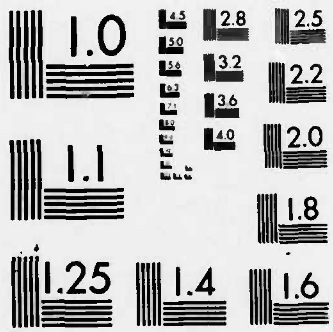
AD-A139 130 PESSHING II SIMULATION STUDIES DEBRIS DYNAMICS AND 275 1/1
ROCKET 6-DOF (DEGR..(U) ALABAMA UNIV IN HUNTSVILLE
SCHOOL OF ENGINEERING P A TILLEY ET AL. JUL 83
UNCLASSIFIED DRSMI/RD-CR-83-20 DAAH01-82-D-A008 F/G 19/4 NL




MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

TECHNICAL REPORT RD-CR-83-20

# PERSHING II SIMULATION STUDIES
# (Debris Dynamics and 2.75 Rocket 6-DOF)

Patrick A. Tilley and Donn A. Hall
School of Engineering
The University of Alabama in Huntsville
Huntsville, AL 35899

July 1983

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898

Approved for public release; distribution unlimited.

## DISPOSITION INSTRUCTIONS



## DISCLAIMER



## TRADE NAMES

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RD-CR-83-20
2. GOVT ACCESSION NO.: AD-A139 130
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): PERSHING II SIMULATION STUDIES (Debris Dynamics and 2.75 Rocket 6-DOF)
5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Patrick A. Tilley and Donn A. Hall
8. CONTRACT OR GRANT NUMBER(s): DAAH01-82-D-A008 Delivery Order 0007
9. PERFORMING ORGANIZATION NAME AND ADDRESS: School of Engineering, The University of Alabama in Huntsville, Huntsville, AL 35899
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: COMMANDER, US Army Missile Command, ATTN: DRSMI-RPT, Redstone Arsenal, AL 35898
12. REPORT DATE: July 1983
13. NUMBER OF PAGES: 84
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): COMMANDER, US Army Missile Command, ATTN: DRSMI-RD, Redstone Arsenal, AL 35898
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):
PERSHING II
Debris Study
MSIXB1 Simulation
2.75 Rocket
6-DOF Simulation
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Pershing II simulation studies were performed using the program MSIXB1 during this task. Analysis and documentation of MSIXB1 and the 2.75 6-DOF simulation were completed for future use in debris studies.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

## PREFACE

This technical report was prepared by the Research Staff of the Electrical Engineering Department, School of Engineering, the The University of Alabama in Huntsville. The purpose of the report is to provide documentation of the technical work performed and results obtained under delivery order 0007 of MICOM Contract No. DAAH01-82-D-A008; Dr. N. A. Kheir, Principal Investigator.

The project documented herein was performed by Patrick A. Tilley and Donn Hall. Dr. M. M. Hallum, III, Chief, Systems Evaluation Branch, Army Missile Laboratory, US Army Missile Command, was technical monitor, and Mr. D. L. Cobb provided technical coordination.

The authors wish to acknowledge the valuable discussions and assistance provided throughout the task by D. L. Cobb and Dr. M. M. Hallum of the Systems Evaluation Branch.

The 2.75 Rocket 6-DOF simulation program, described in Section IV, was developed by H. Lanier, MICOM, and modified by R. A. Dillard, MICOM.

The technical viewpoints, opinions and conclusions expressed in this report are those of the authors and do not necessarily express or imply policies or positions of the US Army Missile Command.

## I. INTRODUCTION

The primary goal of this task was to find impact points of the second stage and adapter of the PERSHING II (PII) missile. This task was necessary because an accurate approximation of major debris impact points was needed for PERSHING II test flights. To accomplish this goal, the MSIXB1 6-DOF Simulation was the primary tool used. It is described in Section II. The results of debris impact studies, using MSIXB1, are then presented in Section III. Additionally, another method of debris evaluation, the 2.75 Rocket 6-DOF simulation, is described in Section IV to facilitate future use in debris studies. Then, in Section V, applicable plotting programs are described.

## II. DEBRIS SIMULATION PROGRAM (MSIXB1)

### A. Introduction

The method chosen to approximate impact points uses MSIXB1 and input missile characteristics form the U-70 6-DOF Simulation of the PII missile. Specifically, an input file containing values such as velocity and position is used by MSIXB1 to find an impact point for each set of inputs. Thrust tables generated for various cutoff times are also used by the program.

The main MSIXB1 program tracks the second stage to impact while an alternate path in the program can track the adapter or any other debris. MSIXB1 finds the impact points by thrusting until cutoff and then letting the debris fall ballistically until impact. Inputs can be varied to supply a footprint of impact points which can be analyzed statistically.

### B. Data Inputs

Data changes are input into MSIXB1 in an efficient manner. The type of data is determined by the first character in each line of data. If this character is numeric, the line of data is interpreted as input data to be stored in an array which will be used by the program. If the first character is alphabetic, it will trigger the program to run, to go into its ballistic phase, or to end execution.

After a line of data is detected an input data, the first number read defines the element number of the input array. Input data is begun after a space is encountered. Two methods can then be used to read the data on the line. The first method records the second number as the value of the element defined by the first number. All of the following numbers are then entered into the data array sequentially. An example is given below with an explanation presented in Table I.

73 15.2 16.3 17.4 3.277 1086.0 -107.32 0.

TABLE I. MSIXB1 SINGLE VARIABLE INPUT SCHEME EXAMPLE

| Data Array Element Number | Value |
|---|---|
| 73 | 15.2 |
| 74 | 16.3 |
| 75 | 17.4 |
| 76 | 3.277 |
| 77 | 1086.0 |
| 78 | -107.32 |
| 79 | 0. |

The second method allows a table of values to be read in from one or more lines in the input data file. The first number is again used as the beginning data array element number. The second number indicates the number of elements in the table of values, while the third number is set to zero in order to align the inputs correctly. The fourth number is the first number in the table of values and is placed in the data array as an element defined by the first number. The remaining values are then entered into the data array in sequential order as above. An example follows with the explanation given in Table II.

400 14 0 5. 77.32 10. 89.15 15. 97.98 20. 123.50 25. 141.34 30. 136.2 35. -17.1

TABLE II. MSIXB1 TABLE INPUT SCHEME EXAMPLE

| Data Array Element Number | Corresponding Array Value |
|---|---|
| 400 | 5. |
| 401 | 77.32 |
| 402 | 10. |
| 403 | 89.15 |
| 404 | 15. |
| 405 | 97.98 |
| 406 | 20. |
| 407 | 123.50 |
| 408 | 25. |
| 409 | 141.34 |
| 410 | 30. |
| 411 | 136.2 |
| 412 | 35. |
| 413 | -17.1 |

The first number in each set of active data represents a time corresponding to the second number. Tables are called in an interpolation routine, and an interpolated value is calculated according to the surrounding times and their values.

The alphabetic data inputs mentioned earlier are available to perform special operations in the program. Presently, only three alphabetic inputs are tapped for use in the program. These are "R" for running, "C" for going into ballistics, and "E" for ending. Operation of the program ceases only when an "E" is encountered. Thus, the data array can be changed at any time, and the program can be rerun using only one input data file. An example is given in Table III.

TABLE III. MSIXB1 ALPHABETIC INPUT SCHEME EXAMPLE

| | |
|---|---|
| 161 5. 10. 12. | data array values |
| R | program runs using above values |
| C | program goes into ballistics after run is completed |
| 161 7. 15. 32. | new data array values |
| 161 10. 25. | data array values replacing part of above line |
| R | program runs using new values 161=10. 162=25. 163=32. |
| E | program ends. |

As shown in the above example, only the final values corresponding to data array elements which precede each "R" are used during the run. This proves to be a very useful tool since an existing input data file can be altered by simply adding a new line of data after the old line. This is convenient since experimentation with new inputs can be made without disturbing functioning inputs. A list of all possible active inputs is given in Tables IV-VI. The English system is used, and time inputs have units of seconds.

C. Aerodynamics

MSIXB1 utilizes aerodynamics based on coordinate transformations using quaternions. The four quaternion parameters are calculated in the subroutine KINE. Quaternions are an analogy to complex numbers, but use three components instead of two. They are used because of their excellent coordinate system rotation properties which reduce the number of calculations required. QAP(1), QAP(2), QAP(3), and QAP(4) are quaternions used to find coefficients for coordinate transformations. These coordinate transformation coefficients are then used in subroutine AERO to properly simulate the missile motion.

The coordinate system used in MSIXB1 is shown in Figure 1. Also shown in Figure 1 is the U70 coordinate system, the other coordinate system used in the PERSHING II study. The x-axis is downrange in both systems, and the following transformations are needed.

TABLE IV. MSIXB1 SINGLE INPUTS

| Array Element Number | Variable | Subroutine |
|---|---|---|
| 1 | X Position (Ft) | KINE |
| 2 | Y Position (Ft) | KINE |
| 3 | -Z Position (Ft) | KINE |
| 4 | Total Velocity (Ft/S) | KINE |
| 5 | Initial Yaw Angle (Deg) | KINE |
| 6 | Initial Pitch Angle (Deg) | KINE |
| 7 | PSI - Yaw Euler Angle (Deg) | KINE |
| 8 | THETA - Pitch Euler Angle (Deg) | KINE |
| 9 | PHI - Roll Euler Angle (Deg) | KINE |
| 10 | Roll Rate (Deg/S) | KINE |
| 11 | Yaw Rate (Deg/S) | KINE |
| 12 | Pitch Rate (Deg/S) | KINE |
| 13 | Distance Between Force & CG in Z Direction (In.) | AERO |
| 14 | Distance Between Force & CG in Y Direction (In.) | AERO |
| 15 | Acceleration of Gravity (Ft/$S^2$) | MAIN, AERO |
| 19 | Weight (Lbs) | AERO |
| 20 | CG Mass Property Table Number | AERO |
| 21 | X Mass Property Table Number | AERO |
| 22 | Y Mass Property Table Number | AERO |
| 23 | Moment arm used to calculate Variable M2 | AERO |
| 25 | Print Option for Metrics, Etc. | MAIN |
| 26 | Total Body Rate Limit 0 = No Limit | MAIN |
| 27 | Print Option for Delta Values | MAIN |
| 28 | Roll Rate Limit | MAIN |
| 29 | QR Rate Limit | MAIN |
| 30 | Iteration Time | MAIN |
| 31 | Print Iteration Interval | MAIN |
| 32 | Alternate Print Iteration for TIME.GT. A(205) | MAIN |
| 33 | Time Limit For Run | MAIN |
| 34 | Seconds After Separation That Thrust Reversers Are Activated | AERO |
| 38 | Distance Between Center of Gravity And Thrust Force to Calculate Variable M3 | AERO |
| 40 | Print Option for Deltas | MAIN |
| 41 | Variable Delta Option Time | MAIN |
| 42 | X Velocity Vector Added to Debris After Time A(49) | MAIN |
| 43 | Y Velocity Vector Added to Debris After Time A(49) | MAIN |
| 44 | Z Velocity Vector Added to Debris After Time A(49) | MAIN |

TABLE IV (cont'd). MSIXB1 SINGLE INPUTS

| Array Element Number | Variable | Subroutine |
|---|---|---|
| 45 | Ballistic Coefficient of Debris | BALLST |
| 46 | Iteration Time For Ballistic Subroutine | BALLST |
| 47 | Print Iteration For Ballistic Subroutine | BALLST |
| 49 | Time After Separtion When Secondary Debris is Tracked | MAIN |
| 50 | IN-LB/PSI Coefficient For Roll | AERO |
| 51 | IN-LB/PSI Coefficient For Pitch-Yaw | AERO |
| 52 | Pitch Yaw Phi Angle | AERO |
| 90 | Print Start Time | MAIN |
| 101 | X Offset Distance Between A(1) & Start Point | MAIN |
| 102 | Y Offset Distance Between A(2) & Liftoff Point | MAIN |
| 160 | Weight From Which Thrust Ratio is Calculated | AERO |
| 161 | Moment 1 Input | AERO |
| 162 | Moment 2 Input | AERO |
| 163 | Moment 3 Input | AERO |
| 164 | Upper Time Limit At Which Moment 2 Equals 0 | AERO |
| 165 | Lower Time Limit At Which Moment 2 Equals 0 | AERO |
| 205 | Time Which Changes Print Iteration to A(32) | MAIN |

TABLE V. MSIXB1 INPUTS IN AERO

| Table Array Start Number | Table Subject |
|---|---|
| 305 | Delta Pitch Attack Angles |
| 320 | Delta Yaw Attack Angles |
| 500 | Forward Thrust Before TR (Thrust Reversal) |
| 550 | Reverse Thrust Before TR |
| 600 | Rate of Weight Change Before TR |
| 1500 | Forward Thrust After TR |
| 1550 | Reverse Thrust After TR |
| 1600 | Rate of Weight Change After TR |
| 1700 | Thrust of Dome Reverser Ports |
| 2600 | CG Mass Property |
| 2620 | X Mass Property |
| 2640 | Y Mass Property |

TABLE VI. MSIXB1 ALPHABETIC INPUTS

| Letter | Command |
|---|---|
| C | Calls BALLST After Run |
| E | Ends Run |
| R | Runs Program |

x = x, y - z, y = -y

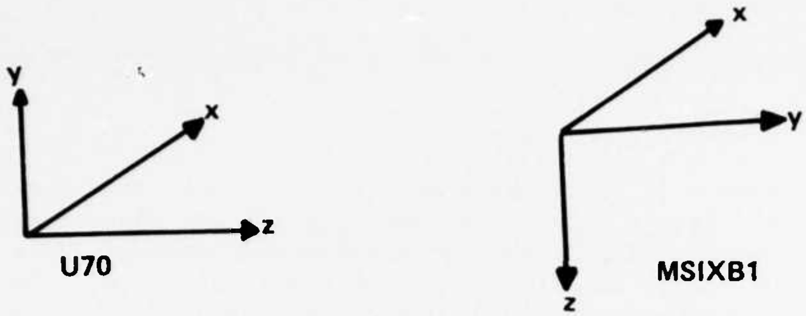



Figure 1. U70 and MSIXB1 coordinate systems.

However, it should be noted that the negative sign for z is taken into account when the z value is entered into the program as -A(3).

D. Ballistic Phase

New subroutines, BALLST and ATMO, were developed and added to the basic MSIX program to permit simulation of missile debris. The only forces acting upon debris are gravity and drag. The weight of the debris is taken into account by the ballistic coefficient "BC," which is weight divided by drag coefficient. Since the debris begins falling from a point outside the atmosphere, ATMO has been included to supply the changing density of the air which affects drag. Time and print iteration intervals, specifically for the ballistic phase, are included as inputs A(46) and A(47). Iterations continue until the z component of the debris' position is greater than zero. This impact point is then printed in both English and metric coordinates, and the run is terminated.

E. Program Operation

A complete set of MSIXB1 files consists of a CSS file, a FTN (fortran) file, and TSK (task) file. Alterations to the program are made in the FTN file. This FTN file is compiled into the machine code which comprises the TSK file. The CSS file is a file which contains instructions to run the TSK file. This CSS file is initiated by simply typing MSIXB1, the name of data file to be used, and return. The CSS program is presented in Figure 2.

```
L   MSIXB1
AS  1,CON:
AS  3,NULL:
AS  5,@1.DAT
AS  6,CON:
AS  7,CON:
AS  8,CON:
AS  9,CON:
AS  4,CON:
ST
$EXIT
```

Figure 2. MSIXB1 CSS file.

The data file name should have the extension .DAT to function properly. A flow chart of the FTN file is given in Appendix A, and the FTN file itself is presented in Appendix B.

## III. DEBRIS IMPACT STUDIES

### A. Introduction

Three pieces of debris have been studied thus far. These are the second stage, the adapter between the second stage and the re-entry vehicle, and the dome closures covering the thrust reversal stacks. The studies used several different separation times for each piece of debris. Thus, several sets of initial conditions had to be used in the data files. Also, various initial velocity vectors were added to yield a circle of impact data. Input A(49) controls the time at which the secondary debris begins falling. The value that is used to describe the weight and drag properties of the object is the ballistic coefficient. This value is defines as weight divided by the drag coefficient.

### B. Adapter Study

A ballistic coefficient of 8.8 was used for the adapter. Four sets of added initial velocity vectors were input, and a run was made for each set at three different separation times. The vectors were all combinations of positive and negative vectors added in z and y directions. Several different impulse moments were used on the missile so that a circular error pattern could be found for each set along with an impact footprint. The adapter impacted at shorter range and had a smaller cross-range deviation than did the second stage. Thus, the debris stopped short in downrange causing a smaller deviation in the y-direction.

### C. Port Closure Study

Two different weight port closures were studied which had ballistic coefficients of 7.4 and 22.2. Chamber prelsure of the second stage was analyzed to yield the average and limiting values of added initial velocity vectors. Twelve runs were made for each of eight separation times to fully evaluate the debris pattern of each piece of debris. Plots were made showing the characteristic impact ellipse for each debris mass, with the boundary of the ellipse being determined by the limiting initial added velocities.

### D. Second Stage Impact Study

The second stage study was similar to the secondary debris studies except that forces continued to act on the body even after thrust cutoff. These thrusts occur due to the release of pressure through the thrust reversal ports. The program simulated motion until all thrusts were zero. Then ballistic and atmospheric models permit simulation from this cutoff point until impact.

Runs using fifty random moments were made for the second stage with various velocity vectors added. A circular error pattern routine, which found 99% and 50% error circles, was used to analyze this data.

## IV. 2.75 ROCKET 6-DOF SIMULATION PROGRAM

### A. Introduction

The 2.75 Rocket is a multi-mode rocket. It can function as either a surface-to-surface or air-to-surface rocket. Additionally, it can be fired from either fixed wing aircraft or from helicopters. Its simulation is based on the most difficult of the three launch platforms, the helicopter. The simulation is versatile enough, though, to easily adapt to the other platforms. And this versatility also makes the simulation useful for debris study.

### B. Algorithm

The algorithm for the simulation is common although some of the processes are uncommon. A flowchart and complete program listing can be found in Appendix C and D, respectively.

The first step of the algorithm is to predefine the values of all variables at launch. A "cycle" begins when the rocket is launched and starts with an initial update of variables. Added to these variables are first the effects of wind and rotor downwash, if applicable, and then the corrections for aerodynamic and atmospheric conditions. From these values and the previous values, derivatives of the variables (Table VII) are defined and calculated. Finally, a Runge-Kutta integration routine is used to integrate the derivatives (Table VIII), thereby completing the cycle.

### C. Coordinate System

There are three coordinate systems to be considered for this simulation. These are centered on (1) the launch platform, (2) the rocket, and (3) the ground. All three have the same conventions and are shown in Figure 3. It should be noted that altitude is positive downward.

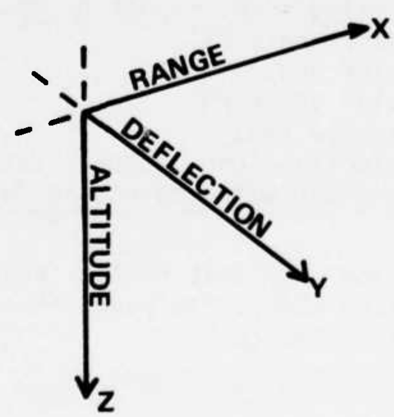



Figure 3. 2.75 uniform coordinate systems.

Table VII is a list of the derivative array and the derivatives' definitions. Table VIII is a list of the variables that are the result of integrating the derivatives array and their definitions.

TABLE VII. 2.75 DERIVATIVE ARRAY

| | |
|---|---|
| XX(1,1) | - Angular acceleration about the missile X-Axis |
| XX(1,2) | - Angular acceleration about the missile Y-Axis |
| XX(1,3) | - Angular acceleration about the missile Z-Axis |
| XX(1,4) | - Linear acceleration along the missile X-Axis |
| XX(1,5) | - Linear acceleration along the missile Y-Axis |
| XX(1,6) | - Linear acceleration along the missile Z-Axis |
| XX(1,7) | - Linear velocity along the ground X(range)-Axis |
| XX(1,8) | - Linear velocity along the ground Y(deflection)-Axis |
| XX(1,9) | - Linear velocity along the ground Z(vertical)- Axis |
| XX(1,10) | - Derivative of the Euler angle PSI |
| XX(1,11) | - Derivative of the Euler angle THETA |
| XX(1,12) | - Derivative of the Euler angle PHI |
| XX(1,13) | - Derivative of the mass (time rate of change of mass) |
| XX(1,14) | - Acceleration along launcher center line |
| XX(1,15) | - Velocity along launcher centerline |

TABLE VIII. 2.75 STATE VARIABLE ARRAY

| | |
|---|---|
| XX(2,1) | - Angular velocity about missile X-Axis (Roll Rate) |
| XX(2,2) | - Angular velocity about missile Y-Axis (Pitch Rate) |
| XX(2,3) | - Angular velocity about missile Z-Axis (Yaw Rate) |
| XX(2,4) | - Linear velocity along missile X-Axis |
| XX(2,5) | - Linear velocity along missile Y-Axis |
| XX(2,6) | - Linear velocity along missile Z-Axis |
| XX(2,7) | - Position with respect to ground X-Axis (Range) |
| XX(2,8) | - Position with respect to ground Y-Axis (Deflection) |
| XX(2,9) | - Position with respect to ground Z-Axis (Altitude) |
| XX(2,10) | - Euler angle PSI |
| XX(2,11) | - Euler angle THETA |
| XX(2,12) | - Euler angle PHI |
| XX(2,13) | - Missile mass |
| XX(2,14) | - Velocity along launcher centerline |
| XX(2,15) | - Position with respect to launcher |

It is important to note that XX(1,1) does not go through the integration routine. The roll rate XX(2,1) is found through the use of a look-up table based on the time since launch.

In the helicopter system, all of the rocket's launch conditions are calculated from the position of the helicopter. Important angles in the helicopter system are illustrated in Figures 4 and 5. More information on the helicopter as a launch platform can be obtained from Reference [1].

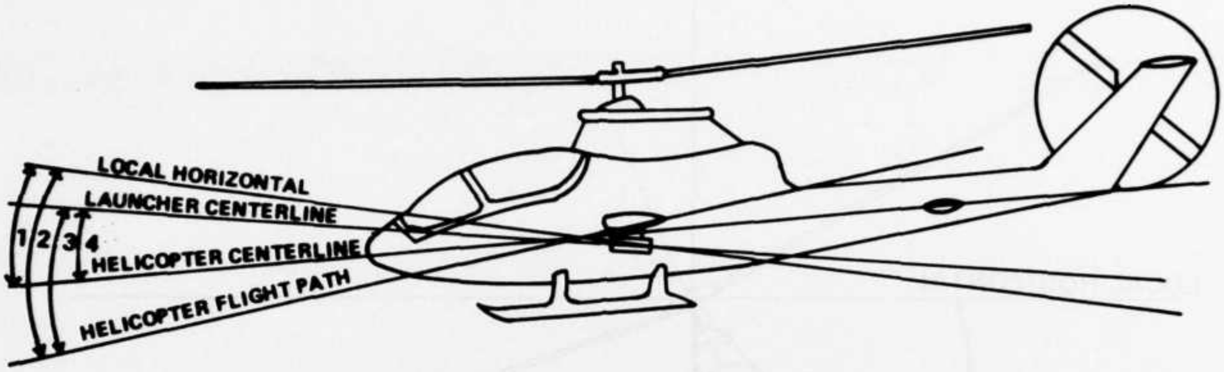



Figure 4. AH-1G helicopter angles (side view).

The numbered angles in Figure 4 are defined as follows:

1. Total horizontal angle of attack
2. Dive angle
3. Helicopter Altitude
4. Quadrant elevation of launcher

The missile coordinate system, illustrated in Figures 6 and 7, is the system in which all of the flight characteristics of the missile and all of the external forces on the missile are calculated. Figure 5 illustrates the rocket with respect to the launcher tube shortly after launch. Figure 7 illustrates the basic body angles and velocity vectors of the rocket.

The ground coordinate system (Figure 8) is based on the position of the target. Through use of this system, the inertial position of the rocket is determined as well as the position of impact. The simulation assumes a flat earth and a uniform gravitational field.

Transformation between the helicopter and missile systems is performed simply by adding and subtracting predefined distances based on parameters existing at the time of calculation. The transformation between the rocket and the earth-fixed coordinate systems is more difficult, and the process used in this simulation is rather unusual.

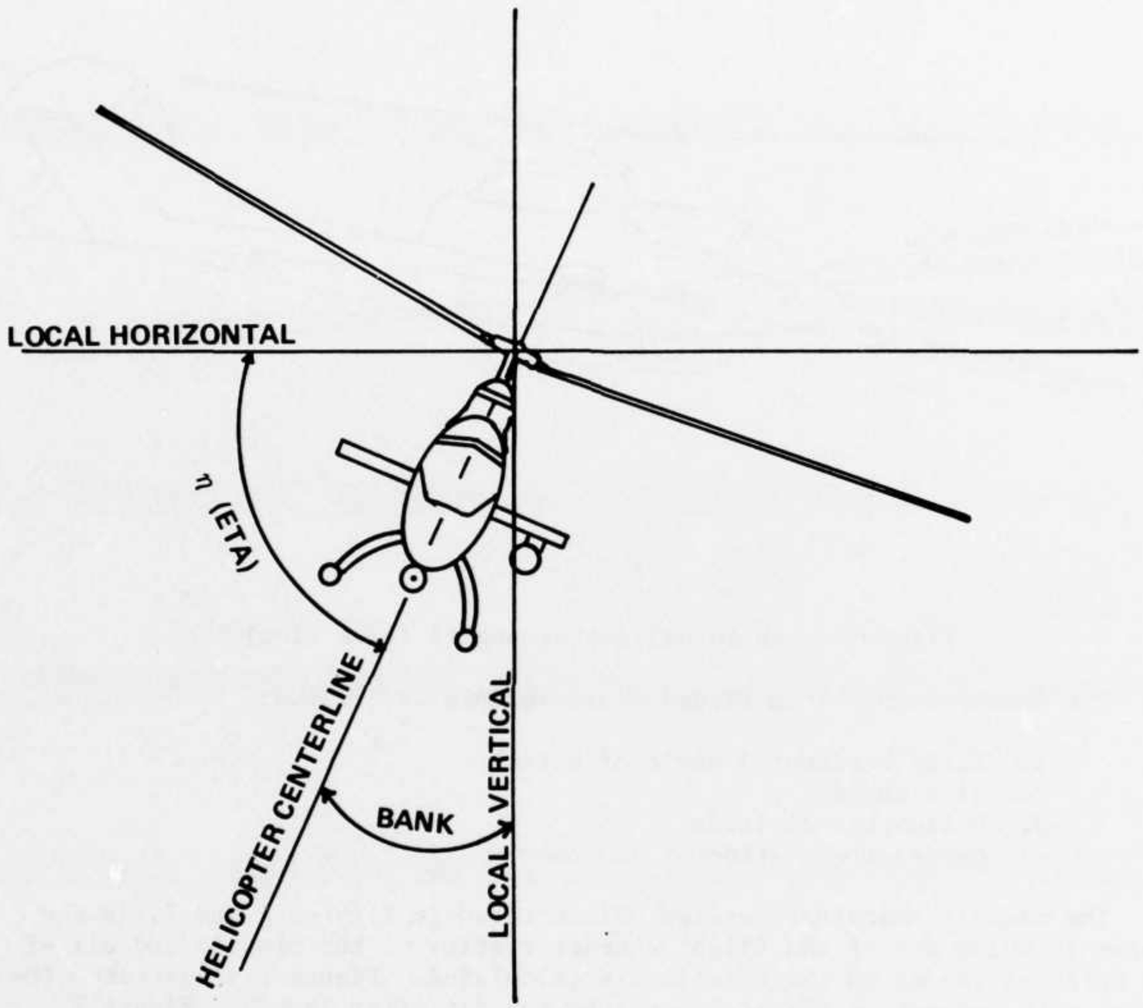



Figure 5. AH-1G helicopter angles (front view).

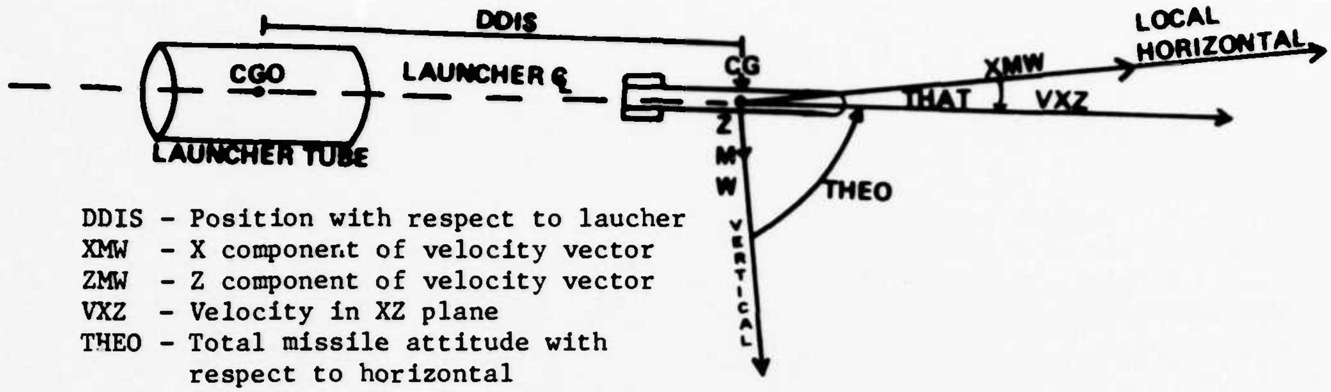



Figure 6. 2.75 Rocket coordinate system with respect to launcher.

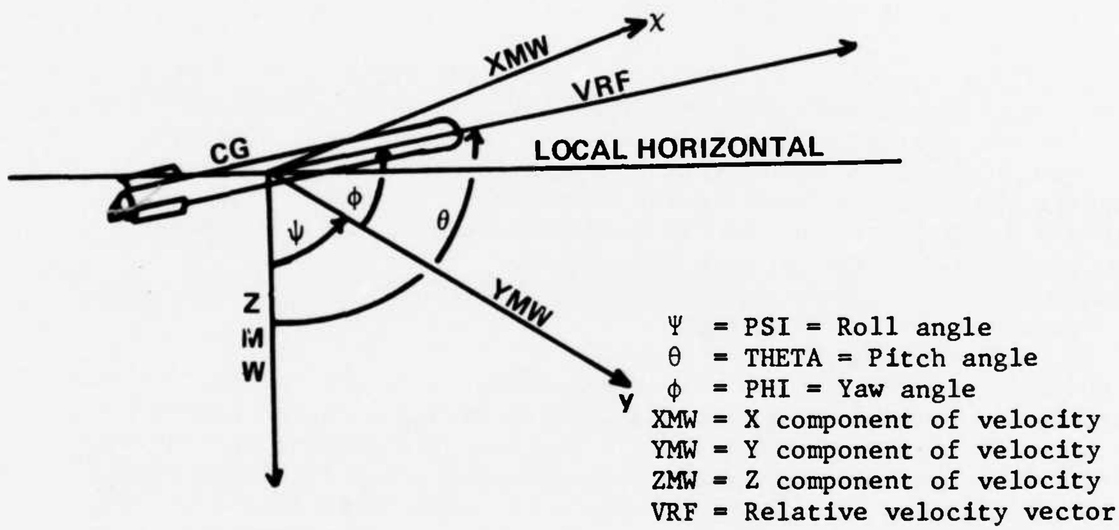



Figure 7. 2.75 Rocket coordinate system body angles and velocity vectors.

XMXH = X component of inertial position
YMYH = Y component of inertial position
ZMZH = Z component of inertial position
VRF  = Relative velocity vector

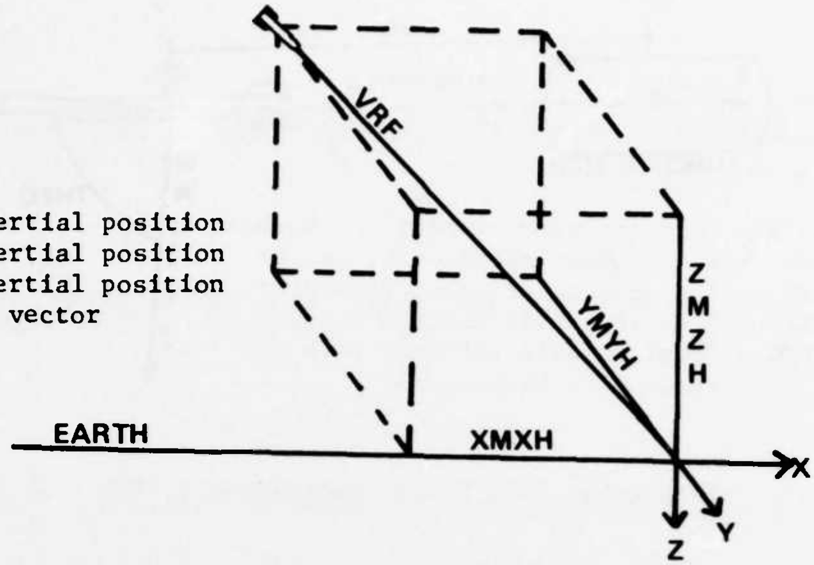



Figure 8. 2.75 Rocket earth fixed coordinate system.

Usually, the derivatives of the Euler angles are repeatedly integrated to update the coordinate transformation matrix (Data Conversion Matrix-DCM) in terms of sines and cosines of the Euler angles. However, this simulation digressed from that method. Integration of the Euler angle derivatives is used to update the Euler derivatives via an altered DCM array. Thus, the body rates are used to refresh the DCM matrix instead of the basic Euler angles, eliminating one step in each calculation.

D. Downwash Profile Description

During the first few seconds of the rocket's flight, after launched from a helicopter, several transients associated with the launch platform become involved. These transients include vibration, translation and rotation of launch platform and rotor downwash. As soon as the rocket clears the launcher tube, the first two of these transients cease to be involved. The downwash affects the rocket until the time that the rocket clears the rotor blades, and the effect changes with respect to helicopter velocity. These effects have been studied at MICOM for many years and are modeled in the program through subroutine HELVEL. It uses a data pack containing the downwash characteristics of an AH-1G helicopter. Refer to Reference [1] for more information on these transients and their effects.

E. Program Variables

The purpose of this section is to define the three types of variables in the 2.75 digital simulation. They are input, computed, and output variables. Due to the number of variables and the uncertainty of some of their definitions, there is a possibility that a complete listing is not presented in this section. However, all efforts have been made to ensure that the definitions given are correct. Any variable that is given in parenthesis without a definition is done so because a definition is not known at this time. All variable lists are, for the most part, in the order that they appear within the program.

1. Input Variables

Table IX is a list of the variables input to the 2.75 digital simulation with their definitions. (NAMELIST/PUTT/)

TABLE IX. 2.75 INPUT VARIABLES

REALS

ALTIN - Initial missile altitude
DLETT - Delta time (step size) for integration routine
DELO2 - Delay time
DIVANG - Angle between helicopter velocity vector VINIT and local horizontal
DNSCH - Dynamic pressure scaler
DPIT - Offset in pitch of motor nozzle
DT - Delta time for integration routine after TDC
DYAW - Offset in yaw of motor nozzle
DZZ - Z component of downwash force
EOL - End of launch flag
FACTOR - Variable associated with motor nozzle (ft.)
FCA - Forward cant angle of helicopter
FTORM - Output format flag (ft. or meters)
GREFF - Force of gravity (ft./sec$^2$)
HELIC - Flag for use of downwash profile (1.-use,0.-do not use)
PITMAL - Pitch mallaunch rate
QELNCH - Launcher elevation angle with respect to helicopter centerline
REF - Radius of the earth
RK - Flag for type of Runge-Kutta integration routine (2-2nd order,4-4th order).
ROTRAD - Radius of helicopter rotor blades
SCARF - Variable associated with motor nozzle (ft.)
TDC - Time that integration delta time DELTT changes to DT.
TFINAL - Maximum time of flight
TIME - Program counter
TLL - Time missile left launcher
TPRINT - Time increment for printing results
TROT - Time of rotation (sec.)
VINIT - Intial helicopter velocity vector's magnitude (airspeed ft./sec.)
WCF - Crossrange wind force
WDF - Downrange wind force
XMV-X - Component of missile velocity at launch (ft./sec.)
XTAR - Range of target (ft. Earth fixed coordinate system)
YAWMAL - Yaw mallaunch rate
YTAR - Crossrange of target (ft. Earth fixed coordinate system)
ZTAR - Altitude of target (ft. Earth fixed coordinate system)

INTERGERS

N- Integration routine upper limit
NM - Integration routine lower limit

2. Computed Variables

Table X is a list of variables computed in the 2.75 digital simulation.

TABLE X. 2.75 COMPUTED VARIABLES

Logicals

BURNOUT - Motor has burned out
EOL - [END OF LAUNCH] Trajectory begins with parameters at time rocket clears tube
IMPACT - Impact occurred
LNEXIT - Rocket clear of launcher
LRATE - Rocket CG has cleared launcher
NOTIME - Runtime exceeds allowed TFINAL
SETBO - Set rocket parameters to burnout values
SKIP - Overrides SETBO

Reals

IXO - Initial axial moment of inertia (slugs/ft$^2$)
IXF - Final axial moment of inertia (slugs/ft$^2$)
IYO - Initial transverse moment of inertia (slugs/ft$^2$)
IYF - Final transverse moment of inertia (slugs/ft$^2$)
IX - Axial moment of inertia
IY - Transverse moment of inertia
IZ - Vertical moment of inertia
IXDOT - Time rate of change of axial moment of inertia (slugs.sec/ft$^2$)
IYDOT - Time rate of change of transverse moment of inertia (slugs.sec/ft$^2$)
IZDOT - Time rate of change of vertical moment of inertia (slugs.sec/ft$^2$)
ISP - Specific impulse
MCDOFF (21) - Mach number table for coefficients of DRAG w/power off
MCDON (21) - Mach number table for coefficients of DRAG w/power on
MCMQ (18) - Mach number table for pitch damping coefficients
MCNA (7) - Mach number table for normal force
MCP (9) - Mach number table for center of pressure
MDOT - Mass flow rate as a function of thrust
ACTI (3,3) - Inverse of DCM Matrix
CDPOFF (21) - Coefficients of DRAG with power off (function of Mach data)
CDPON (21) - Coefficients of DRAG with power on (function of Mach data)
CMQT (17) - Coefficients of pitch damping (rad/sec - function of mach data)
CNAT (7) - Slope coefficient of normal force (f/deg. - function of mach data)
CPT (8,2) - Center of pressure (calibers from nose - function of mach data and angle of attack)
CPAF (2) - Angle of attack table (degrees)
CK (4, 50) Calculated integrating constants
LABEL (8) - Input of trajectory characteristics
IDAY (3) - Date (internal to computer)
DCM (3,3) - Euler angle transformation matrix
DTDH (12) - Atmospheric density reference table
HGEOPB (13) - Atmospheric altitude reference table

TABLE X. 2.75 COMPUTED VARIABLES (Cont'd)

PRESB (12) - Atmospheric pressure reference table
SPINT (30) - Time breakpoints for SPNRAT (table)
SPNRAT (30) - Average roll rate (rev/sec - table)
TEMPMB (12) - Atmospheric temperature reference tables
THRUST (9) - Thrust table (pounds)
TTIME (9) - Time table for thrust
WUVW (3,3) - Result of DCM (3,3) * WXYZ (3,3) (Effect of wind on rocket body)
WXYZ (3,3) - Coefficients of wind
XX (3,50) - XX (1,50) Derivatives of rocket trajectory parameters
XX (2,50) State of rocket trajectory parameters
XX (3,50) Not used
YY (3,50) - YY (1,50) Derivative at beginning of time step
YY (2,50) State at beginning of time step
YY (3,50) Not used
CLP - Roll damping coefficient
CG - Instantaneous center of gravity (ft. from tail)
CGO - Initial center of gravity (ft. from tail)
CGF - Final center of gravity (ft. from tail)
DREF - Reference diameter of rocket (ft)
F - Air density constant
PI - Variable equivalent to PI = 3.14159
RADCON - Degrees per radian = 57.2957795
S - Surface area of earth
SUSOFF - Motor burn time
TI - Total impulse
WO - Initial rocket weight
WF - Final rocket weight
WL - Weight when rocket clears launcher
XLENG - Missile length
CGDOT - Time rate of change of center of gravity
XMASS - Initial mass of rocket
QEMILS - Quadrant elevation (mils)
AIRSP - Airspeed (knots)
HELAT - Helicopter attitude (degrees)
AREFF - Reference area of rocket ($ft^2$)
THL - Angle between launcher centerline and local horizontal
ALV - Angle between launcher centerline and helicopter velocity vector VINIT
TTO - Time till thrust equals zero
TEXIT - Time missile exits launcher
DXLD - X - Coordinate of launcher CG with respect to the helicopter rotor hub
DYLN - Y - Coordinate of launcher CG with respect to the helicopter rotor hub
DZLN - Z - Coordinate of launcher CG with respect to the helicopter rotor hub
DXX - X - Components of downwash coefficient
DXY - Y - Components of downwash coefficient
DZZ - Z - Components of downwash coefficient
SI - Euler Angle PSI
TH - Euler Angle THETA
FI - Euler Angle PHI

TABLE X. 2.75 COMPUTED VARIABLES (Cont'd)

SSI - SIN (PSI)
CSI - COS (PSI)
STH - SIN (THETA)
CTH - COS (THETA)
SFI - SIN (PHI)
CFI - COS (PHI)
P - Angular acceleration about rocket Z-axis
UDOT - Linear acceleration along rocket X-axis
VDOT - Linear acceleration along rocket Y-axis
WDOT - Linear acceleration along rocket Z-axis
CMQ - Rate Damping coefficient for rates about rocket
AUX - Variable equal to $\frac{\text{(Dynamic Pressure * rocket diameter squared)}}{\text{2 * Relative Velocity}}$
XMP1AS - Aerodynamic torques about rocket X-axis
XMQ1AS - Aerodynamic torques about rocket Y-axis
XMR1AS - Aerodynamic torques about rocket Z-axis
CP - Center of pressure as a function of Mach No. and Angle of Attack
AMA - Aerodynamic moment arm
SM - Stability margin (calibers)
XMAY - Aerodynamic restoring torques about rocket Y-axis
XMAZ - Aerodynamic restoring torques about rocket Z-axis
TAX - Torque about rocket X-axis due to angular velocities
TAY - Torque about rocket Y-axis due to angular velocities
TAZ - Torque about rocket Z-axis due to angular velocities
TIX - Torque about rocket X-axis due to time rate of change of inertia
TIY - Torque about rocket Y-axis due to time rate of change of inertia
TIZ - Torque about rocket Z-axis due to time rate of change of inertia
XMPIGS - Torque about rocket X-axis due to components of thrust
ARM - Length of moment arm
XMQ1GS - Torque about rocket Y-axis due to components of thrust
XMR1GS - Torque about rocket Z-axis due to components of thrust
TORQX - Total torque about rocket X-axis
TORQY - Total torque about rocket Y-axis
TORQZ - Total torque about rocket Z-axis
ROLL - Roll rate (hz)
GAX - Gravitational acceleration along rocket X-axis
GAY - Gravitational acceleration along rocket Y-axis
GAZ - Gravitational acceleration along rocket Z-axis
UDOTL - Acceleration rocket along X-AXIS due to drag, thrust + gravity only
UL - Velocity of rocket along launcher centerline
REZIF - Rocket altitude from center of earth
RF - Distance from center of earth to launcher
HEIGHT - Distance from earth's crust to launcher
SQIRT - Horizontal distance from launcher to rocket
DYNAR - Dynamic pressure times reference area (ft)
DZRFF - Reference diameter (ft)
TANG - Cosine of trajectory angle from center of earth
RGF - Gravitation reference height (ft)
WIND - Magnitude of wind force
TPHASE - (TIME + TROT = .187)
XH - Instantaneous distance from rotor hub to rocket along X-axis

TABLE X. 2.75 COMPUTED VARIABLES (Cont'd)

YH - Instantaneous distance from rotor hub to rocket along Y-axis
ZH - Instantaneous distance from rotor hub to rocket along Z-axis
XMXH - Initial position of rocket with respect to rotor hub along X-axis
YMYH - Initial position of rocket with respect to rotor hub along Y-axis
ZMZH - Initial position of rocket with respect to rotor hub along Z-axis
DDIS - Vertical position of rocket with respect to launcher
THAT - Total Horizontal Attitude of Helicopter (deg)
THEO - Downrange rocket trajectory with respect to vertical (deg)
THTH - Downrange rocket trajectory with respect to horizontal (deg)
XD - Horizontal position of rocket with respect to rotor hub
YD - Deflection position of rocket with respect to rotor hub
ZD - Vertical position of rocket with respect to rotor hub
DDISS - Horizontal distance from rocket to rotor hub
WXYZ (1,1) - Magnitude of wind X-axis vector
WXYZ (2,1) - Magnitude of wind Y-axis vector
WXYZ (3,1) - Magnitude of wind Z-axis vector
HGTIND - Height indicator
HGEOP - Atmospheric altitude
WMOL - Molecular weight
TEMPMD - Instantaneous observed atmospheric temperature
HGEOPO - Instantaneous observed atmospheric altitude
DODH - Instantaneous observed atmospheric density
PRESO - Instantaneous observed atmospheric pressure
TEMPM - Median temperature (molecular temperature)
TEMP - Actual temperature
VSOUND - Velocity of sound (function of altitude, temperature, & air denstiy)
PRES - Air pressure
XMW - Apparent velocity of rocket in X vector (m-wind)
YMW - Apparent velocity of rocket in Y vector
ZMW - Apparent velocity of rocket in Z vector
VVW - Vertical velocity (squared)
VRF - Relative velocity of rocket (ft/s)
XMACHN - Relative velocity (MACHS)
VALF - Slope of velocity vector
WUVW (1,1) - Wind X vector on rocket
WUVW (2,1) - Wind Y vector on rocket
WUVW (3,1) - Wind Z vector on rocket
ALFTOF - Total angle of flight
BANK - Angle of heliocopter body with respect to vertical
ALPPRN - Vertical body angle with respect to downrange
BETPRN - Horizontal body angle with respect to downrange
CN - Normal force coefficient
CY - Side force coefficient
CA - Axial force coefficient
THRSTP - Magnitude of thrust vector
DP - Dynamic pressure
DPS - Dynamic pressure on surface of missile
FU1AP - Aerodynamic force along rocket X-axis
FV1AP - Aerodynamic force along rocket Y-axis
FW1AP - Aerodynamic force along rocket Z-axis

TABLE X. 2.75 COMPUTED VARIABLES (Cont'd)

FU1GP - Thrust force along rocket X-axis
FV1GP - Thrust force along rocket Y-axis
FW1GP - Thrust force along rocket Z-axis
UMF1 - Linear acceleration along rocket X-axis
VMF1 - Linear acceleration along rocket Y-axis
WMF1 - Linear acceleration along rocket Z-axis
U - Linear velocity along rocket X-axis
V - Linear velocity along rocket Y-axis
W - Linear velocity along rocket Z-axis
P - Angular velocity about rocket X-axis
Q - Angular velocity about rocket Y-axis
R - Angular velocity about rocket Z-axis
PDOT - Angular acceleration about X-axis
QDOT - Angular acceleration about Y-axis
RDOT - Angular acceleration about Z-axis
PNOM - Length of rocket vertical velocity vector
TWOPI - Two times the radian angle PI (Circumference of unit circle)
PHII - Angle PHI
SNPHI - Sine of PHII
CMPHI - Cosine of PHII
ETAPRN = Angle between horizontal and bank vector
ALPHPR - Angle Alpha (Rad) - Angle between Z and X velocity vectors
BETAPR - Angle Beta (Rad) - Angle bewteen Y and X velocity vectors
SIDOT - Derivative of PSI
THDOT - Derivative of THETA
FIDOT - Derivative of PHI
TPR - Time till print (Run time [TIME - TPRINT] Time of period)
ALPTD - Angle Alpha in degrees
BETTD - Angle Beta in degrees
ETATD - Angle Eta in degrees
ALF - Angle Alpha in radians
BETO - Angle Beta in radians
TP1 - Time (counter 1)
XP1 - Rocket downrange position (time 1)
YP1 - Rocket cross-range position (time 1)
ZP1 - Rocket altitude position (time 1)
XP2 - Rocket downrange impact
YP2 - Rocket cross-range impact
ZP2 - Rocket altitude (impact - ground coordinate system)
TP2 - Time rocket impact
XMP - Impact range
TMP - Time of impact
PID - Roll rate (rev/sec) at impact
WINX - Velocity along X-axis including wind
WINY - Velocity along Y-axis including wind
WINZ - Velocity along Z-axis including wind

TABLE X. 2.75 COMPUTED VARIABLES (Cont'd)

| |
|---|
| ALFO - Vertical angle of impact |
| QID - Pitch at impact (Radians) |
| RID - YAW at impact (Radians) |
| DIST - Distance to go (rocket to target) |
| PID1 - Angular acceleration about rocket X-axis (degrees) |
| Q1D1 - Angular acceleration about rocket Y-axis (degrees) |
| R1D1 - Angular acceleration about rocket Z-axis (degrees) |
| KN - Integration counter |
| CK (1,4) - Integration coefficient array |

3. Outputs

There are two output sources available to the user of the 2.75 digital simulation. The first is output by the program via the line printer. The second is output to a local file for use with the plotting routine associated with this program.

(b) Printed Variables

There are three sets of outputs to the line printer. The first is the set of all initial conditions for the simulation. The second set is output after the value of counter exceeds the value of a print flag. The last set is output after impact or when the simulation runs past the maximum allowable time. A listing of these variables and their definitions, in the order they are output, follows in Table XI.

TABLE XI. 2.75 PRINTED OUTPUT VARIABLES

| |
|---|
| TIME - Time of Flight |
| RANGE - Downrange |
| DEFL - Crossrange |
| ALT - Altitude |
| VELR - Relative Velocity (Magnitude) |
| THRUST - Magnitude of Thrust |
| XDOT - Linear Velocity Along Ground X(Range)-Axis |
| YDOT - Linear Velocity Along Ground Y(Deflection)-Axis |
| ZDOT - Linear Velocity Along Ground Z(Vertical)-Axis |
| MACH - Mach Speed |
| MASS - Mass of Rocket |
| UUU - Total Linear Velocity Along Rocket X-Axis |
| VVV - Total Linear Velocity Along Rocket Y-Axis |
| WWW - Total Linear Velocity Along Rocket Z-Axis |
| MDOT - Time Rate of Change of Rocket Mass |
| (SM) - |
| UDOT - Total Linear Acceleration Along Rocket X-Axis |
| VDOT - Total Linear Acceleration Along Rocket Y-Axis |
| WDOT - Total Linear Acceleration Along Rocket Z-Axis |
| DP - Dynamic Pressure |
| RHO - Air Density |

TABLE XI. 2.75 PRINTED OUTPUT VARIABLES (Cont'd)

| |
|---|
| P - (Roll Rate) Angular Velocity About Rocket X-Axis |
| Q - (Pitch Rate) Angular Velocity About Rocket Y-Axis |
| R - (Yaw Rate) Angular Velocity About Rocket Z-Axis |
| CA - Axial Force Coefficient |
| ALFTOT - Total Angle of Flight |
| PID1 - Angular Acceleration About Rocket X-Axis (Degrees) |
| QID1 - Angular Acceleration About Rocket Y-Axis (Degrees) |
| RID1 - Angular Acceleration About Rocket Z-Axis (Degrees) |
| CN - Vertical Coefficient of Force |
| ALPTD - Angle Alpha (Degrees) |
| TORQX - Total Torque About Rocket X-Axis |
| TORQY - Total Torque About Rocket Y-Axis |
| TORQZ - Total Torque About Rocket Z-Axis |
| CY - Side Force Coefficient |
| BETTD - Angle Beta (Degrees) |
| WINX - Velocity ALong X-Axis Including Effect of Wind |
| WINY - Velocity Along Y-Axis Including Effect of Wind |
| WINZ - Velocity Along Z-Axis Including Effect of Wind |
| LNEXIT - Logical Representing Rocket Clear Launcher |
| XMQ1AS - Aerodynamic Torque About Rocket Y-Axis |
| XMAY - Aerodynamic Restoring Torque About Rocket Y-Axis |
| TAY - Torque About Rocket Y-Axis due to Angular Velocities |
| TIY - Torque About Rocket Y-Axis due to Time Rate of Change of Inertia |
| CMQ - Rate Damping Coefficient for Rates About Rocket Axes |
| XMR1AS - Aerodynamic Torque About Rocket Z-Axis |
| XMAZ - Aerodynamic Restoring Torque About Rocket Z-Axis |
| TAZ - Torque About Rocket Z-Axis due to Angular velocities |
| TIZ - Torque About Rocket Z-Axis due to Time Rate of Change of Inertia |
| BANK - Angle Between Positive Vertical and Helicopter Vertical Centering |
| PSI - Euler Angle psi |
| THETA - Euler Angle theta |
| PHI - Euler Angle phi |
| UMF1 - Linear Acceleration Along Rocket X-Axis |
| VMF1 - Linear Acceleration Along Rocket Y-Axis |
| WMF1 - Linear Acceleration Along Rocket Z-Axis |
| GAX - Gravitational Acceleration Along Rocket X-Axis |
| GAY - Gravitational Acceleration Along Rocket Y-Axis |
| GAZ - Gravitational Acceleration Along Rocket Z-Axis |
| CGTRVL - Distance Center of Gravity of Rocket has Travelled |

(b) Plot Tape Variables

The following is a list of the 2.75 variables that are written to the plot tape from the 6-DOF simulation. These are reproduced in the order that they are stored on record.

TABLE XII. 2.75 PLOT TAPE VARIABLES

| |
|---|
| Time of Flight (Sec.) |
| Range |
| Deflection - Positive Right |
| Altitude - Positive Down |
| Relative Velocity |
| Mass (Slugs) |
| Range Velocity |
| Cross Range Velocity - Postivie Right |
| Vertical Velocity - Positive Down |
| Mach Number |
| Thrust |
| Roll Rate (Rev/Sec) |
| Pitch Rate (Deg/Sec) |
| Yaw Rate (Deg/Sec) |
| Dynamic Pressure |
| Range to go to Target |
| Roll Acceleration ($Rad/Sec^2$) |
| Pitch Acceleration ($Rad/Sec^2$) |
| Yaw Acceleration ($Rad/Sec^2$) |
| (HALFI) |
| Center of Gravity - Ft. From Tail |
| Rocket X-Axis Velocity - U |
| Rocket Y-Axis Velocity - V |
| Rocket Z-Axis Velocity - W |
| Density |
| Center of Pressure - Ft. From Tail |
| Body X-Axis Acceleration - UDOT |
| Body Y-Axis Acceleration - VDOT |
| Body Z-Axis Acceleration - WDOT |
| (ALFTO) |
| (PMOMA) |
| Angle ALPHA (Degrees) |
| Angle BETA (DEG) |
| Angle ETA (DEG) |
| (FORSU) |
| (QMONA) |
| Angle THETA (DEG) |
| Angle PHI (DEG) |
| Angle PSI (DEG) |
| (FORSV) |
| (RMONA) |
| Relative Range Velocity |
| Relative Deflection Velocity - Positive Right |
| Relative Vertical Velocity - Positive Down |

TABLE XII. 2.75 PLOT TAPE VARIABLES (Cont'd)

| |
|---|
| (FORSW) |
| (PMOMG) |
| Wind Force Along Range Axis |
| Wind Force Along Deflection Axis |
| Wind Force Along Vertical Axis |
| Time Rate of Change in Mass (Slugs/Sec) |
| (QMOMG) - |
| (RMOMG) - |
| Axial Coefficient of Force |
| Side Coefficient of Force |
| Vertical Coefficient of Force |
| (NALP) |
| (BETI) |
| Bank Angle |
| (HCL) |
| Rate Damping Coefficient for Rates About Rocket Axes. |

## V. TEKPLOT PLOTTING PROGRAM

### A. Introduction

The 2.75 digital simulation takes advantage of a multipurpose plot routine developed at MICOM. This program, TEKPLOT, enables the user to quickly generate readable and understandable evaluation packages for each data run. In association with TEKPLOT is a routine, RPLOT, which ensures that the data input to the TEKPLOT routine is in a suitable format.

### B. RPLOT Tape Read/Write Program

RPLOT is a program designed to write data from an unformatted plot file to a formatted tape. Its listing is presented in Appendix E.

### C. TEKPLOT Program Operations

TEKPLOT is a program designed to generate x-y plots at a TEKTRONIX 4014 terminal, where hardcopies can be made. Its listing is presented in Appendix F. The program is designed to run interactively, using a local file as the data file. Normally, TAPE2 is used as the local data file when the following conditions are met.

1. The number of data points for each variable, NPTS, is on the first record of TAPE2.

2. The minimum and maximum values for each variable are stored in the next NV records where NV is the number of variables.

These conditions are present when the output from 2.75 is in its standard form. If both conditions are not met, the format is non-standard and the operator must answer a displayed question regarding the conditions with a "NO". At this time, the program reads through TAPE2 and finds the number of data points for each variable and the minimum and maximum values of each variable. These values are written to TAPE10 in standard format. NPTS blocks of records are then written to TAPE10 with each record holding the first, second, etc. data point of all records. This process of recording is done by SUBROUTINE FORMAT.

This program assumes that there are NV variables on each record of the data tape and that the operator knows the sequential order of the variables. The operator has the option of choosing any pair of variables to be plotted and can designate either of the variables to be the independent variable. The program provides the option for plotting 1 to N pairs of variables at a time, with a maximum N of 60.

Under current system restraints the program enters a pause mode after plotting to allow for a hard copy of the screen. To proceed to the next plot the operator must enter any character from the keyboard. Sometimes it may be necessary to send the cursur home and then enter any character from the keyboard. The program assumes that the current data tape contains one run of data. At the end of each series of plots the operator is given the option of whether or not he wants to continue plotting from the current data tape. If a "NO" is entered, the program stops execution. If "NO" is not entered, the operator can re-initialize the program and continue to plot from the current data tape. If he chooses to plot from the current data tape, he will use TAPE2 if the original data was in standard format. If the data was in non-standard format, he should return TAPE2 and rename TAPE10 as TAPE2. If the operator wants to plot from a different data run, TAPE2 and TAPE10, if applicable, must be returned. Then he must attach or create a new TAPE2 and proceed to re-initialize the program. If one had a file called DATA that contained several runs, each followed by an END OF RECORD, the tenth run could be plotted by copying records onto a file called DUMMY. The tenth record could then be copied into a file called TAPE2. TAPE2 rewound, and the program initialized.

SUBROUTINE LABLR creates tables for the x and y axis from a block data statement called LAB. The program is designed to handle 60 labels with each label containing a maximum of 3 words. The labes., in block data, appear in the same sequential order that the variables appear on the data tape. If the operator wants to change the label of the Ith variable on the record, he simply chang-s the Ith label in block data. The program will correctly select the labels for the specified pair of variables to be plotted.

SUBROUTINE XYGRID draws the grid for the XY plot. If desired, you can change the tic mark form of the grid by changing this subroutine.

SUBROUTINE FORMAT changes data received in non-standard form to standard form (described above).

## VI. CONCLUSIONS AND RECOMMENDATIONS

The study of PERSHING II impact points has been successfully completed for important debris through the use of the six degree of simulation program MSIXB1. The program has been described so that it may be used as a tool to aid in future studies. Important subroutines including ballistic and atmospheric coding have been added and successfully employed in MSIXB1. The 2.75 simulation, which may also be used for debris studies, has been described.

Thus, the PERSHING II debris dynamics simulation study has been successful, not only in providing needed data, but also in providing tools for the future.

Recommendations for the MSIXB1 model are to further investigate the integration routine and the quaternion transformations. When high spin rate debris from PERSHING II was investigated, the integration result varied widely as the integration step A(30) was increased. Extensive testing of the program under high spin rate conditions using various integration steps was performed, and a .0008 second step was determined to be the maximum valid step size. Any larger size caused a response in which the impact piont became continuously further from the precise point calculated using smaller step sizes. Also, a complete documentation of how MSIXB1's quaternion equations are developed from general quaternion equations would be helpful to users.

Of interest to users who use multiple runs, changing inputs under program control each time would be a statistical extension of the program. This could be accomplished by writing values to a data file as they are calculated by the program. This file could then be accessed by a statistical routine. This process is now being developed to analyze impact data using a circular error pattern statistical routine.

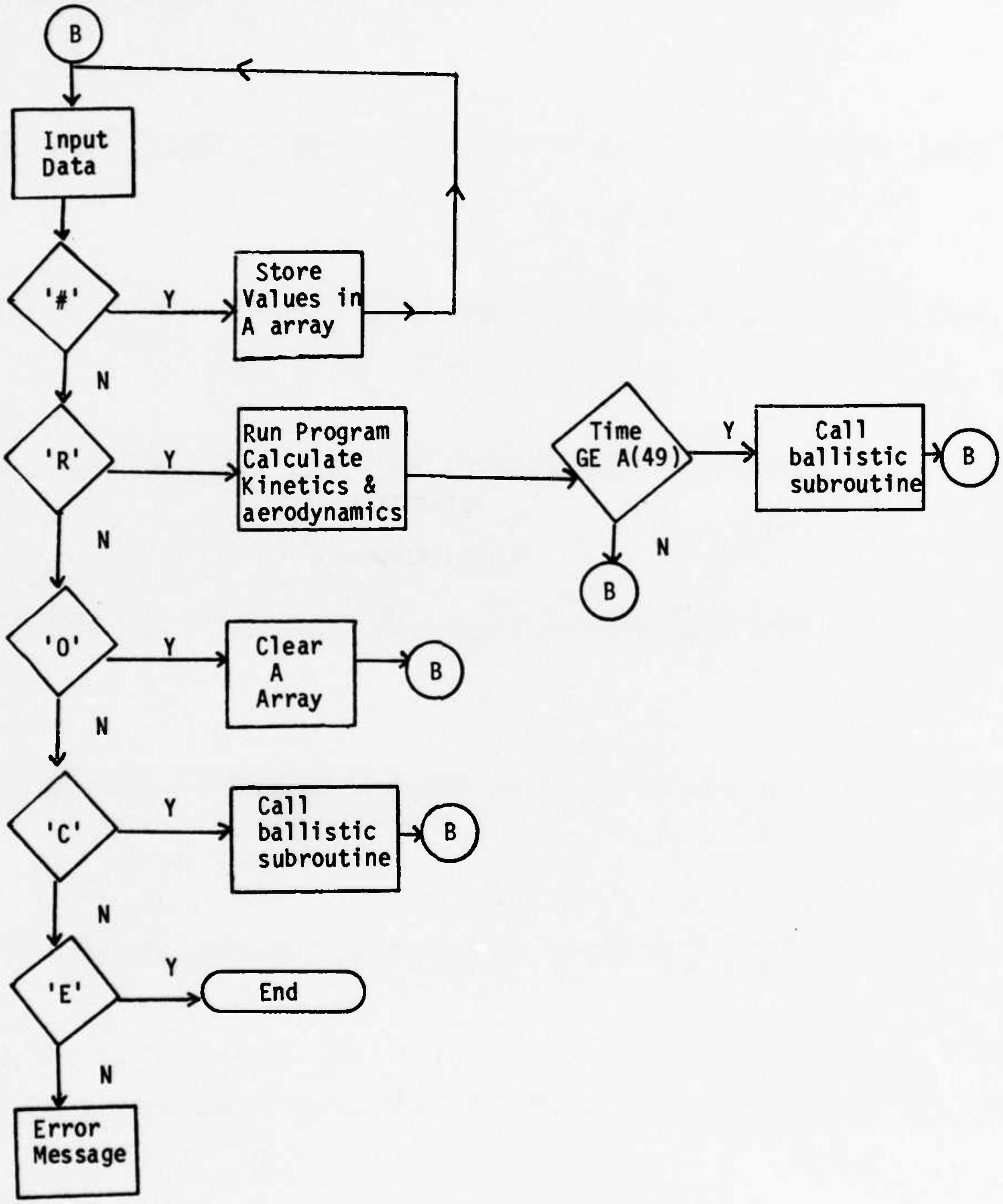
REFERENCES

[1] Jenkins, B. Z., "The Aerodynamic Environment of Rockets Launched from Helicopters," MICOM, Ppaer #19, 105h Navy Symposium on Aeroballistics, Fredericksburg, Virginia, Vol. 2., 1975.

[2] Dillard, R. A., "Notes on Lanier's 2.75 Fortran Simulation," 2.75 Digital Simulation Documentation Working Papers, Systems Simulation and Development Directorate, MICOM, June 1980.

APPENDIX A

MSIXB1 Flowchart


B
Input Data
'#'
Y
Store Values in A array
N
'R'
Y
Run Program Calculate Kinetics & aerodynamics
Time GE A(49)
Y
Call ballistic subroutine
B
N
B
N
'O'
Y
Clear A Array
B
N
'C'
Y
Call ballistic subroutine
B
N
'E'
Y
End
N
Error Message

# APPENDIX B

## MSIXB1 Listing

```
$BATCH
$PROG MAIN
      PROGRAM MSIXB1
C  *********  6DOF SIMULATION  *************
C     THIS PROGRAM WAS ORIGINALLY WRITTEN BY MARTIN MARIETTA   C
C     IT HAS BEEN MODIFIED BY PATRICK TILLEY,GERALDNJOHNSON,   C
C     DONN HALL AND OTHERS AT UAH TO ACCOMODATE DEBRIS STUDIES  C
      LOGICAL PFLAG,QRFLAG
      REAL M1,M2,M3
      EXTERNAL SIGN
      COMMON/DATA/ A(28ØØ)
      COMMON /ALLT/ DELT,TIME,ITIME,DELT2
      COMMON /AKIN/ADR,ADP,ADY,ADF1,ADF2,ADF3,ADM1,ADM2,ADM3,AWX,AWY,AWZ
      COMMON /MONT/NMRC,LRC,NRC,IB
      COMMON/KILL/ PE,CEP5Ø,THETAA,SIGTHE,A249,PK
      COMMON /AKOT/AA1,AA2,AA3,AAX,AAY,AAZ,AVX,AVY,AVZ,AX,AY,AZ,APD,
     1 AQD,ARD,AP,AQ,AR,APSI,ATHETA,APHI,AVA,AVT,AMACH,ALPHA,ABETA,AETA,
     2  AQDP,    AH,  AW,ACG,AIXX,AIYY,ACA,ACMT,ACLP,ACLD,ACL1,
     3  ACMP,ACMY,ACNP,ACNY,AC1X,AC2X,AC3X,AC1Y,AC2Y,AC3Y,AC1Z,AC2Z,AC3Z
     4,APHID,ATHED,APSID
      COMMON/GNC/ACQ,IE,      ESP,ESY,GP,GY,PRS,DBP,DBY,XP,YP,ZP,
     1 ACQRG,ACQALT,TSEQ,      SEARCH,TPRINT,PHASE,CPS,CYS,TSRCH,GGUID
      COMMON /PULSES/ NPLS(36Ø)
      COMMON /TARG/XBGO,YBGO,XFGO,YFGO,NBG,NFG,NTRUE,NTOT,JPLS
      COMMON /CAERO/      CNA,CND,CMA,CMD,V23S,V23,CAFAC,
     1 AV1,AV2,AV3,AV4,ABSA,ABSB,SAD,SBD,XX,CL,
     2 DPS,DYS,DCY,DCG,D2V,V2A,V3A,SA,SB,F1,F2,F3,M1,M2,M3
      COMMON /SPLOT/ QRMX,DIVE,TURN,DELTAX,DELTAY
      COMMON /CPLOT/ HERR,DFLARE,HGO,ADPX,ADYX,ELOS
      COMMON /SAUTO/ QFB,RFB,DTHS,R
      DIMENSION SYM(27)
      CHARACTER*7Ø TITLE
      CHARACTER *3 STA
      CHARACTER *12 FNAME
      CHARACTER *9 FM
      DATA SYM/'TIME','PSI ','PHI ','THET','P   ','Q   ','R   ',
     *         'PDOT','QDOT','RDOT','AX  ','AY  ','AZ  ','M1  ',
     *         'M2  ','M3  ','XM  ','YM  ','ZM  ','VXM ','VYM ',
     *         'VZM','DLAX','DLAY','F1  ','F2  ','F3  '/
      OPEN(1Ø,FILE='RUN.DAT',STATUS='OLD')
      READ(1Ø,8Ø7) NRUN
8Ø7   FORMAT(15)
      NRUN=NRUN+1
      CLOSE(1Ø)
      OPEN(1Ø,FILE='RUN.DAT',STATUS='OLD')
      WRITE(1Ø,BØ7) NRUN
      DATA DPR,ZERO,ZNINE/57.2957795,Ø.,999./
515   IB=Ø
COMMENT - INITIAL CALCULATIONS
    1 CALL INPUT(A,2BØØ,3ØØ,KRN)
      IF(IB.EQ.1) GOTO 515
      IF( NMRC.LE.Ø ) WRITE (3,1ØØ) KRN
      IF( NMRC.LE.Ø ) WRITE (6,1ØØ) NRUN
1ØØ   FORMAT (1X,'RUN',15)
      IF( NMRC.GT.Ø ) WRITE (3,11ØØ) KRN,NMRC
      IF( NMRC.GT.Ø ) WRITE (6,11ØØ) KRN,NMRC
11ØØ  FORMAT(4H1RUN,I5,2X,2HMC,15/)
      WRITE(6,1ØØ1)
      IF( A(25).GT.Ø. ) WRITE(6,1Ø11)
      IF( A(25).GT.Ø. ) WRITE(6,1Ø15)
      PQRMAX=1.E+6
      IF( A(26).GT.1. ) PQRMAX=A(26)*A(26)
      DUMMY=SQRT(PQRMAX)
      PFLAG=.FALSE.
```

```
       QRFLAG=.FALSE.
        IFINAL=A(2Ø)+2
       WFINAL=A(IFINAL)
       WRITE(6,11Ø) A(33),DUMMY,WFINAL
 11Ø   FORMAT(' RUN WILL STOP AT TIME GT',F5.1,
      1 ' TOTAL BODY RATE GT',F7.1,' OR WEIGHT LT ',F7.1)
       DELT=A(3Ø)
       DELPT=A(31)
       TPRINT=A(9Ø)
       DDX=Ø.
       DDY=Ø.
       PT=Ø.
        IPTF=Ø
       IPRINT=99
       TIME=Ø.
       ITIME=-2
       DELT2=DELT*DELT/2.
       IE=-1
       TG=A(33)
C
COMMENT - CONTINUAL CALCULATIONS
     2 CONTINUE
       IF (TG.GT.DELT) GO TO 3
       DELT=TG
       IE=1
     3 ITIME=ITIME+1
       IF (ITIME) 6,5,4
     4 ITIME=1
     5 TIME=TIME+DELT
     6 IF (TIME.GT.A(33)) IE=2
       TSEQ=TIME-A(9Ø)
C
       CALL KINE
C
       CALL AERO
C
C
       IF( TIME.LT.A(41) ) GO TO 15
C      BYPASS VARIABLE DELT
       TEMP=AP*AP+AQ*AQ+AR*AR
       IF(TEMP.LT.1ØØØØ.)TEMP=1ØØØØ.
       DELT=1Ø./SQRT(TEMP)
       DELT2=Ø.5*DELT*DELT
15     IF( (AP*AP+AQ*AQ+AR*AR).GT.PQRMAX ) IE=3
       IF( AW.LT.WFINAL ) IE=4
C
       GAMP=ATAN(-AVZ/AVX)*DPR
       GAMY=ASIN(AVY/AVT)*DPR
       TF=(-AVZ)/A(15)+SQRT((-AVZ)*(-AVZ)+2.*A(15)*AH)/A(15)
       XF=AVX*TF+AX+DDX
       YF=AVY*TF+AY+DDY
       IF( IPTF.NE.Ø ) GO TO 2Ø
       IPTF=99
       XFØ=XF+Ø.ØØ1 + A(1Ø1)
       YFØ=YF+A(1Ø2)
       DDX=A(1Ø1)
       DDY=A(1Ø2)
2Ø     DELTAX=(XF-XFØ)*.3Ø48
       DELTAY=(YF-YFØ)*.3Ø48
       DTOTAL=SQRT(DELTAX*DELTAX+DELTAY*DELTAY)
       HAPO=AH+Ø.5*AVZ*AVZ/A(15)
       QRMX=SQRT(AQ*AQ+AR*AR)
       IF (PFLAG) GO TO 52
       IF(AP.LT.A(28))GO TO 52
```

```
      PFLAG=.TRUE.
      WRITE(6,51)A(28)
   51 FORMAT(//5X,'***ROLL RATE EXCEEDS',F7.1)
      GO TO 1002
   52 IF(QRFLAG) GO TO 54
      IF(QRMX.LT.A(29))GO TO 54
      QRFLAG=.TRUE.
      WRITE(6,53)A(29)
   53 FORMAT(//5X,'*** QR RATE EXCEEDS',F7.1)
      GO TO 1002
   54 CONTINUE
      IF (A(31).LE.0.) GO TO 9
      IF(TIME.GE.A(205)) DELPT = A(32)
      IF(ABS(TIME-TPRINT).LT.DELT/2.) GO TO 22
      IF (IE.GT.0) PT=TIME
      IF (TIME.LT.PT) GO TO 2
      PT=PT+DELPT
      IPOINT=IPOINT+1
      IF( IPOINT.LT.10) GO TO 1002
      IPOINT=0
      WRITE(6,1001)
1001  FORMAT(' TIME',/9X,'PSI',9X,'PHI',9X,'THETA',
     1 9X,'X',11X,'Y',11X,'Z',11X,'P',11X,'Q',11X,'R',
     1 4X,'Q+R(RSS)',
     2 /10X,'VX',10X,'VY',10X,'VZ',10X,'AX',10X,'AY',10X,'AZ',
     3 8X,'PDOT',8X,'QDOT',8X,'RDOT',/10X,'F1',10X,'F2',10X,
     4 'F3',10X,'M1',10X,'M2',10X,'M3'11X,'W',9X,'IXX',9X,'IYY')
      IF( A(25).LT.0.5 ) GO TO 1002
      WRITE(6,1011)
1011  FORMAT(9X,'VXM',9X,'VYM',9X,'VZM',10X,'XM',10X,'YM',
     1 10X,'ZM',10X,'TF',6X,'DELTAX',6X,'DELTAY')
      WRITE(6,1015)
1015  FORMAT(9X,'C1X',9X,'C1Y',9X,'C1Z',8X,'DIVE',8X,'TURN',
     1 8X,'HAPO')
1002  WRITE(6,101) TIME
101   FORMAT(/2X,F10.6)
      WRITE(6,102) APSI,APHI,ATHETA,AX,AY,AZ,AP,AQ,AR,QRMX
      WRITE(6,102) AVX,AVY,AVZ,AAX,AAY,AAZ,APD,AQD,ARD
      WRITE(6,102) F1,F2,F3,M1,M2,M3,AW,AIXX,AIYY
102   FORMAT(10F12.3)
C       CUTOFF TIME FOR TRANSFER INTO BALLISTIC ROUTINE
      IF(TIME .GE. A(49)) THEN
        VXS=5.*(AC2X*AQ + AC3X*AR)
        VYS=5.*(AC2Y*AQ + AC3Y*AR)
        VZS=5.*(AC2Z*AQ + AC3Z*AR)
         AVX=AVX + VXS+A(42)
         AVY=AVY + VYS+A(43)
         AVZ=AVZ + VZS+A(44)
         A(297)=8.8
         WRITE(6,331) A(42),A(43),A(44)
  331    FORMAT(1X,'A(42) ',F12.6,' A(43) ',F12.6,' AA(44) ',
     *   F12.6)
         WRITE(6,332) VXS,VYS,VZS
  332    FORMAT(1X,' VXS   ',F12.6,' VYS   ',F12.6,' VZS    ',
     *   F12.6)
         WRITE(6,333) AVX,AVY,AVZ
333      FORMAT(1X,'DVX=',F12.6,5X,'DVY=',F12.6,5X,'DVZ=',F12.6)
         CALL BALLST(A)
         GOTO 515
      ENDIF
C     CONVERT TO METRIC FOR OPTIONAL PRINT
      IF( A(25).LT.0.5 ) GO TO 22
      XM=AX*.3048
      YM=AY*.3048
```

```
      ZM=AZ*.3048
      VXM=AVX*.3048
      VYM=AVY*.3048
      VZM=AVZ*.3048
      HAPO=HAPO*.3048
      WRITE(6,102) VXM,VYM,VZM,XM,YM,ZM,TF,DELTAX,DELTAY
      WRITE(6,1020) AC1X,AC1Y,AC1Z,DIVE,TURN,HAPO
1020  FORMAT(3F12.6,6F12.3)
C
   22 CONTINUE
9     CONTINUE
      IF (IE.LT.0) GO TO 2
      ENDFILE NT
C
      IF( A(27).GT.0.5 ) WRITE(3,66) DELTAX,DELTAY,DTOTAL
66      FORMAT(/3X,'DELTAX=',F12.1,' DELTAY=',F12.2,
     1 /3X,'TOTAL DELTA IIP=',F12.1)
C
      IF (NMRC-1) 1,68,69
68    CONTINUE
69    CONTINUE
      IF( A(40).LE.0.5 ) GO TO 1
      IF( NMRC.LT.(NRC-LRC) ) WRITE(4,4444) DELTAX,DELTAY,ZERO
      IF( NMRC.GE.(NRC-LRC) ) WRITE(4,4444) DELTAX,DELTAY,ZNINE
4444  FORMAT(3F10.0)
C
      GO TO 1
      END
$PROG AERO
      SUBROUTINE AERO
C     ---------------
      REAL IXX,IYY,M,M1,M2,M3,M2A,M3A
      EXTERNAL SIGN
      COMMON /ALLT/ DELT,TIME,ITIME,DELT2
      COMMON /AKIN/ DR,ADP,ADY,DF1,DF2,DF3,DM1,DM2,DM3,WX,WY,WZ
      COMMON /AKOT/A1,A2,A3,AX,AY,AZ,VX,VY,VZ,X,Y,Z,PD,QD,RD,P,Q,R,
     1PSI,THETA,PHI,VA,VT,AMACH,ALPHA,BETA,ETA,QDP,H,W,CG,IXX,IYY,CA,CMT
     2,CLP,CLD,CL1,CMP,CMY,CNP,CNY,C1X,C2X,C3X,C1Y,C2Y,C3Y,C1Z,C2Z,C3Z
     3,PHID,THED,PSID
      COMMON/DATA/ A(2800)
      COMMON /CAERO/     CNA,CND,CMA,CMD,V23S,V23,CAFAC,
     1 AV1,AV2,AV3,AV4,ABSA,ABSB,SAD,SBD,XX,CL,
     2 DPS,DYS,DCY,DCG,D2V,V2A,V3A,SA,SB,F1,F2,F3,M1,M2,M3
      COMMON /APLOT/ ALPHAA,BETAA,DPAERO,DYAERO,SADA,SBDA,
     & CMPA,CMYA,CNPA,CNYA,P1,P2,P3,P4,P5,P6,M2A,M3A
      COMMON /CPLOT/ HERR,DFLARE,HGO,ADPX,ADYX,ELOS
      DATA DPR/57.2957795/
C
      IF (ITIME.GE.0) GO TO 1
C     CONVERT FROM INCHES TO FEET,+ OFFSET GIVES + MOMENT
      DZCG=A(13)/12.
      DYCG=A(14)/12.
      W=A(19)
      M=A(19)/32.17
      WX=0
      WY=0
      WZ=0
      ALPHA=0.
      BETA=0.
      ETA=0.
C       INPUT ADDRESS POINTERS FOR FUNCTIONS OF WEIGHT IN LBS
      IXCG=A(20)
      IIXX=A(21)
      IIYY=A(22)
```

```
      XGP=A(23)
      SXJB=A(38)
      TSTACK=A(34)
C     INPUT THE IN-LB/PSI COEFFS FOR ROLL AND PITCH/YAW
      RTR=A(5Ø)
      PYTR=A(51)
      PHIPY=A(52)/DPR
C     COMPUTE COEFFS TO CONVERT CHAMBER PRESSURE TO TORQUE
      CEPT=COS(PHIPY)*PYTR/12.
      CEYT=SIN(PHIPY)*PYTR/12.
      CERT=RTR/12.
1     CONTINUE
C
COMMENT - 6 DOF ANGLES-OF-ATTACK
      VXA=VX-WX
      VYA=VY-WY
      VZA=VZ-WZ
      VA=SQRT(VXA**2+VYA**2+VZA**2)
      VIA=CIX*VXA+C1Y*VYA+C1Z*VZA
      V2A=C2X*VXA+C2Y*VYA+C2Z*VZA
      V3A=C3X*VXA+C3Y*VYA+C3Z*VZA
C
C  *****  ROTATE V2A,V3A,Q,R INTO AERO AXES  *********
      V2AS = V2A
      V3AS = V3A
      V2A = .7Ø7*(V2AS+V3AS)
      V3A = .7Ø7*(-V2AS+V3AS)
      OX = Q
      RX = R
      O = .7Ø7*(OX+RX)
      R = .7Ø7*(-OX+RX)
      IF( V2A*V3A ) 18Ø,19Ø,18Ø
18Ø   ALPHA=ATAN2(V3A,V1A)*DPR
      BETA=ATAN2(V2A,V1A)*DPR
      ETA=ATAN2(SORT(V2A**2+V3A**2),V1A)*DPR
19Ø   ALPHAA = ALPHA
      BETAA = BETA
C
      IF( TIME.GE.TSTACK ) GO TO 3ØØ
      FF=FN1V(A,5ØØ,TIME)
      FR=FN1V(A,55Ø,TIME)
      WDOT=FNIV(A,6ØØ,TIME)
      GO TO 4ØØ
3ØØ   TIME2=TIME-TSTACK
      FF=FN1V(A,15ØØ,TIME2)
      FR=FNIV(A,155Ø,TIME2)
      FDR=FN1V(A,17ØØ,TIME2)
      WDOT=FNIV(A,16ØØ,TIME2)
C     UPDATE MASS PROPERTIES HERE INSTEAD OF IN KINE
4ØØ   IF( ITIME.LT.Ø ) GO TO 555
      W=W-WDOT*DELT
      M=W/32.17
C      INTERPOLATE MASS PROPERTIES
555    XCG=FNIV(A,IXCG,W)
       IXX=FNIV(A,IIXX,W)
       IYY=FN1V(A,IIYY,W)
      XBGP=(XGP-XCG)/12.
      XBJB=(XCG-SXJB)/12.
C     F2JB=FNIV(A,165Ø,TIME)
C     F3JB=FNIV(A,17ØØ,TIME)
      DP=FN1V(A,3Ø5,TIME)
      DY=FNIV(A,32Ø,TIME)
      DELTAT=SORT(DP*DP+DY*DY)
      FIN=COS(DELTAT/DPR)*FF
```

```
      F2N=-SIN(DY/DPR)*FF
      F3N=-SIN(DP/DPR)*FF
      F1=F1N-FR-FDR
      F2=F2N
      F3=F3N
      TRATIO=F1/A(16Ø)
      MI=A(I6I)*TRATIO
      M2=A(I62)*TRATIO
      M3=A(163)*TRATIO
      IF(TIME.LT.A(164))M2=Ø.
      IF(TIME.GT.A(165))M2=Ø.
C
COMMENT - 6 DOF ACCELERATIONS
      A1=F1/M
      A2=F2/M
      A3=F3/M
      AX=C1X*A1+C2X*A2+C3X*A3
      AY=CIY*A1+C2Y*A2+C3Y*A3
      AZ=C1Z*A1+C2Z*A2+C3Z*A3+A(I5)
      Q = QX
      R = RX
      PD=DPR*M1/1XX
      QD=(M2*DPR+(1YY-1XX)*P*R/DPR)/IYY
      RD=(M3*DPR+(1XX-1YY)*P*Q/DPR)/IYY
      V2A = V2AS
      V3A = V3AS
      IF( V2A*V3A ) 28Ø,29Ø,28Ø
28Ø   ALPHA=ATAN2(V3A,V1A)*DPR
      BETA=ATAN2(V2A,V1A)*DPR
29Ø   ADPX = .7Ø7*(DP-DY)
      ADYX = .7Ø7*(DP+DY)
      RETURN
      END
$PROG SOLVE
      SUBROUTINE SOLVE(X,Y,XIN,XOUT,A,B,C)
      DIMENSION X(3),Y(3)
C     COMPUTE QUADRATIC COEFFS FOR Y=A*X**2 + B*X + C
      D1=Y(1)/((X(1)-X(2))*(X(1)-X(3)))
      D2=Y(2)/((X(2)-X(1))*(X(2)-X(3)))
      D3=Y(3)/((X(3)-X(1))*(X(3)-X(2)))
      C=D1*X(2)*X(3)+D2*X(1)*X(3)+D3*X(1)*X(2)
      B=-D1*(X(2)+X(3))-D2*(X(1)+X(3))-D3*(X(1)+X(2))
      A=D1+D2+D3
      XOUT=A*XIN*X1N+B*X1N+C
      RETURN
      END
$PROG FN1V
      FUNCTION FNIV (A,I,X)
C     -------------
COMMENT - FUNCTION OF ONE VARIABLE - 4/I5/7Ø
      DIMENSION A(28ØØ)
      N=A(1)+.5
      IF (N.EQ.Ø) GO TO 4
      IF (N.EQ.1) GO TO 5
      DO 1 J=4,N,2
      K=I+J
      FN1V=A(K+I)
      DEL=X-A(K)
      IF (DEL) 2,3,1
    I CONTINUE
    2 CONTINUE
      IF(A(K).EQ.A(K-2)) GO TO 3
      SLOPE=( FN1V -A(K-I))/(A(K)-A(K-2))
      FNIV=FNIV+DEL*SLOPE
```

```
    3 RETURN
    4 FN1V=Ø.
      RETURN
    5 FN1V=A(I+2)
      RETURN
      END
$PROG INPUT
      SUBROUTINE INPUT (A,NDIM,INITFN,NR)
C     ----------------
COMMENT -  DATA INPUT SUBROUTINE - 1/28/78
      DIMENSION LMC(2ØØ),NDIST(2ØØ),VAL1(2ØØ),VAL2(2ØØ),VALØ(2ØØ),IRC(2)
      DIMENSION A(NDIM),ISEQ(1ØØ),RSEQ(1ØØ),IS(3),NS(3),VSE(5,6)
      EQUIVALENCE (ISEQ(1),RSEQ(1))
      LOGICAL PSEQ,POINT,AZERO,DA,LA,CARLO,PMC
      COMMON /MONT/NMRC,LRC,NRC,IB
      INTEGER*2 IN(72),SYMBOL(45),IT,LABEL(72)
      DATA SYMBOL /1H ,1HØ,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H.,1H+,
     11H-,1H*,1H/,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     21HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H(,1H),
     3 1H'/
      DATA JSEQ,NSEQ,LSEQ,PSEQ,AZERO/Ø,Ø,1,2*.FALSE./
      DATA DA,CARLO,PMC,MRC,NMC,II,MMM/3*.FALSE.,4*Ø/
      DATA VSE/-1.,-.6,-.2,.2,.6,.2,-.6,-1.,.6,-.2,.6,-1.,-.2,-.6,.2,
     1  .6,-.2,-.6,.2,-1.,-.6,-.2,.2,-1.,.6,-1.,.2,.6,-.2,-.6/
C
COMMENT - INITIALIZE
      IF (AZERO) GO TO 2
      INFILE=5
      NMRC=Ø
      LRC=Ø
      NRC=Ø
      NR=Ø
      DO 1 I=1,NDIM
    1 A(I)=Ø.
      AZERO=.TRUE.
C
COMMENT - IDENTIFY AND SET UP COMMANDS
    2 IF(CARLO) GO TO 5ØØ
      IF (NSEQ.GT.Ø) GO TO 45Ø
      WRITE(6,4)
    4 FORMAT (5HØDATA)
    3 READ(INFILE,1Ø) IN
   1Ø FORMAT (72A1)
      WRITE(6,11) IN
   11 FORMAT (1X,72A1)
      J=1
      NB=Ø
   18 IT=IN(J)
      DO 2Ø I=3,11
      IF (IT.EQ.SYMBOL(I)) GO TO 1ØØ
   2Ø CONTINUE
C     ALPHABETIC CONTROL STATEMENTS
      IF (IT.EQ.SYMBOL(34)) GO TO 2ØØ
      IF (IT.EQ.SYMBOL(2)) GO TO 5Ø
      IF (IT.EQ.SYMBOL(35)) GO TO 7Ø
      IF (IT.EQ.SYMBOL(1)) GO TO 8Ø
      IF (IT.EQ.SYMBOL(29)) GO TO 9Ø
      IF (IT.EQ.SYMBOL(15)) GO TO 4Ø
      IF (IT.EQ.SYMBOL(19)) THEN
          CALL BALLST(A)
          IB=1
          RETURN
      ENDIF
      IF (IT.EQ.SYMBOL(21)) STOP
```

```
   3Ø WRITE (6,35)
   35 FORMAT (12H INPUT ERROR)
      WRITE (6,11) IN
      J=J-I
      WRITE (6,1I) (SYMBOL(I),I=I,J),SYMBOL(I4)
      STOP
   4Ø IF (INFILE.EQ.5) GO TO 42
      INFILE=5
      GO TO 3
   42 INFILE=2
      REWIND 2
   44 READ(2,1Ø) LABEL
      IF (LABEL(I).NE.IT) GO TO 44
      IF (LABEL(2).EQ.IT) GO TO 3Ø
      DO 48 I=2,72
      IF (IN(I).EQ.SYMBOL(I)) GO TO 49
      IF (LABEL(I).NE.IN(I)) GO TO 44
   48 CONTINUE
   49 WRITE(6,II) LABEL
      GO TO 3
   5Ø DO 55 I=I,NDIM
   55 A(I)=Ø.
      WRITE (6,58)
   58 FORMAT (14H ARRAY CLEARED)
      GO TO 3
   7Ø IF(IN(J+I).EQ.SYMBOL(28)) PSEQ=.TRUE.
      IF (IN(J+1).EQ.SYMBOL(2Ø)) PSEQ=.FALSE.
      GO TO 3
   8Ø NB=NB+I
      IF (NB.EQ.I2) GO TO 3
      J=J+I
      GO TO I8
C
COMMENT - SET UP MONTE CARLO OPTION
   9Ø J=J+I
      IF (IN(J).EQ.SYMBOL(28)) PMC=.TRUE.
      IF (IN(J).EQ.SYMBOL(2Ø)) PMC=.FALSE.
      NB=Ø
      DO 99 I=I,2
      IRC(I)=Ø
   92 J=J+I
      IF (IN(J).EQ.SYMBOL(I)) GO TO 98
      DO 94 K=2,II
      IF (IN(J).EQ.SYMBOL(K)) GO TO 96
   94 CONTINUE
      GO TO 3Ø
   96 IRC(I)=IØ*IRC(I)+K-2
      GO TO 92
   98 NB=NB+I
      IF (NB.GT.I1) GO TO 99
      IF (IRC(I).GT.Ø) GO TO 99
      GO TO 92
   99 NB=Ø
      NRC=IRC(I)
      LRC=IRC(2)
      CARLO=.TRUE.
      GO TO 3
C
COMMENT - DETERMINE DATA LOCATION
  IØØ LOC=Ø
  IØI LOC=IØ*LOC+I-2
      J=J+I
      DO IØ2 I=2,II
      IF (IN(J).EQ.SYMBOL(I)) GO TO IØI
```

```
  IØ2 CONTINUE
      IF (LOC.GT.NDIM) GO TO 3Ø
      IF (IN(J).EQ.SYMBOL(I)) GO TO I15
      IF (IN(J).EQ.SYMBOL(29)) GO TO 1Ø9
      IF (IN(J).EQ.SYMBOL(35)) GO TO I12
C
COMMENT - SET UP FUNCTION MODIFICATION
      IF (LOC.LT.INITFN) GO TO 3Ø
      N1=A(LOC)
      N2=A(LOC+I)
      NT=N1+N2
      NX=NI*N2
      IF (NT.LE.Ø) GO TO 3Ø
      IF (IN(J).NE.SYMBOL(22)) GO TO 1Ø7
      L=3
      IF (N1.EQ.1) L=2
      IF (NX.GT.Ø) L=NI+3
      LI=2
      IF (NX.GT.Ø) LI=-N1-1
      LASTL=NX+NT+1
      GO TO 12Ø
  IØ7 IF (IN(J).NE.SYMBOL(25)) GO TO IØ8
      IF (N1.LT.2) GO TO 3Ø
      L=2
      LI=2
      IF (N2.GT.Ø) LI=1
      LASTL=N1+2-LI
      GO TO 12Ø
  1Ø8 IF (IN(J).NE.SYMBOL(I8)) GO TO 3Ø
      IF (N2.LT.2) GO TO 3Ø
      L=NI+2
      LI=NI+I
      IF (N1.EQ.Ø) LI=2
      LASTL=NX+NT+2-LI
      GO TO I2Ø
C
COMMENT - SET UP MONTE CARLO DATA
  IØ9 IF (LOC.GT.INITFN) GO TO 3Ø
      J=J+I
      DO I1Ø I=3,5
      IF (IN(J).EQ.SYMBOL(I)) GO TO I1I
  I1Ø CONTINUE
      GO TO 3Ø
  I1I J=J+1
      IF (IN(J).NE.SYMBOL(I)) GO TO 3Ø
      NMC=NMC+1
      IF (NMC.GT.2ØØ) GO TO 3Ø
      LMC(NMC)=LOC
      NDIST(NMC)=I-2
      MMM=I
      GO TO I15
C
COMMENT - SET UP SEQUENCED DATA
  I12 ISEQ(LSEQ)=LOC
      NSEQ=NSEQ+1
      JSEQ=LSEQ+I
      LSEQ=LSEQ+2
C
COMMENT - SET UP TO READ DATA
  115 L=Ø
      JP=-1
      LASTL=-1
  I2Ø NB=I
      VAL=Ø.
```

```
      LASTOP=1
  I22 NO=Ø
      NP=Ø
      FN=Ø.
      POINT=.FALSE.
      NI=Ø
C
COMMENT - READ DATA AND COMPUTE VALUES
  I25 J=J+1
      IF (IN(J).EQ.SYMBOL(I)) GO TO 15Ø
      NB=Ø
      DO 135 I=2,11
      IF (IN(J).EQ.SYMBOL(I)) GO TO 14Ø
  I35 CONTINUE
      GO TO 16Ø
  I4Ø NI=1
      NO=IØ*NO+I-2
      IF (POINT)  NP=NP+I
  145 IF (J.LT.72) GO TO 125
  I5Ø NB=NB+1
      IF (NI) 182,152,I8Ø
  152 IF(NB.LT.12) GO TO I45
      IF (L.GT.LASTL) GO TO I87
  155 READ (INFILE,IØ) IN
      J=Ø
      JP=-1
      GO TO 12Ø
  16Ø IF (IN(J).NE.SYMBOL(I2)) GO TO I65
      POINT=.TRUE.
      GO TO I45
  I65 IF (IN(J).NE.SYMBOL(38)) GO TO 17Ø
      POINT=.FALSE.
      LV=LOC+L
      FN=A(LV)
      NI=-1
      GO TO 145
  I7Ø DO 171 I=I3,I6
      IF (IN(J).EQ.SYMBOL(I)) GO TO I79
  17I CONTINUE
      IF (IN(J).NE.SYMBOL(43)) GO TO I72
      JR=J
      JP=NO-I
      IF (LOC.GE.INITFN.AND.NO.EQ.Ø) JP=9999
      IF (J.EQ.72) GO TO 3Ø
      GO TO I2Ø
  172 IF (IN(J).NE.SYMBOL(44)) GO TO 3Ø
      IF (JP) 3Ø,I5Ø,I74
  I74 JP=JP-1
      J=JR
      GO TO I5Ø
  I79 NEXTOP=I-I2
      IF (NI.LT.Ø) GO TO I82
  I8Ø FN=NO
      FN=FN/1Ø.**NP
  I82 IF (LASTOP.EQ.I) VAL=VAL+FN
      IF (LASTOP.EQ.2) VAL=VAL-FN
      IF (LASTOP.EQ.3) VAL=VAL*FN
      IF (LASTOP.EQ.4) VAL=VAL/FN
      IF (NB.EQ.1) GO TO I83
      LASTOP=NEXTOP
      IF (J.EQ.72) GO TO 3Ø
      GO TO 122
C
COMMENT - STORE SEQUENCED VALUES
```

```
  183 IF (JSEQ.EQ.Ø) GO TO 188
      IF (LSEQ.GT.1ØØ) GO TO 186
      RSEQ(LSEQ)=VAL
      IF (LSEQ.EQ.JSEQ+1) ISEQ(LSEQ)=VAL
      LSEQ=LSEQ+1
      GO TO 12Ø
  186 JSEQ=Ø
      LSEQ=JSEQ-1
      NSEQ=NSEQ-1
      GO TO 3Ø
  187 IF (JSEQ.EQ.Ø) GO TO 3
      IF (LSEQ.LT.JSEQ+4) GO TO 186
      ISEQ(JSEQ)=LSEQ-JSEQ-2
      JSEQ=Ø
      GO TO 3
C
COMMENT - STORE MONTE CARLO VALUES
  188 IF (MMM.NE.1) GO TO 192
      I1=I1+1
      IF (I1.GT.1) GO TO 19Ø
      VAL1(NMC)=VAL
      GO TO 12Ø
  19Ø VAL2(NMC)=VAL
      I1=Ø
      MMM=Ø
      GO TO 3
C
COMMENT - STORE DATA VALUES
  192 LV=LOC+L
      IF (LV.GT.NDIM) GO TO 3Ø
      A(LV)=VAL
      IF (LOC.LT.INITFN) GO TO 193
      IF (L-1) 196,194,197
  193 IF (J-72) 196,3,3Ø
  194 L1=1
      LASTL=(A(LOC)+1.)*(A(LOC+1)+1.)
  196 L=L+1
      GO TO 12Ø
  197 IF (L.GE.LASTL) GO TO 3
      IF (L1.LT.Ø) GO TO 198
      L=L+L1
      GO TO 199
  198 IF (MOD(LASTL-L,-L1).EQ.Ø) L=L+1
      L=L+1
  199 IF (J-72) 12Ø,155,3Ø
C
COMMENT - SET RUN NUMBER, PRINT VALUES, AND START RUN
  2ØØ NP=Ø
      NB=Ø
      LA=.FALSE.
  21Ø J=J+1
      IF (IN(J).EQ.SYMBOL(1)) GO TO 27Ø
      NB=Ø
      DO 23Ø I=2,11
      IF (IN(J).EQ.SYMBOL(I)) GO TO 25Ø
  23Ø CONTINUE
      IF (IN(J).NE.SYMBOL(28)) GO TO 24Ø
      LA=.TRUE.
      NA=Ø
      GO TO 21Ø
  24Ø IF (IN(J).NE.SYMBOL(2Ø)) GO TO 3Ø
      DA=.FALSE.
      GO TO 21Ø
  25Ø IF(LA) GO TO 265
```

```
      NP=IØ*NP+I-2
      GO TO 21Ø
  265 NA=IØ*NA+I-2
      GO TO 21Ø
  27Ø NB=NB+1
      IF (NB.LT.I2) GO TO 21Ø
      IF(LA) DA=LA
      IF(NA.EQ.Ø) NA=NDIM
      IF (NP.GT.Ø) GO TO 278
      NR=NR+I
      GO TO 28Ø
  278 NR=NP
  28Ø IF (NSEQ.GT.Ø) GO TO 4ØØ
  282 IF (CARLO) GO TO 5ØØ
  29I IF(.NOT.DA) GO TO 295
      J=Ø
      K=9
      IF (K.GT.NA) K=NA
      WRITE (6,292) J,NR,(A(I),I=I,K)
  292 FORMAT (I2H INPUT ARRAY//I6.5H  RUN,I6,2X,9FI1.4)
  293 IF (K.EQ.NA) GO TO 295
      J=J+IØ
      K=K+IØ
      IF (K.GT.NA) K=NA
      WRITE (6,294) J,(A(I),I=J,K)
  294 FORMAT (I6,2X,IØFII.4)
      GO TO 293
  295 IF (NSEQ.EQ.Ø) RETURN
      IF (.NOT.PSEQ) RETURN
      I=I
      J=I
      WRITE (6,297)
  297 FORMAT (I7H SEQUENCED VALUES)
  298 L=ISEQ(I)
      WRITE (6,299) L,A(L)
  299 FORMAT (I6,2X,IØFII.4/(6X,IØFII.4))
      I=I+ISEQ(I+I)+3
      J=J+I
      IF (J.GT.NSEQ) RETURN
      GO TO 298
C
COMMENT - INITIALIZE SEQUENCED VALUES
  4ØØ DO 4Ø5 I=1,3
      IS(I)=I
  4Ø5 NS(I)=Ø
      I=I
      J=I
  4IØ L=ISEQ(I)
      A(L)=RSEQ(I+3)
      K=ISEQ(I+2)
      NS(K)=ISEQ(I+I)-I
      J=J+I
      I=I+NS(K)+4
      IF (J.GT.NSEQ) GO TO 282
      GO TO 4IØ
C
COMMENT - UPDATE SEQUENCED VALUES
  45Ø DO 455 I=I,3
      IS(I)=IS(I)+I
      IF (IS(I).LE.NS(I)) GO TO 465
  455 IS(I)=I
      I=I
      J=I
      IF (PSEQ) WRITE (6,297)
```

```
  460 L=ISEQ(1)
      I=I+ISEQ(I+1)+3
      A(L)=RSEQ(I-1)
      J=J+1
      IF (PSEQ) WRITE (6,299) L,A(L)
      IF (J.LE.NSEQ) GO TO 460
      NSEQ=0
      LSEQ=1
      JSEQ=0
      GO TO 3
  465 CONTINUE
      NR=NR+1
      I=1
      J=1
  470 K=ISEQ(I+2)
      K=IS(K)+I+2
      L=ISEQ(1)
      A(L)=RSEQ(K)
      I=I+ISEQ(I+1)+3
      J=J+1
      IF (J.GT.NSEQ) GO TO 282
      GO TO 470
C
COMMENT - UPDATE MONTE CARLO VALUES
  500 IF (LRC+MRC.GE.NRC) GO TO 530
      MRC=MRC+1
      NMRC=MRC
      CALL RAN1(MRC+LRC)
      NC=0
      IF (.NOT.PMC) GO TO 504
      WRITE(6,502)
  502 FORMAT (//18H    ARRAY LOCATION,5X,13HNOMINAL VALUE,
     1 5X,9HRUN VALUE,5X,10HDIFFERENCE)
  504 NC=NC+1
      V1=VAL1(NC)
      V2=VAL2(NC)
      ND=NDIST(NC)
      LL=LMC(NC)
      IF (MRC.EQ.1) VAL0(NC)=A(LL)
      NZ=MRC+LRC
      NSE=MOD((NZ-1)/5+NC-1,120)
      ISE=MOD(NSE,6)+1
      JSE=MOD(NSE,5)
      KSE=MOD(NSE+NSE/60,4)+1
      LSE=MOD(JSE+(NZ-1)*KSE,5)+1
      RUNIF=RANU(.4)+VSE(LSE,ISE)
      IF (ND.NE.1) GO TO 506
      A(LL)=V1+V2*RUNIF*(1.1075+.142/(1.0898-ABS(RUNIF)))
      GO TO 520
  506 IF (ND.NE.2) GO TO 510
      A(LL)=V1+V2*RUNIF
      GO TO 520
  510 IF (RUNIF.LT.0.) GO TO 512
      A(LL)=V2
      GO TO 520
  512 A(LL)=V1
  520 IF (.NOT.PMC) GO TO 526
      DIFF=A(LL)-VAL0(NC)
      WRITE(6,522) LL,VAL0(NC),A(LL),DIFF
  522 FORMAT (8X,I4,12X,F10.3,6X,F10.3,4X,F10.3)
  526 IF (NC.LT.NMC) GO TO 504
      IF (MRC.EQ.1) GO TO 291
      RETURN
C
```

```
COMMENT - RESTORE MONTE CARLO VALUES
  53Ø DO 531 NC=1,NMC
      LL=LMC(NC)
  531 A(LL)=VALØ(NC)
      NMRC=Ø
      MRC=Ø
      IF (NSEQ.GT.Ø) GO TO 45Ø
      CARLO=.FALSE.
      NMC=Ø
      GO TO 3
      END
$PROG KINE
      SUBROUTINE KINE
C     ---------------
       REAL IXX,IYY
      COMMON /ALLT/ DELT,TIME.ITIME,DELT2
      COMMON /AKIN/ DR,DP,DY,DF1,DF2,DF3,DM1,DM2,DM3,WX,WY,WZ
      COMMON /AKOT/A1,A2,A3,AX,AY,AZ,VX,VY,VZ,X,Y,Z,PD,QD,RD,P,Q,R,
     1PSI,THETA,PHI,VA,VT,AMACH,ALPHA,BETA,ETA,QDP,H,W,CG,IXX,IYY,CA,CMT
     2,CLP,CLD,CLI,CMP,CMY,CNP,CNY,C1X,C2X,C3X,C1Y,C2Y,C3Y,C1Z,C2Z,C3Z
     3,PHID,THED,PSID
      COMMON/DATA/ A(28ØØ)
      DIMENSION QA(4),QAP(4)
      DATA DPR/57.2957795/
      IF (ITIME.GE.Ø) GO TO I
      X=A(1)
      Y=A(2)
      Z=-A(3)
      V=A(4)
      CA5=COS(A(5)/DPR)
      VX =V*CA5*COS(A(6)/DPR)
      VY =V*SIN(A(5)/DPR)
      VZ =-V*SIN(A(6)/DPR)*CA5
      PSI=A(7)
      THETA=A(8)
      PHI=A(9)
      CPSI=COS(A(7)/DPR)
      SPSI=SIN(A(7)/DPR)
      CPHI=COS(A(9)/DPR)
      SPHI=SIN(A(9)/DPR)
      CTHETA=COS(THETA/DPR)
      STHETA=SIN(THETA/DPR)
      CIX=CTHETA*CPSI
      CIY=SPSI
      C1Z=-STHETA*CPSI
      C2X=-CPHI*SPSI*CTHETA+SPHI*STHETA
      C2Y=CPHI*CPSI
      C2Z=SPHI*CTHETA+STHETA*SPSI*CPHI
      C3X=SPHI*SPSI*CTHETA+CPHI*STHETA
      C3Y=-SPHI*CPSI
      C3Z=CPHI*CTHETA-STHETA*SPSI*SPHI
      IQUAT=Ø
      QAP(4)=Ø.5*SQRT(I.+C1X+C2Y+C3Z)
      QAP4S=Ø.25*(I.+CIX+C2Y+C3Z)
88Ø   FORMAT(IX,'QAP3')
      QAP(3)=SQRT(QAP4S-Ø.5*(CIX+C2Y))
      IF(C2X.GT.CIY)QAP(3)=-QAP(3)
      QAP(2)=SQRT(QAP4S-Ø.5*(CIX+C3Z))
      IF(CIZ.GT.C3X)QAP(2)=-QAP(2)
      QAP(I)=SQRT(QAP4S-Ø.5*(C2Y+C3Z))
      IF(C3Y.GT.C2Z)QAP(I)=-QAP(I)
C
      P=A(1Ø)
      Q=A(II)
```

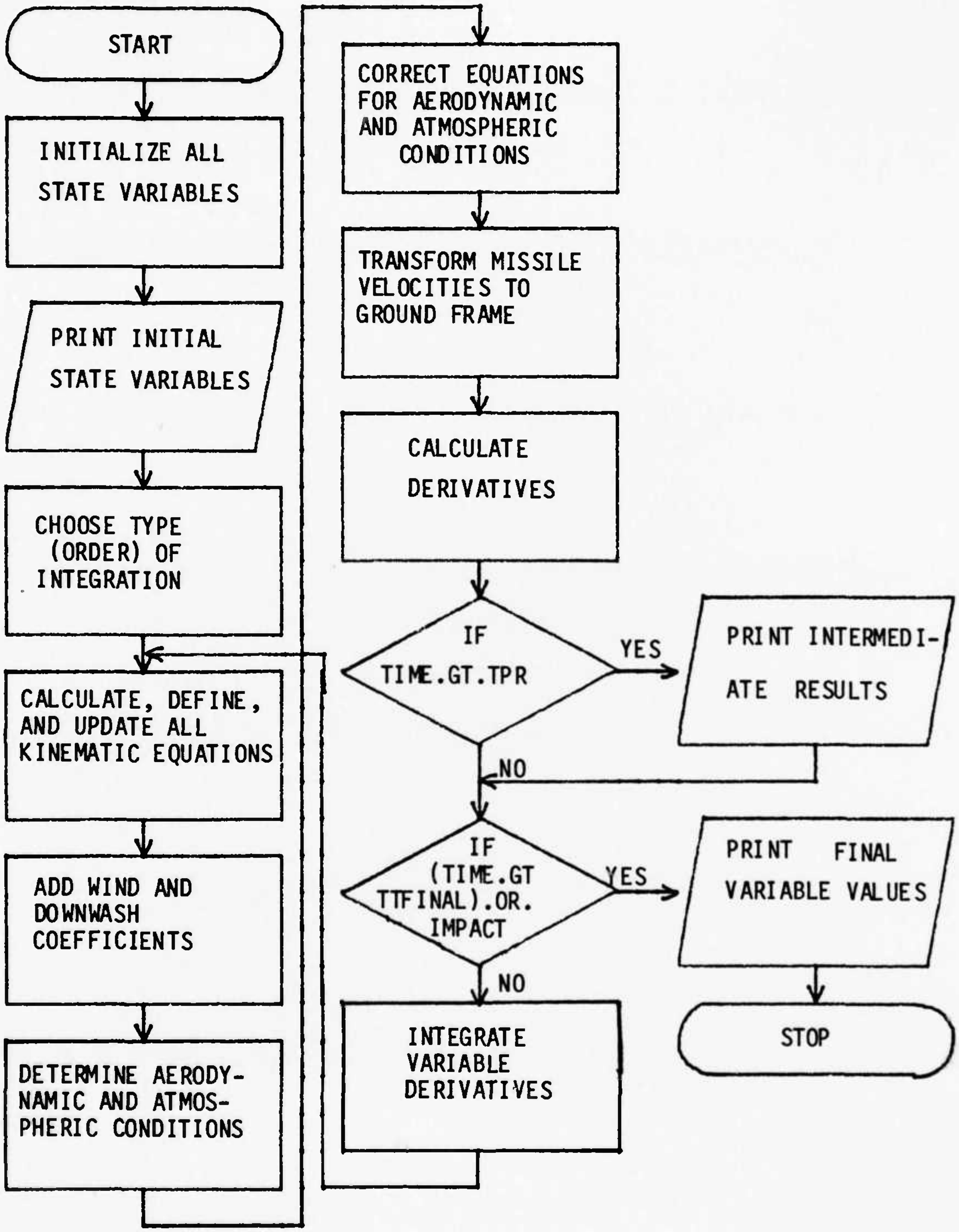
```
      R=A(12)
      GO TO 2
C
    1 CONTINUE
      VX   = VX +AX*DELT
      VY   = VY +AY*DELT
      VZ   = VZ +AZ*DELT
      X=X+VX*DELT+AX*DELT2
      Y=Y+VY*DELT+AY*DELT2
      Z=Z+VZ*DELT+AZ*DELT2
      P=P+PD*DELT
      Q=Q+QD*DELT
      R=R+RD*DELT
C
      DELP=(P*DELT+PD*DELT2)/DPR
      DELQ=(Q*DELT+QD*DELT2)/DPR
      DELR=(R*DELT+RD*DELT2)/DPR
      DELAS=DELP*DELP+DELQ*DELQ+DELR*DELR
      DQA=Ø.125*DELAS*(DELAS/48.-1.)+1.
      SQA=Ø.5-DELAS/48.
      HXA=DELP*SQA
      HYA=DELQ*SQA
      HZA=DELR*SQA
      QA(1)=DQA*QAP(1)+HZA*QAP(2)-HYA*QAP(3)+HXA*QAP(4)
      QA(2)=-HZA*QAP(1)+DQA*QAP(2)+HXA*QAP(3)+HYA*QAP(4)
      QA(3)=HYA*QAP(1)-HXA*QAP(2)+DQA*QAP(3)+HZA*QAP(4)
      QA(4)=-HXA*QAP(1)-HYA*QAP(2)-HZA*QAP(3)+DQA*QAP(4)
      DO 444 I=1,4
  444 QAP(I)=QA(I)
      IQUAT=IQUAT+1
      IF(IQUAT.LT.1ØØ)GO TO 446
      TEMP=Ø.5*(3.-QA(1)*QA(1)-QA(2)*QA(2)-QA(3)*QA(3)-
     * QA(4)*QA(4))
      DO 445 I=1,4
  445 QAP(I)=TEMP*QAP(I)
      IQUAT=Ø
  446 CONTINUE
      C1X=QAP(1)**2-QAP(2)**2-QAP(3)**2+QAP(4)**2
      C1Y=2.*(QAP(1)*QAP(2)+QAP(3)*QAP(4))
      C1Z=2.*(QAP(1)*QAP(3)-QAP(2)*QAP(4))
      C2X=2.*(QAP(1)*QAP(2)-QAP(3)*QAP(4))
      C2Y=QAP(2)**2+QAP(4)**2-QAP(1)**2-QAP(3)**2
      C2Z=2.*(QAP(2)*QAP(3)+QAP(1)*QAP(4))
      C3X=2.*(QAP(1)*QAP(3)+QAP(2)*QAP(4))
      C3Y=2.*(QAP(2)*QAP(3)-QAP(1)*QAP(4))
      C3Z=QAP(3)**2+QAP(4)**2-QAP(1)**2-QAP(2)**2
      THETA=ATAN2(-C1Z,C1X)*DPR
      PHI=ATAN2(-C3Y,C2Y)*DPR
      PSI=ASIN(C1Y)*DPR
C
    2 CONTINUE
      H=-Z
      VT =  SQRT(VX**2+VY**2+VZ**2)
      RETURN
      END
$PROG SIGN
      FUNCTION SIGN(A1,A2)
      Q=ABS(A1)
      IF(Q.NE.Ø.Ø)GO TO 1
      SIGN=A1
      RETURN
    1 CONTINUE
      Z=1.Ø
      IF(A2.LT.Ø.Ø)Z=-1.Ø
```

```
      SIGN=Z*ABS(AI)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
$PROG RANI
      SUBROUTINE RANI(I,N)
C     INCLUDED IN ORIGINAL PROGRAM BUT NOT ACTIVATED BY UAH
C RANDOM NUMBER INITIATOR FOR 8/32
C INITIALIZES A DOUBLE PRECISION FRACTION,N, FOR SUBSEQUENT USE
C BY THE RANU AND/OR RANG FUNCTIONS
C
      DOUBLE PRECISION M,N
      M=I7I79B69184.DØ
C     MODULUS M = 2**34
      N=(562951413.DØ + 2B182B172.DØ*I)/M
      N=N-IDINT(N)
C
C     N IS NOW A DOUBLE PRECISION FRACTION
C     N=2**(-34)  + MISC HIGHER BITS WHICH DEPEND ON I
C
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
$PROG RANU
      FUNCTION RANU(DPFRC,A)
      DOUBLE PRECISION DPFRC,XX
      XX=DPFRC*I3IØ75.DØ
      N=XX-IDINT(XX)
      XI=N
      RANU=A*XI
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
$PROG BALLST
      SUBROUTINE BALLST(A)
      DIMENSION A(28ØØ)
      COMMON /ALLT/ DELT,TIME,ITIME,DELT2
      COMMON /AKOT/AI,A2,A3,AX,AY,AZ,VX,VY,VZ,X,Y,Z,PD,QD,RD,P,Q,R,
     IPSI,THETA,PHI,VA,VT,AMACH,ALPHA,BETA,ETA.QDP,H,W,CG,IXX,IYY,CA,CMT
     2,CLP,CLD,CLI,CMP,CMY,CNP,CNY,CIX,C2X,C3X,CIY,C2Y,C3Y,C1Z,C2Z,C3Z
     3,PHID,THED,PSID
      BC=A(45)
      BDT=A(46)
      BDP=A(47)
      IPT=Ø
      AXO=AX *DELT/BDT
      AYO=AY *DELT/BDT
      AZO=AZ *DELT/BDT
      VXO=VX
      VYO=VY
      VZO=VZ
      TNXTPR=TIME+BDT
      BDTI=BDT/2.
      WRITE(6,7Ø7)TIME,X,Y,Z,VX,VY,VZ,AX,AY,AZ,RHO
1     VT=SQRT(VX**2 + VY**2 + VZ**2)
      CALL ATMO(-Z,RHO,VS)
      AX=-I6.I*RHO*VT*VX/BC
      AY=-I6.I*RHO*VT*VY/BC
      AZ=-I6.I*RHO*VT*VZ/BC + 32.2
      VX= VX+(AX+AXO)*BDTI
      IF(ABS(VX).LE.1.E-35) VX=Ø.Ø
      AXO=AX
      VY= VY+(AY+AYO)*BDTI
      IF(ABS(VY).LE.I.E-35) VY=Ø.Ø
```

```
      AYO=AY
      VZ= VZ+(AZ+AZO)*BDTI
      AZO=AZ
      X= X+(VX+VXO)*BDTI
      VXO=VX
      Y= Y+(VY+VYO)*BDTI
      VYO=VY
      Z= Z+(VZ+VZO)*BDTI
      VZO=VZ
      TIME=TIME+BDT
      IF(IPT.EQ.1) GOTO 767
      WRITE(6,727)
727   FORMAT(6X,'TIME',9X,'X',12X,'Y',9X,'Z',12X,'VX',10X,'VY',
     +         11X,'VZ',11X,'AX',9X,'AY',9X,'AZ',9X,'RHO')
      IPT=I
767   IF(TIME .GE. TNXTPR)  THEN
      WRITE(6,707)TIME,X,Y,Z,VX,VY,VZ,AX,AY,AZ,RHO
707   FORMAT(10F12.3,F12.9)
      TNXTPR=TNXTPR+BDP
      ENDIF
      IF (Z .GE. 0.0) THEN
          WRITE(6,757)TIME,X,Y,Z,VX,VY,VZ,AX,AY,AZ
757       FORMAT(10F12.3)
          RETURN
      ENDIF
      GO TO I
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
$PROG ATMO
      SUBROUTINE ATMO(H,RHO,VS)
C     H IS MEASURED POSITIVE UP
C     H HAS UNITS OF FEET
C     REFERENCE AEROPLANE AERODYNAMICS,DOMMASCH,1951
C     HNA=HEIGHT ABOVE WHICH NO SENSIBLE ATMOSPHERE EXIST
      HNA=300000.0
      RHOSL=.002378
C     FOR HNA=30000 FEET THE DENSITY IS ONE MILLIONTH THAT OF SL
C     RHOSL IS ATMOSPHERIC DENSITY OF SEA LEVEL IN SLUGS/FT**3
      IF(H.LE.35332.0) THEN
          RR=(1.0-.0000068*H)**4.256
      ELSE IF(H.LE.HNA) THEN
          RR=1.32/EXP(1.452+((H-35332.)/20950.))
      ELSE
          RR=0.0
      ENDIF
      RHO=RR*RHOSL
C     VS IS SPEED OF SOUND IN FT/SEC
C     IF(H.LE.35332.0) THEN
C         VS=1116.4 - .0042*H
C     ELSE IF(H.LE.83000.0) THEN
C             VS=968.5
C         ELSE IF(H.LE.173885.0) THEN
C                 VS=968.5 + .001515*(H-83000.)
C               ELSE IF(H.LE.HNA) THEN
C                     VS=1106.19- .0025324*(H-173885.)
C                   ELSE
C                     VS=786.8
C     ENDIF
      RETURN
C      M.M. HALLUM 13 JAN 1983 - AERO EQUATIONS
      END
```

APPENDIX C

2.75 6-DOF Flowchart




2.75 DIGITAL SIMULATION

## APPENDIX D

## 2.75 Listing

```
$BATCH
$XREF
$PROG MAIN275
C
C                          WRHD151
C
C     PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2,TAPE10)
C                   ******************************************
C                   *           6-DOF SIMULATION             *
C                   *           2.75 INCH ROCKET             *
C                   *                 AND                    *
C                   *           DOWNWASH PROFILE             *
C                   *                                        *
C                   ******************************************
C       LOGICAL VARIABLE IMPACT STOPS TRAJECTORY WHEN ROUND REACHES
C   GROUND,THAT IS ALTITUDE XX(2,9) GOES POSTIVE.
C       LOGICAL VARIABLE 'NOTIME' STOPS TRAJECTORY WHEN FLIGHT TIME HAS
C   EXCEEDED ALLOTED RUNTIME TIME OF 'TFINAL'.
      LOGICAL       BURNOUT , EOL      , IMPACT  , LRATE    , NOTIME
      LOGICAL       LNEXIT
      LOGICAL       SETBO   , SKIP
      REAL          IXO     , IXF      , IYO     , IYF
      REAL          IX      , IY       , IZ
      REAL          IXDOT   , IYDOT    , IZDOT   , ISP
      REAL          MCDOFF  , MCDON    , MCMQ    , MCNA     , MCP
      REAL          MDOT
      DIMENSION ACTI(3,3)
      DIMENSION CDPOFF(21)  , CDPON(21)   , CMQT(17)   , CNAT(7)
      DIMENSION CPT(8,2)      , CPAF(2)
      DIMENSION CK(4,50), IDAY(3)
      DIMENSION DCM(3,3)    , DTDH(12)
      DIMENSION HGEOPB(13)
C     DIMENSION LABEL(8)
      DIMENSION MCDOFF(21)  , MCDON(21)   , MCMQ(18)      , MCNA(7)
      DIMENSION MCP(8)
      DIMENSION PRESB(12)
      DIMENSION SPINT(30)   , SPNRAT(30)
      DIMENSION TEMPMB(12) , THRUST(9)     , TTIME(9)
      DIMENSION WUVW(3,3)   , WXYZ(3,3)
      DIMENSION XX(3,50)
      DIMENSION Y(3,50)
C       SPNRAT IS THE 2.75 AVERAGE ROLL RATE IN RPS AS OF 30 MAY 80
C   FROM MEASURED NAVY DATA PREPARED BY D.BROWN. THE VALUES IN SPINT
C   ARE THE TIME BREAKPOINTS FOR SPNRAT.
      DATA SPINT/ 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 1.0, 2.0,
     A 2.2, 2.4, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0,
     B 9.0, 10.0, 12.0, 16.0, 20.0, 25.0/
      DATA SPNRAT/ 0.0, 17.0, 28.0, 34.0, 39.0, 43.5, 46.3, 48.7, 51.5,
     A 54.0, 19.0, 14.0, 8.3, 6.7, 8.5, 11.0, 12.0, 12.2, 11.0, 9.2,
     B 6.0, 3.6, 0.0, -1.0, -2.0, -2.5, -2.5, -2.6, -3.0, -3.0/

C   POWER OFF DRAG COEFFICIENT-CDPOFF-FUNCTION OF MACH
      DATA(CDPOFF(I),I=1,20)/
     A 0.700,0.700,0.730,0.809,0.863,0.960,0.977,0.989,1.000,0.1008,
     B 1.010,1.012,1.005,0.990,0.970,0.940,0.875,0.811,0.765,0.730/
C   POWER OFF DRAG COEFFICIENT MACH NUMBER TABLE-MCDOFF
      DATA(MCDOFF(I), I = 1, 20)/
     A 0.0, 0.76, 0.82, 0.90, 0.94, 1.0, 1.03, 1.06, 1.10, 1.15,
     B 1.18, 1.28,1.34, 1.48, 1.58,1.71,1.94, 2.20, 2.40,2.6/

C   POWER ON DRAG COEFFICIENT-CDPON-FUNCTION OF MACH
```

```
      DATA(CDPON(I), I=1,20)/
     A 0.55,0.55,0.576,0.629,0.65,.685,0.699,0.71,0.727,0.7415,0.747,
     B.76,0.757,0.753,0.742,.724,0.681,0.65,0.628,0.612/
C    POWER ON DRAG COEFFICIENT MACH NO. TABLE-MCDON
      DATA(MCDON(I), I =1, 20)/
     A 0.0,.76, 0.82, 0.90,.94, 1.0, 1.03, 1.06, 1.10, 1.15, 1.18,
     B 1.28,1.34,1.48 , 1.58, 1.71, 1.94, 2.20,2.40,2.60/

C    CLP-ROLL DAMPING COEFFICIENT
      DATA CLP/-29.539/

C    PITCH DAMPING COEFFICIENT-CMQT-PER RAD/SEC/-FUNCTION OF- MACH
      DATA(CMQT(I), I=1,17)/
     A -230.5, -230.5, -261.0, -290.3, -301.8, -308.1, -313.2, -340.0,
     B -379.4, -368.0, -325.9, -294.1, -253.4, -94.2, -36.7, -31.3,
     C -30.8/
C    PITCH DAMPING MACH NO. TABLE-MCMQ
      DATA(MCMQ(I), I = 1, 18)/
     A 0.0, 0.60, 0.70, 0.80, 0.90, 0.95, 1.00, 1.05, 1.10, 1.20,
     B 1.30, 1.40, 1.50, 2.00, 3.00, 4.0, 5.0, 10.0/

C    NORMAL FORCE COEFFICIENT SLOPE-PER DEGREE-CNAT-FUNCTION OF MACH
      DATA(CNAT(I), I=1, 7)/
     A 12.44,12.44,12.597,8.874,6.585,6.012,6.012/
C    NORMAL FORCE MACH NO. TABLE-MCNA
      DATA(MCNA(I), I = 1, 7)/
     A 0.0, 0.60, 0.80, 1.75, 2.50, 3.00, 10.0/

C    ANGLE OF ATTACH TABLE FOR CP-IN DEGREES-CPAF
      DATA(CPAF(I), I = 1, 2)/
     A 0.0, 90.0/

C    CENTER OF PRESSURE-CPT-FUNCTION OF MACH AND ANGLE OF ATTACK
C    IN CALIBERS FROM NOSE
      DATA(CPT(I,1), I = 1,7)/
     A 14.54, 14.54, 14.75, 15.29, 15.07, 13.31, 12.21/
     B (CPT(1,2), I=1,7)/
     C 14.54, 14.54, 14.75, 15.29, 15.07, 13.31, 12.21/
C    CENTER OF PRESSURE MACH NO. TABLE-MCP
      DATA(MCP(I), I = 1, 8)/
     A 0.0, 0.60, 0.80, 1.00, 1.75, 2.50, 3.00, 10.0/

C    INITIAL AND FINAL VALUES OF CENTER OF GRAVITY-CGO,CGF-FEET FROM NOSE
      DATA CGO, CGF/2.116667, 1.800000/

C    REFERENCE DIAMETER-DREF-2.75 INCHES-IN FEET
      DATA DREF/0.2292/
C    ATMOSPHERIC DENSITY REFERENCE TABLE
      DATA(DTDH(I), I=2,12)/
     A -0.0065, 0.0, 0.003, 0.0, 0.004, -0.0045, 0.0, 0.02, 0.01,
     C 0.005, 0.0035/

      DATA F/0.34838395/

      DATA(HGEOPB(I), I= 2, 13)/
     A 0.0, 11000., 25000., 47000., 53000., 79000., 90000., 105000.,
     B 160000., 170000., 200000., 700000./
C    1 SLUG= 32.1739 LB. OF MASS (LB-SQ.IN.=0.0002158 SLUG-SQ.FT.)

C    INITIAL AND FINAL AXIAL MOMENT OF INERTIA-IXO,IXF-SLUG-FEET*FEET
       DATA IXO, IXF/5.6755E-3, 2.4621E-3/
C    INITIAL AND FINAL TRANSVERSE MOMENT OF INERTIA-IYO,IYF
```

```
      DATA IYO,IYF/1.4449968, 1.1869/

      DATA IMPACT/.FALSE./

      DATA NOTIME/.FALSE./

      DATA(PRESB(I), I = 2,12)/
     A 1.0332E4, 2.3078E3, 2.5381E2, 1.2285E1, 5.9478, 1.029E-1,
     B 1.066E-2, 7.618E-4, 3.691E-5, 2.88E-5, 1.454E-5/

      DATA PI/3.14159/

      DATA RADCON/57.2957795/
C     REFERENCE SURFACE AREA OF EARTH
      DATA S/6.378178E+6/


C    MOTOR BURN TIME-SUSOFF
      DATA SUSOFF/1.075/

C    ATMOSPHERIC TEMPERATURE REFERENCE TABLE
      DATA(TEMPMB(I), I = 2, 12)/
     A 288.16, 216.66, 216.66, 282.66, 282.66, 165.66,
     B 165.66, 225.66, 1325.66, 1425.66, 1575.66/

C    THRUST TABLE-THRUST-POUNDS;TIME TABLE OF THRUST
      DATA(THRUST(I), I = 1, 9)/
     A 0.0, 1600.0, 1350.0, 1300.0, 1550.0, 1750.0, 1550.0, 0.0, 0.0/
      DATA(TTIME(I), I = 1, 9)/
     A 0.0, 0.05, 0.2, 0.3, 0.8, 0.95, 1.00, 1.075, 10.0/

C    TOTAL IMPULSE-TI
      DATA TI/1510./

C    INITIAL AND FINAL VALUE OF MISSILE WEIGHT WO, WF
      DATA WO, WF/23.75, 16.59/

C    WEIGHT WHEN ROUND CLEARS LAUNCHER-WL
      DATA WL/23.296/

C    MISSILE LENGTH IN FEET-XLENG
      DATA XLENG/5.49167/

C     FIND THE CURRENT DATE AS DATE(IDAY)
      CALL DATE(IDAY)
C    DATA STATEMENTS INITIALIZING ALL VARIABLES IN NAMELIST PUTT
      DATA ALTIN/-50.0/
      DATA DELTT, DELO2, DIVANG/0.001, 0.0005, 0.0/
      DATA DNSCH, DPIT, DT, DYAW, DZZ/0.5, 0.0, 0.001, 2*0.0/
      DATA EOL/.FALSE./
      DATA FACTOR, FCA, FTORM, GREFF/0.0, -6.56, 0.0, 32.174/
      DATA HELIC, N, NN/0.0, 15, 1/
      DATA PHIZD, PITMAL, PSIZD, QELNCH/ 3*0.0, 16.0/
      DATA RAMP, REF, RK, ROTRAD/0.0, 20902703.0, 4.0, 22.0/
      DATA SCARF/0.0/
      DATA TDC,THTZD,TFINAL,TIME/3.0,0.,7.0,.09/
      DATA TLL,TPRINT,TROT/.09,.1,.092/
      DATA VINIT, WCF, WDF/68.0, 0.0, 0.0/
      DATA XMV, XTAR/146.5, 13000.0/
      DATA YAWMAL, YTAR/0.0, 0.0/
      DATA ZTAR/0.0/
      WRITE(6,3000) IDAY
```

```
      WRITE(6,36) ALTIN,DELTT,DELO2,DIVANG
      WRITE(6,37) DNSCH,DPIT,DT,DYAW
      WRITE(6,38) DZZ,EOL,FACTOR,FCA
      WRITE(6,39) FTORM,GREFF,HELIC,N
      WRITE(6,40) NN,PHIZD,PITMAL,PSIZD
      WRITE(6,41) QELNCH,RAMP,REF,RK
      WRITE(6,42) ROTRAD,SCARF,TDC,THTZD
      WRITE(6,43) TFINAL,TIME,TLL,TPRINT
      WRITE(6,44) TROT,VINIT,WCF,WDF
      WRITE(6,45) XMV,XTAR,YAWMAL,YTAR
      WRITE(6,46) ZTAR
C VINIT IS THE AIRSPEED;XTAR IS THE RANGE;QELNCH IS THE LAUNCHER
C     ELEVATION (ANGLE BETWEEN LAUNCHER CENTERLINE AND HELICOPTER
C     CENTERLINE);XMV IS THE MISSLE VELOCITY AT LAUNCHER EXIT; TIME,
C     TLL ARE VARIABLES OF TIME AT LAUNCHER EXIT, HELIC IS THE DOWNWASH
C  INDICATOR:1=DOWNWASH USED,0=NO DOWNWASH USED;FTORM IS UNIT USAGE:
C ENGLISH SYSTEM =0,METRIC SYSTEM=1.
      NAMELIST/PUTT/
     A ALTIN,
     B DELTT, DELO2, DIVANG,
     C DNSCH, DPIT, DT, DYAW, DZZ,
     D EOL,
     E FACTOR, FCA, FTORM, GREFF,
     F HELIC, N, NN,
     H PHIZD, PITMAL, PSIZD, QELNCH,
     I RAMP, REF, RK, ROTRAD,
     J SCARF,
     K TDC, THTZD, TFINAL, TIME,
     L TLL, TPRINT, TROT,
     M VINIT, WCF, WDF,
     N XMV, XTAR,
     O YAWMAL, YTAR,
     P ZTAR

      READ(5,PUTT)
C     THE DERIVATIVES ARE FOUND IN THE XX(1,J) SECTION OF THE XX ARRAY.
C   XX(2,J),S ARE INTEGRALS OF XX(1,J),S. THE FOLLOWING IS A LIST OF
C   THE DERIVATIVES BY XX(1,J) LOCATIONS.
C   XX(1,1)-ANGULAR ACCELERATION ABOUT THE MISSILE X-AXIS.
C   XX(1,2)-ANGULAR ACCELERATION ABOUT THE MISSILE Y-AXIS.
C   XX(1,3)-ANGULAR ACCELERATION ABOUT THE MISSILE Z-AXIS.
C   XX(1,4)-LINEAR ACCELERATION ALONG THE MISSILE X-AXIS.
C   XX(1,5)-LINEAR ACCELERATION ALONG THE MISSILE Y-AXIS.
C   XX(1,6)-LINEAR ACCELERATION ALONG THE MISSILE Z-AXIS.
C   XX(1,7)-LINEAR VELOCITY ALONG THE GROUND X(RANGE)-AXIS.
C   XX(1,8)-LINEAR VELOCITY ALONG THE GROUND Y(DEFLECTION)-AXIS.
C   XX(1,9)-LINEAR VELOCITY ALONG THE GROUND Z(VERTICAL)-AXIS.
C   XX(1,10)-DERIVATIVE OF THE EULER ANGLE SI = PSI
C   XX(1,11)-DERIVATIVE OF THE EULER ANGLE TH = THETA
C   XX(1,12)-DERIVATIVE OF THE EULER ANGLE FI = PHI.
C   XX(1,13)-DERIVATIVE OF THE MASS.
C   XX(1,14)-ACCELERATION ALONG LAUNCHER CENTERLINE.
C   XX(1,15)-VELOCITY ALONG LAUNCHER CENTERLINE.
C                  THIS IS THE INITIAL SECTION OF THE PROGRAM AND IS
C              ONLY PASSED THRU AT THE BEGINNING OF EACH RUN.

C   SET ARRAYS TO ZERO
      DO 50 I = 1, 3
      DO 50 J = 1, 50
      XX(I,J)=0.0
   50 Y(I,J) = 0.0
```

```
C     INITIALIZE BURNOUT VARIABLE
      BURNOUT = .FALSE.
C     WHEN THE VARIABLE LRATE IS TRUE,THE ROUND CG HAS CLEARED THE
C    LAUNCH TUBE.AT THIS TIME,THE MALLAUNCH RATES PITMAL AND YAWMAL ARE
C    ADDED TO THE BODY RATES.ASSUMPTION-ROUND LENGTH XLENG=LAUNCHER
C    LENGTH.
      LRATE = .FALSE.
C      INITIALIZE VARIABLE THAT SETS MISSILE PARAMETERS TO THER VALUES
C    WHEN BURNOUT OCCURS.
      SETBO = .FALSE.
C     INITIALIZE LOGICAL VARIABLE THAT ALLOWS PROGRAM TO SKIP SETTING
C    VALUES AT BURNOUT WHEN THE LOGICAL SETBO IS TRUE.
      SKIP = .FALSE.
C     CALCULATE TIME RATE OF CHANGE OF CENTER OF GRAVITY-CGDOT
      CGDOT = (CGO -CGF)/SUSOFF
C     INITIALIZE CENTER OF GRAVITY-CG
      CG = CGO
C     CALCULATE MASS AT TIME ZERO
      XMASS=WO/GREFF
      XX(2,13) = XMASS
C     CALCULATE TIME RATE OF CHANGE IN AXIAL AND TRANSVERSE MOMENTS
C    OF INERTIA-IXDOT, IYDOT
      IXDOT = (IXO - IXF)/SUSOFF
      IYDOT = (IYO - IYF)/SUSOFF
      IZDOT = IYDOT
C     CALCULATE SPECIFIC IMPULSE-ISP=TOTAL IMPULSE/PROPELLANT WEIGHT
      ISP = TI/(WO - WF)
C     SET FLAG FOR SUBROUTINE HELVEL
      IFL = 0
C     CALCULATE QUADRANT ELEVATION IN MILS AND AIRSPEED IN KNOTS
      QEMILS = QELNCH*1000./RADCON
      AIRSP = VINIT*3600./6080.
C     CALCULATE HELICOPTER ATTITUDE-HELAT-IN DEGS
      HELAT = 0.0
      IF(AIRSP .GE. 50.) HELAT = FCA*(AIRSP - 50.)/70.
C     CALCULATE REFERENCE AREA-AREFF
      AREFF = (PI*DREF*DREF)/4.
C     SET MISSILE ALTITUDE XX(2,9)=INITIAL ALTITUDE ALTIN
      XX(2,9) = ALTIN
C     QELNCH - ANGLE BETWEEN LAUNCHER CENTERLINE AND HELICOPTER
C    CENTERLINE
C     HELAT - ANGLE BETWEEN HELICOPTER CENTERLINE AND LOCAL HORIZONTAL
C     THL - ANGLE BETWEEN LAUNCHER CENTERLINE AND LOCAL HORIZONTAL
C     DIVANG - ANGLE BETWEEN HELICOPTER VELOCITY VECTOR VINIT AND LOCAL
C    HORIZONTAL
C     ALV - ANGLE BETWEEN LAUNCHER CENTERLINE AND HELICOPTER VELOCITY
C    VECTOR VINIT
      THL= (HELAT + QELNCH)/RADCON
      ALV = THL-(DIVANG/RADCON)
C     CALCULATE VELOCITIES IN THE BODY X AND Z DIRECTION DUE TO
C    HELICOPTER VELOCITY VINIT.
      XX(2,4) = VINIT*COS(ALV)
      XX(2,6) = VINIT*SIN(ALV)
C     SET THE EULER ANGLES TO THEIR INITIAL VALUES. XX(2,10) = SI,
C    XX(2,11)=TH,XX(2,12)=FI.
      XX(2,10) = PSIZD/RADCON
      XX(2,11) = THL
      XX(2,12) = PHIZD/RADCON
C     IF EOL(END OF LAUNCH)VARIABLE IS TRUE,TRAJECTORY BEGINS WITH
C    PHYSICAL PARAMETERS AT THE TIME THE ROUND CLEARS THE TUBE.
      IF(.NOT. EOL) GO TO 60
C    XMV=MISSILE VELOCITY AT LAUNCH
```

```
      XX(2,4) = VINIT*COS(ALV) + XMV
C   WL=WEIGHT OF ROUND AT TUBE EXIT.
      XMASS = WL/GREFF
      XX(2,13) = XMASS
C   SET THE BODY RATES Q AND R TO THE MALLAUNCH RATES
      XX(2,2) = PITMAL
      XX(2,3) = YAWMAL
C   SET LRATE TRUE TO INDICATE MALLAUNCH RATES HAVE BEEN ADDED
      LRATE = .TRUE.
C     CALCULATE CENTER OF GRAVITY AT END OF LAUNCH
      CG = CGO - TIME*CGDOT
   60 CONTINUE
C   INITIALIZE OTHER PARAMETERS
      IPRNT = 1
      TTO = SUSOFF + DELO2
      TEXIT = TLL + DELO2
C     DXLN, DYLN, DZLN REPRESENT THE XYZ COORDINATES OF THE CG OF THE
C   LAUNCHER W/R TO THE HELICOPTER ROTOR HUB.
      DXLN = 2.
      DYLN = 3.5
      DZLN = 7.
      DXX = 0.
      DYY = 0.
C
C   END OF INITIALIZATION SECTION OF MAIN PROGRAM
C
C          READ ONE CARD TO DESCRIBE PERTINENT FEATURES ABOUT PARTICULAR
C   TRAJECTORY RUN.
C  30 FORMAT(8A10)
C  CHANGED READ/WRITE LABEL TO COMMENT
C     READ(5,30) LABEL
C  31 FORMAT(1H1)
C     WRITE(6,31)
C     WRITE(6,30) LABEL
   32 FORMAT(1H0)
 3000 FORMAT(60X,I2,'/',I2,'/',I2,)
      WRITE(6,32)
      WRITE(6,25)
   25 FORMAT( '********** 2.75 MISSLE WITH M151 WARHEAD')
   34 FORMAT(20X,'TRAJECTORY RUN WAS MADE WITH HELICOPTER DOWNWASH PROFI
     1LE')
   35 FORMAT(20X,' $$$ NO HELICOPTER DOWNWASH PROFILE WAS USED IN THIS T
     1RAJECTORY RUN')
      IF(HELIC .EQ. 1.) WRITE(6,34)
      IF(HELIC .EQ. 0.) WRITE(6,35)
C     WRITE(6,PUTT)
C THE GROUP OF WRITE STATEMENTS THAT FOLLOW CAUSE A PRINTOUT DOWN
C  THE PAGE,IF ACROSS THE PAGE GO TO THE PRECEEDING WRITE,CWRITE(6,PUTT)
      WRITE(6,36) ALTIN,DELTT,DELO2,DIVANG
  36  FORMAT( ' ALTIN=',F6.2,2X,'DELTT=',E9.4,2X,'DELO2=',E9.4,2X,'DIVAN
     1G=',F5.2)
      WRITE(6,37) DNSCH,DPIT,DT,DYAW
  37  FORMAT( ' DNSCH=',F5.2,2X,'DPIT=',F5.2,2X,'DT=',E9.4,2X,'DYAW=',
     1F5.2)
      WRITE(6,38) DZZ,EOL,FACTOR,FCA
  38  FORMAT( ' DZZ=',F5.2,2X,'EOL=',L3,2X,'FACTOR=',F5.2,2X,'FCA=',
     1F8.3)
      WRITE(6,39) FTORM,GREFF,HELIC,N
  39  FORMAT( ' FTORM=',F5.2,2X,'GREFF=',F8.3,2X,'HELIC=',F3.2,2X,'N=',
     1I10.4)
      WRITE(6,40) NN,PHIZD,PITMAL,PSIZD
  40  FORMAT( ' NN=',I10.2,2X,'PHIZD=',F5.2,2X,'PITMAL=',F5.2,2X,'PSIZD=
```

```
     1',F5.2)
      WRITE(6,41) QELNCH,RAMP,REF,RK
   41 FORMAT( ' QELNCH=',F6.1,2X,'RAMP=',F5.2,2X,'REF=',E12.4,2X,'RK=',
     1F5.2)
      WRITE(6,42) ROTRAD,SCARF,TDC,THTZD
   42 FORMAT( ' ROTRAD=',F6.2,2X,'SCARF=',F5.2,2X,'TDC=',F5.2,2X,'THTZD=
     1',F5.2)
      WRITE(6,43) TFINAL,TIME,TLL,TPRINT
   43 FORMAT( ' TFINAL=',F5.2,2X,'TIME=',E9.4,2X,'TLL=',E9.4,2X,'TPRINT=
     1',F5.4)
      WRITE(6,44) TROT,VINIT,WCF,WDF
   44 FORMAT( ' TROT=',E9.4,2X,'VINIT=',F6.2,2X,'WCF=',F5.2,2X,'WDF=',
     1F5.2)
      WRITE(6,45) XMV,XTAR,YAWMAL,YTAR
   45 FORMAT( ' XMV=',F7.2,2X,'XTAR=',F9.2,2X,'YAWMAL=',F5.2,2X,'YTAR=',
     1F5.2)
      WRITE(6,46) ZTAR
   46 FORMAT( ' ZTAR='F5.2)
      WRITE(6,33) DIVANG, HELAT, QEMILS, AIRSP, WO
   33 FORMAT('   SIX DEGREE OF FREEDOM SIMULATION FOR 2.75 ROCKET WITH H
     1ELICOPTER DOWNWASH PROFILE'//'   HELICOPTER DIVE ANGLE = ',F7.2,'
     2DEGREES'/'   HELICOPTER ATTITUDE =',F6.2,' DEGREES'/'   LAUNCHER Q
     3.E. = ',F8.2,' MILS'/'   INDICATED AIRSPEED = ',F7.2,' KNOTS'/'
     4INITIAL ROCKET WEIGHT=',F6.2,'POUNDS'/)
C        THE PROGRAM NOW OFFERS A CHOICE BETWEEN 2ND ORDER AND 4TH
C   ORDER RUNGE -KUTTA INTEGRATION. IF RK=2.0,2ND ORDER RUNGE-KUTTA
C   INTEGRATION IS USED. THE  DEFAULT VALUE OF RK IS 4.0, AND CAN BE
C   CHANGED THRU NAMELIST PUTT.
      K2 = 2
      K=4
      TO=TIME
   85 IF(K2 - 2)2,9,2060
    7 IF(K-4)2,9,2060
 2060 WRITE(6,2061)
 2061 FORMAT(' ERROR -K EXCEEDS 4')
    9 SI = XX(2,10)
      TH = XX(2,11)
      FI = XX(2,12)
      SSI = SIN(SI)
      CSI = COS(SI)
      STH = SIN(TH)
      CTH = COS(TH)
      SFI = SIN(FI)
      CFI = COS(FI)
C        CALCULATE THE DCM MATRIX,THAT TRANSFORMS FROM GROUND TO BODY,
C   USING THE CURRENT VALUES OF THE EULER ANGLES.
      DCM(1,1) = CTH*CSI
      DCM(1,2) = CTH*SSI
      DCM(1,3) = -STH
      DCM(2,1) = STH*CSI*SFI - SSI*CFI
      DCM(2,2) = STH*SSI*SFI + CSI*CFI
      DCM(2,3) = CTH*SFI
      DCM(3,1) = STH*CSI*CFI + SSI*SFI
      DCM(3,2) = STH*SSI*CFI - CSI*SFI
      DCM(3,3) = CTH*CFI
C        CHANGE INTEGRATION DELTA T AT 3.0 SECONDS
      IF(TIME .GE. TDC) DELTT = DT
      DELO2 = 0.5*DELTT
C     COMPUTE THE COMPONENTS OF THE GRAVITATIONAL ACCELERATION ALONG
C   THE MISSILE XYZ AXES A GAX, GAY, GAZ.
      GAX = DCM(1,3)*GREFF
      GAY = DCM(2,3)*GREFF
```

```
      GAZ = DCM(3,3)*GREFF
C      ORTHOGONALITY CONSTRAINT
      CALL INVERT(DCM,ACTI)
      DO 84 I=1,3
      DO 84 J=1,3
   84 DCM(I,J)=(ACTI(J,I)+DCM(I,J))/2.
    2 IF(RAMP.NE.0.)WCF=RAMP*TIME
C     CALCULATE ROLL RATE,IN REV/SEC,AS FUNCTION OF TIME AS THE VARIABLE
C   ROLL.
      ROLL = TABLK(SPNRAT, SPINT, TIME, 30)
      XX(2,1) = ROLL*2.*PI
      IF(.NOT.LNEXIT) XX(2,1) = 0.0
C     THE COMPONENT OF THE TOTAL ROUND VELOCITY VECTOR ALONG THE MISSILE
C   X AXIS IS U=XX(2,4). U IS FOUND BY INTEGRATING ALL THE LINEAR
C   ACCELERATION TERMS. WHAT IS DESIRED IS ONLY THE LINEAR ACCELERATIONS
C   ALONG THE X AXIS THAT ARE DUE TO THRUST, DRAG, AND GRAVITY, IN
C   ORDER THAT WE CAN INTEGRATE TWICE AND OBTAIN THE DISTANCE THE ROUND
C   HAS TRAVELED ALONG THE LAUNCHER CENTERLINE. LET UDOTL BE THE ACCEL.
C   ALONG THE MISSILE X AXIS DUE ONLY TO THRUST,DRAG,AND GRAVITY.
      UDOTL = UMF1 + GAX
C     ESTABLISH UDOTL AS A DERIVATIVE.
      XX(1,14) = UDOTL
C     THE STATE,OR VELOCITY,OBTAINED BY INTEGRATING XX(1,14) IS XX(2,14)
C   ESTABLISH THE DERIVATIVE OF DISTANCE ALONG THE LAUNCHER CENTERLINE
C    THE VELOCITY, AS XX(1,15).
      UL = XX(2,14)
      XX(1,15) = UL
C     THE DISTANCE THE ROUND HAS TRAVELED ALONG THE LAUNCHER CENTERLINE
C   IS THE STATE XX(2,15),THE INTEGRAL OF XX(1,15).
      CGTRVL = XX(2,15)
C     THE ROUND WILL HAVE CLEARED THE LAUNCHER WHEN THE CENTER OF
C   GRAVITY HAS MOVED A DISTANCE = THE LENGTH OF THE ROUND,XLENG. THIS
C   WILL BE SIGNALED BY THE LOGICAL VARIABLE LNEXIT CHANGING FROM FALSE
C   TO TRUE.
      LNEXIT = CGTRVL .GE. XLENG
C     THIS SECTION ADDS THE MALLAUNCH RATE WHEN THE ROUND CG HAS
C   CLEARED THE LAUNCHER.ALSO,ADDS THE HELICOPTER VELOCITY IN X AND Z
C   AXES.
      IF(LRATE) GO TO 2062
      IF( .NOT.LNEXIT) GO TO 2062
      LRATE = .TRUE.
      XX(2,2) = XX(2,2) + PITMAL
      XX(2,3) = XX(2,3) + YAWMAL
      JUMP=3
      GO TO 8
 2062 REZIF=REF-XX(2,9)
      RF=SQRT(XX(2,7)**2+XX(2,8)**2+REZIF**2)
      HEIGHT=RF-REF
      IF(HEIGHT)4,5,5
    4 IF(HEIGHT+1.)3,6,6
    6 HEIGHT=0.
      GO TO 5
    3 WRITE(6,1501)
      JUMP=2
      GO TO 8
    5 SQIRT=SQRT(XX(2,7)**2+XX(2,8)**2)
      TANG=SQIRT/RF
      RGF=REF*ATAN(TANG/SQRT(1.-TANG**2))
      WIND=SQRT(WCF*WCF+WDF*WDF)
      IF(.NOT.LNEXIT) GO TO 230
      IF(HELIC.EQ.0.) GO TO 230
      DXX=0.
```

```
      DYY=0.
      DZZ=0.
      IF(TIME.GT..3) GO TO 230
      TPHASE=TIME+TROT
      XH=VINIT*COS(DIVANG)*TIME+DXLN
      YH=DYLN
      ZH=VINIT*SIN(DIVANG)*TIME-ALTIN+DZLN
      XMXH=XX(2,7)-XH
      YMYH=XX(2,8)-YH
      ZMZH=XX(2,9)+ZH
      DDIS=SQRT(XMXH**2+YMYH**2+ZMZH**2)
      THAT=DIVANG+HELAT
      THEO=0.
      IF(ZMZH.NE.0.) THEO=ATAN(XMXH/ZMZH)
      THTH=THEO-THAT
      XD=-DDIS*SIN(THTH)
      ZD=-DDIS*COS(THTH)
      YD=-YH
      DDISS=SQRT(XD*XD+YD*YD)
      IF(DDISS.LE.ROTRAD)CALL HELVEL(XD,YD,ZD,DWX,DWY,DWZ,TPHASE,IFL)
      IFL=1
      DXX=DWX*COS(THAT)-DWZ*SIN(THAT)
      DYY=DWY
      DZZ=-DWZ*COS(THAT)-DWX*SIN(THAT)
  230 CONTINUE
      IF(SQIRT.GT.ROTRAD) DZZ=0.
      IF(WIND.GT.0..AND.SQIRT.GT.0.) GO TO 10
      WXYZ(1,1)=WDF+DXX
      WXYZ(2,1)=WCF+DYY
      WXYZ(3,1)=DZZ
      GO TO 12
   10 CONTINUE
      WXYZ(1,1)=(-WCF*XX(2,8)+WDF*XX(2,7)*REZIF/RF)/SQIRT+DXX
      WXYZ(2,1)=(WCF*XX(2,7)+WDF*XX(2,8)*REZIF/RF)/SQIRT+DYY
      WXYZ(3,1)=WDF*SQIRT/RF+DZZ
   12 CALL MULTY(DCM,WXYZ,1,WUVW)
      IF(FTORM)223,222,223
  222 HEIGHT=HEIGHT/3.2808333
  223 HGTIND=0.0
      HGEOP=S*HEIGHT/(S+HEIGHT)
      IF(HGEOP-90000.)201,201,202
  201 WMOL=28.966
      GO TO 205
  202 IF(HGEOP-180000.)203,203,204
  203 WMOL=22.0-5.04483574*ATAN(6356.766*.04*HEIGHT/(6356.766+
     1HEIGHT)-8.8)
      GO TO 205
  204 WMOL=27.106-7.9356971*ATAN((HGEOP-180000.0)/140000.0)
  205 I=1
  206 I=I+1
      IF(HGEOPB(I)-HGEOP)207,209,210
  207 IF(I-13)206,208,208
  208 HGTIND=1.0
      GO TO 15
  209 TEMPMO=TEMPMB(I)
      HGEOPO=HGEOPB(I)
      DODH=DTDH(I)
      PRESO=PRESB(I)
      GO TO 211
  210 DODH=DTDH(I-1)
      PRESO=PRESB(I-1)
      HGEOPO=HGEOP3(I-1)
```

```
      TEMPMO=TEMPMB(I-1)
  211 TEMPM=TEMPMO+DODH*(HGEOP-HGEOPO)
      TEMP=TEMPM*WMOL/28.966
      IF(HEIGHT-90000.)216,216,217
  216 VSOUND=20.04633*SQRT(TEMPM)
      GO TO 218
  217 VSOUND=20.04633*SQRT(165.66)
  218 IF(DODH)212,213,212
  212 PRES=PRESO*(TEMPMO/(TEMPMO+DODH*(HGEOP-HGEOPO)))**(.034165/DODH)
      GO TO 214
  213 PRES=PRESO*EXP(-.034165*(HGEOP-HGEOPO)/TEMPMO)
  214 RHO=(F*PRES)/(TEMPM*10.1971)
      IF(FTORM.NE.0.) GO TO 15
      VSOUND=3.2808333*VSOUND
C   NOTE   DENSITY IS COMPUTED IN NEWTONS/M**3 OR SLUGS/FT**3
      RHO=RHO/515.375
      HEIGHT=3.2808333*HEIGHT
   15 IF(HGTIND.LE.0.) GO TO 20
      WRITE(6,1500)
      JUMP=2
      GO TO 8
 1500 FORMAT(38H HEIGHT EXCEEDS MAXIMUM VALUE IN TABLE)
 1501 FORMAT(19H HEIGHT IS NEGATIVE)
   20 XMW=XX(2,4)-WUVW(1,1)
      YMW=XX(2,5)-WUVW(2,1)
      ZMW=XX(2,6)-WUVW(3,1)
      VVW=YMW*YMW+ZMW*ZMW
      VRF=SQRT(VVW+XMW*XMW)
      XMACHN=VRF/VSOUND
      VALF=SQRT(VVW/(XMW**2))
      ALFTOT=ATAN(VALF)*RADCON
      IF(.NOT.LNEXIT) ALFTOT = 0.0
      BANK=1.5707
      IF(YMW.EQ.0.)BANK=0.
      IF(ZMW.NE.0.)BANK=ATAN2(YMW,ZMW)
      IF(XMW.EQ.0.) GO TO 503
      ALPPRN=ATAN2(ZMW,XMW)
      BETPRN=ATAN2(YMW,XMW)
      GO TO 510
  503 ALPPRN=0.
      BETPRN=0.
  510 IF(VRF .NE. 0.) GO TO 512
  511 FORMAT(10X,'RELATIVE VELOCITY VRF WAS ZERO-RUN STOPPED TO PREVENT
     1DIVISION BY ZERO')
      WRITE(6,511)
      STOP
  512 CONTINUE
C     CALCULATE NORMAL AND SIDE FORCE COEFFICIENTS CN AND CY AS
C   FUNCTIONS OF MACH NUMBER.
      CN = TABLK(CNAT, MCNA, XMACHN, 7)*ALFTOT
      CY = CN
C     MODIFY CN AND CY BY THE BANK ANGLE
      CN = -CN*COS(BANK)
      CY = -CY*SIN(BANK)
C     CHANGE BANK ANGLE TO DEGREES FOR PRINTING
      BANK = BANK*RADCON
C     CALCULATE THE AXIAL FORCE COEFFICIENT-CA-AS A FUNCTION OF MACH
C   NUMBER AND WETHER OR NOT THE MOTOR IS BURNING.
      IF(BURNOUT) GO TO 513
C     POWER ON DRAG COEFFICIENT - CA
      CA = TABLK(CDPON, MCDON, XMACHN, 21)
      GO TO 514
```

```
  513 CONTINUE
C       POWER OFF DRAG COEFFICIENT - CA
        CA = TABLK(CDPOFF, MCDOFF, XMACHN, 21)
  514 CONTINUE
C       CALCULATE THRUST-THRSTP-AS A FUNCTION OF TIME
        THRSTP = TABLK(THRUST, TTIME, TIME, 9)
C     CALCULATE THE AERODYNAMIC FORCES ALONG THE MISSILE XYZ AXES AS
C   FU1AP,FV1AP,FW1AP
C     DP = DYNAMIC PRESSURE
        DP =DNSCH*RHO*VRF**2
        DPS = DP*AREFF
        FU1AP = -CA*DPS
        FV1AP = CY*DPS
        FW1AP = CN*DPS
C     IN ORDER TO ALIGN THE BODY WITH THE THRUST,ROTATE FROM BODY TO
C   THRUST AXIS FIRST THRU DYAW ABOUT Z AXIS,THEN THRU DPIT ABOUT NEW
C   Y AXIS.
C     CALCULATE THE THRUST FORCES ALONG THE MISSILE XYZ AXES AS FU1GP,
C   FV1GP, FW1GP.
        FU1GP = THRSTP*COS(DPIT)*COS(DYAW)
        FV1GP = THRSTP*COS(DPIT)*SIN(DYAW)
        FW1GP = THRSTP*SIN(DPIT)
C     CALCULATE THE LINEAR ACCELERATIONS ALONG MISSILE XYZ AXES AS UMF1,
C   VMF1, WMF1 DUE TO THRUST AND DRAG.
        XMASS = XX(2,13)
        UMF1 = (FU1AP + FU1GP)/XMASS
        VMF1 = (FV1AP + FV1GP)/XMASS
        WMF1 = (FW1AP + FW1GP)/XMASS
C     DEFINE THE LINEAR AND ANGULAR VELOCITIES ALONG MISSILE XYZ AXES AS
C   U, V, W, P, Q, R.
        U = XX(2,4)
        V = XX(2,5)
        W = XX(2,6)
        P = XX(2,1)
        Q = XX(2,2)
        R = XX(2,3)
C     DEFINE THE TOTAL LINEAR ACCELERATIONS ALONG THE MISSILE XYZ AXES AS
C   UDOT, VDOT, WDOT.
        UDOT = UMF1 + GAX - Q*W + R*V
        VDOT = VMF1 + GAY - R*U + P*W
        WDOT = WMF1 + GAZ - P*V + Q*U
        XX(1,4) = UDOT
        XX(1,5) = VDOT
        XX(1,6) = WDOT
C       ZERO OUT LINEAR ACCELERATIONS ALONG MISSILE Y AND Z AXES UNTIL
C     ROUND HAS CLEARED LAUNCHER.
        IF(LNEXIT) GO TO 100
        XX(1,5) = 0.0
        XX(1,6) = 0.0
  100 CONTINUE
C     CALCULATE AXIAL AND TRANSVERSE MOMENTS OF INERTIA IX, AND IY=IZ.
C   CALCULATE THE CENTER OF GRAVITY CG.
        IF(SKIP) GO TO 130
        IF(.NOT. BURNOUT) GO TO 125
        IX = IXF
        IY = IYF
        IZ = IY
        IXDOT = 0.0
        IYDOT = 0.0
        CG = CGF
        MDOT = 0.0
        XMASS = WF/GREFF
```

```
      XMASS = XX(2,13)
      SKIP = .TRUE.
      GO TO 130
  125 IX = IX0 - IXDOT*TIME
      IY = IY0 - IYDOT*TIME
      IZ = IY
      CG = CG0 - CGDOT*TIME
  130 CONTINUE
C     CALCULATE MASS FLOW RATE MDOT AS A FUNCTION OF THRUST
      IF(SKIP) GO TO 140
      MDOT = -(THRSTP)/(ISP*GREFF)
  140 XX(1,13) = MDOT
C   CALCULATE CMQ,THE RATE DAMPING COEFFICIENT FOR RATES ABOUT MISSILE
C Y AND Z AXES, AS A FUNCTION OF MACH NO.
      CMQ = TABLK(CMQT, MCMQ, XMACHN, 16)
C   CALCULATE THE DAMPING MOMENTS(TORQUES) ABOUT THE MISSILE XYZ AXES
C AS XMP1AS, XMQ1AS, XMR1AS.
      AUX = (DPS*DREF**2)/(2.*VRF)
      XMP1AS = P*AUX*CLP
      XMQ1AS = Q*AUX*CMQ
      XMR1AS = R*AUX*CMQ
C   CALCULATE CENTER OF PRESSURE-CP-AS FUNCTION OF MACH NO. AND ANGLE
C OF ATTACH ALFTOT.
      CP = DREF*TABL2(CPT, MCP, CPAF, XMACHN, ALFTOT, 8, 2)
C   CALCULATE AERODYNAMIC MOMENT ARM-AMA
      AMA = CP - CG
C   CALCULATE THE  STABILITY MARGIN IN CALIBERS
      SM = AMA/DREF
C   CALCULATE THE AERODYNAMIC RESTORING TORQUES ABOUT THE MISSILE Y AND
C Z AXES AS XMAY, XMAZ.
      XMAY = FW1AP*AMA
      XMAZ = -FV1AP*AMA
C   CALCULATE TORQUES ABOUT MISSILE XYZ AXES DUE TO PRODUCT OF ANGULAR
C VELOCITIES ABOUT THE OTHER TWO AXES TIMES THE DIFFERENCE BETWEEN
C INERTIA OF THE OTHER AXES AS TAX, TAY, TAZ.
      TAX = Q*R*(IZ - IY)
      TAY = -P*R*(IX-IZ)
      TAZ = -P*Q*(IY-IX)
C   CALCULATE THE TORQUES ABOUT MISSILE XYZ AXES DUE TO TIME RATE OF
C CHANGE OF INERTIA, AS TIX, TIY, TIZ. TORQUE = DERIVATIVE OF
C ANGULAR MOMENTUM = (D/DT) IW = I*WDOT + IDOT*W
      TIX = P*IXDOT
      TIY = Q*IYDOT
      TIZ = R*IYDOT
C   CALCULATE THE TORQUES ABOUT THE MISSILE XYZ AXES DUE TO THRUST
C COMPONENTS ALONG THE MISSILE XYZ AXES, AS XMP1GS, XMQ1GS, XMR1GS.
      XMP1GS = THRSTP*SCARF
      ARM = CG - FACTOR
      XMQ1GS = ARM*FW1GP
      XMR1GS = -ARM*FV1GP
C   CALCULATE THE TOTAL TORQUES ABOUT THE MISSILE XYZ AXES AS TORQX,
C TORQY, TORQZ.
      TORQX = XMP1AS + XMP1GS + TAX -TIX
      TORQY = XMQ1AS + XMQ1GS + XMAY + TAY - TIY
      TORQZ = XMR1AS + XMR1GS + XMAZ + TAZ - TIZ
C     ZERO OUT THE TORQUES ABOUT MISSILE Y AND Z ACES UNTIL ROUND
C   HAS CLEARED THE LAUNCHER. THIS PUTS THOSE ANGULAR ACCELERATIONS TO
C   ZERO.
      IF(LNEXIT) GO TO 300
      TORQY = 0.0
      TORQZ = 0.0
  300 CONTINUE
```

```
C    CALCULATE ANGULAR ACCELERATIONS ABOUT MISSILE XYZ AXES AS PDOT,
C QDOT, AND RDOT. THE ANGULAR ACCELERATION ABOUT ANY AXIS IS THE
C TOTAL TORQUE DIVIDED BY THE MOMENENT OF INERTIA.
      PDOT = TORQX/IX
      QDOT = TORQY/IY
      RDOT = TORQZ/IZ
C    XX(2,1) IS THE ROLL RATE  AND WOULD NORMALLY BE CALCULATED BY
C EQUATING XX(1,1) TO PDOT,THE ANGULAR ACCELERATION,AND INTEGRATING
C XX(1,1). AT THE PRESENT XX(2,1) IS CALCULATED FROM A TABLE OF ROLL
C VERSUS TIME, SO XX(1,1) WILL NOT BE EQUATED TO PDOT.
      XX(1,2) = QDOT
      XX(1,3) = RDOT
C     CALCULATE THE MISSILE VELOCITIES IN THE GROUND FRAME-XDOT,YDOT,
C    ZDOT-AS XX(1,7), XX(1,8), XX(1,9)
      DO 305 J=1,3
  305 XX(1,J+6)=DCM(1,J)*XX(2,4)+DCM(2,J)*XX(2,5)+DCM(3,J)*XX(2,6)
      PNOM=SQRT(ZMW**2+YMW**2)
      TWOPI = 2.*PI
      PHII=AMOD(XX(2,20),TWOPI)
      SNPHI=SIN(PHII)
      CSPHI=COS(PHII)
      IF(XMW)310,311,310
  311 IF(PNOM)312,313,312
  312 ETAPRN=1.5707963
      GO TO 314
  313 ETAPRN=0.0
      GO TO 314
  310 ETAPRN=ATAN(PNOM/XMW)
  314 ALPHPR=BETPRN*SNPHI+ALPPRN*CSPHI
      BETAPR=BETPRN*CSPHI-ALPPRN*SNPHI
C     CALCULATE THE EULER ANGLE DERIVATIVES SIDOT, THDOT, AND FIDOT AND
C    EQUATE THEM TO XX(1,10), XX(1,11), XX(1,12).
      SIDOT = (SFI*Q +CFI*R)/CTH
      THDOT = CFI*Q - SFI*R
      FIDOT = P + (SFI*STH*Q + CFI*STH*R)/CTH
      XX(1,10) = SIDOT
      XX(1,11) = THDOT
      XX(1,12) = FIDOT
      JUMP=1
      IF(TIME.GE.TLL.AND.TIME.LT.TEXIT) GO TO 5030
      IF(TIME.GE.SUSOFF.AND.TIME.LT.TTO) GO TO 5030
      IF(K-4)23,8,2060
    8 IF(TIME.LE.0.) GO TO 5010
      GO TO 5020
 5010 NPAGE=0
      LPAGE=0
      TPR=TIME-TPRINT
 5020 IF(JUMP.NE.1) GO TO 5030
      IF(TIME-(TPR+TPRINT))5040,5030,5030
 5030 ALPTD=RADCON*ALPPRN
      BETTD=RADCON*BETPRN
      ETATD=RADCON*ETAPRN
      ALF=ALPHPR
      BETO=BETAPR
      IF(XX(2,9).GE.0.) GO TO 2002
      TP1=TIME
      XP1=XX(2,7)
      YP1=XX(2,8)
      ZP1=XX(2,9)
      GO TO 2003
```


```
 2002 IF(IPRNT.EQ.2)GO TO 2003
      XP2=XX(2,7)
```

```
      YP2=XX(2,8)
      ZP2=XX(2,9)
      TP2=TIME
      IPRNT=2
      DOLL=ZP1/(ZP2-ZP1)
      XMP=XP1-DOLL*(XP2-XP1)
      YMP=YP1-DOLL*(YP2-YP1)
      TMP=TP1-DOLL*(TP2-TP1)
      WRITE(6,2005) TMP,XMP,YMP
 2005 FORMAT(/' TIME OF IMPACT=',F8.4,' IMPACT RANGE=',F10.3,
     1' IMPACT CROSS RNG=',F9.3/)
C        WRITE THE VALUES OF THE VARIABLES AT GROUND IMPACT TO THE
C   PLOT FILE TAPE2.
C        THIS SECTION OF CODE IS ONLY REACHED WHEN THE ALTITUDE XX(2,9)
C   HAS GONE POSTIVE,MEANING  THAT GROUND IMPACT HAS OCCURED.
      WRITE(2)
     A TIME, XX(2,7), XX(2,8), XX(2,9), VRF,
     B THRSTP, XX(1,7), XX(1,8), XX(1,9), XMACHN,
     C XMASS, XX(2,4), XX(2,5), XX(2,6), MDOT,
     D SM, XX(1,4), XX(1,5), XX(1,6), DP,
     E RHO, PID, QID, RID, CA,
     F ALFTOT, PID1, QID1, RID1, CN,
     G ALPTD, TORQX, TORQY, TORQZ, CY,
     H BETTD, WINX, WINY, WINZ, CLP,
     I XMQ1AS, XMAY, TAY, TIY, CMQ,
     J XMR1AS, XMAZ, TAZ, TIZ, BANK,
     K SI, TH, FI, UMF1, VMF1,
     L WMF1, GAX, GAY, GAZ, DIST
C        SET THE LOGICAL VARIABLE IMPACT=.TRUE. TO STOP TRAJECTORY.
      IMPACT  = .TRUE.
 2003 CONTINUE
C     CALCULATE ROLL RATE AS PID IN REV/SECS
      PID = XX(2,1)/TWOPI
      WINX=XX(1,7)+WXYZ(1,1)
      WINY=XX(1,8)-WXYZ(2,1)
      WINZ=WXYZ(3,1)-XX(1,9)
      ALFO=0.
      IF(WINX.NE.0.) ALFO=ATAN2(WINZ,WINX)
      QID= RADCON* XX(2,2)
      RID= RADCON*XX(2,3)
      PID1= RADCON* XX(1,1)
      DIST=SQRT((XTAR-XX(2,7))**2+(YTAR-XX(2,8))**2+(ZTAR+XX(2,9))**2)
      QID1= RADCON*XX(1,2)
      RID1= RADCON* XX(1,3)
      SI = XX(2,10)*RADCON
      TH = XX(2,11)*RADCON
      FI =AMOD(XX(2,12),TWOPI)*RADCON
      IF(MOD(NPAGE,4).NE.0)GO TO 5050
      WRITE(6,5005)
      LPAGE=LPAGE+1
      WRITE(6,5006) IDAY, LPAGE
 5050 WRITE(6,5007)
      IF(JUMP.EQ.1) TPR=TIME
      NPAGE=NPAGE+1
      WRITE(6,4000) TIME, XX(2,7), XX(2,8), XX(2,9), VRF
 4000 FORMAT(2X,'TIME   =',E14.8,2X,'RANGE =',E14.8,2X,'DEFL   =',E14.8,2X
     *,'ALT    =',E14.8,2X,'VELR  =',E14.8)
      WRITE(6,4010) THRSTP, XX(1,7), XX(1,8), XX(1,9), XMACHN
 4010 FORMAT(2X,'THRUST=',E14.8,2X,'XDOT  =',E14.8,2X,'YDOT   =',E14.8,2X
     *,'ZDOT   =',E14.8,2X,'MACH  =',E14.8)
      WRITE(6,4020) XMASS, XX(2,4), XX(2,5), XX(2,6), MDOT
 4020 FORMAT(2X,'MASS   =',E14.8,2X,'UUU   =',E14.8,2X,'VVV    =',E14.8,2X
```

```
     *,'WWW   =',E14.8,2X,'MDOT  =',E14.8)
      WRITE(6,4030) SM, XX(1,4), XX(1,5), XX(1,6), DP
 4030 FORMAT(2X,'SM     =',E14.8,2X,'UDOT  =',E14.8,2X,'VDOT  =',E14.8,2X
     *,'WDOT  =',E14.8,2X,'DP    =',E14.8)
      WRITE(6,4040) RHO, PID, QID, RID, CA
 4040 FORMAT(2X,'RHO    =',E14.8,2X,'P     =',E14.8,2X,'Q     =',E14.8,2X
     *,'R     =',E14.8,2X,'CA    =',E14.8)
      WRITE(6,4050) ALFTOT, PID1, QID1, RID1, CN
 4050 FORMAT(2X,'ALFTOT=',E14.8,2X,'PID1  =',E14.8,2X,'QID1  =',E14.8,2X
     *,'RID1  =',E14.8,2X,'CN    =',E14.8)
      WRITE(6,4060) ALPTD, TORQX, TORQY, TORQZ, CY
 4060 FORMAT(2X,'ALPTD =',E14.8,2X,'TORQX =',E14.8,2X,'TORQY =',E14.8,2X
     *,'TORQZ =',E14.8,2X,'CY    =',E14.8)
      WRITE(6,4070) BETTD, WINX, WINY, WINZ, LNEXIT
 4070 FORMAT(2X,'BETTD =',E14.8,2X,'WINX  =',E14.8,2X,'WINY  =',E14.8,2X
     *,'WINZ  =',E14.8,2X,'LNEXIT=',L3)
      WRITE(6,4080) XMQ1AS, XMAY, TAY, TIY, CMQ
 4080 FORMAT(2X,'XMQ1AS=',E14.8,2X,'XMAY  =',E14.8,2X,'TAY   =',E14.8,2X
     *,'TIY   =',E14.8,2X,'CMQ   =',E14.8)
      WRITE(6,4090) XMR1AS, XMAZ, TAZ, TIZ, BANK
 4090 FORMAT(2X,'XMR1AS=',E14.8,2X,'XMAZ  =',E14.8,2X,'TAZ   =',E14.8,2X
     *,'TIZ   =',E14.8,2X,'BANK  =',E14.8)
      WRITE(6,4100) SI, TH, FI, UMF1, VMF1
 4100 FORMAT(2X,'PSI    =',E14.8,2X,'THETA =',E14.8,2X,'PHI   =',E14.8,2X
     *,'UMF1  =',E14.8,2X,'VMF1  =',E14.8)
      WRITE(6,4110) WMF1, GAX, GAY, GAZ, CGTRVL
 4110 FORMAT(2X,'WMF1  =',E14.8,2X,'GAX   =',E14.8,2X,'GAY   =',E14.8,2X
     *,'GAZ   =',E14.8,2X,'CGTRVL=',E14.8)
C        STOP PROGRAM WHEN IMPACT IS TRUE. IMPACT IS TRUE WHEN ALTITUDE
C   XX(2,9) GOES POSTIVE.
      IF(IMPACT)STOP'GROUND IMPACT'
 5005 FORMAT(1H1)
 5006 FORMAT(5X,'6-D TRAJECTORY OF 2.75 MK66I2, M151 WARHEAD',10X,I2,'/'
     !,I2,'/',I2,10X,'PAGE',I2)
 5007 FORMAT(1HO)
      WRITE(2)
     A TIME, XX(2,7), XX(2,8), XX(2,9), VRF,
     B THRSTP, XX(1,7), XX(1,8), XX(1,9), XMACHN,
     C XMASS, XX(2,4), XX(2,5), XX(2,6), MDOT,
     D SM, XX(1,4), XX(1,5), XX(1,6), DP,
     E RHO, PID, QID, RID, CA,
     F ALFTOT, PID1, QID1, RID1, CN,
     G ALPTD, TORQX, TORQY, TORQZ, CY,
     H BETTD, WINX, WINY, WINZ, CLP,
     I XMQ1AS, XMAY, TAY, TIY, CMQ,
     J XMR1AS, XMAZ, TAZ, TIZ, BANK,
     K SI, TH, FI, UMF1, VMF1,
     L WMF1, GAX, GAY, GAZ, DIST
C
C   THESE EQUATIONS PERFORM 4TH ORDER RUNGE-KUTTA INTEGRATION
C
 5040 IF(JUMP-2)23,21,2062
   23 IF(RK .EQ. 2.0) GO TO 300
      IF(K .EQ. 4) KN = 0
      KN=KN+1
      DO 22 J=1,N
      CK(KN,J)=XX(1,J)*DELTT
   22 CONTINUE
      GO TO (407,409,411,420),K
  420 K=1
      DO 404 I=1,2
      DO 404 J=1,N
```

```
  404 Y(I,J)=XX(I,J)
      DO 405 J=NN,N
  405 XX(2,J)=CK(1,J)/2.+Y(2,J)
      TIME=TIME+DELO2
      GO TO 413
  407 K=2
      DO 408 J=NN,N
  408 XX(2,J)=CK(2,J)/2.+Y(2,J)
      GO TO 413
  409 K=3
      DO 410 J=NN,N
  410 XX(2,J)=CK(3,J)+Y(2,J)
      TIME=TIME+DELO2
      GO TO 413
  411 K=4
      DO 412 J=NN,N
  412 XX(2,J)=(CK(1,J)+2.*CK(2,J)+2.*CK(3,J)+CK(4,J))/6.+Y(2,J)
  413 CONTINUE
      GO TO 598
C         THESE EQUATIONS PERFORM 2ND ORDER RUNGE-KUTTA INTEGRATION
  800 CONTINUE
      GO TO(900, 850) K2
  850 K2 = 1
      DO 860 I = 1, 2
      DO 860 J = NN, N
  860 Y(I,J) = XX(I,J)
C         Y(1,J)= DERIVATIVE AT THE BEGINNING OF THE TIME STEP AND
C    Y(2,J)=THE STATE.
      DO 870 J = NN, N
  870 XX(2,J) = Y(2,J) + 0.6666667*DELTT*Y(1,J)
      TIME = TIME + DELTT*0.6666667
      GO TO 1000
  900 K2 = 2
      DO 910 J = NN, N
  910 XX(2,J) = Y(2,J) + 0.25*DELTT*(Y(1,J) + 3.*XX(1,J))
      TIME = TIME + DELTT*0.3333333
      GO TO 1000
  598 CONTINUE
C     CALCULATE LOGICAL VARIABLE BURNOUT
      BURNOUT = TIME .GE. SUSOFF
 1000 IF(HEIGHT) 600, 1010, 1010
 1010 IF(RK .EQ. 4.0) GO TO 1020
      IF(TIME - TFINAL)85,85,700
 1020 IF(TIME - TFINAL)7,7,700
  700 WRITE(6,701)
  701 FORMAT(' TIME EXCEEDS MAX,RUN NEXT CASE')
  600 JUMP=2
C         TIME OF FLIGHT HAS EXCEEDED ALLOTED RUNTIME TFINAL. SET NOTIME
C    EQUAL TO TRUE TO STOP PROGRAM.
      NOTIME  = .TRUE.
      GO TO 8
   21 STOP'FLT.TIME GREATER THAN RUN TIME'
      END
$PROG TABLK
$XREF
      FUNCTION TABLK(X,Y,XNUMB,LENGTH)
      DIMENSION X(30),Y(30)
      DO 1 I=2,LENGTH
      IF(XNUMB-Y(I))3,3,1
    1 CONTINUE
      I=LENGTH
    3 TABLK=X(I-1)+(XNUMB-Y(I-1))*(X(I)-X(I-1))/(Y(I)-Y(I-1))
```

```
      RETURN
      END
$PROG MULTY
$XREF
      SUBROUTINE MULTY(A,B,N,C)
      DIMENSION A(3,3),B(3,3),C(3,3)
      DO 1 L=1,N
      DO 1 I=1,3
    1 C(I,L)=0.
      DO 2 L=1,N
      DO 2 I=1,3
      DO 2 J=1,3
    2 C(I,L)=C(I,L)+A(I,J)*B(J,L)
      RETURN
      END
$PROG INVERT
$XREF
      SUBROUTINE INVERT(A,C)
      DIMENSION A(3,3),C(3,3)
      C(1,1)=A(2,2)*A(3,3)-A(2,3)*A(3,2)
      C(1,2)=A(1,3)*A(3,2)-A(1,2)*A(3,3)
      C(1,3)=A(1,2)*A(2,3)-A(1,3)*A(2,2)
      C(2,1)=A(2,3)*A(3,1)-A(2,1)*A(3,3)
      C(2,2)=A(1,1)*A(3,3)-A(1,3)*A(3,1)
      C(2,3)=A(1,3)*A(2,1)-A(1,1)*A(2,3)
      C(3,1)=A(2,1)*A(3,2)-A(2,2)*A(3,1)
      C(3,2)=A(1,2)*A(3,1)-A(1,1)*A(3,2)
      C(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)
      DETER=A(1,1)*C(1,1)+A(1,2)*C(2,1)+A(1,3)*C(3,1)
      DO 5 I=1,3
      DO 5 J=1,3
    5 C(I,J)=C(I,J)/DETER
      RETURN
      END
$PROG TABL2
$XREF
      FUNCTION TABL2(TAB,TMAC,TALF,X,Y,NO1,NO2)
      DIMENSION TAB(17,5),TMAC(30),TALF(30)
      DO 1 I=2,NO1
      IF(X.LT.TMAC(I)) GO TO 2
    1 CONTINUE
      I=NO1
    2 CONTINUE
      DO 3 J=2,NO2
      IF(Y.LT.TALF(J)) GO TO 4
    3 CONTINUE
      J=NO2
    4 CONTINUE
      PALF=(Y-TALF(J-1))/(TALF(J)-TALF(J-1))
      PMAC=(X-TMAC(I-1))/(TMAC(I)-TMAC(I-1))
      P1=TAB(I-1,J-1)+PALF*(TAB(I-1,J)-TAB(I-1,J-1))
      P2=TAB(I,J-1)+PALF*(TAB(I,J)-TAB(I,J-1))
      TABL2=P1+(P2-P1)*PMAC
      RETURN
      END
$PROG HELVEL
$XREF
      SUBROUTINE HELVEL(XCALL,YCALL,ZCALL,VX,VY,VZ,TIME,IREAD)
      COMMON /KEEP/ RAD,XNRB,ORL,NUMREC,RROT,DT,PERIOD
      COMMON/FUSE/ NPTS,  VXINF(160),  VZINF(160)
     1,XBAR(160),   YBAR(160),   ZBAR(160),   SIGX(160),   SIGZ(160)
     2, DDJ(160,4), ETAK(160,4),  XIK(160,4)
```

```
     3, XLX(160),    XMX(160),    XNX(160)
     4, XLE(160),    XME(160),    XNE(160)
     5, XLZ(160),    XMZ(160),    XNZ(160)
      COMMON /VORT/ X(6,90,2),Y(6,90,2),Z(6,90,2),A(6,90,2),
     1 GAMA(6,90,2),SEG(6,90,2),GAMB1(100),VXF,VYF,VZF,NPS(6),
     2 NA,NB,NW,NAB,XMCL,XMSL
      DIMENSION DUM(90,2),DUMM(160),DUMMY(100),DUMY(6)
      IF(IREAD .NE. 0) GO TO 50
      RAD=.01745329252
      REWIND 10
      READ(10,115) NREC
      NUMREC=NREC+1
      REWIND 10
      DO 10 I=1,NUMREC
      READ(10,115) IDUM
      READ(10,155) ((X(I,J,K),J=1,90),K=1,2),((Y(I,J,K),J=1,90),K=1,2)
     *,((Z(I,J,K),J=1,90),K=1,2),((GAMA(I,J,K),J=1,90),K=1,2)
     *,((A(I,J,K),J=1,90),K=1,2),((SEG(I,J,K),J=1,90),K=1,2)
      READ(10,135) PSI,PSIF,DPSI,NPS(I)
   10 CONTINUE
      READ(10,145) D,D,RROT,D,D,NRB,NPTS
      READ(10,155) DUMM,DUMM,D,D,D,D
      READ(10,165) D,NA,NB,NAB,NW,D
      READ(10,155) D,D,D,XMCL,XMSL,XLAM,D,D,D,D,D,D,D,D,D,DUMMY,GAMB1,
     *XBAR,YBAR,ZBAR,SIGX,SIGZ,DDJ,XIK,ETAK,XLX,XMX,XNX,XLE,XME,XNE,
     *XLZ,XMZ,XNZ,DUMMY,DUMMY,DUMY,VXINF,VZINF
115   FORMAT(I10)
  135 FORMAT(3(1E12.5),I10)
145   FORMAT(5(1PE12.5),2I8)
155   FORMAT(6(1PE12.5))
165   FORMAT(1PE12.5,4I10,1PE12.5)
      XNRB=NRB
      ORL=XNRB*6.*RAD*RROT*XLAM
      REWIND 10
      PERIOD=60./XNRB/FLOAT(NB)
      DT=PERIOD/FLOAT(NUMREC)
   50 RTIME=AMOD(TIME,PERIOD)
      IF(RTIME .LT. 0.) RTIME=RTIME+PERIOD
      DEX=RTIME/DT
      FRAC=AMOD(DEX,1.)
      FRACC=1.-FRAC
      M=DEX+1
      XX=XCALL/RROT
      YY=YCALL/RROT
      ZZ=ZCALL/RROT
      CALL FUSLGE(XX,YY,ZZ,VXF,VYF,VZF)
      CALL VLCTY(XX,YY,ZZ,V1,V2,V3,M)
      CALL VLCTY(XX,YY,ZZ,V11,V22,V33,M+1)
      VX=(V1*FRACC+V11*FRAC)*ORL
      VY=(V2*FRACC+V22*FRAC)*ORL
      VZ=(V3*FRACC+V33*FRAC)*ORL
      RETURN
      END
$PROG VLCTY
$XREF
      SUBROUTINE VLCTY(XIPT,YIPT,ZIPT,VX,VY,VZ,MM)
      COMMON /VORT/ X(6,90,2),Y(6,90,2),Z(6,90,2),A(6,90,2),
     1 GAMA(6,90,2),SEG(6,90,2),GAMB1(100),VXF,VYF,VZF,NPS(6),
     2 NA,NB,NW,NAB,XMCL,XMSL
      M=MOD(MM-1,6)+1
      VX=0.
      VY=0.
```

```
      VZ=0.
      XXX=XIPT
      YYY=YIPT
      ZZZ=ZIPT
      DO 50  J=1,NB
      SIG2=SQRT((XXX-X(M,1,J))**2+(YYY-Y(M,1,J))**2+(ZZZ-Z(M,1,J))**2)
      DO 40  K=1,NW
      SIG1=SIG2
      SIG2=SQRT((XXX-X(M,K+1,J))**2+(YYY-Y(M,K+1,J))**2
     *+(ZZZ-Z(M,K+1,J))**2)
      SEGSQ=SEG(M,K,J)**2
      HM1   = SIG1**2+SIG2**2
      IF(HM1.GT.SEGSQ)GO TO 30
      HM2 = .25*((SIG1+SIG2)**2-SEGSQ)*(SEGSQ-(SIG1-SIG2)**2)/SEGSQ
      IF(HM2 .GT. A(M,K,J)**2) GO TO 30
      GGG=GAMA(M,K,J)/SEG(M,K,J)
      GO TO 31
   30 GGG=GAMA(M,K,J)*(SIG1+SIG2)/(SIG1*SIG2*((SIG1+SIG2)**2-SEGSQ))
   31 DIFX=XXX-X(M,K,J)
      DIFY=YYY-Y(M,K,J)
      DIFZ=ZZZ-Z(M,K,J)
      XNU1=DIFY*(Z(M,K,J)-Z(M,K+1,J))-DIFZ*(Y(M,K,J)-Y(M,K+1,J))
      XNU2=DIFZ*(X(M,K,J)-X(M,K+1,J))-DIFX*(Z(M,K,J)-Z(M,K+1,J))
      XNU3=DIFX*(Y(M,K,J)-Y(M,K+1,J))-DIFY*(X(M,K,J)-X(M,K+1,J))
      VX    = VX   +XNU1*GGG
      VY    = VY   +XNU2*GGG
      VZ    = VZ   +XNU3*GGG
   40 CONTINUE
   50 CONTINUE
      SIG1  = SQRT(XXX**2+YYY**2+ZZZ**2)
      DO 60  L=1,NB
      LPS=MOD(NPS(M)+(NA*(L-1))/NB+NAB,NA)
      IF(LPS.EQ.0)       LPS = NA
      XNU1=ZZZ*Y(M,1,L)-YYY*Z(M,1,L)
      XNU2=XXX*Z(M,1,L)-ZZZ*X(M,1,L)
      XNU3=YYY*X(M,1,L)-XXX*Y(M,1,L)
      SIG2=SQRT((XXX-X(M,1,L))**2+(YYY-Y(M,1,L))**2+(ZZZ-Z(M,1,L))**2)
      DEN   = SIG1*SIG2*((SIG1+SIG2)**2-1.0)
      IF(DEN.EQ.0.0) DEN = .001
      GGG   = GAMB1(LPS)*(SIG1+SIG2)/DEN
      VX=VX+XNU1*GGG
      VY=VY+XNU2*GGG
      VZ=VZ+XNU3*GGG
   60 CONTINUE
      VX=VX + XMCL +VXF
      VY=VY +VYF
      VZ=VZ -XMSL +VZF
      RETURN
      END
$PROG FUSLGE
$XREF
      SUBROUTINE FUSLGE(XD,YD,ZD,UFD,VFD,WFD)
      COMMON/FUSE/ NPTS,  VXINF(160),  VZINF(160)
     1,XBAR(160),   YBAR(160),   ZBAR(160),   SIGX(160),   SIGZ(160)
     2, DDJ(160,4), ETAK(160,4),  XIK(160,4)
     5, XLX(160),      XMX(160),      XNX(160)
     6, XLE(160),      XME(160),      XNE(160)
     7, XLZ(160),      XMZ(160),      XNZ(160)
      DIMENSION    RJ(4),    EJ(4),    HJ(4),     EMJ(4)
      SUMU = 0.0
      SUMV = 0.0
      SUMW = 0.0
```

```
      IF(NPTS.EQ.0)      GO TO 13
      DO 12  J=1,NPTS
      NFLG = 1
      XB = XD-XBAR(J)
      YB = YD-YBAR(J)
      ZB = ZD-ZBAR(J)
      D6=(XIK(J,4)-XIK(J,2))**2+(ETAK(J,4)-ETAK(J,2))**2
      D5=(XIK(J,3)-XIK(J,1))**2+(ETAK(J,3)-ETAK(J,1))**2
      D7 = AMAX1(D5,D6)
3     XI    = XLX(J) * XB + XMX(J) * YB + XNX(J) * ZB
      ETA   = XLE(J) * XB + XME(J) * YB + XNE(J) * ZB
      ZETA  = XLZ(J) * XB + XMZ(J) * YB + XNZ(J) * ZB
      RO = XI**2+ETA**2+ZETA**2
      TJ = RO/D7
      IF(RO.NE.0.0)   GO TO 4
      VXI   = 0.0
      VETA  = 0.0
      VZETA = 6.2831853072
      GO TO 10
4     IF( TJ.LT.6.)      GO TO 5
      SJ=.5*(XIK(J,3)-XIK(J,1))*(ETAK(J,2)-ETAK(J,4))/(RO*SQRT(RO))
      VXI   = SJ *  XI
      VETA  = SJ *  ETA
      VZETA = SJ * ZETA
      GO TO 10
5     VXI   = 0.0
      VETA  = 0.0
      VZETA = 0.0
      DO 9  K=1,2
      DO 9  I=1,4
      I1    = I + 1
      IF(I.EQ.4) I1 = 1
      TRMX  =  XIK(J,I1)- XIK(J,I)
      TRME  = ETAK(J,I1)-ETAK(J,I)
      IF(K.EQ.2)         GO TO 7
      TRM1  =  XI -  XIK(J,I)
      TRM2  = ETA - ETAK(J,I)
      HJ(I) =  TRM1 * TRM2
      EJ(I) =  TRM1**2 + ZETA**2
      RJ(I) = (TRM2**2 + EJ(I))**.5
      IF(TRMX.EQ.0.0)    TRMX = 1.0E-6
      EMJ(I)=  TRME / TRMX
      GO TO 9
7     TRMR = (RJ(I)+RJ(I1)-DDJ(J,I))/(RJ(I)+RJ(I1)+DDJ(J,I))
      IF(TRMR.LT.0.) WRITE(6,99) J,I,XD,YD,ZD,TRMR,RJ(I),RJ(I1),DDJ(J,I)
99    FORMAT( 4X,52HTRMR IS NEG--J,I,XD,YD,ZD,TRMR,RJ(I),RJ(I1),DDJ(J,I.
     1/(2I5,7E15.5))
      TRMR  = ALOG(ABS(TRMR))
      TRME  = TRME / DDJ(J,I)
      TRMX  =-TRMX / DDJ(J,I)
      VXI   = VXI  + TRME * TRMR
      VETA  = VETA + TRMX * TRMR
      IF(ZETA.EQ.0.0)   GO TO 9
      VZETA = VZETA +  ATAN((EMJ(I)*EJ(I) -HJ(I) )/(ZETA*RJ(I) )) -
     *                 ATAN((EMJ(I)*EJ(I1)-HJ(I1))/(ZETA*RJ(I1)))
9     CONTINUE
10    VVX = XLX(J)*VXI+XLE(J)*VETA+XLZ(J)*VZETA
      VVY = XMX(J)*VXI+XME(J)*VETA+XMZ(J)*VZETA
      VVZ = XNX(J)*VXI+XNE(J)*VETA+XNZ(J)*VZETA
      IF(NFLG.EQ.2)      GO TO 11
      VVVX = VVX
      VVVY = VVY
```

```
      VVVZ = VVZ
      YB = -YD-YBAR(J)
      NFLG = 2
      GO TO 3
   11 TRM = SIGX(J)*VXINF(J)+SIGZ(J)*VZINF(J)
      SUMU = SUMU+TRM*(VVVX+VVX)
      SUMV = SUMV+TRM*(VVVY-VVY)
      SUMW = SUMW+TRM*(VVVZ+VVZ)
   12 CONTINUE
   13 UFD = SUMU
      VFD = SUMV
      WFD = SUMW
      RETURN
      END
$BEND
```

APPENDIX E

RPLOT Listing

```
      PROGRAM RPLOT(TAPE5,TAPE6,TAPE15,TAPE20,INPUT,OUTPUT)
******* THIS PROGRAM WRITES DATA FROM UNFORMATTED PLOT FILE

      DIMENSION A(60)
      REAL NO
      DATA YES,NO/3HYES,3H NO/
      REWIND 15
      CALL CONNEC(5)
      CALL CONNEC(6)
   50 FORMAT(5X,*IS DATA TAPE FORMATTED ? ENTER YES OR NO*)
      WRITE(6,50)
   51 FORMAT(A3)
      READ(5,51)ANS
      IF(ANS .EQ. NO)GO TO 70
      READ(15)NRECS
   46 FORMAT(10X,*NUMBER OF RECORDS = *,I6)
      WRITE(20,46)NRECS
   52 FORMAT(10X,*VARIABLE NUMBER*,5X,*MINIMUM*,13X,*MAXIMUM*)
      WRITE(20,52)
      WRITE(6,100)
      READ(5,200) NV
   47 FORMAT(10X,I6,14X,E14.8,6X,E14.8)
      DO 60 I=1,NV
      READ(15)XMIN,XMAX
      WRITE(20,47)I,XMIN,XMAX
   60 CONTINUE
      GO TO 75
   70 CONTINUE
  100 FORMAT(5X,*ENTER 2 DIGIT NUMBER=NO.OF VARIABLES ON EACH RECORD*)
      WRITE(6,100)
  200 FORMAT(I2)
      READ(5,200) NV
   75 CONTINUE
  300 FORMAT(5X,*ENTER TSTOP,F6.3,TO STOP READING PLOT TAPE AT*)
  301 FORMAT(5X,*GIVEN TIME*)
      WRITE(6,300)
      WRITE(6,301)
  400 FORMAT(F6.3)
      READ(5,400) TSTOP
  425 FORMAT(1H1)
      WRITE(20,425)
      N = 0
      L = 4
    5 READ(15)(A(K),K=1,NV)
      IF(EOF(15))80,10
   10 WRITE(20,5000)(A(K),K=1,NV)
 5000 FORMAT(7H TIME =,E15.8,3X6HXXX  =,E15.8,3X6HYYY  =,E15.8,
     *3X6HZZZ  =,E15.8,3X6HVEL R=,E15.8/7H MASS =,E15.8,3X6HXDOT =,
     *E15.8,3X6HYDOT =,E15.8,3X6HZDOT =,E15.8,3X6HMACH =,E15.8/
     *7H THRST=,E15.8,3X6HPPP  =,E15.8,3X6HQQQ  =,E15.8,3X6HRRR  =,
     *E15.8,3X6HDYNPR=,E15.8/7H DIST =,E15.8,3X6HPDOT =,E15.8,
     *3X6HQDOT =,E15.8,3X6HRDOT =,E15.8,3X6HALFI =,E15.8/7H CG   =,
     *E15.8,3X6HUUU  =,E15.8,3X6HVVV  =,E15.8,3X6HWWW  =,E15.8,
```

```
     *3X6HRHO  =,E15.8/7H CP    =,E15.8,3X6HUDUT =,E15.8,3X6HVDOT =,
     *E15.8,3X6HWDOT =,E15.8,3X6HALFTO=,E15.8/7H PMOMA=,E15.8,
     *3X6HALF D=,E15.8,3X6HBET D=,E15.8,3X6HETA D=,E15.8,3X6HFORSU=,
     *E15.8/7H QMOMA=,E15.8,3X6HTHETA=,E15.8,3X6HPHI D=,E15.8,
     *3X6HPSI D=,E15.8,3X6HFORSV=,E15.8/7H RMUMA=,E15.8,3X6HRELVX=,
     *E15.8,3X6HRELVY=,E15.8,3X6HRELVZ=,E15.8,3X6HFORSW=,E15.8/
     *7H PMOMG=,E15.8,3X6HWINDX=,E15.8,3X6HWINDY=,E15.8,3X6HWINDZ=,
     *E15.8,3X6HMDOT =,E15.8/7H QMOMG=,E15.8,3X6HRMOMG=,E15.8,3X
     *6HCA   =,E15.8,3X6HCM   =,E15.8,3X6HCN   =,E15.8/7H CNALP=,E15.8,
     *3X6HBETI =,E15.8,3X6HBANK =,E15.8,3X6HCL   =,E15.8,3X6HCMQ  =,
     *E15.8)
  450 FORMAT(1H )
      WRITE(20,450)

      N = N + 1
      IF( N .LT. L) GO TO 15
      WRITE(20,425)
      L = L + 4
   15 IF(A(1) .GE. TSTOP) GO TO 90
      GO TO 5
   80 WRITE(20, 600)
  600 FORMAT(/10X,*END OF FILE HAS BEEN READ*)
      STOP
   90 WRITE(20,700) TSTOP
  700 FORMAT(/10X,*SPECIFIED TIME TSTOP=*,F7.3,* HAS BEEN REACHED*)
      STOP
      END
```

APPENDIX F

TEKPLOT Listing

```
      PROGRAM TEKPLOT(TAPE2,TAPE5=65,TAPE6=65,TAPE10,OUTPUT)
C*********************************************************************
      INTEGER YES,NO
      DIMENSION A(60), XMIN(60), XMAX(60), INDV(60), IDEPV(60)
      DIMENSION TITLE(8)
      DATA YES,NO/3HYES,3H NO/
      CALL INITT(120)
      CALL BINITT
      CALL ANMODE
      CALL ERASE
      CALL HOME
      CALL BELL
      NT=2
    1 REWIND NT
      CALL ERASE
      CALL HOME
      CALL BELL
C******THE FOLLOWING STATEMENT SETS THE RUN NUMBER TO ONE**********
      IRUN=001
C
C DISPLAY A MESSAGE ON THE TEKTRONIX SCREEN ASKING THE OPERATOR TO ENTER
C THE NUMBER OF VARIABLES ON THE DATA RUN. IF MORE THAN TWENTY, CHANGE
C THE DIMENSIONS FOR A, XMIN, XMAX AND ADD MORE LABELS TO SUBROUTINE
C LABLR
      CALL ANMODE
      WRITE(6,120)
  120 FORMAT(10X,*ENTER 2 DIGIT NO.=TO NO OF VARIABLES ON DATA RUN*)
C READ THE OPERATORS ANS FROM THE TERMINAL
      READ(5,130)NV
  130 FORMAT(I2)
      CALL ERASE
      CALL HOME
      CALL BELL
  140 FORMAT(2E14.8)
C DISPLAY A MESSAGE ASKING THE OPERATOR IF THE DATA TAPE IS IN THE
C STANDARD FORMAT.
      CALL ANMODE
      WRITE(6,131)
  131 FORMAT(10(/))
      WRITE(6,132)
  132 FORMAT(5X,*THE STANDARD FORMAT FOR THE DATA TAPE IS FOR THE *)
      WRITE(6,133)
  133 FORMAT(5X,*NUMBER OF DATA POINTS IN THE RUN TO APPEAR SOMEWHERE*)
      WRITE(6,134)
  134 FORMAT(5X,*IN THE FIRST RECORD. THE NEXT NV RECORDS,WHERE NV=THE*)
      WRITE(6,135)
  135 FORMAT(5X,*THE NUMBER OF VARIABLES,EACH CONTAIN THE MIN AND MAX.*)
      WRITE(6,136)
  136 FORMAT(5X,*VALUES OF THE ITH VARIABLE,WHERE I=1,NV*)
      WRITE(6,137)
  137 FORMAT(5X,*IF THE DATA IS IN THE STANDARD FORMAT,ENTER YES*)
      WRITE(6,138)
  138 FORMAT(5X,*IF NOT,ENTER NO.*)
      WRITE(6,139)
  139 FORMAT(5X,*IF NO,THE PROGRAM WILL CONVERT AND PROCEED NORMALLY*)
```

```
145 FORMAT(A3)
    READ(5,145)IANS1
    IF(IANS1.EQ.YES)GO TO 5
    CALL FORMAT(NV,NT,NPTS)
DISPLAY A MESSAGE ASKING THE OPERATOR TO ENTER A 2 DIGIT NO.=TO THE NO
OF PLOTS TO BE MADE
  5 CALL HOME
    CALL ERASE
    CALL BELL
    CALL ANMODE
    WRITE(6,131)
    WRITE(6,150)
150 FORMAT(10X,*ENTER A 2 DIGIT NO.=TO NO. OF PLOTS TO BE MADE*)
READ THE OPERATORS ANSWER
    READ(5,160)NPLTS
160 FORMAT(I2)
    CALL ERASE
    CALL HOME
    CALL BELL
WRITE A MESSAGE ASKING THE USER TO CHOOSE THE VARIABLES TO BE PLOTTED
BY ENTERING PAIRS OF 2 DIGIT NUMBERS CORRESPONDING TO THE INDEPENDENT
AND DEPENDENT VARIABLE PAIRS
    CALL ANMODE
    WRITE(6,131)
    WRITE(6,161)
161 FORMAT(10X,*CHOOSE THE PAIRS OF VARIABLES TO BE PLOTTED BY *)
    WRITE(6,162)
162 FORMAT(10X,*ENTERING PAIRS OF 2 DIGIT NUMBERS.THE FIRST NUMBER*)
    WRITE(6,163)
163 FORMAT(10X,* OF THE PAIR DESIGNATES THE IND. VARIABLE*)
    WRITE(6,164)
164 FORMAT(10X,*THE FORMAT FOR THE PAIRS IS I2,1X,I2*)
    WRITE(6,165)
165 FORMAT(10X,*THE NUMBER OF PAIRS OF NUMBERS ENTERED MUST EQUAL    *)
    WRITE(6,166)
166 FORMAT(10X,*THE NUMBER OF PLOTS TO BE MADE OR YOU WILL BE STUCK *)
    WRITE(6,167)
167 FORMAT(10X,*IN A READ LOOP*)
READ THE OPERATORS ANSWER FROM THE SCREEN
    DO 20 I=1,NPLTS
 20 READ(5,170)INDV(I),IDEPV(I)
170 FORMAT(I2,1X,I2)
PROMPT OPERATOR TO ENTER TITLE FOR PLOTS
    CALL HOME
    CALL ERASE
    CALL BELL
    CALL ANMODE
171 FORMAT(1X,*ENTER TITLE FOR PLOTS-UP TO 80 CHARACTERS*)
    WRITE(6,171)
    READ(5,146)TITLE
146 FORMAT(8A10)
BEGIN PLOTTING
180 CONTINUE
    DO 500 I=1,NPLTS
```

```
      IF(IANS.EQ.YES)GO TO 5
      CALL BELL
      CALL HOME
      CALL ANMODE
      WRITE(6,131)
      WRITE(6,250)
  250 FORMAT(20X,*PLOTTING COMPLETE-*)
      CALL FINITT(0,400)
      STOP
      END
      SUBROUTINE XYGRID(XMIN,XMAX,YMIN,YMAX)
      DIMENSION X(3),Y(3)
      X(1)=2.
      Y(1)=2.
      X(2)=0.
      Y(2)=0.
      X(3)=0.
      Y(3)=0.
      CALL BINITT
      CALL DLIMX(XMIN,XMAX)
      CALL DLIMY(YMIN,YMAX)
      CALL XFRM(5)
      CALL YFRM(5)
      CALL XNEAT(1)
      CALL YNEAT(1)
      CALL ERASE
      CALL HOME
      CALL CHECK(X,Y)
      CALL DSPLAY(X,Y)
      RETURN
      END
      SUBROUTINE LABLR(IX,IY,TITLE)
      DIMENSION TITLE(8)
      COMMON/LABL/LAB(60,3)
      INTEGER HL(3), VL(3), IHL(30), IVL(30)
      DO 1 K=1,3
      VL(K)=LAB(IY,K)
    1 HL(K)=LAB(IX,K)
      CALL ANMODE
      CALL D2A(HL,30,IHL)
      CALL D2A(VL,30,IVL)
      CALL MOVABS(200,25)
      CALL HLABEL(30,IHL)
      CALL MOVABS(0,700)
      CALL VLABEL(30,IVL)
      CALL HOME
      CALL ANMODE
      WRITE(6,2)TITLE
    2 FORMAT(1X,8A10)
      RETURN
      END
      SUBROUTINE FORMAT(NV,NT,NPTS)
      DIMENSION A(60),B(60),C(60)
      NPTS=0
```

```
     REWIND NT
     READ(NT)NPTS
     DO 190 KK=1,NV
190  READ(NT)XMIN(KK),XMAX(KK)
SELECT THE SUBSCRIPTS OF THE INDEPENDENT AND DEPENDENT VARIABLE
     K1=INDV(I)
     K2=IDEPV(I)
CALL SUBROUTINE XYGRID.THIS DRAWS THE GRID AND LEBELS IT WITH THE MIN.
VALUES OF THE INDEPENDENT,DEPENDENT VARIABLE
     CALL ERASE
     CALL HOME
     THE FIRST VARIABLE ON A RECORD,A(1),IS USUALLY TIME. IF YOU WANT
THE LOWEST VALUE ON THE HORIZONTAL AXIS TO BE ZERO,REGARDLESS OF THE
SMALLEST VALUE OF TIME,ACTIVATE THE FOLLOWING STATEMENTS
                    IF(K1.NE.1) GO TO 200
                    XMI=0.0
                    XMA=XMAX(K1)-XMIN(K1)
                    CALL XYGRID(XMI,XMA,XMIN(K2),XMAX(K2))
                    GO TO 201
200  CONTINUE
     CALL XYGRID(XMIN(K1),XMAX(K1),XMIN(K2),XMAX(K2))
201  CONTINUE
MOVE THE CURSOR TO THE STARTING POINT OF THE GRAPH,A(K1),A(K2)
DRAW THE SPECIFIED GRAPH
     DO 220 IA=1,NPTS
     READ(NT)(A(J),J=1,NV)
     IF(EOF(NT))221,222
222  CONTINUE
     IF THE FIRST VARIABLE,A(1),USUALLY TIME,DOES NOT START AT ZERO
YOU CAN INSERT A STATEMENT THAT SUBRACTS OFF THE MINIMUM VALUE OF TIME
AND THE ZERO TIME ON THE PLOTS WILL REPRESENT THE STARTING TEST TIME
     SUCH A STATEMENT WOULD BE  IF(K1.EQ.1)A(K1)=A(K1)-XMIN(1)
     IF(IA.EQ.1)CALL MOVEA(A(K1),A(K2))
     IF(IA.EQ.1)GO TO 220
     CALL DRAWA(A(K1),A(K2))
220  CONTINUE
221  CONTINUE
LABEL THE X AND Y AXIS
     CALL LABLR(K1,K2,TITLE)
     CALL HDCOPY
     CALL BELL
500  CONTINUE
DISPLAY A MESSAGE ON THE TEKTRONIX SCREEN ASKING THE OPERATOR IF HE
WANTS TO CONTINUE PLOTTING FROM THE CURRENT RUN
     CALL HOME
     CALL ERASE
     CALL ANMODE
     WRITE(6,230)
230  FORMAT(10X,*DO YOU WISH TO CONTINUE PLOTTING FROM THE CURRENT RUN*
    1)
     WRITE(6,240)
240  FORMAT(10X,*ENTER YES OR NO*)
READ THE OPERATORS ANSWER FROM SCREEN
     READ(5,145)IANS
```

```
      REWIND 2
      DO 5 I=1,NV
      B(I)=10000000.
    5 C(I)=-10000000.
    6 READ(2)(A(J),J=1,NV)
      IF(EOF(2))20,10
   10 NPTS=NPTS+1
C FIND THE MIN.,B(I),AND THE MAX.,C(I),OF EACH VARIABLE
      DO 15 I=1,NV
      B(I)=AMIN1(A(I),B(I))
   15 C(I)=AMAX1(A(I),C(I))
      GO TO 6
   20 REWIND 2
      NT=10
      WRITE(NT)NPTS
      DO 30 I=1,NV
   30 WRITE(NT)B(I),C(I)
      DO 40 I=1,NPTS
      READ(2)(A(J),J=1,NV)
   40 WRITE(NT)(A(J),J=1,NV)
      ENDFILE NT
      REWIND NT
      RETURN
      END
      BLOCK DATA
      COMMON /LABL/LAB(60,3)
      DATA ((LAB(I,J),J=1,3),I=1,19)
     A/10HTIME OF FL,10HIGHT IN SE,10HCS.       ,
     B10H RANGE     ,10H          ,10H          ,
     C10HDEFLECTION,10H-POSTIVE R,10HIGHT      ,
     D10HALTITUDE  ,10H          ,10H          ,
     E10HRELATIVE V,10HELOCITY   ,10H          ,
     F10HMASS IN SL,10HUGS      ,10H          ,
     G10HRANGE VELO,10HCITY     ,10H          ,
     H10HCROSS RANG,10HE VELOCITY,10H POS-RIGHT,
     I10HVERTICAL V,10HELOCITY-PO,10HSTIVE DOWN,
     J10HMACH NUMBE,10HR         ,10H          ,
     K10HTHRUST    ,10H          ,10H          ,
     L10HROLL RATE ,10HIN REV/SEC,10H          ,
     M10HPITCH RATE,10H IN DEGS/S,10HEC        ,
     N10HYAW RATE I,10HN DEGS/SEC,10H          ,
     O10HDYNAMIC PR,10HESSURE    ,10H          ,
     P10HRANGE TO G,10HO TO TARGE,10HT         ,
     Q10HROLL ACCEL,10H. IN RAD/S,10HEC/SEC    ,
     R10HPITCH ACCE,10HL. IN RAD/,10HSEC/SEC   ,
     S10HYAW ACCEL.,10H IN RAD/SE,10HC/SEC     /
      DATA ((LAB(I,J),J=1,3),I=20,38)
     A/10HALFI      ,10H          ,10H          ,
     B10HCENTER OF ,10HGRAVITY-FT,10H. W/R TAIL,
     C10HMISSILE AX,10HIS VELOCIT,10HY-U       ,
     D10HMISSILE AX,10HIS VELOCIT,10HY-V       ,
     E10HMISSILE AX,10HIS VELOCIT,10HY-W       ,
     F10HDENSITY   ,10H          ,10H          ,
     G10HCENT. OF P,10HRESSUR-FT.,10H W/R TAIL ,
```

```
     H10HBODY AXIS ,10HACCELERATI,10HON-UDOT    ,
     I10HBODY AXIS ,10HACCELERATI,10HON-CDOT    ,
     J10HBODY AXIS ,10HACCELERATI,10HON-WDOT    ,
     K10HALFTU     ,10H          ,10H           ,
     L10HPMOMA     ,10H          ,10H           ,
     M10HALF D     ,10H          ,10H           ,
     N10HBET D     ,10H          ,10H           ,
     O10HTHETA D   ,10H          ,10H           ,
     P10HFORSU     ,10H          ,10H           ,
     Q10HQMONA     ,10H          ,10H           ,
     R10HTHETA     ,10H          ,10H           ,
     S10HPHI D     ,10H          ,10H           /
      DATA ((LAB(I,J),J=1,3),I=39,57)
     1/10HPSI D     ,10H          ,10H           ,
     A10HFORSV     ,10H          ,10H           ,
     B10HRMONA     ,10H          ,10H           ,
     C10HRELVX     ,10H          ,10H           ,
     D10HRELVY     ,10H          ,10H           ,
     E10HRELVZ     ,10H          ,10H           ,
     F10HFORSW     ,10H          ,10H           ,
     G10HPMOMG     ,10H          ,10H           ,
     H10HWINDX     ,10H          ,10H           ,
     I10HWINDY     ,10H          ,10H           ,
     J10HWINDZ     ,10H          ,10H           ,
     K10HMDOT      ,10H          ,10H           ,
     L10HQMOMG     ,10H          ,10H           ,
     M10HRMOMG     ,10H          ,10H           ,
     N10HCA        ,10H          ,10H           ,
     O10HCM        ,10H          ,10H           ,
     P10HCN        ,10H          ,10H           ,
     Q10HCNALP     ,10H          ,10H           ,
     R10HBETI      ,10H          ,10H           /
      DATA ((LAB(I,J),J=1,3),I=58,60)
     1/10HBANK      ,10H          ,10H           ,
     A10HCL        ,10H          ,10HH          ,
     B10HCMO       ,10H          ,10H           /
      END
```

## DISTIRBUTION LIST

| | |
|---|---|
| UAH | |
| Dr. Noim Kheir | 20 |
| The University of Alabama in Huntsville | |
| School of Engineering | |
| Huntsville, AL 35899 | |
| | |
| US Army Material Systems Analysis Activity | |
| ATTN: DRXSY-MP | 1 |
| Aberdeen Proving Ground, MD | |
| | |
| DRSMI-RDF | 1 |
| DRSMI-RPR | 15 |
| DRSMI-LP, Mr. Voigt | 1 |
| DRSMI-R, Dr. McCorkle | 1 |
| DRSMI-RPT | 1 |



☆U.S. GOVERNMENT PRINTING OFFICE: 1984—746-115/3075

END

FILMED

4-84

DTIC