(NASA-TM-82511) DESCRIPTION OF A
MATHEMATICAL MODEL AND COMPUTER SIMULATION
OF SEPARATION OF THE NOSE CAP FROM THE SOLID
ROCKET BOOSTER (NASA) 33 p HC A03/MF A01
CSCL 22B G3/18

N83-18837

Unclas
02853

NASA TECHNICAL MEMORANDUM

NASA TM-82511

DESCRIPTION OF A MATHEMATICAL MODEL AND COMPUTER SIMULATION OF SEPARATION OF THE NOSE CAP FROM THE SOLID ROCKET BOOSTER

By Arthur J. Schwaniger, Jr. and Hughlen I. Murphree
Systems Dynamics Laboratory

December 1982

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA TM-82511		
4. TITLE AND SUBTITLE Description of a Mathematical Model and Computer Simulation of Separation of the Nose Cap from the Solid Rocket Booster		5. REPORT DATE December 1982 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Arthur J. Schwaniger, Jr. and Hughlen I. Murphree		8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		10. WORK UNIT NO. 11. CONTRACT OR GRANT NO.
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546		13. TYPE OF REPORT & PERIOD COVERED Technical Memorandum 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by Systems Dynamics Laboratory, Science and Engineering Directorate.

16. ABSTRACT

A system of equations which models the motion of the Solid Rocket Booster Nose Cap upon separation is described. The computer program which utilizes these equations to generate nose cap trajectories is described in detail. Application of the program to simulate a rocket sled test of the nose cap separation is discussed and the results of the applications are presented. With the information given a user should be able to exercise the computer program with a minimum of effort.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
	Unclassified – Unlimited

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	32	NTIS

MSFC - Form 3292 (May 1969)

For sale by National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

LIST OF ILLUSTRATIONS

LIST OF TABLES

TECHNICAL MEMORANDUM

DESCRIPTION OF A MATHEMATICAL MODEL AND COMPUTER SIMULATION OF SEPARATION OF THE NOSE CAP FROM THE SOLID ROCKET BOOSTER

I. INTRODUCTION

The design of the decelerator subsystem for the Space Shuttle Solid Rocket Boosters includes a nose cap mounted at the forward end of the booster over the Drogue and Pilot parachute packs. The decelerator subsystem sequence of events begins with the ejection of the nose cap. In designing the system, it was necessary to determine what velocity increment had to be imparted to the nose cap so that it would separate without contacting the chute packs. Several different approaches were taken. A series of wind tunnel tests was conducted and a full scale test, using a rocket sled to propel a nose cap to high speed conditions, were run. In order to apply an analytic approach, a mathematical model was formulated. The equations of motion of the nose cap were derived by Steve Winder and Hughlen Murphree, both of the Systems Dynamics Laboratory, in 1974. The original computer program to implement the generation of nose cap trajectories was done by Hughlen Murphree. This report documents the model and computer program giving the system of equations, coordinate systems, and details of input and output data required to simulate a nose cap trajectory.

The aerodynamic characteristics of the nose cap depend on whether or not it is in close proximity to the booster. When the cap is close to the booster the aerodynamic characteristics are functions of both angle of attack and displacement from the booster. When the cap moves without a large afterbody, as it did in the rocket sled tests, the aerodynamic characteristics are functions of only angle of attack. (Mach number is essentially constant during the separation movement.) The program provides an option for simulation of either case.

Because of design features of the SRB and the nose cap, the internal pressure in the cap could be very low at the time of separation. Thus, the external pressure would act to retard the separation. Provisions are made to include this effect by incorporating the internal pressure during separation as a function of separation velocity and displacement.

The model incorporates two dimensions of translation and one of rotation. The British system of units is used throughout.

II. THE MATHEMATICAL MODEL

A. Coordinate Systems

The basic coordinate system used is an inertial system with origin fixed at the center of gravity of the nose cap at time zero. The X_{II} axis is horizontal and the Z_{II} axis is vertical with positive direction downward. Flight path angle, γ, is measured positive upward from the X_{II} axis to the velocity vector. Attitude angle, θ, is measured positive upward from the X_{II} axis to the long axis of the nose cap, and angle of attack, α, is measured from the velocity vector to the long axis of the nose cap. The velocity components of the cap in this system, $\dot{X}_{1I}$ and $\dot{Z}_{1I}$ are called U_{1I} and W_{1I}, respectively. The velocity components of the booster are also given in this system and are called U_{1B} and W_{1B}.

A body fixed coordinate system, with the origin fixed at the center of gravity of the nose cap, the X_1 axis fixed on the long axis of the cap with positive direction toward the nose, and Z_1 axis perpendicular to X_1, is used as the system in which the cap motion is described. The velocity in this system is denoted by U_1 and W_1. These systems are illustrated in Figure 1.

Finally, a coordinate system whose origin moves with the booster and whose axes are fixed parallel to X_{1I} and Z_{1I} is used to record the motion of points on the nose cap relative to the booster. These coordinates are denoted by X and Z with subscripts to identify the point being tracked. The velocity components of the cap in this system are called U and W.

B. Equations of Motion and Integration

A flat earth with constant gravity magnitude and direction is assumed. The booster motion during the time of interest is assumed to be of constant velocity magnitude and direction and the atmospheric density is assumed constant. These assumptions are warranted by the fact that the nose cap separates from the booster and clears the drogue pack in less than 0.1 sec.

The velocity and attitude of the mated booster and nose cap are given as U_{1I}, W_{1I}, and θ. The ΔV of nose cap separation is given as DV. The first step is to express the velocity of the cap in the body fixed system. This is done by the equations:

$$U_1 = U_{1I} \cos\theta - W_{1I} \sin\theta + DV$$

$$W_1 = U_{1I} \sin\theta + W_{1I} \cos\theta \quad .$$

Next using X_m to represent nose cap mass and X_I for moment of inertia, the time derivatives of U_1, W_1, and θ are given as:

$$\dot{U}_1 = -W_1 \dot{\theta} - G \sin\theta + \frac{F_A + F_P}{X_m}$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$\dot{W}_1 = U_1 \dot{\theta} + G \cos\theta + \frac{F_N}{X_m}$$

$$\ddot{\theta} = \frac{-F_N(CG - CP) + X_{m1}}{X_I}$$

where

G = acceleration due to gravity

ORIGINAL PAGE IS
OF POOR QUALITY

F_A = aerodynamic axial force

F_N = aerodynamic normal force

F_P = force due to difference between ambient pressure and nose cap internal pressure.

CG = cap center of gravity location

CP = cap center of aerodynamic pressure location

X_{m1} = pitch moment coefficient times reference diameter times $\rho V^2 A/2$.

These equations are numerically integrated by the Runge-Kutta method yielding U_1, W_1, and $\dot{\theta}$. U_1 and W_1 are transformed to U and W as follows:

$$U_{1I} = U_1 \cos\theta + W_1 \sin\theta$$

$$W_{1I} = -U_1 \sin\theta + W_1 \cos\theta$$

$$U = U_{1I} - U_{1B}$$

$$W = W_{1I} - W_{1B} \quad .$$

Finally, U, W, and $\dot{\theta}$ are integrated to produce X, Z, and θ.

C. Aerodynamics

The aerodynamics of the nose cap are given in Reference 1 as functions of angle of attack, α, with separation distance from the booster as a parameter. If motion of the cap from the vicinity of the booster is to be simulated, the aerodynamic coefficients are determined by linear interpolation among points in tables with both α and separation distance as independent variables. If motion of the cap without an afterbody present, as with the rocket sled tests, is to be simulated, the coefficients are determined as functions of α alone using the data for separation distance equal to one diameter or greater.

The differences between internal and external pressure are given as functions of separation distance for various cap separation velocities in Reference 2. This pressure difference applied over the base area of the cap is introduced in the equations of motion of the cap as a retarding force along the X_1 axis.

D. Data Form

Most applications of the nose cap trajectory model are done in terms of the angle of attack, dynamic pressure, altitude, and flight path angle of the mated booster and nose cap at the instant of separation. These parameters do not appear explicitly in the model and were not used in the original computer program. The latest version of the program, however, has these parameters for input and incorporates the simple transformations that were originally done by hand each time a simulation was set up. One exception to this is altitude. The atmospheric density, ρ, which appears in the aerodynamic force and moment, and the gravitational acceleration, G, are functions of the altitude at which separation takes place. These must be calculated or determined from tabular data beforehand and must be input to the program.

III. USING THE COMPUTER PROGRAM

A. General Information

The source program is written in Fortran IV and is stored and functions on the Honeywell Sigma V computer. It is called NOSECAP. The data file is called DCAP. DCAP and NOSECAP are listed in Tables 1 and 2.

B. Input

The computer program is set up specifically for the aerodynamic data as it is given in Reference 3. This reference gives axial force coefficient, normal force coefficient, and pitching moment coefficient as functions of angle of attack, α, with the parameter, separation distance divided by length, $\Delta X/D$. Thirteen values of α are chosen and listed as AL(I). Axial force coefficients are specified as CAT1(I) through CAT4(I) where the 1 signifies the value at separation distance, $\Delta X/D$, equals 1 and 4 signifies separation distance equals 0. XCMT1(I) through XCMT4(I) similarly represents pitch moment coefficient for separation distance 1 through 0. CNT1(I) through CNT6(I) represents the normal force coefficient since the reference gives six curves for normal force coefficient instead of four. If the number of entries in these tables is changed, the dimension statements must be changed accordingly.

If the data are to be used to represent separation from the booster, the statement in line 146 of the program is used as part of the comment started in line 145, that is the letter C remains in the first column. If the sled test is to be simulated, the letter C is removed from the beginning of line 146 and the program proceeds with the quantity XCHK = 1.

The above data are part of the program and can only be changed by modification of the program. The following data are located in the data file and are read by the program.

The difference between internal and ambient pressure is given in Reference 2 as a function of separation distance with nose cap velocity as a parameter. If the pressure difference is to be taken into account, the appropriate curve should be chosen and the data loaded as separation distance equals A01 and pressure differential equals DPR. This table is dimensioned for twelve points. If more or fewer points are to be loaded, the dimension statement should be changed accordingly. If pressure difference is to be neglected, DPR should be loaded as zero.

The next data loaded are G, the gravitational acceleration in ft/sec^2; RHO, the atmospheric density in slugs/ft^3; and XLENG, the length of the nose cap in inches.

The next line of data is CPP, the location of aerodynamic center of pressure in percent of body length, and REFD, the reference diameter in inches.

The next line of data is XI, the moment of inertia in slug ft^2; ALPH, the angle of attack of the booster and cap before separation in degrees; DYNPR, the dynamic pressure before separation in lb/ft^2; GAMMA, the flight path angle before separation in degrees; THID, the initial rate of change of attitude in degrees/sec; and DV, the separation velocity increment in ft/sec.

The next line of data is WT, the weight of the cap in lb; CGP, the location of the center of gravity of the cap in percent of body length; XCG, the X coordinate of the center of gravity of the cap, in body fixed coordinates (usually zero); ZCG, the Z coordinate of center of gravity of the cap in body fixed coordinates (usually zero); and AREA, the reference area of the cap in ft^2.

The last line of input is XCONE, the distance from the cap CG to the base of the cap in feet; and ZCONE, the distance in feet from the center line of the nose cap to a point on the base of the cap. XCONE and ZCONE establish the points P and Q which are tracked in the trajectory printout (Fig. 1). This completes the data file.

Besides printing the trajectory of nose cap CG, and points P and Q as functions of time, the computer program has a subroutine called DIST which tracks an extreme point on the nose cap (usually the corner of the thruster post) and determines the distance up the side of the drogue pack where this point intersects the drogue pack envelope (Fig. 2). This distance is printed between time steps where the crossing of the pack envelope occurs. This subroutine contains constants that could be changed if a different cap configuration is studied. These constants are illustrated in Figure 2. Note that the angle PP is used in the subroutine in radians.

The integration time step may be adjusted if a different cap configuration is studied (for instance if wind tunnel tests are simulated). The time step DELTA is specified on line 130 of the main routine. Also, the value of DELTA is used in line 289. These values should always be equal. The value, 2^{-8}, has been found appropriate for simulating motion of the full scale SRB nose cap. For simulation of wind tunnel tests where the scale factor is 1/8, for example, the integration time step should be scaled by a factor $\sqrt{1/8}$.

The time point at which to stop the computation is given at line 87 as TSTOP. At present, the value TSTOP = 0.16 is used.

C. Output

The input data file is printed out in the first block of output. The printout then proceeds with trajectory parameters at regular time intervals. A list of the mnemonics and the parameters they represent is given in Table 3.

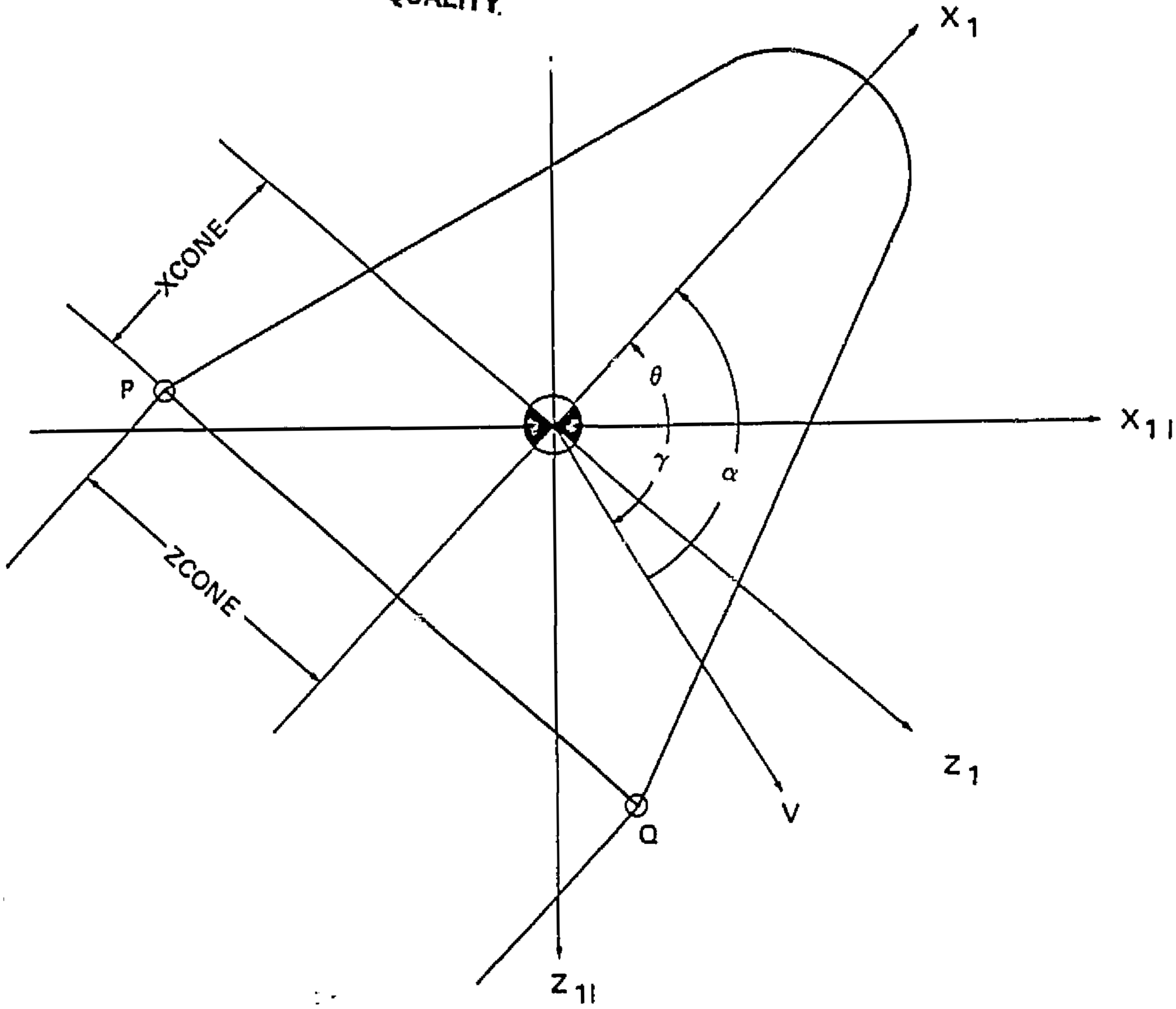
IV. CONCLUSION

Completion of the sled test presented an opportunity to compare the nose cap separation simulation with test results. Motion picture film of the test was analyzed and a graphic representation of the nose cap motion was drawn. A simulation was then made using the test conditions and a similar graph was overlaid on the test results. This is shown in Figure 3. The coordinates used are those of the film analysis. The abscissa is along the base of the nose cap before separation. The origin is at the leading edge of the nose cap before separation. Positions of the nose cap are shown for both test results (solid line) and simulation (broken line) at various times. The times do not coincide exactly, in general, because the simulation time step was 2^{-8} sec and the film speed was 0.0125 sec per frame. Nevertheless, the simulated motion is very similar to the observed motion.

The test condition was the design limit condition of 80 deg angle of attack and dynamic pressure 200 lb/ft^2. The data file shown in Table 1 is the one used for this simulation and all of the test conditions are included there. This particular simulation serves as a sample of the program function. It is identified as run 0464 and is kept in the files of the Dynamics and Trajectory Analysis Branch.

The good agreement between the test and simulated data has established the validity of use of the simulation to analyze the separation motion.

ORIGINAL PAGE IS
OF POOR QUALITY



ANGLES MEASURED COUNTERCLOCKWISE ARE POSITIVE
FLIGHT PATH ANGLE, γ, IS NEGATIVE AS SHOWN

Figure 1. Coordinate systems and preload parameters.

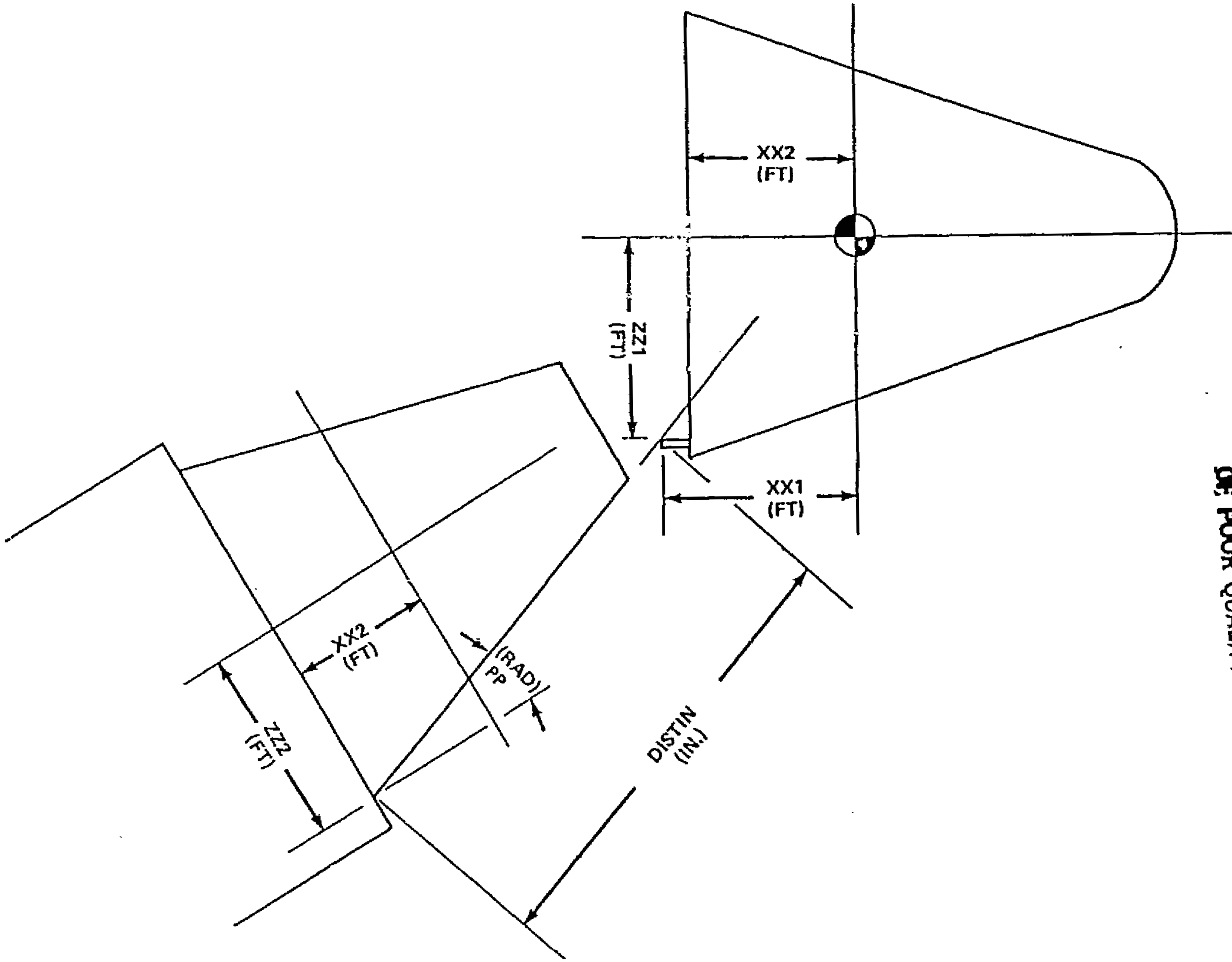


Figure 2. Parameters used in DIST subroutine.

ORIGINAL PAGE IS
OF POOR QUALITY

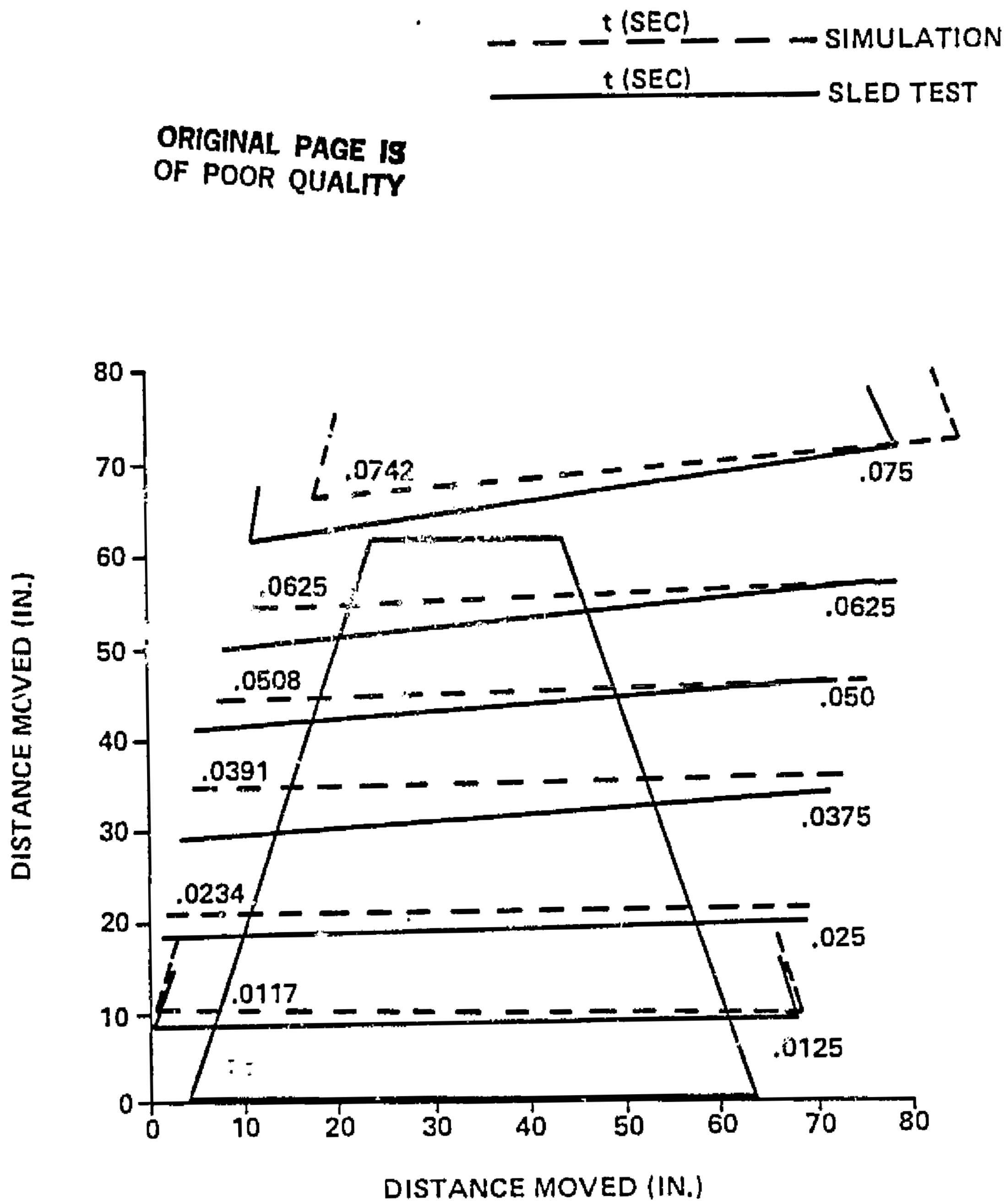


Figure 3. Comparison of motion of base of nose cap as simulated and observed in sled test.

TABLE 1. LISTING OF DCAP

```
0.0,5.0,10.0,15.0,20.0,25.0,
30.0,35.0,40.0,45.0,50.0,300.0,
0.,0.,0.,0.,0.,0.,
0.,0.,0.,0.,0.,0.,
32.2,.002,75.,
.667,68.03
52.,80.,200.,0.,0.,76.,
300.,.66,0.,0.,25.24,
2.125,2.83333,
*
```

TABLE 2. LISTING OF NOSECAP

```
C      NOSE CAP SEPARATION STUDIES
       REAL M,M1
       DIMENSION CNT1(13),XCMT1(13),CAT1(13),CNT5(13),CNT6(13),A
L(13)
       DIMENSION CNT2(13),XCMT2(13),CAT2(13)
       DIMENSION CNT3(13),XCMT3(13),CAT3(13)
       DIMENSION CNT4(13),XCMT4(13),CAT4(13)
       DIMENSION A01(12),DPR(12)
       COMMON/SECD/TH1DD,DUM0(59)
       COMMON/FUNT/Y(3),DUM1(57)
       COMMON/FIRT/F(3),DUM2(57)
       COMMON/LAB/U1P,TH1P,REFD
       COMMON XCONE,ZCONE
       EQUIVALENCE (TH1D,F(1)),(W1D,F(2)),(U1D,F(3))
       EQUIVALENCE (TH1,Y(1)),(W1,Y(2)),(U1,Y(3))
       DATA (CAT1(I),I=1,13)/.4,.42,.38,.25,.1,.02,.22,-.18,-.8,
-1.3,
      *-1.38,-1.38,-1.4/
       DATA (CAT2(I),I=1,13)/.4,.42,.3,.18,0.,-.06,.16,-.19,-.9,
-1.3,
      *-1.38,-1.38,-1.4/
       DATA (CAT3(I),I=1,13)/.4,.42,.28,.1,-.02,-.16,.04,-.2,-1.
,-1.3,
      *-1.38,-1.38,-1.4/
       DATA (CAT4(I),I=1,13)/.4,.42,.21,.06,-.1,-.18,0.,-.21,-1.
1,-1.3,
      *-1.38,-1.38,-1.4/
       DATA (XCMT1(I),I=1,13)/0.,.02,.03,.02,-.02,-.05,-.1,-.1,0
.,.04,
      *.05,.04,0./
       DATA (XCMT2(I),I=1,13)/0.,.02,.03,0.,-.03,-.07,-.13,-.12,
-.02,
      *.04,.05,.04,0./
       DATA (XCMT3(I),I=1,13)/0.,.01,.02,-.02,-.04,-.09,-.15,-.1
3,
      *-.03,.03,.04,.04,0./
*
```

TABLE 2. (Continued)

```
      DATA (XCMT4(I),I=1,13)/0.,.01,.01,-.04 ,-.08,-.12,-.18,-.
14,-.06,
     *.01,.03,.04,0./
      DATA (AL(I),I=1,13)/0.,15.,30.,45.,60.,75.,90.,105.,120.,
135.,
     *150.,165.,180./
      DATA (CNT1(I),I=1,13)/0.,.38,.8,.82,.8,.7,.55,.44,.33,.12
,
     *-.02,-.1,0./
      DATA (CNT2(I),I=1,13)/0.,.39,.82,.9,.86,.76,.57,.45,.34,.
13,
     *-.02,-.1,0./
      DATA (CNT3(I),I=1,13)/0.,.4,.84,.98,.96,.81,.61,.46,.35,.
14,
     *-.02,-.1,0./
      DATA (CNT4(I),I=1,13)/0.,.41,.9,1.01,1.02,.9,.65,.49,.36,
.15,
     *-.02,-.1,0./
      DATA (CNT5(I),I=1,13)/0.,.42,.97,1.13,1.16,1.05,.72,.51,.
37,.16,
     *-.02,-.1,0./
      DATA (CNT6(I),I=1,13)/0.,.43,1.12,1.35,1.46,1.4,.87,.53,.
39,.17,
     *-.02,-.1,0./
   25 READ (5,92),(A01)
      READ (5,92),(DPR)
      WRITE (6,92),(A01)
      WRITE (6,92),(DPR)
      READ (5,2),(G,RHO,XLENG)
      READ (5,2),(CPP,REFD)
      READ (5,92),(XI,ALPH,DYNPR,GAMMA,TH1D,DV)
      READ (5,2),(WT,CGP,X1,Z1,AREA)
      READ (5,2), (XCONE,ZCONE)
      XLEN=XLENG
      ANGLEA = ATAN(XLENG/(REFD*.5))
      SIDEL = XLENG/  N(ANGLEA)
*
```

```
      SIDEL=SIDEL/12.
      U1P = DV
      TH1P = ALPH+GAMMA
      WRITE(6,92),(G,RHO,XLENG)
      WRITE(6,92),(CPP,REFD)
      WRITE(6,92) ,(XI,ALPH,DYNPR,GAMMA,TH1D,DV)
      WRITE(6,92),(WT,CGP,X1,Z1,AREA)
      WRITE(6,92),(XCONE,ZCONE)
   92 FORMAT(     6E15.6)
      C=57.29578
      ALPH=ALPH/C
      GAMMA=GAMMA/C
      TH1=ALPH+GAMMA
      VVV=SQRT(2.0*DYNPR/RHO)
      U1I=VVV*COS(GAMMA)
      W1I=-VVV*SIN(GAMMA)
      U1B=U1I
      W1B=W1I
      TH1DD=0.
      PSI=ATAN2(ZCONE,XCONE)
      TH1D = TH1D/C
      TH11=TH1
      THETA = TH1
      U1 = U1I*COS(TH1) - W1I*SIN(TH1) + DV
      W1 = U1I*SIN(TH1) + W1I*COS(TH1)
C     THE ABOVE EQUATIONS PUT U1 AND W1 IN THE BODY AXIS SYSTEM
      III=0
      K=0
      L=12
      N=13
      TSTOP=0.16
      XM=WT/G
      THA=PSI-TH1
      R=8.*0.46/12.
```


*

TABLE 2. (Continued)

```
TLEN=(2.56*8.)/12.
FLEN=(8.1*8)/12.
THI=-TH1
XA=X1-XCONE*COS(THI)-ZCONE*SIN(THI)+R*SIN(THI)
ZA=Z1-XCONE*SIN(THI)+ZCONE*COS(THI)-R*COS(THI)
G1=108./57.3-THI
G2=(72./57.3)+TH1-(90./57.3)
M=TAN(G2)
B=ZA-M*XA
XB=X1-XCONE*COS(THI)+ZCONE*SIN(THI)-R*SIN(THI)
ZB=Z1-XCONE*SIN(THI)-ZCONE*COS(THI)+R*COS(THI)
AC1=M*M+1.
BC1=-2.*XA-2.*M*ZA+2.*B*M
CC1=XA*XA+ZA*ZA-2.*B*ZA+B*B-FLEN*FLEN
XA1=(-BC1+SQRT(BC1*BC1-4.*AC1*CC1))/(2.*AC1)
ZA1= (M*XA1+B)
M1=(ZB-ZA)/(XB-XA)
B2=ZA1-M1*XA1
AC2=M1*M1+1.
BC2=-2.*XA1+2.*M1*B2-2.*M1*ZA1
CC2=XA1*XA1+B2*B2-2.*ZA1*B2+ZA1*ZA1-TLEN*TLEN
XB1=(-BC2+SQRT(BC2*BC2-4.*AC2*CC2))/(2.*AC2)
ZB1= (M1*XB1+B2)
M=TAN(G1)
XTP=XA
ZTP=ZA
WRITE(3),XTP,ZTP
XTP=XB
ZTP=ZB
WRITE(3),XTP,ZTP
XTP=XB1
ZTP=ZB1
WRITE(3),XTP,ZTP
XTP=XA1
*
```

TABLE 2. (Continued)

```
      ZTP=ZA1
      WRITE(3),XTP,ZTP
      XTP=XA
      ZTP=ZA
      WRITE(3),XTP,ZTP
      DELT=.00390625
      TIME=0.0
      PRTIM = 0.
      PLTSAV = 0.
      I = 1
    2 FORMAT (5E15.6)
  200 IND=1
  500 CONTINUE
      ALPHA=XTAN2(W1,U1)
      IF(ALPHA)501,502,502
  501 ALPHA=360.0+ALPHA
  502 CONTINUE
      XCHK=X1/(REFD/12.)
C
C
C     THIS IS ONLY USED FOR THE SLED TEST EVALUATIONS
C     XCHK=1.0
C
C
      IF(XCHK.GE.1.) GO TO 900
C     THIS IS FOR XCHK=0.3
  901 CALL TBL(AL,CNT2,ALPHA,N,CN2)
      XCHK1=0.3
      CNC1=CN2
      IF(XCHK.GE.0.3) GO TO 900
C     THIS IS FOR  XCHK=0.2
  902 CALL TBL(AL,CNT3,ALPHA,N,CN3)
      IF(XCHK.GE.0.2) GO TO 802
      XCHK1=0.2
```


*

TABLE 2. (Continued)

```
      CNC1=CN3
      GO TO 903
  802 XCHK2=0.2
      CNC2=CN3
      GO TO 909
C     THIS IS FOR XCHK=0.1
  903 CALL TBL(AL,CNT4,ALPHA,N,CN4)
      IF(XCHK.GE.0.1) GO TO 803
      XCHK1=0.1
      CNC1=CN4
      GO TO 904
  803 XCHK2=0.1
      CNC2=CN4
      GO TO 909
C     THIS IS FOR XCHK=0.05
  904 CALL TBL(AL,CNT5,ALPHA,N,CN5)
      IF(XCHK.GE.0.05) GO TO 804
      XCHK1=0.05
      CNC1=CN5
      GO TO 905
  804 XCHK2=0.05
      CNC2=CN5
      GO TO 909
C     THIS IS FOR XCHK=0.0
  905 CALL TBL(AL,CNT6,ALPHA,N,CN6)
      IF(XCHK.GE.0.0) GO TO 805
      CN=CN6
      GO TO 911
  805 XCHK2=0.0
      CNC2=CN6
      GO TO 909
C     THIS IS FOR XCHK=1.0
  900 CALL TBL(AL,CNT1,ALPHA,N,CN1)
      XCHK2=1.0
*
```

```
      CNC2=CN1
      IF(XCHK.GE.1.0) GO TO 910
  909 CN=((XCHK-XCHK1)/(XCHK1-XCHK2))*(CNC1-CNC2)+CNC1
      GO TO 911
  910 CN=CN1
  911 CONTINUE
      IF(XCHK.GE.1.) GO TO 700
C     THIS IS FOR XCHK=0.2
  701 CALL TBL(AL,XCMT2,ALPHA,N,XCM2)
      CALL TBL(AL,CAT2,ALPHA,N,CA2)
      XCHK1=0.2
      XCM1=XCM2
      CA1=CA2
      IF(XCHK.GE.0.2) GO TO 700
C     THIS IS FOR XCHK=0.05
  702 CALL TBL(AL,XCMT3,ALPHA,N,XCM3)
      CALL TBL(AL,CAT2,ALPHA,N,CA3)
      IF(XCHK.GE.0.05) GO TO 601
      XCHK1=0.05
      XCM1=XCM3
      CA1=CA3
      GO TO 703
  601 XCHK2=0.05
      XCM2=XCM3
      CA2=CA3
      GO TO 709
C     THIS IS FOR XCHK=0.0
  703 CALL TBL(AL,XCMT4,ALPHA,N,XCM4)
      CALL TBL(AL,CAT4,ALPHA,N,CA4)
      IF(XCHK.GE.0.0) GO TO 605
      XCM=XCM4
      CA=CA4
      GO TO 711
  605 XCHK2=0.0
```


*

TABLE 2. (Continued)

```
            XCM2=XCM4
            CA2=CA4
            GO TO 709
C           THIS IS FOR XCHK=1.0
      700   CALL TBL(AL,XCMT1,ALPHA,N,XCM6)
            CALL TBL(AL,CAT1,ALPHA,N,CA6)
            XCHK2=1.0
            XCM2=XCM6
            CA2=CA6
            IF(XCHK.GE.1.) GO TO 710
      709   XCM=((XCHK-XCHK1)/(XCHK1-XCHK2))*(XCM1-XCM2)+XCM1
            CA=((XCHK-XCHK1)/(XCHK1-XCHK2))*(CA1-CA2)+CA1
            GO TO 711
      710   XCM=XCM6
            CA=CA6
      711   CONTINUE
            AA= 0.5*RHO*(U1*U1+W1*W1)*AREA
            XM1=AA*XCM*(REFD/12.0)
            C1P=-(XLEN/12.0)*(CGP-CPP)
            FA =-AA*CA
            FN = -AA*CN
            XMOMEN=TH1DD*XI
            TH1DD=((C1P*FN)+XM1)/XI
            W1D=(1.0/XM)*(XM*U1*TH1D+XM*G*COS(TH1)+FN)
            U1D=(1.0/XM)*(-XM*W1*TH1D-XM*G*SIN(TH1)+FA+FP)
            IF(TIME-0.0)600,600,41
       41   CONTINUE
            CALL RUNGE(IND,TIME,DELT,3,1)
            IND=IND+1
            IF(IND-4)500,500,600
     600    CONTINUE
            U1I = U1*COS(TH1) +W1*SIN(TH1)
            W1I = -U1*SIN(TH1) + W1*COS(TH1)
            FPA = ATAN2(W1I,U1I)*C
*
```

```
      W1I = W1I - W1B
      U1I = U1I - U1B
      VEL=SQRT(U1I*U1I+W1I*W1I)
      WRITE(6,39),TIME
        :
      WRITE(6,3),(ALPHA)
        :
      WRITE(6,1001),(FN,FA)
      WRITE(6,4),(XMOMEN)
        :
      WRITE(6,1002),(U1I,W1I,F(1)),FPA
        :
      WRITE(6,2222),(VEL)
        :
      WRITE(6,8),XCM
        :
 8    FORMAT (1X,4HXCM=,E15.6)
        :
      WRITE(6,7),     AA,XM1,C1P,U1,W1,CN
        :
 7    FORMAT (1X,4H AA=, E15.6,4HXM1=,E15.6,4HC1P=,E15.6,4H U1=E15.6,
        :
     *4H W1=,E15.6,4H CN=,E15.6)
        :
      TH1DS=F(1)
      TH1S=Y(1)
      TH1S1=TH1S*C
      W1S=Y(2)
      U1S=Y(3)
      F(1)=0.0
      F(2)=W1I
      F(3)=U1I
      Y(1)=0.0
      Y(2)=Z1
      Y(3)=X1
```

*

TABLE 2. (Continued)

```
      IND=1
      IF(TIME-0.0)800,800,300
  300 CONTINUE
      TIME=TIME-.00390625
  400 CONTINUE
      CALL RUNGE(IND,TIME,DELT,3,0)
      IND=IND+1
      IF(IND-4)400,400,800
  800 TH1=TH1S
      Z1=Y(2)
      X1=Y(3)
      X11=(X1*COS(TH11)-Z1*SIN(TH11))*12.
      CALL TBL(A01,DPR,X11,L,DPRE)
      FP=-DPRE*AREA*144.
      WRITE(6,1003),(Y(3),Y(2),TH1S1,FP)
      THA=PSI-TH1
      TH3=TH1
      XP = X1 - XCONE*COS(-TH3) + ZCONE*SIN(-TH3)
      ZP = Z1 - XCONE*SIN(-TH3) - ZCONE*COS(-TH3)
      XQ = X1 - XCONE*COS(-TH3) - ZCONE* SIN(-TH3)
      ZQ = Z1 - XCONE*SIN(-TH3) + ZCONE*COS(-TH3)
      IF(TIME-.08)401,402,402
  401 CONTINUE
      XTP=XQ
      ZTP=ZQ
      WRITE(1),XTP,ZTP
      XTP=XP
      ZTP=ZP
      WRITE(1),XTP,ZTP
  402 CONTINUE
      WRITE(6,6),(XQ,ZQ)
       C
      WRITE(6,5),(XP,ZP)
      WRITE(6,1013),W1B,DELT
      F(1)=TH1DS
*
```

```
      Y(1)=TH1S
      Y(2)=W1S
      Y(3)=U1S
      I = I + 1
490   CONTINUE
      IF(TIME .LE. 0.0195313) GO TO 201
      CALL DIST(TH11,TH1S,X1,Z1,K)
  201 CONTINUE
      XS1=X1
      ZS1=Z1
      THS=TH1
      IF(TIME-0.0)41,41,49
   49 CONTINUE
      IF(TIME-TSTOP)24,251,251
   24 CONTINUE
      GO TO 200
251   CONTINUE
      GO TO 25
 1001 FORMAT (1H,2HFN,3X,E15.6,2X,2HFA,3X,E15.6)
 1002 FORMAT (1H,2HU ,3X,E15.6,2X,2HW ,3X,E15.6,2X,3HTHD,3X,E15.6,10X,
     *20HFLIGHT PATH ANGLE = ,E15.6)
 1003 FORMAT (1H,3HXCG,2X,E15.6,2X,3HZCG,2X,E15.6,2X,3HTH1,3X,E15.6,
     *2X,2HFP,3X,E15.6)
    3 FORMAT (1H,5HALPHA,3X,E15.6)
    4 FORMAT (1H,6HMOMENT,3X,E15.6)
    5 FORMAT (1H,2HXP,3X,E15.6,2X,2HZP,3X,E15.6)
    6 FORMAT (1H,2HXQ,3X,E15.6,2X,2HZQ,3X,E15.6)
 2222 FORMAT (1H,3HVEL,3X,E15.6)
   39 FORMAT (1H,//1X,4HTIME,3X,E15.6)
 1013 FORMAT(1X,5HW1B= ,E15.8,10X,5HDELT=,E15.8)
      END
*
```

```
      SUBROUTINE RUNGE (KUTTA,TIME,DT,NVAR,NDVAR)
C  KUTTA IS A CONTROL INTEGER
C  TIME IS TIME OF INTEGRATION
C  DT IS THE TIME INCREMENT
C  NVAR = NUMBER OF VARIABLES TO BE INTEGRATED ONCE
C  NDVAR= NUMBER OF VARIABLES TO BE INTEGRATED TWICE
      DIMENSION C1( 60),C2( 60),C3( 60),C4( 60),CD1( 60),CD2( 6
0),
     1 CD3( 60),CD4( 60),SX( 60),SXD( 60)
      COMMON/FUNT/  X( 60)
      COMMON/FIRT/ XD( 60)
      COMMON/SECD/XDD(30)
      GO TO (1,2,3,4),KUTTA
    1 DO 10 I=1,NVAR
      SX(I) = X(I)
      C1(I) = XD(I)*DT
   10 X(I)  = SX(I)+0.5*C1(I)
      IF(NDVAR.EQ.0)GO TO 50
      DO 100 I=1,NDVAR
      SXD(I) = XD(I)
      CD1(I) = XDD(I)*DT
  100 XD(I) = SXD(I)+0.5*CD1(I)
 50   CONTINUE
      TIME = TIME + 0.5*DT
      RETURN
    2 DO 20 I=1,NVAR
      C2(I) = XD(I)*DT
   20 X(I) = SX(I)+0.5*C2(I)
      IF(NDVAR.EQ.0)GO TO 51
      DO 200 I=1,NDVAR
      CD2(I) = XDD(I)*DT
  200 XD(I) = SXD(I)+0.5*CD2(I)
 51   CONTINUE
      RETURN
    3 DO 30 I=1,NVAR
*
```

```
      C3(I) = XD(I)*DT
   30 X(I) = SX(I)+C3(I)
      IF(NDVAR.EQ.0)GO TO 52
      DO 300 I=1,NDVAR
      CD3(I) = XDD(I)*DT
  300 XD(I) = SXD(I)+CD3(I)
52    CONTINUE
      TIME = TIME+0.5*DT
      RETURN
    4 DO 40 I=1,NVAR
      C4(I) = XD(I)*DT
   40 X(I) = SX(I)+(C1(I)+C4(I)+2.*(C2(I)+C3(I)))/6.0
      IF(NDVAR.EQ.0)GO TO 53
      DO 400 I=1,NDVAR
      CD4(I) = XDD(I)*DT
  400 XD(I) = SXD(I)+(CD1(I)+CD4(I)+2.0*(CD2(I)+CD3(I)))/6.0
53    CONTINUE
      RETURN
      END
*
```

TABLE 2. (Continued)

```
      SUBROUTINE TBL(X,Y,X1,N,Y1)
C     THE DATA MUST BE  IN ASCENDING ORDER
      DIMENSION X(1),Y(1)
      DO 93 I=1,N
      IF(X(I)-X1)93,95,91
   91 DX=X1-X(I-1)
      DX1=X(I)-X(I-1)
      DY=Y(I)-Y(I-1)
      Y1=Y(I-1)+D'*(DX/DX1)
      GO TO 94
   95 Y1=Y(I)
      GO TO 94
   93 CONTINUE
   94 CONTINUE
      RETURN
      END
*
```

TABLE 2. (Continued)

```
      FUNCTION XTAN2(A,B)
      IF(B .EQ. 0.0) GO TO 100
      XTAN2=ATAN2(A,B)*57.3
      RETURN
  100 IF(A)10,16,15
   10 XTAN2=-90.0
      RETURN
   16 XTAN2=0.0
      RETURN
   15 XTAN2=90.0
      RETURN
      END
*
```

TABLE 2. (Concluded)

```
      SUBROUTINE DIST(TH11,TH1S,X1,Z1,K)
      IF(K .EQ. 1) GO TO 330
      K=0
      XX1=-2.552083
      ZZ1=2.588367
      XX2=-2.125
      ZZ2=2.516667
      PP=.3131086
      S0=SIN(TH11)
      C0=COS(TH11)
      XX3=XX2*C0+ZZ2*S0
      ZZ3=ZZ2*C0-XX2*S0
      S=SIN(TH1S)
      C=COS(TH1S)
      XX5=X1+XX1*C+ZZ1*S
      ZZ5=Z1+ZZ1*C-XX1*S
      XX6=XX5-XX3
      ZZ6=ZZ5-ZZ3
      PP0=-TH11-PP
      S1=SIN(PP0)
      C1=COS(PP0)
      XX7=XX6*C1+ZZ6*S1
      ZZ7=ZZ6*C1-XX6*S1
      IF(ZZ7 .LT. 0.) GO TO 340
      XX8=XX7
      ZZ8=ZZ7
  330 RETURN
  340 XXX=(XX7-((ZZ7*(XX8-XX7))/(ZZ8-ZZ7)))*12.
      WRITE(6,360),(XXX)
  360 FORMAT(///,7H DISTIN,3X,E15.6,///)
      K=1
      RETURN
      END
*
```

TABLE 3. OUTPUT PARAMETERS

Mnemonic	Definition and Units
TIME	Time (sec)
ALPHA	Angle of attack (deg)
FN	Normal force (lb)
FA	Axial force (lb)
MOMENT	Product of angular acceleration and moment of inertia (lb/ft)
U	Velocity relative to booster component in X direction (ft/sec)
W	Velocity relative to booster component in Z direction (ft/sec)
THD	Angular rate, $\dot{\theta}$, (deg/sec)
FLIGHT PATH ANGLE	Flight path angle, γ, (deg)
VEL	Velocity of nose cap relative to booster (ft/sec)
XCM	Aerodynamic moment coefficient
AA	Product of reference area times dynamic pressure (lb)
XM1	Aerodynamic moment (lb/ft)
C1P	Difference between location of CG and center of pressure (ft)
U1	Body fixed velocity component in X1 direction (ft/sec)
W1	Body fixed velocity component in Z1 direction (ft/sec)
CN	Normal force coefficient
XCG	X coordinate of CG (ft)
ZCG	Z coordinate of CG (ft)
TH1	Attitude angle (deg)
FP	Force due to pressure difference (lb)
XQ	X coordinate of point Q
ZQ	Z coordinate of point Q
XP	X coordinate of point P
ZP	Z coordinate of point P
W1B	Velocity component of booster in Z11 direction (ft/sec)
DELT	Integration time step (sec)

REFERENCES

1. ED15-81-7, "Revised Requirements for Velocity to Separate the SRB Nose Cap," January 22, 1981.
2. ED31-75-1, "SRB Nose Cap Internal Pressure Variation During Ejection," May 1, 1975.
3. ED31-76-59, "Revised Aerodynamic Characteristics of the SRB Nose Cap," December 10, 1976.
4. ED32-77-9, "Analysis of SRB Nose Cap Ejection Wind Tunnel Test SA38F," January 18, 1977.

APPROVAL

DESCRIPTION OF A MATHEMATICAL MODEL AND COMPUTER SIMULATION OF SEPARATION OF THE NOSE CAP FROM THE SOLID ROCKET BOOSTER

By Arthur J. Schwaniger, Jr. and Hughlen I. Murphree

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

GEORGE F. McDONOUGH
Director, Systems Dynamics Laboratory