AFFDL-TR-65-194

A FORTRAN IV PROGRAM TO DERIVE THE EQUATIONS OF MOTION OF SYSTEMS

R. A. WESTERWICK AND J. B. BROWN

CONVAIR DIVISION OF GENERAL DYNAMICS

Distribution of this document is unlimited

FOREWORD

This report was prepared by General Dynamics Convair dynamics group personnel as a part of the R&D program performed under Contract Number AF 33(615)-1836. The program was monitored by the Flight Control Division, Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Systems Command, under Project Number 8225, Task Number 822501. The Air Force project engineer was Mr. Dave Frearson. The Convair project leader was Mr. R. A. Westerwick.

The purpose of this program was to develop a digital computer program to derive the general nonnumeric equations of motion necessary for the analysis of flight control systems. This report presents this program along with its principles of operation and directions for use.

Manuscript of this report released by the authors October 1965 for publication as an RTD Technical Report.

This technical report has been reviewed and approved.

H. W. Basham
Chief, Control Elements Branch
Flight Control Division
AF Flight Dynamics Laboratory

ABSTRACT

The state of the art of computer development has reached the point where it is now possible to sacrifice some of the extremely high speed of arithmetic computations in order to relieve man of tedium in another area - that of algebraic computation. This report presents the results of a development which uses an IBM 7090/94 computer to derive in nonnumeric terms the equations of motion of a system. This system can be composed of any number of masses which are acted upon by forces due to springs, dampers, gravity, and external forces.

CONTENTS

Section		Page
1	INTRODUCTION	1
2	DISCUSSION	3
	2.1 General Concept	3
	2.2 Lagrange's Equations for a Moving Coordinate System	5
3	COMPUTER PROGRAM	13
	3.1 General Form of the Input	13
	3.2 Input Cards for Computer Control	13
	3.3 Computer Output	21
4	REFERENCES	25

Appendix		
I	ILLUSTRATIVE COMPUTER RUNS	27
II	COMPUTER PROGRAM LISTING	69
III	DERIVATION OF THE GRAVITATIONAL FORCING FUNCTION BETWEEN TWO ARBITRARILY SHAPED BODIES	119

SECTION 1

INTRODUCTION

Digital computers have been designed basically to relieve the tedium of repeated manual arithmetic operations. However, with the extensive logic capabilities of the newer generations of computers, together with their extremely high speeds, it is possible to program these machines to perform some rudimentary algebraic operations. This use of these machines for algebraic operations has been illustrated in References 1 and 5.

This report presents the results of programming an IBM 7090 computer to find the generalized equations of motion of a system. The work is a generalization of that presented in Reference 1. In Reference 1 the equations were limited to the kinetic-energy terms of Lagrange's equation, the only variables being generalized coordinates. The program discussed in this report will consider both holonomic and nonholonomic variables, and it will consider the terms of the equations due to certain forces (springs, dashpots and gravity). The general form of the inputs and the basic philosophy of use between the initial work and this follow-on effort have remained the same in order to cause as little confusion as possible. This report is complete in itself so that prior knowledge of the subject is not necessary; however, because it is complete, one may notice many repetitions between the two reports.

The philosophy behind the writing of this computer program is: 1) the engineer can spend more time thinking about a problem instead of performing the completely mechanical operation of applying Lagrange's (or Kirchoff's) equations, 2) the equations for all feasible coordinate systems can be derived prior to making a choice of which would be the most applicable for a particular task, and 3) no assumptions are made about the relative magnitude of terms and the effect of different variables upon the solution until the complete equations are derived, at which time the engineer may choose to retain or delete any term from an equation.

SECTION 2

DISCUSSION

The computer program developed to derive the equations of motion is written in FORTRAN IV. The program is large, and as such it can only be used on one of the larger computers such as the IBM 7090/7094, which was the machine available to the author.

The mathematics used in this report is basically that of Cartesian tensors. Thus the metric tensor $g_{i_j} = g^{ij} = \delta_i^j$, means that the contravariant vector X^i and the covariant vector X_i are the same length.

The tensor summation convention is used throughout. This convention can be briefly explained by saying that if the same index appears twice in one term of an equation, the term involving this duplication is actually the sum of all values which the index may assume. For example

$$R_{i_j} X_j = \sum_{j=1}^{3} R_{i_j} X_j = R_{i_1} X_1 + R_{i_2} X_2 + R_{i_3} X_3.$$

If an index is not a repeated index, then it must also appear in each term of the equation.

An excellent explanation of Cartesian tensors is found in Reference 6.

2.1 GENERAL CONCEPT. The position of a point in Euclidian space with respect to an inertial coordinate system may be represented by the three components X_1, X_2, and X_3. If the inertial components are represented by X_i, then these three components may be represented as X_i.

If the point being represented is in a body B with the coordinates B_1, B_2, and B_3, and the reference system of the body coincides with the basic reference system, then the point in the body measured in the basic or inertial system, X, will be exactly that of body system B, and the three equations for the point may be written

$$X_i = B_i.$$

There are now two operations that may be performed on body reference system B that will cause it to differ from the inertial reference system, X. These are rotation and translation.

A translation is merely a vector that is added componentially. For example, assume that the reference point in body B is translated a distance T as measured in the inertial system; then every point in the body is also translated this distance, or

$$X_i = T_i + B_i,$$

which expresses in inertial coordinates the position of any point in the body.

A rotation merely changes portions of one coordinate into those of another.

For example, a rotation of angle α about the X_1 inertial axis would be

$$R_{ij} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{vmatrix}$$

and the position in inertial space of the components of the body would be

$$X_i = R_{ij}\,B_j$$

The transformation of a point in a body into inertial coordinates may be carried out with as many transformations or rotations as desired. For example, if the body were initially translated a distance T_i from the center of the inertial system and then the whole system rotated about the X inertial axis, this could be written as

$$X_i = R_{ij}\,[T_j + B_j]$$

Since X_i defines the position of a point, the velocity of this point is $\dot{X}_i$, which equals

$$\dot{X}_i = \frac{dX_i}{dt} = \sum_n \frac{\partial X_i}{\partial q_n}\,\frac{dq_n}{dt} = \frac{\partial X_i}{\partial q_n}\,\dot{q}_n$$

where the q_i's are the generalized coordinates of the point.

When the position of the point is expressible as generalized coordinates, the kinetic energy of the point is equal to

$$KE = \frac{m}{2}\,\dot{X}_i\dot{X}_i = \frac{m}{2}\,\frac{\partial X_i}{\partial q_j}\,\frac{\partial X_i}{\partial q_k}\,\dot{q}_j\,\dot{q}_k$$

where m is the mass of the point.

Knowing this, the equations of motion may be derived by applying Lagrange's equation

$$\frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{q}_i}\right) - \frac{\partial KE}{\partial q_i} = - \frac{\partial PE}{\partial q_i} = Q_i$$

where PE is the potential energy of the point and Q_i is the generalized force term.

This is the point of departure between the computer program outlined in this report and that of Reference 1. In Reference 1, only generalized coordinates were used and the calculation of generalized force term Q_i was left to the engineer.

2.2 LAGRANGE'S EQUATION FOR A MOVING COORDINATE SYSTEM. When the position of a point, a is referenced to a coordinate system that is not a fixed inertial system, the kinetic energy of the point may be expressed as

$$T_{(a)} = \frac{1}{2} m_{(a)} \left(U_i + \epsilon_{ijk}\,\omega_j\, X_{k_{(a)}} + \dot{X}_{i_{(a)}}\right)\left(U_i + \epsilon_{ilm}\,\omega_l\, X_{m_{(a)}} + \dot{X}_{i_{(a)}}\right)$$

where: $m_{(a)}$ is the mass at point a.

U_i is the velocity of the moving coordinate system measured along the i^{th} axis.

ω_i is the angular velocity of the moving coordinate system measured about the i^{th} axis.

$X_{i_{(a)}}$ is the displacement of the mass, $m_{(a)}$, as measured in generalized coordinates in the moving reference system.

ϵ_{ijk} is a skew symmetric tensor which has the component values

$$= \frac{(k - i)\ (k - j)\ (j - i)}{2}$$

There are now several sets of variables for which equations must be written, namely the U_i's, ω_i's, and q_i's. To use the simplest equation forms possible, three sets of equations are used. The derivation of these equations has been attributed to Kirchhoff and may be found in vector form on Page 503 of Reference 2. The three equations are

$$\frac{d}{dt}\left(\frac{\partial T}{\partial U_i}\right) + \epsilon_{ijk}\,\omega_j \frac{\partial T}{\partial U_k} = F_i$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \omega_i}\right) + \epsilon_{ijk}\,\omega_j \frac{\partial T}{\partial \omega_k} + \epsilon_{ijk}\, U_j \frac{\partial T}{\partial U_k} = M_i$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} = Q_i$$

where: $T = \sum_{\alpha} T_{(\alpha)}$ the total kinetic energy of a system.

F_i is the linear accelerating force acting on the moving coordinate system as measured along the moving axis.

M_i is the torque exerted on the moving coordinate system as measured about the moving axes.

Q_i is the generalized force associated with q_i.

2.2.1 Expansion of the Kinetic Energy Terms. To use these forms of Lagrange's equations in the computer, the equations were expanded as follows:

$$\begin{aligned} F_i = {} & \dot{U}_i - \dot{\omega}_j\, \epsilon_{jik}\, X_k - 2\,\omega_j \epsilon_{jik} \frac{\partial X_k}{\partial q_m}\, \dot{q}_m \\ & + \frac{\partial^2 X_i}{\partial q_m\, q_n}\, \dot{q}_m\, \dot{q}_n + \frac{\partial X_i}{\partial q_m}\, \ddot{q}_m + U_k\, \epsilon_{kij}\, \omega_j \\ & - \omega_j\, \epsilon_{jik}\, \epsilon_{klm}\, X_m\, \omega_l . \end{aligned} \tag{1}$$

$$\begin{aligned} M_i = {} & \dot{U}_j\, \epsilon_{jik}\, X_k + X_l\, \epsilon_{lim}\, \epsilon_{mkj}\, \omega_j\, U_k - X_k\, \epsilon_{kij}\, \epsilon_{jlm}\, X_m\, \dot{\omega}_l \\ & - X_k\, \epsilon_{kij} \frac{d^2 X_i}{\partial q_m\, q_n}\, \dot{q}_m\, \dot{q}_n - X_k\, \epsilon_{kij} \frac{\partial X_i}{\partial q_m}\, \ddot{q}_m \\ & - X_m\, \epsilon_{mkn}\, \epsilon_{nlp}\, X_p\, \omega_l\, \epsilon_{ijk}\, \omega_j - 2X_k\, \epsilon_{kil}\, \epsilon_{ljn} \frac{\partial X_n}{\partial q_m}\, q_m\, \omega_j . \end{aligned} \tag{2}$$

$$Q_m = \frac{\partial X_i}{\partial q_m} \frac{\partial^2 X_i}{\partial q_k \partial q_n} \dot{q}_k \dot{q}_n + \frac{\partial X_i}{\partial q_m} \frac{\partial X_i}{\partial q_k} \ddot{q}_k + \frac{\partial X_i}{\partial q_m} \epsilon_{ijk} X_k \dot{\omega}_j$$

$$+ \frac{\partial X_i}{\partial q_m} \dot{U}_i + \frac{\partial X_i}{\partial q_m} \epsilon_{ijk} U_k \omega_j + \frac{\partial X_i}{\partial q_m} \epsilon_{ijk} \epsilon_{kln} X_n \omega_j \omega_l \qquad (3)$$

$$+ 2 \frac{\partial X_i}{\partial q_m} \epsilon_{ijk} \frac{\partial X_k}{\partial q_n} \dot{q}_n \omega_j .$$

When the position of the origin of the moving axis system is expressed in terms of generalized coordinates in the inertial system it is possible to change the form of the three equations in the following manner.

Let $\dot{Y}_i = D_{ij} U_j$

where Y_i is the position of the origin of the moving axis system

and D_{ij} is the set of direction cosines defining the orientation of the moving axes system.

Then

$$U_n = D_{in} \dot{Y}_i$$

and

$$\dot{U}_n = D_{ik} \epsilon_{kjn} \omega_j \dot{Y}_i + D_{in} \ddot{Y}_i$$

The two terms in Equation 1,

$$\dot{U}_i + U_k \epsilon_{kij} \omega_j$$

reduce to

$$\dot{U}_i + U_k \epsilon_{kij} \omega_j = \ddot{Y}_n D_{ni} = \frac{\partial^2 Y_n}{\partial y_i \partial y_k} D_{ni} \dot{y}_i \dot{y}_k + \frac{\partial Y_n}{\partial y_i} D_{ni} \ddot{y}_i$$

where the y_i's are the generalized coordinates of the origin of the moving axes.

Using the above relationships in the two terms of Equation 2,

$$\dot{U}_j \epsilon_{jik} X_k - U_k \epsilon_{kjm} \epsilon_{mil} X_l \omega_j ,$$

gives

$$\ddot{Y}_l D_{lj} \epsilon_{jil} X_k = \frac{\partial^2 Y_l}{\partial y_n \partial y_m} D_{lj} \epsilon_{jik} X_k \dot{y}_n \dot{y}_m + \frac{\partial Y_l}{\partial y_n} D_{lj} \epsilon_{jik} X_k \ddot{y}_n$$

Using the same definitions for the origin, the terms in Equation 3 reduce as follows:

$$\frac{\partial X_i}{\partial q_m} \dot{U}_i + \frac{\partial X_i}{\partial q_m} \epsilon_{ijk} U_k \omega_j = \frac{\partial X_i}{\partial q_m} D_{ni} \ddot{Y}_n$$

$$= \frac{\partial^2 Y_n}{\partial y_j \partial y_k} D_{ni} \frac{\partial X_i}{\partial q_m} \dot{y}_j \dot{y}_k + \frac{\partial Y_n}{\partial y_j} D_{ni} \frac{\partial X_i}{\partial q_m} \ddot{y}_j$$

There was no similar reduction of the equations for the case when the rotational orientation, D_{ij}, could be expressed as generalized coordinates, because when this occurs, the whole system can be expressed as generalized coordinates and provisions for this condition had already been made.

2.2.2 Expansion of the Force and Moment Terms. To include the terms F_i, M_i, and Q_i, it was necessary to expand the definitions to fit the computer program. The forces considered in the program are linear spring force, rotary spring force, linear viscous force, rotary viscous force, gravitational force, and an arbitrary external force. Any or all of these force terms may be included in any system, and, whereas the number of independent variables is limited on the computer, no limit is placed on the allowable forces in a system.

2.2.2.1 Calculation of F_i (for Equation 1). The general force term of Equation 1 is equal to the sum of all the forces acting on the system. This is written

$$F_i = \sum_{\beta} F_{(\beta)i}$$

where $F_{(\beta)i}$ is the i^{th} component of the β^{th} force.

For an arbitrary force whose components in the moving system are G_i, the $F_{(\beta)i}$ associated with these is simply

$$F_{(\beta)i} = G_i .$$

Such a force could arise from rocket engines, aerodynamic effects, or some other external source.

If there is a linear spring whose ends are connected between points R_i and S_i and whose spring constant is K_s, then

$$F_{(\beta)i} = K_s (R_i - S_i)$$

where the R_i and S_i positions are measured in the moving coordinate system.

Care must be taken when deriving the equations by hand (the computer takes it into account) that the $F_{(\beta)i}$'s due to a linear spring are acting at both end points. If both end points R_i and S_i are internal to the moving system, then the forces at both ends of the spring must be considered and the net reaction to the moving system itself is zero, leaving only the effects to the internal motions in the system.

If there is a gravitational attraction from a point R_i in the moving system and a point S_i external to the moving system, then

$$F_{(\beta)i} = \frac{K_g}{L^3} (R_i - S_i)$$

where K_g is the gravitational constant between the two points and L is the scalar distance as defined by

$$L = \sqrt{(R_i - S_i)(R_i - S_i)}$$

The above relationship for gravitational forces is merely the inverse square attraction, and would therefore be applicable to electrostatic and magnetic force fields also.

For the case of a linear damper connected between points R_i and S_i whose force exerted upon the two ends is equal to $B_L \dot{L}$, then the components $F_{(\beta)i}$ are

$$F_{(\beta)i} = \frac{B_L}{L^2} (R_i - S_i)(R_j - S_j) \frac{\partial (R_j - S_j)}{q_n} \dot{q}_n$$

$$= \frac{B_L}{L} (R_i - S_i) \frac{\partial L}{\partial q_n} \dot{q}_n$$

For both the rotary spring and the rotary damper, no $F_{(\beta)i}$ terms will be present.

2.2.2.2 <u>Calculation of M_i (for Equation 2).</u> The general moment term of Equation 2 is equal to the sum of all the moments acting upon the system,

$$M_i = \sum_{\beta} M_{(\beta)i}$$

where $M_{(\beta)i}$ is the i^{th} component of the β^{th} force.

Again using X_i to represent the moving axes coordinates of a point in the system, the moment due to a force whose components in the moving system are F_j would be

$$M_{(\beta)i} = F_j \epsilon_{jik} X_k$$

If the components of the force are given in the inertial system as f_j then

$$F_j = D_{nj} f_n$$

and

$$M_{(\beta)i} = D_{nj} f_n \epsilon_{jik} X_k$$

For a linear spring whose ends are connected at points R_i and S_i (as measured in inertial coordinates), the force exerted by the spring is

$$F = K_s L = K_s \sqrt{(R_i - S_i)(R_i - S_i)}$$

If the end of the spring which is attached to the moving coordinate system is at point R, then the position of R may be expressed as

$$R_i = D_{ij} X_j + Z_i$$

where:

D_{ij} is the direction cosine orientation of the moving system.

X_j is the attachment point of the spring.

Z_i is the position in inertial coordinates of the origin of the moving system.

The moment due to this linear spring is therefore

$$M_{(\beta)i} = K_s D_{nj} (Z_n - S_n) \epsilon_{jik} X_k$$

For gravitational attraction between point R_i in the moving system and point S_i, external to the system, then

$$M_{(\beta)j} = \frac{K_g}{L^3} D_{ni} (Z_n - S_n) \epsilon_{ijk} X_k$$

where K_g is the gravitational coefficient.

For a linear damper whose force is

$$F = B_L \dot{L}$$

then

$$M_{(\beta)j} = \frac{B_L}{L^2} D_{ni} (Z_n - S_n) \epsilon_{ijk} X_k (R_m - S_m) \frac{\partial (R_m - S_m)}{\partial q_r} \dot{q}_r$$

$$= \frac{B_L}{L} D_{ni} (Z_n - S_n) \epsilon_{ijk} X_k \frac{\partial L}{\partial q_r} \dot{q}_r$$

Again, no terms exist for a rotary spring or a rotary damper.

2.2.2.3 Calculation of Q_i (for Equation 3). The generalized force, Q_i, is expressed as the sum of all the partial forces

$$Q_i = \sum_{\beta} Q_{(\beta)i}$$

If there is a force with components F_i expressed in the moving coordinate system acting at point X_i, then the generalized force term for this force is

$$Q_{(\beta)m} = F_i \frac{\partial X_i}{\partial q_m}$$

If the force is due to a linear spring between points R_i and S_i, then the generalized force due to the spring is

$$Q_{(\beta)m} = K_s (R_i - S_i) \frac{\partial(R_i - S_i)}{\partial q_m} = K_s L \frac{\partial L}{\partial q_m}$$

For a linear damper connected physically between R_i and S_i whose force is $F = B_L \dot{L}$, the generalized force is

$$Q_{(\beta)m} = \frac{B_L}{L^2} (R_i - S_i)(R_j - S_j) \frac{\partial(R_i - S_i)}{\partial q_m} \frac{\partial(R_i - S_i)}{\partial q_m} \dot{q}_n$$

$$= B_L \frac{\partial L}{\partial q_m} \frac{\partial L}{\partial q_n} \dot{q}_n$$

For a gravitational force between R_i and S_i,

$$Q_{(\beta)m} = \frac{K_g}{L^3} (R_i - S_i) \frac{\partial(R_i - S_i)}{\partial q_m} = \frac{K_g}{L^2} \frac{\partial L}{\partial q_m} = - K_g \frac{\partial(1/L)}{\partial q_m}$$

An alternative expansion of this term is explained in Appendix III.

For a rotary spring and a rotary damper, the generalized forces will be simply

$$Q_{(\beta)m} = K_r q_m$$

and $$Q_{(\beta)m} = B_r \dot{q}_m$$

where the q_m and $\dot{q}_m$ may be only the generalized rotation angle and angular rate about the axis about which the rotary spring and damper are mounted. K_r and B_r are the spring and damper constants.

SECTION 3

COMPUTER PROGRAM

3.1 GENERAL FORM OF THE INPUT. The basic concept of controlling the computer requires that two basic pieces of information be supplied to it. These are a) the variables that exist in the system and b) the relationships that exist between the variables. The format of providing this information to the computer is essentially the same as was developed in Reference 1.

3.2 INPUT CARDS FOR COMPUTER CONTROL. There are five basic card types for computer inputs. These are a) comment cards, b) mass cards (kinetic energy), c) title cards for masses, d) force cards, and e) title cards for forces. Each of these types will be described separately.

3.2.1 Comment Cards. These cards may be placed only at the start of each desired set of equations. Their primary function is to furnish operator written comments that will remain with the final system output, comments such as the title of the problem and verbal descriptions of the variables and their functions. There may be as many comments preceding a program as necessary to identify the problem.

The computer program utilizes the comment cards to separate systems. For this reason, the program requires a comment card following the last system-of-equation cards. For convenience, the comment on this card is usually "End of Test" or a similar statement.

Comment cards are identified by either * or C in card column 1. Both marks are recognized because the C is used in FORTRAN comments and the * is used in comments in some machine languages. There may be any recognized symbol or blank in columns 12 through 80. The * or C must be in column 1 of each comment card.

3.2.2 Input Cards for Masses. Since kinetic energy can only be associated with masses, the portion of Lagrange's equations pertaining to the kinetic energy of a body (or a point mass) is calculated from these cards. The computer requires one control card for each mass. A mass control card is recognized by having an M in column 1.

The code used for inserting the necessary information into the computer basically represents the position of a point in the inertial coordinate system. This position can be expressed as

$$P_i = R_{1_{ij}} \left[T_{1_j} + R_{2_{jk}} \left[T_{2_k} + \dots + R_{8_{mn}} \left(T_{8_n} \right) \dots \right] \right]$$

where R_{ijk} represents the i^{th} rotation matrix and T_{ij} represents the i^{th} translation vector. It is possible that any or all of the translations or rotations will be the dependent variables of the equations of motion. To put this position information into the computer, the mass control card represents the position P_i by a repeated seven-column coding system. The sequence of this repeated system is Blank, Rotation, Blank, Translation 1, Translation 2, Translation 3, Blank, with rotation as a single code letter to represent the i^{th} square rotation matrix R_{ijk}, and the three translations are three code letters to represent the three terms of the i^{th} translation T_{ij}.

The mass control cards are set up as follows.

M A S S	R T T T	R T T T	R T T T	. . .
T I T L E	R O T I T	R O T I T	R O T I T	. . .
I X Y C O D	T R T I T X	T R T I T X	T R T I T X	. . .
I X Z C O D	T R T I T Y	T R T I T Y	T R T I T Y	. . .
I Y Z C O D	T R T I T Z	T R T I T Z	T R T I T Z	. . .

where

R are the rotation code letters
T are the translation code letters
T I T L E is the name given to this mass
R O T I T is the title given to the rotation directly above it
T R T I T X is the title given to the X translation component directly above it
T R T I T Y is the title given to the Y translation component directly above it
T R T I T Z is the title given to the Z translation component directly above it
I X Y C O D is the code to tell whether IXY exists or is equal to zero
I X Z C O D is the code to tell whether IXZ exists or is equal to zero
I Y Z C O D is the code to tell whether IYZ exists or is equal to zero

The rotational code letters that are permissible and their meanings are

X A variable rotation about the X axis.

Y A variable rotation about the Y axis.

Z A variable rotation about the Z axis.

I A constant rotation about the X axis.

J A constant rotation about the Y axis.

K A constant rotation about the Z axis.

D A variable direction-cosine orientational rotation.

C A constant direction-cosine orientational rotation.

U Unit Matrix, no rotation.

P Used only as the last letter in a control line. If it appears, the inertial coordinates of the body are printed out.

The translational code letters and their meanings are

A A constant distance (for integration in a body) measured from the center of mass of the body.

C A constant distance.

O Zero, no translation. (A blank is also accepted for no translation.)

U A nonholonomic variable (one whose distance is not known -- only its velocity) used only in conjunction with direction-cosine rotation for coordinates in a moving reference system.

V A holonomic variable.

All of the code letters that represent variables will indicate to the program that these functions are the dependent variables in the set of differential equations (time is always the independent variable), and as such an equation will be derived for each dependent variable.

Any constants in the coded mass word are for the purpose of allowing more freedom to the engineer who is setting up the system. The constants may represent a constant distance, such as the distance from the center of gravity of an airplane to the hinge line of the elevators, or a constant orientation such as the inclination of an orbital plane with respect to the equatorial plane. Any distance or rotational orientation that is not a function of the time should be represented by the correct constant code letter in lieu of the code variable letter.

Of special interest is the code letter A in the translation codes. This is actually a constant distance; however, it has the added characteristic that it can never appear with any other distance -- either variable or constant. This was programmed in this fashion to allow the computer to perform the integration over a complete body. It was known that the total kinetic energy of a body A is equal to the integral, over the complete body, of the kinetic energy of all points α in the body, or

$$T = \frac{1}{2} \int_A \left(U_i + \epsilon_{ijk}\, \omega_j\, X_{k_{(\alpha)}} + \dot{X}_{i_{(\alpha)}} \right) \left(U_i + \epsilon_{iln}\, \omega_l\, X_{n_{(\alpha)}} + \dot{X}_{i_{(\alpha)}} \right) dm_{(\alpha)}$$

If the generalized position of a point α is expanded to

$$X_{i_{(\alpha)}} = Y_i + S_{ij}\, Z_{j_{(\alpha)}}$$

where Y_i is the generalized distance to the center of mass of the body, $Z_{i(\alpha)}$'s are the body coordinates of the point (α), and S_{ij} is the generalized orientation of the body, then

$$T = \frac{1}{2} \int_A \left(U_i + \epsilon_{ijk} \omega_j Y_k + \epsilon_{ijk} \omega_j S_{kp} Z_{p_{(\alpha)}} + \dot{Y}_i + S_{ij} Z_{j_{(\alpha)}} \right) \left(U_i + \epsilon_{iln} \omega_l Y_n + \epsilon_{iln} \omega_l S_{nr} Z_{r_{(\alpha)}} + \dot{Y}_i + \dot{S}_{il} Z_{l_{(\alpha)}} \right) dm_{(\alpha)}$$

$$= \frac{1}{2} \int (U_i + \epsilon_{ijk} \omega_j Y_k + Y_i)(U_i + \epsilon_{iln} \omega_l Y_n + \dot{Y}_n)\, dm_{(\alpha)}$$

$$+ \frac{1}{2} \int \left(\epsilon_{ijk} \omega_j S_{kp} Z_{p_{(\alpha)}} + \dot{S}_{ip} Z_p (\alpha) \right) \left(\epsilon_{iln} \omega_l S_{nr} Z_{r_{(\alpha)}} + \dot{S}_{ir} Z_r (\alpha) \right) dm_{(\alpha)}$$

$$+ \int (U_i + \epsilon_{ijk} \omega_j Y_k + \dot{Y}_i) \left(\epsilon_{iln} \omega_l S_{nr} Z_{r_{(\alpha)}} + \dot{S}_{ir} Z_{r_{(\alpha)}} \right) dm_{(\alpha)}$$

$$= \frac{M_A}{2} (U_i + \epsilon_{ijk} \omega_j Y_k + \dot{Y}_i)(U_i + \epsilon_{iln} \omega_l Y_n + \dot{Y}_i)$$

$$+ \frac{1}{2} (\epsilon_{ijk} \omega_j S_{kp} + \dot{S}_{ip})(\epsilon_{iln} \omega_l S_{nr} + \dot{S}_{ir}) \int_A Z_{p_{(\alpha)}} Z_{r_{(\alpha)}}\, dm_{(\alpha)}$$

where M_A is the total mass of body A.

The third term in the above equation is zero because

$$\int_A Z_{r_{(\alpha)}}\, dm_\alpha = 0$$

by definition of the center of mass.

The computer is programmed to perform the remaining integration

$$\int_A Z_{p_{(\alpha)}} Z_{r_{(\alpha)}}\, dm_\alpha$$

by recognizing that for the case of $p \neq r$, then

$$\int_A Z_{p_{(\alpha)}} Z_{r_{(\alpha)}} dm_{(\alpha)} = I_{pr} = I_{rp}$$

which is the cross product of the inertial term.

For the case of $p = r$, then

$$\int_A Z_{p_{(\alpha)}} Z_{p_{(\alpha)}} dm_{(\alpha)} = - I_p + \sum_{i=1}^{3} \frac{I_i}{2}$$

where I_i is the moment of inertia about the i^{th} body axis.

3.2.3 Title Cards for Masses. To give recognizable names to the constants and variables, provisions have been made for up to four cards of titles following the mass control card. These title cards are recognized by a blank in column 1.

The first card following the mass control card will assign these titles:

1. In columns 2 through 7, the title of the particular mass described by the control card. Such a title may be any combination of letters, numbers, or symbols, such as: AIRP, BODY-1, A + B, POINTM, or 1.

2. In columns 9 through 14, the title of the first rotational variable. For example, if the rotation code is a Z and the title given is ALPHA, this means that the rotation is an angle ALPHA about the Z axis. Again, any combination of six letters, numbers, or symbols may be given.

3. In each of the six columns starting with column 16, 23, 30, 37, ... 65 will be the titles of rotations 2, 3, ... 8, in the same manner as described for the first rotation in item 2 above.

The second, third, and fourth cards following the mass control card give titles to the first, second, and third translations in each of the six columns starting in columns 9, 16, ... 65. There are two exceptions to this:

1. If the starting column corresponds to a direction cosine variable, (which requires that the translation codes be zeros), then the second, third, and fourth cards give respectively the rotational velocities about the X, Y, and Z axes of the moving coordinate system. If no title is given, the particular velocity is assumed to be restrained to zero.

2. If the translations are U's, corresponding to a velocity along the moving axis coordinates, then the titles given will be the velocity comp nents instead of the position components.

In columns 2 through 7 of the second, third, and fourth title cards is a command to tell the computer whether the I_{XY}, I_{XZ}, and I_{YZ} values of the mass considered are zero. Any nonblank character (except the number 0) in one of these six columns will indicate to the computer that the particular cross product of inertia term exists and is nonzero. A zero in one of these columns will say that the corresponding inertia term is zero.

For the sake of compressing the output, all imbedded blanks in the titles are removed.

If a title is not given but is required in order to print the output, the computer will arbitrarily assign a title. The rotation titles given are R"N" where N will be the number of the rotation when counting from left to right. The translation titles given are V"RC" for a variable at the row-column number "RC" or C"RC" for a constant at the variable "RC."

3.2.4 Input and Title Cards for Force Control. One control card is required for each force acting upon the system, and a force control card is distinguished by having the letter F in column 1.

There are six control titles, one for each type of a force, which describe the force and determine which equation type to use. These titles must start in column 2 (with an F in column 1) on the force control card and are:

LINSPR	for a linear spring
GRAV	for gravity
ROTSPR	for a rotary spring
EXT	for an external force
DLIN	for a linear damper
DROT	for a rotary damper

Four title cards, (no more, no less) are required for all types except ROTSPR and DROT. The title cards for these, again, have a blank in column 1.

These control and title cards will be considered separately.

3.2.4.1 Linear Spring Control Cards. This force card assumes that there is a spring connected between points R_i and S_i in the system. The cards for a linear spring will be set up as follows:

FLINSPR	R TTT	R TTT	R TTT	R TTT	R TTT	R TTT
title	ro tit	ro tit	ro tit	ro tit	ro tit	ro tit
A, B	tr tit	tr tit	tr tit	tr tit	tr tit	tr tit
C, D	tr tit	tr tit	tr tit	tr tit	tr tit	tr tit
E, F	tr tit	tr tit	tr tit	tr tit	tr tit	tr tit

where R - a rotation code.

T - a translation code.

title - the title of the spring constant.

ro tit - the rotation title of the rotation directly above it in the control card.

tr tit - the translation title of the corresponding translation in the control card.

A - a one-digit number indicating the first group in the control word common to both R_i and S_i.

B - a one-digit number indicating the last group in the control word common to both R_i and S_i.

C - a one-digit number corresponding to the first group in the control word used only by R_i.

D - a one-digit number corresponding to the last group in the control word used only by R_i.

E - a one-digit number corresponding to the first group in the control word used only by S_i.

F - a one-digit number corresponding to the last group in the control word used only by S_i.

It must be noted that a comma is required to separate the letters A and B, C and D, and E and F although the two integers and the comma may be placed anywhere in the six-column field with the remaining spaces being blanks. Also, the rotational and translational variables must have the names of the variables as defined for at least one of the mass control sets.

This force control set is an abbreviation of the two positions, R_i and S_i. For example, assume that the position of one end of a spring (whose spring constant is KO) is

$$R_i = A_{ij}\left[B_j + C_{jk}\left[D_k + E_{kl}\left[F_l\right]\right]\right] ,$$

while the other end is at

$$S_i = A_{ij} \left[B_j + G_{jk} \left[H_k \right] \right] .$$

Where A_{ij} is a rotation alpha about the X axis.

B_j is a translation along Y the distance R.

C_{jk} is a rotation beta about the Z axis.

D_k is a translation along Z the distance X1.

E_{kl} is a rotation delta about the X axis.

F_l is the attach point of one end.

G_{jk} is the rotation gamma about the Y axis.

H_k is the attach point of the other end.

The five control cards for this would then be:

FLINSPR	X OVO	Z OOV	X CCC	Y COO
KO	ALPHA	BETA	DELTA	GAMMA
1,1			R1	S1
2,3	R		R2	
4,4		X1	R3	

3.2.4.2 Gravitational Force Control Cards. The control and title cards for a gravitational force between two points R_i and S_i are identical to the set as shown for the linear spring, except for having FGRAV in columns 1 through 5 of the force control card. The relative positions of R_i and S_i are again abbreviated into one control card, and the four title cards are again required.

3.2.4.3 Rotary Spring. Only one title card is required in addition to the control card for a rotary spring force. These two cards are of the form

```
FROTSPR  R
 title    ro tit
```

where R is the rotational code for the variable about which a rotary spring exists

title is the name given to the spring coefficient, and

ro tit is the title of the rotational variable.

Any or all of the additional title cards may be inserted, but they will neither have any meaning to the program nor will they affect the results.

3.2.4.4 External Force Control Cards. If an external force is acting upon a system, it is assumed to have a direction and a point of application. The control card is abbreviated as for the linear spring, however, with minor variations to make it acceptable to this application. The point of application is assumed to be at the position called R_i for the linear spring, with the force components at the S_i position. In the title areas of the force components, the components of the force may be listed directly, or they may be some normalized components. The title area assigned to the spring constant in the linear spring illustration will for this case contain the normalizing factor (either numeric or algebraic), and if this area is left blank, a normalizing factor of one will be assumed and will not be printed.

As a simple example of this, assume a force with the components G1, G2, and G3 is acting at a point X1, X2, X3. The five control cards for this case are:

FEXT	U OOO	U CCC	U CCC
	BLANK CARD		
1, 1		X1	G1
2, 2		X2	G2
3, 3		X3	G3

3.2.4.5 Linear Damper Control Cards. This set of control cards will be the same as that for a linear spring (again with the exception of the control title on the force control card). The two ends of the damper are assumed to be at R_i and S_i. The control card is abbreviated identically, and the damper constant title is inserted in lieu of the spring constant title.

3.2.4.6 Rotary Damper Control Cards. As the linear damper uses exactly the same input format as the linear spring, the rotary damper input is the same as the rotary spring. Only the one rotation code and title are given along with the damper constant.

3.3 COMPUTER OUTPUT. In general, the computer output assumes the appearance of statements written in FORTRAN. There are some peculiarities in the system which will be described here.

3.3.1 Reprint of Input Cards. All cards are printed on the output pages exactly as they are punched with only one exception: for a comment card which uses a C to indicate that it is a comment, the C in column 1 is not printed.

There is a code checking subroutine in the program. If a code letter exists that is illegal, an extra line of output will be printed above the line in error. This extra line is recognized by ***ERROR*** on the right end of the line. Also, in this ERROR line will be one or more "V"s, each V being directly above a coding error. If a coding error is detected in any system of masses, the processing of that particular set of data is terminated with the code checking phase, in other words no equations will be found for this set.

3.3.2 Position Expansion. If a rotation code letter "P" is given for any masses in a set, the next output to be given will be the X, Y, and Z components of the input command card, expanded and expressed in terms of the input rotations and translations. There will be three equations for each P in the system.

3.3.3 Auxiliary Equations. To simplify the writing of some of the terms in the output of the equations, occasionally a set of auxiliary equations will be defined. The term on the left-hand side of an auxiliary equation is defined by the expression on the right-hand side. These equations are set up when it is anticipated they might be used; consequently, there are times when they are set up but not used.

3.3.4 Output Equations. Each equation is preceded by the statement, "This is the equation for the variable ." The variable is the generalized coordinate q_n, the velocity u_i, or the rotational rate ω_i described in Kirchoff's equations on page 5. One equation is written for every variable in a system.

As has been stated, the equations appear to be written in FORTRAN, with only one term on each line. The following, however, are specifically used in the output of this program and it is recommended that they not be used as titles in order to avoid confusion in the output:

D (var) is actually $\frac{d}{dt}$ (var)

DD (var) is actually $\frac{d^2}{dt^2}$ (var)

SINSIN (angle) is actually $\sin^2$ (angle)

COSCOS (angle) is actually $\cos^2$ (angle)

SINCOS (angle) is actually sin (angle)*cos (angle)

IX (mass title) is actually the moment of inertia about the X body axis of (mass title)

IY (mass title) is actually the moment of inertia about the Y body axis of (mass title)

IZ (mass title) is actually the moment of inertia about the Z body axis of (mass title)

IXY (mass title) is actually the cross product of inertia $\int XYdm$ of the body (mass title)

IXZ (mass title) is actually the cross product of inertia $\int XZdm$ of the body (mass title)

IYZ (mass title) is actually the cross product of inertia $\int YZdm$ of the body (mass title)

direction cosine title (i, j) is actually the i^{th} row, j^{th} column component of the direction cosine whose title is given.

SECTION 4

REFERENCES

1. D. H. Jirauch and R. A. Westerwick, A Use of Digital Computers for Algebraic Computation, U. S. Air Force Aeronautical Systems Division Report RTD-TDR-63-4200, December 1963,

2. L. M. Milne-Thompson, Theoretical Hydrodynamics, The Macmillan Co., N. Y., 1957.

3. A. J. McConnell, Applications of Tensor Analysis, Dover Publications, Inc., N. Y., 1957.

4. R. A. Frazer, W. J. Duncan, and A. R. Collar, Elementary Matrices and Some Applications to Dynamics and Differential Equations, Cambridge University Press, Cambridge, England, 1938.

5. FORmula MAnipulation Complier, Users' Reference Manual, International Business Machines Corp., 1954.

6. H. Jeffreys, Cartesian Tensors, Cambridge University Press, Cambridge, England, 1931.

APPENDIX I

ILLUSTRATIVE COMPUTER RUNS

This appendix contains 14 cases of computer runs made with this program. The first case is the simplest and the subsequent cases are progressively more complex.

```
*            CASE 1
*         THIS CASE IS A POINT MASS MOVING ONLY UP AND DOWN.  THE MASS IS
*      CONNECTED BY A SPRING-DAMPER TO POINT IG, AND FORCED DOWNWARDS BY
*      GRAVITY (CONSIDERED HERE AS A CONSTANT EXTERNAL FORCE EQUAL TO THE
*      WEIGHT OF THE POINT MASS).
*
*      Z IS THE DOWNWARD DISPLACEMENT OF THE POINT MASS.
*      K IS THE SPRING CONSTANT.
*      B IS THE DAMPING CONSTANT.
*
MASS    U 00V
 POINT
        BLANK
        BLANK
        Z
FEXT    U 00V   U 00C
        BLANK
 1,1
 1,1
 2,2      Z    WEIGHT
FLINSPR U 00V   U 00C
  K
 1,1
 1,1
 2,2      Z     ZO
FDLIN   U 00V   U 00C
  B
 1,1
 1,1
 2,2      Z      ZO


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K5D/FL=(B)/(+(Z)*(Z)-2*(Z)*(ZO)+(ZO)*(ZO))


THESE ARE THE EQUATIONS FOR THE VARIABLE Z

0=
+DD(Z)*MASS(POINT)
-(WEIGHT)
+(Z)*(K)
-(ZO)*(K)
+(Z)*(Z)*D(Z)*(K5D/FL)
-(Z)*(ZO)*D(Z)*(K5D/FL)
-(ZO)*(Z)*D(Z)*(K5D/FL)
+(ZO)*(ZO)*D(Z)*(K5D/FL)
```

```
*           CASE 2
*        THIS IS A POINT MASS FREE TO MOVE IN THE Y-Z PLANE.  AGAIN THE
*     MASS IS RESTRAINED BY A SPRING DAMPER ATTACHED TO POINT (0,ZO).
*
*     Z IS THE DOWNWARD DISPLACEMENT.
*     Y IS THE HORIZONTAL DISPLACEMENT.
*     K1 IS THE SPRING CONSTANT.
*     B1 IS THE DAMPING COEFFICIENT


MASS     U OVV
         BLANK
         BLANK
          Y
          Z
FLINSPR U OVV   U OCC
   K1
 1,1
 1,1     Y
 2,2     Z         ZO
FDLIN    U OVV  U OOC
  B1
  1,1
  1,1     Y
  2,2      Z       ZO
FEXT     U OVV  U OCC
         BLANK
  1,1
  1,1       Y
 2,2       Z     WEIGHT


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K5C/FL=(B1)/(+(Y)*(Y)+(Z)*(Z)-2*(Z)*(ZO)+(ZO)*(ZO))


THESE ARE THE EQUATIONS FOR THE VARIABLE Y

0=
+DD(Y)*MASS
+(Y)*(K1)
+(Y)*(Y)*D(Y)*(K5C/FL)
+(Y)*(Z)*D(Z)*(K5C/FL)
-(Y)*(ZO)*D(Z)*(K5C/FL)


THESE ARE THE EQUATIONS FOR THE VARIABLE Z

0=
+DD(Z)*MASS
+(Z)*(K1)
-(ZO)*(K1)
+(Z)*(Y)*D(Y)*(K5C/FL)
```

```
+(Z)*(Z)*D(Z)*(K5C/FL)
-(Z)*(ZO)*D(Z)*(K5C/FL)
-(ZO)*(Y)*D(Y)*(K5C/FL)
-(ZO)*(Z)*D(Z)*(K5C/FL)
+(ZO)*(ZO)*D(Z)*(K5C/FL)
-(WEIGHT)
```

```
*             CASE 3

         THIS IS A POINT MASS FREE TO MOVE IN ALL THREE EUCLIDIAN DIRECTIONS,
      AND RESTRAINED BY THREE SPRING DAMPERS.
      K1 AND B1 ARE THE SPRING-DAMPER COEFFICIENTS CONNECTED TO (X1,Y1,Z1).
      K2 AND B2 ARE THE SPRING-DAMPER COEFFICIENTS CONNECTED TO (X2,Y2,Z2).
      K3 AND B3 ARE THE SPRING-DAMPER COEFFICIENTS CONNECTED TO (X3,Y3,Z3).
      X, Y, AND Z ARE THE EUCLIDIAN COORDINATES OF THE MASS.
      THE FORCE OF GRAVITY PUSHES THE MASS IN THE Z DIRECTION WITH A FORCE
         PROPORTIONAL TO WEIGHT.


MASS     U VVV   P
 POINT
          X
          Y
          Z
FEXT     U VVV   U OOC
                 BLANK
 1,1       X
 1,1      Y
 2,2     Z       WEIGHT
FLINSPR U VVV   U CCC
 K1
 1,1      X       X1
 1,1      Y         Y1
 2,2      Z         Z1
FLINSPR U VVV   U CCC
  K2
 1,1      X       X2
 1,1      Y       Y2
 2,2       Z      Z2
FLINSPR U VVV   U CCC
  K3
 1,1      X       X3
 1,1      Y       Y3
 2,2      Z       Z3
FDLIN    U VVV   U CCC
  B1
 1,1      X       X1
  1,1     Y       Y1
  2,2     Z       Z1
FDLIN    U VVV   U CCC
  B2
   1,1     X      X2
   1,1     Y      Y2
   2,2     Z      Z2
FDLIN    U VVV   U CCC
  B3
 1,1        X     X3
 1,1        Y     Y3
 2,2        Z     Z3


THIS IS THE X POSITION EXPANSION FOR THE MASS POINT

+(X)
```

```
THIS IS THE Y POSITION EXPANSION FOR THE MASS POINT

+(Y)

THIS IS THE Z POSITION EXPANSION FOR THE MASS POINT

+(Z)


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K5F/FL=(B1)/(+(X)*(X)-2*(X)*(X1)+(Y)*(Y)-2*(Y)*(Y1)+(Z)*(Z)-2*(Z)*(Z1)+(X1)*(X1)+(Y1)*(Y1)
        +(Z1)*(Z1))

K5G/FL=(B2)/(+(X)*(X)-2*(X)*(X2)+(Y)*(Y)-2*(Y)*(Y2)+(Z)*(Z)-2*(Z)*(Z2)+(X2)*(X2)+(Y2)*(Y2)
        +(Z2)*(Z2))

K5H/FL=(B3)/(+(X)*(X)-2*(X)*(X3)+(Y)*(Y)-2*(Y)*(Y3)+(Z)*(Z)-2*(Z)*(Z3)+(X3)*(X3)+(Y3)*(Y3)
        +(Z3)*(Z3))


THESE ARE THE EQUATIONS FOR THE VARIABLE X

0=
+DD(X)*MASS(POINT)
+(X)*(K1)
-(X1)*(K1)
+(X)*(K2)
-(X2)*(K2)
+(X)*(K3)
-(X3)*(K3)
+(X)*(X)*D(X)*(K5F/FL)
-(X)*(X1)*D(X)*(K5F/FL)
+(X)*(Y)*D(Y)*(K5F/FL)
-(X)*(Y1)*D(Y)*(K5F/FL)
+(X)*(Z)*D(Z)*(K5F/FL)
-(X)*(Z1)*D(Z)*(K5F/FL)
-(X1)*(X)*D(X)*(K5F/FL)
+(X1)*(X1)*D(X)*(K5F/FL)
-(X1)*(Y)*D(Y)*(K5F/FL)
+(X1)*(Y1)*D(Y)*(K5F/FL)
-(X1)*(Z)*D(Z)*(K5F/FL)
+(X1)*(Z1)*D(Z)*(K5F/FL)
+(X)*(X)*D(X)*(K5G/FL)
-(X)*(X2)*D(X)*(K5G/FL)
+(X)*(Y)*D(Y)*(K5G/FL)
-(X)*(Y2)*D(Y)*(K5G/FL)
+(X)*(Z)*D(Z)*(K5G/FL)
-(X)*(Z2)*D(Z)*(K5G/FL)
-(X2)*(X)*D(X)*(K5G/FL)
+(X2)*(X2)*D(X)*(K5G/FL)
-(X2)*(Y)*D(Y)*(K5G/FL)
```

```
+(X2)*(Y2)*D(Y)*(K5G/FL)
-(X2)*(Z)*D(Z)*(K5G/FL)
+(X2)*(Z2)*D(Z)*(K5G/FL)
+(X)*(X)*D(X)*(K5H/FL)
-(X)*(X3)*D(X)*(K5H/FL)
+(X)*(Y)*D(Y)*(K5H/FL)
-(X)*(Y3)*D(Y)*(K5H/FL)
+(X)*(Z)*D(Z)*(K5H/FL)
-(X)*(Z3)*D(Z)*(K5H/FL)
-(X3)*(X)*D(X)*(K5H/FL)
+(X3)*(X3)*D(X)*(K5H/FL)
-(X3)*(Y)*D(Y)*(K5H/FL)
+(X3)*(Y3)*D(Y)*(K5H/FL)
-(X3)*(Z)*D(Z)*(K5H/FL)
+(X3)*(Z3)*D(Z)*(K5H/FL)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE Y

```
0=
+DD(Y)*MASS(POINT)
+(Y)*(K1)
-(Y1)*(K1)
+(Y)*(K2)
-(Y2)*(K2)
+(Y)*(K3)
-(Y3)*(K3)
+(Y)*(X)*D(X)*(K5F/FL)
-(Y)*(X1)*D(X)*(K5F/FL)
+(Y)*(Y)*D(Y)*(K5F/FL)
-(Y)*(Y1)*D(Y)*(K5F/FL)
+(Y)*(Z)*D(Z)*(K5F/FL)
-(Y)*(Z1)*D(Z)*(K5F/FL)
-(Y1)*(X)*D(X)*(K5F/FL)
+(Y1)*(X1)*D(X)*(K5F/FL)
-(Y1)*(Y)*D(Y)*(K5F/FL)
+(Y1)*(Y1)*D(Y)*(K5F/FL)
-(Y1)*(Z)*D(Z)*(K5F/FL)
+(Y1)*(Z1)*D(Z)*(K5F/FL)
+(Y)*(X)*D(X)*(K5G/FL)
-(Y)*(X2)*D(X)*(K5G/FL)
+(Y)*(Y)*D(Y)*(K5G/FL)
-(Y)*(Y2)*D(Y)*(K5G/FL)
+(Y)*(Z)*D(Z)*(K5G/FL)
-(Y)*(Z2)*D(Z)*(K5G/FL)
-(Y2)*(X)*D(X)*(K5G/FL)
+(Y2)*(X2)*D(X)*(K5G/FL)
-(Y2)*(Y)*D(Y)*(K5G/FL)
+(Y2)*(Y2)*D(Y)*(K5G/FL)
-(Y2)*(Z)*D(Z)*(K5G/FL)
+(Y2)*(Z2)*D(Z)*(K5G/FL)
+(Y)*(X)*D(X)*(K5H/FL)
-(Y)*(X3)*D(X)*(K5H/FL)
+(Y)*(Y)*D(Y)*(K5H/FL)
-(Y)*(Y3)*D(Y)*(K5H/FL)
+(Y)*(Z)*D(Z)*(K5H/FL)
-(Y)*(Z3)*D(Z)*(K5H/FL)
-(Y3)*(X)*D(X)*(K5H/FL)
+(Y3)*(X3)*D(X)*(K5H/FL)
```

```
-(Y3)*(Y)*D(Y)*(K5H/FL)
+(Y3)*(Y3)*D(Y)*(K5H/FL)
-(Y3)*(Z)*D(Z)*(K5H/FL)
+(Y3)*(Z3)*D(Z)*(K5H/FL)


THESE ARE THE EQUATIONS FOR THE VARIABLE Z

0=
+DD(Z)*MASS(POINT)
-(WEIGHT)
+(Z)*(K1)
-(Z1)*(K1)
+(Z)*(K2)
-(Z2)*(K2)
+(Z)*(K3)
-(Z3)*(K3)
+(Z)*(X)*D(X)*(K5F/FL)
-(Z)*(X1)*D(X)*(K5F/FL)
+(Z)*(Y)*D(Y)*(K5F/FL)
-(Z)*(Y1)*D(Y)*(K5F/FL)
+(Z)*(Z)*D(Z)*(K5F/FL)
-(Z)*(Z1)*D(Z)*(K5F/FL)
-(Z1)*(X)*D(X)*(K5F/FL)
+(Z1)*(X1)*D(X)*(K5F/FL)
-(Z1)*(Y)*D(Y)*(K5F/FL)
+(Z1)*(Y1)*D(Y)*(K5F/FL)
-(Z1)*(Z)*D(Z)*(K5F/FL)
+(Z1)*(Z1)*D(Z)*(K5F/FL)
+(Z)*(X)*D(X)*(K5G/FL)
-(Z)*(X2)*D(X)*(K5G/FL)
+(Z)*(Y)*D(Y)*(K5G/FL)
-(Z)*(Y2)*D(Y)*(K5G/FL)
+(Z)*(Z)*D(Z)*(K5G/FL)
-(Z)*(Z2)*D(Z)*(K5G/FL)
-(Z2)*(X)*D(X)*(K5G/FL)
+(Z2)*(X2)*D(X)*(K5G/FL)
-(Z2)*(Y)*D(Y)*(K5G/FL)
+(Z2)*(Y2)*D(Y)*(K5G/FL)
-(Z2)*(Z)*D(Z)*(K5G/FL)
+(Z2)*(Z2)*D(Z)*(K5G/FL)
+(Z)*(X)*D(X)*(K5H/FL)
-(Z)*(X3)*D(X)*(K5H/FL)
+(Z)*(Y)*D(Y)*(K5H/FL)
-(Z)*(Y3)*D(Y)*(K5H/FL)
+(Z)*(Z)*D(Z)*(K5H/FL)
-(Z)*(Z3)*D(Z)*(K5H/FL)
-(Z3)*(X)*D(X)*(K5H/FL)
+(Z3)*(X3)*D(X)*(K5H/FL)
-(Z3)*(Y)*D(Y)*(K5H/FL)
+(Z3)*(Y3)*D(Y)*(K5H/FL)
-(Z3)*(Z)*D(Z)*(K5H/FL)
+(Z3)*(Z3)*D(Z)*(K5H/FL)
```

```
*        CASE 4
*
*     IN ORDER TO SHOW HOW A DIFFERENT COORDINATE SYSTEM MAY BE USED, THIS
*     CASE IS FOR A POINT MASS MOVING IN THE Y-Z PLANE WITH IT'S POSITION
*     EXPRESSED IN POLAR COORDINATES.  (THIS IS THE SAME MOTION AS FOR
*     CASE 2).
*
*     R IS THE SCALAR DISTANCE TO THE ORIGIN.
*     THETA IS THE ANGULAR ORIENTATION WITH RESPECT TO DOWN.
*
*     THE SPRING-DAMPER IS AGAIN CONNECTED BETWEEN THE POINT MASS AND THE
*     POINT (0,0,ZO) , AND HAS THE COEFFICIENT K AND B.
*
MASS     X 00V   P
         THETA
 0
 0
          R
FEXT     X 00V   U 00C
         THETA
 1,1
 1,1
 2,2       R     WEIGHT
FLINSPR  X 00V   U 00C
   K     THETA
  1,1
  1,1
  2,2     R       ZO
FDLIN    X 00V   U 00C
  B       THETA
 1,1
 1,1
 2, 2     R       ZO
```

THIS IS THE X POSITION EXPANSION FOR THE MASS

THIS IS THE Y POSITION EXPANSION FOR THE MASS

-(R)*SIN(THETA)

THIS IS THE Z POSITION EXPANSION FOR THE MASS

+(R)*COS(THETA)

THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K5D/FL=(B)/(+(R)*(R)-2*COS(THETA)*(R)*(ZO)+(ZO)*(ZO))

THESE ARE THE EQUATIONS FOR THE VARIABLE R

```
0=
+DD(R)*MASS
-(R)*D(THETA)*D(THETA)*MASS
-(WEIGHT)*COS(THETA)
+(R)*(K)
-(ZO)*(K)*COS(THETA)
+(R)*(R)*D(R)*(K5D/FL)
-(R)*(ZO)*D(R)*(K5D/FL)*COS(THETA)
+(R)*(R)*(ZO)*D(THETA)*(K5D/FL)*SIN(THETA)
-(ZO)*COS(THETA)*(R)*D(R)*(K5D/FL)
+(ZO)*COS(THETA)*(ZO)*D(R)*(K5D/FL)*COS(THETA)
-(ZO)*COS(THETA)*(R)*(ZO)*D(THETA)*(K5D/FL)*SIN(THETA)


THESE ARE THE EQUATIONS FOR THE VARIABLE THETA

0=
+2*(R)*D(THETA)*D(R)*MASS
+(R)*(R)*DD(THETA)*MASS
+(WEIGHT)*(R)*SIN(THETA)
+(R)*(ZO)*(K)*SIN(THETA)
+(R)*(ZO)*SIN(THETA)*(R)*D(R)*(K5D/FL)
-(R)*(ZO)*SIN(THETA)*(ZO)*D(R)*(K5D/FL)*COS(THETA)
+(R)*(ZO)*SIN(THETA)*(R)*(ZO)*D(THETA)*(K5D/FL)*SIN(THETA)
```

```
           CASE 5
      THIS EXAMPLE CONSIDERS A POINT MASS SYSTEM IN WHICH THE COORDINATES
   OF A SATELLITE ARE EXPRESSED IN TERMS OF

      R = DISTANCE FROM CENTER OF EARTH TO SATELLITE (IN THE ORBITAL PLANE).
      THETA = POLAR ANGLE OF SATELLITE POSITION (MEASURED IN THE ORBITAL
              PLANE FROM THE POINT OF MAXIMUM LATITUDE).
      ALPHA = INCLINATION ANGLE OF ORBITAL PLANE.
      BETA = ANGLE TO MOVE ORBIT ABOUT THE N-S EARTH AXIS TO DESIRED
             ORIENTATION.
      GR = GRAVITATIONAL ATTRACTION CONSTANT OF THE EARTH-POINT MASS
           COMBINATION.

MASS    K 000   I 000   Z 000
        BETA    ALPHA   THETA
 BLANK
                        R
FGRAV   U 000   K 000   I 000   Z 000
 GR             BETA    ALPHA   THETA
 1,1
 1,1                            R
 2,4


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K2B/FL=(GR)/((+(R)*(R))**1.5)


THESE ARE THE EQUATIONS FOR THE VARIABLE R

0=
+DD(R)*MASS
-(R)*D(THETA)*D(THETA)*MASS
+(R)*(K2B/FL)


THESE ARE THE EQUATIONS FOR THE VARIABLE THETA

0=
+2*(R)*D(THETA)*D(R)*MASS
+(R)*(R)*DD(THETA)*MASS
```

```
            CASE 6
        IF THE POSITION OF A SATELLITE IS EXPRESSED IN SPHERICAL COORDINATES
      IT COULD BE REPRESENTED AS FOLLOWS

       R=DISTANCE FROM CENTER OF EARTH.
       THETA=LATITUDE OF THE SATELLITE.
       PHI=LONGITUDE OF THE SATELLITE.

MASS     Z 000   X 0V0   P
 SATELL PHI      THETA
                 BLANK
                 R
FGRAV    U 000   Z 000   X 0V0
 GR               PHI    THETA
 1,1
 1,1                      R
 2,3


THIS IS THE X POSITION EXPANSION FOR THE MASS SATELL

-(R)*SIN(PHI)*COS(THETA)


THIS IS THE Y POSITION EXPANSION FOR THE MASS SATELL

+(R)*COS(PHI)*COS(THETA)


THIS IS THE Z POSITION EXPANSION FOR THE MASS SATELL

+(R)*SIN(THETA)


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K2B/FL=(GR)/((+(R)*(R))**1.5)


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI

0=
+(R)*(R)*DD(PHI)*MASS(SATELL)*COSCOS(THETA)
+2*(R)*D(R)*D(PHI)*MASS(SATELL)*COSCOS(THETA)
-2*(R)*(R)*D(THETA)*D(PHI)*MASS(SATELL)*SINCOS(THETA)


THESE ARE THE EQUATIONS FOR THE VARIABLE R

0=
-(R)*D(PHI)*D(PHI)*MASS(SATELL)*COSCOS(THETA)
+DD(R)*MASS(SATELL)
-(R)*D(THETA)*D(THETA)*MASS(SATELL)
+(R)*(K2B/FL)
```

```
THESE ARE THE EQUATIONS FOR THE VARIABLE THETA

0=
+(R)*(R)*D(PHI)*D(PHI)*MASS(SATELL)*SINCOS(THETA)
+2*(R)*D(THETA)*D(R)*MASS(SATELL)
+(R)*(R)*DD(THETA)*MASS(SATELL)
```

```
CASE 7

   THIS IS AN EXAMPLE OF A TWO MASS SYSTEM, A DOUBLE PENDULUM IN THE X-Y
PLANE.  IN ORDER TO SET THIS UP EASILY, X IS TAKEN AS DOWN AND Y IS
PERPENDICULAR TO X.
   PHI1 IS THE ANGLE BETWEEN THE FIRST PENDULUM AND THE X ASIS.
   PHI2 IS THE ANGLE BETWEEN THE SECOND PENDULUM AND THE FIRST PENDULUM.
   M1 IS THE NAME OF THE FIRST POINT MASS.
   M2 IS THE NAME OF THE SECOND POINT MASS.
L1 IS THE LENGTH OF THE FIRST PENDULUM.
   L2 IS THE LENGTH OF THE SECOND PENDULUM.
   WT1 IS THE WEIGHT OF THE FIRST MASS (=G*MASS(M1)).


MASS    Z CCO
 M1     PHI1
        L1
MASS    Z CCO   Z CCO
 M2     PHI1    PHI2
        L1      L2
FEXT    Z COO   J COO
        PHI1
 1,1    L1      WT1
 1,1
 2,2
FEXT    Z COO   Z CCO   J CCO
        PHI1    PHI2
 1,1    L1      L2      WT2
 1,2
 3,3


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI1

0=
+(L1)*(L1)*DD(PHI1)*MASS(M1)
+(L1)*(L1)*DD(PHI1)*MASS(M2)
+2*(L2)*(L1)*DD(PHI1)*MASS(M2)*COS(PHI2)
+(L2)*(L2)*DD(PHI1)*MASS(M2)
-2*(L1)*(L2)*D(PHI2)*D(PHI1)*MASS(M2)*SIN(PHI2)
+(L1)*(L2)*DD(PHI2)*MASS(M2)*COS(PHI2)
-(L1)*(L2)*D(PHI2)*D(PHI2)*MASS(M2)*SIN(PHI2)
+(L2)*(L2)*DD(PHI2)*MASS(M2)
+(WT1)*(L1)*SIN(PHI1)
+(WT2)*(L1)*SIN(PHI1)*COS(R3)
+(WT2)*(L2)*SIN(PHI1)*COS(PHI2)*COS(R3)
+(WT2)*(L2)*COS(PHI1)*SIN(PHI2)*COS(R3)


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI2

0=
+(L2)*(L1)*DD(PHI1)*MASS(M2)*COS(PHI2)
+(L2)*(L1)*D(PHI1)*D(PHI1)*MASS(M2)*SIN(PHI2)
+(L2)*(L2)*DD(PHI1)*MASS(M2)
```

```
+(L2)*(L2)*DD(PHI2)*MASS(M2)
+(WT2)*(L2)*COS(PHI1)*SIN(PHI2)*COS(R3)
+(WT2)*(L2)*SIN(PHI1)*COS(PHI2)*COS(R3)
```

```
            CASE 8
         THIS IS ANOTHER EXAMPLE OF A TWO MASS SYSTEM, TWO PENDULUMS
      CONNECTED BY A LINEAR SPRING.  X IS AGAIN TAKEN AS DOWN.  THE TWO
      PENDULUMS ARE LIMITED TO MOTION IN THE X-Y PLANE.  THE PIVOT POINTS
      OF THE TWO ARE ALSO IN THE X-Y PLANE AT THE SAME VALUE OF X.

         L1 AND L2 ARE THE LENGTHS OF THE TWO PENDULUMS
         M1 AND M2 ARE THE MASSES OF THE TWO PENDULUMS
         A1 AND A2 ARE THE ATTACH POINTS OF THE ENDS OF THE SPRING ON THE
           LENGTHS L1 AND L2
         PHI1 AND PHI2 ARE THE ANGLES FROM VERTICAL OF L1 AND L2.
         K IS THE SPRING CONSTANT
         SEP IS THE HORIZONTAL DISTANCE BETWEEN THE PIVOT POINTS


MASS    Z COO
 M1     PHI1
        L1
MASS    U OCO   Z CCO
 M2             PHI2
                L2
        SEP
FLINSPR Z COO   U OCO   Z CCO
  K     PHI1            PHI2
 1,1    A1              A2
 1,1             SEP
 2,3


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI1

0=
+(L1)*(L1)*DD(PHI1)*MASS(M1)
-(A1)*(SEP)*(K)*COS(PHI1)
+(A1)*(A2)*(K)*SIN(PHI1)*COS(PHI2)
-(A1)*(A2)*(K)*COS(PHI1)*SIN(PHI2)


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI2

0=
+(L2)*(L2)*DD(PHI2)*MASS(M2)
+(A2)*(SEP)*(K)*COS(PHI2)
```

```
*           CASE 9
*        THIS IS THE SAME AS CASE 8 EXCEPT THAT IN CASE 8 THE SPRING IS
*      CONSIDERED TO HAVE A FORCE EXCEPT WHEN A1 AND A2 ARE AT THE SAME
*      POINT IN SPACE.  IN ORDER TO CONSIDER A FINITE LENGTH SPRING, IT IS
*      NECESSARY TO HAVE A MASSLESS LENGTH CONNECTING THE ATTACH POINT OF
*      ONE ROD (SAY A1) WITH THE END OF THE SPRING.  THE OTHER END OF THE
*      SPRING IS THEN ATTACHED TO POINT A2.  THIS ADDITIONAL MASS WILL BE
*      DESIGNATED SIMPLY A MO, AND THE ANGLE WHICH IT MAKES WITH THE FIRST
*      PENDULUM AS PHIO.  THE FORCELESS LENGTH OF THE SPRING IS CALLED
*      L-SPR.  ALL OTHER NOTATION IS THE
*      SAME AS FOR CASE 6.
*
MASS    Z CCO
 M1     PHI1
        L1
MASS    Z COC
 M2     PHI2
        L2
MASS    Z COO  Z CCO
 MO     PHI1   PHIO
        A1     L-SPR
FLINSPR Z COO  Z COC  O OCO  Z CCO
  K     PHI1   PHIO          PHI2
 1,1    A1     L-SPR         A2
 1,2                  SEP
 3,4


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI1

0=
+(L1)*(L1)*DD(PHI1)*MASS(M1)
+(A1)*(A1)*DD(PHI1)*MASS(MO)
+2*(L-SPR)*(A1)*DD(PHI1)*MASS(MO)*COS(PHIO)
+(L-SPR)*(L-SPR)*DD(PHI1)*MASS(MO)
-2*(A1)*(L-SPR)*D(PHIO)*D(PHI1)*MASS(MO)*SIN(PHIO)
+(A1)*(L-SPR)*DD(PHIO)*MASS(MO)*COS(PHIO)
-(A1)*(L-SPR)*D(PHIO)*D(PHIO)*MASS(MO)*SIN(PHIO)
+(L-SPR)*(L-SPR)*DD(PHIO)*MASS(MO)
-(A1)*(SEP)*(K)*COS(PHI1)
+(A1)*(A2)*(K)*SIN(PHI1)*COS(PHI2)
-(A1)*(A2)*(K)*COS(PHI1)*SIN(PHI2)
+(L-SPR)*(SEP)*(K)*SIN(PHI1)*SIN(PHIO)
-(L-SPR)*(SEP)*(K)*COS(PHI1)*COS(PHIO)
+(L-SPR)*(A2)*(K)*COS(PHI1)*SIN(PHIO)*COS(PHI2)
+(L-SPR)*(A2)*(K)*SIN(PHI1)*COS(PHIO)*COS(PHI2)
+(L-SPR)*(A2)*(K)*SIN(PHI1)*SIN(PHIO)*SIN(PHI2)
-(L-SPR)*(A2)*(K)*COS(PHI1)*COS(PHIO)*SIN(PHI2)


THESE ARE THE EQUATIONS FOR THE VARIABLE PHI2

0=
+(L2)*(L2)*DD(PHI2)*MASS(M2)
+(A2)*(SEP)*(K)*COS(PHI2)
```

```
THESE ARE THE EQUATIONS FOR THE VARIABLE PHIO

0=
+(L-SPR)*(A1)*DD(PHI1)*MASS(MO)*COS(PHIO)
+(L-SPR)*(A1)*D(PHI1)*D(PHI1)*MASS(MO)*SIN(PHIO)
+(L-SPR)*(L-SPR)*DD(PHI1)*MASS(MO)
+(L-SPR)*(L-SPR)*DD(PHIO)*MASS(MO)
-(L-SPR)*(A1)*(K)*SIN(PHIO)
+(L-SPR)*(SEP)*(K)*SIN(PHI1)*SIN(PHIO)
-(L-SPR)*(SEP)*(K)*COS(PHI1)*COS(PHIO)
+(L-SPR)*(A2)*(K)*COS(PHI1)*SIN(PHIO)*COS(PHI2)
+(L-SPR)*(A2)*(K)*SIN(PHI1)*COS(PHIO)*COS(PHI2)
+(L-SPR)*(A2)*(K)*SIN(PHI1)*SIN(PHIO)*SIN(PHI2)
-(L-SPR)*(A2)*(K)*COS(PHI1)*COS(PHIO)*SIN(PHI2)
```

TO DERIVE THE EQUATIONS OF MOTION OF A BODY WITH MASS, THE FOLLOWING EXAMPLES (CASES 10 THROUGH 14) ARE GIVEN.

CASE 10

THIS IS FOR AN AIRPLANE.
U,V,W ARE THE VELOCITIES OF THE AIRPLAE ALONG THE X,Y,Z BODY AXES.
P,Q,R ARE THE ROTATIONAL RATES ABOUT THE X,Y,Z BODY AXES.
DIRCOS IS THE DIRECTION COSINE MATRIX OF THE X,Y,Z BODY AXIS WITH RESPECT TO EARTH X,Y,Z AXES.
THE X,Y,Z AXES ARE TAKE IN THE NORMAL MANNER, X IS TOWARDS THE NOSE, Y IS TOWARD THE RIGHT WINGTIP, AND Z IS TOWARD THE BOTTOM OF THE PLANE - THE ORIGIN BEING AT THE CENTER OF MASS OF THE VEHICLE.
THE AERODYNAMIC FORCES ARE CONSIDERED TO BE EXTERNAL FORCES. THESE FORCES ARE -

FRTW (FORCE AT RIGHT WING)-ACTS IN THE Z DIRECTION AT POINT (XFRTW, YFRTW,ZFRTW)
FLTW (FORCE AT LEFT WING)-Z FORCE AT POINT (XFLTW,YFLTW,ZFLTW)
FEL (FORCE AT ELEVATORS)-Z FORCE AT POINT (XFEL,0,ZFEL)
FRUD (FORCE AT RUDDER)- Y FORCE AT POINT (XFRUD,0,ZFRUD)
FLAT (LATERAL FORCE OF BODY)-Y FORCE AT POINT (0,0,0).
F(T-D) (THRUST MINUS DRAG)- X FORCE AT POINT (0,0,0).

IT IS RECOGNIZED THAT THESE FORCES MAY BE EXPRESSED IN TERMS OF CONTROL SURFACE DEFLECTIONS, ANGLE OF ATTACK, ANGLE OF SIDESLIP, ETC.

IN ADDITION, THE FORCE OF GRAVITY IS CONSIDERED TO BE THE WEIGHT OF THE AIRPLANE ACTING AS AN EXTERNAL FORCE AND PUSHING IN THE Z EARTH DIRECTION.
IXY=IYZ=0 , IXZ IS NOT ZERO.

```
MASS     0 000   0 CC0   0 AAA
                 DIRCOS
          U       P       A1
  IXZ     V       Q       A2
          W       R       A3
FEXT     0 000   0 0C0   0 CCC   0 00C
                 DIRCOS
 1,2      U       P      XFRTW
 3,3      V       Q      YFRTW
 4,4      W       R      ZFRTW   FRTW
FEXT     0 000   0 0C0   0 CCC   0 00C
                 DIRCOS
 1,2      U       P      XFLTW
  3,3     V       Q      YFLTW
  4,4     W       R      ZFLTW   FLTW
FEXT     0 000   0 0C0   0 C0C   0 C0C
                 DIRCOS
 1,2      U       P      XFEL
 3,3      V       Q
 4,4      W       R      ZFEL    FEL
FEXT     0 000   0 0C0   0 C0C   0 CC0
                 DIRCOS
 1,2      U       P      XFRUD
```

```
 3,3     V      Q                   FRUD
 4,4     W      R          ZFRUD
FEXT     U UUU  D CCO   U CCO    U CCO
                DIRCOS
  1,2    U      P                   F(T-D)
  3,3    V      Q                   FLAT
  4,4    W      R
FEXT     U UUU  D UCC   U UOC
                DIRCOS
 1,1     U      P
 1,2     V      Q
 3,3     W      R       WEIGHT
```

THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

```
D(ZPOSA1)=DIRCOS(1,1)*(U)+DIRCOS(1,2)*(V)+DIRCOS(1,3)*(W)

D(ZPOSA2)=DIRCOS(2,1)*(U)+DIRCOS(2,2)*(V)+DIRCOS(2,3)*(W)

D(ZPOSA3)=DIRCOS(3,1)*(U)+DIRCOS(3,2)*(V)+DIRCOS(3,3)*(W)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE U

```
0=
+D(U)*MASS
-(R)*(V)*MASS
+(Q)*(W)*MASS
-(F(T-D))
-(WEIGHT)*DIRCOS(3,1)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE V

```
0=
+D(V)*MASS
+(R)*(U)*MASS
-(P)*(W)*MASS
-(FRUD)
-(FLAT)
-(WEIGHT)*DIRCOS(3,2)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE W

```
0=
+D(W)*MASS
-(Q)*(U)*MASS
+(P)*(V)*MASS
-(FRTW)
-(FLTW)
-(FEL)
-(WEIGHT)*DIRCOS(3,3)
```

```
THESE ARE THE EQUATIONS FOR THE VARIABLE P

0=
-(Q)*(P)*IXZ
+D(P)*IX
-(R)*(Q)*IY
+(R)*(Q)*IZ
-D(R)*IXZ
-(FRTW)*(YFRTW)
-(FLTW)*(YFLTW)
+(FRUD)*(ZFRUD)


THESE ARE THE EQUATIONS FOR THE VARIABLE Q

0=
+(P)*(P)*IXZ
+(R)*(P)*IX
-(R)*(P)*IZ
+D(Q)*IY
-(R)*(R)*IXZ
+(FRTW)*(XFRTW)
+(FLTW)*(XFLTW)
+(FEL)*(XFEL)


THESE ARE THE EQUATIONS FOR THE VARIABLE R

0=
-(Q)*(P)*IX
+(Q)*(P)*IY
-D(P)*IXZ
+(R)*(Q)*IXZ
+D(R)*IZ
-(FRUD)*(XFRUD)
```

```
CASE 11

   EQUATIONS FOR A FLEXIBLE MISSILE.

   CONSIDER THE MISSILE TO BE MOVING ONLY IN THE X-Y PLANE.  X IS
THE LONGITUDINAL AXIS OF THE MISSILE AND Y IS THE AXIS PERPENDICULAR
TO THIS.  THE POINT MASSES WHICH COMPOSE THE BODY ARE ASSUMED TO BE
LOCATED IN THE X DIRECTION AT A DISTANCE OF XTOPNT (X TO POINT) FROM
THE ORIGIN OF THE X-Y AXIS SYSTEM, WHILE THE Y POSITION IS THE SUM
OF THE BENDING MODE DISPLACEMENTS(QI) TIMES THE BENDING MODE SHAPES
(PI(X)).  IN ORDER TO DEVELOP THESE EQUATIONS, THE VARIABLE QI IS
USED TO REPRESENT ONLY THE I-TH VARIABLE AND QJ REPRESENTS ANY OTHER,
HOWEVER, THE ACTUAL VARIABLES USED ARE P(X)QI AND P(X)QJ WHERE P(X)QI
IS ACTUALLY PI(X)*QI.  THIS MEANS THAT TO OBTAIN THE EQUATION FOR
THE VARIABLE QI, THE EQUATION AS DERIVED FOR P(X)QI MUST BE MULTIPLIED
BY PI(X).
   U,V ARE THE VELOCITIES ALONG THE MOVING X-Y COORDINATE SYSTEM
   OM IS THE ROTATIONAL RATE ABOUT THE MOVING Z AXIS.
   PNT IS THE DESIGNATION OF THE MASS POINT.

   THE FINAL EQUATIONS MUST BE INTEGRATED OVER ALL OF THE POINT MASSES
IN THE BODY.

MASS     U UUU   D COO   U CVO   U OVO
 PNT             DIRCOS
           U             XTOPNT
           V             P(X)QI  P(X)QJ
                 OM


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

D(ZPOSA1)=DIRCOS(1,1)*(U)+DIRCOS(1,2)*(V)

D(ZPOSA2)=DIRCOS(2,1)*(U)+DIRCOS(2,2)*(V)

D(ZPOSA3)=DIRCOS(3,1)*(U)+DIRCOS(3,2)*(V)


THESE ARE THE EQUATIONS FOR THE VARIABLE U

0=
+D(U)*MASS(PNT)
-(OM)*(V)*MASS(PNT)
-2*(OM)*D(P(X)QI)*MASS(PNT)
-2*(OM)*D(P(X)QJ)*MASS(PNT)
-(XTOPNT)*(OM)*(OM)*MASS(PNT)
-(P(X)QI)*D(OM)*MASS(PNT)
-(P(X)QJ)*D(OM)*MASS(PNT)


THESE ARE THE EQUATIONS FOR THE VARIABLE V

0=
```

```
+D(V)*MASS(PNT)
+(OM)*(U)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
-(P(X)QI)*(OM)*(CM)*MASS(PNT)
-(P(X)QJ)*(OM)*(OM)*MASS(PNT)
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE OM

```
0=
+(XTOPNT)*(OM)*(U)*MASS(PNT)
+(XTOPNT)*D(V)*MASS(PNT)
-(P(X)QI)*D(U)*MASS(PNT)
+(P(X)QI)*(OM)*(V)*MASS(PNT)
-(P(X)QJ)*D(U)*MASS(PNT)
+(P(X)QJ)*(OM)*(V)*MASS(PNT)
+(XTOPNT)*DD(P(X)QI)*MASS(PNT)
+(XTOPNT)*DD(P(X)QJ)*MASS(PNT)
+2*(P(X)QI)*(OM)*D(P(X)QI)*MASS(PNT)
+2*(P(X)QJ)*(OM)*D(P(X)QI)*MASS(PNT)
+2*(P(X)QI)*(OM)*D(P(X)QJ)*MASS(PNT)
+2*(P(X)QJ)*(OM)*D(P(X)QJ)*MASS(PNT)
+(XTOPNT)*(XTOPNT)*D(OM)*MASS(PNT)
+(P(X)QI)*(P(X)QI)*D(OM)*MASS(PNT)
+2*(P(X)QJ)*(P(X)QI)*D(OM)*MASS(PNT)
+(P(X)QJ)*(P(X)QJ)*D(CM)*MASS(PNT)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE P(X)QI

```
0=
+(OM)*(U)*MASS(PNT)
+D(V)*MASS(PNT)
-(P(X)QI)*(OM)*(OM)*MASS(PNT)
-(P(X)QJ)*(OM)*(OM)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE P(X)QJ

```
0=
+(OM)*(U)*MASS(PNT)
+D(V)*MASS(PNT)
-(P(X)QI)*(OM)*(OM)*MASS(PNT)
-(P(X)QJ)*(OM)*(CM)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
```

```
*              CASE 12

*         THIS IS AN EXTENSION OF CASE NUMBER 11.  FOR THIS CASE, AN ENGINE
*      IS ATTACHED AT POINT XENG.  THE ENGINE IS ROTATED IN THE X-Y PLANE
*      AN ANGLE DEL' FROM THE MOVING X AXIS.  THE ENGINE IS ASSUMED RIGID
*      WITH RESPECT TO OTHER PARTS OF THE ENGINE, THEREFORE THE INTEGRATION
*      IS CARRIED OUT IN THE PROGRAM OVER THIS MASS.  THE DISTANCE FROM THE
*      ENGINE PIVOT POINT XENG TO THE CENTER OF MASS OF THE ENGINE IS XCMENG.
*      ALL OTHER NOTATIONS ARE THE SAME AS FOR CASE 11.

MASS    U UUO   D UUU   U CVC   U OVO
 PNT            DIRCOS
         U              XTOPNT
         V              P(X)QI  P(X)QJ
                 OM
MASS    U UUO   D OOO   U CVO   U OVO   Z CCO   U AAA
 ENG            DIRCOS                  DE L'
          U             XENG            XCMENG    A1
        V               P(X)QI  P(X)QJ            A2
                OM                                A3


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

D(ZPOSA1)=DIRCOS(1,1)*(U)+DIRCOS(1,2)*(V)

D(ZPOSA2)=DIRCOS(2,1)*(U)+DIRCOS(2,2)*(V)

D(ZPOSA3)=DIRCOS(3,1)*(U)+DIRCOS(3,2)*(V)


THESE ARE THE EQUATIONS FOR THE VARIABLE U

0=
+D(U)*MASS(PNT)
-(OM)*(V)*MASS(PNT)
-2*(OM)*D(P(X)QI)*MASS(PNT)
-2*(OM)*D(P(X)QJ)*MASS(PNT)
-(XTOPNT)*(OM)*(OM)*MASS(PNT)
-(P(X)QI)*D(OM)*MASS(PNT)
-(P(X)QJ)*D(OM)*MASS(PNT)
+D(U)*MASS(ENG)
-(OM)*(V)*MASS(ENG)
-2*(OM)*D(P(X)QI)*MASS(ENG)
-2*(OM)*D(P(X)QJ)*MASS(ENG)
-2*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*COS(DEL')
-(XENG)*(OM)*(OM)*MASS(ENG)
-(P(X)QI)*D(OM)*MASS(ENG)
-(P(X)QJ)*D(OM)*MASS(ENG)
-(XCMENG)*(OM)*(OM)*MASS(ENG)*COS(DEL')
-(XCMENG)*D(OM)*MASS(ENG)*SIN(DEL')
-(XCMENG)*DD(DEL')*MASS(ENG)*SIN(DEL')
```

```
THESE ARE THE EQUATIONS FOR THE VARIABLE V

0=
+D(V)*MASS(PNT)
+(OM)*(U)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
-(P(X)QI)*(OM)*(OM)*MASS(PNT)
-(P(X)QJ)*(OM)*(OM)*MASS(PNT)
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
+D(V)*MASS(ENG)
+(OM)*(U)*MASS(ENG)
-2*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*SIN(DEL')
+(XENG)*D(OM)*MASS(ENG)
-(P(X)QI)*(OM)*(OM)*MASS(ENG)
-(P(X)QJ)*(OM)*(OM)*MASS(ENG)
-(XCMENG)*(OM)*(OM)*MASS(ENG)*SIN(DEL')
+(XCMENG)*D(OM)*MASS(ENG)*COS(DEL')
+DD(P(X)QI)*MASS(ENG)
+DD(P(X)QJ)*MASS(ENG)
+(XCMENG)*DD(DEL')*MASS(ENG)*COS(DEL')


THESE ARE THE EQUATIONS FOR THE VARIABLE OM

0=
+(XTOPNT)*(OM)*(U)*MASS(PNT)
+(XTOPNT)*D(V)*MASS(PNT)
-(P(X)QI)*D(U)*MASS(PNT)
+(P(X)QI)*(OM)*(V)*MASS(PNT)
-(P(X)QJ)*D(U)*MASS(PNT)
+(P(X)QJ)*(OM)*(V)*MASS(PNT)
+(XTOPNT)*DD(P(X)QI)*MASS(PNT)
+(XTOPNT)*DD(P(X)QJ)*MASS(PNT)
+2*(P(X)QI)*(OM)*D(P(X)QI)*MASS(PNT)
+2*(P(X)QJ)*(OM)*D(P(X)QI)*MASS(PNT)
+2*(P(X)QI)*(OM)*D(P(X)QJ)*MASS(PNT)
+2*(P(X)QJ)*(OM)*D(P(X)QJ)*MASS(PNT)
+(XTOPNT)*(XTOPNT)*D(OM)*MASS(PNT)
+(P(X)QI)*(P(X)QI)*D(OM)*MASS(PNT)
+2*(P(X)QJ)*(P(X)QI)*D(OM)*MASS(PNT)
+(P(X)QJ)*(P(X)QJ)*D(OM)*MASS(PNT)
+(XENG)*(OM)*(U)*MASS(ENG)
+(XENG)*D(V)*MASS(ENG)
-(P(X)QI)*D(U)*MASS(ENG)
+(P(X)QI)*(OM)*(V)*MASS(ENG)
-(P(X)QJ)*D(U)*MASS(ENG)
+(P(X)QJ)*(OM)*(V)*MASS(ENG)
+(XCMENG)*(OM)*(U)*MASS(ENG)*COS(DEL')
-(XCMENG)*D(U)*MASS(ENG)*SIN(DEL')
+(XCMENG)*(OM)*(V)*MASS(ENG)*SIN(DEL')
+(XCMENG)*D(V)*MASS(ENG)*COS(DEL')
+(XENG)*DD(P(X)QI)*MASS(ENG)
+(XCMENG)*DD(P(X)QI)*MASS(ENG)*COS(DEL')
+(XENG)*DD(P(X)QJ)*MASS(ENG)
+(XCMENG)*DD(P(X)QJ)*MASS(ENG)*COS(DEL')
+(XENG)*(XCMENG)*DD(DEL')*MASS(ENG)*COS(DEL')
-(XENG)*(XCMENG)*D(DEL')*D(DEL')*MASS(ENG)*SIN(DEL')
```

```
+(P(X)QI)*(XCMENG)*DD(DEL')*MASS(ENG)*SIN(DEL')
+(P(X)QI)*(XCMENG)*D(DEL')*D(DEL')*MASS(ENG)*COS(DEL')
+(P(X)QJ)*(XCMENG)*DD(DEL')*MASS(ENG)*SIN(DEL')
+(P(X)QJ)*(XCMENG)*D(DEL')*D(DEL')*MASS(ENG)*COS(DEL')
+(XCMENG)*(XCMENG)*DD(DEL')*MASS(ENG)
+DD(DEL')*IZ(ENG)
+2*(P(X)QI)*(OM)*D(P(X)QI)*MASS(ENG)
+2*(P(X)QJ)*(OM)*D(P(X)QI)*MASS(ENG)
+2*(XCMENG)*(OM)*D(P(X)QI)*MASS(ENG)*SIN(DEL')
+2*(P(X)QI)*(OM)*D(P(X)QJ)*MASS(ENG)
+2*(P(X)QJ)*(OM)*D(P(X)QJ)*MASS(ENG)
+2*(XCMENG)*(OM)*D(P(X)QJ)*MASS(ENG)*SIN(DEL')
-2*(XENG)*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*SIN(DEL')
+2*(P(X)QI)*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*COS(DEL')
+2*(P(X)QJ)*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*COS(DEL')
+(XENG)*(XENG)*D(OM)*MASS(ENG)
+2*(XCMENG)*(XENG)*D(OM)*MASS(ENG)*COS(DEL')
+(P(X)QI)*(P(X)QI)*D(OM)*MASS(ENG)
+2*(P(X)QJ)*(P(X)QI)*D(OM)*MASS(ENG)
+2*(XCMENG)*(P(X)QI)*D(OM)*MASS(ENG)*SIN(DEL')
+(P(X)QJ)*(P(X)QJ)*D(OM)*MASS(ENG)
+2*(XCMENG)*(P(X)QJ)*D(OM)*MASS(ENG)*SIN(DEL')
+(XCMENG)*(XCMENG)*D(OM)*MASS(ENG)
+D(OM)*IZ(ENG)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE P(X)QI

```
0=
+(OM)*(U)*MASS(PNT)
+D(V)*MASS(PNT)
-(P(X)QI)*(OM)*(OM)*MASS(PNT)
-(P(X)QJ)*(OM)*(OM)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
+(OM)*(U)*MASS(ENG)
+D(V)*MASS(ENG)
-(P(X)QI)*(OM)*(OM)*MASS(ENG)
-(P(X)QJ)*(OM)*(OM)*MASS(ENG)
-(XCMENG)*(OM)*(OM)*MASS(ENG)*SIN(DEL')
+(XENG)*D(OM)*MASS(ENG)
+(XCMENG)*D(OM)*MASS(ENG)*COS(DEL')
-2*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*SIN(DEL')
+DD(P(X)QI)*MASS(ENG)
+DD(P(X)QJ)*MASS(ENG)
+(XCMENG)*DD(DEL')*MASS(ENG)*COS(DEL')
-(XCMENG)*D(DEL')*D(DEL')*MASS(ENG)*SIN(DEL')
```

THESE ARE THE EQUATIONS FOR THE VARIABLE P(X)QJ

```
0=
+(OM)*(U)*MASS(PNT)
+D(V)*MASS(PNT)
-(P(X)QI)*(OM)*(OM)*MASS(PNT)
-(P(X)QJ)*(OM)*(OM)*MASS(PNT)
+(XTOPNT)*D(OM)*MASS(PNT)
```

```
+DD(P(X)QI)*MASS(PNT)
+DD(P(X)QJ)*MASS(PNT)
+(OM)*(U)*MASS(ENG)
+D(V)*MASS(ENG)
-(P(X)QI)*(OM)*(OM)*MASS(ENG)
-(P(X)QJ)*(OM)*(OM)*MASS(ENG)
-(XCMENG)*(OM)*(OM)*MASS(ENG)*SIN(DEL')
+(XENG)*D(OM)*MASS(ENG)
+(XCMENG)*D(OM)*MASS(ENG)*COS(DEL')
-2*(XCMENG)*(OM)*D(DEL')*MASS(ENG)*SIN(DEL')
+DD(P(X)QI)*MASS(ENG)
+DD(P(X)QJ)*MASS(ENG)
+(XCMENG)*DD(DEL')*MASS(ENG)*COS(DEL')
-(XCMENG)*D(DEL')*D(DEL')*MASS(ENG)*SIN(DEL')


THESE ARE THE EQUATIONS FOR THE VARIABLE DEL'

0=
+(XCMENG)*(OM)*(U)*MASS(ENG)*COS(DEL')
-(XCMENG)*D(U)*MASS(ENG)*SIN(DEL')
+(XCMENG)*(OM)*(V)*MASS(ENG)*SIN(DEL')
+(XCMENG)*D(V)*MASS(ENG)*COS(DEL')
+(XCMENG)*(XENG)*(OM)*(OM)*MASS(ENG)*SIN(DEL')
-(XCMENG)*(P(X)QI)*(OM)*(OM)*MASS(ENG)*COS(DEL')
-(XCMENG)*(P(X)QJ)*(OM)*(OM)*MASS(ENG)*COS(DEL')
+(XCMENG)*(XENG)*D(OM)*MASS(ENG)*COS(DEL')
+(XCMENG)*(P(X)QI)*D(OM)*MASS(ENG)*SIN(DEL')
+(XCMENG)*(P(X)QJ)*D(OM)*MASS(ENG)*SIN(DEL')
+(XCMENG)*(XCMENG)*D(OM)*MASS(ENG)
+(A3)*(A3)*D(OM)*IZ(ENG)
+2*(XCMENG)*(OM)*D(P(X)QI)*MASS(ENG)*SIN(DEL')
+2*(XCMENG)*(OM)*D(P(X)QJ)*MASS(ENG)*SIN(DEL')
+(XCMENG)*DD(P(X)QI)*MASS(ENG)*COS(DEL')
+(XCMENG)*DD(P(X)QJ)*MASS(ENG)*COS(DEL')
+(XCMENG)*(XCMENG)*DD(DEL')*MASS(ENG)
+DD(DEL')*IZ(ENG)
```

```
             CASE 13
          EQUATIONS OF A SATELLITE WHOSEVARIABLES ARE GENERALIZED COORDINATES,
       THAT IS HOLONOMIC.

          ASSUME THE CENTER OF MASS IS IN AN ORBITAL PLANE, THE POSITION
       IN THE PLANE BEING    DETERMINED BY THE PLAR COORDINATES R AND THETA.
       PITCH, ROLL, AND YAW ARE THE ANGLES OF THE BODY WITH RESPECT TO THE
       ROTATING COORDINATE SYSTEM, TAKENAS GIMBALLED ROTATIONS IN THE ORDER
       SHOWN.  AX,AY,AZ ARE THE POSITIONS IN THE BODY OF ANY POINT.

MASS    Z OVO  Y OOC  X OCO  Z AAA
        THETA  YAW    ROLL    PITCH
 IXY                           AX
 IXZ     R                     AY
 IYZ                           AZ
FGRAV   U 000  Z OVO  Y OCO  X COO  Z CCC
 KGRAV         THETA  YAW    ROLL   PITCH
 1,1                                AX
 1,1             R                  AY
 2,5                                AZ


THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

K2B/FL=(KGRAV)/(((+(R)*(R)+2*COS(ROLL)*SIN(PITCH)*(R)*(AX)+2*COS(ROLL)*COS(PITCH)*(R)*(AY)

       -2*SIN(ROLL)*(R)*(AZ)+(AX)*(AX)+(AY)*(AY)+(AZ)*(AZ))**1.5)


THESE ARE THE EQUATIONS FOR THE VARIABLE R

0=
+DD(R)*MASS
-(R)*D(THETA)*D(THETA)*MASS
+(R)*(K2B/FL)
+(AX)*(K2B/FL)*COS(ROLL)*SIN(PITCH)
+(AY)*(K2B/FL)*COS(ROLL)*COS(PITCH)
-(AZ)*(K2B/FL)*SIN(ROLL)


THESE ARE THE EQUATIONS FOR THE VARIABLE THETA

0=
+2*(R)*D(THETA)*D(R)*MASS
+(R)*(R)*DD(THETA)*MASS
-2*DD(THETA)*IXY*COSCOS(YAW)*SINCOS(PITCH)
+2*DD(THETA)*IXY*SINCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+2*DD(THETA)*IXY*COSCOS(ROLL)*SINCOS(PITCH)
-2*DD(THETA)*IXY*SINCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+2*DD(THETA)*IXY*SINSIN(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
+2*DD(THETA)*IXZ*SINCOS(YAW)*COS(ROLL)*COS(PITCH)
-2*DD(THETA)*IXZ*COSCOS(YAW)*SINCOS(ROLL)*SIN(PITCH)
-2*DD(THETA)*IYZ*SINCOS(YAW)*COS(ROLL)*SIN(PITCH)
```

```
-2*DD(THETA)*IYZ*COSCOS(YAW)*SINCOS(ROLL)*COS(PITCH)
-0.5*DD(THETA)*IX*COSCOS(YAW)*COSCOS(PITCH)
+0.5*DD(THETA)*IY*COSCOS(YAW)*COSCOS(PITCH)
-2*DD(THETA)*IX*SINCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*DD(THETA)*IY*SINCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-0.5*DD(THETA)*IX*COSCOS(ROLL)*SINSIN(PITCH)
+0.5*DD(THETA)*IY*COSCOS(ROLL)*SINSIN(PITCH)
-0.5*DD(THETA)*IX*SINSIN(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+0.5*DD(THETA)*IY*SINSIN(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+DD(THETA)*IZ*COSCOS(YAW)*COSCOS(ROLL)
+0.5*DD(THETA)*IX*COSCOS(YAW)*SINSIN(PITCH)
-0.5*DD(THETA)*IY*COSCOS(YAW)*SINSIN(PITCH)
+0.5*DD(THETA)*IX*COSCOS(ROLL)*COSCOS(PITCH)
-0.5*DD(THETA)*IY*COSCOS(ROLL)*COSCOS(PITCH)
+0.5*DD(THETA)*IX*SINSIN(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-0.5*DD(THETA)*IY*SINSIN(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+0.5*DD(THETA)*IX*SINSIN(ROLL)
+0.5*DD(THETA)*IY*SINSIN(ROLL)
+0.5*DD(THETA)*IX*SINSIN(YAW)*COSCOS(ROLL)
+0.5*DD(THETA)*IY*SINSIN(YAW)*COSCOS(ROLL)
+2*D(YAW)*D(THETA)*IXY*COSCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+4*D(YAW)*D(THETA)*IXY*SINCOS(YAW)*SINCOS(PITCH)
+4*D(YAW)*D(THETA)*IXY*SINCOS(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
+2*D(YAW)*D(THETA)*IXY*SINSIN(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(YAW)*D(THETA)*IXY*COSCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(YAW)*D(THETA)*IXY*SINSIN(YAW)*SIN(ROLL)*COSCOS(PITCH)
+2*D(YAW)*D(THETA)*IXZ*COSCOS(YAW)*COS(ROLL)*COS(PITCH)
+4*D(YAW)*D(THETA)*IXZ*SINCOS(YAW)*SINCOS(ROLL)*SIN(PITCH)
-2*D(YAW)*D(THETA)*IXZ*SINSIN(YAW)*COS(ROLL)*COS(PITCH)
-2*D(YAW)*D(THETA)*IYZ*COSCOS(YAW)*COS(ROLL)*SIN(PITCH)
+4*D(YAW)*D(THETA)*IYZ*SINCOS(YAW)*SINCOS(ROLL)*COS(PITCH)
+2*D(YAW)*D(THETA)*IYZ*SINSIN(YAW)*COS(ROLL)*SIN(PITCH)
-2*D(YAW)*D(THETA)*IX*COSCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*D(YAW)*D(THETA)*IY*COSCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+D(YAW)*D(THETA)*IX*SINCOS(YAW)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IY*SINCOS(YAW)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IX*SINCOS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+D(YAW)*D(THETA)*IY*SINCOS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
-2*D(YAW)*D(THETA)*IZ*SINCOS(YAW)*COSCOS(ROLL)
+2*D(YAW)*D(THETA)*IX*SINSIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
-2*D(YAW)*D(THETA)*IY*SINSIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
-D(YAW)*D(THETA)*IX*SINCOS(YAW)*SINSIN(PITCH)
+D(YAW)*D(THETA)*IY*SINCOS(YAW)*SINSIN(PITCH)
+D(YAW)*D(THETA)*IX*SINCOS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IY*SINCOS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+D(YAW)*D(THETA)*IX*SINCOS(YAW)*COSCOS(ROLL)
+D(YAW)*D(THETA)*IY*SINCOS(YAW)*COSCOS(ROLL)
+2*D(ROLL)*D(THETA)*IXY*SINCOS(YAW)*COS(ROLL)*COSCOS(PITCH)
-4*D(ROLL)*D(THETA)*IXY*COSCOS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-2*D(ROLL)*D(THETA)*IXY*SINCOS(YAW)*COS(ROLL)*SINSIN(PITCH)
-2*D(ROLL)*D(THETA)*IXZ*SINCOS(YAW)*SIN(ROLL)*COS(PITCH)
+2*D(ROLL)*D(THETA)*IXZ*COSCOS(YAW)*SINSIN(ROLL)*SIN(PITCH)
-2*D(ROLL)*D(THETA)*IXZ*COSCOS(YAW)*COSCOS(ROLL)*SIN(PITCH)
+2*D(ROLL)*D(THETA)*IYZ*SINCOS(YAW)*SIN(ROLL)*SIN(PITCH)
+2*D(ROLL)*D(THETA)*IYZ*COSCOS(YAW)*SINSIN(ROLL)*COS(PITCH)
-2*D(ROLL)*D(THETA)*IYZ*COSCOS(YAW)*COSCOS(ROLL)*COS(PITCH)
-2*D(ROLL)*D(THETA)*IX*SINCOS(YAW)*COS(ROLL)*SINCOS(PITCH)
+2*D(ROLL)*D(THETA)*IY*SINCOS(YAW)*COS(ROLL)*SINCOS(PITCH)
+2*D(ROLL)*D(THETA)*IX*COSCOS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
+2*D(ROLL)*D(THETA)*IY*COSCOS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
```

```
-2*D(ROLL)*D(THETA)*IZ*COSCOS(YAW)*SINCOS(ROLL)
-2*D(PITCH)*D(THETA)*IXY*COSCOS(YAW)*COSCOS(PITCH)
-8*D(PITCH)*D(THETA)*IXY*SINCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(THETA)*IXY*COSCOS(ROLL)*SINSIN(PITCH)
-2*D(PITCH)*D(THETA)*IXY*SINSIN(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+2*D(PITCH)*D(THETA)*IXY*COSCOS(YAW)*SINSIN(PITCH)
+2*D(PITCH)*D(THETA)*IXY*COSCOS(ROLL)*COSCOS(PITCH)
+2*D(PITCH)*D(THETA)*IXY*SINSIN(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-2*D(PITCH)*D(THETA)*IXZ*SINCOS(YAW)*COS(ROLL)*SIN(PITCH)
-2*D(PITCH)*D(THETA)*IXZ*COSCOS(YAW)*SINCOS(ROLL)*COS(PITCH)
-2*D(PITCH)*D(THETA)*IYZ*SINCOS(YAW)*COS(ROLL)*COS(PITCH)
+2*D(PITCH)*D(THETA)*IYZ*COSCOS(YAW)*SINCOS(ROLL)*SIN(PITCH)
+2*D(PITCH)*D(THETA)*IX*COSCOS(YAW)*SINCOS(PITCH)
-2*D(PITCH)*D(THETA)*IY*COSCOS(YAW)*SINCOS(PITCH)
-2*D(PITCH)*D(THETA)*IX*SINCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+2*D(PITCH)*D(THETA)*IY*SINCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-2*D(PITCH)*D(THETA)*IX*COSCOS(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(THETA)*IY*COSCOS(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(THETA)*IX*SINCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(PITCH)*D(THETA)*IY*SINCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(PITCH)*D(THETA)*IX*SINSIN(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(THETA)*IY*SINSIN(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
-2*DD(YAW)*IXY*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-DD(YAW)*IXY*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
+DD(YAW)*IXY*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
-D(YAW)*D(YAW)*IXY*COS(YAW)*COS(ROLL)*SINSIN(PITCH)
+2*D(YAW)*D(YAW)*IXY*SIN(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
+D(YAW)*D(YAW)*IXY*COS(YAW)*COS(ROLL)*COSCOS(PITCH)
-DD(YAW)*IXZ*COS(YAW)*COSCOS(ROLL)*SIN(PITCH)
+DD(YAW)*IXZ*COS(YAW)*SINSIN(ROLL)*SIN(PITCH)
-DD(YAW)*IXZ*SIN(YAW)*SIN(ROLL)*COS(PITCH)
+D(YAW)*D(YAW)*IXZ*SIN(YAW)*COSCOS(ROLL)*SIN(PITCH)
-D(YAW)*D(YAW)*IXZ*COS(YAW)*SIN(ROLL)*COS(PITCH)
-D(YAW)*D(YAW)*IXZ*SIN(YAW)*SINSIN(ROLL)*SIN(PITCH)
-DD(YAW)*IYZ*COS(YAW)*COSCOS(ROLL)*COS(PITCH)
+DD(YAW)*IYZ*COS(YAW)*SINSIN(ROLL)*COS(PITCH)
+DD(YAW)*IYZ*SIN(YAW)*SIN(ROLL)*SIN(PITCH)
+D(YAW)*D(YAW)*IYZ*SIN(YAW)*COSCOS(ROLL)*COS(PITCH)
+D(YAW)*D(YAW)*IYZ*COS(YAW)*SIN(ROLL)*SIN(PITCH)
-D(YAW)*D(YAW)*IYZ*SIN(YAW)*SINSIN(ROLL)*COS(PITCH)
+DD(YAW)*IX*COS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
+DD(YAW)*IY*COS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
-DD(YAW)*IZ*COS(YAW)*SINCOS(ROLL)
-DD(YAW)*IX*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
+DD(YAW)*IY*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
-D(YAW)*D(YAW)*IX*COS(YAW)*COS(ROLL)*SINCOS(PITCH)
+D(YAW)*D(YAW)*IY*COS(YAW)*COS(ROLL)*SINCOS(PITCH)
-D(YAW)*D(YAW)*IX*SIN(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
-D(YAW)*D(YAW)*IY*SIN(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
+D(YAW)*D(YAW)*IZ*SIN(YAW)*SINCOS(ROLL)
-4*D(ROLL)*D(YAW)*IXY*COS(YAW)*COSCOS(ROLL)*SINCOS(PITCH)
+4*D(ROLL)*D(YAW)*IXZ*COS(YAW)*SINCOS(ROLL)*SIN(PITCH)
+4*D(ROLL)*D(YAW)*IYZ*COS(YAW)*SINCOS(ROLL)*COS(PITCH)
+D(ROLL)*D(YAW)*IX*COS(YAW)*COSCOS(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(YAW)*IY*COS(YAW)*COSCOS(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(YAW)*IZ*COS(YAW)*COSCOS(ROLL)
-D(ROLL)*D(YAW)*IX*COS(YAW)*COSCOS(ROLL)*COSCOS(PITCH)
+D(ROLL)*D(YAW)*IY*COS(YAW)*COSCOS(ROLL)*COSCOS(PITCH)
-D(ROLL)*D(YAW)*IX*COS(YAW)*SINSIN(ROLL)
-D(ROLL)*D(YAW)*IY*COS(YAW)*SINSIN(ROLL)
```

```
+D(ROLL)*D(YAW)*IZ*COS(YAW)*SINSIN(ROLL)
+2*D(PITCH)*D(YAW)*IXY*COS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
-4*D(PITCH)*D(YAW)*IXY*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(YAW)*IXY*COS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
+2*D(PITCH)*D(YAW)*IXZ*COS(YAW)*SINSIN(ROLL)*COS(PITCH)
+2*D(PITCH)*D(YAW)*IXZ*SIN(YAW)*SIN(ROLL)*SIN(PITCH)
-2*D(PITCH)*D(YAW)*IYZ*COS(YAW)*SINSIN(ROLL)*SIN(PITCH)
+2*D(PITCH)*D(YAW)*IYZ*SIN(YAW)*SIN(ROLL)*COS(PITCH)
+2*D(PITCH)*D(YAW)*IX*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(YAW)*IY*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
+D(PITCH)*D(YAW)*IX*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-D(PITCH)*D(YAW)*IY*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-D(PITCH)*D(YAW)*IZ*SIN(YAW)*COS(ROLL)
-D(PITCH)*D(YAW)*IX*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
+D(PITCH)*D(YAW)*IY*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
-DD(ROLL)*IXY*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-2*DD(ROLL)*IXY*SIN(YAW)*SINCOS(PITCH)
+DD(ROLL)*IXY*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(ROLL)*IXY*COS(YAW)*COS(ROLL)*COSCOS(PITCH)
+D(ROLL)*D(ROLL)*IXY*COS(YAW)*COS(ROLL)*SINSIN(PITCH)
-DD(ROLL)*IXZ*COS(YAW)*COS(ROLL)*COS(PITCH)
+D(ROLL)*D(ROLL)*IXZ*COS(YAW)*SIN(ROLL)*COS(PITCH)
+DD(ROLL)*IYZ*COS(YAW)*COS(ROLL)*SIN(PITCH)
-D(ROLL)*D(ROLL)*IYZ*COS(YAW)*SIN(ROLL)*SIN(PITCH)
+DD(ROLL)*IX*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-DD(ROLL)*IY*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-DD(ROLL)*IX*SIN(YAW)*COSCOS(PITCH)
-DD(ROLL)*IY*SIN(YAW)*SINSIN(PITCH)
+D(ROLL)*D(ROLL)*IX*COS(YAW)*COS(ROLL)*SINCOS(PITCH)
-D(ROLL)*D(ROLL)*IY*COS(YAW)*COS(ROLL)*SINCOS(PITCH)
+4*D(PITCH)*D(ROLL)*IXY*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(ROLL)*IXY*SIN(YAW)*SINSIN(PITCH)
-2*D(PITCH)*D(ROLL)*IXY*SIN(YAW)*COSCOS(PITCH)
+D(PITCH)*D(ROLL)*IX*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(ROLL)*IY*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(ROLL)*IZ*COS(YAW)*SIN(ROLL)
+2*D(PITCH)*D(ROLL)*IX*SIN(YAW)*SINCOS(PITCH)
-2*D(PITCH)*D(ROLL)*IY*SIN(YAW)*SINCOS(PITCH)
-D(PITCH)*D(ROLL)*IX*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(ROLL)*IY*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-DD(PITCH)*IXZ*COS(YAW)*SIN(ROLL)*SIN(PITCH)
+DD(PITCH)*IXZ*SIN(YAW)*COS(PITCH)
-D(PITCH)*D(PITCH)*IXZ*COS(YAW)*SIN(ROLL)*COS(PITCH)
-D(PITCH)*D(PITCH)*IXZ*SIN(YAW)*SIN(PITCH)
-DD(PITCH)*IYZ*COS(YAW)*SIN(ROLL)*COS(PITCH)
-DD(PITCH)*IYZ*SIN(YAW)*SIN(PITCH)
+D(PITCH)*D(PITCH)*IYZ*COS(YAW)*SIN(ROLL)*SIN(PITCH)
-D(PITCH)*D(PITCH)*IYZ*SIN(YAW)*COS(PITCH)
+DD(PITCH)*IZ*COS(YAW)*COS(ROLL)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE YAW

```
0=
+DD(THETA)*IXY*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
-2*DD(THETA)*IXY*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-DD(THETA)*IXY*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-2*D(THETA)*D(THETA)*IXY*SINCOS(YAW)*SINCOS(PITCH)
+D(THETA)*D(THETA)*IXY*SINSIN(YAW)*SIN(ROLL)*COSCOS(PITCH)
```

```
+D(THETA)*D(THETA)*IXY*COSCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(THETA)*D(THETA)*IXY*SINCOS(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IXY*SINSIN(YAW)*SIN(ROLL)*SINSIN(PITCH)
-D(THETA)*D(THETA)*IXY*COSCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-DD(THETA)*IXZ*SIN(YAW)*SIN(ROLL)*COS(PITCH)
+DD(THETA)*IXZ*COS(YAW)*SINSIN(ROLL)*SIN(PITCH)
-DD(THETA)*IXZ*COS(YAW)*COSCOS(ROLL)*SIN(PITCH)
+D(THETA)*D(THETA)*IXZ*SINSIN(YAW)*COS(ROLL)*COS(PITCH)
-2*D(THETA)*D(THETA)*IXZ*SINCOS(YAW)*SINCOS(ROLL)*SIN(PITCH)
-D(THETA)*D(THETA)*IXZ*COSCOS(YAW)*COS(ROLL)*COS(PITCH)
+DD(THETA)*IYZ*SIN(YAW)*SIN(ROLL)*SIN(PITCH)
+DD(THETA)*IYZ*COS(YAW)*SINSIN(ROLL)*COS(PITCH)
-DD(THETA)*IYZ*COS(YAW)*COSCOS(ROLL)*COS(PITCH)
-D(THETA)*D(THETA)*IYZ*SINSIN(YAW)*COS(ROLL)*SIN(PITCH)
-2*D(THETA)*D(THETA)*IYZ*SINCOS(YAW)*SINCOS(ROLL)*COS(PITCH)
+D(THETA)*D(THETA)*IYZ*COSCOS(YAW)*COS(ROLL)*SIN(PITCH)
-DD(THETA)*IX*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
+DD(THETA)*IY*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
+DD(THETA)*IX*COS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
+DD(THETA)*IY*COS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
-DD(THETA)*IZ*COS(YAW)*SINCOS(ROLL)
-0.5*D(THETA)*D(THETA)*IX*SINCOS(YAW)*COSCOS(PITCH)
+0.5*D(THETA)*D(THETA)*IY*SINCOS(YAW)*COSCOS(PITCH)
-D(THETA)*D(THETA)*IX*SINSIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
+D(THETA)*D(THETA)*IY*SINSIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
+D(THETA)*D(THETA)*IX*COSCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IY*COSCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+0.5*D(THETA)*D(THETA)*IX*SINCOS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
-0.5*D(THETA)*D(THETA)*IY*SINCOS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+D(THETA)*D(THETA)*IZ*SINCOS(YAW)*COSCOS(ROLL)
+0.5*D(THETA)*D(THETA)*IX*SINCOS(YAW)*SINSIN(PITCH)
-0.5*D(THETA)*D(THETA)*IY*SINCOS(YAW)*SINSIN(PITCH)
-0.5*D(THETA)*D(THETA)*IX*SINCOS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+0.5*D(THETA)*D(THETA)*IY*SINCOS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-0.5*D(THETA)*D(THETA)*IX*SINCOS(YAW)*COSCOS(ROLL)
-0.5*D(THETA)*D(THETA)*IY*SINCOS(YAW)*COSCOS(ROLL)
-2*D(ROLL)*D(THETA)*IXY*SIN(YAW)*SIN(ROLL)*COSCOS(PITCH)
+4*D(ROLL)*D(THETA)*IXY*COS(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
+2*D(ROLL)*D(THETA)*IXY*SIN(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*D(ROLL)*D(THETA)*IXZ*SIN(YAW)*COS(ROLL)*COS(PITCH)
+4*D(ROLL)*D(THETA)*IXZ*COS(YAW)*SINCOS(ROLL)*SIN(PITCH)
+2*D(ROLL)*D(THETA)*IYZ*SIN(YAW)*COS(ROLL)*SIN(PITCH)
+4*D(ROLL)*D(THETA)*IYZ*COS(YAW)*SINCOS(ROLL)*COS(PITCH)
+2*D(ROLL)*D(THETA)*IX*SIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
-2*D(ROLL)*D(THETA)*IY*SIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
-D(ROLL)*D(THETA)*IX*COS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+D(ROLL)*D(THETA)*IY*COS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
+D(ROLL)*D(THETA)*IZ*COS(YAW)*SINSIN(ROLL)
+D(ROLL)*D(THETA)*IX*COS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-D(ROLL)*D(THETA)*IY*COS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+D(ROLL)*D(THETA)*IX*COS(YAW)*COSCOS(ROLL)
+D(ROLL)*D(THETA)*IY*COS(YAW)*COSCOS(ROLL)
-D(ROLL)*D(THETA)*IZ*COS(YAW)*COSCOS(ROLL)
-4*D(PITCH)*D(THETA)*IXY*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(THETA)*IXY*COS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
-2*D(PITCH)*D(THETA)*IXY*COS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
-2*D(PITCH)*D(THETA)*IXZ*COS(YAW)*COSCOS(ROLL)*COS(PITCH)
+2*D(PITCH)*D(THETA)*IYZ*COS(YAW)*COSCOS(ROLL)*SIN(PITCH)
-D(PITCH)*D(THETA)*IX*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
+D(PITCH)*D(THETA)*IY*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
```

```
+D(PITCH)*D(THETA)*IZ*SIN(YAW)*COS(ROLL)
+2*D(PITCH)*D(THETA)*IX*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(THETA)*IY*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
+D(PITCH)*D(THETA)*IX*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-D(PITCH)*D(THETA)*IY*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-2*DD(YAW)*IXY*COSCOS(ROLL)*SINCOS(PITCH)
+2*DD(YAW)*IXZ*SINCOS(ROLL)*SIN(PITCH)
+2*DD(YAW)*IYZ*SINCOS(ROLL)*COS(PITCH)
+DD(YAW)*IX*COSCOS(ROLL)*SINSIN(PITCH)
+DD(YAW)*IY*COSCOS(ROLL)*COSCOS(PITCH)
+DD(YAW)*IZ*SINSIN(ROLL)
+4*D(ROLL)*D(YAW)*IXY*SINCOS(ROLL)*SINCOS(PITCH)
-2*D(ROLL)*D(YAW)*IXZ*SINSIN(ROLL)*SIN(PITCH)
+2*D(ROLL)*D(YAW)*IXZ*COSCOS(ROLL)*SIN(PITCH)
-2*D(ROLL)*D(YAW)*IYZ*SINSIN(ROLL)*COS(PITCH)
+2*D(ROLL)*D(YAW)*IYZ*COSCOS(ROLL)*COS(PITCH)
-2*D(ROLL)*D(YAW)*IX*SINCOS(ROLL)*SINSIN(PITCH)
-2*D(ROLL)*D(YAW)*IY*SINCOS(ROLL)*COSCOS(PITCH)
+2*D(ROLL)*D(YAW)*IZ*SINCOS(ROLL)
+2*D(PITCH)*D(YAW)*IXY*COSCOS(ROLL)*SINSIN(PITCH)
-2*D(PITCH)*D(YAW)*IXY*COSCOS(ROLL)*COSCOS(PITCH)
+2*D(PITCH)*D(YAW)*IXZ*SINCOS(ROLL)*COS(PITCH)
-2*D(PITCH)*D(YAW)*IYZ*SINCOS(ROLL)*SIN(PITCH)
+2*D(PITCH)*D(YAW)*IX*COSCOS(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(YAW)*IY*COSCOS(ROLL)*SINCOS(PITCH)
-DD(ROLL)*IXY*COS(ROLL)*COSCOS(PITCH)
+DD(ROLL)*IXY*COS(ROLL)*SINSIN(PITCH)
+D(ROLL)*D(ROLL)*IXY*SIN(ROLL)*COSCOS(PITCH)
-D(ROLL)*D(ROLL)*IXY*SIN(ROLL)*SINSIN(PITCH)
+DD(ROLL)*IXZ*SIN(ROLL)*COS(PITCH)
+D(ROLL)*D(ROLL)*IXZ*COS(ROLL)*COS(PITCH)
-DD(ROLL)*IYZ*SIN(ROLL)*SIN(PITCH)
-D(ROLL)*D(ROLL)*IYZ*COS(ROLL)*SIN(PITCH)
+DD(ROLL)*IX*COS(ROLL)*SINCOS(PITCH)
-DD(ROLL)*IY*COS(ROLL)*SINCOS(PITCH)
-D(ROLL)*D(ROLL)*IX*SIN(ROLL)*SINCOS(PITCH)
+D(ROLL)*D(ROLL)*IY*SIN(ROLL)*SINCOS(PITCH)
+4*D(PITCH)*D(ROLL)*IXY*COS(ROLL)*SINCOS(PITCH)
+D(PITCH)*D(ROLL)*IX*COS(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(ROLL)*IY*COS(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(ROLL)*IZ*COS(ROLL)
-D(PITCH)*D(ROLL)*IX*COS(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(ROLL)*IY*COS(ROLL)*SINSIN(PITCH)
-DD(PITCH)*IXZ*COS(ROLL)*SIN(PITCH)
-D(PITCH)*D(PITCH)*IXZ*COS(ROLL)*COS(PITCH)
-DD(PITCH)*IYZ*COS(ROLL)*COS(PITCH)
+D(PITCH)*D(PITCH)*IYZ*COS(ROLL)*SIN(PITCH)
-DD(PITCH)*IZ*SIN(ROLL)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE ROLL

```
0=
+DD(THETA)*IXY*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-2*DD(THETA)*IXY*SIN(YAW)*SINCOS(PITCH)
-DD(THETA)*IXY*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+D(THETA)*D(THETA)*IXY*SINCOS(YAW)*COS(ROLL)*SINSIN(PITCH)
+2*D(THETA)*D(THETA)*IXY*COSCOS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IXY*SINCOS(YAW)*COS(ROLL)*COSCOS(PITCH)
```

```
-DD(THETA)*IXZ*COS(YAW)*COS(ROLL)*COS(PITCH)
+D(THETA)*D(THETA)*IXZ*COSCOS(YAW)*COSCOS(ROLL)*SIN(PITCH)
+D(THETA)*D(THETA)*IXZ*SINCOS(YAW)*SIN(ROLL)*COS(PITCH)
-D(THETA)*D(THETA)*IXZ*COSCOS(YAW)*SINSIN(ROLL)*SIN(PITCH)
+DD(THETA)*IYZ*COS(YAW)*COS(ROLL)*SIN(PITCH)
+D(THETA)*D(THETA)*IYZ*COSCOS(YAW)*COSCOS(ROLL)*COS(PITCH)
-D(THETA)*D(THETA)*IYZ*SINCOS(YAW)*SIN(ROLL)*SIN(PITCH)
-D(THETA)*D(THETA)*IYZ*COSCOS(YAW)*SINSIN(ROLL)*COS(PITCH)
+DD(THETA)*IX*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-DD(THETA)*IY*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-DD(THETA)*IX*SIN(YAW)*COSCOS(PITCH)
-DD(THETA)*IY*SIN(YAW)*SINSIN(PITCH)
+D(THETA)*D(THETA)*IX*SINCOS(YAW)*COS(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IY*SINCOS(YAW)*COS(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IX*COSCOS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
-D(THETA)*D(THETA)*IY*COSCOS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
+D(THETA)*D(THETA)*IZ*COSCOS(YAW)*SINCOS(ROLL)
-4*D(YAW)*D(THETA)*IXY*COS(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
-2*D(YAW)*D(THETA)*IXY*SIN(YAW)*SIN(ROLL)*SINSIN(PITCH)
+2*D(YAW)*D(THETA)*IXY*SIN(YAW)*SIN(ROLL)*COSCOS(PITCH)
-4*D(YAW)*D(THETA)*IXZ*COS(YAW)*SINCOS(ROLL)*SIN(PITCH)
+2*D(YAW)*D(THETA)*IXZ*SIN(YAW)*COS(ROLL)*COS(PITCH)
-4*D(YAW)*D(THETA)*IYZ*COS(YAW)*SINCOS(ROLL)*COS(PITCH)
-2*D(YAW)*D(THETA)*IYZ*SIN(YAW)*COS(ROLL)*SIN(PITCH)
+D(YAW)*D(THETA)*IX*COS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
-D(YAW)*D(THETA)*IY*COS(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
-D(YAW)*D(THETA)*IZ*COS(YAW)*SINSIN(ROLL)
-2*D(YAW)*D(THETA)*IX*SIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*D(YAW)*D(THETA)*IY*SIN(YAW)*SIN(ROLL)*SINCOS(PITCH)
-D(YAW)*D(THETA)*IX*COS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+D(YAW)*D(THETA)*IY*COS(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IX*COS(YAW)*COSCOS(ROLL)
-D(YAW)*D(THETA)*IY*COS(YAW)*COSCOS(ROLL)
+D(YAW)*D(THETA)*IZ*COS(YAW)*COSCOS(ROLL)
+4*D(PITCH)*D(THETA)*IXY*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*D(PITCH)*D(THETA)*IXY*SIN(YAW)*SINSIN(PITCH)
-2*D(PITCH)*D(THETA)*IXY*SIN(YAW)*COSCOS(PITCH)
+2*D(PITCH)*D(THETA)*IXZ*COS(YAW)*COS(ROLL)*SIN(PITCH)
+2*D(PITCH)*D(THETA)*IYZ*COS(YAW)*COS(ROLL)*COS(PITCH)
-D(PITCH)*D(THETA)*IX*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(THETA)*IY*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(THETA)*IZ*COS(YAW)*SIN(ROLL)
+2*D(PITCH)*D(THETA)*IX*SIN(YAW)*SINCOS(PITCH)
-2*D(PITCH)*D(THETA)*IY*SIN(YAW)*SINCOS(PITCH)
+D(PITCH)*D(THETA)*IX*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(THETA)*IY*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+DD(YAW)*IXY*COS(ROLL)*SINSIN(PITCH)
-DD(YAW)*IXY*COS(ROLL)*COSCOS(PITCH)
-2*D(YAW)*D(YAW)*IXY*SINCOS(ROLL)*SINCOS(PITCH)
+DD(YAW)*IXZ*SIN(ROLL)*COS(PITCH)
-D(YAW)*D(YAW)*IXZ*COSCOS(ROLL)*SIN(PITCH)
+D(YAW)*D(YAW)*IXZ*SINSIN(ROLL)*SIN(PITCH)
-DD(YAW)*IYZ*SIN(ROLL)*SIN(PITCH)
-D(YAW)*D(YAW)*IYZ*COSCOS(ROLL)*COS(PITCH)
+D(YAW)*D(YAW)*IYZ*SINSIN(ROLL)*COS(PITCH)
+DD(YAW)*IX*COS(ROLL)*SINCOS(PITCH)
-DD(YAW)*IY*COS(ROLL)*SINCOS(PITCH)
+D(YAW)*D(YAW)*IX*SINCOS(ROLL)*SINSIN(PITCH)
+D(YAW)*D(YAW)*IY*SINCOS(ROLL)*COSCOS(PITCH)
-D(YAW)*D(YAW)*IZ*SINCOS(ROLL)
```

```
+4*D(PITCH)*D(YAW)*IXY*COS(ROLL)*SINCOS(PITCH)
-2*D(PITCH)*D(YAW)*IXZ*SIN(ROLL)*SIN(PITCH)
-2*D(PITCH)*D(YAW)*IYZ*SIN(ROLL)*COS(PITCH)
-D(PITCH)*D(YAW)*IX*COS(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(YAW)*IY*COS(ROLL)*SINSIN(PITCH)
+D(PITCH)*D(YAW)*IZ*COS(ROLL)
+D(PITCH)*D(YAW)*IX*COS(ROLL)*COSCOS(PITCH)
-D(PITCH)*D(YAW)*IY*COS(ROLL)*COSCOS(PITCH)
+2*DD(ROLL)*IXY*SINCOS(PITCH)
+DD(ROLL)*IX*COSCOS(PITCH)
+DD(ROLL)*IY*SINSIN(PITCH)
-2*D(PITCH)*D(ROLL)*IXY*SINSIN(PITCH)
+2*D(PITCH)*D(ROLL)*IXY*COSCOS(PITCH)
-2*D(PITCH)*D(ROLL)*IX*SINCOS(PITCH)
+2*D(PITCH)*D(ROLL)*IY*SINCOS(PITCH)
-DD(PITCH)*IXZ*COS(PITCH)
+D(PITCH)*D(PITCH)*IXZ*SIN(PITCH)
+DD(PITCH)*IYZ*SIN(PITCH)
+D(PITCH)*D(PITCH)*IYZ*COS(PITCH)
-(AX)*(R)*(K2B/FL)*SIN(ROLL)*SIN(PITCH)
-(AY)*(R)*(K2B/FL)*SIN(ROLL)*COS(PITCH)
-(AZ)*(R)*(K2B/FL)*COS(ROLL)


THESE ARE THE EQUATIONS FOR THE VARIABLE PITCH

0=
-D(THETA)*D(THETA)*IXY*COSCOS(YAW)*SINSIN(PITCH)
+4*D(THETA)*D(THETA)*IXY*SINCOS(YAW)*SIN(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IXY*COSCOS(ROLL)*COSCOS(PITCH)
-D(THETA)*D(THETA)*IXY*SINSIN(YAW)*SINSIN(ROLL)*COSCOS(PITCH)
+D(THETA)*D(THETA)*IXY*COSCOS(YAW)*COSCOS(PITCH)
+D(THETA)*D(THETA)*IXY*COSCOS(ROLL)*SINSIN(PITCH)
+D(THETA)*D(THETA)*IXY*SINSIN(YAW)*SINSIN(ROLL)*SINSIN(PITCH)
-DD(THETA)*IXZ*COS(YAW)*SIN(ROLL)*SIN(PITCH)
+DD(THETA)*IXZ*SIN(YAW)*COS(PITCH)
+D(THETA)*D(THETA)*IXZ*SINCOS(YAW)*COS(ROLL)*SIN(PITCH)
+D(THETA)*D(THETA)*IXZ*COSCOS(YAW)*SINCOS(ROLL)*COS(PITCH)
-DD(THETA)*IYZ*COS(YAW)*SIN(ROLL)*COS(PITCH)
-DD(THETA)*IYZ*SIN(YAW)*SIN(PITCH)
+D(THETA)*D(THETA)*IYZ*SINCOS(YAW)*COS(ROLL)*COS(PITCH)
-D(THETA)*D(THETA)*IYZ*COSCOS(YAW)*SINCOS(ROLL)*SIN(PITCH)
+DD(THETA)*IZ*COS(YAW)*COS(ROLL)
-D(THETA)*D(THETA)*IX*COSCOS(YAW)*SINCOS(PITCH)
+D(THETA)*D(THETA)*IY*COSCOS(YAW)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IX*SINCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+D(THETA)*D(THETA)*IY*SINCOS(YAW)*SIN(ROLL)*SINSIN(PITCH)
+D(THETA)*D(THETA)*IX*COSCOS(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IY*COSCOS(ROLL)*SINCOS(PITCH)
+D(THETA)*D(THETA)*IX*SINCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
-D(THETA)*D(THETA)*IY*SINCOS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+D(THETA)*D(THETA)*IX*SINSIN(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
-D(THETA)*D(THETA)*IY*SINSIN(YAW)*SINSIN(ROLL)*SINCOS(PITCH)
+2*D(YAW)*D(THETA)*IXY*COS(YAW)*SINCOS(ROLL)*COSCOS(PITCH)
+4*D(YAW)*D(THETA)*IXY*SIN(YAW)*COS(ROLL)*SINCOS(PITCH)
-2*D(YAW)*D(THETA)*IXY*COS(YAW)*SINCOS(ROLL)*SINSIN(PITCH)
+2*D(YAW)*D(THETA)*IXZ*COS(YAW)*COSCOS(ROLL)*COS(PITCH)
-2*D(YAW)*D(THETA)*IYZ*COS(YAW)*COSCOS(ROLL)*SIN(PITCH)
-2*D(YAW)*D(THETA)*IX*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
```

```
+2*D(YAW)*D(THETA)*IY*COS(YAW)*SINCOS(ROLL)*SINCOS(PITCH)
+D(YAW)*D(THETA)*IX*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IY*SIN(YAW)*COS(ROLL)*COSCOS(PITCH)
-D(YAW)*D(THETA)*IZ*SIN(YAW)*COS(ROLL)
-D(YAW)*D(THETA)*IX*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
+D(YAW)*D(THETA)*IY*SIN(YAW)*COS(ROLL)*SINSIN(PITCH)
-4*D(ROLL)*D(THETA)*IXY*COS(YAW)*SIN(ROLL)*SINCOS(PITCH)
+2*D(ROLL)*D(THETA)*IXY*SIN(YAW)*COSCOS(PITCH)
-2*D(ROLL)*D(THETA)*IXY*SIN(YAW)*SINSIN(PITCH)
-2*D(ROLL)*D(THETA)*IXZ*COS(YAW)*COS(ROLL)*SIN(PITCH)
-2*D(ROLL)*D(THETA)*IYZ*COS(YAW)*COS(ROLL)*COS(PITCH)
+D(ROLL)*D(THETA)*IX*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(THETA)*IY*COS(YAW)*SIN(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(THETA)*IZ*COS(YAW)*SIN(ROLL)
-2*D(ROLL)*D(THETA)*IX*SIN(YAW)*SINCOS(PITCH)
+2*D(ROLL)*D(THETA)*IY*SIN(YAW)*SINCOS(PITCH)
-D(ROLL)*D(THETA)*IX*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+D(ROLL)*D(THETA)*IY*COS(YAW)*SIN(ROLL)*COSCOS(PITCH)
+D(YAW)*D(YAW)*IXY*COSCOS(ROLL)*COSCOS(PITCH)
-D(YAW)*D(YAW)*IXY*COSCOS(ROLL)*SINSIN(PITCH)
-DD(YAW)*IXZ*COS(ROLL)*SIN(PITCH)
-D(YAW)*D(YAW)*IXZ*SINCOS(ROLL)*COS(PITCH)
-DD(YAW)*IYZ*COS(ROLL)*COS(PITCH)
+D(YAW)*D(YAW)*IYZ*SINCOS(ROLL)*SIN(PITCH)
-DD(YAW)*IZ*SIN(ROLL)
-D(YAW)*D(YAW)*IX*COSCOS(ROLL)*SINCOS(PITCH)
+D(YAW)*D(YAW)*IY*COSCOS(ROLL)*SINCOS(PITCH)
-4*D(ROLL)*D(YAW)*IXY*COS(ROLL)*SINCOS(PITCH)
+2*D(ROLL)*D(YAW)*IXZ*SIN(ROLL)*SIN(PITCH)
+2*D(ROLL)*D(YAW)*IYZ*SIN(ROLL)*COS(PITCH)
+D(ROLL)*D(YAW)*IX*COS(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(YAW)*IY*COS(ROLL)*SINSIN(PITCH)
-D(ROLL)*D(YAW)*IZ*COS(ROLL)
-D(ROLL)*D(YAW)*IX*COS(ROLL)*COSCOS(PITCH)
+D(ROLL)*D(YAW)*IY*COS(ROLL)*COSCOS(PITCH)
-D(ROLL)*D(ROLL)*IXY*COSCOS(PITCH)
+D(ROLL)*D(ROLL)*IXY*SINSIN(PITCH)
-DD(ROLL)*IXZ*COS(PITCH)
+DD(ROLL)*IYZ*SIN(PITCH)
+D(ROLL)*D(ROLL)*IX*SINCOS(PITCH)
-D(ROLL)*D(ROLL)*IY*SINCOS(PITCH)
+DD(PITCH)*IZ
+(AX)*(R)*(K2B/FL)*COS(ROLL)*COS(PITCH)
-(AY)*(R)*(K2B/FL)*COS(ROLL)*SIN(PITCH)
```

CASE 14

EQUATIONS OF A SATELLITE WHOSEATTITUDE WITH RESPECT TO ROTATING COORDINATE IS IN TERMS OF NON-HOLONOMIC VARIABLES.

OMX,OMY,OMZ ARE THE ROTATIONALRATES ABOUT THE BODY X,Y,Z AXES.
DCOS IS THE DIRECTION COSINE DESIGNATION BETWEEN THE X,Y,Z BODY AXES AND THE X1,Y1,Z1 ROTATING SYSTEM.
R AND THETA ARE THE POLAR COORDINATES OF THE CENTER OF MASS OF THE BODY AS IT ROTATES ABOUT THE EARTH IN AN ORBITAL PLANE.
X1,Y1,Z1 IS A ROTATING COORDINATE SYSTEM WHOSE ORIGIN IS AT THE CENTER OF MASS OF THE BODY. Z1 IS DIRECTLY AWAY FROM THE CENTER OF THE EARTH, X1 IS IN THE ORBITAL PLANE.

```
MASS     Y OOV   D CCO  U AAA
         THETA   DCOS
 IXY             OMX     AX
 IXZ             OMY     AY
 IYZ       R     OMZ     AZ
FGRAV    U OOO   Y OOV  D OCO   U AAA
 KGRAV           THETA  DCOS
 1,1                     OMX     AX
 1,1                     OMY     AY
 2,4               R     OMZ     AZ
```

THESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS SET.

```
K2B/FL=(KGRAV)/((+(R)*(R)+DCOS(1,1)*DCOS(1,1)*(AX)*(AX)+DCOS(2,1)*DCOS(2,1)*(AX)*(AX)+DCOS(3,1)*DCOS(3,1)*(AX)*(AX)
       +2*DCOS(1,2)*DCOS(1,1)*(AX)*(AY)+2*DCOS(2,2)*DCOS(2,1)*(AX)*(AY)+2*DCOS(3,2)*DCOS(3,1)*(AX)*(AY)
       +2*DCOS(1,3)*DCOS(1,1)*(AX)*(AZ)+2*DCOS(2,3)*DCOS(2,1)*(AX)*(AZ)+2*DCOS(3,3)*DCOS(3,1)*(AX)*(AZ)
       +DCOS(1,2)*DCOS(1,2)*(AY)*(AY)+DCOS(2,2)*DCOS(2,2)*(AY)*(AY)+DCOS(3,2)*DCOS(3,2)*(AY)*(AY)
       +2*DCOS(1,3)*DCOS(1,2)*(AY)*(AZ)+2*DCOS(2,3)*DCOS(2,2)*(AY)*(AZ)+2*DCOS(3,3)*DCOS(3,2)*(AY)*(AZ)
       +DCOS(1,3)*DCOS(1,3)*(AZ)*(AZ)+DCOS(2,3)*DCOS(2,3)*(AZ)*(AZ)+DCOS(3,3)*DCOS(3,3)*(AZ)*(AZ)
       )**1.5)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE R

```
0=
+DD(R)*MASS
-(R)*D(THETA)*D(THETA)*MASS
+(R)*(K2B/FL)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE THETA

```
0=
```

```
+2*(R)*D(THETA)*D(R)*MASS
+(R)*(R)*DD(THETA)*MASS
+2*DD(THETA)*IXY*DCOS(1,2)*DCOS(1,1)
+2*DD(THETA)*IXY*DCOS(3,2)*DCOS(3,1)
+2*DD(THETA)*IXZ*DCOS(1,3)*DCOS(1,1)
+2*DD(THETA)*IXZ*DCOS(3,3)*DCOS(3,1)
+2*DD(THETA)*IYZ*DCOS(1,3)*DCOS(1,2)
+2*DD(THETA)*IYZ*DCOS(3,3)*DCOS(3,2)
-0.5*DD(THETA)*IX*DCOS(1,1)*DCOS(1,1)
+0.5*DD(THETA)*IY*DCOS(1,1)*DCOS(1,1)
+0.5*DD(THETA)*IZ*DCOS(1,1)*DCOS(1,1)
-0.5*DD(THETA)*IX*DCOS(3,1)*DCOS(3,1)
+0.5*DD(THETA)*IY*DCOS(3,1)*DCOS(3,1)
+0.5*DD(THETA)*IZ*DCOS(3,1)*DCOS(3,1)
+0.5*DD(THETA)*IX*DCOS(1,2)*DCOS(1,2)
-0.5*DD(THETA)*IY*DCOS(1,2)*DCOS(1,2)
+0.5*DD(THETA)*IZ*DCOS(1,2)*DCOS(1,2)
+0.5*DD(THETA)*IX*DCOS(3,2)*DCOS(3,2)
-0.5*DD(THETA)*IY*DCOS(3,2)*DCOS(3,2)
+0.5*DD(THETA)*IZ*DCOS(3,2)*DCOS(3,2)
+0.5*DD(THETA)*IX*DCOS(1,3)*DCOS(1,3)
+0.5*DD(THETA)*IY*DCOS(1,3)*DCOS(1,3)
-0.5*DD(THETA)*IZ*DCOS(1,3)*DCOS(1,3)
+0.5*DD(THETA)*IX*DCOS(3,3)*DCOS(3,3)
+0.5*DD(THETA)*IY*DCOS(3,3)*DCOS(3,3)
-0.5*DD(THETA)*IZ*DCOS(3,3)*DCOS(3,3)
+2*(OMX)*D(THETA)*IXY*DCOS(1,3)*DCOS(1,1)
+2*(OMX)*D(THETA)*IXY*DCOS(3,3)*DCOS(3,1)
-2*(OMX)*D(THETA)*IXZ*DCOS(1,2)*DCOS(1,1)
-2*(OMX)*D(THETA)*IXZ*DCOS(3,2)*DCOS(3,1)
-2*(OMX)*D(THETA)*IYZ*DCOS(1,2)*DCOS(1,2)
-2*(OMX)*D(THETA)*IYZ*DCOS(3,2)*DCOS(3,2)
+2*(OMX)*D(THETA)*IYZ*DCOS(1,3)*DCOS(1,3)
+2*(OMX)*D(THETA)*IYZ*DCOS(3,3)*DCOS(3,3)
-2*(OMX)*D(THETA)*IY*DCOS(1,3)*DCOS(1,2)
+2*(OMX)*D(THETA)*IZ*DCOS(1,3)*DCOS(1,2)
-2*(OMX)*D(THETA)*IY*DCOS(3,3)*DCOS(3,2)
+2*(OMX)*D(THETA)*IZ*DCOS(3,3)*DCOS(3,2)
-2*(OMY)*D(THETA)*IXY*DCOS(1,3)*DCOS(1,2)
-2*(OMY)*D(THETA)*IXY*DCOS(3,3)*DCOS(3,2)
+2*(OMY)*D(THETA)*IXZ*DCOS(1,1)*DCOS(1,1)
+2*(OMY)*D(THETA)*IXZ*DCOS(3,1)*DCOS(3,1)
-2*(OMY)*D(THETA)*IXZ*DCOS(1,3)*DCOS(1,3)
-2*(OMY)*D(THETA)*IXZ*DCOS(3,3)*DCOS(3,3)
+2*(OMY)*D(THETA)*IYZ*DCOS(1,2)*DCOS(1,1)
+2*(OMY)*D(THETA)*IYZ*DCOS(3,2)*DCOS(3,1)
+2*(OMY)*D(THETA)*IX*DCOS(1,3)*DCOS(1,1)
-2*(OMY)*D(THETA)*IZ*DCOS(1,3)*DCOS(1,1)
+2*(OMY)*D(THETA)*IX*DCOS(3,3)*DCOS(3,1)
-2*(OMY)*D(THETA)*IZ*DCOS(3,3)*DCOS(3,1)
-2*(OMZ)*D(THETA)*IXY*DCOS(1,1)*DCOS(1,1)
-2*(OMZ)*D(THETA)*IXY*DCOS(3,1)*DCOS(3,1)
+2*(OMZ)*D(THETA)*IXY*DCOS(1,2)*DCOS(1,2)
+2*(OMZ)*D(THETA)*IXY*DCOS(3,2)*DCOS(3,2)
+2*(OMZ)*D(THETA)*IXZ*DCOS(1,3)*DCOS(1,2)
+2*(OMZ)*D(THETA)*IXZ*DCOS(3,3)*DCOS(3,2)
-2*(OMZ)*D(THETA)*IYZ*DCOS(1,3)*DCOS(1,1)
-2*(OMZ)*D(THETA)*IYZ*DCOS(3,3)*DCOS(3,1)
-2*(OMZ)*D(THETA)*IX*DCOS(1,2)*DCOS(1,1)
+2*(OMZ)*D(THETA)*IY*DCOS(1,2)*DCOS(1,1)
```

```
-2*(OMZ)*D(THETA)*IX*DCOS(3,2)*DCOS(3,1)
+2*(OMZ)*D(THETA)*IY*DCOS(3,2)*DCOS(3,1)
-D(OMX)*IXY*DCOS(3,3)*DCOS(1,1)
+D(OMX)*IXY*DCOS(3,1)*DCOS(1,3)
+(OMX)*(OMX)*IXY*DCOS(3,2)*DCOS(1,1)
-(OMX)*(OMX)*IXY*DCOS(3,1)*DCOS(1,2)
+D(OMX)*IXZ*DCOS(3,2)*DCOS(1,1)
-D(OMX)*IXZ*DCOS(3,1)*DCOS(1,2)
+(OMX)*(OMX)*IXZ*DCOS(3,3)*DCOS(1,1)
-(OMX)*(OMX)*IXZ*DCOS(3,1)*DCOS(1,3)
-D(OMX)*IX*DCOS(3,3)*DCOS(1,2)
+D(OMX)*IX*DCOS(3,2)*DCOS(1,3)
-2*(OMY)*(OMX)*IXZ*DCOS(3,2)*DCOS(1,3)
+2*(OMY)*(OMX)*IXZ*DCOS(3,3)*DCOS(1,2)
+(OMY)*(OMX)*IX*DCOS(3,2)*DCOS(1,1)
-(OMY)*(OMX)*IY*DCOS(3,2)*DCOS(1,1)
-(OMY)*(OMX)*IZ*DCOS(3,2)*DCOS(1,1)
-(OMY)*(OMX)*IX*DCOS(3,1)*DCOS(1,2)
+(OMY)*(OMX)*IY*DCOS(3,1)*DCOS(1,2)
+(OMY)*(OMX)*IZ*DCOS(3,1)*DCOS(1,2)
-2*(OMZ)*(OMX)*IXY*DCOS(3,3)*DCOS(1,2)
+2*(OMZ)*(OMX)*IXY*DCOS(3,2)*DCOS(1,3)
+(OMZ)*(OMX)*IX*DCOS(3,3)*DCOS(1,1)
-(OMZ)*(OMX)*IY*DCOS(3,3)*DCOS(1,1)
-(OMZ)*(OMX)*IZ*DCOS(3,3)*DCOS(1,1)
-(OMZ)*(OMX)*IX*DCOS(3,1)*DCOS(1,3)
+(OMZ)*(OMX)*IY*DCOS(3,1)*DCOS(1,3)
+(OMZ)*(OMX)*IZ*DCOS(3,1)*DCOS(1,3)
+D(OMY)*IXY*DCOS(3,3)*DCOS(1,2)
-D(OMY)*IXY*DCOS(3,2)*DCOS(1,3)
+(OMY)*(OMY)*IXY*DCOS(3,1)*DCOS(1,2)
-(OMY)*(OMY)*IXY*DCOS(3,2)*DCOS(1,1)
-D(OMY)*IYZ*DCOS(3,1)*DCOS(1,2)
+D(OMY)*IYZ*DCOS(3,2)*DCOS(1,1)
+(OMY)*(OMY)*IYZ*DCOS(3,3)*DCOS(1,2)
-(OMY)*(OMY)*IYZ*DCOS(3,2)*DCOS(1,3)
+D(OMY)*IY*DCOS(3,3)*DCOS(1,1)
-D(OMY)*IY*DCOS(3,1)*DCOS(1,3)
-2*(OMZ)*(OMY)*IXY*DCOS(3,3)*DCOS(1,1)
+2*(OMZ)*(OMY)*IXY*DCOS(3,1)*DCOS(1,3)
-(OMZ)*(OMY)*IX*DCOS(3,3)*DCOS(1,2)
+(OMZ)*(OMY)*IY*DCOS(3,3)*DCOS(1,2)
-(OMZ)*(OMY)*IZ*DCOS(3,3)*DCOS(1,2)
+(OMZ)*(OMY)*IX*DCOS(3,2)*DCOS(1,3)
-(OMZ)*(OMY)*IY*DCOS(3,2)*DCOS(1,3)
+(OMZ)*(OMY)*IZ*DCOS(3,2)*DCOS(1,3)
-D(OMZ)*IXZ*DCOS(3,2)*DCOS(1,3)
+D(OMZ)*IXZ*DCOS(3,3)*DCOS(1,2)
+(OMZ)*(OMZ)*IXZ*DCOS(3,1)*DCOS(1,3)
-(OMZ)*(OMZ)*IXZ*DCOS(3,3)*DCOS(1,1)
+D(OMZ)*IYZ*DCOS(3,1)*DCOS(1,3)
-D(OMZ)*IYZ*DCOS(3,3)*DCOS(1,1)
+(OMZ)*(OMZ)*IYZ*DCOS(3,2)*DCOS(1,3)
-(OMZ)*(OMZ)*IYZ*DCOS(3,3)*DCOS(1,2)
-D(OMZ)*IZ*DCOS(3,2)*DCOS(1,1)
+D(OMZ)*IZ*DCOS(3,1)*DCOS(1,2)


THESE ARE THE EQUATIONS FOR THE VARIABLE OMX
```

```
0=
-(OMY)*(OMX)*IXZ
+(OMZ)*(OMX)*IXY
+D(OMX)*IX
-(OMY)*(OMY)*IYZ
-(OMZ)*(OMY)*IY
+(OMZ)*(OMY)*IZ
-D(OMY)*IXY
+(OMZ)*(OMZ)*IYZ
-D(OMZ)*IXZ
+(AY)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,3)
+(AY)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,3)
-(AZ)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,2)
-(AZ)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,2)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE OMY

```
0=
+(OMX)*(OMX)*IXZ
+(OMY)*(OMX)*IYZ
+(OMZ)*(OMX)*IX
-(OMZ)*(OMX)*IZ
-D(OMX)*IXY
-(OMZ)*(OMY)*IXY
+D(OMY)*IY
-(OMZ)*(OMZ)*IXZ
-D(OMZ)*IYZ
-(AX)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,3)
-(AX)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,3)
+(AZ)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,1)
+(AZ)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,1)
```

THESE ARE THE EQUATIONS FOR THE VARIABLE OMZ

```
0=
-(OMX)*(OMX)*IXY
-(OMY)*(OMX)*IX
+(OMY)*(OMX)*IY
-(OMZ)*(OMX)*IYZ
-D(OMX)*IXZ
+(OMY)*(OMY)*IXY
+(OMZ)*(OMY)*IXZ
-D(OMY)*IYZ
+D(OMZ)*IZ
+(AX)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,2)
+(AX)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,2)
-(AY)*(R)*(K2B/FL)*SIN(THETA)*DCOS(1,1)
-(AY)*(R)*(K2B/FL)*COS(THETA)*DCOS(3,1)
```

* END OF CASES

APPENDIX II

COMPUTER PROGRAM LISTING

This appendix contains the FORTRAN IV source decks for this program. To use this program on another machine, the only major change required would be to change the tape numbers of the four work tapes plus input and output tapes. The numbers of these tapes are set in subroutine DATA in the area NWTAPE, with NWTAPE (I), I=1, 2, 3, 4 being the work tapes, NWTAPE (5) the input tape, and NWTAPE (6) the output tape.

Throughout the programming, it was kept in mind that the machine used might not be a 7090; consequently, the FORTRAN IV coding was kept as simple as possible.

```
$IBFTC MAIN     LIST
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),INLINE(80),ROT(33),VARFUN(30),VARLO
     1C(30),VARTIT(6,30),WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(653),INLINE(1)),(CM(773),NOFO
     1NT),(CM(775),NOFVAR),(CM(776),NOMONT),(CM(777),NVAR),(CM(977),PRDC
     2ON),(CM(1146),VARFUN(1)),(CM(1176),VARTIT(1,1)),(CM(1356),WTAPE(1)
     3),(CM(1356),WTAPE1),(CM(1357),WTAPE2),(CM(1358),WTAPE3),(CM(1359),
     4WTAPE4),(CM(1361;,NTAPE6),(CM(1418),NFORTP),(CM(1421),ROT(1)),(CM(
     51663),VARLOC(1))
      INTEGER VARLOC,VARFUN,FORNO
      INTEGER WTAPE,WTAPE1,WTAPE2,WTAPE3,WTAPE4
  200 CALL DATA
  250 WRITE(NTAPE6,5300)
  300 CALL READCK
  400 IF(NVAR.EQ.0)GO TO 5200
  500 DO 5100 NOFVAR=1,NVAR
  600 REWIND WTAPE2
  650 IF(VARFUN(NOFVAR).EQ.15) GO TO 5100
  700 WRITE(NTAPE6,800)(VARTIT(I,NOFVAR),I=1,6)
  800 FORMAT(///39H THIS IS THE EQUATION FOR THE VARIABLE ,6A1//3H 0= )
  900 IF(NOMONT.EQ.0)GO TO 6000
 1000 DO 2200 MASNO=1,NOMONT
 1100 IF(MASNO.GT.4) GO TO 1150
 1110 MNO=MASNO+1
 1115 DO 1120 I=1,325
 1120 COMON(I,1)=COMON(I,MNO)
 1130 GO TO 1300
 1150 READ(WTAPE1)(COMON(I,1),I=1,325)
 1300 IF(VARLOC(NOFVAR).EQ.0)GO TO 2200
 1400 IF(VARFUN(NOFVAR).GE.11)GO TO 1900
 1500 IF(VARFUN(NOFVAR).LE.3)GO TO 2100
 1600 IF(VARFUN(NOFVAR).EQ.10)GO TO 2100
 1700 CALL EXPEQ2
 1800 GO TO 2200
 1900 CALL EXPEQ1
 2000 GO TO 2200
 2100 CALL EXPEQ3
 2200 CONTINUE
 2300 REWIND WTAPE1
 2400 IF(NOFONT.EQ.0)GO TO 5100
 2500 DO 5000 FORNO=1,NOFONT
 2600 IF(FORNO.GT.4) GO TO 2640
 2605 MNO=FORNO+5
 2610 DO 2620 I=1,325
 2620 COMON(I,1)=COMON(I,MNO)
 2630 GO TO 2650
 2640 READ(WTAPE2)(COMON(I,1),I=1,325)
 2650 I=VARFUN(NOFVAR)
 2700 IF(I.LE.3) GO TO 3000
 2800 IF(I.LE.6) GO TO 3500
 2900 IF(I.GT.20) GO TO 4000
 3000 GO TO (3100,3100,3400,3200,3300,3400), NFORTP
 3100 PRDCON=1.0
```

```
3110 CALL FTP3A(NOFVAR,0,1,N,WTAPE(3),7)
3150 GO TO 5000
3200 CALL FTP34
3250 GO TO 5000
3300 CALL FTP35
3350 GO TO 5000
3400 CALL FTP36
3450 GO TO 5000
3500 GO TO (3600,3600,5000,3700,3800,5000),NFORTP
3600 CALL FTP2A(0,NO,1,WTAPE(3),7)
3650 GO TO 5000
3700 CALL FTP24
3750 GO TO 5000
3800 CALL FTP25
3850 GO TO 5000
4000 I=I-20
4100 GO TO (4200,4200,5000,4400,4600,5000) ,NFORTP
4200 CALL FTP1A(I,0,NO)
4300 GO TO 5000
4400 CALL FTP14(I)
4500 GO TO 5000
4600 CALL FTP15(I)
5000 CONTINUE
5100 CONTINUE
5200 WRITE(NTAPE6,5300)INLINE
5300 FORMAT(1H1///1X,80A1)
5900 GO TO 300
6000 WRITE(NTAPE6,6100)
6100 FORMAT(42H WITHOUT MASSES THE PROBLEM IS MEANINGLESS,38X,
    111H***ERROR***)
6200 GO TO 5200
     END
$IBFTC DATA     LIST
      SUBROUTINE DATA
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),COSINE(4),IXY(4),IXZ(4),IY
     1Z(4),MASS(5),NALPH(46),NCOSIN(4),NIXY(4),NIXZ(4),NIYZ(4),NMASS(5),
     2NSINE(4),NWTAPE(6),SINE(4),WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(CM(647),COSINE(1
     1)),(CM(733),IXY(1)),(CM(737),IXZ(1)),(CM(741),IYZ(1)),(CM(746),LOC
     2FAC),(CM(748),MASS(1)),(CM(753),MINUS),(CM(778),ONE),(CM(1142),SIN
     3E(1)),(CM(1356),WTAPE(1))
      INTEGER ONE,WTAPE,ALPH,COSINE,SINE
      DATA(NALPH(I),I=1,46)/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
     1       1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,
     2       1HZ,1H=,1H(,1H),1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,
     3       1H*,1H,,1H.,1H/,1H-,1H+,1H /
C ****** WHEN THE DIRECTION COSINES ARE TESTED CHANGE LOCFAC TO 10 *****
      DATA LOCFA/10/,NONE/0/,MINU/8589934592/,(NWTAPE(I),I=1,6)/
     1     1,2,3,4,5,6/,(NSINE(I),I=1,4)/1HS,1HI,1HN,1H /,
     2     (NCOSIN(I),I=1,4)/1HC,1HO,1HS,1H /,(NMASS(I),I=1,5)/1HM,1HA,
     3       1HS,1HS,1H /
      DATA (NIXY(I),I=1,4)/1HI,1HX,1HY,1H /,
```

```
     1(NIXZ(I),I=1,4)/1HI,1HX,1HZ,1H /,(NIYZ(I),I=1,4)/1HI,1HY,1HZ,1H /
  100 DO 200 I=1,46
  200 ALPH(I)=NALPH(I)
  300 LOCFAC=LOCFA
  400 ONE=NONE
  500 MINUS=MINU
  600 DO 1100 I=1,4
  700 COSINE(I)=NCOSIN(I)
  800 SINE(I)=NSINE(I)
  900 IXY(I)=NIXY(I)
 1000 IXZ(I)=NIXZ(I)
 1100 IYZ(I)=NIYZ(I)
 1200 DO 1300 I=1,5
 1300 MASS(I)=NMASS(I)
 1400 DO 1500 I=1,6
 1500 WTAPE(I)=NWTAPE(I)
      RETURN
      END
$IBFTC READCK  LIST
      SUBROUTINE READCK
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),INLINE(80),OUTWRD(180),WTA
     1PE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(3),LC),(ALP
     1H(6),LF),(ALPH(13),LM),(ALPH(24),LX),(ALPH(40),LAS),(ALPH(46),LBL)
     2,(CM(652),IFEND),(CM(653),INLINE(1)),(CM(745),LINAUX),(CM(772),NER
     3TOT),(CM(773),NOFONT),(CM(774),NOFPS),(CM(776),NOMONT),(CM(777),NV
     4AR),(CM(779),OUTWRD(1)),(CM(959),MORF),(CM(1356),WTAPE(1)),(CM(135
     56),WTAPE1),(CM(1357),WTAPE2),(CM(1359),WTAPE4),(CM(1360),NTAPE5),(
     6CM(1361),NTAPE6),(CM(1380),IFUS),(CM(1429),IFPRNT)
C THE READCK SUBROUTINE READS IN THE INPUT DATA FOR EACH CASE AND
C CHECKS ITS VALIDITY
      INTEGER WTAPE1,WTAPE2,WTAPE4,OUTWRD
  100 CALL ERROR(1,3)
  200 MORF=LX
  230 LINAUX=0
  300 CALL STLIST
  400 NOFONT=0
  500 NOMONT=0
  600 NVAR=0
  700 NERTOT=0
  800 NOFPS=0
  900 IFEND=-1
 1000 READ(NTAPE5,1100)(INLINE(I),I=1,80)
 1100 FORMAT(80A1)
 1200 IF(INLINE(1).EQ.LC)GO TO 4400
 1300 IF(INLINE(1).EQ.LAS)GO TO 4500
 1400 IF(INLINE(1).EQ.LBL)GO TO 3200
 1500 IF(IFEND.LT.0)GO TO 2200
 1600 IF(NERTOT.GT.0)GO TO 2000
 1700 CALL VARCK
 1750 IF(NERTOT.GT.0) GO TO 2000
 1800 CALL STLIST
 1900 GO TO 2200
```

```
2000 REWIND WTAPE1
2100 REWIND WTAPE2
2150 REWIND WTAPE4
2200 IFEND=0
2300 MORF=INLINE(1)
2400 IF(MORF.EQ.LF)GO TO 2800
2500 IF(MORF.EQ.LM)GO TO 3000
2600 CALL ERROR(1,1)
2700 GO TO 5800
2800 CALL FORCK
2900 GO TO 5800
3000 CALL MASSCK
3100 GO TO 5800
3200 IF(IFEND.GE.0)GO TO 3700
3300 WRITE(NTAPE6,3400)
3400 FORMAT(46H CANNOT HAVE TITLE PRIOR TO MASS OR FORCE CARD)
3500 CALL ERROR(81,1)
3600 GO TO 5800
3700 IFEND=IFEND+1
3800 IF(IFEND.LE.4)GO TO 4200
3900 WRITE(NTAPE6,4000)
4000 FORMAT(29H TOO MANY TITLE CARDS PRESENT)
4100 GO TO 3500
4200 CALL STITLE
4300 GO TO 5800
4400 INLINE(1)=LBL
4500 IF(IFEND.EQ.-1)GO TO 5900
4600 IF(NERTOT.GT.0)GO TO 6200
4700 CALL VARCK
4750 IF(NERTOT.GT.0) GO TO 6200
4800 CALL STLIST
5100 REWIND WTAPE1
5135 REWIND WTAPE4
5200 REWIND WTAPE2
5250 NERTOT=1
5300 IF(NOFPS)6500,6500,5350
5350 NERTOT=NERTOT+1
5410 IF(NERTOT.GT.5) GO TO 5440
5420 DO 5425 I=1,325
5425 COMON(I,1)=COMON(I,NERTOT)
5430 GO TO 5450
5440 READ(WTAPE1)(COMON(I,1),I=1,325)
5450 IF(IFPRNT.EQ.0) GO TO 5300
5500 CALL EXPP
5600 NOFPS=NOFPS-1
5700 GO TO 5300
5800 CALL ERROR(1,2)
5900 WRITE(NTAPE6,6000)(INLINE(I),I=1,80)
6000 FORMAT(1H ,80A1)
6100 GO TO 1000
6200 WRITE(NTAPE6,6300)INLINE
6300 FORMAT(1H1,80A1)
6325 REWIND WTAPE1
```

```
 6350 REWIND WTAPE2
 6375 REWIND WTAPE4
 6400 GO TO 200
 6500 REWIND WTAPE1
 6550 IF(LINAUX.GT.0) GO TO 6650
 6600 RETURN
 6650 DO 7100 J=1,LINAUX
 6700 READ(WTAPE4,6800)(OUTWRD(I),I=1,120)
 6800 FORMAT(120A1)
 6900 WRITE(NTAPE6,7000)(OUTWRD(I),I=1,120)
 7000 FORMAT(/1X,120A1)
 7100 CONTINUE
 7150 REWIND WTAPE4
 7200 GO TO 6600
      END
$IBFTC ZEROT   LIST
      SUBROUTINE ZEROT
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(753),MINUS),(CM(979),ROTMAT(1
     1,1,1)),(CM(1141),ROTNUM)
C THE ZEROT SUBROUTINE SETS THE NEXT AVAILABLE ROTMAT TO ALL CODE ZEROS.
      INTEGER ROTNUM,ROTMAT
  100 ROTNUM=ROTNUM+1
  200 DO 300 I=1,9
  300 ROTMAT(I,1,ROTNUM)=-MINUS
  400 RETURN
      END
$IBFTC ERROR   LIST
      SUBROUTINE ERROR(I,J)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),ERROUT(80)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(22),LV),(AL
     1PH(46),LBL),(CM(772),NERTOT),(CM(1361),NTAPE6)
      INTEGER ERROUT
C THE ERROR SUBROUTINE WILL PLACE A 'V' OVER COLUMN(I) OF DATA WHICH IS
C IN ERROR. ALSO ***ERROR*** IS WRITTEN BEGINNING IN COLUMN 80 OF THE
C DATA PRINTOUT. WHEN J EQUALS 1 THE 'V' IS PUT OVER THE INCORRECT
C COLUMN. WHEN J EQUALS 2 ALL OF THE V'S ARE PRINTED. WHEN J EQUALS 3
C THE ARRAY OF ERROR POINTERS IS BLANKED OUT.
  100 GO TO(1000,200,600),J
  200 IF(NOFERR.EQ.0)GO TO 1200
  300 NERTOT=NERTOT+1
  400 WRITE(NTAPE6,500)(ERROUT(N),N=1,80)
  500 FORMAT(1H ,80A1,11H***ERROR***)
  600 DO 700 N=1,80
  700 ERROUT(N)=LBL
  800 NOFERR=0
  900 GO TO 1200
 1000 ERROUT(I)=LV
 1100 NOFERR=NOFERR+1
 1200 RETURN
      END
$IBFTC STLIST  LIST
```

```
      SUBROUTINE STLIST
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),IXYTIT(6,4),ROT(33),TITLES
     1(192)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(6),LF),(ALP
     1H(13),LM),(ALPH(46),LBL),(CM(773),NOFONT),(CM(776),NOMONT),(CM(959
     2),MORF),(CM(1356),WTAPE1),(CM(1357),WTAPE2),(CM(1391),IXYTIT(1,1))
     3,(CM(1421),ROT(1)),(CM(1462),TITLES(1))
      INTEGER WTAPE1,WTAPE2
      INTEGER CM,COMON
      INTEGER ROT,TITLES
  100 IF(MORF.NE.LF.AND.MORF.NE.LM)GO TO 1100
  200 IF(MORF.EQ.LF)GO TO 700
  300 NOMONT=NOMONT+1
  400 IF(NOMONT.LE.4) GO TO 500
  450 WRITE(WTAPE1)(COMON(I,1),I=1,325)
  475 GO TO 1300
  500 NOM=NOMONT+1
  525 DO 550 I=1,325
  550 COMON(I,NOM)=COMON(I,1)
  600 GO TO 1300
  700 NOFONT=NOFONT+1
  725 IF(NOFONT.LE.4) GO TO 800
  750 WRITE(WTAPE2)(COMON(I,1),I=1,325)
  775 GO TO 1300
  800 NOM=NOFONT+5
  900 GO TO 525
 1000 GO TO 1300
 1100 REWIND WTAPE1
 1200 REWIND WTAPE2
 1300 DO 1400 I=1,33
 1400 ROT(I)=0
 1500 DO 1600 I=1,192
 1600 TITLES(I)=LBL
 1625 DO 1650 I=1,24
 1650 IXYTIT(I,1)=LBL
 1700 RETURN
      END
$IBFTC WOTAST  LIST
      SUBROUTINE WOTAST
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),OUTWRD(180)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(40),LAS),(C
     1M(747),LOC1ST),(CM(779),OUTWRD(1))
C THE WOTAST SUBROUTINE STORES A MULTIPLICATION SIGN(*) IT THE POSITION
C LOC1ST OF THE OUTPUT WORD
      INTEGER OUTWRD
  100 OUTWRD(LOC1ST)=LAS
  200 LOC1ST=LOC1ST+1
  300 RETURN
      END
$IBFTC COMPRS  LIST
      SUBROUTINE COMPRS(NTERMS)
```

```
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11))
C THE COMPRS SUBROUTINE COMPRESSES THE LIST OF VARIABLES BY REMOVING THE
C TERMS WHICH HAVE A ZERO COEFFICIENT. THE COUNT OF TERMS IS REDUCED
C ACCORDINGLY.
      INTEGER CODWRD
  100 NT1=0
  200 DO 700 I=1,NTERMS
  300 IF(COEF(I).EQ.0.)GO TO 700
  400 NT1=NT1+1
  500 COEF(NT1)=COEF(I)
  600 CODWRD(NT1)=CODWRD(I)
  700 CONTINUE
  800 NTERMS=NT1
  900 RETURN
      END
$IBFTC NADD
      FUNCTION NADD(I,J)
      INTEGER TWOMIN
      DATA TWOMIN/17179869184/
C THE NADD FUNCTION DOES THE SIMULATED MULTIPLICATION. THE TWOMIN IS
C NECESSARY TO GIVE THE CORRECT SIGN FOR THE RESULTING PRODUCT.
  100 NADD=MOD(I+J,TWOMIN)
  200 RETURN
      END
$IBFTC MODI3   LIST
      FUNCTION MODI3(I)
  100 MODI3=MOD(I,3)+1
  200 RETURN
      END
$IBFTC WOTITL  LIST
      SUBROUTINE WOTITL(NTITLE,NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),OUTWRD(180),NTITLE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(28),LPO),(A
     1LPH(29),LPC),(ALPH(46),LBL),(CM(747),LOC1ST),(CM(779),OUTWRD(1))
C THE WOTITL SUBROUTINE LEFT JUSTIFIES, REMOVES EMBEDDED BLANKS, AND
C PRINTS THE CONTENTS OF NTITLE. IF NTYPE IS 2 THEN NTITLE IS PLACED
C INSIDE PARENTHESES. IF NTYPE IS 1 NO PARENTHESES ARE PRESENT.
      INTEGER OUTWRD
   90 IF(NTITLE(1).EQ.LBL)GO TO 1100
  100 IF(NTYPE.EQ.2)GO TO 400
  200 OUTWRD(LOC1ST)=LPO
  300 LOC1ST=LOC1ST+1
  400 DO 700 N=1,6
  500 IF(NTITLE(N).EQ.LBL)GO TO 800
  600 OUTWRD(LOC1ST)=NTITLE(N)
  700 LOC1ST=LOC1ST+1
  800 IF(NTYPE.EQ.2)GO TO 1100
  900 OUTWRD(LOC1ST)=LPC
 1000 LOC1ST=LOC1ST+1
 1100 RETURN
```

```
      END
$IBFTC MASSCK  LIST
      SUBROUTINE MASSCK
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),INLINE(80),INVLOC(9),ROT(3
     13),ROTLOC(8),TRAN(3,8),TRGLOC(8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(1),LA),(ALP
     1H(3),LC),(ALPH(4),LD),(ALPH(6),LF),(ALPH(9),LI),(ALPH(10),LJ),(ALP
     2H(11),LK),(ALPH(16),LP),(ALPH(21),LU),(ALPH(22),LV),(ALPH(24),LX),
     3(ALPH(25),LY),(ALPH(26),LZ),(ALPH(30),LO),(ALPH(46),LBL),(CM(653),
     4INLINE(1)),(CM(746),LOCFAC),(CM(774),NOFPS),(CM(959),MORF),(CM(978
     5),ROTFAC),(CM(1380),IFUS),(CM(1381),INVLOC(1)),(CM(1390),INVLST),(
     6CM(1419),NODCOS),(CM(1420),NOROT),(CM(1421),ROT(1)),(CM(1429),IFPR
     7NT),(CM(1430),TRAN(1,1)),(CM(1454),ROTLOC(1)),(CM(1654),TRGLOC(1))
     8,(CM(1662),TRGLST)
C THE MASSCK SUBROUTINE CHECKS THE LEGALITY OF THE ROTATIONS AND
C TRANSLATIONS ON THE MASS INPUT CARD. ROTLOC,INVLOC,AND TRGLOC
C ARE SET UP IN THIS ROUTINE.
      INTEGER ROTFAC,TRGLST,TRGLOC,ROT,TRAN
      INTEGER ROTLOC
  100 ROTFAC=1
  150 IF(MORF.EQ.LF) ROTFAC=LOCFAC
  200 TRGLST=0
  300 INVLST=0
  350 IF(MORF.EQ.LF) INVLST=1
  400 NODCOS=0
  500 IFPRNT=0
  600 IFUS=0
  700 DO 4800 I=1,9
  800 LETTST=INLINE(7*I+2)
  900 IF(LETTST.EQ.LU)GO TO 4800
 1000 IF(LETTST.EQ.LZ)GO TO 4100
 1100 IF(LETTST.EQ.LY)GO TO 3900
 1200 IF(LETTST.EQ.LX)GO TO 3700
 1300 IF(LETTST.EQ.LK)GO TO 3500
 1400 IF(LETTST.EQ.LJ)GO TO 3300
 1500 IF(LETTST.EQ.LI)GO TO 3100
 1600 IF(LETTST.EQ.LD)GO TO 2400
 1700 IF(LETTST.EQ.LC)GO TO 2200
 1800 IF(LETTST.EQ.LP)GO TO 5000
 1900 IF(LETTST.EQ.LBL)GO TO 5100
 2000 CALL ERROR(7*I+2,1)
 2100 GO TO 4800
 2200 ROT(I)=-4
 2300 GO TO 2600
 2400 ROT(I)=4
C NODCOS GIVES THE POSITION OF THE DIRECTION COSINE (IE. 1-7)
 2500 NODCOS=I
 2600 ROTLOC(I)=ROTFAC
C THE DIRECTION COSINE REQUIRES TWICE THE AMOUNT OF SPACE OF ANY OTHER
C ROTATION.
C ROTLOC(I) GIVES THE POSITION OF THE ROTATION IN THE CODED WORD
C (IE. 1-10**8).
```

```
 2700 ROTFAC=ROTFAC*LOCFAC
 2800 INVLST=INVLST+1
 2900 INVLOC(INVLST)=I
C THE INVLOC IS AN INVERSE POSITION LOCATOR NECESSARY BECAUSE THE
C DIRECTION COSINE REQUIRES TWO POSITIONS IN THE CODED WORD
 3000 GO TO 4500
 3100 ROT(I)=-1
 3200 GO TO 4200
 3300 ROT(I)=-2
 3400 GO TO 4200
 3500 ROT(I)=-3
 3600 GO TO 4200
 3700 ROT(I)=1
 3800 GO TO 4200
 3900 ROT(I)=2
 4000 GO TO 4200
 4100 ROT(I)=3
 4200 ROTLOC(I)=ROTFAC
 4250 IF(ROTFAC.EQ.1)GO TO 4500
C TRGLOC GIVES THE LOCATION OF POSSIBLE TRIGONOMETRIC REDUCTIONS
 4300 TRGLST=TRGLST+1
 4400 TRGLOC(TRGLST)=ROTFAC
 4500 ROTFAC=ROTFAC*LOCFAC
 4600 INVLST=INVLST+1
 4700 INVLOC(INVLST)=I
 4800 CONTINUE
 4900 GO TO 5100
 5000 IFPRNT=1
 5050 NOFPS=NOFPS+1
 5100 NOROT=I-1
 5200 IF(NOROT.GT.0)GO TO 5500
 5300 CALL ERROR(9,1)
 5400 GO TO 8700
 5500 NTOT=0
 5600 NC=6
 5700 DO 8100 I=1,NOROT
 5800 IF(ROT(I).NE.0) NTOT=NTOT+1
 5900 NC=NC+4
 6000 DO 8100 J=1,3
 6100 NC=NC+1
 6200 LETTST=INLINE(NC)
 6300 IF(LETTST.EQ.LO.OR.LETTST.EQ.LBL)GO TO 8100
 6400 IF(LETTST.EQ.LC)GO TO 7000
 6500 IF(LETTST.EQ.LA)GO TO 7200
 6600 IF(LETTST.EQ.LV)GO TO 7400
 6700 IF(LETTST.EQ.LU)GO TO 7700
 6800 CALL ERROR(NC,1)
 6900 GO TO 8100
 7000 TRAN(J,I)=-1
 7100 GO TO 7500
 7200 TRAN(J,I)=-2
 7300 GO TO 7500
 7400 TRAN(J,I)=1
 7500 NTOT=NTOT+1
```

```
 7600 GO TO 8100
 7700 TRAN(J,I)=2
 7800 IF(I+1.NE.NODCOS) CALL ERROR(NC,1)
 7900 IFUS=1
 8000 IF(NTOT.GT.0) CALL ERROR(NC,1)
 8100 CONTINUE
 8200 IF(NODCOS.EQ.0)GO TO 8700
 8300 DO 8600 J=1,3
 8400 IF(TRAN(J,NODCOS).EQ.0)GO TO 8600
 8500 CALL ERROR(7*NODCOS+3+J,1)
 8600 CONTINUE
 8700 K=7*NOROT+10
 8800 DO 9000 I=K,72
 8900 IF(INLINE(I).NE.LBL) CALL ERROR(I,1)
 9000 CONTINUE
 9100 RETURN
      END
$IBFTC VARCK    LIST
      SUBROUTINE VARCK
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),INVLOC(9),ROT(33),ROTLOC(8
     1),TITLES(6,8,4),TRAN(3,8),TRGLOC(8),TRNTIT(6,8,3),VARFUN(30),VARLO
     2C(30),VARTIT(6,30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(13),LM),(AL
     1PH(30),LO),(ALPH(46),LBL),(CM(652),IFEND),(CM(746),LOCFAC),(CM(777
     2),NVAR),(CM(959),MORF),(CM(1146),VARFUN(1)),(CM(1176),VARTIT(1,1))
     3,(CM(1361),NTAPE6),(CM(1380),IFUS),(CM(1381),INVLOC(1)),(CM(1390),
     4INVLST),(CM(1418),NFORTP),(CM(1420),NOROT),(CM(1421),ROT(1)),(CM(1
     5430),TRAN(1,1)),(CM(1454),ROTLOC(1)),(CM(1462),TITLES(1,1,1)),(CM(
     61510),TRNTIT(1,1,1)),(CM(1654),TRGLOC(1)),(CM(1663),VARLOC(1))
      INTEGER ROT,TRAN,VARFUN,VARLOC,TITLES,VARTIT,TRNTIT,ROTLOC,TRGLOC
  150 NOTU=10
  200 IF(IFEND.GE.4)GO TO 600
  300 IFEND=IFEND+1
  400 CALL TITLCK
  500 GO TO 200
  600 DO 700 I=1,30
  700 VARLOC(I)=0
  800 DO 5000 NC=1,NOROT
  900 DO 5000 NT=1,4
 1000 IF(NT.LT.4)GO TO 1400
 1100 IF(ROT(NC).LE.0.OR.ROT(NC).EQ.4) GO TO 5000
 1200 NVF=ROT(NC)
 1300 GO TO 1700
 1340 NVF=3+NT
 1360 GO TO 1700
 1400 IF(ROT(NC).EQ.4) GO TO 1340
 1450 IF(TRAN(NT,NC).LE.0) GO TO 5000
 1500 NVF=10
 1600 IF(TRAN(NT,NC).EQ.2) NVF=20+NT
 1650 IF(NVF.LT.20) GO TO 1700
 1660 NOTU=MIN0(NOTU,5)
 1670 IF(MORF.NE.LM) GO TO 5000
```

```
1700 IF(NVAR.EQ.0)GO TO 3320
1800 DO 3300 NV=1,NVAR
1900 IF(VARFUN(NV).NE.NVF) GO TO 3300
2600 NROW=MODI4(NT)
2700 DO 2900 NCT=1,6
2800 IF(TITLES(NCT,NC,NROW).NE.VARTIT(NCT,NV))GO TO 3300
2900 CONTINUE
2950 IF(VARLOC(NV).EQ.0) GO TO 3100
2975 CALL ERROR(81,1)
3000 WRITE(NTAPE6,3010)(VARTIT(I,NV),I=1,6)
3010 FORMAT(48HOTHIS TITLE IS GIVEN TO MORE THAN ONE VARIABLE, ,6A1)
3100 VARLOC(NV)=10*(NROW-1)+NC
3150 IF(ROT(NC).EQ.4) VARLOC(NV)=NC
3200 GO TO 5000
3300 CONTINUE
3320 IF(MORF.EQ.LM) GO TO 3400
3322 NROW=MODI4(NT)
3325 IF(ROT(NC).NE.4) GO TO 3355
3335 IF(TITLES(1,NC,NROW).EQ.LBL.OR .TITLES(1,NC,NROW).EQ.LO)GO TO 5000
3355 WRITE(NTAPE6,3360)(TITLES(I,NC,NROW),I=1,6)
3360 FORMAT(12H THE TITLE  ,6A1,98H HAS NOT BEEN DEFINED PREVIOUSLY FOR
    1 A MASS.  MOVE THIS SET OF FORCE CONTROL CARDS TO LAST IN SET.)
3370 CALL ERROR (81,1)
3380 GO TO 5000
3400 IF(NVF.LT.4.OR.NVF.GT.6) GO TO 4500
3600 IF(TRNTIT(1,NC,NT).EQ.LBL.OR.TRNTIT(1,NC,NT).EQ.LO) GO TO 5000
3700 NVAR=NVAR+1
3800 VARFUN(NVAR)=NVF
4000 VARLOC(NVAR)=NC
4100 DO 4200 NCT=1,6
4200 VARTIT(NCT,NVAR)=TRNTIT(NCT,NC,NT)
4350 CALL SETAUX(2)
4400 GO TO 5000
4500 NVAR=NVAR+1
4550 VARFUN(NVAR)=NVF
4600 NROW=MODI4(NT)
4700 VARLOC(NVAR)=10*(NROW-1)+NC
4800 DO 4900 NCT=1,6
4900 VARTIT(NCT,NVAR)=TITLES(NCT,NC,NROW)
4925 IF(NVF.GT.20) NOTU=1
5000 CONTINUE
5010 IF(NOTU.LT.10) CALL SETAUX(NOTU)
5020 IF(MORF.EQ.LM) GO TO 5100
5030 I=3
5040 IF(NFORTP.EQ.5) I=4
5050 IF(NFORTP.EQ.2.OR.NFORTP.EQ.5) CALL SETAUX(I)
5100 IF(IFUS.EQ.0.OR.MORF.NE.LM) GO TO 5900
5200 N=9
5300 DO 5700 I=1,8
5400 ROTLOC(I)=ROTLOC(I)*LOCFAC
5500 INVLOC(N)=INVLOC(N-1)
5600 TRGLOC(I)=TRGLOC(I)*LOCFAC
5700 N=N-1
5800 INVLST=INVLST+1
```

```
 5900 RETURN
      END
$IBFTC TITLCK  LIST
      SUBROUTINE TITLCK
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),IXYTIT(6,4),ROT(33),ROTTIT
     1(6,8),TRAN(3,8),TRNTIT(6,8,3)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(1),LA),(ALP
     1H(3),LC),(ALPH(13),LM),(ALPH(18),LR),(ALPH(21),LU),(ALPH(22),LV),(
     2ALPH(30),LO),(ALPH(41),LCOM),(ALPH(46),LBL),(CM(652),IFEND),(CM(13
     376),FIRSTC),(CM(1377),FIRSTR),(CM(1378),FIRSTS),(CM(1391),IXYTIT(1
     4,1)),(CM(1415),LASTC),(CM(1416),LASTR),(CM(1417),LASTS),(CM(1418),
     5NFORTP),(CM(1420),NOROT),(CM(1421),ROT(1)),(CM(1430),TRAN(1,1)),(C
     6M(1462),ROTTIT(1,1)),(CM(1510),TRNTIT(1,1,1)),(CM(959),MORF)
C THE TITLCK SUBROUTINE GIVES TITLES TO THOSE FACTORS REQUIRING TITLES
C WHICH HAVE NONE GIVEN.
      INTEGER FIRSTC,FIRSTS,FIRSTR,FIRSTX
      INTEGER ROT,TRAN,ROTTIT,TRNTIT,ALPH
      ISHIFT(I)=I/(2**30)
  100 DO 2000 I=1,NOROT
  200 IF(IFEND.GT.1)GO TO 800
  300 IF(ROT(I).EQ.0)GO TO 2000
  400 IF(ROTTIT(1,I).NE.LBL)GO TO 2000
  500 ROTTIT(1,I)=LR
  600 ROTTIT(2,I)=ALPH(I+30)
  700 GO TO 2000
  800 IF(TRNTIT(1,I,IFEND-1).NE.LBL)GO TO 2000
  900 NC=TRAN(IFEND-1,I)+3
 1000 GO TO(1100,1300,2000,1500,1700),NC
 1100 TRNTIT(1,I,IFEND-1)=LA
 1200 GO TO 1800
 1300 TRNTIT(1,I,IFEND-1)=LC
 1400 GO TO 1800
 1500 TRNTIT(1,I,IFEND-1)=LV
 1600 GO TO 1800
 1700 TRNTIT(1,I,IFEND-1)=LU
 1720 TRNTIT(2,I,IFEND-1) =ALPH(IFEND+29)
 1740 GO TO 2000
 1800 TRNTIT(2,I,IFEND-1)=ALPH(IFEND+29)
 1900 TRNTIT(3,I,IFEND-1)=ALPH(I+30)
 2000 CONTINUE
 2100 NC=IFEND-1
 2200 IF(IFEND.EQ.1) NC=4
 2250 IF(NC.EQ.4)GO TO 2400
 2275 IF(MORF.NE.LM)GO TO 2500
 2300 IF(IXYTIT(1,NC).EQ.LBL.OR.IXYTIT(1,NC).EQ.LO) IXYTIT(1,NC)=0
 2400 RETURN
 2500 IF(IXYTIT(2,NC).EQ.LCOM) GO TO 3000
 2600 IF(NFORTP.EQ.3.OR.NFORTP.EQ.6) GO TO 2400
 2700 DO 2800 I=2,7
 2800 CALL ERROR(I,1)
 2900 GO TO 2400
 3000 IFR=ISHIFT(IXYTIT(1,NC))
```

```
 3100 IF(IFR.EQ.0) IFR=1
 3200 ILR=ISHIFT(IXYTIT(3,NC))
 3300 IF(IFR.LE.0.OR.IFR.GT.8) GO TO 2600
 3400 IF(ILR.LE.0.OR.ILR.GT.8) GO TO 2600
 3500 GO TO (3600,3900,4200),NC
 3600 FIRSTC=IFR
 3700 LASTC =ILR
 3800 GO TO 2400
 3900 FIRSTR=IFR
 4000 LASTR =ILR
 4100 GO TO 2400
 4200 FIRSTS=IFR
 4300 LASTS =ILR
 4700 GO TO 2400
      END
$IBFTC STITLE  LIST
      SUBROUTINE STITLE
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),INLINE(80),IXYTIT(6,4),TIT
     1LES(6,8,4)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(46),LBL),(C
     1M(652),IFEND),(CM(653),INLINE(1)),(CM(1391),IXYTIT(1,1)),(CM(1420)
     2,NOROT),(CM(1462),TITLES(1,1,1))
C THE STITLE SUBROUTINE STORES ALL GIVEN TITLES, AFTER LEFT JUSTIFYING
C AND REMOVING EMBEDDED BLANKS.
      INTEGER TITLES
  100 NC=7
  200 DO 1000 I=1,NOROT
  300 NC=NC+1
  400 NTOT=0
  500 DO 1000 J=1,6
  600 NC=NC+1
  700 IF(INLINE(NC).EQ.LBL)GO TO 1000
  800 NTOT=NTOT+1
  900 TITLES(NTOT,I,IFEND)=INLINE(NC)
 1000 CONTINUE
 1100 NC=IFEND-1
 1200 IF(IFEND.EQ.1) NC=4
 1300 NTOT=0
 1400 DO 1800 I=2,7
 1500 IF(INLINE(I).EQ.LBL)GO TO 1800
 1600 NTOT=NTOT+1
 1700 IXYTIT(NTOT,NC)=INLINE(I)
 1800 CONTINUE
 2000 CALL TITLCK
 2100 RETURN
      END
$IBFTC EXPP    LIST
      SUBROUTINE EXPP
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),MASTIT(6),TRAN(3,8),TRNTIT
     1(6,8,3)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(CM(747),LOCIST),
     1(CM(1141),ROTNUM),(CM(1361),NTAPE6),(CM(1409),MASTIT(1)),(CM(1420)
```

```
     2,NOROT),(CM(1430),TRAN(1,1)),(CM(1510),TRNTIT(1,1,1))
C THE EXPP SUBROUTINE CALCULATES THE POSITION VECTOR OF THE MASSES
C WHICH HAVE A 'P' AFTER THE ROTATIONS ON THE INPUT CARD.
      INTEGER ROTNUM,TRAN
  100 DO 1400 IROW=1,3
  200 WRITE(NTAPE6,300)ALPH(IROW+23),(MASTIT(I),I=1,6)
  300 FORMAT(//13H THIS IS THE ,A1,33H POSITION EXPANSION FOR THE MASS ,
     16A1//)
  400 ROTNUM=0
  500 DO 1400 I=1,NOROT
  800 CALL SETROT(I,1)
  900 DO 1400 J=1,3
 1000 IF(TRAN(J,I).EQ.0) GO TO 1400
 1100 LOC1ST=2
 1200 CALL WOTITL(TRNTIT(1,I,J),1)
 1300 CALL EXPROT(IROW,J,1)
 1400 CONTINUE
 1500 RETURN
      END
$IBFTC PRNTRO  LIST
      SUBROUTINE PRNTRO(INROT,NTYPE,IPR1)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),COSINE(4),INVLOC(9),ITEMP(
     14),MASS(5),MASTIT(6),OUTWRD(180),ROT(33),ROTTIT(6,8),SINE(4),IXY(4
     2),IXZ(4),IYZ(4)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(9),LI),(ALP
     1H(41),LCOM),(ALPH(44),LMINUS),(ALPH(45),LPLUS),(ALPH(46),LBL),(CM(
     2647),COSINE(1)),(CM(733),IXY(1)),(CM(737),IXZ(1)),(CM(741),IYZ(1))
     3,(CM(746),LOCFAC),(CM(747),LOC1ST),(CM(748),MASS(1)),(CM(753),MINU
     4S),(CM(779),OUTWRD(1)),(CM(1142),SINE(1)),(CM(1361),NTAPE6),(CM(13
     581),INVLOC(1)),(CM(1390),INVLST),(CM(1409),MASTIT(1)),(CM(1421),RO
     6T(1)),(CM(1462),ROTTIT(1,1))
C THE PRNTRO SUBROUTINE SETS UP THE OUTPUT OF THE CODED WORD TO BE
C PRINTED. INROT IS THE CODED WORD. NTYPE EQUALS TWO MEANS AN INERTIA
C TERM IS PRESENT. NO INERTIA TERM IS PRESENT IF NTYPE EQUALS ONE.
      INTEGER OUTWRD,ALPH,ROT
  100 IF(INROT.LT.0) RETURN
C IF THE CODED WORD IS LESS THS
C IF THE CODED WORD IS LESS THAN ZERO THIS MEANS THE ACTUAL ELEMENT
C EQUALS ZERO.
  200 IF(NTYPE.EQ.2)GO TO 800
  300 IF(INROT.LT.MINUS)GO TO 700
  400 OUTWRD(1)=LMINUS
  500 INROT=INROT-MINUS
  600 GO TO 800
  700 OUTWRD(1)=LPLUS
  800 IF(INROT.EQ.0)GO TO 5700
  900 DO 5500 I=1,INVLST
 1000 N=MOD(INROT,LOCFAC)
 1100 INROT=INROT/LOCFAC
 1200 IF(N.NE.0)GO TO 1300
 1250 IF(I.NE.1. OR.IABS(NTYPE).LE.1)GO TO 5500
 1300 CALL WOTAST
```

```
1400 IF(I.NE.1)GO TO 3300
1500 IF(IABS(NTYPE).LE.1)GO TO 3250
1600 IF(N.EQ.4)GO TO 2200
1700 IF(N.EQ.5)GO TO 2400
1800 IF(N.EQ.6)GO TO 2600
1850 IF(N.EQ.7)GO TO 3100
1900 IF(N.LT.4.AND.N.GT.0)GO TO 2800
2000 CALL WOTITL(MASS(1),2)
2100 GO TO 3100
2200 CALL WOTITL(IXY(1),2)
2300 GO TO 3100
2400 CALL WOTITL(IXZ(1),2)
2500 GO TO 3100
2600 CALL WOTITL(IYZ(1),2)
2700 GO TO 3100
2800 LOC1ST=LOCUP(LI)
2900 LOC1ST=LOCUP(ALPH(N+23))
3100 CALL WOTITL(MASTIT(1),1)
3150 IF(MASTIT(1).EQ.LBL.AND.N.EQ.7) LOC1ST=LOC1ST-1
3200 GO TO 5500
3250 IF(N.NE.7.OR.NTYPE.NE.-1) GO TO 3300
3260 LOC1ST=LOC1ST-1
3270 GO TO 5500
3300 IN=INVLOC(I)
3400 L=IABS(ROT(IN))
3500 IF(L.GE.4)GO TO 4900
3600 IF(N.EQ.2)GO TO 4500
3700 IF(N.EQ.1)GO TO 4600
3800 IF(N.EQ.3)GO TO 4300
3900 IF(N.EQ.4)GO TO 4200
4000 CALL WOTITL(COSINE(1),2)
4100 GO TO 4300
4200 CALL WOTITL(SINE(1),2)
4300 CALL WOTITL(COSINE(1),2)
4400 GO TO 4700
4500 CALL WOTITL(SINE(1),2)
4600 CALL WOTITL(SINE(1),2)
4700 CALL WOTITL(ROTTIT(1,IN),1)
4800 GO TO 5500
4900 CALL WOTITL(ROTTIT(1,IN),2)
5000 NT=31+(N-1)/3
5050 ITEMP(1)=ALPH(NT)
5100 ITEMP(2)=LCOM
5200 NT=30 + MODI3(N-1)
5250 ITEMP(3)=ALPH(NT)
5300 ITEMP(4)=LBL
5400 CALL WOTITL(ITEMP(1),1)
5500 CONTINUE
5550 IF(NTYPE.LE.0)GO TO 6400
5600 GO TO 6100
5700 IF(NTYPE.EQ.1)GO TO 6100
5750 IF(NTYPE.LE.0)GO TO 6400
5800 CALL WOTAST
5900 CALL WOTITL(MASS(1),2)
```

```
 6000 CALL WOTITL(MASTIT(1),1)
 6100 LOC1=LOC1ST-1
 6150 WRITE(NTAPE6,6200)(OUTWRD(1),(OUTWRD(I),I=IPR1,LOC1))
 6200 FORMAT(1H ,120A1/10X,60A1)
 6400 RETURN
      END
$IBFTC SETROT  LIST
      SUBROUTINE SETROT(I,NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROT(33),ROTLOC(8),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(746),LOCFAC),(CM(753),MINUS),
     1(CM(778),ONE),(CM(979),ROTMAT(1,1,1)),(CM(1141),ROTNUM),(CM(1421),
     2ROT(1)),(CM(1454),ROTLOC(1))
C THE SETROT SUBROUTINE SETS UP THE ROTATION MATRIX. THE I DESIGNATES
C WHICH ROTATION THAT IS PRESENT. FOR NTYPE EQUALS 1 THE NORMAL MATRIX
C IS SET UP. FOR NTYPE EQUALS 2 ITS TRANSPOSE IS SET.
      INTEGER ROT,ROTLOC,ROTMAT,ROTNUM,ONE
  100 IF(ROT(I).EQ.0) RETURN
  200 CALL ZEROT
  300 J=IABS(ROT(I))
  400 IF(J.GE.4)GO TO 1600
  500 NDIAG=3*ROTLOC(I)
  600 NSKEW=ROTLOC(I)
  700 IF(NTYPE.EQ.2) NSKEW=NADD(NSKEW,MINUS)
  800 DO 900 L=1,3
  900 ROTMAT(L,L,ROTNUM)=NDIAG
 1000 ROTMAT(J,J,ROTNUM)=ONE
 1100 J=MODI3(J)
 1200 L=MODI3(J)
 1300 ROTMAT(L,J,ROTNUM)=NSKEW
 1400 ROTMAT(J,L,ROTNUM)=NADD(NSKEW,MINUS)
 1500 RETURN
 1600 ND=ROTLOC(I)
 1700 IF(NTYPE.EQ.2)GO TO 2300
 1800 DO 2100 J=1,3
 1900 DO 2100 L=1,3
 2000 ROTMAT(J,L,ROTNUM)=ND
 2100 ND=ND+ROTLOC(I)
 2200 RETURN
 2300 ND=ND*LOCFAC
 2400 ND1=ND
 2500 DO 2700 J=1,9
 2600 ROTMAT(J,1,ROTNUM)=ND
 2700 ND=ND+ND1
 2800 RETURN
      END
$IBFTC EXPROT  LIST
      SUBROUTINE EXPROT(NROW1,LSTCOL,NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300),NCOLMN(18),PA
     1IRPRD(18),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11)),(CM(651),ICOUNT),(CM(747),LOC1ST),(CM(753),MINUS),(CM(754),NCO
```

```
    2LMN(1)),(CM(778),ONE),(CM(959),PARPRD(1)),(CM(977),PRDCON),(CM(979
    3),ROTMAT(1,1,1)),(CM(1141),ROTNUM)
      INTEGER ROTNUM,ROTMAT,PARPRD,CODWRD
  100 LOC1=LOC1ST
  150 IF(ROTNUM.EQ.0) CALL SETDEL(ONE)
  200 DO 300 I=1,ROTNUM
  300 NCOLMN(I)=1
  400 NCOLMN(ROTNUM)=LSTCOL
  500 I=1
  600 NC=NCOLMN(I)
  700 NCM1=NCOLMN(I-1)
  800 IF(I.GT.1)GO TO 1400
  900 PARPRD(1)=ROTMAT(NROW1,NC,1)
 1000 IF(PARPRD(1).LT.0)GO TO 1500
 1100 IF(ROTNUM.EQ.1)GO TO 2400
 1200 I=I+1
 1300 GO TO 600
 1400 IF(ROTMAT(NCM1,NC,I).GE.0)GO TO 2200
 1500 IF(I.GE.ROTNUM)GO TO 1900
 1600 NCOLMN(I)=NCOLMN(I)+1
 1700 IF(NCOLMN(I).LE.3)GO TO 600
 1800 NCOLMN(I)=1
 1900 I=I-1
 2000 IF(I.LE.0) RETURN
 2100 GO TO 1600
 2200 PARPRD(I)=NADD(PARPRD(I-1),ROTMAT(NCM1,NC,I))
 2300 IF(I.LT.ROTNUM)GO TO 1200
 2400 IF(NTYPE.EQ.2)GO TO 2800
 2500 LOC1ST=LOC1
 2600 CALL PRNTRO (PARPRD(I),NTYPE,2)
 2700 GO TO 1900
 2800 ICOUNT=ICOUNT+1
 2900 COEF(ICOUNT)=PRDCON
 3000 IF(PARPRD(I).GE.MINUS) COEF(ICOUNT)=-PRDCON
 3100 CODWRD(ICOUNT)=MOD(PARPRD(I),MINUS)
 3200 GO TO 1900
      END
$IBFTC SETDEL  LIST
      SUBROUTINE SETDEL(N)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(778),ONE),(CM(979),ROTMAT(1,1
     1,1)),(CM(1141),ROTNUM)
      INTEGER ROTNUM,ONE,ROTMAT
C THE SETDEL SUBROUTINE SETS UP THE KRONECKER DELTA.
  100 CALL ZEROT
  200 DO 300 J=1,3
  300 ROTMAT(J,J,ROTNUM)=N
  400 RETURN
      END
$IBFTC MODI4   LIST
      FUNCTION MODI4(I)
  100 MODI4=MOD(I,4)+1
  200 RETURN
```

```
 300 END
$IBFTC FORCK   LIST
      SUBROUTINE FORCK
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),INLINE(80)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(4),LD),(ALP
     1H(5),LE),(ALPH(7),LG),(ALPH(12),LL),(ALPH(18),LR),(CM(653),INLINE(
     21)),(CM(1418),NFORTP)
 100 IN=INLINE(2)
 200 IF(IN.EQ.LD)GO TO 1700
 300 IF(IN.EQ.LE)GO TO 1500
 400 IF(IN.EQ.LR)GO TO 1300
 500 IF(IN.EQ.LG)GO TO 1100
 600 IF(IN.EQ.LL)GO TO 900
 700 CALL ERROR(2,1)
 800 GO TO 2500
 900 NFORTP=1
1000 GO TO 2400
1100 NFORTP=2
1200 GO TO 2400
1300 NFORTP=3
1400 GO TO 2400
1500 NFORTP=4
1600 GO TO 2400
1700 IF(INLINE(3).EQ.LR)GO TO 2100
1800 IF(INLINE(3).EQ.LL)GO TO 2300
1900 CALL ERROR(3,1)
2000 GO TO 2500
2100 NFORTP=6
2200 GO TO 2400
2300 NFORTP=5
2400 CALL MASSCK
2500 RETURN
      END
$IBFTC EXPEQ3  LIST
      SUBROUTINE EXPEQ3
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(775),NOFVAR),(CM(1379),FIRSTX
     1),(CM(1380),IFUS),(CM(1419),NODCOS),(CM(1420),LASTX),(CM(1663),VAR
     2LOC(1))
C THE EXPEQ3 SUBROUTINE DETERMINES THE LAGRANGE EQUATION FOR THE
C GENERALIZED COORDINATES
      INTEGER FIRSTX,FIRSTY,VARLOC
 200 IF(NODCOS.EQ.0)GO TO 400
 300 IF(MOD(VARLOC(NOFVAR),10).GE.NODCOS)GO TO 600
 400 FIRSTX=1
 500 GO TO 1600
 600 FIRSTX=1+NODCOS
 700 IF(IFUS.EQ.0)GO TO 1000
 800 CALL EQ3335
 900 GO TO 1400
1000 IF(NODCOS.EQ.1)GO TO 1400
```

```
 1100 FIRSTY=1
 1200 LASTY=NODCOS-1
 1300 CALL EQ3132(FIRSTY,LASTY)
 1400 CALL EQ3436
 1500 CALL EQ37
 1600 CALL EQ3132(FIRSTX,LASTX)
 1700 RETURN
      END
$IBFTC SETXQM  LIST
      SUBROUTINE SETXQM(FIRX,LASX,NOCQM)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROT(33)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(778),ONE),(CM(1141),ROTNUM),(
     1CM(1421),ROT(1))
      INTEGER FIRX,ROTNUM,ROT
C THE SETXQM SUBROUTINE EITHER SETS UP DX/DQ(M) FOR ROTATIONS WHERE
C THE Q(M) IS BETWEEN THE X LIMITS FIRX AND LASX OR IT SETS UP THE
C MATRICES BETWEEN FIRX AND LASX. NOCQM IS THE COLUMN WHERE THE Q(M)
C IS LOCATED. IF NOCQM IS NOT BETWEEN FIRX AND LASX NO DERIVATIVE IS
C TAKEN.
  200 J=LASX
  300 DO 800   I=FIRX,LASX
  400 IF(J.EQ.NOCQM) CALL SETEPS(ROT(J),2)
  700 CALL SETROT(J,2)
  800 J=J-1
  900 RETURN
      END
$IBFTC SETEPS  LIST
      SUBROUTINE SETEPS(ITYPE,NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(753),MINUS),(CM(778),ONE),(CM
     1(979),ROTMAT(1,1,1)),(CM(1141),ROTNUM)
      INTEGER ROTMAT,ONE,ROTNUM
C THE SETEPS SUBROUTINE SETS UP THE EPSILON MATRIC IF NTYPE IS 1. FOR
C NTYPE EQUALS 2 THE TRANSPOSE EPSILON MATRIX IS GENERATED.
  100 IF(ITYPE.LE.0)GO TO 1100
  200 CALL ZEROT
  300 J=MODI3(ITYPE)
  400 K=MODI3(J)
  500 IF(NTYPE.EQ.1)GO TO 900
  600 L=K
  700 K=J
  800 J=L
  900 ROTMAT(J,K,ROTNUM)=MINUS
 1000 ROTMAT(K,J,ROTNUM)=ONE
 1100 RETURN
      END
$IBFTC PRINT2  LIST
      SUBROUTINE PRINT2(NTERMS)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),CODWRD(300),COEF(300),OUTW
     1RD(180)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(CM(47),CODWRD(1)
```

```
     1),(CM(347),COEF(1)),(CM(747),LOC1ST),(CM(779),OUTWRD(1))
  100 LOC1=LOC1ST
  200 DO 2300 I=1,NTERMS
  225 N=COEF(I)/ABS(COEF(I))+0.5
  250 OUTWRD(1)=ALPH(N+44)
  300 ICOEF=COEF(I)*100.
  350 ICOEF=IABS(ICOEF)
  400 OUTWRD(3)=ALPH(30)
  450 OUTWRD(4)=ALPH(30)
  500 K=7
  600 DO 2100 J=1,5
  700 IF(J-3)1200,1600,800
  800 IF(ICOEF-1) 900,1100,1200
  900 IF(J.EQ.4) K=K-1
 1000 GO TO 2200
 1100 IF(K.LT.7)GO TO 1200
 1125 K=8
 1150 GO TO 2200
 1200 N=MOD(ICOEF,10)
 1300 ICOEF=ICOEF/10
 1400 IF(N.NE.0) GO TO 1900
 1500 GO TO 1700
 1600 N=12
 1700 IF(K.LT.7) GO TO 1900
 1800 IF(J.LE.3) GO TO 2100
 1900 K=K-1
 2000 OUTWRD(K)=ALPH(N+30)
 2100 CONTINUE
 2200 LOC1ST=LOC1
 2300 CALL PRNTRO(CODWRD(I),2 ,K)
 2400 LOC1ST=LOC1
 2500 RETURN
      END
$IBFTC SETEPP  LIST
      SUBROUTINE SETEPP(I,J)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROTMAT(3,3,18)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(753),MINUS),(CM(778),ONE),(CM
     1(979),ROTMAT(1,1,1)),(CM(1141),ROTNUM)
      INTEGER ROTMAT,ROTNUM,ONE
  100 CALL ZEROT
  200 IF(I.EQ.J)GO TO 600
  300 ROTMAT(I,J,ROTNUM)=ONE
  400 ROTMAT(J,I,ROTNUM)=ONE
  500 GO TO 1000
  600 DO 900 K=1,3
  700 IF(K.EQ.I)GO TO 900
  800 ROTMAT(K,K,ROTNUM)=MINUS
  900 CONTINUE
 1000 RETURN
      END
$IBFTC TESTRN  LIST
      SUBROUTINE TESTRN(IR1,IC1,IR2,IC2,IT,NT)
```

```
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),IXYTIT(6,4),ROT(33),TRAN(3,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1391),IXYTIT(1,1)),(CM(1421),
     1ROT(1)),(CM(1430),TRAN(1,1))
      INTEGER TRAN
  100 NT=0
  200 IT=0
  300 IF(IC1.NE.IC2)GO TO 1700
  400 IF(IR2-IR1)1100,500,700
  500 NT=1
  600 IF(TRAN(IR1,IC1)+2)2000,1300,2000
  700 IF(TRAN(IR2,IC2).NE.-2)GO TO 1800
  800 IF(TRAN(IR1,IC1).NE.-2)GO TO 1100
  900 IN=IR1+IR2-2
 1000 IF(IXYTIT(1,IN).NE. 0)GO TO 1500
 1100 NT=-1
 1200 GO TO 2000
 1300 IT=1
 1400 GO TO 2000
 1500 IT=2
 1600 GO TO 2000
 1700 IF(TRAN(IR2,IC2).EQ.-2)GO TO 1100
 1800 IF(TRAN(IR1,IC1).EQ.-2)GO TO 1100
 1900 IF(TRAN(IR2,IC2).EQ. 0)GO TO 1100
 2000 RETURN
      END
$IBFTC TESTQM  LIST
      SUBROUTINE TESTQM(JR,JC,NQ,LIM,IFX,ILX)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1379),FIRSTX),(CM(1663),VARLO
     1C(1))
C THE TESTQM SUBROUTINE CHECKS TO SEE IF THE VARIABLE NQ IS IN THE RANGE
C BETWEEN IFX AND ILX
      INTEGER VARLOC,FIRSTX
  100 IF(JC.LT.IFX)GO TO 1200
  200 IF(JC.GT.ILX)GO TO 1200
  300 LIM=VARLOC(NQ)
  400 IF(LIM.GE.10)GO TO 800
  500 IF(JC.LT.LIM)GO TO 1200
  600 IF(LIM.LT.IFX    ) LIM=-1
  700 GO TO 1300
  800 IF(MOD(LIM,10).NE.JC) GO TO 1200
  900 IF(LIM/10.NE.JR)GO TO 1200
 1000 LIM=0
 1100 GO TO 1300
 1200 LIM=-1
 1300 RETURN
      END
$IBFTC SETPQN  LIST
      SUBROUTINE SETPQN(IFX,ILX,NQN,LIM)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1146),VARFUN(1))
```

```
      INTEGER VARFUN
  100 IF(IFX.GT.ILX) GO TO 800
  200 DO 700 I=IFX,ILX
  300 CALL SETROT(I,1)
  400 IF(LIM.NE.1)GO TO 700
  500 LIN=MODI3(VARFUN(NQN)-1)
  600 CALL SETEPS(LIN,1)
  700 CONTINUE
  800 RETURN
      END
$IBFTC MODSTI  LIST
      SUBROUTINE MODSTI(NCOUNT,IROCOL,NWT,IXY)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11))
      INTEGER CODWRD
  100 DO 800 I=1,NCOUNT
  200 X=COEF(I)/2.0
  300 J=CODWRD(I)
  400 DO 800 II=1,3
  500 Y=X
  600 IF(IROCOL.EQ.II) Y=-X
  700 CODWRD(1)= J+II
  725 COEF(1)=Y
  800 CALL W34RW(2,IXY,1,NWT)
 1000 RETURN
      END
$IBFTC NQNCK   LIST
      SUBROUTINE NQNCK(NQ,IFX,ILX,ITEST)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1663),VARLOC(1))
      INTEGER VARLOC
  100 ITEST=MOD(VARLOC(NQ),10)
  200 IF(VARLOC(NQ).EQ.0)GO TO 500
  300 IF(ITEST.LT.IFX)GO TO 500
  400 IF(ITEST.LE.ILX)GO TO 600
  500 ITEST=-1
  600 RETURN
      END
$IBFTC TRIG    LIST
      SUBROUTINE TRIG(NTERMS)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300),ROT(33),ROTLO
     1C(8),TRGLOC(8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11)),(CM(746),LOCFAC),(CM(1421),ROT(1)),(CM(1454),ROTLOC(1)),(CM(16
     254),TRGLOC(1)),(CM(1662),TRGLST)
C THE TRIG SUBROUTINE DOES ARITHMETIC AND TRIGONOMETRIC REDUCTIONS
      INTEGER ROT,CODWRD,EXTI,EXTJ,TRGLST,TRGLOC,REMI,REMJ,ROTLOC
C THE FOLLOWING SECTION ARRANGES THE DIRECTION COSINE TERMS SO THAT
C ACTUAL REDUCTIONS MAY TAKE PLACE
```

```
  100 IF(NTERMS.LE.1) RETURN
  200 DO 1250 I=1,8
  300 K=IABS(ROT(I))
  400 IF(K.LT.4)GO TO 1250
  500 K=ROTLOC(I)
  600 DO 1200 J=1,NTERMS
  700 L=CODWRD(J)/K
  800 EXTJ=MOD(L,LOCFAC)
  900 EXTI=MOD(L/LOCFAC,LOCFAC)
 1000 IF(EXTJ.GE.EXTI)GO TO 1200
 1100 CODWRD(J)=CODWRD(J)+K*(EXTJ-EXTI)*(LOCFAC-1)
 1200 CONTINUE
 1250 CONTINUE
 1300 GO TO 1500
 1400 IF(NTERMS.LE.1) RETURN
C THE FOLLOWING SECTION DOES THE ARITHMETIC REDUCTIONS
 1500 NCHANG=0
 1600 NTMIN1=NTERMS-1
 1700 DO 2800 I=1,NTMIN1
 1800 IF(COEF(I).EQ.0.)GO TO 2800
 1900 IPLUS1=I+1
 2000 DO 2700 J=IPLUS1,NTERMS
 2100 IF(COEF(J).EQ.0.)GO TO 2700
 2200 IF(CODWRD(I).NE.CODWRD(J))GO TO 2700
 2300 COEF(I)=COEF(I)+COEF(J)
 2400 COEF(J)=0.
 2500 NCHANG=NCHANG+1
 2600 IF(COEF(I).EQ.0.)GO TO 2800
 2700 CONTINUE
 2800 CONTINUE
 2900 CALL COMPRS(NTERMS)
 3000 IF(NTERMS.LE.1) RETURN
C THE FOLLOWING SECTION DOES THE TRIGONOMETRIC REDUCTIONS
 3100 DO 5800 I=1,NTERMS
 3200 IF(COEF.EQ.0.)GO TO 5800
 3300 DO 5700 K=1,TRGLST
 3400 KK=TRGLOC(K)
 3500 EXTI=MOD(CODWRD(I)/KK,LOCFAC)
 3600 IF(EXTI.NE.2.AND.EXTI.NE.6)GO TO 5700
 3700 REMI=CODWRD(I)-EXTI*KK
 3800 DO 5600 J=1,NTERMS
 3900 IF(ABS(COEF(J)).NE.ABS(COEF(I)))GO TO 5600
 4000 IF(I.EQ.J)GO TO 5600
 4100 EXTJ=MOD(CODWRD(J)/KK,LOCFAC)
 4200 REMJ=CODWRD(J)-KK*EXTJ
 4300 IF(REMJ.NE.REMI)GO TO 5600
 4400 IF(EXTJ.EQ.0)GO TO 4900
 4500 IF(EXTJ+EXTI.NE.8)GO TO 5600
 4600 IF(COEF(I).NE.COEF(J))GO TO 5600
 4700 CODWRD(I)=REMI
 4800 GO TO 5300
 4900 IF(COEF(I)+COEF(J).NE.0.)GO TO 5600
 5000 COEF(I)=-COEF(I)
 5100 EXTI=8-EXTI
```

```
 5200 CODWRD(I)=REMI+KK*EXTI
 5300 COEF(J)=0.
 5400 NCHANG=NCHANG+1
 5500 GO TO 5800
 5600 CONTINUE
 5700 CONTINUE
 5800 CONTINUE
 5900 CALL COMPRS(NTERMS)
 6000 IF(NCHANG.GT.0)GO TO 1400
 6100 RETURN
      END
$IBFTC EXPEQ2  LIST
      SUBROUTINE EXPEQ2
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1379),FIRSTX),(CM(1380),IFUS)
     1,(CM(1419),NODCOS),(CM(1420),LASTX)
      INTEGER FIRSTX,FIRSTY
  100 FIRSTX=1+NODCOS
  200 IF(IFUS.EQ.0)GO TO 500
  300 CALL EQ3335
  400 GO TO 900
  500 IF(NODCOS.EQ.1)GO TO 900
  600 FIRSTY=1
  700 LASTY=NODCOS-1
  800 CALL EQ3132(FIRSTY,LASTY)
  900 CALL EQ3132(FIRSTX,LASTX)
 1000 CALL EQ37
 1100 CALL EQ2225
 1200 RETURN
      END
$IBFTC SETPRN  LIST
      SUBROUTINE SETPRN(LQM,JR1,JC1,LQN,LQK,JR2,JC2,NQN,NQK,NTYPE,LOC1)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),OUTWRD(180),TRNTIT(6,8,3),VARTIT(6,
     130),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(4),LD),(CM(
     1747),LOC1ST),(CM(779),OUTWRD(1)),(CM(1146),VARFUN(1)),(CM(1176),VA
     2RTIT(1,1)),(CM(1510),TRNTIT(1,1,1))
      INTEGER TRNTIT,OUTWRD,VARTIT ,VARFUN
  100 LOC1ST=IABS(LOC1)
  200 IF(LOC1.GT.0) CALL WOTAST
  300 IF(LQM.EQ.0)GO TO 600
  400 CALL WOTITL(TRNTIT(1,JC1,JR1),1)
  500 CALL WOTAST
  600 IF(MIN0(IABS(LQN),IABS(LQK)).EQ.0)GO TO 900
  700 CALL WOTITL(TRNTIT(1,JC2,JR2),1)
  800 CALL WOTAST
  900 GO TO (2200,1700,1000,1300), NTYPE
 1000 OUTWRD(LOC1ST)=LD
 1100 LOC1ST=LOC1ST+1
 1200 GO TO 1700
 1300 IF(VARFUN(NQK).GE.4.AND.VARFUN(NQK).LT.10) GO TO 1600
```

```
 1350 IF(VARFUN(NQK).GT.19) GO TO 1600
 1400 OUTWRD(LOC1ST)=LD
 1500 LOC1ST=LOC1ST+1
 1600 CALL WOTITL(VARTIT(1,NQK),1)
 1650 CALL WOTAST
 1700 IF(VARFUN(NQN).GE.4.AND.VARFUN(NQN).LT.10) GO TO 2000
 1750 IF(VARFUN(NQN).GT.19) GO TO 2000
 1800 OUTWRD(LOC1ST)=LD
 1900 LOC1ST=LOC1ST+1
 2000 CALL WOTITL(VARTIT(1,NQN),1)
 2100 RETURN
 2200 LOC1ST=LOC1ST-1
 2300 GO TO 2100
      END
$IBFTC PRINT3  LIST
      SUBROUTINE PRINT3(ICOUNT,ITEST,JR1,JR2,IXY,NWT)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300),OUTWRD(180)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11)),(CM(779),OUTWRD(1)),(CM(747),LOC1ST)
      INTEGER OUTWRD,CODWRD
  100 CALL TRIG(ICOUNT)
  200 IF(ICOUNT.EQ.0)GO TO 1200
  300 IF(ITEST.EQ.0) GO TO 1100
  310 GO TO (500,800,1600), ITEST
  500 CALL MODSTI(ICOUNT,JR1,NWT,IXY)
  700 GO TO 1200
  800 IADD=JR1+JR2+1
  900 DO 1000 I=1,ICOUNT
 1000 CODWRD(I)=CODWRD(I)+IADD
 1100 CALL PRINT2(ICOUNT)
 1200 RETURN
 1600 WRITE(NWT) ICOUNT,LOC1ST,OUTWRD
 1700 DO1800 I=1,ICOUNT
 1800 WRITE(NWT) COEF(I),CODWRD(I)
 1900 IXY=IXY+1
 2000 GO TO 1200
      END
$IBFTC EXPEQ1  LIST
      SUBROUTINE EXPEQ1
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(775),NOFVAR),(CM(1146),VARFUN
     1(1)),(CM(1379),FIRSTX),(CM(1419),NODCOS)
      INTEGER FIRSTX,VARFUN
  100 FIRSTX=1+NODCOS
  200 I=VARFUN(NOFVAR)-20
  300 CALL EQ1116(I)
  400 CALL EQ13(I)
  500 CALL EQ1217(I)
  600 CALL EQ1415(I)
  700 RETURN
      END
$IBFTC EQ2225  LIST
```

```
      SUBROUTINE EQ2225
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROTMAT(3,3,18),TRAN(3,8),VARFUN(30)
     1,WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(775),NOFVAR),(CM(778),ONE),(CM(977),PRDCON),(CM(979),ROTMAT(1,
     21,1)),(CM(1141),ROTNUM),(CM(1146),VARFUN(1)),(CM(1356),WTAPE(1)),(
     3CM(1379),FIRSTX),(CM(1420),LASTX),(CM(1430),TRAN(1,1))
      INTEGER VARFUN,FIRSTX,ROTMAT,ROTNUM,ONE,TRAN
  100 IOMEGA=VARFUN(NOFVAR)-3
  200 PRDCON=-1.0
  300 DO 7700 LOMEGA=1,3
  400 CALL FINDNQ(LOMEGA+3,NQL)
  500 IF(NQL.LE.0)GO TO 7700
  600 DO 7600 JOMEGA=LOMEGA,4
  800 IF(JOMEGA.EQ.4) GO TO 2400
 1100 IF(JOMEGA.EQ.IOMEGA.AND.IOMEGA.EQ.LOMEGA) GO TO 7600
 1200 CALL FINDNQ(JOMEGA+3,NQJ)
 1300 IF(NQJ.LE.0)GO TO 7600
 2400 IXYZ=0
 2500 DO 7200 ICOL1=FIRSTX,LASTX
 2600 DO 7200 IROW1=1,3
 2700 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 7200
 2800 DO 7100 ICOL2=ICOL1,LASTX
 2900 DO 7100 IROW2=1,3
 3000 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
 3100 IF(NTEST.LT.0)GO TO 7100
 3200 ICOUNT=0
 3300 JROW1=IROW1
 3400 JROW2=IROW2
 3500 JCOL1=ICOL1
 3600 JCOL2=ICOL2
 3700 ROTNUM=0
 3800 CALL SETXQM(FIRSTX,JCOL1,0)
 3900 IF(JOMEGA.LT.4) GO TO 4300
 4000 CALL SETEPS(LOMEGA,1)
 4100 CALL SETEPS(IOMEGA,1)
 4200 GO TO 5000
 4300 CALL ZEROT
 4310 IF(LOMEGA.EQ.JOMEGA)GO TO 4520
 4320 JL=(JOMEGA-IOMEGA)*(JOMEGA-LOMEGA)*(LOMEGA-IOMEGA)
 4330 IF(JL.EQ.0)GO TO 4390
 4340 JL=MODI3(IOMEGA)
 4350 ROTMAT(JL,JL,ROTNUM)=MINUS
 4360 JL=MODI3(JL)
 4370 ROTMAT(JL,JL,ROTNUM)=ONE
 4380 GO TO 5000
 4390 KL=IOMEGA
 4400 JL=JOMEGA
 4500 IF(IOMEGA.EQ.3) JL=LOMEGA
 4510 GO TO 4550
 4520 IF(IOMEGA.EQ.LOMEGA) GO TO 5000
 4530 JL=LOMEGA
```

```
4540 KL=JOMEGA
4550 IF(JL.EQ.MODI3(IOMEGA))GO TO 4590
4560 NT=MINUS
4570 L=MODI3(IOMEGA)
4580 GO TO 4610
4590 NT=ONE
4600 L=MODI3(JL)
4610 ROTMAT(L,KL,ROTNUM)=NT
5000 CALL SETPQN(FIRSTX,JCOL2,1,0)
5100 CALL EXPROT(JROW1,JROW2,2)
5200 IF(ITEST.EQ.1)GO TO 6300
5300 IF(ITEST.EQ.2)GO TO 5900
5400 IF(JOMEGA.EQ.4)GO TO 5700
5500 CALL SETPRN(1,JROW1,JCOL1,1,1,JROW2,JCOL2,NQL,NQJ,4,7)
5600 GO TO 6300
5700 CALL SETPRN(1,JROW1,JCOL1,1,1,JROW2,JCOL2,NQL,0,3 ,7)
5800 GO TO 6300
5900 IF(JOMEGA.EQ.4)GO TO 6200
6000 CALL SETPRN(0,1,1,0,0,1,1,NQL,NQJ,4,7)
6100 GO TO 6300
6200 CALL SETPRN(0,1,1,0,0,1,1,NQL,0,3,7)
6300 IF(NTEST.EQ.1)GO TO 7000
6400 NTEST=NTEST+1
6500 JCOL1=ICOL2
6600 JCOL2=ICOL1
6700 JROW1=IROW2
6800 JROW2=IROW1
6900 GO TO 3700
7000 IF(ICOUNT.GT.0) CALL PRINT3(ICOUNT,ITEST,JROW1,JROW2,IXYZ,WTAPE(3)
    1)
7100 CONTINUE
7200 CONTINUE
7300 IF(IXYZ.EQ.0)GO TO 7600
7400 CALL SETPRN(0,1,1,0,0,1,1,NQL,NQJ,7-MAX0(3,JOMEGA),7)
7500 CALL PRINT4(IXYZ,WTAPE(3))
7600 CONTINUE
7700 CONTINUE
7800 RETURN
     END
$IBFTC EQ3132  LIST
      SUBROUTINE EQ3132(IFX,ILX)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),IXYZ(2),ROT(33),TRAN(3,8),VARFUN(30
     1),WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(775),NOFVAR)
     1,(CM(777),NVAR),(CM(778),ONE),(CM(977),PRDCON),(CM(1141),ROTNUM),(
     2CM(1146),VARFUN(1)),(CM(1356),WTAPE(1)),(CM(1379),FIRSTX),(CM(1419
     3),NODCOS),(CM(1420),LASTX),(CM(1421),ROT(1)),(CM(1430),TRAN(1,1)),
     4(IXYZ(1),IXYQDD),(IXYZ(2),IXYQD)
      INTEGER FIRSTX,TRAN,VARFUN,ROTNUM,ROT,WTAPE
  100 IFRSTX=1
  200 IF(FIRSTX.GT.IFX) IFRSTX=2
  300 DO 10400 NQN=1,NVAR
  400 CALL NQNCK(NQN,IFX,ILX,ITEST)
```

```
 500 IF(ITEST.LT.0)GO TO 10400
 600 DO 10300 NQK=NQN,NVAR
 700 CALL NQNCK(NQK,IFX,ILX,ITEST)
 800 IF(ITEST.LT.0)GO TO 10300
 900 PRDCON=2.0
1000 IF(NQN.EQ.NQK) PRDCON=1.0
1100 IXYQD=0
1200 IXYQDD=0
1300 DO 9700 ICOL1=IFX,LASTX
1400 DO 9700 IROW1=1,3
1500 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 9700
1600 DO 9600 ICOL2=ICOL1,LASTX
1700 DO 9600 IROW2=1,3
1800 NQNDD=PRDCON
1900 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
2000 IF(NTEST.LT.0)GO TO 9600
2100 ICOUNT=0
2200 JROW1=IROW1
2300 JROW2=IROW2
2400 JCOL1=ICOL1
2500 JCOL2=ICOL2
2600 IF(JCOL1.LT.FIRSTX)GO TO 8500
2700 IF(VARFUN(NOFVAR).LE.6.AND.VARFUN(NOFVAR).GT.3)GO TO 3100
2800 CALL TESTQM(JROW1,JCOL1,NOFVAR,LIMQM,FIRSTX,LASTX)
2900 IF(LIMQM.LT.0)GO TO 8500
3000 GO TO 3200
3100 LIMQM=-1
3200 CALL TESTQM(JROW2,JCOL2,NQN,LIMQN,IFX,ILX)
3300 IF(LIMQN.LT.0)GO TO 8500
3400 CALL TESTQM(JROW2,JCOL2,NQK,LIMQK,IFX,ILX)
3500 IF(LIMQK.LT.0)GO TO 8500
3600 ROTNUM=0
3700 CALL SETXQM(FIRSTX,JCOL1,LIMQM)
3800 IF(LIMQM.EQ.-1) CALL SETEPS(VARFUN(NOFVAR)-3,2)
3900 IF(IFRSTX.EQ.2) CALL SETROT(NODCOS,2)
4200 LM1=LIMQM
4250 IF(LIMQM.EQ.0) LM1=JCOL1
4300 NQN1=NQN
4350 NQK1=NQK
4400 LIMQN1=LIMQN
4450 LIMQK1=LIMQK
4500 IF(NQNDD.EQ.1) GO TO 5100
4600 IF(LIMQN.LE.LIMQK)GO TO 4900
4700 NQN1=NQK
4750 NQK1=NQN
4800 LIMQN1=LIMQK
4850 LIMQK1=LIMQN
4900 IF(LIMQN1.NE.0) GO TO 5200
4950 IF(LIMQK1.EQ.0) GO TO 9600
5000 LN1=LIMQK1
5050 GO TO 5300
5100 IF(LIMQN.NE.0) GO TO 5200
5150 LN1=JCOL2
```

```
 5175 GO TO 5300
 5200 LN1=LIMQN1
 5300 MIN=MIN0(LM1,LN1)+1
 5400 IF(LIMQM.GE.0.AND.IFRSTX.EQ.1) ROTNUM=IRNUM(ROTNUM,IFX,MIN)
 5450 IF(LIMQM.LT.0.OR.IFRSTX.EQ.2) MIN=IFX
 5500 IF(NQNDD.EQ.1)GO TO 6300
 5550 IF(LIMQK1+1.NE.MIN) GO TO 5700
 5600 CALL SETEPS(MODI3(VARFUN(NQK1)-1),1)
 5650 GO TO 6300
 5700 IF(LIMQN1.GT.0)GO TO 5800
 5750 CALL SETPQN(MIN,JCOL2,NQK1,LIMQK1)
 5775 GO TO 7900
 5800 IF(LIMQN1+1.EQ.MIN) CALL SETEPS(MODI3(VARFUN(NQN1)-1),1)
 6000 CALL SETPQN(MIN,LIMQK1,NQN1,LIMQN1)
 6100 CALL SETEPS(MODI3(VARFUN(NQK1)-1),1)
 6200 MIN=LIMQK1+1
 6250 GO TO 6600
 6300 IF(LIMQN1+1.EQ.MIN) CALL SETEPS(MODI3(VARFUN(NQN1)-1),1)
 6600 CALL SETPQN(MIN,JCOL2,NQN1,LIMQN1)
 6700 IF(ROTNUM.EQ.0) CALL SETDEL(ONE)
 6800 IF(ROTNUM.GE.0)GO TO 7900
 6900 CALL PDUMP
 7000 GO TO 10500
 7900 CALL EXPROT(JROW1,JROW2,2)
 8000 IF(ITEST.EQ.1)GO TO 8500
 8100 IF(ITEST.EQ.0)GO TO 8400
 8200 LIMQM=0
 8300 LIMQN=0
 8400 CALL SETPRN(LIMQM,JROW1,JCOL1,LIMQN,LIMQK,JROW2,JCOL2,NQN,NQK,
     1NQNDD+2,7)
 8500 IF(NTEST.EQ.1)GO TO 9200
 8600 NTEST=1
 8700 JROW1=IROW2
 8800 JROW2=IROW1
 8900 JCOL1=ICOL2
 9000 JCOL2=ICOL1
 9100 GO TO 2700
 9200 IF(ICOUNT.GT.0) CALL PRINT3(ICOUNT,ITEST,JROW1,JROW2,IXYZ(NQNDD),
     1WTAPE(NQNDD+2))
 9300 IF(NQNDD.EQ.2)GO TO 9600
 9400 NQNDD=2
 9500 GO TO 1900
 9600 CONTINUE
 9700 CONTINUE
 9800 DO 10200 N=1,2
 9900 IF(IXYZ(N).LE.0)GO TO 10200
10000 CALL SETPRN(0,1,1,0,0,1,1,NQN,NQK,N+2,7)
10100 CALL PRINT4(IXYZ(N),WTAPE(N+2))
10200 CONTINUE
10300 CONTINUE
10400 CONTINUE
10500 RETURN
      END
$IBFTC FINDNQ  LIST
```

```
      SUBROUTINE FINDNQ(NOM,NQ)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(777),NVAR),(CM(1146),VARFUN(1
     1))
      INTEGER VARFUN
  100 NQ=-1
  200 DO 400 I=1,NVAR
  300 IF(VARFUN(I).EQ.NOM) NQ=I
  400 CONTINUE
  500 RETURN
      END
$IBFTC PRINT4  LIST
      SUBROUTINE PRINT4(IXY,NWT)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11))
  100 REWIND NWT
  200 CALL W34RW(1,IXY,IXY,NWT)
  400 REWIND NWT
  500 CALL TRIG(IXY)
  600 IF(IXY.GT.0) CALL PRINT2(IXY)
  700 RETURN
      END
$IBFTC EQ37     LIST
      SUBROUTINE EQ37
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8),VARFUN(30),WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(775),NOFVAR)
     1,(CM(777),NVAR),(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1146),VARFU
     2N(1)),(CM(1356),WTAPE(1)),(CM(1379),FIRSTX),(CM(1420),LASTX),(CM(1
     3430),TRAN(1,1))
      INTEGER TRAN,FIRSTX,VARFUN,ROTNUM,WTAPE
  100 PRDCON=2.0
  200 DO 4600 NON=1,NVAR
  300 CALL NONCK(NON,FIRSTX,LASTX,ITEST)
  400 IF(ITEST.LT.0)GO TO 4600
  500 DO 4500 JOMEGA=1,3
  600 CALL FINDNQ(JOMEGA+3,NQJ)
  700 IF(NQJ.LE.0)GO TO 4500
  800 IXYZNO=0
  900 DO 4400 ICOL1=FIRSTX,LASTX
 1000 DO 4400 IROW1=1,3
 1100 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 4400
 1200 DO 4300 ICOL2=ICOL1,LASTX
 1300 DO 4300 IROW2=1,3
 1400 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
 1500 IF(NTEST.LT.0)GO TO 4300
 1600 ICOUNT=0
 1700 JROW1=IROW1
 1800 JROW2=IROW2
 1900 JCOL1=ICOL1
```

```
2000 JCOL2=ICOL2
2050 ROTNUM=0
2100 IF(VARFUN(NOFVAR).LE.3)GO TO 2500
2200 IF(VARFUN(NOFVAR).GT.6)GO TO 2500
2300 LIMQM=-1
2400 GO TO 2700
2500 CALL TESTQM(JROW1,JCOL1,NOFVAR,LIMQM,FIRSTX,LASTX)
2600 IF(LIMQM.LT.0)GO TO 3500
2700 CALL TESTQM(JROW2,JCOL2,NQN,LIMQN,FIRSTX,LASTX)
2800 IF(LIMQN.LT.0)GO TO 3500
2900 CALL SETXQM(FIRSTX,JCOL1,LIMQM)
3000 IF(LIMQM.EQ.-1) CALL SETEPS(MODI3(VARFUN(NOFVAR)-1),2)
3100 CALL SETEPS(JOMEGA,1)
3200 CALL SETPQN(FIRSTX,JCOL2,NQN,LIMQN)
3300 CALL EXPROT(JROW1,JROW2,2)
3310 IF(ITEST-1) 3400,3500,3320
3320 CALL SETPRN(0,1,1,0,0,1,1,NQN,NQJ,4,7)
3330 GO TO 3500
3400 CALL SETPRN(LIMQM,JROW1,JCOL1,LIMQN,1,JROW2,JCOL2,NQN,NQJ,4,7)
3500 IF(NTEST.EQ.1)GO TO 4200
3600 NTEST=1
3700 JROW1=IROW2
3800 JROW2=IROW1
3900 JCOL1=ICOL2
4000 JCOL2=ICOL1
4100 GO TO 2050
4200 IF(ICOUNT.GT.0) CALL PRINT3(ICOUNT,ITEST,JROW1,JROW2,IXYZNO,
    1WTAPE(3))
4300 CONTINUE
4400 CONTINUE
4500 CONTINUE
4600 CONTINUE
4700 RETURN
     END
$IBFTC EQ3436  LIST
     SUBROUTINE EQ3436
     COMMON CM
     DIMENSIONCM(1700),COMON(325,9),TRAN(3,8),WTAPE(6)
     EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(775),NOFVAR)
    1,(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1356),WTAPE1),(CM(1379),FI
    2RSTX),(CM(1420),LASTX),(CM(1430),TRAN(1,1)),(CM(1356),WTAPE(1))
     INTEGER TRAN,ROTNUM,FIRSTX,WTAPE
 100 PRDCON=1.0
 200 DO 4500 JOMEGA=1,3
 300 CALL FINDNQ(JOMEGA+3,NQJ)
 400 IF(NQJ.LE.0)GO TO 4500
 500 DO 4400 LOMEGA=JOMEGA,4
 600 IF(LOMEGA.EQ.4)GO TO 900
 700 CALL FINDNQ(LOMEGA+3,NQL)
 800 IF(NQL.LE.0)GO TO 4400
 900 IXYZNO=0
1000 DO 4200 ICOL1=FIRSTX,LASTX
1100 DO 4200 IROW1=1,3
1200 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 4200
```

```
 1300 DO 4100 ICOL2=ICOL1,LASTX
 1400 DO 4100 IROW2=1,3
 1500 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
 1600 IF(NTEST.LT.0)GO TO 4100
 1700 ICOUNT=0
 1800 JROW1=IROW1
 1900 JROW2=IROW2
 2000 JCOL1=ICOL1
 2100 JCOL2=ICOL2
 2200 CALL TESTQM(JROW1,JCOL1,NOFVAR,LIMQM,FIRSTX,LASTX)
 2300 IF(LIMQM.LT.0)GO TO 3300
 2350 ROTNUM=0
 2400 CALL SETXQM(FIRSTX,JCOL1,LIMQM)
 2500 IF(LOMEGA.EQ.4)GO TO 2900
 2600 CALL SETEPP(JOMEGA,LOMEGA)
 2700 N=4
 2800 GO TO 3100
 2900 CALL SETEPS(JOMEGA,1)
 3000 N=3
 3100 CALL SETPRN(LIMQM,JROW1,JCOL1,1,1,JROW2,JCOL2,NQJ,NQL,N,7)
 3200 CALL SETPQN(FIRSTX,JCOL2,1,0)
 3250 CALL EXPROT(JROW1,JROW2,2)
 3300 IF(NTEST.EQ.1)GO TO 4000
 3400 NTEST=1
 3500 JROW1=IROW2
 3600 JROW2=IROW1
 3700 JCOL1=ICOL2
 3800 JCOL2=ICOL1
 3900 GO TO 2200
 4000 IF(ICOUNT.GT.0) CALL PRINT3(ICOUNT,ITEST,JROW1,JROW2,IXYZNO,
     1WTAPE(3))
 4100 CONTINUE
 4200 CONTINUE
 4300 IF(IXYZNO.GT.0) CALL PRINT4(IXYZNO,WTAPE(3))
 4400 CONTINUE
 4500 CONTINUE
 4600 RETURN
      END
$IBFTC EQ3335  LIST
      SUBROUTINE EQ3335
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),MASS(5),MASTIT(6),TRAN(3,8),VARFUN(
     130),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(748),MASS(1)),(CM(775),NOFVAR
     1),(CM(1141),ROTNUM),(CM(1146),VARFUN(1)),(CM(1379),FIRSTX),(CM(140
     29),MASTIT(1)),(CM(1420),LASTX),(CM(1430),TRAN(1,1)),(CM(1663),VARL
     3OC(1))
      INTEGER VARLOC,VARFUN,TRAN,ROTNUM,FIRSTX
  100 LIMCLO=MOD(VARLOC(NOFVAR),10)
  200 IF(VARFUN(NOFVAR).LT.10)GO TO 800
  300 LIMCHI=LIMCLO
  400 LIMRLO=VARLOC(NOFVAR)/10
  500 LIMRHI=LIMRLO
```

```
  600 LIMQM=0
  700 GO TO 1400
  800 LIMCHI=LASTX
  900 LIMRLO=1
 1000 LIMRHI=3
 1050 LIMQM=LIMCLO
 1100 IF(VARFUN(NOFVAR).LE.3)GO TO 1400
 1200 LIMQM=-1
 1300 LIMCLO=FIRSTX
 1400 DO 3900 ICOL=LIMCLO,LIMCHI
 1500 DO 3900 IROW=LIMRLO,LIMRHI
 1600 IF(IABS(TRAN(IROW,ICOL)).NE.1)GO TO 3900
 1650 ROTNUM=0
 1700 CALL SETXQM(FIRSTX,ICOL,LIMQM)
 1800 IF(LIMQM.EQ.-1) CALL SETEPS(MODI3(VARFUN(NOFVAR)-1),2)
 1900 NUMEND=ROTNUM
 2000 DO 3800 NOFU=1,3
 2100 CALL FINDNQ(NOFU+20,NQU)
 2200 IF(NQU.LE.0)GO TO 3800
 2300 DO 3700 JOMEGA=1,4
 2400 ROTNUM=NUMEND
 2500 IF(JOMEGA.LT.4)GO TO 2800
 2600 N=3
 2700 GO TO 3200
 2800 CALL FINDNQ(JOMEGA+3,NQJ)
 2900 IF(NQJ.LE.0)GO TO 3700
 3000 CALL SETEPS(JOMEGA,1)
 3100 N=4
 3200 CALL SETPRN(LIMQM,IROW,ICOL,0,0,1,1,NQU,NQJ,N,-2)
 3300 CALL WOTAST
 3400 CALL WOTITL(MASS(1),2)
 3500 CALL WOTITL(MASTIT(1),1)
 3600 CALL EXPROT(IROW,NOFU,1)
 3700 CONTINUE
 3800 CONTINUE
 3900 CONTINUE
 4000 RETURN
      END
$IBFTC EQ1415  LIST
      SUBROUTINE EQ1415(I)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),MASS(5),MASTIT(6),OUTWRD(1
     180),TRAN(3,8),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(CM(748),MASS(1))
     1,(CM(777),NVAR),(CM(778),ONE),(CM(779),OUTWRD(1)),(CM(1141),ROTNUM
     2),(CM(1146),VARFUN(1)),(CM(1379),FIRSTX),(CM(1409),MASTIT(1)),(CM(
     31420),LASTX),(CM(1430),TRAN(1,1))
      INTEGER FIRSTX,ROTNUM,OUTWRD,ALPH,TRAN
  100 DO 4700 NQN=1,NVAR
  200 CALL NQNCK(NQN,FIRSTX,LASTX,ITEST)
  300 IF(ITEST.LT.0)GO TO 4700
  400 DO 4600 NQK=NQN,NVAR
  500 CALL NQNCK(NQK,FIRSTX,LASTX,ITEST)
  600 IF(ITEST.LT.0)GO TO 4600
```

```
 700 DO 4500 JCOL=FIRSTX,LASTX
 800 DO 4500 JROW=1,3
 900 IF(IABS(TRAN(JROW,JCOL)).NE.1)GO TO 4500
1000 CALL TESTQM(JROW,JCOL,NQN,LIMQN,FIRSTX,LASTX)
1100 IF(LIMQN.LT.0)GO TO 4500
1200 CALL TESTQM(JROW,JCOL,NQK,LIMQK,FIRSTX,LASTX)
1300 IF(LIMQK.LT.0)GO TO 4500
1400 ROTNUM=0
1600 IF(NQN.NE.NQK)GO TO 2100
1700 NQNDD=1
1800 CALL SETPRN(LIMQN,JROW,JCOL,0,0,1,1,NQN,1,3,-2)
1900 CALL SETPQN(FIRSTX,JCOL,NQN,LIMQN)
2000 GO TO 4100
2100 CALL SETPRN(LIMQN,JROW,JCOL,0,0,1,1,NQN,NQK,4,3)
2200 OUTWRD(2)=ALPH(NQNDD+30)
2300 NQNDD=2
2400 NQN1=NQN
2500 NQK1=NQK
2600 IF(LIMQN.LE.LIMQK)GO TO 3200
2700 NQN1=NQK
2800 NQK1=NQN
2900 ITEMP=LIMQN
3000 LIMQN=LIMQK
3100 LIMQK=ITEMP
3200 IF(LIMQN.GT.0)GO TO 3600
3300 IF(LIMQK.EQ.0)GO TO 4600
3400 LIMAX=FIRSTX-1
3500 GO TO 4000
3600 CALL SETPQN(FIRSTX,LIMQK,NQN1,LIMQN)
3700 LIMAX=LIMQK
3800 CALL SETEPS(VARFUN(NQK1),1)
3900 IF(LIMAX.EQ.JCOL)GO TO 4100
4000 CALL SETPQN(LIMAX+1,JCOL,NQK1,LIMQK)
4100 CALL WOTAST
4200 CALL WOTITL(MASS(1),2)
4300 CALL WOTITL(MASTIT(1),1)
4400 CALL EXPROT(I,JROW,1)
4500 CONTINUE
4600 CONTINUE
4700 CONTINUE
     RETURN
     END
$IBFTC EQ1217  LIST
      SUBROUTINE EQ1217(I)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),MASS(5),MASTIT(6),TRAN(3,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(748),MASS(1)),(CM(1141),ROTNU
     1M),(CM(1379),FIRSTX),(CM(1409),MASTIT(1)),(CM(1420),LASTX),(CM(143
     20),TRAN(1,1))
      INTEGER TRAN,FIRSTX,ROTNUM
 100 DO 2400 JCOL=FIRSTX,LASTX
 200 DO 2400 JROW=1,3
 300 IF(IABS(TRAN(JROW,JCOL)).NE.1)GO TO 2400
```

```
 400 DO 2300 JOMEGA=1,3
 500 CALL FINDNQ(JOMEGA+3,NQJ)
 600 IF(NQJ.LT.0)GO TO 2300
 700 DO 2200 LOMEGA=JOMEGA,4
 800 ROTNUM=0
 900 IF(LOMEGA.EQ.4)GO TO 1500
1000 CALL FINDNQ(LOMEGA+3,NQL)
1100 IF(NQL.LT.0)GO TO 2200
1200 CALL SETEPP(JOMEGA,LOMEGA)
1300 CALL SETPRN(1,JROW,JCOL,0,0,1,1,NQJ,NQL,4,-2)
1400 GO TO 1700
1500 CALL SETPRN(1,JROW,JCOL,0,0,1,1,NQJ,1,3,-2)
1600 CALL SETEPS(JOMEGA,1)
1700 CALL WOTAST
1800 CALL WOTITL(MASS(1),2)
1900 CALL WOTITL(MASTIT(1),1)
2000 CALL SETPQN(FIRSTX,JCOL,1,0)
2100 CALL EXPROT(I,JROW,1)
2200 CONTINUE
2300 CONTINUE
2400 CONTINUE
2500 RETURN
     END
$IBFTC EQ1116  LIST
     SUBROUTINE EQ1116(I)
     COMMON CM
     DIMENSIONCM(1700),COMON(325,9),ALPH(46),MASS(5),MASTIT(6),OUTWRD(1
    180),VARTIT(6,30)
     EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(4),LD),(CM(
    1748),MASS(1)),(CM(775),NOFVAR),(CM(779),OUTWRD(1)),(CM(1141),ROTNU
    2M),(CM(1176),VARTIT(1,1)),(CM(1409),MASTIT(1)),(CM(747),LOCIST)
     INTEGER OUTWRD,ROTNUM
 100 OUTWRD(2)=LD
 200 LOCIST=3
 300 CALL WOTITL(VARTIT(1,NOFVAR),1)
 400 CALL WOTAST
 500 CALL WOTITL(MASS(1),2)
 600 CALL WOTITL(MASTIT(1),1)
 700 CALL PRNTRO(0,1,2)
 800 DO 2200 NOFU=1,3
 900 CALL FINDNQ(20+NOFU,NQU)
1000 IF(NQU.LE.0)GO TO 2200
1100 DO 2100 JOMEGA=1,3
1200 CALL FINDNQ(3+JOMEGA,NQJ)
1300 IF(NQJ.LE.0)GO TO 2100
1400 ROTNUM=0
1500 CALL SETEPS(JOMEGA,1)
1600 CALL SETPRN(0,1,1,0,0,1,1,NQU,NQJ,4,-2)
1700 CALL WOTAST
1800 CALL WOTITL(MASS(1),2)
1900 CALL WOTITL(MASTIT(1),1)
2000 CALL EXPROT(I,NOFU,1)
2100 CONTINUE
2200 CONTINUE
```

```
 2300 RETURN
      END
$IBFTC EQ13     LIST
      SUBROUTINE EQ13(I)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),MASS(5),MASTIT(6),OUTWRD(180),TRAN(
     13,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(32),L2),(CM
     1(748),MASS(1)),(CM(777),NVAR),(CM(779),OUTWRD(1)),(CM(1141),ROTNUM
     2),(CM(1379),FIRSTX),(CM(1409),MASTIT(1)),(CM(1420),LASTX),(CM(1430
     3),TRAN(1,1))
      INTEGER FIRSTX,TRAN,ROTNUM,OUTWRD
  100 DO 2100 JCOL1=FIRSTX,LASTX
  200 DO 2100 JROW1=1,3
  300 IF(IABS(TRAN(JROW1,JCOL1)).NE.1)GO TO 2100
  400 DO 2000 NQN=1,NVAR
  500 CALL TESTQM(JROW1,JCOL1,NQN,LIMQN,FIRSTX,LASTX)
  600 IF(LIMQN.EQ.-1)GO TO 2000
  700 DO 1900 JOMEGA=1,3
  800 CALL FINDNQ(3+JOMEGA,NQJ)
  900 IF(NQJ.LE.0)GO TO 1900
 1000 ROTNUM=0
 1100 CALL SETEPS(JOMEGA,1)
 1200 CALL SETPQN(FIRSTX,JCOL1,NQN,LIMQN)
 1300 OUTWRD(2)=L2
 1400 CALL SETPRN(LIMQN,JROW1,JCOL1,0,0,1,1,NQN,NQJ,4,3)
 1500 CALL WOTAST
 1600 CALL WOTITL(MASS(1),2)
 1700 CALL WOTITL(MASTIT(1),1)
 1800 CALL EXPROT(I,JROW1,I)
 1900 CONTINUE
 2000 CONTINUE
 2100 CONTINUE
 2200 RETURN
      END
$IBFTC IRNUM    LIST
      FUNCTION IRNUM(ROTNUM,FIRSTX,MIN)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ROT(33)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1421),ROT(1))
      INTEGER ROTNUM,ROT,FIRSTX
  100 K=0
  200 MINM1=MIN-1
  300 DO 500 I=FIRSTX,MINM1
  400 IF(ROT(I).EQ.0) K=K+1
  500 CONTINUE
  600 IRNUM=ROTNUM+FIRSTX+K-MIN
  700 RETURN
      END
$IBFTC FTP34    LIST
      SUBROUTINE FTP34
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8),VARLOC(30)
```

```
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(775),NOFVAR),(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1376),FIRS
     2TC),(CM(1377),FIRSTR),(CM(1378),FIRSTS),(CM(1415),LASTC),(CM(1416)
     3,LASTR),(CM(1417),LASTS),(CM(1430),TRAN(1,1)),(CM(1663),VARLOC(1))
      INTEGER VARLOC,TRAN,ROTNUM,FIRSTS,FIRSTR
  100 CALL MQTYPE(NOFVAR,NTYPQM)
  150 IF(NTYPQM.EQ.5)GO TO 3700
  175 PRDCON=-1.0
  200 IF(NTYPQM.LE.1)GO TO 800
  300 LIMCLO=VARLOC(NOFVAR)
  400 LIMCHI=LASTR
  500 LIMRLO=1
  600 LIMRHI=3
  700 GO TO 1200
  800 LIMRLO=VARLOC(NOFVAR)/10
  900 LIMRHI=LIMRLO
 1000 LIMCLO=MOD(VARLOC(NOFVAR),10)
 1100 LIMCHI=LIMCLO
 1200 DO 3600 ICOL=LIMCLO,LIMCHI
 1300 DO 3600 IROW=LIMRLO,LIMRHI
 1400 IF(TRAN(IROW,ICOL).EQ.0)GO TO 3600
 1500 ROTNUM=0
 1600 CALL SETDEL(7)
 1700 CALL TESTQM(IROW,ICOL,NOFVAR,LIMQM,FIRSTC,LASTR)
 1800 IF(LIMQM.LT.0)GO TO 3600
 1900 IF(NTYPQM.EQ.0)GO TO 2100
 2000 GO TO(2700,2700,2300,2700),NTYPQM
 2100 IF(ICOL.LT.LASTC) CALL SETPQN(ICOL+1,LASTC,1,0)
 2200 GO TO 2800
 2300 CALL SETXQM(LIMQM,ICOL,LIMQM)
 2400 ROTNUM=ROTNUM-1
 2500 IF(LIMQM.LT.LASTC) CALL SETPQN(LIMQM+1,LASTC,1,0)
 2600 GO TO 2800
 2700 CALL SETXQM(FIRSTR,ICOL,LIMQM)
 2800 CALL SETPQN(FIRSTS,LASTS,1,0)
 2900 DO 3500 IF=1,3
 3000 IF(TRAN(IF,LASTS).EQ.0)GO TO 3500
 3100 ICOUNT=0
 3200 CALL SETPRN(1,IF,LASTS,LIMQM,1,IROW,ICOL,NOFVAR,1,1,7)
 3300 CALL EXPROT(IROW,IF,2)
 3400 IF(ICOUNT.GT.0) CALL PRINT3( ICOUNT,0,3,3,L,3)
 3500 CONTINUE
 3600 CONTINUE
 3700 RETURN
      END
$IBFTC MQTYPE  LIST
      SUBROUTINE MQTYPE(MQ,NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1377),FIRSTR),(CM(1416),LASTR
     1),(CM(1663),VARLOC(1))
      INTEGER VARLOC,FIRSTR
  100 IR=VARLOC(MQ)/10
  200 IC=MOD(VARLOC(MQ),10)
```

```
  300 IF(IC.GE.FIRSTR)GO TO 700
  400 NTYPE=0
  500 IF(IR.EQ.0) NTYPE=3
  600 GO TO 1300
  700 IF(IC.LE.LASTR)GO TO 1100
  800 NTYPE=5
  900 IF(IR.EQ.0) NTYPE=4
 1000 GO TO 1300
 1100 NTYPE=1
 1200 IF(IR.EQ.0) NTYPE=2
 1300 RETURN
      END
$IBFTC FTP3A    LIST
      SUBROUTINE FTP3A(NQK,I1,NT,I4,NWT,LOC1)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1377),FIRSTR),(CM(1378),FIR
     2STS),(CM(1416),LASTR),(CM(1417),LASTS),(CM(1430),TRAN(1,1))
      INTEGER FIRSTR,FIRSTS,TRAN,ROTNUM
  100 CALL MQTYPE(NQK,NTYPQM)
  110 PRD=PRDCON
  200 IF(NTYPQM.EQ.3) RETURN
  300 DO 6600 ICOL1=FIRSTR,LASTS
  400 DO 6600 IROW1=1,3
  500 IF(TRAN(IROW1,ICOL1).EQ.0) GO TO 6600
  600 DO 6500 ICOL2=ICOL1,LASTS
  700 DO 6500 IROW2=1,3
 1000 NTYPRS=1
 1100 PRDCON=PRD
 1200 ICOUNT=0
 1210 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
 1220 IF(NTEST.LT.0) GO TO 6500
 1300 JCOL1=ICOL1
 1400 JCOL2=ICOL2
 1500 JROW1=IROW1
 1600 JROW2=IROW2
 1700 IF(NTYPRS.LE.2) GO TO 2100
 1800 LOLIM1=FIRSTS
 1900 CALL TESTQM(JROW1,JCOL1,NQK,LIMQM,FIRSTS,LASTS)
 2000 GO TO 2300
 2100 LOLIM1=FIRSTR
 2200 CALL TESTQM(JROW1,JCOL1,NQK,LIMQM,FIRSTR,LASTR)
 2300 IF(LIMQM.LT.0) GO TO 5100
 2400 IF(NTYPRS.EQ.2*(NTYPRS/2))GO TO 2800
 2500 LOLIM2=FIRSTR
 2600 IF(JCOL2.GT.LASTR) GO TO 5100
 2700 GO TO 3000
 2800 LOLIM2=FIRSTS
 2900 IF(JCOL2.LT.FIRSTS) GO TO 5100
 3000 ROTNUM=0
 3100 CALL SETDEL(7)
 3200 CALL SETXQM(LOLIM2,JCOL2,0)
```

```
3300 CALL SETPQN(LOLIM1,JCOL1,NQK,LIMQM)
4900 CALL SETPRN(LIMQM,JROW1,JCOL1,1,1,JROW2,JCOL2,NQK,1,NT,LOC1)
5000 CALL EXPROT(JROW2,JROW1,2)
5100 IF(NTEST.EQ.1)GO TO 5800
5200 NTEST=1
5300 JROW1=IROW2
5400 JROW2=IROW1
5500 JCOL1=ICOL2
5600 JCOL2=ICOL1
5700 GO TO 1700
5800 NTYPRS=NTYPRS+1
5900 IF(NTYPRS.EQ.3)GO TO 1300
6000 IF(ICOUNT.GT.0) CALL PRINT3( ICOUNT,I1,0,0,I4,NWT)
6100 IF(NTYPRS.EQ.4)GO TO 1100
6200 IF(NTYPRS.EQ.5)GO TO 6500
6300 PRDCON=-PRD
6400 GO TO 1200
6500 CONTINUE
6600 CONTINUE
6700 RETURN
     END
$IBFTC FTP35    LIST
     SUBROUTINE FTP35
     COMMON CM
     DIMENSIONCM(1700),COMON(325,9)
     EQUIVALENCE(CM(1376),COMON(1,1)),(CM(775),NOFVAR),(CM(977),PRDCON)
    1,(CM(1358),WTAPE3)
     INTEGER WTAPE3
 100 REWIND WTAPE3
 200 NO=0
 300 PRDCON=1.0
 400 CALL FTP3A(NOFVAR,3,1,NO,WTAPE3,7)
 500 REWIND WTAPE3
 600 IF(NO.EQ.0) RETURN
 700 CALL FTP5(NO)
 800 REWIND WTAPE3
 900 RETURN
     END
$IBFTC FTP5     LIST
     SUBROUTINE FTP5(NO)
     COMMON CM
     DIMENSIONCM(1700),COMON(325,9),CODWRD(300),OUTWRD(180),VARLOC(30)
     EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(747),LOC1S
    1T),(CM(777),NVAR),(CM(779),OUTWRD(1)),(CM(977),PRDCON),(CM(1358),W
    2TAPE3),(CM(1359),WTAPE4),(CM(1377),FIRSTR),(CM(1417),LASTS),(CM(16
    363),VARLOC(1))
     INTEGER WTAPE3,CODWRD,FIRSTR,WTAPE4,VARLOC
 700 DO 1700 NONO=1,NO
 800 READ (WTAPE3) NCOUNT,LOC11,OUTWRD
 900 DO 1700 NONCO=1,NCOUNT
1000 LOC1ST=LOC11
1050 READ(WTAPE3)PRDCON,CODWRD(1)
1100 CALL PRNTRO(CODWRD(1),-1,2)
1200 LOC12=LOC1ST
```

```
 1300 DO 1700 NQN=1,NVAR
 1400 CALL NQNCK(NQN,FIRSTR,LASTS,ITEST)
 1500 IF(ITEST.LT.0)GO TO 1700
 1600 CALL FTP3A(NQN,0,2,0,WTAPE4,LOC12)
 1700 CONTINUE
 1800 RETURN
      END
$IBFTC FTP2A    LIST
      SUBROUTINE FTP2A(I1,I4,NT,NWT,LOC1)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8),VARLOC(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(775),NOFVAR),(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1377),FIRS
     2TR),(CM(1378),FIRSTS),(CM(1415),LASTC),(CM(1416),LASTR),(CM(1417),
     3LASTS),(CM(1430),TRAN(1,1)),(CM(1663),VARLOC(1))
      INTEGER VARLOC,FIRSTR,FIRSTS,TRAN,ROTNUM
  100 NCOLD=VARLOC(NOFVAR)
  200 IF(NCOLD.LT.FIRSTR) GO TO 4200
  300 IF(NCOLD.LE.LASTR)GO TO 900
  400 LASTX=LASTS
  500 LO1=FIRSTS
  600 LO2=FIRSTR
  700 NHI2=LASTR
  800 GO TO 1300
  900 LASTX=LASTR
 1000 LO1=FIRSTR
 1100 LO2=FIRSTS
 1200 NHI2=LASTS
 1300 NHI1=NCOLD-1
 1400 DO 4100 ICOL1=NCOLD,LASTX
 1500 DO 4100 IROW1=1,3
 1600 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 4100
 1800 IF(LO1.GT.NHI1)GO TO 2300
 1900 PRDCON=1.0
 1950 NTIMES=1
 2000 LO=LO1
 2100 NHI=NHI1
 2200 GO TO 2700
 2300 PRDCON=-1.0
 2400 LO=LO2
 2500 NHI=NHI2
 2600 NTIMES=2
 2700 DO 3900 ICOL2=LO,NHI
 2800 DO 3900 IROW2=1,3
 2900 IF(TRAN(IROW2,ICOL2).EQ.0)GO TO 3900
 3000 ROTNUM=0
 3100 ICOUNT=0
 3200 CALL SETDEL(7)
 3300 CALL SETXQM(LO,ICOL2,0)
 3400 CALL SETPQN(NCOLD,ICOL1,NOFVAR,NCOLD)
 3500 CALL EXPROT(IROW2,IROW1,2)
 3600 IF(ICOUNT.EQ.0)GO TO 3900
 3700 CALL SETPRN(1,IROW1,ICOL1,1,1,IROW2,ICOL2,NOFVAR,0,NT,LOC1)
```

```
 3800 CALL PRINT3( ICOUNT,I1,0,0,I4,NWT)
 3900 CONTINUE
 4000 IF(NTIMES.EQ.1)GO TO 2300
 4100 CONTINUE
 4200 RETURN
      END
$IBFTC FTP24    LIST
      SUBROUTINE FTP24
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8),VARFUN(30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(775),NOFVAR),(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1146),VARF
     2UN(1)),(CM(1377),FIRSTR),(CM(1378),FIRSTS),(CM(1415),LASTC),(CM(14
     316),LASTR),(CM(1417),LASTS),(CM(1419),NODCOS),(CM(1430),TRAN(1,1))
      INTEGER ROTNUM,FIRSTS,FIRSTR,TRAN
  100 LO=NODCOS
  150 PRDCON=1.0
  200 ROTNUM=0
  300 CALL SETXQM(FIRSTS,LASTS,0)
  400 IF(NODCOS.GE.FIRSTR)GO TO 800
  500 LO=FIRSTR
  600 CALL SETEPS(VARFUN(NOFVAR),1)
  700 GO TO 900
  800 CALL SETPQN(FIRSTR,NODCOS,NOFVAR,NODCOS)
  900 CALL SETDEL(7+MINUS)
 1100 DO 2400 ICOL=LO,LASTR
 1200 CALL SETROT(ICOL,1)
 1300 DO 2300 IFROW=1,3
 1400 IF(TRAN(IFROW,LASTS).EQ.0)GO TO 2300
 1500 DO 2200 IXROW=1,3
 1600 IF(TRAN(IXROW,ICOL).EQ.0)GO TO 2200
 1700 ICOUNT=0
 1800 CALL EXPROT(IFROW,IXROW,2)
 1900 IF(ICOUNT.EQ.0)GO TO 2200
 2000 CALL SETPRN(1,IFROW,LASTS,1,1,IXROW,ICOL,1,1,1,7)
 2100 CALL PRINT3( ICOUNT,0,1,2,3,4)
 2200 CONTINUE
 2300 CONTINUE
 2400 CONTINUE
 2500 RETURN
      END
$IBFTC FTP25    LIST
      SUBROUTINE FTP25
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1358),WTAPE3)
      INTEGER WTAPE3
  100 REWIND WTAPE3
  200 NO=0
  400 CALL FTP2A(3,NO,1,WTAPE3,7)
  500 REWIND WTAPE3
  600 IF(NO.EQ.0)GO TO 900
  700 CALL FTP5(NO)
  800 REWIND WTAPE3
```

```
  900 RETURN
      END
$IBFTC LOCUP    LIST
      FUNCTION LOCUP(N)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),OUTWRD(180)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(747),LOC1ST),(CM(779),OUTWRD(
     11))
      INTEGER OUTWRD
  100 OUTWRD(LOC1ST)=N
  200 LOCUP=LOC1ST+1
  300 RETURN
      END
$IBFTC PRNT5    LIST
      SUBROUTINE PRNT5
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),OUTWRD(180)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(46),LBL),(C
     1M(745),LINAUX),(CM(747),LOC1ST),(CM(779),OUTWRD(1)),(CM(1359),WTAP
     2E4)
      INTEGER OUTWRD,WTAPE4,ALPH
  100 IF(LOC1ST.LT.90) RETURN
  200 WRITE(WTAPE4,300)(OUTWRD(I),I=2,121)
  300 FORMAT(120A1)
  400 DO 500 I=2,121
  500 OUTWRD(I)=LBL
  600 LINAUX=LINAUX+1
  610 IF(LOC1ST.GT.122) GO TO 700
  620 LOC1ST=9
  630 GO TO 100
  700 DO 800 I=122,LOC1ST
  800 OUTWRD(I-110)=OUTWRD(I)
  900 LOC1ST=LOC1ST-110
 1000 GO TO 100
      END
$IBFTC SETAUX   LIST
      SUBROUTINE SETAUX(NTYPE)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),ALPH(46),CODWRD(300),COEF(300),MAST
     1IT(6),OUTWRD(180),ROTTIT(6,8),TITLES(6,8,4),TRNTIT(6,8,3),TRAN(3,8
     2),VARFUN(30),VARLOC(30),VARTIT(6,30)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(4),LD),(ALP
     1H(6),LF),(ALPH(11),LK),(ALPH(12),LL),(ALPH(15),LO),(ALPH(16),LP),(
     2ALPH(19),LS),(ALPH(26),LZ),(ALPH(27),LEQ),(ALPH(28),LPO),(ALPH(29)
     3,LPC),(ALPH(40),LAS),(ALPH(41),LCOM),(ALPH(42),LPER),(ALPH(43),LSL
     4ASH),(ALPH(44),LMI),(ALPH(45),LPL),(ALPH(46),LBL),(CM(47),CODWRD(1
     5)),(CM(347),COEF(1)),(CM(651),ICOUNT),(CM(745),LINAUX),(CM(747),LO
     6C1ST),(CM(753),MINUS),(CM(773),NOFONT),(CM(776),NOMONT),(CM(777),N
     7VAR),(CM(779),OUTWRD(1)),(CM(959),MORF),(CM(1141),ROTNUM),(CM(1146
     8),VARFUN(1)),(CM(1176),VARTIT(1,1)),(CM(1359),WTAPE4),(CM(1377),FI
     9RSTR),(CM(1378),FIRSTS),(CM( 1409 ),MASTIT(1)),(CM(1416),LASTR),(CM(
     11417),LASTS),(CM(1418),NFORTP),(CM(1419),NODCOS),(CM(1430),TRAN(1,
     21)),(CM(1462),TITLES(1,1,1)),(CM(1462),ROTTIT(1,1)),(CM(1510),TRNT
```

```
   3IT(1,1,1)),(CM(1663),VARLOC(1)),(CM(977),PRDCON)
    INTEGER OUTWRD,ALPH,WTAPE4,ROTNUM,CODWRD,TRAN,FIRSTR,FIRSTS
    INTEGER TRNTIT
    INTEGER VARFUN,VARTIT,VARLOC
100 DO 110 I=2,121
110 OUTWRD(I)=LBL
115 NOFPS=ICOUNT
120 NS=NOMONT+NOFONT+1
130 IF(LINAUX.GT.0.OR.NTYPE.EQ.2) GO TO 160
135 LOC1ST=122
136 CALL PRNT5
140 WRITE(WTAPE4,145)
145 FORMAT(61HTHESE ARE THE AUXILIARY EQUATIONS TO BE USED WITH THIS S
   1ET.     59X)
150 LINAUX=2
160 LOC1ST=2
170 GO TO (200,2000,4000,4000,200),NTYPE
200 LOC1ST=LOCUP(LD)
201 IF(LFIRST.NE.1) NS1=NS
202 LFIRST=1
203 NS=NS1
220 LOC1ST=LOCUP(LPO)
230 LOC1ST=LOCUP(LZ)
240 LOC1ST=LOCUP(LP)
250 LOC1ST=LOCUP(LO)
260 LOC1ST=LOCUP(LS)
270 LOC1ST=LOCUP(ALPH(NS))
280 LOC1ST=LOCUP(LBL)
290 LOC1ST=LOCUP(LPC)
300 LOC1ST=LOCUP(LEQ)
305 IF(NTYPE.EQ.5) GO TO 490
310 DO 480 J=1,3
320 OUTWRD(9)=ALPH(J+30)
330 LOC1ST=12
340 DO 450 K=1,3
350 IF(TRAN(K,NODCOS-1).NE.2) GO TO 450
360 IF(LOC1ST.NE.12) LOC1ST=LOCUP(LPL)
370 CALL WOTITL(ROTTIT(1,NODCOS),2)
380 LOC1ST=LOCUP(LPO)
390 LOC1ST=LOCUP(ALPH(J+30))
400 LOC1ST=LOCUP(LCOM)
410 LOC1ST=LOCUP(ALPH(K+30))
420 LOC1ST=LOCUP(LPC)
430 CALL WOTAST
440 CALL WOTITL(TRNTIT(1,NODCOS-1,K),1)
450 CONTINUE
460 WRITE(WTAPE4,470)(OUTWRD(I),I=2,121)
470 FORMAT(120A1)
480 LINAUX=LINAUX+1
481 DO 486 I=1,3
482 NVAR=NVAR+1
483 DO 484 J=1,5
484 VARTIT(J,NVAR)=OUTWRD(J+3)
485 VARFUN(NVAR)=15
```

```
486 VARTIT(6,NVAR)=ALPH(I+30)
490 DO 520 I=1,3
500 DO 510 J=1,5
510 TRNTIT(J,NODCOS-1,I)=OUTWRD(J+3)
520 TRNTIT(6,NODCOS-1,I)=ALPH(I+30)
530 IF(MORF.NE.LF) GO TO 1000
540 DO 630 NV=1,NVAR
550 IF(VARFUN(NV).NE.15) GO TO 630
560 DO 580 I=1,5
570 IF(VARTIT(I,NV).NE.OUTWRD(I+3)) GO TO 630
580 CONTINUE
590 NNN=NV-1
600 DO 620 I=1,3
610 NNN=NNN+1
620 VARLOC(NNN)=10*I+NODCOS-1
625 GO TO 640
630 CONTINUE
640 DO 650 I=1,3
650 TRAN(I,NODCOS-1)=1
1000 ICOUNT=NOFPS
1100 RETURN
2000 IFIRST=2
2010 GO TO 1000
4000 LOCIST=LOCUP(LK)
4005 FORM=PRDCON
4010 LOCIST=LOCUP(ALPH(NFORTP+30))
4020 LOCIST=LOCUP(ALPH(NS))
4030 LOCIST=LOCUP(LSLASH)
4040 LOCIST=LOCUP(LF)
4050 LOCIST=LOCUP(LL)
4060 LOCIST=LOCUP(LEQ)
4070 CALL WOTITL(MASTIT(1),1)
4080 LOCIST=LOCUP(LSLASH)
4090 LOCIST=LOCUP(LPO)
4100 IF(NTYPE.EQ.3) LOCIST=LOCUP(LPO)
4110 DO 4120 I=1,6
4120 MASTIT(I)=OUTWRD(I+1)
4130 PRDCON=1.0
4140 DO 4490 ICOL1=FIRSTR,LASTS
4150 DO 4490 IROW1=1,3
4160 IF(TRAN(IROW1,ICOL1).EQ.0)GO TO 4490
4170 DO 4480 ICOL2=ICOL1,LASTS
4180 DO 4480 IROW2=1,3
4190 CALL TESTRN(IROW1,ICOL1,IROW2,ICOL2,ITEST,NTEST)
4200 IF(NTEST.LT.0)GO TO 4480
4210 ICOUNT=0
4220 JROW1=IROW1
4222 JCOL1=ICOL1
4224 JROW2=IROW2
4226 JCOL2=ICOL2
4230 ROTNUM=0
4235 IF(JCOL1.GT.LASTR)GO TO 4250
4240 CALL SETXQM(FIRSTR,JCOL1,0)
```

```
4245 GO TO 4260
4250 CALL SETDEL(MINUS)
4255 CALL SETXQM(FIRSTS,JCOL1,0)
4260 IF(JCOL2.GT.LASTR)GO TO 4270
4265 CALL SETPQN(FIRSTR,JCOL2,0)
4268 GO TO 4280
4270 CALL SETDEL(MINUS)
4275 CALL SETPQN(FIRSTS,JCOL2,0,0)
4280 CALL EXPROT(JROW1,JROW2,2)
4290 IF(NTEST.EQ.1) GO TO 4320
4300 NTEST=1
4310 JROW1=IROW2
4312 JROW2=IROW1
4314 JCOL1=ICOL2
4316 JCOL2=ICOL1
4318 GO TO 4230
4320 CALL TRIG(ICOUNT)
4330 IF(ICOUNT.EQ.0)GO TO 4480
4340 DO 4470 L=1,ICOUNT
4350 N=1
4360 IF(COEF(L).GT.0.0) GO TO 4390
4370 COEF(L)=-COEF(L)
4380 N=0
4390 LOT=LOC1ST
4400 IF(COEF(L).LE.1.0) GO TO 4420
4410 CALL WOTAST
4414 N1=COEF(L)
4416 LOC1ST=LOCUP(ALPH(N1+30))
4420 CALL PRNTRO(CODWRD(L),0,1)
4430 CALL WOTAST
4440 CALL WOTITL(TRNTIT(1,ICOL1,IROW1),1)
4450 CALL WOTAST
4455 CALL WOTITL(TRNTIT(1,ICOL2,IROW2),1)
4460 OUTWRD(LOT)=ALPH(N+44)
4470 CALL PRNT5
4480 CONTINUE
4490 CONTINUE
4500 LOC1ST=LOCUP(LPC)
4510 IF(NTYPE.EQ.4) GO TO 4580
4520 LOC1ST=LOCUP(LAS)
4530 LOC1ST=LOCUP(LAS)
4540 LOC1ST=LOCUP(ALPH(31))
4550 LOC1ST=LOCUP(LPER)
4560 LOC1ST=LOCUP(ALPH(35))
4570 LOC1ST=LOCUP(LPC)
4580 CALL PRNT5
4590 LOC1ST=122
4600 CALL PRNT5
4605 PRDCON=FORM
4610 GO TO 1000
     END
$IBFTC FTP14   LIST
     SUBROUTINE FTP14(I)
     COMMON CM
```

```
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(753),MINUS),(CM(1141),ROTNUM)
     1,(CM(1377),FIRSTR),(CM(1378),FIRSTS),(CM(1416),LASTR),(CM(1417),LA
     2STS),(CM(1419),NODCOS),(CM(1430),TRAN(1,1))
      INTEGER ROTNUM,FIRSTS,FIRSTR,TRAN
  100 IF(NODCOS.GT.LASTR) GO TO 1000
  200 ROTNUM=0
  250 CALL SETDEL(MINUS)
  300 CALL SETXQM(FIRSTS,LASTS,0)
  400 IF(NODCOS.GE.FIRSTR) CALL SETROT(NODCOS,1)
  500 DO 900 IROW=1,3
  600 IF(TRAN(IROW,LASTS).EQ.0)GO TO 900
  700 CALL SETPRN(1,IROW,LASTS,0,0,0,0,1,1,1,-2)
  800 CALL EXPROT(IROW,1,1)
  900 CONTINUE
 1000 RETURN
      END
$IBFTC FTP15    LIST
      SUBROUTINE FTP15(I)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1358),WTAPE3)
      INTEGER WTAPE3
  100 REWIND WTAPE3
  200 NO=0
  300 CALL FTP1A(I,3,NO)
  400 REWIND WTAPE3
  500 IF(NO.EQ.0)GO TO 800
  600 CALL FTP5(NO)
  700 REWIND WTAPE3
  800 RETURN
      END
$IBFTC FTP1A    LIST
      SUBROUTINE FTP1A(I,I1,NO)
      COMMON CM
      DIMENSIONCM(1700),COMON(325,9),TRAN(3,8)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(651),ICOUNT),(CM(753),MINUS),
     1(CM(977),PRDCON),(CM(1141),ROTNUM),(CM(1358),WTAPE3),(CM(1376),FIR
     2STC),(CM(1377),FIRSTR),(CM(1378),FIRSTS),(CM(1415),LASTC),(CM(1416
     3),LASTR),(CM(1417),LASTS),(CM(1419),NODCOS),(CM(1430),TRAN(1,1))
      INTEGER TRAN,FIRSTC,FIRSTR,FIRSTS,WTAPE3,ROTNUM
  100 PRDCON=1.0
  200 IFC=FIRSTC
  300 ILC=LASTC
  400 IFR=FIRSTR
  500 ILR=LASTR
  600 IFS=FIRSTS
  700 ILS=LASTS
  800 IF(NODCOS.GT.LASTR) GO TO 1200
  900 IFR=NODCOS-1
 1000 ILC=IFR-1
 1100 GO TO 1700
 1200 IFR=FIRSTS
```

```
1300 ILR=LASTS
1400 IFS=FIRSTR
1500 ILS=LASTR
1600 IF(NODCOS.NE.IFR+1) RETURN
1700 ROTNUM=0
1800 CALL SETDEL(7)
1825 IF(NODCOS.GE.ILR) GO TO 2900
1850 NDCOS1=NODCOS+1
1900 DO 2800 ICOL=NDCOS1,ILR
2000 CALL SETROT(ICOL,1)
2100 DO 2800 IROW=1,3
2200 IF(TRAN(IROW,ICOL).EQ.0)GO TO 2800
2300 ICOUNT=0
2400 CALL EXPROT(I,IROW,2)
2500 IF(ICOUNT.EQ.0)GO TO 2800
2600 CALL SETPRN(0,0,0,1,1,IROW,ICOL,0,0,1,7)
2700 CALL PRINT3 (ICOUNT,I1,0,0,NO,WTAPE3)
2800 CONTINUE
2900 ROTNUM=1
3000 CALL SETROT(NODCOS,2)
3100 DO 3700 IROW=1,3
3200 ICOUNT=0
3300 CALL EXPROT(I,IROW,2)
3400 IF(ICOUNT.EQ.0)GO TO 3700
3500 CALL SETPRN(1,IROW,IFR,0,0,0,0,0,0,1,7)
3600 CALL PRINT3 (ICOUNT,I1,0,0,NO,WTAPE3)
3700 CONTINUE
3800 CALL SETDEL(MINUS)
3900 DO 4800 ICOL=IFS,ILS
4000 CALL SETROT(ICOL,1)
4100 DO 4800 IROW=1,3
4200 IF(TRAN(IROW,ICOL).EQ.0)GO TO 4800
4300 ICOUNT=0
4400 CALL EXPROT(I,IROW,2)
4500 IF(ICOUNT.EQ.0)GO TO 4800
4600 CALL SETPRN(1,IROW,ICOL,0,0,0,0,0,0,1,7)
4700 CALL PRINT3 (ICOUNT,I1,0,0,NO,WTAPE3)
4800 CONTINUE
4900 RETURN
     END
$IBFTC FTP36    LIST
     SUBROUTINE FTP36
     COMMON CM
     DIMENSIONCM(1700),COMON(325,9),ALPH(46),MASTIT(6),OUTWRD(180),VARL
    1OC(30),VARTIT(6,30)
     EQUIVALENCE(CM(1376),COMON(1,1)),(CM(1),ALPH(1)),(ALPH(4),LD),(ALP
    1H(44),LMI),(CM(747),LOC1ST),(CM(775),NOFVAR),(CM(779),OUTWRD(1)),(
    2CM(1176),VARTIT(1,1)),(CM(1361),NTAPE6),(CM(1409),MASTIT(1)),(CM(1
    3418),NFORTP),(CM(1663),VARLOC(1))
     INTEGER VARLOC,OUTWRD
 100 IF(VARLOC(NOFVAR).EQ.0) GO TO 1000
 200 LOC1ST=1
 300 LOC1ST=LOCUP(LMI)
 400 CALL WOTITL(MASTIT(1),1)
```

```
  500 CALL WOTAST
  600 IF(NFORTP.EQ.6) LOC1ST=LOCUP(LD)
  700 CALL WOTITL(VARTIT(1,NOFVAR),1)
  800 WRITE(NTAPE6,900)(OUTWRD(I-1),I=2,LOC1ST)
  900 FORMAT (1X,99A1)
 1000 RETURN
      END
$IBFTC W34RW    LIST
      SUBROUTINE W34RW(INORWT,NOS,NOE,NWT)
      COMMON CM
      DIMENSIONCO(100,2),NCD(100,2)
      DIMENSIONCM(1700),COMON(325,9),CODWRD(300),COEF(300),WTAPE(6)
      EQUIVALENCE(CM(1376),COMON(1,1)),(CM(47),CODWRD(1)),(CM(347),COEF(
     11)),(CM(1356),WTAPE(1))
      INTEGER CODWRD,WTAPE
  100 NT=1
  110 IF(NWT.EQ.WTAPE(4)) NT=2
  120 IF(INORWT.EQ.2) GO TO 200
  130 DO 190 I=1,NOS
  140 IF(I.GT.100) GO TO 180
  150 COEF(I)=CO(I,NT)
  160 CODWRD(I)=NCD(I,NT)
  170 GO TO 190
  180 READ(NWT) COEF(I),CODWRD(I)
  190 CONTINUE
  195 GO TO 280
  200 DO 270 I=1,NOE
  210 NOS=NOS+1
  220 IF(NOS.GT.100) GO TO 260
  230 CO(NOS,NT)=COEF(I)
  240 NCD(NOS,NT)=CODWRD(I)
  250 GO TO 270
  260 WRITE(NWT) COEF(I),CODWRD(I)
  270 CONTINUE
  280 RETURN
      END
```

APPENDIX III

DERIVATION OF THE GRAVITATIONAL FORCING FUNCTION BETWEEN TWO ARBITRARILY SHAPED BODIES

An alternative method to that derived on page 11 of the report for the generalized force due to the gravitational attraction may also be derived. If the potential function is assumed to be U, then the term Q_m is actually $\frac{\partial U}{\partial q m}$. The problem is now to determine U for an inverse square attraction.

If an arbitrary two-body system may be represented by Figure 1 then, the position of any point in Body X (in terms of an orthogonal coordinate system whose origin is in Body Y) is

$$P_j = R_j + A_{ji} X_i$$

and any point in Body Y is

$$P'_j = B_{ji} Y_i .$$

The vector between these two arbitrary points is

$$S_j = R_j + A_{ji} X_i - B_{ji} Y_i .$$

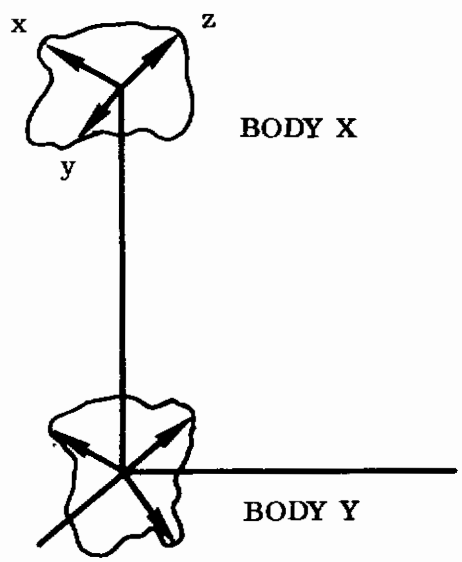

Figure 1. Coordinates of an Arbitrary Two-Body System

The scalar distance between the two points is

$$D = \sqrt{S_j S_j} = \sqrt{R_j R_j + 2R_j A_{ji} X_i - 2R_j B_{ji} Y_i + A_{ji} A_{jk} X_i X_k - 2A_{ji} B_{jk} X_i Y_k + B_{ji} B_{jk} Y_i Y_k}$$

However, as A and B are pure orthogonal rotations (in the cartesian coordinate system chosen),

$$D = \sqrt{R_j R_j + 2R_j A_{ji} X_i - 2R_i B_{ji} Y_i + X_i X_i - 2A_{ji} B_{jk} X_i Y_k + Y_i Y_i}$$

or

$$D = \sqrt{R_j R_j}\ \sqrt{1 + \frac{(2R_j A_{ji} X_i - 2R_j B_{ji} Y_i + X_i X_i - 2A_{ji} B_{jk} X_i Y_k + Y_i Y_i)}{\sum_\ell (R_\ell R_\ell)}}$$

The potential energy of the two body system will be

$$U = - K \int^X \int^Y \frac{dX\, dY}{D}$$

$$U = - K \int^X \int^Y \frac{dX\, dY}{\sqrt{R_j R_j}\ \sqrt{1 + Q}} .$$

As no loss in generality will result if $R = \begin{vmatrix} 0 \\ 0 \\ r \end{vmatrix}$

and $R_j R_j = r^2$, then

$$U = - \frac{K}{r} \int^X \int^Y \frac{dX\, dY}{(1+Q)^{1/2}} .$$

The binomial expansion of $(1 + Q)^{-1/2}$ is

$$(1 + Q)^{-1/2} = 1 - \frac{Q}{2} + \frac{3}{8} Q^2 - \frac{5}{16} Q^3 + \cdots$$

where in this case

$$Q = \frac{1}{r^2} (2R_j A_{ji} X_i - 2R_j B_{ji} Y_i + X_i X_i - 2A_{ji} B_{jk} X_i Y_k + Y_i Y_i)$$

$$\begin{aligned} Q^2 = \frac{1}{r^4} \Big(& 4r^2 A_{3i} A_{3k} X_i X_k - 8r^2 A_{3i} B_{3j} X_i Y_j + 4r A_{3i} X_i X_j X_j \\ & + 4r A_{3i} X_i Y_j Y_j - 8r A_{3i} A_{j\ell} B_{jk} X_i X_\ell Y_k + 4r^2 B_{3i} B_{3k} Y_i Y_k \\ & -4r B_{3i} X_j X_j Y_i - 4r B_{3i} Y_i Y_j Y_j + 8r B_{3i} A_{j\ell} B_{jk} X_\ell Y_i Y_k \\ & + X_i X_i X_j X_j + Y_i Y_i Y_j Y_j + 2 X_i X_i Y_j Y_j - 4 A_{ji} B_{jk} X_i X_\ell X_\ell Y_k \end{aligned}$$

$$- 4 A_{ji} B_{jk} X_i Y_k Y_\ell Y_\ell + 4 A_{ji} A_{\ell m} B_{jk} B_{\ell n} X_i X_m Y_k Y_n \Big)$$

$$Q^3 = Q\,Q^2$$

$$Q^4 = \cdots$$

Making these substitutions in the expression for U gives

$$U = -\frac{k}{r}\int^X\!\!\int^Y \left[1 - \frac{1}{r}\left(A_{3i}X_i - B_{3i}Y_i\right) + \frac{1}{r^2}\left(\frac{-X_iX_i}{2} - \frac{Y_iY_i}{2} + A_{ji}B_{jk}X_iY_k + \frac{3}{2}A_{3i}A_{3k}X_iX_k + \frac{3}{2}B_{3i}B_{3k}Y_iY_k - 3A_{3i}B_{3j}X_iY_j \right) + \frac{1}{r^3}\left(\frac{3}{2}A_{3i}X_iX_jX_j + \frac{3}{2}A_{3i}X_iY_jY_j - 3A_{3i}A_{j\ell}B_{jk}X_iX_\ell Y_k - \frac{3}{2}B_{3i}X_jX_jY_i - \frac{3}{2}B_{3i}Y_iY_jY_j + 3B_{3i}A_{j\ell}B_{jk}X_\ell Y_iY_k - \frac{5}{2}A_{3i}A_{3j}A_{3k}X_iX_jX_k + \frac{15}{2}A_{3i}A_{3j}B_{3k}X_iX_jY_k - \frac{15}{2}A_{3i}B_{3j}B_{3k}X_iY_jY_k + \frac{5}{2}B_{3i}B_{3j}B_{3k}Y_iY_jY_k \right) + \frac{1}{r^4}\left(\frac{3}{8}X_iX_iX_jX_j + \frac{3}{4}X_iX_iY_jY_j - \frac{3}{2}A_{ji}B_{jk}X_iX_\ell X_\ell Y_k + \frac{3}{8}Y_iY_iY_jY_j - \frac{3}{2}A_{ji}B_{jk}X_iY_kY_\ell Y_\ell + \frac{3}{2}A_{ji}A_{\ell m}B_{jk}B_{\ell n}X_iX_mY_kY_n + \cdots \right) + \cdots \right] dX\,dY$$

By making the following substitutions, a considerable simplification in the potential function may be made. The coordinates Z_i describes the position of an infinitesimal mass of volume ($dz_1\, dz_2\, dz_3$) in body z.

$$F_{z_i} = \iiint Z_i\, dz_1 dz_2 dz_3 \quad = \text{first moment of body } z$$

$$m_z = \iiint dz_1 dz_2 dz_3 \quad = \text{mass of body } z$$

$$I_{z_i} = \iiint \left[\left(\sum_{j=1}^{3} (Z_j)^2\right) - (Z_i)^2\right] dz_1 dz_2 dz_3 = \text{moment of inertia about axis } i \text{ in body } z$$

$$T_z = \iiint Z_i Z_i dz_1 dz_2 dz_3 = \frac{I_{z_1} + I_{z_2} + I_{z_3}}{2} = \frac{1}{2} \text{ Trace of the inertia tensor}$$

$$I_{z_{ij}} = T_z \delta_{ij} - \iiint Z_i Z_j\, dz_1 dz_2 dz_3 = \begin{bmatrix} I_{z1} & -J_{z12} & -J_{z13} \\ -J_{z21} & I_{z2} & -J_{z23} \\ -J_{z31} & -J_{z32} & J_{z3} \end{bmatrix} = \text{inertia tensor of body } z$$

$$J_{z_{ij}} = + \iiint Z_i Z_j\, dz_1 dz_2 dz_3 \quad i \neq j = \text{the cross product of inertia terms in the above tensor}$$

$$TM_{z_{ijk}} = \iiint Z_i Z_j Z_k\, dz_1 dz_2 dz_3 = \text{third moment tensor of body } z$$

$$TV_{z_i} = \iiint Z_i Z_j Z_j\, dz_1 dz_2 dz_3 = \text{third moment vector of body } z$$

$$FM_{z_{ijk\ell}} = \iiint Z_i Z_j Z_k Z_\ell\, dz_1 dz_2 dz_3 = \text{fourth moment tensor of body } z$$

(Undoubtedly some relationships must exist between the higher moment tensors but they are unknown at the present).

The potential function between the two masses X and Y is therefore,

$$U = -\frac{k}{r}\left[\begin{aligned} & m_x m_y - \frac{1}{r}\left(m_y A_{3i} F_{xi} - m_x B_{3i} F_{yi}\right) \\ & + \frac{1}{r^2}\left(\begin{aligned} & T_x m_y + T_y m_x - \frac{3}{2} A_{3i} A_{3k} I_{xik} m_y - \frac{3}{2} B_{3i} B_{3k} I_{yik} m_k \\ & + A_{ji} B_{jk} F_{xi} F_{xk} - 3 A_{3i} B_{3k} F_{xi} F_{yk} \end{aligned}\right) \\ & + \frac{1}{r^3}\left(\begin{aligned} & \frac{3}{2} TV_{xi} A_{3i} m_y + \frac{3}{2} A_{3i} F_{xi} T_y - 3 A_{3i} A_{j\ell} B_{jk} (T_x I - I_{xi\ell}) F_{yk} \\ & \frac{-3}{2} B_{3i} T_x F_{yi} - \frac{3}{2} B_{3i} TV_{yi} m_x + 3 B_{3i} A_{j\ell} B_{jk} (T_y I - I_{yik}) F_{x\ell} \\ & - \frac{5}{2} A_{3i} A_{3j} A_{3k} TM_{xijk} m_y + \frac{15}{2} A_{3i} A_{3j} B_{3k} (T_x I - I_{xij}) F_{yk} \\ & \frac{-15}{2} A_{3i} B_{3j} B_{3k} (T_y I - I_{yjk}) F_{xi} + \frac{5}{2} B_{3i} B_{3j} B_{3k} TM_{yijk} m_k \end{aligned}\right) \\ & + \cdots \end{aligned}\right]$$

Up to this point, no assumptions as to the shapes of the bodies have been made other than the assumption that the bodies are separated enough so that the integral

$$\int^X \int^Y f_1(x)\, f_2(y)\ dX\, dY$$

may be considered as

$$\left(\int^X f_1(x)\, dX\right)\left(\int^Y f_2(y)\, dY\right).$$

This assumption is valid as long as no portions of the two bodies occupy the same space at the same time.

If now the assumption is made that the reference points for both masses are their centroids, then

$$F_{Xi} = F_{Yi} = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

Using this relationship, the equation for the potential function of two arbitrary bodies is

$$U = -\frac{k}{r}\left[\begin{array}{l} m_x m_y + \frac{1}{r^2}\left(T_x m_y + T_y m_x - \frac{3}{2} A_{3i} A_{3k} I_{x_{ik}} m_y \right. \\ \qquad \left. - \frac{3}{2} B_{3i} B_{3k} I_{y_{ik}} m_x \right) \\ + \frac{1}{r^3}\left(\frac{3}{2} TV_{x_i} A_{3i} m_y - \frac{3}{2} TV_{yi} B_{3i} m_x - \frac{5}{2} A_{3i} A_{3j} A_{3k} TM_{x_{ijk}} m_y \right. \\ \qquad \left. + \frac{5}{2} B_{3i} B_{3j} B_{3k} TM_{y_{ijk}} m_x \right) \\ + \cdots \end{array}\right]$$

It is now possible, to determine Q_m by taking the partials of V with respect to the generalized coordinate q_m.

As a general rule, one of the bodies (say Y in this case) will be the earth. Therefore, in order to remain completely general, the attitude of the earth in inertial space may be represented as

$$B = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ (\cos\theta \sin\psi) & (\cos\theta \cos\psi) & -\sin\theta \\ (\sin\theta \sin\psi) & (\sin\theta \cos\psi) & \cos\theta \end{bmatrix}$$

where

ψ = longitude angle

θ = co-latitude angle measured from North.

Axis 1 is through centroid of earth and Greenwich meridian, and perpendicular to Axis 3.

Axis 2 is through centroid and 90° east, and is perpendicular to Axis 3.

Axis 3 is from centroid to North Pole.

Using this relationship, the potential function for the two-body system may be given as:

$$U = -\frac{Km_Y}{r}\left[\begin{array}{l} m_X + \dfrac{1}{r^2}\left(\begin{array}{l} T_X + m_X\dfrac{T_Y}{m_Y} - \dfrac{3}{2}A_{3i}A_{3k}I_{X_{ik}} \\ -\dfrac{3}{2}B_{3i}B_{3k}\left(\dfrac{I_{Y_{ik}}}{m_Y}\right)m_X \end{array}\right) \\ +\dfrac{1}{r^3}\left(\begin{array}{l} \dfrac{3}{2}TV_{Xi}A_{3i} - \dfrac{3}{2}B_{3i}\dfrac{TV_{Y_i}}{m_Y}m_X \\ -\dfrac{5}{2}A_{3i}A_{3j}A_{3k}TM_{X_{ijk}} + \dfrac{5}{2}B_{3i}B_{3j}B_{3k}\dfrac{TM_{Y_{ijk}}}{m_Y}m_X \end{array}\right) \\ + \cdots \end{array}\right]$$

To evaluate the constant K, we know that for a point mass at the surface of the earth

$$U = -m_X gR_o \doteq -\frac{Km_Y}{R_o}m_X$$

$$\therefore \quad Km_Y = gR_o^2$$

As an example of how this potential energy function is applied for various body configurations, take the special case of a spherical earth. In this case, all of the moments of inertia are the same, or

$$I_{Y_i} = I_{Y_2} = I_{Y_3} = I_Y,$$

then,

$$U = -\frac{gR_o^2}{r}\left[m_X + \frac{1}{r^2}\left(T_X - \frac{3}{2}A_{3i}A_{3k}I_{X_{ik}}\right) + \cdots\right]$$

For the special case of an oblate spheroid earth with a point mass second body,

$$I_{Y_{ik}} = \begin{bmatrix} I_{Y_1} & 0 & 0 \\ 0 & I_{Y_1} & 0 \\ 0 & 0 & I_{Y_3} \end{bmatrix}$$

$$\therefore \quad B_{3i} B_{3k} I_{Y_{ik}} = \sin^2\theta \, I_{Y_1} + \cos^2\theta \, I_{Y_3}$$

$$U = -\frac{gR_o^2}{r}\left[m_X + \frac{m_X}{r^2}\left(\frac{T_Y}{m_Y} - \frac{3}{2}\left(\sin^2\theta \, I_{Y_1} + \cos^2\theta \, I_{Y_3} \right)\right) + \cdots \right]$$

$$= -\frac{gR_o^2}{r} m_X \left[1 + \frac{1}{r^2}\left(\frac{I_{Y_3} - I_{Y_1}}{2m_Y} \right) (1 - 3\cos^2\theta) \quad + \cdots \right]$$

For the case of an oblate earth and an arbitrary body

$$U = -\frac{gR_o^2}{r}\left[m_X + \frac{m_X}{r^2}\left(\frac{I_{Y_3} - I_{Y_1}}{2m_Y} \right)(1 - 3\cos^2\theta) + \frac{1}{r^2}\left(T_X - \frac{3}{2} A_{3i} A_{3k} I_{X_{ik}} \right) + \cdots \right]$$

For more exact answers, more terms of the expansion could be carried, however, for most orbital mechanic problems, it is probably sufficient to carry only the terms of up to $\frac{1}{r^2}$.

It must be recognized that in the above examples the transformation for B given on page 119 is used to represent the position of the Earth, and A_{ij} is a generalized rotation of the position of the body.

It is felt that there are too many possible combinations of bodies to include this form of gravitational attraction in the computer expansion of the equations. The method is included here for completeness.

Equivalence Between the Two Forms

The form of the generalized force as given in the computer runs may be made equivalent to this form by the following method.

$$Q_m = K_g \int_X \int_Y \frac{(R_i - S_i)}{L^3} \frac{\partial(R_i - S_i)}{\partial q_m} \, dX \, dY$$

$$= \frac{K_g}{2} \int_X \int_Y \frac{1}{L^3} \frac{\partial(R_i - S_i)(R_i - S_i)}{\partial q_m} \, dX \, dY \ .$$

However,

$$(R_i - S_i)(R_i - S_i) \equiv L^2,$$

therefore,

$$Q_m = \frac{K_g}{2} \int_X \int_Y \frac{1}{L^3} \frac{\partial L^2}{\partial q_m} \, dX \, dY$$

$$= K_g \int_X \int_Y \frac{1}{L^2} \frac{\partial L}{\partial q_m} \, dX \, dY$$

$$= - K_g \int_X \int_Y \frac{\partial (1/L)}{\partial q_m} \, dX \, dY$$

$$= + \frac{\partial}{\partial q_m} \left[- K_g \int_X \int_Y \frac{1}{L} \, dX \, dY \right]$$

$$Q_m = \frac{\partial U}{\partial q_m} \ ,$$

Where the U's may be seen to be identically equal by comparing the above term to the equation for U on page 120

$$U = - K \int \int \frac{dX \, dY}{D} \ .$$

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* General Dynmics Convair Division P. O. Box 1128, San Diego, Calif 92112	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

A Fortran IV Program to Derive the Equations of Motion of Systems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report

5. AUTHOR(S) *(Last name, first name, initial)*

Westerwick, Robert A.
Brown, J. B.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1966	134	6

8a. CONTRACT OR GRANT NO. AF33(615)-1836 b. PROJECT NO. 8225 c. Task No. 822501 d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-65-194 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this report is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Air Force Flight Dynamics Laboratory Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT

The state of the art of computer development has reached the point where it is now possible to sacrifice some of the extremely high speed of arithmetic computations in order to relieve man of tedium in another area-that of algebraic computation. This report presents the results of a development which uses an IBM 7090/94 computer to derive in non-numeric terms the equations of motion of a system. This system can be composed of any number of masses which are acted upon by forces due to springs, dampers, gravity, and external forces.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Equation of Motion Computer Uses (digital) Non numeric						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

UNCLASSIFIED

Security Classification