AD614699

NOLTR 64-225

AN IBM 7090 SIX-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM

COPY 2 OF 3
HARD COPY $. —
MICROFICHE $. —
DDC USERS ONLY

9 FEBRUARY 1965

NOL

UNITED STATES NAVAL ORDNANCE LABORATORY, WHITE OAK, MARYLAND

NOLTR 64-225

DDC
RECEIVED
MAY 11 1965
DDC-IRA E

ARCHIVE COPY

NOTICE

Requests for additional copies by Agencies of the Department of Defense, their contractors, and other Government agencies should be directed to:

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia

Department of Defense contractors who have established DDC services or have their 'need-to-know' certified by the cognizant military agency of their project or contract should also request copies from DDC.

NOLTR 64-225

Aerodynamics Research Report 237

AN IBM 7090 SIX-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM

Prepared by:
W. E. DeGrafft

ABSTRACT: This report presents a six-degree-of-freedom trajectory program which describes the unpowered flight of a rigid body in the vicinity of a spherical rotating earth. The program is written for an IBM 7090 computer operating under the IBSYS monitor.

The mathematical formulation of the problem and the computer program itself are presented. The program description includes a FORTRAN listing and instructions for its use.

PUBLISHED MAY 1965

U. S. NAVAL ORDNANCE LABORATORY
WHITE OAK, MARYLAND

NOLTR 64-225 9 February 1965

AN IBM 7090 SIX-DEGREE-OF-FREEDOM TRAJECTORY PROGRAM

The purpose of this report is to present a complete documentation of an IBM 7090 six-degree-of-freedom trajectory program in current use at the Naval Ordnance Laboratory (NOL).

This report is based partially on techniques contained in informal reports and internal memoranda from the Naval Ordnance Test Station, China Lake and the Lockheed Missiles and Space Company, Sunnyvale. The author would like to gratefully acknowledge the work done by Mr. Jerry Linnekin and Miss Mary Lou Lyons of the NOL Mathematics Department in the development of this program. Appreciation is also extended to Mrs. Carole Young and Mr. Norman Moore of the Applied Aerodynamics Division for their help in putting this program in its final form.

R. E. ODENING
Captain, USN
Commander

K. R. ENKENHUS
By direction

CONTENTS

	Page
Introduction	1
Symbols	2
Mathematical Formulation of Problem	6
Rigid Body Equations of Motion	6
Aerodynamic Force and Moment Equations	7
Aerodynamic Orientation Angles	9
Functional Calculations	11
Initial Calculations	14
Auxiliary Calculations	18
Description of Machine Program	19
Input	19
Options and Controls	22
Output	24
Concluding Remarks	24
References	25
Appendix A	A-1

ILLUSTRATIONS

Figure	Title
1	Aerodynamic and Body Axes Systems
2	Inertial and Local Axes Systems
3	Velocity Components in Local Axes System
4	Local and Aerodynamic Data Axes Systems
5	Earth and Inertial Axes Systems

TABLES

Table	Title
1	FORTRAN Listing of MAIN Routine
2	FORTRAN Listing of SETUP Subroutine
3	FORTRAN Listing of AERO Subroutine
4	FORTRAN Listing of DERIV Subroutine
5	FORTRAN Listing of TERM Subroutine
6	FORTRAN Listing of OUT Subroutine
7	FORTRAN Listing of FNOL-3 Subroutine
8	KY Array Variables
9	Y Array Variables
10	C Array Variables
11	D Array Variables
12	Output Format

INTRODUCTION

Unguided fin-stabilized vehicles find wide applications in aircraft ordnance and sounding rockets due to their relative simplicity and low cost.

One of the more important problems encountered in unguided vehicle design is that both the nutational and roll frequencies may be equal at some time during the flight. For certain combinations of vehicle and flight parameters the ensuing pitch-roll resonance condition can cause sufficient amplification of the vehicle angle of attack to produce structural failure or significant deviation from the intended flight path. The designer must attempt to ensure that pitch-roll resonance is not encountered. In the event that this is not possible, he must ensure that the crossover of the two frequencies takes place as rapidly as possible.

When the dynamic pressure and spin rate are nearly constant and the aerodynamic forces and moments are linear (i.e. no dependence on roll angle and linear dependence on angle of attack) then analytic approaches to this problem are quite satisfactory (ref. (1)). If these rather stringent conditions are not fulfilled, then a six-degree-of-freedom method of analysis will probably be required for an accurate assessment of the problem. The primary purpose of this report is to present a computer program which can be used for this type of analysis as well as the analysis of spin-stabilized vehicles.

This program is similar to those described in references (2) and (3) because, among other things, the Euler angle formulation of the problem is not used. The reason for this is twofold. First, each angle occurring in the initial construction of the inertial-to-body-axes-transfer matrix can be readily defined in terms of the physical variables describing the motion of the vehicle. Second, the singularity occurring when one of Euler's angles is equal to 90° is eliminated.

There are several special features which distinguish this program from many other six-degree-of-freedom trajectory

programs written in the past.

a. Aerodynamic force and moment coefficients which are arbitrary functions of Mach number, total angle of attack and roll angle are introduced in table form. Correct use of the options afforded in the handling of the tabulated data can result in considerable time savings in preparation of the tables and running time of the program.

b. The integration subroutine provides the user with the option of automatically adjusting the integration time step on the basis of estimates of the truncation error. This feature is quite important when the dynamic pressure or spin rate vary over wide ranges during the vehicle flight.

c. Nonrolling body axes can be used even when aerodynamic asymmetries exist. This is important when the vehicle is rapidly spinning since considerable time savings over using rolling body axes can be realized.

Along with the mathematical description of the program, a program listing along with identification of the key arrays is included. The reasons for doing this are to insure that the user has complete understanding of the program and facilitate any changes which may be desired.

SYMBOLS

A	body reference area
A_A	azimuth of $\bar{V}_A$ in local frame (Figure 3)
A_E	azimuth of $\bar{V}_E$ in local frame (Figure 3)
A_W	azimuth of local wind (Figure 3)
c	speed of sound
C_X, C_Y, C_Z	aerodynamic force coefficients defined in rolling or nonrolling body frame (Equation 4)
C_L, C_M, C_N	aerodynamic moment coefficients defined in rolling or nonrolling body frame (Equation 5)
C_x, C_y, C_z	aerodynamic force coefficients defined in aerodynamic data frame
C_ℓ, C_m, C_n	aerodynamic moment coefficients defined in aerodynamic data frame

C_{x_p} $= \dfrac{\partial C_x}{\partial \left(\frac{pd}{2V_A}\right)}$

C_{ℓ_δ} $= \dfrac{\partial C_\ell}{\partial \delta}$

C_{ℓ_p} $= \dfrac{\partial C_\ell}{\partial \left(\frac{pd}{2V_A}\right)}$

C_{n_p} $= \dfrac{\partial C_n}{\partial \left(\frac{pd}{2V_A}\right)}$

C_{m_q} $= \dfrac{\partial C_m}{\partial \left(\frac{qd}{2V_A}\right)}$

$C_{m_{\delta_A}}$ $= \dfrac{\partial C_m}{\partial \delta_A}$

d	body reference diameter
D	atmospheric density deviation function (Equation 21)
F_x, F_y, F_z	aerodynamic forces defined in rolling or nonrolling body frame (Equation 4)
g_o	acceleration due to gravity at earth's surface
h	altitude of body above earth's surface
I_x	axial moment of inertia of vehicle
I_y	transverse moment of inertia of vehicle about body y axis
I_z	transverse moment of inertia of vehicle about body z axis

$\left[\ell_E\right]$	inertial to earth frame transfer matrix (Equation 33)
$\left[\ell_B\right]$	body to inertial frame transfer matrix (Equations 1 and 28)
$\left[\ell_L\right]$	inertial to local frame transfer matrix (Equation 13)
ℓ_{ij}	elements of $\left[\ell_B\right]$
m	mass of vehicle
M	Mach number (Equation 19)
M_x, M_y, M_z	aerodynamic moments defined in rolling or non-rolling body frame (Equation 5)
p, q, r	components of angular velocity of vehicle in body frame
$\bar{Q}$	dynamic pressure
R_E	radius of earth
R_τ	longitudinal range with respect to the earth (Equation 29)
R_ϕ	latitudinal range with respect to the earth (Equation 30)
t	time
$\bar{V}_A$	velocity of vehicle with respect to local air mass
V_A	speed of vehicle with respect to local air mass = $\lvert\bar{V}_A\rvert$
$\bar{V}_E$	velocity of vehicle with respect to earth
V_E	speed of vehicle with respect to earth = $\lvert\bar{V}_E\rvert$
V_W	speed of local wind
$V_{x_{BA}}, V_{y_{BA}}, V_{z_{BA}}$	components of V_A in body frame
$V_{x_{LA}}, V_{y_{LA}}, V_{z_{LA}}$	components of V_A in local frame

$V_{x_{LE}}, V_{y_{LE}}, V_{z_{LE}}$	components of V_E in local frame
X_A, Y_A, Z_A	aerodynamic data frame axes
X_B, Y_B, Z_B	rolling or nonrolling body frame axes
X_E, Y_E, Z_E	earth frame axes
X_L, Y_L, Z_L	local frame axes
X_R, Y_R, Z_R	inertial frame axes
α	angle of attack (Equation 7)
$\bar{\alpha}$	total angle of attack (Equation 9)
β	angle of sideslip (Equation 8)
γ_A	elevation angle of $\bar{V}_A$ in local frame (Equation 27)
γ_E	elevation angle of $\bar{V}_E$ in local frame (Equation 24)
δ	average fin cant angle
δ_A	effective configurational asymmetry angle in pitch plane
δ_B	effective configurational asymmetry angle in yaw plane
θ	orientation of plane of total angle of attack ($\bar{\alpha}$) and plane of elevation of $\bar{V}_A$ (γ_A) (Figure 4)
ρ	atmospheric density
ρ_A	standard atmospheric density
τ_E	earth longitude (Equation 31)
τ_R	inertial longitude (Equation 14)
φ	orientation of rolling body frame with respect to X_A-Z_A plane (Equation 12)
φ'	orientation of rolling body frame with respect to nonrolling body frame (Equation 11)
ϕ	orientation of rolling or nonrolling body frame with respect to X_A-Z_A plane (Equation 10)

ϕ'	aerodynamic roll angle
ψ_E	earth latitude (Equation 32)
ψ_R	inertial latitude (Equation 15)
ω_E	angular velocity of earth
$(\dot{\ })$	$\frac{d}{dt}$
$(\ddot{\ })$	$\frac{d^2}{dt^2}$
$[\ \ ]^T$	transpcse of matrix
$[\ \ ]^{-1}$	inverse of matrix
$(\ \ )_t$	value of variable at which calculations are automatically terminated

MATHEMATICAL FORMULATION OF PROBLEM

Rigid Body Equations of Motion

The equations of motion are written in a form such that the force equations are integrated in the inertial frame and the moment equations are integrated in the body frame. They are also written in a form which allows either a rolling or non-rolling body frame to be used.

The force equations in the inertial frame are

$$\begin{bmatrix} \ddot{X}_R \\ \ddot{Y}_R \\ \ddot{Z}_R \end{bmatrix} = \frac{\left[\ell_B\right]}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} - \frac{g_o R_E^2}{\left(X_R^2+Y_R^2+Z_R^2\right)^{3/2}} \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} \tag{1}$$

and the moment equations in the inertial frame are

$$\begin{bmatrix} I_x\dot{p} \\ I_y\dot{q} \\ I_z\dot{r} \end{bmatrix} = - \begin{bmatrix} \Omega \end{bmatrix} \begin{bmatrix} I_x p \\ I_y q \\ I_z r \end{bmatrix} + \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{2}$$

where

$$\begin{bmatrix} \Omega \end{bmatrix} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -kp \\ -q & kp & 0 \end{bmatrix}$$

If the rolling body axes are used then k=1 and if the non-rolling body axes are used k=0. When $I_x \neq I_y$, then the rolling body axes must be used.

The derivatives of the elements of the direction cosine matrix $\begin{bmatrix} \ell_B \end{bmatrix}$ are given by the expressions

$$\dot{\ell}_{ij} = \sum_{k=1}^{3} \ell_{ik}\,\Omega_{kj} \qquad i,j = 1,2,3 \tag{3}$$

Equations 1, 2 and 3 form a set of 18 first order simultaneous differential equations describing the motion of the vehicle.

Aerodynamic Force and Moment Equations

The aerodynamic forces and moments in the body frame are now defined. The definitions apply for both rolling and non-rolling body frames. The forces are

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \bar{Q}A \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \tag{4}$$

and the moments are

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \bar{Q}Ad \begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} \tag{5}$$

The force and moment coefficients appearing in the above equations are now written in terms of force and moment coefficients defined in the aerodynamic data frame, with the exception of the moments due to configurational asymmetries which are defined in the rolling body frame. The aerodynamic data and body axes systems are shown in Figure 1. The expressions for the coefficients appearing in equations (4) and (5) are

$$\begin{aligned}
C_X &= C_x + C_{x_p} \frac{pd}{2V_A} \\
C_Y &= C_n \cos \phi + C_z \sin \phi \\
C_Z &= C_z \cos \phi - C_y \sin \phi \\
C_L &= C_\ell + C_{\ell_\delta} \delta + C_{\ell_p} \frac{pd}{2V_A} \\
C_M &= C_m \cos \phi + C_n \sin \phi + C_{n_p} \frac{pd}{2V_A} \sin \phi \\
&\quad + C_{m_q} \frac{qd}{2V_A} + C_{m_{\delta_A}} (\delta_A \cos \varphi' - \delta_B \sin \varphi') \\
C_N &= C_n \cos \phi - C_m \sin \phi + C_{n_p} \frac{pd}{2V_A} \cos \phi \\
&\quad + C_{m_q} \frac{rd}{2V_A} + C_{m_{\delta_A}} (\delta_B \cos \varphi' + \delta_A \sin \varphi')
\end{aligned} \tag{6}$$

where C_x, C_{x_p}, C_{ℓ_δ}, C_{ℓ_p}, C_{m_q}, C_{n_p} and $C_{m_{\delta_A}}$ are functions of M and $\bar{\alpha}$ and C_y, C_z, C_ℓ, C_m and C_n are functions of M, $\bar{\alpha}$ and ϕ'.

When looking up C_{m_q} appearing in the expression for C_M, $\bar{\alpha}$ is taken to be $|\alpha|$ and in the equation for C_N, $\bar{\alpha}$ is taken to be $|\beta|$.

Aerodynamic Orientation Angles

In this section the aerodynamic orientation angles, required for the calculation of the force and moment coefficients given in equation (6), are defined. The angle of attack, sideslip and total angle of attack are shown in Figure 1. It should be noted that for a given orientation of the body with respect to the velocity vector the values of α and β depend upon whether the body axes are rolling or nonrolling. The value of $\bar{\alpha}$, however, is independent of the choice of body axes. Also note that $\bar{V}_A$ and $\bar{\alpha}$ always lie in the $X_A - Z_A$ plane.

$$\alpha = \tan^{-1}\left[\frac{V_{z_{BA}}}{V_{x_{BA}}}\right] \qquad +180^\circ \geq \alpha > -180^\circ \tag{7}$$

Range of α	$V_{z_{BA}}$	$V_{x_{BA}}$
$90^\circ > \alpha > 0^\circ$	+	+
$180^\circ > \alpha > 90^\circ$	+	-
$0^\circ > \alpha > -90^\circ$	-	+
$-90^\circ > \alpha > -180^\circ$	-	-
$\alpha = 0$	0	0

$$\beta = \tan^{-1}\left[\frac{V_{y_{BA}}}{V_{x_{BA}}}\right] \qquad +180^\circ \geq \beta > -180^\circ \tag{8}$$

$$\bar{\alpha} = \tan^{-1}\left[\frac{\sqrt{V_{y_{BA}}^2 + V_{z_{BA}}^2}}{V_{x_{BA}}}\right] \qquad 180° \geq \bar{\alpha} \geq 0° \tag{9}$$

Range of $\bar{\alpha}$	$V_{x_{BA}}$	$V_{y_{BA}}$	$V_{z_{BA}}$
$90° > \bar{\alpha} > 0°$	+		
$180° > \bar{\alpha} > 0°$	-		
$\bar{\alpha} = 0°$	0	0	0

The roll angle ϕ, as is the case with α and β, depends upon whether or not the body frame is rolling

$$\phi = \tan^{-1}\left[\frac{V_{y_{BA}}}{V_{z_{BA}}}\right] \qquad 360° > \phi \geq 0° \tag{10}$$

Range of ϕ	$V_{y_{BA}}$	$V_{z_{BA}}$
$90° > \phi > 0°$	+	+
$180° > \phi > 90°$	+	-
$270° > \phi > 180°$	-	-
$360° > \phi > 270°$	-	-
$\phi = 0°$	0	0

Initially the rolling and nonrolling body frames are coincident. If it is desired to use a nonrolling body frame when the vehicle has a configurational asymmetry or other roll dependent forces and moments, then the roll orientation of the rolling body frame with respect to nonrolling body frame (φ') must be calculated. This roll orientation angle is given by the expression

$$\dot{\varphi}' = (1-k)p \qquad 360° > \varphi' \geq 0° \tag{11}$$

where again

$$k = \begin{cases} 0 & \text{nonrolling body axes} \\ 1 & \text{rolling body axes} \end{cases}$$

The roll orientation of the rolling body axes with respect to the plane of the total angle of attack (φ) is given by

$$\varphi = \phi + \phi' \tag{12}$$

In the case of cruciform finned bodies the roll dependent aerodynamic coefficients can have a 90° or 180° symmetry in φ. If such a symmetry exists, the amount of aerodynamic data required can be reduced considerably by making the coefficients functions of the dummy variable ϕ' instead of φ.

If there is a 90° symmetry in the roll dependent coefficients then

$$\begin{aligned} \phi' &= \varphi & 90° \geq \varphi \geq 0° \\ \phi' &= \varphi - 90° & 180° \geq \varphi > 90° \\ \phi' &= \varphi - 180° & 270° \geq \varphi > 180° \\ \phi' &= \varphi - 270° & 360° > \varphi > 270° \end{aligned}$$

and if there is a 180° symmetry in the roll dependent coefficients then

$$\begin{aligned} \phi' &= \varphi & 180° \geq \varphi \geq 0° \\ \phi' &= \varphi - 180° & 360° > \varphi > 180° \end{aligned}$$

For all other cases $\phi' = \varphi$ and the roll dependent data must be supplied over the range $360° > \phi' \geq 0°$.

Functional Calculations

The functional calculations are those which must be performed in order for the program to operate. The aerodynamic force and moment calculations described in the previous section are of course program calculations but are described separately because of their significance. The remainder of the functional calculations described in this section essentially express the

variables upon which the aerodynamic forces and moments depend, in terms of the integrated variables. Specifically the variables $\bar{Q}$, M, V_A, $V_{x_{BA}}$, $V_{y_{BA}}$, and $V_{z_{BA}}$ must be defined in terms of X_R, Y_R, Z_R, $\dot{X}_R$, $\dot{Y}_R$, $\dot{Z}_R$, and ℓ_{ij}.

The components of the velocity of the body, with respect to the earth, in the local axes (L) are given by the expression

$$\begin{bmatrix} V_{x_{LE}} \\ V_{y_{LE}} \\ V_{z_{LE}} \end{bmatrix} = \left[\ell_L\right] \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{Z}_R \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \omega_E \sqrt{Y_R^2 + Z_R^2} \end{bmatrix} \tag{13}$$

The direction cosine matrix $\left[\ell_L\right]$ is given by the expression

$$\left[\ell_L\right] = \begin{bmatrix} c_{\psi_R} & -s_{\psi_R} c_{\tau_R} & -s_{\psi_R} s_{\tau_R} \\ s_{\psi_R} & c_{\psi_R} c_{\tau_R} & c_{\psi_R} s_{\tau_R} \\ 0 & -s_{\tau_R} & c_{\tau_R} \end{bmatrix}$$

The angles τ_R and ψ_R are shown in Figure 2 and are defined as follows

$$\tau_R = \tan^{-1}\left[\frac{Z_R}{Y_R}\right] \qquad 360° > \tau_R \geq 0° \tag{14}$$

Range of τ_R	Z_R	Y_R
$90° > \tau_R > 0°$	+	+
$180° > \tau_R > 90°$	+	-
$270° > \tau_R > 180°$	-	-
$360° > \tau_R > 270°$	-	+
$\tau_R = 0$	0	0

$$\psi_R = \tan^{-1}\left[\frac{X_R}{\sqrt{Y_R^2 + Z_R^2}}\right] \qquad +180° \geq \psi_R > -180° \tag{15}$$

Range of ψ_R	X_R
$180° > \psi_R > 0°$	+
$0° > \psi_R > -180°$	-
$\psi_R = 0$	0

The components of the velocity of the vehicle, with respect to the local air mass, in the local frame are given by the expression

$$\begin{bmatrix} V_{x_{LA}} \\ V_{y_{LA}} \\ V_{z_{LA}} \end{bmatrix} = \begin{bmatrix} V_{x_{LE}} + V_W \cos A_W \\ V_{y_{LE}} \\ V_{z_{LE}} + V_W \sin A_W \end{bmatrix} \tag{16}$$

The winds are assumed to be locally horizontal with no vertical component (see Figure 3). In the absence of winds the atmosphere is assumed to rotate with the earth and be uniform in density for a given altitude. The local wind speed (V_W) and azimuth (A_W) are measured with respect to the earth. The wind azimuth has the range $360° > A_W \geq 0°$ and is the direction from north from which the wind is moving.

The components of the velocity of the body with respect to the local air mass in the body (B) frame are given by

$$\begin{bmatrix} V_{x_{BA}} \\ V_{y_{BA}} \\ V_{z_{BA}} \end{bmatrix} = \left[\ell_B\right]^T \left[\ell_L\right]^T \begin{bmatrix} V_{x_{LA}} \\ V_{y_{LA}} \\ V_{z_{LA}} \end{bmatrix} \tag{17}$$

The elements of the direction cosine matrix $\left[\ell_B\right]$ are the terms ℓ_{ij} obtained from the integration of the equations of motion.

The total velocity of the body with respect to the local air mass is

$$V_A = \sqrt{V_{x_{BA}}^2 + V_{y_{BA}}^2 + V_{z_{BA}}^2} \tag{18}$$

The Mach number is given by the expression

$$M = \frac{V_A}{c} \tag{19}$$

where the speed of sound (c) is given in reference (4).

The altitude of the vehicle is given by the expression

$$h = \sqrt{X_R^2 + Y_R^2 + Z_R^2} - R_E \tag{20}$$

and the dynamic pressure is

$$\bar{Q} = \tfrac{1}{2}\rho V_A^2$$

The atmospheric density is calculated by the expression

$$\rho = \rho_A(D + 1) \tag{21}$$

where ρ_A is the standard atmospheric density given in reference (4) and D is the density deviation, given as a function of altitude, which allows the use of an arbitrary density profile.

Initial Calculations

These calculations must be performed in order to convert the input data into the variables required for the initiation of the integration process. This means that X_R, Y_R, Z_R, $\dot{X}_R$, $\dot{Y}_R$, $\dot{Z}_R$, and ℓ_{ij} must be expressed in terms of h, V_E, A_E, γ_E, ϕ_E, τ_E,

θ, $\bar{\alpha}$, ϕ, p, q, and r.

Since the earth and interial frames are initially coincident, the initial values of the inertial longitude and latitude are

$$\tau_R = \tau_E, \quad \psi_R = \psi_E$$

The angle of attack and sideslip angle are

$$\alpha = \bar{\alpha} \cos \phi$$

$$\beta = \bar{\alpha} \sin \phi$$

The initial values of the inertial coordinates of the body (X_R, Y_R, and Z_R) are calculated using equations (14), (15) and the equation

$$h = \sqrt{X_R^2 + Y_R^2 + Z_R^2} - R_E \tag{22}$$

The components of V_E in the local frame can now be calculated from the expressions for the earth referenced flight path azimuth and elevation angles

$$A_E = \tan^{-1} \left[\frac{V_{z_{LE}}}{V_{x_{LE}}} \right] \qquad 360° > A_E \geq 0° \tag{23}$$

Range of A_E	$V_{z_{LE}}$	$V_{x_{LE}}$
$90° > A_E > 0°$	+	+
$190° > A_E > 90°$	+	+
$270° > A_E > 180°$	-	+
$360° > A_E > 270°$	-	+
$A_E = 0$	0	0

$$\gamma_E = \tan^{-1}\left[\frac{V_{y_{LE}}}{\sqrt{V_{x_{LE}}^2 + V_{z_{LE}}^2}}\right] \quad -90° \geq \gamma_E \geq +90° \tag{24}$$

Range of γ_E	$V_{y_{LE}}$
$90° \geq \gamma_E < 0$	+
$0° > \gamma_E \geq -90°$	-
$\gamma_E = 0$	0

along with the equation

$$V_E = \sqrt{V_{x_{LE}}^2 + V_{y_{LE}}^2 + V_{z_{LE}}^2} \tag{25}$$

Having the components of V_E in the local frame, the initial values of the inertial velocity components in the inertial frame can be obtained by inverting equation (13).

The initial elements of the direction cosine matrix $\left[\ell_B\right]$ must now be calculated. The matrix $\left[\ell_B\right]$ is the transpose of the direction cosine matrix formed by rotating an axes system, initially coincident with the inertial axes, through a series of angles so that in its final orientation it is coincident with the body frame. The sequence of rotation is

τ_R about X

$-\phi_R$ about Z

$-A_A$ about Y

γ_A about Z

$90°-\theta$ about X

$\bar{a}$ about Y

ϕ about X

The inertial longitude τ_R and latitude ψ_R shown in Figure 2 are giver by equations (14) and (15).

The azimuth and elevation angles of the vehicle's velocity vector with respect to the local air mass are shown in Figure 3 and are defined by the equations

$$A_A = \tan^{-1}\left[\frac{V_{z_{LA}}}{V_{x_{LA}}}\right] \qquad 360° > A_A \geq 0° \tag{26}$$

$$\gamma_A = \tan^{-1}\left[\frac{V_{y_{LA}}}{\sqrt{V_{x_{LA}}^2 + V_{z_{LA}}^2}}\right] \qquad -90° \geq \gamma_A \geq +90° \tag{27}$$

The sign conventions for A_A and γ_A are the same as those for A_E and γ_E given in equations (23) and (24). The components V_E in the local frame are calculated from equations (23), (24), and (25). These components used in equation (16) give the velocity components $V_{x_{LA}}$, $V_{y_{LA}}$, and $V_{z_{LA}}$ required in equations (26) and (27). The angles θ, $\bar{\alpha}$, and ϕ are shown in Figures 1 and 4, and are input data. The angle θ is measured in a plane perpendicular to $\bar{V}_A$ and for $\theta = 0°$ or $180°$ X_A lies in the plane of ν_A. The initial elements of $\left[\ell_B\right]$ are obtained by expanding the following matrix

$$\left[\ell_B\right] = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\bar{\alpha} & 0 & -\sin\bar{\alpha} \\ 0 & 1 & 0 \\ \sin\bar{\alpha} & 0 & \cos\bar{\alpha} \end{bmatrix}\right.$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\theta & \cos\theta \\ 0 & -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} \cos\gamma_A & \sin\gamma_A & 0 \\ -\sin\gamma_A & \cos\gamma_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos A_A & 0 & \sin A_A \\ 0 & 1 & 0 \\ -\sin A_A & 0 & \cos A_A \end{bmatrix}$$

$$\begin{bmatrix} \cos\psi_R & -\sin\psi_R & 0 \\ \sin\psi_R & \cos\psi_R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\tau_R & \sin\tau_R \\ 0 & -\sin\tau_R & \cos\tau_R \end{bmatrix}^T \tag{28}$$

Auxiliary Calculations

Calculations which are optional are described in this section.

The longitudinal and latudinal ranges are given by the expressions

$$R_\tau = R_E \tau_E \tag{29}$$

and

$$R_\psi = R_E \psi_E \tag{30}$$

where

$$\tau_E = \tan^{-1}\left[\frac{Z_E}{Y_E}\right] \qquad 360° \geq \tau_E \geq 0° \tag{31}$$

$$\psi_E = \tan^{-1}\left[\frac{X_E}{\sqrt{Y_E^2 + Z_E^2}}\right] \qquad +180° \geq \psi_E > -180° \tag{32}$$

The sign conventions on the earth longitude and latitude are the same as those in equations (14) and (15). The longitudinal range is always positive and the latitudinal range has the sign of ψ_E.

The earth position coordinates of the body are obtained from the inertial coordinates by the expression

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \left[\ell_E\right] \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} \tag{33}$$

noting that initially the earth and inertial frames are coincident (see Figure 5). The direction cosine matrix $\left[\ell_E\right]$ is given by

$$\left[\ell_E\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega_E t & \sin \omega_E t \\ 0 & -\sin \omega_E t & \cos \omega_E t \end{bmatrix}$$

The time rate of change of the aerodynamic roll angle is

$$\dot{\phi} = \frac{V_{z_{BA}}\left[\frac{F_y}{m} - r\,V_{x_{BA}}\right] - V_{y_{BA}}\left[\frac{F_z}{m} + q\,V_{x_{BA}}\right]}{V_{z_{BA}}^{2} + V_{y_{BA}}^{2}} + p \qquad (34)$$

DESCRIPTION OF MACHINE PROGRAM

This program is coded in FORTRAN IV and has been prepared for running on an IBM 7090 with the IBSYS monitor. It consists of a routine, entitled MAIN, which exercises overall control of the program and fourteen subroutines. Descriptions of all subroutines are given in Appendix A and FORTRAN listings of MAIN and the subroutines SETUP, AERO, DERIV, TERM, OUT and FNOL-3 are given in Tables 1 through 7.

The most important arrays appearing in this program are the KY, Y, C, and D arrays. These arrays are identified in Tables 8 through 11. The KY's are program options and controls and identify tabulated data described below. The Y array contains additional program controls and most of the program variables appearing in the equations of motion and the D's are their first derivatives.

Input

The cards necessary to run this program contain controls, initial conditions, constants and tabulated data.

The tabulated data consist of aerodynamic coefficients, the density deviation (D) and the wind velocity (V_W) and

wind azimuth (A_W). The aerodynamic coefficients depend upon Mach number, angle of attack and roll angle as explained in the section "Aerodynamic Force and Moment Equations." The density deviation, wind velocity and wind azimuth are functions of altitude alone. These tables are prepared in accordance with the instructions in Appendix A, section 7. These data may be introduced into machine memory directly from cards or from cards to special tape and then to machine memory.

a. Tables introduced directly into memory from cards - When using this mode of operation subroutine MEMTAB places the tables in memory. The card deck required in this case is as follows:

Card No.	Item	Format
1	$JOB	
2	$EXECUTE IBJOB	
3	$IBJOB NOSOURCE	
	Trajectory Program Binary Deck	
4	$DATA	
5	ITAB, JO	10I7
6,7	KY(21) - KY(JO)	"
	Tabulated Data	
8	Blank	
9	INT	10I7
10,11	KY(1) - KY(20)	"
12-19	Y(200) - Y(INT)	5E14.6
	Repeat Cards 9-19 for Each Additional Run Required	
20	Blank	
21	Termination (7-8 punch 1st col.)	

For this mode of running, ITAB=0. The number of KY inputs is JO. If there are no winds then the tables V_W and A_W are not required and JO=33 and if winds are present JO=35. The number of Y inputs is INT-200. For this program INT=238.

When the tables are introduced into memory in this manner they must appear in the following order: C_x, C_{x_p}, C_y, C_z, C_ℓ, C_{ℓ_δ}, C_{ℓ_p}, C_m, C_n, C_{m_q}, C_{n_p}, $C_{m_{\delta_A}}$, D, V_W, A_W. If multiple runs are made then all runs must utilize the same tabulated data.

b. Tables placed on special tape and then introduced into memory - The tables must be placed on tape by a separate program, TAPER. When this is done the tables are then transferred to machine memory by subroutine TABLIN. The required card deck for this mode of operation is as follows:

Card No.	Item	Format
1	$JOB	
2	$EXECUTE	
3	$IBJOB	
	TAPER	
4	$DATA	
	Tabulated Data	
5	Blank	
6	Termination	
7	$JOB	
8	$EXECUTE	
9	$IBJOB	
	Trajectory Binary Deck	
10	$DATA	
11	ITAB, JO	10I7
12	INT	"
13-16	KY(1) - KY(JO)	"
17-24	Y(200) - Y(INT)	5E14.6
	Repeat Cards 12-24 for Each Additional Run Required	
25	Blank	
26	Termination (7-8 punch 1st col.)	

For this mode of running ITAB=1. JO and INT are as previously described. If the tabulated data are to be supplied by a previously prepared tape then the first card of the deck will be card number 7.

When tables are placed on the special tape they are implicitly numbered in the order in which they appear. The numbers assigned to KY(21) through KY(35) on the input cards assign a given table on the special tape to a particular KY. In other words, if KY(21)=1 then the first table on the tape contains C_x and so on. It should also be noted that more tables than are required for a given run may be placed on the special tape. The advantage of doing this can be seen for the case when a series of runs are desired each having a different value for C_x. The tables would be prepared just as they would for a single run but with additional tables of C_x included. The value assigned to KY(21) would change, from run to run, depending on the particular table of C_x desired. The numbers assigned to KY(22) through KY(35) would remain the same. This feature is not available when the tables are read directly into memory by MEMTAB since, in this case, C_x is always the first table, C_{x_p} the second, and so on.

Options and Controls

Options and controls, other than those which must be exercised in order to use the required subroutines, will be discussed in this section.

a. Rolling or nonrolling body axes - The value of KY(12) determines whether the body axes roll with the body or are nonrolling.

KY(12) = 0, nonrolling
KY(12) = 1, rolling

The only restriction on the selection of the type of axes to be used is that if $I_y \neq I_z$ the rolling frame must be used. The aerodynamic forces and moments are calculated in a manner such that the data required are the same for both types of axes. The nonrolling axes can be used even when the vehicle has an aerodynamic asymmetry.

Several factors which may influence the choice of the body axes to be used are the output desired and the running time. Since the quantities α, β, ϕ, q, r, F_y, F_z, M_y and M_z are defined in the body axes, their particular values will depend upon the axes chosen. If the body is rapidly spinning the use of nonrolling axes will in general allow a significantly larger integration time step than will the rolling axes.

b. Aerodynamic symmetry - This option applies only to the aerodynamic coefficients which are functions of the aerodynamic roll angle C_y, C_z, C_ℓ, C_m and C_n. The value assigned to KY(11) indicates the degree of roll symmetry which the coefficients have.

KY(11) = 1, 90° symmetry
KY(11) = 2, 180° "
KY(11) = 3, 360° "

For the case of 90° or 180° symmetry in roll the roll dependent coefficients need only be tabulated for roll angles $0° \leq \phi' \leq 90°$ or $0° \leq \phi' \leq 180°$, respectively, instead of the full range 0° to 360°. This option can result in considerable time savings in data preparation and reduction of storage required for aerodynamic data. A more complete discussion of this option is given in the section "Aerodynamic Orientation Angles."

c. Force free calculations - Above the altitude A_H (Y(203)) the aerodynamic forces and moments are set equal to zero. This option considerably shortens the computation procedure when this type of calculation is desired.

d. Winds - If there are no winds KY(7)=0 and if winds are present KY(7)=1. In the event that there are no winds the wind velocity and azimuth tables, designated by KY(34) and KY(35) respectively, do not have to be included.

e. Rotating earth - If Y(204) is equal to zero then the earth is treated as nonrotating. The effects of a rotating earth are included by setting Y(204)= ω_E rad/sec.

f. Correction of $\left[\ell_B\right]$ - This matrix must remain very nearly orthogonal throughout the calculational procedure in order to accurately predict the flight path of the body. Experience has shown that errors in this matrix can be held to

a minimum if it is corrected 3 times each time step by the iterative equation

$$\left[\ell_B\right]_{i+1} = \frac{\left[\ell_B\right]_i + \left[\left[\ell_B\right]^T\right]_i^{-1}}{2}$$

As a check on the orthogonality of $\left[\ell_B\right]$ the matrix $\left[\ell_B\right]\left[\ell_B\right]^T$ can be printed every KY(10) integration time steps.

g. Output printing frequency - The frequency of the output print-out is once every KY(9) integration time steps.

h. Termination - Calculations are automatically terminated when any of the conditions below are satisfied.

$$h = h_t$$
$$t = t_t$$
$$\bar{\alpha} = \bar{\alpha}_t$$

Output - The output data format is shown in Table 13. On the page preceding the first page of output data, the input data are printed.

CONCLUDING REMARKS

As previously stated, one reason for the detailed documentation of this program is to enable the user to modify it to fit his particular needs with a minimum of effort. Once familiar with the program, it is easy to see how the effects of powered flight, thrust and mass asymmetries or a guidance system can be incorporated with a modest effort.

It is felt that this program is capable of very efficient operation. The use of nonrolling body axes and automatic time step adjustment, when the situation favors their application, are proven time savers. The input tables allow an accurate representation of the aerodynamic coefficients with a minimum of time required for their preparation. It should also be noted that by holding the auxiliary calculations and the output to a minimum considerable savings in running time can be realized.

REFERENCES

(1) DeGrafft, W. E., "Analytic Solutions to the Equations of Motion of Missiles Having Six Degrees of Freedom," NOLTR 63-241, Jan 1963 (Unclassified)

(2) Butler, J. F., NAVORD 6766, 1959 (Confidential)

(3) Peterson, V. L., "Motions of a Short 10° Blunted Cone Entering a Martian Atmosphere at Arbitrary Angles of Attack and Arbitrary Pitching Rates," NASA TN D-1326, 1962 (Unclassified)

(4) Minzer, R. A. and others, "The ARDC Model Atmosphere," AFCRC TR 59-267, Aug 1959 (Unclassified)

(5) Linnekin, J. S., Belliveau, L. J., "FNOL-2, A FORTRAN (IBM 7090) Subroutine for the Solution of Ordinary Differential Equations with Automatic Adjustment of the Interval of Integration," NOLTR 63-171, 1963 (Unclassified)

APPENDIX A

DESCRIPTION OF PROGRAM SUBROUTINES

In this appendix the function of each subroutine, as it applies to this program, is briefly described. The FORTRAN II version of FNOL-3 is documented in reference (5). The subroutines TABLIN, MEMTAB, FROMRA, ARDCFT, SINCOS, MATVEC, MATINV and ARKTAN are also documented and available from NOL.

For convenience the call line variables are also given. Unless otherwise indicated T is time, the independent variable, C is the dependent variable and D is its derivative.

1. SETUP (T, C, D) - This subroutine initializes variables and along with MAIN performs initial calculations required for the first integration of the equations of motion.

2. AERO (T, C, D) - The aerodynamic force and moment coefficients are calculated by this subroutine.

3. DERIV (T, C, D) - The derivatives of the dependent variables appearing in the equations of motion are calculated here.

4. TERM (T, C, D, F) - The purpose of this subroutine is twofold. TERM corrects the elements of $\left[\ell_B\right]$ in order to help maintain its orthogonality. This routine also provides for the termination of the calculations as specified by the input data. These functions of TERM are discussed in another section. The variable F in the call line is the variable which determines termination.

5. OUT (T, C, D, ERROR, N, L, H) - Auxiliary calculations which are not required in the actual integration of the equations of motion are performed in this subroutine. OUT also provides for the storage and print-out of the data called for in the output format. H is the integration time step at time T. The terms ERROR, N and L are defined in section 6.

6. FNOL-3 (J, N, G, L, M, ENE, T, C, D, DERIV, TERM, OUT) - This subroutine accomplishes the integration of the equations of motion. It is the FORTRAN IV version of FNOL-2 described in reference (5). This subroutine will numerically integrate a system of up to 30 simultaneous, first order differential equations. The interval of integration can be automatically adjusted to hold the absolute or relative truncation error within specified bounds. The call line variables are defined as follows:

a. J=KY(3) controls the mode of integration. If automatic time step adjustment is desired J=2. Under this mode of integration the Runge-Kutta method is used for the first four time steps, and then the Adams-Moulton predictor-corrector method is used until termination. The integration is completed with four iterations again using the Runge-Kutta method.

b. N=KY(4) is the number of differential equations to be integrated. For the program described in this report, N=19.

c. G=Y(207) is the initial interval of integration.

d. L=KY(5) is always equal to zero in this program.

e. M=KY(16) is always equal to 1 in this program.

f. ENE=Y(206) controls the interval of integration. If ENE=0, the differential equations are integrated with a constant time step equal to G. If ENE>0 then the interval of integration is adjusted so that either the absolute or relative truncation error is maintained within the bounds $\geq 10^{-ENE-3}$ and $\leq 10^{-ENE}$. If the automatic time step adjustment feature is used, the program may restart several times if G is such that the initial absolute or relative truncation errors are not maintained within the specified bounds.

g. ERROR=Y(205) specifies whether or not the time step interval is adjusted on the basis of absolute (T_E) or relative (R_E) truncation error. If ERROR=-1, the adjustment is on the basis of absolute truncation error and if ERROR=0 it is based on the relative truncation error. The formulas

for the errors are

$$T_E = \frac{C^{(p)} - C^{(c)}}{14}, \qquad R_E = \frac{T_E}{C^{(c)}}$$

where $C^{(p)}$ and $C^{(c)}$ are given by the Adams-Moulton predictor-corrector formulas. Care must be exercised in the selection of absolute or relative truncation error as a basis for time step adjustment. For example, if $C^{(p)} = 0(10^{-3})$ and $C^{(c)} = 0(10^{-5})$, then $T_E = 0(10^{-4})$ and $R_E = 0(10)$. In this case if relative truncation error is used the time step interval may be adjusted to a value many times smaller than that required for accurate calculations.

This subroutine has greater generality than indicated here. Reference (5) should be consulted for further details.

7. MEMTAB (KY, Y) - The aerodynamic coefficients, atmospheric density deviation and wind data required by this program are punched on cards in tabular form. These tables are introduced into the machine memory by this subroutine. KY is the array of table numbers given on the input cards and Y is the array in which the tables are stored. In this program the tabulated functions are functions of 1, 2 or 3 variables, with each function in a separate table. The tabulation of a function of 3 variables would be as follows:

a. Control card

L, N, M, n_1, n_2, ... n_N (FORMAT 14I5) where

L=0 all cases

N=3 number of independent variables

M=1 each table contains only 1 function

n_1, n_2, n_3 numbers of values of each independent variable for which values of the function are tabulated.

b. Listing of values of independent variables for which function is tabulated - On the first card/cards the n_1

values of the first independent variable are listed. On the succeeding cards the n_2 and n_3 values of the second and third independent variables are listed. The following restrictions apply. Values of 2 different independent variables may not appear on the same card. At least 2 values of each independent variable must be listed, all values must be distinct and must be in ascending order. The format for these cards is 6E12.7.

c. Listing of values of dependent function - The three independent variables are designated N_1, N_2 and N_3. The number of values of each of these variables is n_1, n_2 and n_3, respectively. A block of data contains the values of the function for all values of N_3 listed and one particular value of N_1 and N_2. The first block corresponds to the first value of N_1 and N_2 listed. The second block corresponds to the first value of N_1 and the second value of N_2. These blocks are repeated until a set of blocks for the first value of N_1 and all values of N_2 have been presented. Sets for the remaining values of N_1 follow until the table is completed. As a check there are n_1 sets, $n_1 \times n_2$ blocks and $n_1 \times n_2 \times n_3$ distinct values of the function. The format for this tabulation is also 6E12.7.

Subroutine LOCOCT is listed as a separate subroutine but is an integral part of MEMTAB.

8. TABLIN (I, J, KY, Y) - This subroutine serves the same purpose as MEMTAB except that it reads tables from tape into memory. I is the number of the tape unit on which the tape containing the tabulated data is mounted. The number of tables on the tape is J, which is automatically set by the program. KY and Y are the same as their counterparts in MEMTAB.

9. FROMRA (KY, N, 1, U_1, U_2,..., U_N, V, Z) - This subroutine extracts data from the tables placed in memory by MEMTAB or TABLIN. Entering FROMRA with the N independent variables U_1, U_2, ..., U_N it linearly interpolates or extrapolates 2^N-1 times and supplies the desired value of the function V. The table is again designated by KY. Z is an array of temporary storage which must have the dimension of ZN + 2^N or greater.

10. ARDCFT (H, P, T, D, C, G) - The earth's atmospheric properties as presented in reference (4) are supplied by this subroutine up to an altitude of 10^6 feet. Entering with the altitude H, the ratios to the corresponding sea level value of pressure (P), temperature (T), density (D), speed of sound (C) and acceleration due to gravity (G) are given.

11. SINCOS (A, SA, CA, N) - This subroutine supplies sin A=SA or cos A=CA for the range $0° \leq A \leq 360°$. If N=0 A is in degrees and if N=1 A is in radians. Subroutines sin and cos must be included as part of the system library.

12. MATVEC (A, B, C, N) - Products of matrices of orders (3,3) and (3,1) are computed by this subroutine.

N=0, $C=AB^*$

N=1, $C=A^TB^*$

N=2, $C=AB$

N=3, $C=A^TB$

N=4, $C=AB^T$

N=5, $C=A^TB^T$

The A, B and C arrays are stored column-wise starting at the left. The symbol * indicates that B in these cases is a (3,1) array. In all other cases A, B, C are (3,3) arrays.

13. MATINV (A, B, C) - This subroutine computes the transpose (B) and inverse (C) of (3,3) matrices (A). If the determinant of A is zero, neither B or C is calculated, a comment is printed and control is returned to the calling program.

14. ARKTAN (A, B, C, N) - Arctangents, defined by the expression $C = \tan^{-1} (A/B)$, are computed by this subroutine. If N=0, C is in degrees and if N=1, C is in radians. The range of the output is $-90° \leq C \leq +90°$. If A=0 B<0, C=-90° and if B>0, C=+90°. Subroutine ATAN must also be included as part of the system library.

TABLE 1

```
C     MAIN ROUTINE - TRAJECTORY PROGRAM 6UUR1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50)
      EXTERNAL DERIV,TERM,OUT
      READ(5,1000)ITAB,JO
      IF(ITAB)1,98,1
   98 READ(5,1000)(KY(I),I=21,JO)
      WRITE(6,1002)(KY(I),I=21,JO)
      CALL MEMTAB(KY(21),Y(600))
    1 DO 3 IJ=1,500
    3 Y(IJ)=0.0
      DO 4 IJ=1,25
      C(IJ)=0.0
    4 D(IJ)=0.0
      DO 5 IJ=1,20
    5 KY(IJ)=0
      READ(5,1000)INT
      IF(INT)100,100,2
    2 IF(ITAB)7,6,7
    6 READ(5,1000)(KY(I),I=1,20)
      WRITE(6,1002)KY(1),KY(2)
      WRITE(6,1002)(KY(I),I=3,20)
      GO TO 8
    7 READ(5,1000)(KY(I),I=1,JO)
      WRITE(6,1002)KY(1),KY(2)
      WRITE(6,1002)(KY(I),I=3,JO)
    8 READ(5,1001)(Y(I),I=200,INT)
      WRITE(6,1003)(Y(I),I=200,INT)
      WRITE(6,1004)
 1000 FORMAT (10I7)
 1001 FORMAT (5E14.6)
 1002 FORMAT(1H0,10I10)
 1003 FORMAT(1H0,1P5E17.6)
 1004 FORMAT (1H1)
      IF(ITAB)26,22,26
   26 IF(KY(7))25,24,25
   24 KYZ=13
      GO TO 21
   25 KYZ=15
   21 CALL TABLIN(18,KYZ,KY(21),Y(600))
   22 CALL SINCOS(Y(223),SN,CN,0)
      Y(299)=(Y(222))*CN
      Y(298)=(Y(222))*SN
      CALL SINCOS(Y(217),SA,CA,0)
      CALL SINCOS(Y(218),SG,CG,0)
      Y(66)=Y(216)*CG*CA
      Y(67)=Y(216)*SG
      Y(68)=Y(216)*CG*SA
      IF(KY(7)) 27,27,28
   27 Y(160)=Y(66)
```

TABLE 1 (Continued)

```
      Y(161)=Y(67)
      Y(162)=Y(68)
      GO TO 29
   28 CALL SINCOS(Y(228),SA,CA,0)
      Y(160)=Y(66)+Y(227)*CA
      Y(161)=Y(67)
      Y(162)=Y(68)+Y(227)*SA
   29 CALL SETUP(T,C,D)
C     LB MATRIX COMPUTED IN SETUP
      C(4)= (Y(229)+Y(215))*Y(47)
      C(5)= (Y(229)+Y(215))*Y(46)*Y(52)
      C(6)= (Y(229)+Y(215))*Y(46)*Y(53)
      Y(37)=0.
      Y(38)=0.
      Y(39)=Y(204)*SQRT(C(5)**2+C(6)**2)
      CALL MATVEC (Y(72),Y(37),Y(40),1)
      CALL MATVEC(Y(72),Y(66),Y(37),1)
      DO 12 I=1,3
      C(I)=Y(I+36)+Y(I+39)
   12 C(I+6)=Y(I+223)
      C(19)=0.
      T=Y(208)
      C(21)=0.
      C(22)=Y(205)
      CALL FNOL3(KY(3),KY(4),Y(207),KY(5),KY(6),Y(206),T,C,D,
     1DERIV,TERM,OUT)
      GO TO 1
  100 STOP
       END
```

```
      SUBROUTINE SETUP(T,C,D)
C     TRAJECTORY PROGRAM 6UUR1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50)
      Y(190)=Y(223)*.0174532925
      CALL SINCOS(Y(190),Y(51),Y(50),1)
      Y(190)=Y(222)*.0174532925
      CALL SINCOS(Y(190),Y(57),Y(56),1)
      Y(190)=Y(219)*.0174532925
      CALL SINCOS(Y(190),Y(59),Y(58),1)
      Y(158)=SQRT(Y(160)**2+Y(162)**2)
      Y(159)=SQRT(Y(160)**2+Y(161)**2+Y(162)**2)
      Y(48)=Y(160)/Y(158)
      Y(49)=Y(162)/Y(158)
      Y(54)=Y(158)/Y(159)
      Y(55)=Y(161)/Y(159)
      Y(157)=Y(221)
      CALL SINCOS (Y(157),Y(47),Y(46),0)
      Y(156)=Y(220)
      CALL SINCOS (Y(156),Y(53),Y(52),0)
      Y(1)=1.
      Y(2)=0.
      Y(3)=0.
      Y(4)=0.
      Y(5)=Y(50)
      Y(6)=-Y(51)
      Y(7)=0.
      Y(8)=Y(51)
      Y(9)=Y(50)
      Y(10)=Y(56)
      Y(11)=0.
      Y(12)=Y(57)
      Y(13)=0.
      Y(14)=1.
      Y(15)=0.
      Y(16)=-Y(57)
      Y(17)=0.
      Y(18)=Y(56)
      Y(19)=1.
      Y(20)=0.
      Y(21)=0.
      Y(22)=0.
      Y(23)=Y(59)
      Y(24)=-Y(58)
      Y(25)=0.
      Y(26)=Y(58)
      Y(27)=Y(59)
      Y(28)=Y(54)
      Y(29)=-Y(55)
      Y(30)=0.
      Y(31)=Y(55)
```

```
Y(32)=Y(54)
Y(33)=0.
Y(34)=0.
Y(35)=0.
Y(36)=1.
Y(163)=Y(48)
Y(164)=0.
Y(165)=-Y(49)
Y(166)=0.
Y(167)=1.
Y(168)=0.
Y(169)=Y(49)
Y(170)=0.
Y(171)=Y(48)
Y(172)=Y(46)
Y(173)=Y(47)
Y(174)=0.
Y(175)=-Y(47)
Y(176)=Y(46)
Y(177)=0.
Y(178)=0.
Y(179)=0.
Y(180)=1.
Y(181)=1.
Y(182)=0.
Y(183)=0.
Y(184)=0.
Y(185)=Y(52)
Y(186)=-Y(53)
Y(187)=0.
Y(188)=Y(53)
Y(189)=Y(52)
CALL MATVEC(Y(163),Y(172),Y(37),2)
CALL MATVEC(Y(28),Y(37),Y(37),2)
CALL MATVEC(Y(19),Y(37),Y(37),2)
CALL MATVEC(Y(10),Y(37),Y(37),2)
CALL MATVEC(Y(181),Y(37),Y(37),2)
CALL MATVEC(Y(37),Y(1),C(10),5)
Y(72)=Y(46)
Y(73)=Y(47)
Y(74)=0.
Y(75)=-Y(47)*Y(52)
Y(76)=Y(46)*Y(52)
Y(77)=-Y(53)
Y(78)=-Y(47)*Y(53)
Y(79)=Y(46)*Y(53)
Y(80)=Y(52)
RETURN
 END
```

TABLE 3

```
      SUBROUTINE AERO(T,C,D)
C     TRAJECTORY PROGRAM 600R1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50),Z(50)
      IF(T) 100,100,200
  100 J=0
  200 IF(ABS(Y(91))-10.E-7)1,1,4
    1 Y(91)=0.
      Y(129)=0.
      IF(Y(92))2,3,3
    2 Y(223)=180.
      Y(130)=-1.
      GO TO 5
    3 Y(223)=0.
      Y(130)=1.0
      GO TO 5
    4 CALL ARKTAN(Y(91),Y(92),Y(223),0)
    5 Y(190)=SQRT(Y(91)**2+Y(92)**2)
      CALL ARKTAN(Y(190),Y(90),Y(222),0)
      CALL ARKTAN(Y(92),Y(90),Y(299),0)
      CALL ARKTAN(Y(91),Y(90),Y(298),0)
      IF(Y(223))15,6,6
   15 Y(223)=Y(223)+360.
    6 CALL SINCOS(Y(223),Y(129),Y(130),0)
      Y(151)=AMOD(C(19),6.28318531)*57.2957795
      CALL SINCOS(Y(151),Y(191),Y(192),0)
      Y(149)=Y(223)+Y(151)
      IF(KY(11)-2) 20,22,25
   20 Y(131)=AMOD(Y(149),90.)
      GO TO 30
   22 Y(131)=AMOD(Y(149),180.0)
      GO TO 30
   25 Y(131)=AMOD(Y(149),360.)
   30 CONTINUE
      DO 40 I=1,3
   40 Y(I+152)=C(I+6)*Y(235)*.5/Y(93)
      CALL FROMRA(KY(21),2,1,Y(110),Y(222),Y(132),Z)
      CALL FROMRA(KY(22),2,1,Y(110),Y(222),Y(117),Z)
      CALL FROMRA(KY(23),3,1,Y(110),Y(222),Y(131),Y(133),Z)
      CALL FROMRA(KY(24),3,1,Y(110),Y(222),Y(131),Y(134),Z)
      CALL FROMRA(KY(25),3,1,Y(110),Y(222),Y(131),Y(135),Z)
      CALL FROMRA(KY(26),2,1,Y(110),Y(222),Y(118),Z)
      CALL FROMRA(KY(27),2,1,Y(110),Y(222),Y(136),Z)
      CALL FROMRA(KY(28),3,1,Y(110),Y(222),Y(131),Y(137),Z)
      CALL FROMRA(KY(29),3,1,Y(110),Y(222),Y(131) Y(138),Z)
      Y(109)=ABS(Y(299))
      CALL FROMRA(KY(30),2,1,Y(110),Y(109),Y(139),Z)
      Y(109)=ABS(Y(298))
      CALL FROMRA(KY(30),2,1,Y(110),Y(109),Y(140),Z)
      CALL FROMRA(KY(31),2,1,Y(110),Y(222),Y(119),Z)
      CALL FROMRA(KY(32),2,1,Y(110),Y(222),Y(105),Z)
```

TABLE 3 (Continued)

```
 Y(111)=Y(152)+Y(117)*Y(153)
 Y(112)=Y(153)*Y(130)+Y(154)*Y(129)
 Y(113)=Y(154)*Y(130)-Y(153)*Y(129)
 Y(114)=Y(135)+Y(236)*Y(118)+Y(136)*Y(153)
 Y(115)=Y(137)*Y(130)+Y(138)*Y(129)+Y(119)*Y(153)*Y(129)+Y(139)*
1Y(154)+Y(237)*Y(105)*Y(192)-Y(238)*Y(105)*Y(191)
 Y(116)=Y(138)*Y(130)-Y(137)*Y(129)+Y(119)*Y(153)*Y(130)+Y(140)*
1Y(155)+Y(238)*Y(105)*Y(192)+Y(237)*Y(105)*Y(191)
 RETURN
  END
```

TABLE 4

```
      SUBROUTINE DERIV(T,C,D)
C     TRAJECTORY PROGRAM 00OR1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50),Z(50)
      Y(108)=SQRT(C(4)**2+C(5)**2+C(6)**2)
      Y(215)=Y(108)-Y(229)
      CALL ARDCFT(Y(215),Y(100),Y(101),Y(102),Y(103),Y(104))
      CALL FROMRA(KY(33),1,1,Y(215),DDV,Z)
      Y(102)=Y(102)+Y(102)*DDV
      Y(109)=SQRT(C(5)**2+C(6)**2)
      Y(46)=Y(109)/Y(108)
      Y(47)=C(4)/Y(108)
      Y(52)=C(5)/Y(109)
      Y(53)=C(6)/Y(109)
C     SETUP LL MATRIX
      Y(72)=Y(46)
      Y(73)=Y(47)
      Y(74)=0.
      Y(75)=-Y(47)*Y(52)
      Y(76)=Y(46)*Y(52)
      Y(77)=-Y(53)
      Y(78)=-Y(47)*Y(53)
      Y(79)=Y(46)*Y(53)
      Y(80)=Y(52)
C     END SETUP OF LL MATRIX
      CALL MATVEC(Y(72),C(1),Y(66),0)
      Y(68)=Y(68)-Y(204)*Y(109)
      IF(KY(7)) 9,9,8
    8 CALL FROMRA(KY(34),1,1,Y(215),Y(227),Z)
      CALL FROMRA(KY(35),1,1,Y(215),Y(228),Z)
      CALL SINCOS(Y(228),SW,CW,0)
      Y(160)=Y(66)+Y(227)*CW
      Y(161)=Y(67)
      Y(162)=Y(68)+Y(227)*SW
      GO TO 10
    9 Y(160)=Y(66)
      Y(161)=Y(67)
      Y(162)=Y(68)
   10 CALL MATVEC(Y(72),Y(160),Y(37),1)
      CALL MATVEC(C(10),Y(37),Y(90),1)
      Y(93)=SQRT(Y(90)**2+Y(91)**2+Y(92)**2)
      Y(110)=Y(93)/(Y(103)*1116.4)
      Y(106)=.00118845*Y(102)*Y(93)**2
      Y(107)=32.174*Y(229)**2/Y(108)**3
      IF(Y(215)-Y(203))15,15,20
   20 Y(111)=0.
      Y(112)=0.
      Y(113)=0.
      Y(114)=0.
      Y(115)=0.
      Y(116)=0.
```

```
      GO TO 16
   15 CALL AERO(T,C,D)
   16 DO 17 I=1,3
      Y(I+142)=Y(106)*Y(234)*Y(I+110)
   17 Y(I+145)=Y(106)*Y(234)*Y(235)*Y(I+113)
      CALL MATVEC(C(10),Y(143),Y(37),D)
      DO 18 I=1,3
      D(I)=Y(I+36)/Y(233)-Y(107)*C(I+3)
   18 D(I+3)=C(I)
      D(7)=(C(9)*C(8)*(Y(231)-Y(232))+Y(146))/Y(230)
      A=KY(12)
      D(8)=(C(7)*C(9)*(A*Y(232)-Y(230))+Y(147))/Y(231)
      D(9)=(C(7)*C(8)*(Y(230)-A*Y(231))+Y(148))/Y(232)
      D(10)=C(13)*C(9)-C(16)*C(8)
      D(11)=C(14)*C(9)-C(17)*C(8)
      D(12)=C(15)*C(9)-C(18)*C(8)
      D(13)=-C(10)*C(9)+C(16)*C(7)*A
      D(14)=-C(11)*C(9)+C(17)*C(7)*A
      D(15)=-C(12)*C(9)+C(18)*C(7)*A
      D(16)=C(10)*C(8)-C(13)*C(7)*A
      D(17)=C(11)*C(8)-C(14)*C(7)*A
      D(18)=C(12)*C(8)-C(15)*C(7)*A
      D(19)=(1.-A)*C(7)
      DO 30 I=1,19
      IF(ABS(D(I))-10.E-8)25,25,27
   25 D(I)=0.
   27 IF(ABS(C(I))-10.E-8)28,28,30
   28 C(I)=0.
   30 CONTINUE
      RETURN
       END
```

TABLE 5

```
      SUBROUTINE TERM(T,C,D,F)
C     TRAJECTORY PROGRAM 6UUR1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50)
      IF(T-Y(208))7,6,7
    6 COUNT=0.
    7 F=1.
      IF(KY(8))5,5,1
    1 DO 3 I=1,3
      Y(300)=C(10)
      Y(301)=C(13)
      Y(302)=C(16)
      Y(303)=C(11)
      Y(304)=C(14)
      Y(305)=C(17)
      Y(306)=C(12)
      Y(307)=C(15)
      Y(308)=C(18)
      CALL MATINV(Y(300),Y(309),Y(318))
      DO 2 J=10,18
    2 C(J)=(C(J)+Y(J+308))/2.
    3 CONTINUE
      COUNT=COUNT+1.
      B=KY(10)
      IF(B-COUNT)4,5,5
    4 CALL MATVEC(C(10),C(10),Y(327),4)
      PRINT 100,(Y(I),I=327,335)
  100 FORMAT(1X,1P9E13.5)
      COUNT=0.
    5 CONTINUE
      IF(Y(218)) 12,12,13
   12 F=Y(215)-Y(200)
      IF (F) 20,20,13
   13 IF(Y(201))16,16,14
   14 IF(T-Y(201))16,19,19
   16 IF(Y(222)-Y(202))50,50,19
   19 F=0.
   20 Y(121)=1.0
   50 RETURN
       END
```

TABLE 6

```
      SUBROUTINE OUT(T,C,D,ERROR,DMY,DNYX,DELT)
C     TRAJECTORY PROGRAM 6UUR1
      COMMON Y,KY
      DIMENSION C(25),D(25),Y(8000),KY(50),ERROR(25)
   10 FORMAT(1H0,5X,4HTIME,12X,1HH,16X,2HVE,15X,2HVA,16X,1HM,
     114X,4HQBAR,11X,7HGAMMA E//)
   11 FORMAT(1H0,4X,4HTIME,10X,6HALPHA-,11X,3HPHI,10X,7HPHI PRI
     1,9X,5HALPHA,10X,4HBETA,13X,1HQ,13X,1HR)
   12 FORMAT(1H0,5X,4HTIME,15X,4HDELT,18X,1HP,19X,4HRTAU,17X,
     14HRPSI,16X,6HPHIDOT//)
   13 FORMAT(1H0,5X,4HTIME,12X,2HFX,15X,2HFY,15X,2HFZ,15X,2HMX,
     115X,2HMY,15X,2HMZ//)
   14 FORMAT(1H0,5X,4HTIME,11X,4HDDXR,13X,4HDDYR,13X,4HDDZR,
     113X,3HDXR,14X,3HDYR,14X,3HDZR//)
    1 FORMAT(3X,4HRUN ,I3,95X,7HFORMAT ,I1)
    2 FORMAT(3X,I6,98X,5HPAGE ,I3)
    3 FORMAT(1PE13.4,1P6E17.7)
    4 FORMAT(1PE13.4,1P5E21.7)
    5 FORMAT(1H0)
    6 FORMAT(1H1)
    7 FORMAT(1PE13.4,1P7E15.5)
      IF(T-Y(208))105,100,105
  100 KK=0
      N=0
      KZ=0
      J=0
      GO TO 115
  105 KK=KK+1
      IF(Y(121)) 110,110,115
  110 IF (KK-KY(9)) 320,111,320
  111 KK=0
      IF(Y(142)-T) 115,320,115
  115 KZ=KZ+1
      IF(KZ-46)200,15,15
   15 N=N+1
      DO 30 L=1,5
      WRITE(6,1)KY(2),L
      WRITE(6,2)KY(1),N
      GO TO (20,21,22,23,24),L
   20 WRITE(6,10)
      M=7*KZ+5000
      WRITE(6,3)(Y(I),I=5001,M)
      GO TO 26
   21 WRITE(6,11)
      M=8*KZ+5315
      WRITE(6,7)(Y(I),I=5316,M)
      GO TO 26
   22 WRITE(6,12)
      M=6*KZ+5675
      WRITE(6,4)(Y(I),I=5676,M)
      GO TO 26
```

TABLE 6 (Continued)

```
   23 WRITE(6,13)
      M=7*KZ+5945
      WRITE(6,3)(Y(I),I=5946,M)
      GO TO 26
   24 WRITE(6,14)
      M=7*KZ+6260
      WRITE(6,3)(Y(I),I=6261,M)
      GO TO 26
   26 WRITE(6,6)
   30 CONTINUE
      J=0
      K=0
      JK=0
      KZ=0
      GO TO 320
  200 Y(190)=Y(204)*T
      CALL SINCOS(Y(190),SOT,COT,1)
C     SETUP LE MATRIX
      Y(37)=1.
      Y(38)=0.
      Y(39)=0.
      Y(40)=0.
      Y(41)=COT
      Y(42)=-SOT
      Y(43)=0.
      Y(44)=SOT
      Y(45)=COT
      CALL MATVEC(Y(37),C(4),Y(69),0)
      CALL ARKTAN(Y(71),Y(70),Y(220),1)
      IF(ABS(Y(220))-0.1) 40,40,41
   40 Y(158)=Y(71)/Y(70)
      Y(94)=Y(229)/Y(70)*Y(71)*(1.-Y(158)**2/3.+Y(158)**4/5.)
      GO TO 42
   41 Y(94)=Y(229)*Y(220)
   42 Y(190)=SQRT(Y(70)**2+Y(71)**2)
      CALL ARKTAN(Y(69),Y(190),Y(221),1)
      IF(ABS(Y(221))-0.1) 43,43,44
   43 Y(159)=Y(69)/Y(190)
      Y(95)=Y(229)/Y(190)*Y(69)*(1.-Y(159)**2/3.+Y(159)**4/5.)
      GO TO 45
   44 Y(95)=Y(229)*Y(221)
   45 Y(141)= ((Y(92)*(Y(144)/Y(110)-C(9)*Y(90))-Y(91)*
     1(Y(145)/Y(110)+C(8)*Y(90)))/(Y(92)**2+Y(91)**2))+C(7)
      Y(190)=SQRT(Y(66)**2+Y(68)**2)
      CALL ARKTAN(Y(67),Y(190),Y(218),0)
      Y(216)=SQRT(Y(66)**2+Y(67)**2+Y(68)**2)
      IF (J) 310,300,310
  300 J = 1
      K=1
      JK=1
      GO TO 311
```

TABLE 6 (Continued)

```
310 J=J+7
    K=K+8
    JK=JK+6
311 Y(J+5000)=T
    Y(J+5001)=Y(215)
    Y(J+5002)=Y(216)
    Y(J+5003)=Y(93)
    Y(J+5004)=Y(110)
    Y(J+5005)=Y(106)
    Y(J+5006)=Y(218)
    Y(K+5315)=T
    Y(K+5316)=Y(222)
    Y(K+5317)=Y(223)
    Y(K+5318)=Y(151)
    Y(K+5319)=Y(299)
    Y(K+5320)=Y(298)
    Y(K+5321)=C(8)
    Y(K+5322)=C(9)
    Y(JK+5675)=T
    Y(JK+5676)=DELT
    Y(JK+5677)=C(7)
    Y(JK+5678)=Y(94)
    Y(JK+5679)=Y(95)
    Y(JK+5680)=Y(141)
    Y(J+5945)=T
    Y(J+5946)=Y(143)
    Y(J+5947)=Y(144)
    Y(J+5948)=Y(145)
    Y(J+5949)=Y(146)
    Y(J+5950)=Y(147)
    Y(J+5951)=Y(148)
    Y(J+6260)=T
    Y(J+6261)=D(1)
    Y(J+6262)=D(2)
    Y(J+6263)=D(3)
    Y(J+6264)=D(4)
    Y(J+6265)=D(5)
    Y(J+6266)=D(6)
    IF(Y(121))315,315,15
315 IF(KZ-45) 320,15,320
320 Y(142)=T
350 RETURN
     END
```

TABLE 7

```
      SUBROUTINE FNOL3(J,N,G,L,M,ENE,X,Y,D,DERIV,TERM,OUTPUT)
C     FORTRAN IV      11-63
      DIMENSION Y(50),D(50),YB(30,6),GI2(30),GI3(30),GI4(30),EF(30),
     1EF1(30),EF2(30),EF3(30),Y1(30),ERROR(30),HA(30),YA(50),DA(50),
     2YC(30),YP(30),YD(50)
      DOUBLE PRECISION XD,YD,YA,YC,YP,Y1
      EC=Y(N+3)
    1 H=G
    2 HZ=H
    3 LN=N+MAX0(L,3)
    4 NA=0
    5 NB=1
    6 NF=0
    7 NG=0
    8 F=0.
    9 FA=0.
   10 FB=0.
   11 FC=0.
   12 FD=0.
   13 NE=ENE
      DO 200 I=1,LN
  200 YD(I)=DBLE(Y(I))
      XD=DBLE(X)
   14 IF(J-3)15,21,15
   15 IF(NE)18,16,18
   16 JA=4
   17 GO TO 22
   18 RE1=10.**(-ENE)
   19 RE2=10.**(-ENE-3.0)
   20 REM=10.**(-ENE-1.5)
   21 JA=1
   22 DO 25 I=1,N
   23 DO 24 IC=1,5
   24 YB(I,IC)=0.
   25 ERROR(I)=0.
   26 CALL DERIV(X,Y,D)
      DO 300 I=1,N
      GI2(I)=D(I)
      GI3(I)=D(I)
      GI4(I)=D(I)
  300 EF(I)=D(I)
   27 CALL OUTPUT(X,Y,D,ERROR,N,L,H)
   28 FD=Y(N+1)
   29 IF(J-2) 30,129,30
   30 GO TO(31,37,35,37),JA
   31 DO 33 I=1,LN
   32 YA(I)=YD(I)
   33 DA(I)=D(I)
   34 GO TO 37
   35 HB=H
   36 H=2.*H
```

TABLE 7 (Continued)

```
 37 HD2 = .5*H
    DO 39 I=1,N
 38 YB(I,NB)=D(I)
    XL = D(I) * HD2
 39 Y(I)=SNGL(YD(I)+XL)
    X=SNGL(XD+HD2)
 40 CALL DERIV (X,Y,GI2)
 41 DO 42 I=1,N
    XL = GI2(I)*HD2
 42 Y(I)=SNGL(YD(I)+XL)
 43 CALL DERIV (X,Y,GI3)
 44 DO 45 I=1,N
    XL=GI3(I)*H
 45 Y(I)=SNGL(YD(I)+XL)
    X=SNGL(XD+H)
 46 CALL DERIV(X,Y,GI4)
 47 HD6 =H/6.
    GO TO(48,55,60,66),JA
 48 DO 52 I=1,N
    XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
 49 YC(I)=YD(I)+XL
 51 YD(I)=YA(I)
 52 ERROR(I)=0.
 53 JA=3
 54 GO TO 35
 55 DO 57 I=1,N
    XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
 56 YD(I)=YD(I)+XL
 57 ERROR(I)=SNGL(YD(I)-YP(I))/15.
 58 JA=1
 59 GO TO 681
 60 DO 62 I=1,N
 61 YD(I)=YC(I)
    XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
 62 YP(I)=YA(I)+XL
 63 H=HB
 64 JA=2
 65 GO TO 681
 66 DO 68 I=1,N
    XL=(D(I) + 2.*(GI2(I) + GI3(I)) +GI4(I))*HD6
 67 YD(I)=YD(I)+XL
 68 ERROR(I)=0.
681 DO 69 I=1,N
 69 Y(I)=SNGL(YD(I))
    XD=XD+H
    X=SNGL(XD)
 70 CALL DERIV(X,Y,D)
 71 FC=F
 72 CALL TERM(X,Y,D,F)
 73 IF(ABS(F)-1.0E-5 )731,731,733
731 NF=5
```

TABLE 7 (Continued)

```
732 GO TO 124
733 IF(F)74,124,76
 74 FA=1.
 75 GO TO 77
 76 FB=1.
 77 IF(FA-FB)83,78,83
 78 NF=NF+1
 79 JA=4
 80 NB=1
 81 H=H*F/(FC-F)
 82 IF(NF-4)37,37,124
 83 IF(NE)84,117,84
 84 IF(JA-1)117,85,117
 85 IF(J-3)86,117,86
 86 DO 95 I=1,N
    IF(Y(I))886,885,886
885 HA(I)=1000.
    GO TO 95
886 IF(EC)880,890,87
 87 IF(ABS(Y(I))-EC) 880,880,890
880 IF(ABS(ERROR(I))-RE2) 882,94,881
881 IF(ABS(ERROR(I))-RE1)94,94,882
882 HA(I)=H*(REM/(ABS(ERROR(I))+.0000000001))**(.2)
883 GO TO 95
890 IF(ABS(ERROR(I)/Y(I))-RE2)892,94,891
891 IF(ABS(ERROR(I)/Y(I))-RE1)94,94,892
892 HA(I)=H*(REM/(ABS(ERROR(I)/Y(I))+.0000000001))**(.2)
893 GO TO 95
 94 HA(I)=H
 95 CONTINUE
 96 HB=HA(N)
 97 DO 98 I=1,N
 98 HB=AMIN1(HA(I),HB)
 99 IF(H-HB)100,117,101
100 IF(HZ-H)101,101,116
101 DO 103 I=1,LN
102 YD(I)=YA(I)
    Y(I)=SNGL(YD(I))
103 D(I)=DA(I)
104 IF(NB-6)107,105,105
105 XD=XD-H
106 GO TO 109
107 XD=XD-2.*H
108 HZ=H
109 H=HB
    X=SNGL(XD)
    CALL DERIV(X,Y,D)
110 NB=1
111 XABS=ABS(.000001*X)
112 IF(ABS(H)-XABS)113,113,117
113 NG=NG+1
```

TABLE 7 (Continued)

```
 114 H=SIGN(XABS,HB)
 115 IF(NG-10)124,126,126
 116 HZ=H
 117 IF(M)118,118,121
 118 IF(ABS(Y(N+1)-FD)-Y(N+2))29,119,119
 119 FD=Y(N+1)
 120 GO TO 124
 121 NA=NA+1
 122 IF(M-NA)123,123,29
 123 NA=0
124  CALL OUTPUT(X,Y,D,ERROR,N,L,H)
 125 IF(NF-4)29,29,126
 126 WRITE (6,127)
 127 FORMAT(1H0)
 128 RETURN
 129 NB=NB+1
 130 IF(NB-6)30,131,136
 131 DO 134 I=1,N
 132 EF3(I)=YB(I,3)
 133 EF2(I)=YB(I,4)
 134 EF1(I)=YB(I,5)
 135 GO TO 137
 136 NB=10
 137 HD24 =H/24.
     DO 138 I=1,N
     XL =(55.*D(I) -59.*EF1(I) +37.*EF2(I) -9.*EF3(I))*HD24
     YP(I)=YD(I)+XL
 138 Y(I)=SNGL(YP(I))
     X=SNGL(XD+H)
 139 CALL DERIV(X,Y,EF)
 140 DO 142 I=1,LN
 141 YA(I)=YD(I)
 142 DA(I)=D(I)
 143 DO 148 I=1,N
     XL = (9.*EF(I) +19.*D(I) -5.*EF1(I) +EF2(I))*HD24
 144 YD(I)=YD(I)+XL
 145 ERROR(I)=-SNGL(YD(I)-YP(I))/14.
 146 EF3(I)=EF2(I)
 147 EF2(I)=EF1(I)
 148 EF1(I)=D(I)
 149 GO TO 681
      END
```

TABLE 8

KY ARRAY VARIABLES

KY(1)	Date
KY(2)	Run No.
KY(3)	J
KY(4)	N
KY(5)	L
KY(6)	M
KY(7)	Wind
KY(8)	$\left[\ell_B\right]$ correction
KY(9)	Print Freq.
KY(10)	Print Freq. $\left[\ell_B\right]$ $\left[\ell_B\right]^T$
KY(11)	Aero Option
KY(12)	k
KY(13)-KY(20)	Available for input
KY(21)	C_x
KY(22)	C_{x_p}
KY(23)	C_y
KY(24)	C_z
KY(25)	C_ℓ
KY(26)	C_{ℓ_δ}
KY(27)	C_{ℓ_p}
KY(28)	C_m
KY(29)	C_n
KY(30)	C_{m_q}
KY(31)	C_{n_p}
KY(32)	$C_{m_{\delta_A}}$
KY(33)	D
KY(34)	V_W
KY(35)	A_W
KY(36)-KY(50)	Available for input

TABLE 9

Y ARRAY VARIABLES

Y(1) - Y(36)	Setup only
Y(37) - Y(65)	Intermediate storage
Y(66)	$V_{x_{LE}}$
Y(67)	$V_{y_{LE}}$
Y(68)	$V_{z_{LE}}$
Y(69)	X_E
Y(70)	Y_E
Y(71)	Z_E
Y(72) - Y(80)	$\left[\ell_L\right]$ array
Y(81) - Y(89)	Available for program
Y(90)	$V_{x_{BA}}$
Y(91)	$V_{y_{BA}}$
Y(92)	$V_{z_{BA}}$
Y(93)	V_A
Y(94)	R_τ
Y(95)	R_ψ
Y(96) - Y(104)	Intermediate storage
Y(105)	$C_{m_{\delta_A}}$

Y ARRAY VARIABLES (Continued)

Y(106)	$\bar{Q}$
Y(107) - Y(109)	Intermediate Storage
Y(110)	M
Y(111) - Y(116)	Intermediate Storage
Y(117)	C_{x_p}
Y(118)	C_{ℓ_δ}
Y(119)	C_{n_p}
Y(120) - Y(130)	Intermediate Storage
Y(131)	ϕ'
Y(132)	C_x
Y(133)	C_y
Y(134)	C_z
Y(135)	C_ℓ
Y(136)	C_{ℓ_p}
Y(137)	C_m
Y(138)	C_n
Y(139)	C_{m_q} (pitch)
Y(140)	C_{m_q} (yaw)
Y(141)	$\dot{\phi}$
Y(142)	Intermediate Storage

Y ARRAY VARIABLES (Continued)

Y(143) F_x

Y(144) F_y

Y(145) F_z

Y(146) M_x

Y(147) M_y

Y(148) M_z

Y(149) - Y(162) Intermediate Storage

Y(163) - Y(189) Setup only

Y(190) - Y(199) Intermediate Storage

Y(200) h_t (ft)

Y(201) t_t (sec)

Y(202) $\bar{\alpha}_t$ (deg)

Y(203) A_H (ft)

Y(204) ω_E (rad/sec)

Y(205) Error

Y(206) ENE

Y(207) Δt_o (sec)

Y(208) t_o (sec)

Y(209) - Y(214) Available for input data

Y(215) h (ft)

Y(216) V_E (ft/sec)

Y ARRAY VARIABLES (Continued)

Y(217) A_E (deg)

Y(218) γ_E (deg)

Y(219) θ (deg)

Y(220) τ_E (deg)

Y(221) ψ_E (deg)

Y(222) $\bar{\alpha}$ (deg)

Y(223) ϕ (deg)

Y(224) p (rad/sec)

Y(225) q (rad/sec)

Y(226) r (rad/sec)

Y(227) V_W (ft/sec)

Y(228) A_W (deg)

Y(229) R_E (ft)

Y(230) I_x (slug-ft^2)

Y(231) I_y (slug-ft^2)

Y(232) I_z (slug-ft^2)

Y(233) m (slug)

Y(234) A (ft^2)

Y(235) d (ft)

Y(236) δ (rad)

Y(237) δ_A (rad or deg depending on $C_{m_{\delta_A}}$)

Y ARRAY VARIABLES (Continued)

Y(238) δ_B (rad or deg depending on $C_{m_{\delta_A}}$)

Y(239) - Y(297) Available for input

Y(298) β

Y(299) α

Y(300) - Y(335) Intermediate storage

Y(336) - Y(599) Available for program

Y(600) - Y(4999) Table storage

Y(5000) - Y(8000) Output storage

TABLE 10

C ARRAY VARIABLES

C(1)	$\dot{X}_R$	C(12)	ℓ_{31}
C(2)	$\dot{Y}_R$	C(13)	ℓ_{12}
C(3)	$\dot{Z}_R$	C(14)	ℓ_{22}
C(4)	X_R	C(15)	ℓ_{32}
C(5)	Y_R	C(16)	ℓ_{13}
C(6)	Z_R	C(17)	ℓ_{23}
C(7)	p	C(18)	ℓ_{33}
C(8)	q	C(19)	φ'
C(9)	r	C(20)	Blank
C(10)	ℓ_{11}	C(21)	
C(11)	ℓ_{21}	C(22)	Error Control

TABLE 11

D ARRAY VARIABLES

D(1)	$\ddot{X}_R$	D(10)	$\dot{\ell}_{11}$
D(2)	$\ddot{Y}_R$	D(11)	$\dot{\ell}_{21}$
D(3)	$\ddot{Z}_R$	D(12)	$\dot{\ell}_{31}$
D(4)	$\dot{X}_R$	D(13)	$\dot{\ell}_{12}$
D(5)	$\dot{Y}_R$	D(14)	$\dot{\ell}_{22}$
D(6)	$\dot{Z}_R$	D(15)	$\dot{\ell}_{32}$
D(7)	$\dot{p}$	D(16)	$\dot{\ell}_{13}$
D(8)	$\dot{q}$	D(17)	$\dot{\ell}_{23}$
D(9)	$\dot{r}$	D(18)	$\dot{\ell}_{33}$
		D(19)	$\dot{\varphi}'$

TABLE 12

OUTPUT FORMAT

FORMAT 1

t	h	V_E	V_A	M	$\bar{Q}$	γ_E
(sec)	(ft)	(ft/sec)	(ft/sec)		(lb/ft^2)	(deg)

FORMAT 2

t	$\bar{\alpha}$	ϕ	φ'	α	β	q	r
(sec)	(deg)	(deg)	(deg)	(deg)	(deg)	(rad/sec)	(rad/sec)

FORMAT 3

t	Δt	p	R_τ	R_ψ	$\dot{\phi}$
(sec)	(sec)	(rad/sec)	(ft)	(ft)	(rad/sec)

FORMAT 4

t	F_x	F_y	F_z	M_x	M_y	M_z
(sec)	(lb)	(lb)	(lb)	(ft-lb)	(ft-lb)	(ft-lb)

FORMAT 5

t	$\ddot{X}_R$	$\ddot{Y}_R$	$\ddot{Z}_R$	$\dot{X}_R$	$\dot{Y}_R$	$\dot{Z}_R$
(sec)	(ft/sec^2)	(ft/sec^2)	(ft/sec^2)	(ft/sec)	(ft/sec)	(ft/sec)

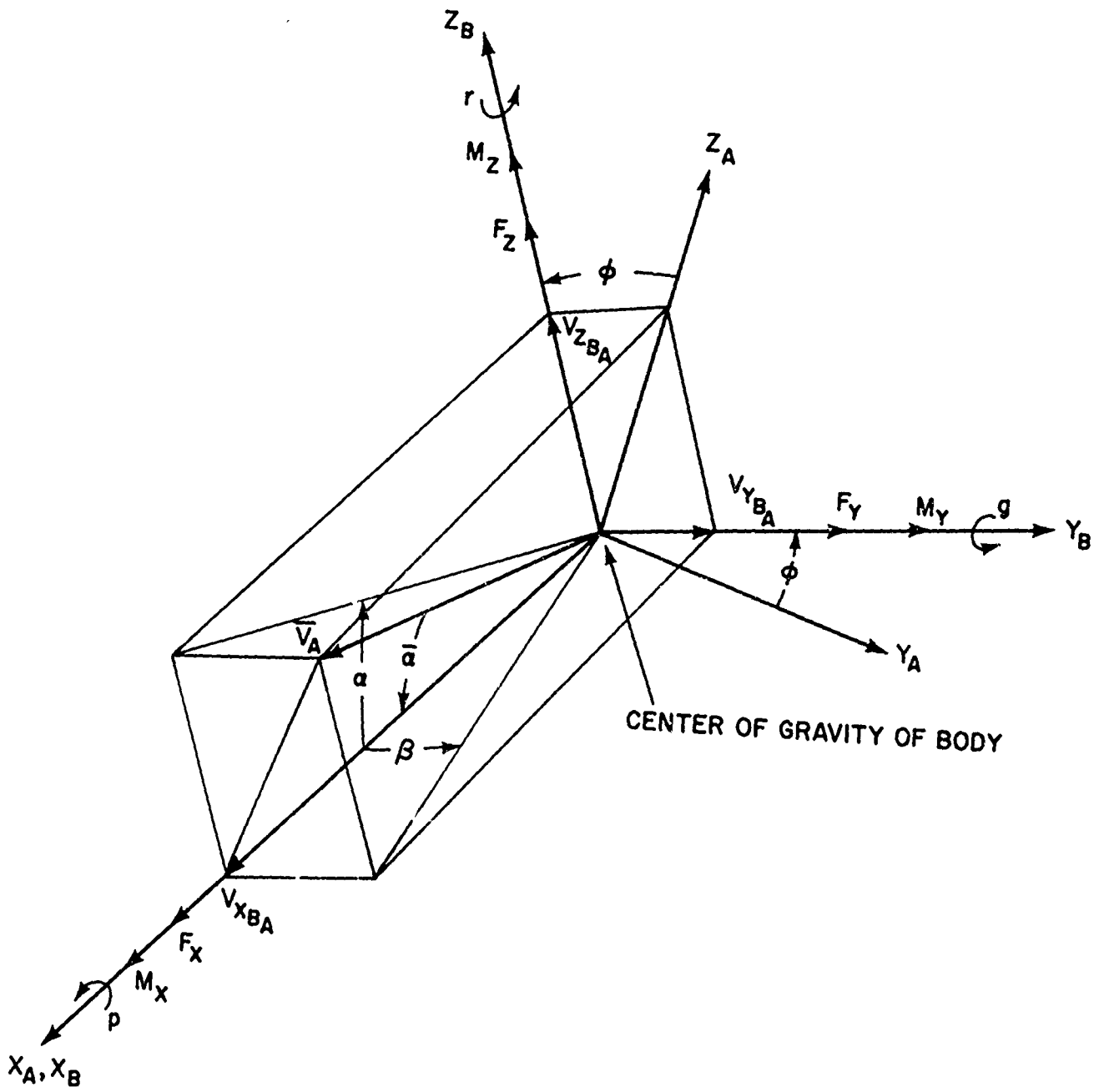

FIG. 1 AERODYNAMIC DATA AND BODY AXES SYSTEMS

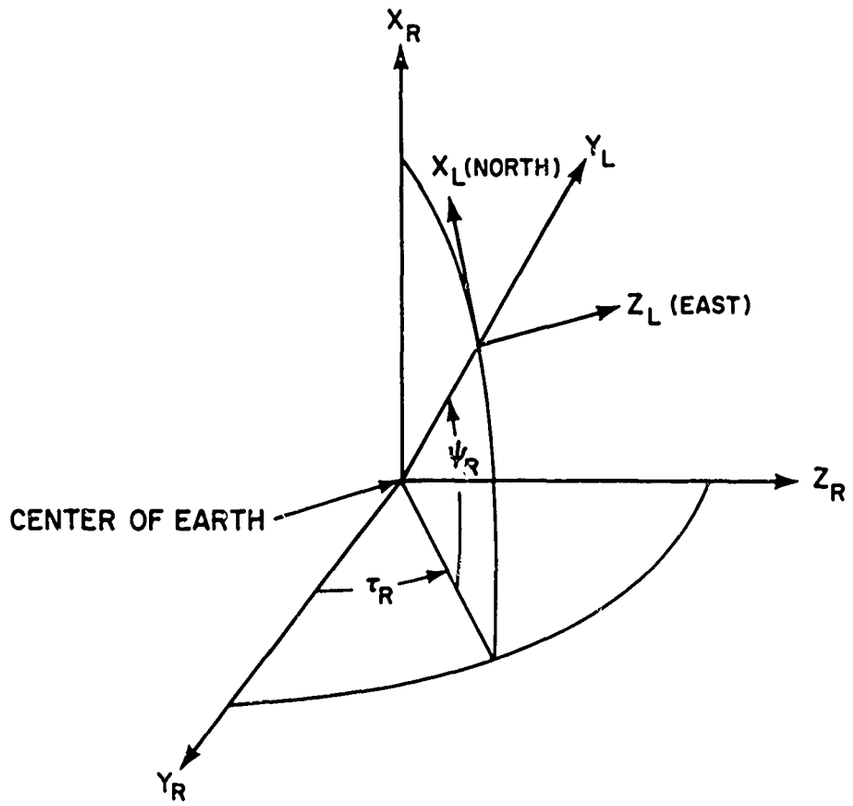

FIG.2 INERTIAL AND LOCAL AXES SYSTEMS

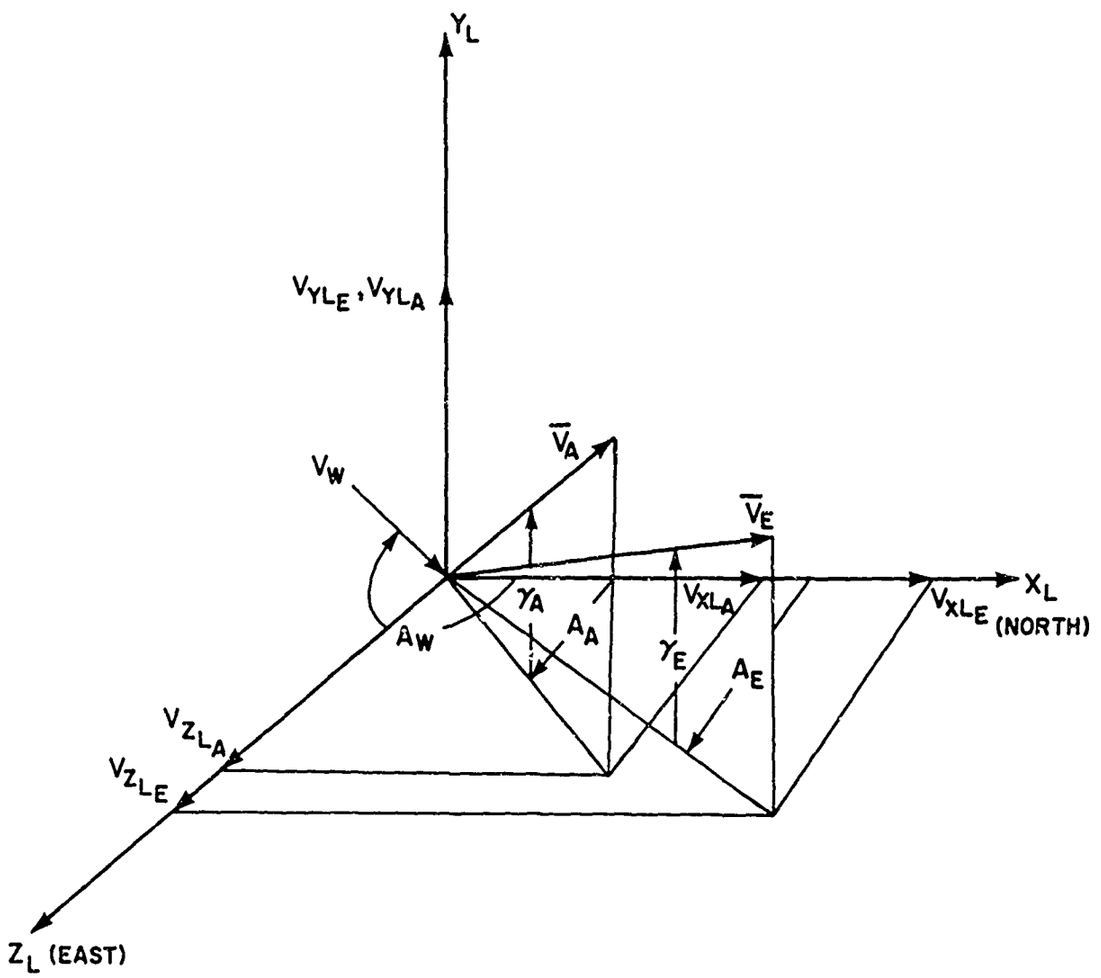

FIG.3 VELOCITY COMPONENTS IN LOCAL AXES SYSTEM

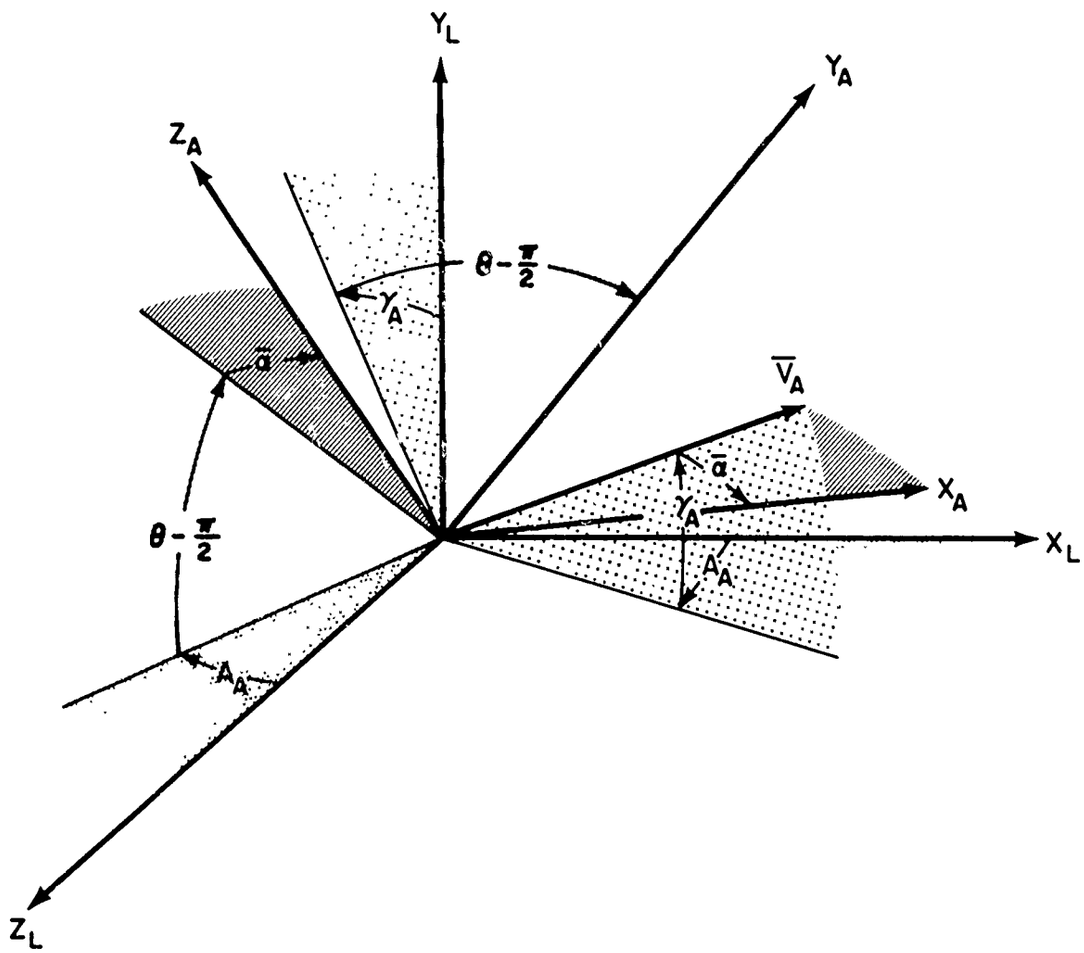

FIG.4 LOCAL AND AERODYNAMIC DATA AXES SYSTEMS

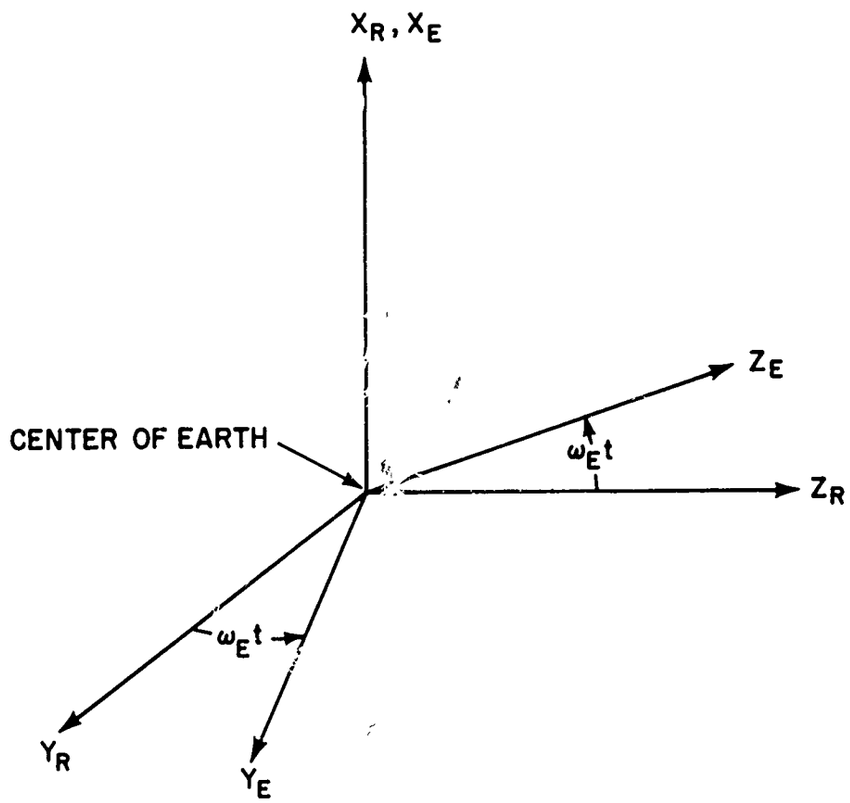

FIG.5 EARTH AND INERTIAL AXES SYSTEMS

AERODYNAMICS DEPARTMENT
EXTERNAL DISTRIBUTION LIST (A1)

	No. of Copies
Chief, Bureau of Naval Weapons Department of the Navy Washington, D. C. 20360 Attn: DLI-3 Attn: R-14 Attn: RRRE-4 Attn: RMGA-811 Attn: RMMC-42	2
Office of Naval Research T-3 Washington, D. C. Attn: Head, Structural Mechanics Branch Attn: Head, Fluid Dynamics Branch	
Director, David Taylor Model Basin Aerodynamics Laboratory Washington, D. C. Attn: Library	
Commander, U. S. Naval Ordnance Test Station China Lake, California Attn: Technical Library Attn: Code 406	
Director, Naval Research Laboratory Washington, D. C. Attn: Code 2027	
Commanding Officer Office of Naval Research Branch Office Box 39, Navy 100 Fleet Post Office New York, New York	
NASA High Speed Flight Station Box 273 Edwards Air Force Base, California	
NASA Ames Research Center Moffett Field, California Attn: Librarian	

AERODYNAMICS DEPARTMENT
EXTERNAL DISTRIBUTION LIST (A1)

	No. of Copies
NASA Langley Research Center Langley Station, Hampton, Virginia Attn: (Mrs.) Elizabeth R. Gilman, Librarian, Bldg. 1244 Attn: C. H. McLellan Attn: Adolph Busemann Attn: Comp. Res. Div. Attn: Theoretical Aerodynamics Division	
NASA Lewis Research Center 21000 Brookpark Road Cleveland 11, Ohio Attn: Librarian Attn: Chief, Propulsion Aerodynamics Division	
NASA 600 Independence Ave., S. W. Washington, D. C. Attn: Chief, Division of Research Information Attn: Dr. H. H. Kurzweg, Director of Research	
Office of the Assistant Secretary of Defense (R&D) Room 3E1065, The Pentagon Washington 25, D. C. Attn: Technical Library	
Research and Development Board Room 3D1041, The Pentagon Washington 25, D. C. Attn: Library	
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
Commander, Pacific Missile Range Point Mugu, California Attn: Technical Library	
Commanding General Aberdeen Proving Ground, Maryland Attn: Technical Information Branch Attn: Ballistic Research Laboratories	

AERODYNAMICS DEPARTMENT
EXTERNAL DISTRIBUTION LIST (A1)

	No. of Copies
Commander, Naval Weapons Laboratory Dahlgren, Virginia Attn: Library	
Director, Special Projects Department of the Navy Washington 25, D. C. Attn: SP-2722	
Director of Intelligence Headquarters, USAF Washington 25, D. C. Attn: AFOIN-3B	
Headquarters - Aero. Systems Division Wright-Patterson Air Force Base Dayton, Ohio Attn: WWAD Attn: RRLA-Library	2
Commander Air Force Ballistic Systems Division Norton Air Force Base San Bernardino, California Attn: BSRVA	2
Chief, Defense Atomic Support Agency Washington 25, D. C. Attn: Document Library	
Headquarters, Arnold Engineering Development Center ARO, Inc. Arnold Air Force Station, Tennessee Attn: Technical Library Attn: AEOR Attn: AEOIM	
Commanding Officer, Harry Diamond Laboratories Washington 25, D. C. Attn: Library, Room 211, Bldg. 92	
Commanding General U. S. Army Missile Command Redstone Arsenal, Alabama Attn: AMSMI-RR (Mr. N. Shapiro) Attn: AMSMI-RB (Redstone Scientific Information Center)	