33443

ARCHIVES

CSDL-R-1582

EQUATIONS OF MOTION FOR A FLEXIBLE SPACECRAFT -- LUMPED PARAMETER IDEALIZATION

by

Joel Storch
Stephen Gates

September 1982

R-1582

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

FLEXIBLE SPACECRAFT-LUMPED PARAMETER
(Draper (Charles Stark) Lab.)
CSCL 22B Unclas
0033443

Draper Laboratory

PUBLICATION AUTHORIZATION FORM

For External Distribution and Formal Program Documents

TCO Work Order No. 8209H217

Document No. assigned R-1582

DESCRIPTION — to be filled in by author

1. Title of document: Equations of Motion for a Flexible Spacecraft - Lumped Parameter Idealizations

2. Author(s): Joel Storch, Stephen Gates

3. Work generated under Lab account No. 92521

4. Classification: Unclassified [X] Confidential [ ] Secret [ ] S/RD [ ]

5. Date(s) due: Draft copy ______ Final copies ______

6. DOCUMENT TYPE

	Series
[X] Technical Report (interim/final)	R
[ ] Specification	S
[ ] Technical Memorandum	C
[ ] Thesis	T
[ ] Technical Paper, slides, etc.	P

Remarks ______

To be prepared by: Publications (65E) [X]

Other ______

7. DISTRIBUTION

[ ] List attached [ ] List included in document

		copies
Program		
Internal	CSDL	
	Author(s)	2
	TIC/Library	
External:	Gov't Agencies/Contractors	
	Other external	3
	TOTAL	~~2~~ 5

Remarks Security Review
3 Copies to Security

8. PAPERS & PRESENTATIONS—public dissemination

[ ] Abstract, for submission/acceptance or oral presentation. — (resulting paper requires a separate authorization form)

[ ] Manuscript, for external publishing

[ ] Slides, vugraphs, etc. not included in a paper

[ ] Copies to be released at seminar/symposium

[ ] Oral presentation only

To be published by ______

Society ______

Subject of meeting ______

Meeting date(s) ______

Location ______

9. Will any invention be disclosed that has not been submitted to CSDL Patent Committee ? YES [ ] NO [X]

10. Is sponsor review required for technical content ? YES [X] NO [ ]

Reviewing Agency ______

11. Charge this effort to Lab account No. 92521

Signed Joel Storch (Senior Author) date 9/82

AUTHORIZATION TO PREPARE — document described above

Signatures [signature] Author's Supervisor, date 9/27/82 and Joseph F. Turnbull (Program Manager) date 9/27/82 if nonproject funded ______ Chief Scientist date ______

APPROVAL TO PUBLISH — camera-ready copy/manuscript

TCO/Publications: CSDL Styleguide and/or sponsor specs met? YES [X] NO [ ]

[signature] Reviewing Editor, date 9/27/82

Security Office: released as is [ ] review required [X]

J P O'Neill [signature] Security Officer, date 9/27/82

camera-ready copy: Joel Storch (Senior Author) date ______

overall approval: [signature] Department Head — Sponsor interface, date 9-27-82

technical approval: Joseph F. Turnbull (Program Manager or Technical Supervisor) date 9/27/82

IF thesis or IR&D ______ Department Head — Education IR&D date ______

overall approval: [signature] (Author's Department Head) date 9-27-82

IF nonproject funded ______ Chief Scientist date ______

Remarks ______

DOCUMENT CONTROL:

received 5 + M copies, from [illegible] signed [signature] date 9/30/82

Distribution: Program ______ copies, on ______ Internal ______ copies, on ______ External ______ copies, on ______

7906M238

1) DOCUMENT CONTROL FILE COPY

BEAR DOWN: YOU ARE WRITING FIVE COPIES!

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Equations of Motion for a Flexible Spacecraft — Lumped Parameter Idealization		5. Report Date September 1982
		6. Performing Organization Code
7. Author(s) Joel Storch, Stephen Gates		8. Performing Organization Report No. CSDL-R-1582
9. Performing Organization Name and Address The Charles Stark Draper Laboratory, Inc. 555 Technology Square Cambridge, Massachusetts 02139		10. Work Unit No.
		11. Contract or Grant No. NAS-9 16023
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Johnson Space Center Houston, Texas 77058		13. Type of Report and Period Covered Technical Report
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The equations of motion for a flexible vehicle capable of arbitrary translational and rotational motions in inertial space accompanied by small elastic deformations are derived in an unabridged form. The vehicle is idealized as consisting of a single rigid body with an ensemble of mass particles interconnected by massless elastic structure. The internal elastic restoring forces are quantified in terms of a stiffness matrix. A transformation and truncation of elastic degrees of freedom is made in the interest of numerical integration efficiency. Deformation dependent terms are partitioned into a hierarchy of significance. The final set of motion equations are brought to a fully assembled first order form suitable for direct digital implementation. A FORTRAN program implementing the equations is given and its salient features described.

17. Key Words Suggested by Author Spacecraft Dynamics Flexible Body Mechanics Lumped Parameter Modal Coordinates Mechanical Vibrations	18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	74	

The final chapter of this report pertains to a FORTRAN computer program which implements and numerically integrates the complete set of equations. The salient features of the program, its subroutines, and the input and output data are described. An annotated flowchart along with a full listing of the code is provided.

It is noteworthy that while the idealization and methodology applied in this report are essentially those of Likins,[5] the equations formulated herein are unique from those developed in Reference 5, and indeed the distinction is fundamental. It was the express desire to avoid the kinematic restrictions required there to effect a coordinate transformation on the elastic deflections which motivated this approach.

From an applications standpoint, the basic discretization of the vehicle of interest is performed in the manner of lumped mass structural dynamics modeling. The required stiffness matrix which quantifies the internal elastic restoring forces can in general be obtained from preprocessed linear structural finite element analysis programs (e.g., NASTRAN). Because of the mass particle idealization of the elastic domain, only translational displacements are defined at those points, hence any finite element model used to provide stiffness matrix information must be purged of any rotational degrees of freedom that may exist. This requirement is easily satisfied through the application of the static condensation procedure. Thus the analyst is afforded these familiar and versatile structural modeling techniques augmented by the arbitrary motion capability..

The motion equations formulated here are complete and unabridged for a single unconstrained flexible vehicle. However, they could, in a straightforward fashion, be coupled to the dynamics equations for other independent bodies to form an articulated system. This can be done through the identification and elimination of interbody constraint forces/torques and redundant kinematic variables. Indeed, it is for just such an application that these equations are intended. Specifically they are to represent a generic flexible payload to be terminally attached to the

Space Shuttle Orbiter remote manipulator system, which is an articulated chain of rigid and flexible bodies. For this case the model's rigid-body is taken to be the payload grapple fixture with all outboard structure represented by the particle assemblage.

CHAPTER 2

PRELIMINARIES

2.1 Vehicle Idealization

The system being analyzed (see Figure 1) consists of a single rigid body and an attached flexible appendage. The appendage is idealized as a system of particles connected by massless elastic structure. There is no articulation between the appendage and rigid base, i.e., the appendage is "cantilevered" to the rigid body. At an arbitrary point, 0_g, of the rigid body we locate the origin of the body fixed frame which rotates as the body rotates in inertial space. The vector $\vec{R}$ serves to determine the position of 0_g relative to the inertially fixed point 0. The particle masses m_i $(i = 1,2,\ldots,n)$ are located via the position vectors $\vec{r}_i$ relative to 0_g in the undeformed state. The elastic displacement of m_i is $\vec{q}_i$, measured in the body frame.

Many space vehicles or parts of spacecraft can be approximated in this manner. A specific example is the Shuttle Remote Manipulation System in which the "appendage" corresponds to a flexible payload and the "rigid base" to the grapple fixture (this component being attached to the orbiter through the links of the manipulator arm).

2.2 External and Internal Forces

With the ultimate goal in mind of applying the present analysis to more complicated situations, we wish to accommodate all forces and torques which will arise when the system in Figure 1 is attached to other spacecraft components. Hence, at the point 0_g, let there be a force-torque pair: $\vec{f}^{o}(t)$, $\vec{\tau}^{o}(t)$. In the domain of the elastic appendage we have an external force $\vec{f}^{i}(t)$ acting upon the point mass m_i.

As an example, in the case of the Shuttle Remote Manipulator Arm, $\vec{f}^{o}$ and $\vec{\tau}^{o}$ would represent the force and torque exerted by the end-effector on the grapple fixture.

We assume that we are given a stiffness matrix reflecting the mutual elastic forces between the mass points of the appendage. Assemble the elastic displacements as

$$\underline{q} = (q_x^1\ q_y^1\ q_z^1\ q_x^2\ q_y^2\ q_z^2,\ \ldots,\ q_x^n\ q_y^n\ q_z^n\ )^T$$

(x,y,z) refer to Cartesian components along the axes of the body fixed frame.

If [K] is the stiffness matrix, then $-[K]\underline{q}$ is the vector of elastic forces exerted on the point masses. Assume that [K] is partitioned such that the vector of elastic forces are ordered exactly as the elements in $\underline{q}$. Note that in generating [K] the appendage is cantilevered to the rigid base.

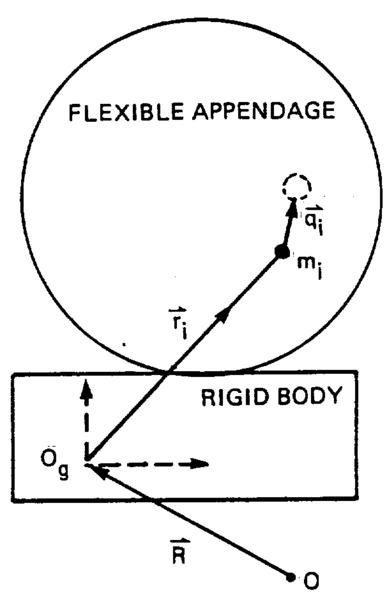


Figure 1. Idealized vehicle.

CHAPTER 3

EQUATIONS OF MOTION

3.1 Particle Translational Equations

$\vec{v}^i$, the inertial velocity of the ith particle, is given by

$$\vec{v}^i = \frac{d}{dt}(\vec{R} + \vec{r}^i + \vec{q}^i)$$

Let $\begin{pmatrix} u \\ v \\ w \end{pmatrix}$ and $\underline{\omega}$ represent the (absolute) velocity and angular velocity of the body frame resolved in body axes, we then have

$$\underline{v}^i = \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \underline{\omega} \times (\underline{r}^i + \underline{q}^i) + \dot{\underline{q}}^i$$

Differentiating this expression we arrive at the particle acceleration

$$\frac{d}{dt}\underline{v}^i = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - (\underline{r}^i + \underline{q}^i) \times \dot{\underline{\omega}} + \ddot{\underline{q}}^i$$

$$+ \underline{\omega} \times \left[\begin{pmatrix} u \\ v \\ w \end{pmatrix} + 2\dot{\underline{q}}^i\right] + \underline{\omega} \times [\underline{\omega} \times (\underline{r}^i + \underline{q}^i)]$$

Expanding the cross product in the last term and using the matrix-vector form for the cross product $\underline{a} \times \underline{b} \equiv [a]^{\sim}\underline{b}$ where $[\underline{a}]^{\sim} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$ the particle acceleration can be written as

$$\frac{d}{dt}\underline{v}^i = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - ([\underline{r}^i]^{\sim} + [\underline{q}^i]^{\sim})\dot{\underline{\omega}} + \ddot{\underline{q}}^i + [\underline{\omega}]^{\sim}\begin{pmatrix} u \\ v \\ w \end{pmatrix} + 2[\underline{\omega}]^{\sim}\dot{\underline{q}}^i$$

$$+ (\underline{\omega}^T\underline{r}^i)\underline{\omega} - ||\underline{\omega}||^2\,\underline{r}^i + (\underline{\omega}^T\underline{q}^i)\underline{\omega} - ||\underline{\omega}||^2\underline{q}^i \qquad (3\text{-}1)$$

If $\underline{f}^i$ is the external force on the i^{th} particle and $\underline{f}_e^i$ the elastic force exerted on the i^{th} particle by the rest of the assemblage (both resolved along body axes), the translational equation is

$$\underline{f}^i + \underline{f}_e^i = m_i \frac{d}{dt}\underline{v}^i$$

$$(i = 1,2,\ldots,n)$$

Partitioning the stiffness matrix into (3×3) arrays

$$[K] = \begin{pmatrix} [K_{11}] & [K_{12}]\ldots[K_{1n}] \\ [K_{21}] & [K_{22}]\ldots[K_{2n}] \\ \vdots & \vdots \qquad \vdots \\ [K_{n1}] & [K_{n2}]\ldots[K_{nn}] \end{pmatrix}$$

$$\underline{f}_e^i = -\sum_{j=1}^{n} [K_{ij}]\,\underline{q}^j$$

Employing Eq. (3-1) for the particle acceleration, the translation equations for the appendage particles may be assembled as

$$\begin{bmatrix} m^1 \\ \hline m^2 \\ \hline \vdots \\ m^n \end{bmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - \begin{bmatrix} m_1(\tilde{\underline{r}}^1 + \tilde{\underline{q}}^1) \\ \hline m_2(\tilde{\underline{r}}^2 + \tilde{\underline{q}}^2) \\ \hline \vdots \\ m_n(\tilde{\underline{r}}^n + \tilde{\underline{q}}^n) \end{bmatrix} \dot{\underline{\omega}} + \begin{bmatrix} m^1 & \underline{0} & \underline{0} & \cdots & \underline{0} \\ \underline{0} & m^2 & \underline{0} & \cdots & \underline{0} \\ \underline{0} & \underline{0} & m^3 & \cdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \underline{0} & \underline{0} & \cdots & \cdots & m^n \end{bmatrix} \ddot{\underline{q}}$$

$$+ [K]\underline{q} = \begin{bmatrix} \underline{f}^1 \\ \underline{f}^2 \\ \vdots \\ \underline{f}^n \end{bmatrix} + \underline{u}_v \qquad (3\text{-}2)$$

where we have introduced the symbol $m^i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} (i = 1,2,\ldots,n)$

and the nonlinear kinematic term $\underline{u}_v$ is given by

$$\underline{u}_v = -\begin{bmatrix} m_1\tilde{\underline{\omega}} \\ \hline m_2\tilde{\underline{\omega}} \\ \hline \vdots \\ \hline m_n\tilde{\underline{\omega}} \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2\begin{bmatrix} m_1\tilde{\underline{\omega}}\,\dot{\underline{q}}^1 \\ \hline m_2\tilde{\underline{\omega}}\,\dot{\underline{q}}^2 \\ \hline \vdots \\ \hline m_n\tilde{\underline{\omega}}\,\dot{\underline{q}}^n \end{bmatrix} - \begin{bmatrix} m_1\underline{\omega}\cdot(\underline{r}^1 + \underline{q}^1)\underline{\omega} \\ \hline m_2\underline{\omega}\cdot(\underline{r}^2 + \underline{q}^2)\underline{\omega} \\ \hline \vdots \\ \hline m_n\underline{\omega}\cdot(\underline{r}^n + \underline{q}^n)\underline{\omega} \end{bmatrix}$$

$$+ ||\underline{\omega}||^2 \begin{bmatrix} m_1(\underline{r}^1 + \underline{q}^1) \\ \hline m_2(\underline{r}^2 + \underline{q}^2) \\ \hline \vdots \\ \hline m_n(\underline{r}^n + \underline{q}^n) \end{bmatrix} \qquad (3\text{-}3)$$

Equation (3-2) constitutes a set of 3n scalar differential equations.

3.2 Vehicle Translational Equations

For the composite system (rigid body and appendage) the sum of the external forces equals the total mass times the acceleration of the mass center.

If m_b is the mass of the rigid body and $\underline{s}$ is the vector from O_g to the mass center of the rigid body (expressed in the body frame)

$$m\underline{c} = \sum_{i=1}^{n} m_i(\underline{r}^i + \underline{q}^i) + m_b\underline{s} \tag{3-4}$$

$m = m_b + \sum_{i=1}^{n} m_i$ is the total mass.

$\underline{c}(t)$ is the vector position of the instantaneous mass center relative to O_g.

The acceleration of the mass center is: $\frac{d^2}{dt^2}(\vec{R} + \vec{c})$

$$\frac{d^2\vec{R}}{dt^2} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + \underline{\omega} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} \qquad \text{(in body frame)}$$

$$m\frac{d^2\underline{c}}{dt^2} = \sum_{i=1}^{n} m_i\ddot{\underline{q}}^i - m\underline{c} \times \dot{\underline{\omega}} + 2\underline{\omega} \times \sum_{i=1}^{n} m_i\dot{\underline{q}}^i + \underline{\omega} \times (\underline{\omega} \times m\underline{c})$$

Expressing this last term as: $(\underline{\omega} \cdot m\underline{c})\underline{\omega} - ||\underline{\omega}||^2 m\underline{c}$ the vehicle translational equation assumes the form

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - [m\underline{c}]^{\sim}\dot{\underline{\omega}} + [m^1 \quad m^2 \quad \cdots \quad m^n]\ddot{\underline{q}} = \sum_{i=0}^{n} \underline{f}^i + \underline{u}_t \tag{3-5}$$

The nonlinear term $\underline{u}_t$ is given by

$$\underline{u}_t = -m\underline{\omega} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2\underline{\omega} \times \sum_{i=1}^{n} m_i \dot{\underline{q}}^i - (\underline{\omega} \cdot m\underline{c})\underline{\omega} + ||\underline{\omega}||^2 m\underline{c} \tag{3-6}$$

3.3 Vehicle Rotational Equations

For the composite system (rigid body and appendage) the sum of the external torques taken about the mass center equals the time rate of change of the angular momentum taken about the mass center.

Let $[I_b]$ be the inertia matrix of the rigid body with respect to a coordinate system located at the mass center of the rigid body and parallel to the body fixed axes system at O_g.

$$[I_b] = \iiint [\underline{\lambda} \cdot \underline{\lambda}E - \underline{\lambda}\underline{\lambda}^T] \, dm$$

$\underline{\lambda}$ is the position vector of a mass element dm in the rigid body relative to the rigid body mass center and the integration is performed over the region occupied by the rigid base. [E] denotes the unit matrix.

The system angular momentum can be split into two parts:

$\underline{H}_b$ — angular momentum of rigid base

$\sum_{i=1}^{n} \underline{H}_i$ — angular momentum of appendage particles

Let $\underline{\ell}^i$ and $\underline{\ell}$ denote the position vectors from the system mass center to m_i and a generic mass element in the rigid body respectively.

$$\underline{H}_b = \iiint \underline{\ell} \times \frac{d}{dt} \underline{\ell} \, dm$$

$$\underline{H}_i = \underline{\ell}^i \times m_i \frac{d}{dt} \underline{\ell}^i$$

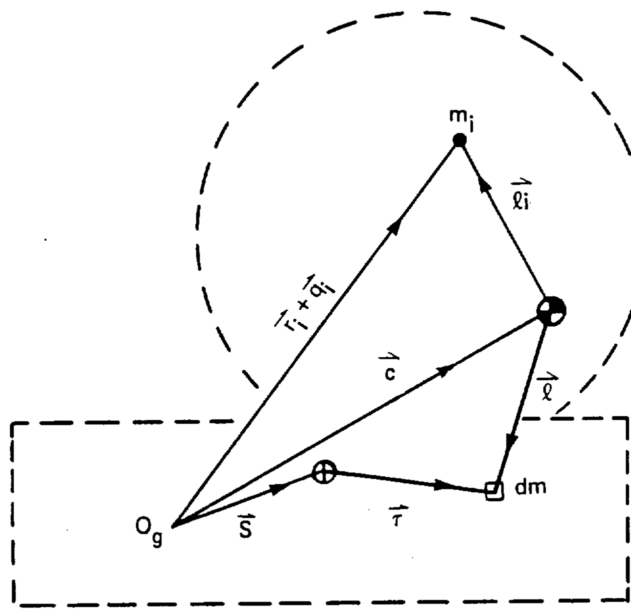


Figure 2. Vector geometry.

From Figure 2, $\underline{\ell}^i = \underline{r}^i + \underline{q}^i - \underline{c}$, $\underline{\ell} = \underline{s} + \underline{\lambda} - \underline{c}$.

Inserting this expression for $\underline{\ell}$ into the integral definition of $\underline{H}_b$ and recalling the definition of $[I_b]$ and the fact that $\iiint \underline{\lambda}\ dm = \underline{0}$ we arrive at the following expression for $\underline{H}_b$

$$\underline{H}_b = [I_b]\underline{\omega} + m_b(\underline{s} - \underline{c}) \times \frac{d}{dt}(\underline{s} - \underline{c})$$

Thus

$$\frac{d}{dt}\underline{H}_b = [I_b]\dot{\underline{\omega}} + \underline{\omega} \times [I_b]\underline{\omega} + m_b(\underline{s} - \underline{c}) \times [\dot{\underline{\omega}} \times \underline{s} + \underline{\omega} \times (\underline{\omega} \times \underline{s})]$$
$$- m_b(\underline{s} - \underline{c}) \times \frac{d^2}{dt^2}\underline{c}$$

Turning to the angular momentum of the i^{th} particle

$$\frac{d}{dt}\underline{H}^i = \underline{\ell}^i \times m_i \frac{d^2}{dt^2}\underline{\ell}^i$$

Now

$$\frac{d^2}{dt^2}\underline{\ell}^i = -(\underline{r}^i + \underline{q}^i) \times \dot{\underline{\omega}} + \ddot{\underline{q}}^i + 2\underline{\omega} \times \dot{\underline{q}}^i + [\underline{\omega} \cdot (\underline{r}^i + \underline{q}^i)]\underline{\omega}$$

$$- ||\underline{\omega}||^2(\underline{r}^i + \underline{q}^i) - \frac{d^2}{dt^2}\underline{c}$$

Combining the above expressions for $\frac{d}{dt}\underline{H}_b$ and $\frac{d}{dt}\underline{H}^i$ (with the substitution for $\frac{d^2}{dt^2}\underline{\ell}^i$) the terms involving $\frac{d^2}{dt^2}\underline{c}$ conveniently cancel leaving the following result for the time derivative of the angular momentum

$$\frac{d}{dt}\underline{H} = ([I_b] - m_b(\underline{s} - \underline{c})^{\sim}\tilde{\underline{s}})\dot{\underline{\omega}} + \underline{\omega} \times [I_b]\underline{\omega} + \sum_{i=1}^{n} m_i[\underline{\ell}^i]^{\sim}\ddot{\underline{q}}^i$$

$$+ m_b(\underline{s} - \underline{c}) \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})] - \sum_{i=1}^{n} m_i[\underline{\ell}^i]^{\sim}(\underline{r}^i + \underline{q}^i)^{\sim}\dot{\underline{\omega}}$$

$$+ 2\sum_{i=1}^{n} m_i[\underline{\ell}^i]^{\sim}\tilde{\underline{\omega}}\,\dot{\underline{q}}^i + \sum_{i=1}^{n} [\underline{\omega} \cdot (\underline{r}^i + \underline{q}^i)]m_i\,\underline{\ell}^i \times \underline{\omega}$$

$$- ||\underline{\omega}||^2 \sum_{i=1}^{n} m_i\,\underline{\ell}^i \times (\underline{r}^i + \underline{q}^i) \tag{3-7}$$

The system rotational motion equation is

$$\frac{d}{dt}\underline{H} = \underline{\tau}^o - \underline{c} \times \underline{f}^o + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i - \underline{c}) \times \underline{f}^i$$

$$= \underline{\tau}^o - \underline{c} \times \sum_{j=0}^{n} \underline{f}^j + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i) \times \underline{f}^i$$

If we were to insert Eq. (3-7) into this last equation we would have a valid equation for system rotation. Note, however, that this equation would not depend upon $(\dot{u},\dot{v},\dot{w})$ explicitly. Since the translational equations (3-5) depend upon $\dot{\underline{\omega}}$ explicitly, and we wish to have a final set of equations with a symmetric coefficient matrix of the generalized accelerations, we force the coupling between the rotational equations and the acceleration vector $(\dot{u}\ \dot{v}\ \dot{w})$ by the following device.

Take Eq. (3-5) and (3-6) and solve for $\sum_{j=0}^{n} \underline{f}^j$. The system rotational motion equation then becomes

$$\frac{d}{dt}\underline{H} = \underline{\tau}^o + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i) \times \underline{f}^i - [m\underline{c}]^\sim \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + [\underline{c}]^\sim [m\underline{c}]^\sim \dot{\underline{\omega}}$$

$$- \underline{c} \times \sum_{i=1}^{n} m_i \ddot{\underline{q}}^i - [m\underline{c}]^\sim \tilde{\underline{\omega}} \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2[\underline{c}]^\sim \tilde{\underline{\omega}} \sum_{i=1}^{n} m_i \dot{\underline{q}}^i$$

$$- (\underline{\omega} \cdot m\underline{c})\, \underline{c} \times \underline{\omega} \qquad (3\text{-}8)$$

Combining Eq. (3-7) and (3-8), we arrive at the final desired form for the vehicle rotational equation

$$[m\underline{c}]^{\sim} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + [I(t)]\ \dot{\underline{\omega}} + \left[m_1(\underline{r}^1 + \underline{q}^1)^{\sim} \mid m_2(\underline{r}^2 + \underline{q}^2)^{\sim} \mid \ldots \mid m_n(\underline{r}^n + \underline{q}^n)^{\sim}\right]\ddot{\underline{q}}$$

$$= \underline{\tau}^{o} + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i) \times \underline{f}^i + \underline{u}_r \qquad (3\text{-}9)$$

Here

$$[I(t)] = [I_b] - m_b(\underline{s} - \underline{c})^{\sim}\underline{\tilde{s}} - \sum_{i=1}^{n} m_i [\underline{\ell}^i]^{\sim}(\underline{r}^i + \underline{q}^i)^{\sim} - [\underline{\tilde{c}}][m\underline{c}]^{\sim}$$

which can be simplified to

$$[I(t)] = [I_b] - m_b[\underline{\tilde{s}}]^2 - \sum_{i=1}^{n} m_i(\underline{\tilde{r}}^i + \underline{\tilde{q}}^i)^2 \qquad (3\text{-}10)$$

In this form we recognize [I(t)] as the inertia matrix of the vehicle about point O_g.

The nonlinear rotation term $\underline{u}_r$ is given by

$$\underline{u}_r = -[m\underline{c}]^{\sim}\underline{\tilde{\omega}} \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2[\underline{c}]^{\sim}[\underline{\omega}]^{\sim} \sum_{i=1}^{n} m_i \dot{\underline{q}}^i - (\underline{\omega} \cdot m\underline{c})[\underline{c}]^{\sim}\underline{\omega}$$

$$- [\underline{\omega}]^{\sim}[I_b]\underline{\omega} - m_b(\underline{s} - \underline{c}) \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})] - 2 \sum_{i=1}^{n} m_i [\underline{\ell}^i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i$$

$$- \sum_{i=1}^{n} [\underline{\omega} \cdot (\underline{r}^i + \underline{q}^i)] m_i\ \underline{\ell}^i \times \underline{\omega} + ||\underline{\omega}||^2 \sum_{i=1}^{n} m_i\ \underline{\ell}^i \times (\underline{r}^i + \underline{q}^i)$$

(3-11)

CHAPTER 4

EXPANSION AND PARTITIONING OF TERMS

4.1 Deformation Dependent Coefficients

In the equations developed thus far, specifically Eq. (3-2), (3-5), and (3-9), we have isolated the accelerations on the left hand sides of the respective equations. The acceleration coefficients are time dependent through the elastic deformations. It is quite desirable from an applications viewpoint to rank the constituents in these coefficients in accordance with their relative magnitude. Thus, in a computer simulation, one can choose to omit certain terms and speed up execution with a minimal impact on computed results.

We will rank terms amongst three categories:

(1) Terms independent of $\underline{q}$.

(2) Terms first order in $\underline{q}$.

(3) Terms second order in $\underline{q}$.

The majority of coefficients are directly identifiable in this hierarchy. We have two coefficients which require additional attention: $m\underline{c}$ and $[I(t)]$.

From Eq. (3-4), $m\underline{c} = m_b\, \underline{s} + \sum_{i=1}^{n} m_i \underline{r}^i + \sum_{i=1}^{n} m_i \underline{q}^i(t)$. The first two terms are of category 1 and the sum will be denoted by $m\underline{c}_0$. The time-dependent term will be denoted by $m\underline{c}_1(t)$.

The matrix $[I(t)]$ is given by Eq. (3-10) and can be written as

$$[I(t)] = [I_1] + [I_2(t)] + [I_3(t)]$$

The three matrices $[I_1]$, $[I_2(t)]$, and $[I_3(t)]$ are of category 1, 2, and 3 respectively and are given by

$$[I_1] = [I_b] - m_b[\tilde{\underline{s}}]^2 - \sum_{i=1}^{n} m_i[\tilde{\underline{r}}^i]^2$$

$$[I_2(t)] = -\sum_{i=1}^{n} m_i([\tilde{\underline{r}}^i][\tilde{\underline{q}}^i] + [\tilde{\underline{q}}^i][\tilde{\underline{r}}^i])$$

$$[I_3(t)] = -\sum_{i=1}^{n} m_i[\tilde{\underline{q}}^i]^2$$

4.2 Nonlinear Kinematic Terms

In this section, we concentrate upon the three nonlinear terms: $\underline{u}_t$, $\underline{u}_v$, and $\underline{u}_r$ appearing on the right hand sides of the motion equations. Following a procedure similar to that of the previous section, the nonlinear terms are partitioned amongst three categories:

(1) Nonlinear terms independent of $\underline{q}$, $\dot{\underline{q}}$

(2) Nonlinear terms first order in $\underline{q}$, $\dot{\underline{q}}$

(3) Nonlinear terms second order in $\underline{q}$, $\dot{\underline{q}}$

Accordingly, from Eq. (3-6), $\underline{u}_t = \underline{u}_t^{(1)} + \underline{u}_t^{(2)}$ with

$$\underline{u}_t^{(1)} = -m[\underline{\omega}]^{\sim} \begin{pmatrix} u \\ v \\ w \end{pmatrix} - (\underline{\omega} \cdot m\underline{c}_0)\underline{\omega} + ||\underline{\omega}||^2 m\underline{c}_0 \qquad (4\text{-}1)$$

$$\underline{u}_t^{(2)} = -2[\underline{\omega}]^{\sim} \sum_{i=1}^{n} m_i \dot{\underline{q}}^i - (\underline{\omega} \cdot m\underline{c}_1)\underline{\omega} + ||\underline{\omega}||^2 m\underline{c}_1 \qquad (4\text{-}2)$$

In a similar manner from Eq. (3-3)

$$\underline{u}_v = \underline{u}_v^{(1)} + \underline{u}_v^{(2)}$$

$$\underline{u}_v^{(1)} = -\begin{bmatrix} m_1\tilde{\underline{\omega}} \\ m_2\tilde{\underline{\omega}} \\ \vdots \\ m_n\tilde{\underline{\omega}} \end{bmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix} - \begin{bmatrix} m_1(\underline{\omega} \cdot \underline{r}^1)\underline{\omega} \\ m_2(\underline{\omega} \cdot \underline{r}^2)\underline{\omega} \\ \vdots \\ m_n(\underline{\omega} \cdot \underline{r}^n)\underline{\omega} \end{bmatrix} + ||\underline{\omega}||^2 \begin{bmatrix} m_1\underline{r}^1 \\ m_2\underline{r}^2 \\ \vdots \\ m_n\underline{r}^n \end{bmatrix} \quad (4\text{-}3)$$

$$\underline{u}_v^{(2)} = -2\begin{bmatrix} m_1\tilde{\underline{\omega}}\,\dot{\underline{q}}^1 \\ m_2\tilde{\underline{\omega}}\,\dot{\underline{q}}^2 \\ \vdots \\ m_n\tilde{\underline{\omega}}\,\dot{\underline{q}}^n \end{bmatrix} - \begin{bmatrix} m_1(\underline{\omega} \cdot \underline{q}^1)\underline{\omega} \\ m_2(\underline{\omega} \cdot \underline{q}^2)\underline{\omega} \\ \vdots \\ m_n(\underline{\omega} \cdot \underline{q}^n)\underline{\omega} \end{bmatrix} + ||\underline{\omega}||^2 \begin{bmatrix} m_1\underline{q}^1 \\ m_2\underline{q}^2 \\ \vdots \\ m_n\underline{q}^n \end{bmatrix} \quad (4\text{-}4)$$

<u>Expansion of $\underline{u}_r$</u>

The two terms in $\underline{u}_r$ (Eq. (3-11)) depending upon $\dot{\underline{q}}$ can be combined as

$$-2\sum_{i=1}^{n} m_i([\underline{c}]^{\sim} + [\underline{\ell}^i]^{\sim})[\underline{\omega}]^{\sim}\dot{\underline{q}}^i = -2\sum_{i=1}^{n} m_i(\underline{r}^i + \underline{q}^i)^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i$$

$$= -2\sum_{i=1}^{n} m_i[\underline{r}^i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i - 2\sum_{i=1}^{n} m_i[\underline{q}^i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i$$

The third term in $\underline{u}_r$ can be expressed as

$$(\underline{\omega} \cdot m\underline{c})[\underline{c}]^{\sim}\underline{\omega} = (\underline{\omega} \cdot m\underline{c}_0)[\underline{c}_0]^{\sim}\underline{\omega} + (\underline{\omega} \cdot m\underline{c}_0)[\underline{c}_1]^{\sim}\underline{\omega} + (\underline{\omega} \cdot m\underline{c}_1)[\underline{c}_0]^{\sim}\underline{\omega}$$

$$+ (\underline{\omega} \cdot m\underline{c}_1)[\underline{c}_1]^{\sim}\underline{\omega}$$

For the last two terms in $\underline{u}_r$ the following expansions are useful

$$[\underline{\omega} \cdot (\underline{r}^i + \underline{q}^i)]m_i\underline{\ell}^i \times \underline{\omega} = (\underline{\omega} \cdot \underline{r}^i)m_i(\underline{r}^i - \underline{c}_0) \times \underline{\omega}$$

$$+ (\underline{\omega} \cdot \underline{q}^i)m_i(\underline{r}^i - \underline{c}_0) \times \underline{\omega}$$

$$+ (\underline{\omega} \cdot \underline{r}^i)m_i(\underline{q}^i - \underline{c}_1) \times \underline{\omega}$$

$$+ (\underline{\omega} \cdot \underline{q}^i)m_i(\underline{q}^i - \underline{c}_1) \times \underline{\omega}$$

$$\underline{\ell}^i \times (\underline{r}^i + \underline{q}^i) = \underline{r}^i \times \underline{c}_0 + \underline{r}^i \times \underline{c}_1 + \underline{q}^i \times \underline{c}_0 + \underline{q}^i \times \underline{c}_1$$

Collecting terms in $\underline{u}_r$ independent of deformation

$$\underline{u}_r^{(1)} = -[m\underline{c}_0]^{\sim}[\underline{\omega}]^{\sim}\begin{pmatrix} u \\ v \\ w \end{pmatrix} - (\underline{\omega} \cdot m\underline{c}_0)[\underline{c}_0]^{\sim}\underline{\omega} - \underline{\omega} \times [I_b]\underline{\omega}$$

$$- m_b(\underline{s} - \underline{c}_0) \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})] - \sum_{i=1}^{n} m_i(\underline{\omega} \cdot \underline{r}^i)(\underline{r}^i - \underline{c}_0) \times \underline{\omega}$$

$$+ ||\underline{\omega}||^2 \sum_{i=1}^{n} m_i\underline{r}^i \times \underline{c}_0$$

The expression for $\underline{u}_r^{(1)}$ can be further simplified by use of the following identities

$$\sum_{i=1}^{n} m_i \underline{r}^i \times \underline{c}_0 = -m_b \underline{s} \times \underline{c}_0$$

$$\sum_{i=1}^{n} m_i (\underline{\omega} \cdot \underline{r}^i)(\underline{r}^i - \underline{c}_0) \times \underline{\omega} = -[\underline{\omega}]^{\sim} \sum_{i=1}^{n} m_i \underline{r}^i \underline{r}^{iT} \underline{\omega}$$

$$+ [\underline{\omega} \cdot (m_b \underline{s} - m\underline{c}_0)]\underline{c}_0 \times \underline{\omega}$$

$$-m_b(\underline{s} - \underline{c}_0) \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})] = -(\underline{\omega} \cdot \underline{s}) m_b \underline{s} \times \underline{\omega} + (\underline{\omega} \cdot \underline{s}) m_b \underline{c}_0 \times \underline{\omega}$$

$$- ||\underline{\omega}||^2 m_b \underline{c}_0 \times \underline{s}$$

Incorporating these results we arrive at the final expression

$$\underline{u}_r^{(1)} = -[m\underline{c}_0]^{\sim}[\underline{\omega}]^{\sim} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + [\underline{\omega}]^{\sim} \sum_{i=1}^{n} m_i \underline{r}^i \underline{r}^{iT} \underline{\omega} - \underline{\omega} \times [I_b]\underline{\omega}$$

$$- m_b(\underline{s} \cdot \underline{\omega})\underline{s} \times \underline{\omega} \qquad (4\text{-}5)$$

Collecting first order deformation dependent terms in $\underline{u}_r$

$$\underline{u}_r^{(2)} = -[m\underline{c}_1]^{\sim}[\underline{\omega}]^{\sim} \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2\sum_{i=1}^{n} m_i [\underline{r}^i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i - (\underline{\omega} \cdot m\underline{c}_0)[\underline{c}_1]^{\sim}\underline{\omega}$$

$$- (\underline{\omega} \cdot m\underline{c}_1)[\underline{c}_0]^{\sim}\underline{\omega} + m_b\underline{c}_1 \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})]$$

$$- \sum_{i=1}^{n} [(\underline{\omega} \cdot \underline{q}^i) m_i (\underline{r}^i - \underline{c}_0) \times \underline{\omega} + (\underline{\omega} \cdot \underline{r}^i) m_i (\underline{q}^i - \underline{c}_1) \times \underline{\omega}]$$

$$+ ||\underline{\omega}||^2 \sum_{i=1}^{n} m_i (\underline{r}^i \times \underline{c}_1 + \underline{q}^i \times \underline{c}_0)$$

The expression for $\underline{u}_r^{(2)}$ can be further simplified by use of the following identities

$$\sum_{i=1}^{n} m_i(\underline{r}^i \times \underline{c}_1 + \underline{q}^i \times \underline{c}_0) = m_b\, \underline{c}_1 \times \underline{s}$$

$$-\sum_{i=1}^{n} [(\underline{\omega} \cdot \underline{q}^i)m_i(\underline{r}^i - \underline{c}_0) \times \underline{\omega} + (\underline{\omega} \cdot \underline{r}^i)m_i(\underline{q}^i - \underline{c}_1) \times \underline{\omega}] =$$

$$[\underline{\omega}]^{\sim}\sum_{i=1}^{n} m_i(\underline{r}^i\underline{q}^{iT} + \underline{q}^i\underline{r}^{iT})\underline{\omega} + (\underline{\omega} \cdot m\underline{c}_1)\underline{c}_0 \times \underline{\omega} + [\underline{\omega} \cdot (m\underline{c}_0 - m_b\underline{s})]\underline{c}_1 \times \underline{\omega}$$

$$m_b\, \underline{c}_1 \times [\underline{\omega} \times (\underline{\omega} \times \underline{s})] = m_b(\underline{\omega} \cdot \underline{s})\underline{c}_1 \times \underline{\omega} - m_b||\underline{\omega}||^2\, \underline{c}_1 \times \underline{s}$$

$$\underline{u}_r^{(2)} = -[m\underline{c}_1]^{\sim}[\underline{\omega}]^{\sim}\begin{pmatrix}u\\v\\w\end{pmatrix} - 2\sum_{i=1}^{n} m_i[\underline{r}^i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i$$

$$+ [\underline{\omega}]^{\sim}\sum_{i=1}^{n} m_i(\underline{r}^i\underline{q}^{iT} + \underline{q}^i\underline{r}^{iT})\underline{\omega} \qquad (4\text{-}6)$$

Collecting second order deformation dependent terms in $\underline{u}_r$ and simplifying

$$\underline{u}_r^{(3)} = -2\sum_{i=1}^{n} m_i[\underline{q}_i]^{\sim}[\underline{\omega}]^{\sim}\dot{\underline{q}}^i + [\underline{\omega}]^{\sim}\sum_{i=1}^{n} m_i\underline{q}^i\underline{q}^{iT}\underline{\omega} \qquad (4\text{-}7)$$

$\underline{u}_r = \underline{u}_r^{(1)} + \underline{u}_r^{(2)} + \underline{u}_r^{(3)}$ where the terms on the right hand side are given by Eq. (4-5) - (4-7).

CHAPTER 5

MODAL COORDINATE TRANSFORMATION

When the number of particles in the appendage idealization becomes large, high frequencies obtain which make numerical integration difficult. We will describe a truncated coordinate transformation to circumvent this difficulty. Note that the treatment to follow is somewhat heuristic and hence requires good engineering judgement and caution in its implementation.

Since the high frequencies arise from the appendage vibration, it is natural to start with its governing equation (3-2, 3-3). Consider the case where no external forces act, $\underline{\omega} = \underline{0}$ and $(\dot{u},\dot{v},\dot{w}) = \underline{0}$. The "constrained" appendage equation then assumes the familiar form

$$[M]\ddot{\underline{q}} + [K]\underline{q} = \underline{0} \qquad \text{where } [M] = \text{diag}(m^1, m^2, \ldots, m^n)$$

The natural frequencies, ω_i, and corresponding mode shapes, $\underline{v}^i$, are determined from

$$([K] - \omega^2[M])\underline{V} = \underline{0} \qquad (5\text{-}1)$$

For the vehicle we are treating here, the appendage is rigidly attached to the base body hence no rigid body modes are present in the above eigenvalue problem. Equivalently, [K] is positive definite. Since [K] and [M] are symmetric and positive definite there exists 3n independent eigenvectors $\underline{v}^i$ corresponding to positive eigenvalues ω_i^2, even if there are multiple eigenvalues.

We assume that the eigenvectors are normalized such that $(\underline{v}^i,[M]\underline{v}^i) = 1$. It follows that $(\underline{v}^i,[K]\underline{v}^i) = \omega_i^2$. We can always create a mutually orthogonal set such that

$$(\underline{v}^i,[M]\underline{v}^j) = 0 = (\underline{v}^i,[K]\underline{v}^j) \qquad (i \neq j)$$

In actual computation we can deal with a simpler eigenvalue problem than that presented by Eq. (5-1). Specifically we will transform Eq. (5-1) to an ordinary symmetric eigenvalue problem. Introduce the change of variables: $\underline{W} = [M]^{1/2}\underline{V}$. Since [M] is diagonal, $[M]^{1/2}$ is a diagonal matrix whose elements are the square roots of the corresponding elements in [M]. The eigenvalue problem transforms into

$$[K][M]^{-1/2}\underline{W} = \omega^2[M]^{1/2}\underline{W} \quad \text{or} \quad [\mathcal{G}]\underline{W} = \omega^2\underline{W} \qquad (5\text{-}2)$$

where $[\mathcal{G}]$ is the symmetric matrix: $[\mathcal{G}] = [M]^{-1/2}[K][M]^{-1/2}$

It is easily verified that if the eigenvectors $\underline{W}^i$ $(i = 1,2,\ldots,3n)$ of Eq. (5-2) are orthogonal (which can always be done) then the corresponding eigenvectors of (5-1) satisfy all orthogonality and normality conditions specified above.

Order the eigenvalues such that $\omega_1^2 \leq \omega_2^2 \leq \ldots \leq \omega_{3n}^2$ and let $[\Phi]$ be the $(3n \times t)$ matrix whose columns are the eigenvectors $\underline{v}^1,\underline{v}^2,\ldots,\underline{v}^t$ $(t \leq 3n)$. We now make the transformation

$$\underline{q} = [\Phi]\underline{\eta} \qquad (5\text{-}3)$$

This is not a coordinate transformation in the strict sense, since $[\Phi]$ does not have an inverse when $t < 3n$. The appendage deformation is now characterized by t "modal coordinates" instead of the original 3n deformation coordinates. We formally make the substitution, Eq. (5-3), into the full set of motion equations.

Substituting into the appendage deformation equation (3-2), premultiplying by $[\Phi]^T$ and recalling the orthogonality and normality conditions we arrive at

$$[\Phi]^T \begin{bmatrix} m^1 \\ m^2 \\ \vdots \\ m^n \end{bmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - [\Phi]^T \begin{bmatrix} m_1(\underline{r}^1 + \underline{q}^1)^{\sim} \\ m_2(\underline{r}^2 + \underline{q}^2)^{\sim} \\ \vdots \\ m_n(\underline{r}^n + \underline{q}^n)^{\sim} \end{bmatrix} \dot{\underline{\omega}} + \ddot{\underline{\eta}} + \begin{bmatrix} \omega_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \omega_2^2 & & & 0 \\ 0 & & \ddots & & \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdot & \omega_t^2 \end{bmatrix} \underline{\eta} =$$

$$[\Phi]^T \begin{bmatrix} \underline{f}^1 \\ \underline{f}^2 \\ \vdots \\ \underline{f}^n \end{bmatrix} + [\Phi]^T \underline{u}_v \qquad (5\text{-}4)$$

The vehicle translational equations (3-5) become

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} - [m\underline{c}]^{\sim}\dot{\underline{\omega}} + [m^1\ m^2\ \ldots\ m^n][\Phi]\ddot{\underline{\eta}} = \sum_{i=0}^{n} \underline{f}^i + \underline{u}_t$$

The vehicle rotational equations (3-9) become

$$[m\underline{c}]^{\sim} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + [I(t)]\dot{\underline{\omega}} + \left[m_1(\underline{r}^1 + \underline{q}^1)^{\sim} \,\middle|\, m_2(\underline{r}^2 + \underline{q}^2)^{\sim} \,\middle|\, \ldots \,\middle|\, m_n(\underline{r}^n + \underline{q}^n)^{\sim} \right] [\Phi]\ddot{\underline{\eta}} =$$

$$\underline{\tau}^0 + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i) \times \underline{f}^i + \underline{u}_r$$

The assembled equations of motion in matrix form are presented in Figure 3.

$$
\begin{bmatrix}
\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} & -[m\underline{c}]^{\sim} & [m^1 m^2 \ldots m^n]\,[\Phi] \\
[m\underline{c}]^{\sim} & [I(t)] & [m_1(\underline{r}^1+\underline{g}^1)^{\sim} \cdots m_n(\underline{r}^n+\underline{g}^n)^{\sim}]\,[\Phi] \\
[\Phi]^T \begin{bmatrix} m^1 \\ m^2 \\ \vdots \\ m^n \end{bmatrix} & -[\Phi]^T \begin{bmatrix} m_1(\underline{r}^1+\underline{g}^1)^{\sim} \\ m_2(\underline{r}^2+\underline{g}^2)^{\sim} \\ \vdots \\ m_n(\underline{r}^n+\underline{g}^n)^{\sim} \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & & 1 \end{bmatrix}
\end{bmatrix}
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\underline{\omega}} \\ \ddot{\underline{\eta}} \end{bmatrix}
$$

$$
= \begin{bmatrix}
\sum_{i=0}^{n} \underline{f}^i \\
\underline{\tau}^0 + \sum_{i=1}^{n} (\underline{r}^i + \underline{g}^i) \times \underline{f}^i \\
[\Phi]^T \begin{pmatrix} \underline{f}^1 \\ \underline{f}^2 \\ \vdots \\ \underline{f}^n \end{pmatrix}
\end{bmatrix}
-
\begin{bmatrix}
\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\
\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\
\begin{bmatrix} \omega_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \omega_2^2 & \cdot & & \\ 0 & & & \cdot & \\ \vdots & & & & \cdot \\ 0 & 0 & \cdots & \cdots & \omega_t^2 \end{bmatrix} \underline{\eta}
\end{bmatrix}
+
\begin{bmatrix}
\underline{u}_t \\
\underline{u}_r \\
[\Phi]^T\, \underline{u}_v
\end{bmatrix}
$$

Figure 3. Assembled equations of motion.

CHAPTER 6

SYSTEM KINETIC ENERGY

The kinetic energy of the vehicle is the sum of the translational and rotational kinetic energy of the rigid body and the kinetic energy of the particles comprising the appendage

$$T = \frac{1}{2} m_b v_b^2 + \frac{1}{2} \underline{\omega} \cdot [I_b]\underline{\omega} + \frac{1}{2} \sum_{i=1}^{n} m_i v_i^2$$

$\underline{v}_b = \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \underline{\omega} \times \underline{s}$ is the velocity of the mass center of the base

$\underline{v}^i = \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \underline{\omega} \times (\underline{r}^i + \underline{q}^i) + \dot{\underline{q}}^i$ is the velocity of i^{th} particle

Forming the inner products $(\underline{v}_b, \underline{v}_b);(\underline{v}^i, \underline{v}^i)$ and recalling Eq. (3-10) for $[I(t)]$ the kinetic energy can be written as

$$T = \frac{1}{2} m(u^2 + v^2 + w^2) + \frac{1}{2} \underline{\omega}^T[I(t)]\underline{\omega} + \frac{1}{2} \sum_{i=1}^{n} \{\dot{\underline{q}}^i\}^T m_i \dot{\underline{q}}^i$$

$$- \frac{1}{2} (uvw)[m\underline{c}]^{\sim}\underline{\omega} + \frac{1}{2} \underline{\omega}^T[m\underline{c}]^{\sim} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{2} (uvw) \sum_{i=1}^{n} m_i\dot{\underline{q}}^i$$

$$+ \frac{1}{2} \sum_{i=1}^{n} m_i\{\dot{\underline{q}}^i\}^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{2} \underline{\omega}^T \sum_{i=1}^{n} m_i(\underline{r}^i + \underline{q}^i)^{\sim}\dot{\underline{q}}^i - \frac{1}{2} \sum_{i=1}^{n} m_i\{\dot{\underline{q}}^i\}^T(\underline{r}^i + \underline{q}^i)^{\sim}\underline{\omega}$$

We now rewrite those terms in T which depend upon $\dot{\underline{q}}^i$ in terms of $\dot{\underline{\eta}}$

$$\sum_{i=1}^{n} \{\dot{\underline{q}}^i\}^T m_i \dot{\underline{q}}^i = \dot{\underline{q}}^T [M] \dot{\underline{q}} = \dot{\underline{\eta}}^T [\Phi]^T [M][\Phi] \dot{\underline{\eta}} = \dot{\underline{\eta}}^T [E] \dot{\underline{\eta}}$$

[E] is the $(t \times t)$ identity matrix

$$(uvw) \sum_{i=1}^{n} m_i \dot{\underline{q}}^i = (uvw) \left[m^1 \,\vdots\, m^2 \ldots \,\vdots\, m^n \right] \begin{pmatrix} \dot{\underline{q}}^1 \\ \dot{\underline{q}}^2 \\ \vdots \\ \dot{\underline{q}}^n \end{pmatrix}$$

$$= (uvw)[m^1 \; m^2 \; \ldots \; m^n][\Phi]\dot{\underline{\eta}}$$

$$\underline{\omega}^T \sum_{i=1}^{n} m_i (\underline{r}^i + \underline{q}^i)^{\sim} \dot{\underline{q}}^i = \underline{\omega}^T [m_1 (\underline{r}^1 + \underline{q}^1)^{\sim} \; \ldots \; m_n (\underline{r}^n + \underline{q}^n)^{\sim}][\Phi]\dot{\underline{\eta}}$$

The kinetic energy can be written as the quadratic form $T = \frac{1}{2} \underline{U}^T [A] \underline{U}$ where [A] is the coefficient matrix (symmetric) of the generalized accelerations appearing in the equations of motion (see Figure 3) and $\underline{U}$ is the vector of non-holonomic velocities

$$\underline{U} = \left(uvw \,\vdots\, \underline{\omega}^T \,\vdots\, \dot{\underline{\eta}}^T \right)^T$$

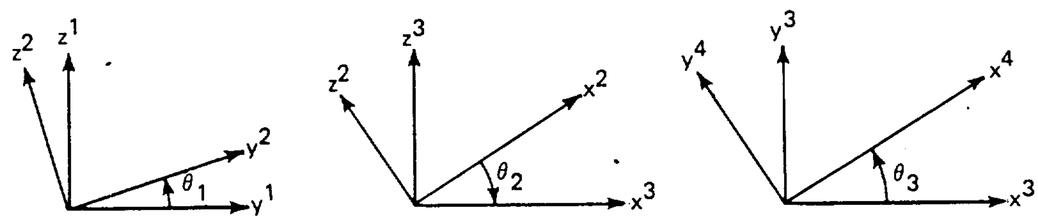
Since $[I_b]$ is positive definite, an inspection of the initial expression for T reveals that $T \geq 0$ for all $\underline{U}$. If $T = 0$ then $\vec{v}_b = \underline{\omega} = \underline{v}^i = \underline{0}$ $(i = 1, 2, \ldots, n)$. But $\vec{v}_b = \underline{0} = \underline{\omega}$ implies $(uvw) = \underline{0}$ and $\underline{v}^i = \underline{0} = (uvw) = \underline{\omega}$ implies $\dot{\underline{q}} = \underline{0}$. Hence $[\Phi]\dot{\underline{\eta}} = \underline{0}$. Since the columns of $[\Phi]$ are linearly independent we must have $\dot{\underline{\eta}} = \underline{0}$ also. In other words, $T = 0$ if and only if $\underline{U} = \underline{0}$. This argument proves that [A] is positive definite and consequently nonsingular (see Chapter 8 where we require $[A]^{-1}$).

Note that if we replace the rigid body by a particle then $\underline{s} = \underline{0}$ and $[I_b] = [0]$. We still have $T \geq 0$ but if $T = 0$ we can only argue that $(uvw) = \underline{0}$. We can have $T = 0$ for nonzero $\underline{\omega}$ and $\dot{\underline{n}}$ as long as $\underline{\omega} \times (\underline{r}^i + \underline{q}^i) + \dot{\underline{q}}^i = 0$ $(i = 1,2,\ldots,n)$. Thus for this later case [A] is positive semi-definite. In particlar [A] will be singular. The situation here can be understood by simply enumerating the degrees of freedom involved. Originally we had a system consisting of a rigid body and n particles: (6 + 3n) degrees of freedom. The number of dynamic equations was also (6 + 3n). When degenerating the rigid body to a particle we have a system of (n + 1) particles: (3n + 3) degrees of freedom. However, when we retain the same equations of motion as in the original case (6 + 3n) there will clearly be a redundancy present. Indeed, this explains why [A] is singular for the degenerate case. Consequently, we cannot use the equations developed here for a system composed solely of particles; at least not without modification.

CHAPTER 7

KINEMATICAL RELATIONSHIPS

Let the transformation from the inertial frame $\{x^1, y^1, z^1\}$ to the body frame $\{x^4, y^4, z^4\}$ be arrived at by a sequence of three Euler angles $\theta_1, \theta_2, \theta_3$ as depicted below.



$$R^{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{pmatrix} \qquad R^{23} = \begin{pmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix}$$

$$R^{34} = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$[R^{ij}]$ is the transformation matrix from frame 'j' to frame 'i'

Concatenating transformations, $[R^{14}] = [R^{12}][R^{23}][R^{34}]$

$$[R^{14}] = \begin{pmatrix} \cos\theta_2\cos\theta_3 & -\cos\theta_2\sin\theta_3 & \sin\theta_2 \\ \cos\theta_1\sin\theta_3 + \sin\theta_1\sin\theta_2\cos\theta_3 & \cos\theta_1\cos\theta_3 - \sin\theta_1\sin\theta_2\sin\theta_3 & -\sin\theta_1\cos\theta_2 \\ \sin\theta_1\sin\theta_3 - \cos\theta_1\sin\theta_2\cos\theta_3 & \sin\theta_1\cos\theta_3 + \cos\theta_1\sin\theta_2\sin\theta_3 & \cos\theta_1\cos\theta_2 \end{pmatrix} \tag{7-1}$$

We next derive the relationship between the body frame angular velocity $\underline{\omega}$ (expressed in body coordinates) and the Euler rates $\dot\theta_1$, $\dot\theta_2$, $\dot\theta_3$. Let $\{\underline{i}_p, \underline{j}_p, \underline{k}_p\}$ be the set of unit vectors along the axes of frame 'p' (p = 1,2,3,4) .

$$\vec\omega = \dot\theta_1\vec i_1 + \dot\theta_2\vec j_2 + \dot\theta_3\vec k_3$$

To express $\vec\omega$ in the body frame, we must use the representation of the unit vectors in frame 4. With the aid of the transformations listed above we arrive at

$$\underline{\omega} = \begin{pmatrix} \dot\theta_1\cos\theta_2\cos\theta_3 + \dot\theta_2\sin\theta_3 \\ \dot\theta_2\cos\theta_3 - \dot\theta_1\cos\theta_2\sin\theta_3 \\ \dot\theta_3 + \dot\theta_1\sin\theta_2 \end{pmatrix} \quad \text{(in body frame)}$$

This system can be inverted to yield

$$\begin{pmatrix} \dot\theta_1 \\ \dot\theta_2 \\ \dot\theta_3 \end{pmatrix} = \begin{bmatrix} \dfrac{\cos\,\theta_3}{\cos\,\theta_2} & \dfrac{-\sin\,\theta_3}{\cos\,\theta_2} & 0 \\ \sin\,\theta_3 & \cos\,\theta_3 & 0 \\ -\cos\,\theta_3\tan\,\theta_2 & \sin\,\theta_3\tan\,\theta_2 & 1 \end{bmatrix} \underline{\omega} \qquad (\cos\,\theta_2 \neq 0) \tag{7-2}$$

CHAPTER 8

EQUATIONS OF MOTION — FIRST ORDER FORM

The assembled motion equations (Figure 3) can be written as

$$[A(t)]\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \hline \dot{\underline{\omega}} \\ \hline \ddot{\underline{\eta}} \end{bmatrix} = \underline{F} + \underline{U} - \begin{bmatrix} \underline{0} \\ \hline \underline{0} \\ \hline \omega_1^2\eta_1 \\ \omega_2^2\eta_2 \\ \vdots \\ \omega_t^2\eta_t \end{bmatrix} \qquad (8\text{-}1)$$

where

$$\underline{F} = \begin{bmatrix} \sum_{i=0}^{n} \underline{f}^i \\ \hline \underline{T}^0 + \sum_{i=1}^{n} (\underline{r}^i + \underline{q}^i) \times \underline{f}^i \\ \hline [\Phi]^T \begin{pmatrix} \underline{f}^1 \\ \underline{f}^2 \\ \vdots \\ \underline{f}^n \end{pmatrix} \end{bmatrix} \qquad (8\text{-}2)$$

and

$$\underline{U} = \begin{bmatrix} \underline{u}_t \\ \hline \underline{u}_r \\ \hline [\Phi]^T \underline{u}_v \end{bmatrix} \tag{8-3}$$

Let R_x, R_y, R_z be the components of the inertial position vector of O_g (origin of body frame) resolved along inertial axes and $[\Gamma]$ denote the matrix in Eq. (7-2). The kinematic relationships can now be written

$$\begin{pmatrix} \dot{R}_x \\ \dot{R}_y \\ \dot{R}_z \end{pmatrix} = [R^{14}] \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \qquad \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = [\Gamma]\underline{\omega}$$

Define $[\Omega^2] = \mathrm{diag}(\omega_1^2, \omega_2^2, \ldots, \omega_t^2)$. The state vector $\underline{Y}$ is defined to be

$$\underline{Y} = (R_x R_y R_z \theta_1 \theta_2 \theta_3 \underline{\eta}^T \; uvw \; \underline{\omega}^T \; \dot{\underline{\eta}}^T)^T \tag{8-4}$$

The equations of motion written in first order form are

$$\frac{d}{dt}\underline{Y} = \begin{bmatrix} [R^{14}] \begin{pmatrix} u \\ v \\ w \end{pmatrix} \\ [\Gamma]\underline{\omega} \\ \dot{\underline{\eta}} \\ A^{-1}\left[\underline{F} + \underline{U} - \begin{pmatrix} \underline{0} \\ [\Omega^2]\underline{\eta} \end{pmatrix}\right] \end{bmatrix} \tag{8-5}$$

This system of (2t + 12) first order equations can be integrated numerically with appropriate initial conditions.

CHAPTER 9

DIGITAL SIMULATION

This chapter is concerned with the FORTRAN computer program which implements and numerically integrates the complete set of first order ordinary differential equations presented in Chapter 8, Eq. (8-5). A description of the main program, its subroutines, and the input data is given. An annotated flowchart of the program is given in Figure 4 and a complete listing of the program and its subroutines is provided in Appendix A. An example of the input data for a sample vehicle is provided in Appendix B. The code is liberally commented throughout and in most instances the FORTRAN variable names are mnemonically similar to the corresponding analytical quantities. Virtually all computations involving real number quantities are performed in (IBM) double precision. External subroutines from the double precision IMSL library[8] are used to perform certain standard computations. IMSL subroutine "EIGRS" is used for eigenvalue/eigenvector extraction and subroutine "LEQT1P" is used to solve simultaneous linear equations. In addition, IMSL subroutine "USPLT" is used to generate time history graphs of selected elements of the vector

$$\{R_x R_y R_z \;\; \theta_1 \theta_2 \theta_3 \;\; \underline{q}^{1^T} \cdots \underline{q}^{n^T} \;\; uvw \;\; \omega_1 \omega_2 \omega_3 \;\; \underline{\dot{q}}^{1^T} \cdots \underline{\dot{q}}^{n^T}\} \qquad (9\text{-}1)$$

via the line printer.

Throughout the program deformation dependent terms are arranged and computed hierarchically as quantities involving structural deflections to the first and second degree. Similarly the nonlinear kinematic terms

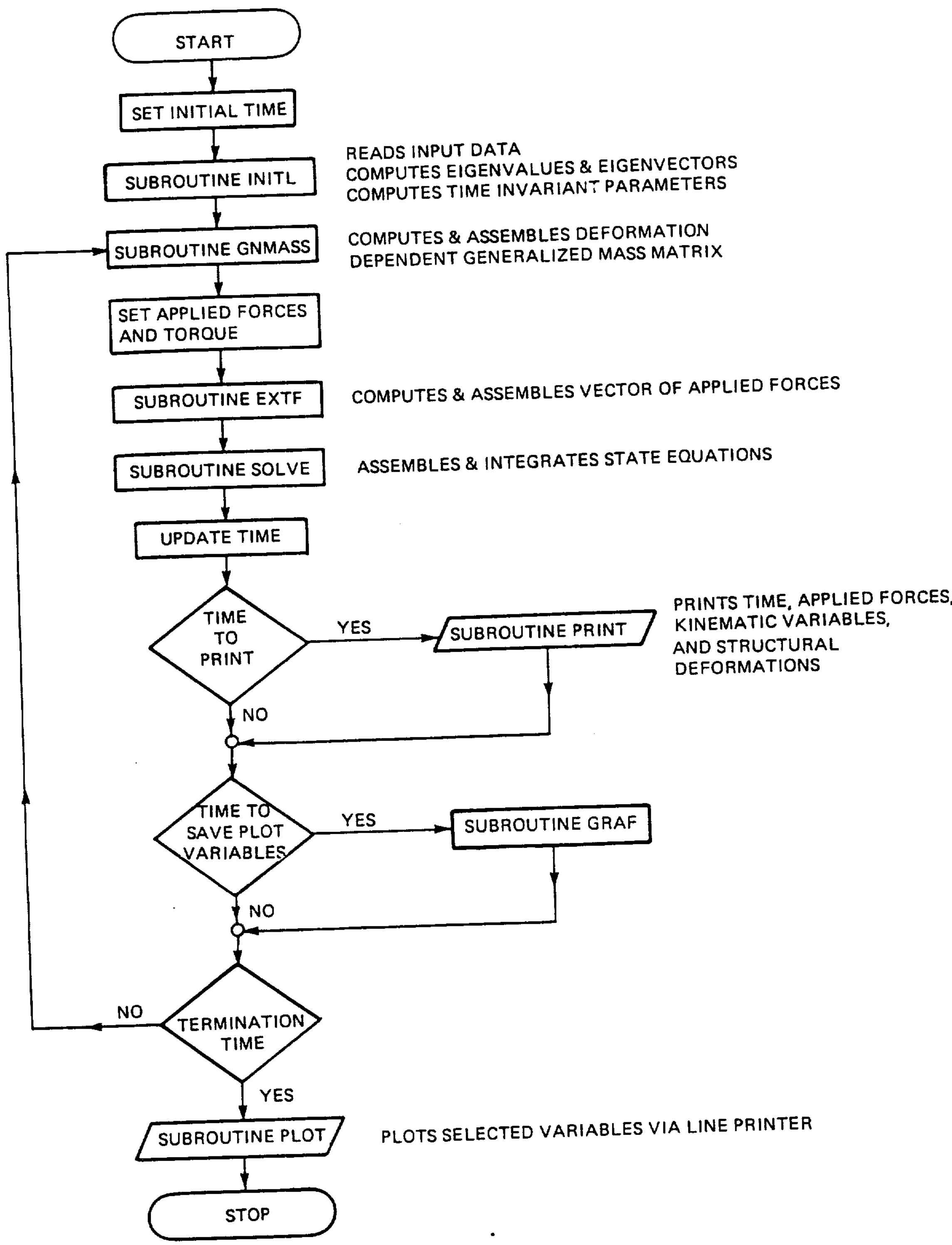


Figure 4. Program flowchart.

are organized into the three categories of Section 4.2 with the contributions of each group of terms being computed independently. This partitioned structure of the computations provides the capability to assess the influence of these higher order terms on the final solution and upon such analysis bypass those deemed negligible.

Main Program

The main program is simply an executive module which calls the appropriate subroutines in the proper order. The reader will note that if external forces are required, these must be explicitly coded either in the main program or as individual subroutines. If the external forces are time dependent, it is essential that they be recomputed prior to each call to subroutine "EXTF" (see comments in main program). For the system in Figure 1 the external excitation is accommodated via the three arrays: FØ,* TAUØ, FP.

FØ — sum of external forces on rigid body

TAUØ — sum of external moments on rigid body taken about body frame origin.

FØ and TAUØ are three-dimensional vectors whose elements refer to components along body frame axes.

FP(I,J) — is the I^{th} component of the external force acting upon particle J in the appendage (I = 1,2,3; J = 1,2,...,N).

The external forces for each of the "N" particles comprising the appendage are resolved along body axes.

Subroutines

Subroutine INITL reads in all program input data and performs consistency checks. Selected input data is echo printed. The eigenvalues

* "Ø" denotes the number zero.

and eigenvectors of the standard symmetric eigenvalue problem given by Eq. (5-2) are computed via a call to IMSL subroutine EIGRS. The eigenvectors are then transformed to those corresponding to Eq. (5-1). All time-invariant terms of the generalized mass matrix of Figure 3 are computed. Finally, the initial conditions on the particle displacements, modal coordinates, and the respective time derivatives are set.

Subroutine GNMASS computes and assembles the deformation dependent generalized mass matrix of Figure 3.

Subroutine EXTF computes and assembles the generalized force vector $\underline{F}$ of Eq. (8-2).

Subroutine NLKT computes and assembles the vector of nonlinear kinematic terms of Eq. (8-3).

Subroutine SOLVE computes the transformation matrices given by Eq. (7-1) and (7-2). The set of simultaneous equations given by Eq. (8-1) are solved via a call to IMSL subroutine LEQT1P. The state vector Eq. (8-4) is assembled and its time derivative, Eq. (8-5), evaluated. The value of the state vector is advanced one time step via a call to subroutine ODESLV.

Subroutine ODESLV integrates the state equation, Eq. (8-5), using the Adams method with third order differences.

Subroutine PRINT is executed only at print-time intervals specified in the input (see below). When called, the subroutine prints the time, force, and torque on the rigid body, applied forces on the particles and all the variables of the vector given in Eq. (9-1).

Entry point GRAF in subroutine PRINT stores selected variables for plotting at a specified time interval (see namelist items DTG and IPLOT below).

Program Input Data

Program input data is read in during execution of subroutine INITL. Input is achieved through four READ-NAMELIST combinations and a single unformatted READ of the stiffness matrix. It is worth noting that while the code given in Appendix A requires the stiffness matrix (described in Section 3.1) and from this and the appendage mass matrix (assembled internally) computes the constrained appendage eigenvalues and eigenvectors, it could be modified to read in the appropriate eigenvalues/eigenvectors directly. The four NAMELIST inputs are defined below, and their use illustrated in Appendix B.

(1) NAMELIST/INPUT/MØ, N, MASS, RM, IØ, S, NT; contains all mass and geometry data as well as the number of modes to be retained.

MØ = mass of rigid body (real)

N = number of particles (integer)

MASS = masses of particles 1 through N (real N × 1 array)

RM = position vectors of particles 1 through N prior to deformation, expressed in body frame (real 3 × N array)

IØ = inertia matrix of the rigid body with respect to a frame located at the rigid body mass center with axes parallel to body frame (real 3 × 3 array)

S = position vector from body frame origin to mass center of rigid body expressed in body frame (real 3 × 1 array)

NT = number of modes to be retained; modes 1 through NT are used (integer)

(2) NAMELIST/KIN/UVW, OMEGA, R, THETA: contains initial conditions for kinematic variables.

UVW = initial velocity vector of body frame origin, expressed in body frame coordinates (real 3 × 1 array)

OMEGA = initial angular velocity vector of body frame with respect to inertial frame, components expressed in body frame (real 3 × 1 array)

R = initial inertial position vector of body frame origin, components expressed in inertial frame (real 3 × 1 array)

THETA = initial 1-2-3 Euler angles of body frame with respect to inertial frame (real 3 × 1 array)

(3) NAMELIST/RUN/DT, TSTOP, DTP, DTG: contains numerical integration parameters and print and plot time intervals.

DT = integration time step in seconds (real)

TSTOP = integration termination time in seconds (real)

DTP = print output time interval in seconds; output printed every DTP seconds (real)

DTG = plot output time interval in seconds; selected variables plotted every DTG seconds (real)

(4) NAMELIST/PLOT/IPLOT: specifies which elements of vector in Eq. (9-1) are to be plotted via line printer.

IPLOT = integer array with the integers corresponding to those elements of the vector in Eq. (9-1) that are to be plotted versus time (see sample use in Appendix B).

APPENDIX A

FORTRAN PROGRAM LISTING

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.271/12.43.05       PAGE   1

REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

               C
               C  ********************************************************************
               C  *THIS PROGRAM SOLVES THE EQUATIONS OF MOTION OF A VEHICLE
               C  *CONSISTING OF A RIGID BASE WITH AN ATTACHED FLEXIBLE APPENDAGE.
               C  *THE APPENDAGE IS IDEALIZED AS A COLLECTION OF PARTICLES CONNECTED
               C  *BY MASSLESS ELASTIC STRUCTURE. IN ADDITION TO EXTERNAL FORCES ACTING
               C  *UPON EACH OF THE PARTICLES, A FORCE AND TORQUE ARE ACCOMMODATED AT
               C  *THE GRAPPLE FIXTURE CORRESPONDING TO THE ORIGIN OF BODY FRAME..
               C  *(WRITTEN BY JOEL STORCH & STEPHEN GATES  C.S.D.L.  BASED UPON
               C  * C.S.D.L.  REPORT R1582  SEPT. 1982)
               C  ********************************************************************
               C
               C
               C   NOTE: ARRAYS ARE DIMENSIONED TO ACCOMODATE A MAXIMUM OF 50 PARTICLES
               C
  ISN 0002           IMPLICIT REAL*8(A-H,O-Z)
  ISN 0003           DIMENSION F0(3),TAU0(3),FP(3,50)
  ISN 0004           COMMON /TIME/ DT,TSTOP,DTP,DTG
  ISN 0005           T=0.0
               C   INPUT PROGRAM DATA AND CALCULATE ALL TIME INVARIANT PARAMETERS.
               C
  ISN 0006           CALL INITL
  ISN 0007           TP=DTP
  ISN 0008           TG=DTG
               C
               C   CALCULATE GENERALIZED MASS MATRIX
               C
  ISN 0009     10    CALL GNMASS
               C
               C   INPUT VALUES REQUIRED FOR 'EXTF'
               C   (ALL VECTORS EXPRESSED IN BODY FRAME)
               C
               C   F0 - EXTERNAL FORCE ON M0 (AT ORIGIN OF BODY FRAME)
               C   TAU0 - EXTERNAL TORQUE AT LOCATION OF BODY FRAME ORIGIN.
               C   FP - VECTOR OF EXTERNAL FORCES ON PARTICLES 1,2,...,N.
               C
  ISN 0010           CALL EXTF(F0,TAU0,FP)
               C
               C   CALCULATE NON-LINEAR KINEMATIC TERMS
               C
  ISN 0011           CALL NLKT
               C
               C   INTEGRATE EQUATIONS OF MOTION
               C
  ISN 0012           CALL SOLVE
  ISN 0013           T=T+DT
  ISN 0014           IF(T .LT. TP) GO TO 15
```

```
                                                                        00005400
  ISN 0026          CALL PRINT(T,F0,TAU0,FP)                            00005500
  ISN 0027          TP=TP+DTP                                           00005600
  ISN 0028     15   IF(T .LT. TG) GO TO 20                              00005700
  ISN 0030          CALL GRAF(T)                                        00005800
  ISN 0031          TG=TG+DTG                                           00005900
  ISN 0032     20   IF(T .LT. TSTOP) GO TO 10                           00006000
  ISN 0034          CALL PLOT                                           00006100
  ISN 0035          STOP                                                00006200
  ISN 0036          END
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      35, PROGRAM SIZE =     2002, SUBPROGRAM NAME =  MAIN

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                280K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.246/12.21.45        PAGE   1

REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

  ISN 0002          SUBROUTINE SKEW(V,A)                                             00006300
            C                                                                        00006400
            C   THIS SUBROUTINE CREATES THE SKEW SYMMETRIC MATRIX CORRESPONDING       00006500
            C   TO THE VECTOR "V".                                                   00006600
            C                                                                        00006700
  ISN 0003          REAL*8 V,A                                                       00006800
  ISN 0004          DIMENSION V(3),A(3,3)                                            00006900
  ISN 0005          A(1,1)=0.0                                                       00007000
  ISN 0006          A(1,2)=-V(3)                                                     00007100
  ISN 0007          A(1,3)=V(2)                                                      00007200
  ISN 0008          A(2,1)=V(3)                                                      00007300
  ISN 0009          A(2,2)=0.0                                                       00007400
  ISN 0010          A(2,3)=-V(1)                                                     00007500
  ISN 0011          A(3,1)=-V(2)                                                     00007600
  ISN 0012          A(3,2)=V(1)                                                      00007700
  ISN 0013          A(3,3)=0.0                                                       00007800
  ISN 0014          RETURN                                                           00007900
  ISN 0015          END                                                              00008000

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       14, PROGRAM SIZE =       388, SUBPROGRAM NAME =  SKEW

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  280K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.246/12.21.46        PAGE   1

REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
                                                                                          00008100
  ISN 0002            SUBROUTINE CROSS(A,B,C)                                             00008200
              C                                                                           00008300
              C   THIS SUBROUTINE CALCULATES THE VECTOR CROSS PRODUCT                     00008400
              C                  A X B =C                                                 00008500
              C                                                                           00008600
  ISN 0003            REAL*8 A(3),B(3),C(3)                                               00008700
  ISN 0004            C(1)=A(2)*B(3)-A(3)*B(2)                                            00008800
  ISN 0005            C(2)=A(3)*B(1)-A(1)*B(3)                                            00008900
  ISN 0006            C(3)=A(1)*B(2)-A(2)*B(1)                                            00009000
  ISN 0007            RETURN                                                              00009100
  ISN 0008            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =        7, PROGRAM SIZE =       484, SUBPROGRAM NAME = CROSS

*STATISTICS*  NO  DIAGNOSTICS GENERATED
                                                                 280K BYTES OF CORE NOT USED
****** END OF COMPILATION ******
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002            SUBROUTINE INITL
              C
              C   THIS SUBROUTINE READS IN PROGRAM DATA AND CALCULATES ALL TIME
              C   INVARIANT PARAMETERS.
              C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            REAL*8 MASS,M0,K,MCO,INERT1,MRM,I0,MOS
ISN 0005            REAL*4 PLTDAT
ISN 0006            DIMENSION MASS(50),RM(3,50),K(150,150),MC0(3),INERT1(3,3),
                   1 MRM(3,50),CMAT(3,3),A12(3,3),A23(3,150),A(156,156),Q(3,50),
                   1 QDOT(3,50),UVW(3),OMEGA(3),R(3),THETA(3),I0(3,3),S(3),MOS(3),
                   2 AV(11325),FREQ(150),EV(150,150),WK(150),IPLOT(42),PLTDAT(100,20),
                   3 WKM(3,150),ETA(150),ETAD(150)
ISN 0007            COMMON /CONST/ TM,MASS,MC0,MRM,CMAT,N,N3,N3P6,NT,NTP6,N0
ISN 0008            COMMON /AMAT/ A,A12,INERT1,A23
ISN 0009            COMMON /STATE/ R,THETA,Q,UVW,OMEGA,QDOT
ISN 0010            COMMON /GEOM/ RM
ISN 0011            COMMON /TIME/ DT,TSTOP,DTP,DTG
ISN 0012            COMMON /RIGID/ MOS,I0,S
ISN 0013            COMMON /PLOTT/ PLTDAT,IPLOT,NP
ISN 0014            COMMON /MODCO/ ETA,ETAD
ISN 0015            COMMON /MODES/ EV,FREQ
ISN 0016            EQUIVALENCE(K(1,1),EV(1,1))
ISN 0017            NAMELIST /INPUT/ M0,N,MASS,RM,I0,S,NT
ISN 0018            NAMELIST /KIN/ UVW,OMEGA,R,THETA
ISN 0019            NAMELIST /RUN/ DT,TSTOP,DTP,DTG
ISN 0020            NAMELIST /PLOT/ IPLOT
              C
              C             DESCRIPTION OF NAMELIST VARIABLES
              C
              C   "M0" IS THE MASS OF THE RIGID BASE
              C
              C   "N" IS THE NUMBER OF PARTICLES THAT COMPRISE THE FLEXIBLE APPENDAGE
              C
              C   "MASS" CONTAINS THE MASSES OF PARTICLES 1 TO N
              C
              C   "RM" CONTAINS THE POSITION VECTORS OF PARTICLES 1 THRU N  IN THE
              C    UNDEFORMED STATE EXPRESSED IN THE BODY FRAME.
              C
              C   "I0" IS THE INERTIA MATRIX OF THE RIGID BASE WITH RESPECT TO A
              C        FRAME LOCATED AT THE MASS CENTER OF THE BASE AND PARALLEL
              C        TO THE BODY FIXED AXIS SYSTEM.
              C
              C   "S"  IS THE VECTOR FROM THE BODY FRAME ORIGIN TO THE MASS CENTER
              C        OF THE RIGID BODY.
              C
              C   "NT" NUMBER OF RETAINED MODES IN APPENDAGE VIBRATION
              C
              C   "UVW" IS THE INITIAL VELOCITY OF THE BODY FRAME ORIGIN EXPRESSED
              C         IN BODY COORDINATES.
              C
              C   "OMEGA" IS THE INITIAL ANGULAR VELOCITY OF THE BODY FRAME EXPRESSED
              C           IN BODY COORDINATES.
```

```
                C
                C
                C   "R" IS THE INITIAL INERTIAL POSITION VECTOR OF THE BODY FRAME
                C      ORIGIN
                C
                C   "THETA" IS THE INITIAL SET OF ATTITUDE ANGLES FOR THE BODY FRAME
                C
                C   "DT" IS THE INTEGRATION TIME STEP
                C
                C   "TSTOP" IS THE TERMINAL TIME FOR THE SIMULATION
                C
                C   "DTP" IS THE TIME INTERVAL BETWEEN PRINTOUTS
                C
                C   "DTG" IS THE TIME INTERVAL BETWEEN PLOTTED POINTS
                C
                C   "IPLOT" IS AN ARRAY INDICATING VARIABLES TO BE PLOTTED.
                C          NUMBERING CORRESPONDS TO LOCATION IN STATE VECTOR.
                C          POINTS ARE PLOTTED EVERY "DTG" SECONDS.
                C
ISN 0021              READ(5,INPUT)
ISN 0022              IF(NT .LE. 3*N) GO TO 8
ISN 0024              WRITE(6,112) NT,N
ISN 0025              STOP
ISN 0026        8     WRITE(6,100) M0
ISN 0027              DO 10 I=1,N
ISN 0028        10    WRITE(6,101) I,MASS(I),(RM(J,I),J=1,3)
ISN 0029        100   FORMAT(1H1,20X,'MASS OF RIGID BASE=',E12.4,' SLUGS',//,1H ,T10,
                     1 'PARTICLE',T21,'MASS(SLUGS)',T41,'POSITION(FT.)',/)
ISN 0030        101   FORMAT(1H ,T12,I3,T21,F7.2,T34,3(F7.2,2X))
ISN 0031              WRITE(6,106) ((I0(I,J),J=1,3),I=1,3)
ISN 0032              WRITE(6,107) S,NT
ISN 0033              READ(5,KIN)
ISN 0034              WRITE(6,104) R,THETA
ISN 0035        103   FORMAT(1H0,5X,'INITIAL VELOCITY=',3F7.2,3X,'FT/SEC',4X,
                     1 'INITIAL ANGULAR VELOCITY=',3F7.2,'  DEG/SEC')
ISN 0036              WRITE(6,103) UVW,OMEGA
ISN 0037        104   FORMAT(1H0,5X,'INITIAL POSITION=',3F7.2,3X,'FT',4X,
                     1 '  INITIAL ATTITUDE=',3F8.3,' DEG')
ISN 0038              READ(5,RUN)
ISN 0039              WRITE(6,105) DT,TSTOP,DTP,DTG
ISN 0040              READ(5,PLOT)
ISN 0041              II=0
ISN 0042              DO 108 I=1,42
ISN 0043              IF(IPLOT(I) .EQ. 0) GO TO 109
ISN 0045        108   II=II+1
ISN 0046        109   IF(II .EQ. 0) GO TO 111
ISN 0048              WRITE(6,110) (IPLOT(I),I=1,II)
ISN 0049        110   FORMAT(1H0,5X,'VARIABLES PLOTTED',4X,42(I2,1X))
ISN 0050        105   FORMAT(1H0,5X,'TIME STEP=',E12.4,' SEC',3X,'TERMINATION TIME=',
                     1 E12.4,' SEC',2X,'PRINT INTERVAL=',E12.4,' SEC','PLOT INTERVAL=',
                     2 E12.4,' SEC')
ISN 0051        112   FORMAT(1H0,3X,I2,' MODES REQUESTED',2X,I3,' PARTICLES IN MODEL')
ISN 0052        106   FORMAT(1H0,T15,'INERTIA MATRIX OF RIGID BODY(SLUG FT**2)',
                     1 //,(T15,3E13.5))
ISN 0053        107   FORMAT(1H0,15X,'S=',3E13.5,' FT',3X,I2,' CONSTRINED APPENDAGE MODE
                     1S RETAINED')
                C
                C   CHANGE ANGULAR VELOCITY & ATTITUDE TO RADIAN MEASURE
```

```
                C
ISN 0054        111   DTR=DATAN(1.000)/45.
ISN 0055              DO 15 I=1,3
ISN 0056              OMEGA(I)=DTR*OMEGA(I)
ISN 0057        15    THETA(I)=DTR*THETA(I)
                C
                C   TM - TOTAL BODY MASS
                C
ISN 0058              TM=M0
ISN 0059              DO 20 I=1,N
ISN 0060        20    TM=TM+MASS(I)
ISN 0061              DO 30 J=1,3
ISN 0062              MOS(J)=M0*S(J)
ISN 0063        30    MC0(J)=0.0
ISN 0064              DO 40 I=1,N
ISN 0065              DO 45 J=1,3
ISN 0066              MRM(J,I)=MASS(I)*RM(J,I)
ISN 0067        45    MC0(J)=MC0(J)+MRM(J,I)
ISN 0068        40    CONTINUE
ISN 0069              DO 42 I=1,3
ISN 0070        42    MC0(I)=MC0(I)+MOS(I)
ISN 0071              N3=3*N
ISN 0072              N3P6=N3+6
ISN 0073              NTP6=NT+6
ISN 0074              NO=2*NTP6
ISN 0075              NP=0
                C
                C   READ IN STIFFNESS MATRIX
                C
ISN 0076              READ(8) NDOF,((K(I,J),J=1,NDOF),I=1,NDOF)
ISN 0077              IF(NDOF .EQ. N3) GO TO 400
ISN 0079              WRITE(6,102) N,NDOF
ISN 0080              STOP
ISN 0081        400   CONTINUE
ISN 0082        102   FORMAT(1H0,10X,'INCONSISTENT DATA',2X,I3,' PARTICLES',2X,I3,
                     1 ' DEGREES OF FREEDOM IN STIFFNESS MATRIX')
                C
                C   GET CONSTRAINED FREQUENCIES AND MODE SHAPES OF APPENDAGE
                C
ISN 0083              L=1
ISN 0084              DO 300 J=1,N3
ISN 0085              LC=1+J/3
ISN 0086              IF( (J-3*(J/3)) .EQ. 0) LC=LC-1
ISN 0088              DO 300 I=1,J
ISN 0089              LR=1+I/3
ISN 0090              IF( (I-3*(I/3)) .EQ. 0) LR=LR-1
ISN 0092              AV(L)=K(I,J)/DSQRT(MASS(LR)*MASS(LC))
ISN 0093              L=L+1
ISN 0094        300   CONTINUE
ISN 0095              CALL EIGRS(AV,N3,1,FREQ,EV,150,WK,IER)
ISN 0096              IF(IER .EQ. 0) GO TO 310
ISN 0098              WRITE(6,301) IER
ISN 0099        301   FORMAT(1H0,10X,'ERROR FROM IMSL ROUTINE "EIGRS" ERROR CODE=',I4)
ISN 0100              STOP
                C
                C   TRANSFORM EIGENVECTORS
                C
ISN 0101        310   DO 311 L=1,N
```

```
ISN 0102          C1=DSQRT(MASS(L))
ISN 0103          DO 311 I=1,3
ISN 0104          IR=3*(L-1)+I
ISN 0105          DO 311 J=1,N3
ISN 0106    311   EV(IR,J)=EV(IR,J)/C1
ISN 0107          DO 320 I=1,N3
ISN 0108          FHZ=DSQRT(FREQ(I))/6.283185
ISN 0109          WRITE(6,321) I,FHZ,(EV(J,I),J=1,N3)
ISN 0110    321   FORMAT(1H0,3X,'MODE ',I2,' FREQUENCY=',E12.4,1X,'(HZ)',/,1H ,
                 1 'MODE SHAPE: ',(9(G12.4,1X)))
ISN 0111    320   CONTINUE
            C
            C    THE ROUTINE "EIGRS" RETURNS AN ORTHONORMAL SET OF EIGENVECTORS.
            C    THIS IS ESSENTIAL SINCE WE ASSUME IN THE DERIVATION THAT THE
            C    EIGENVECTORS(OF THE ORIGINAL GENERALIZED EIGENVALUE PROBLEM)
            C    ARE ORTHOGONAL WITH RESPECT TO MASS(STIFFNESS) AND NORMALIZED
            C    WITH RESPECT TO MASS.
            C
            C
            C    CALCULATE TIME INVARIANT PART OF INERTIA MATRIX "INERT1"
            C    AND "CMAT"
            C
ISN 0112          DO 50 I=1,3
ISN 0113          DO 50 J=1,3
ISN 0114          INERT1(I,J)=0.0
ISN 0115          CMAT(I,J)=0.0
ISN 0116     50   CONTINUE
ISN 0117          DO 60 I=1,N
ISN 0118          XS=RM(1,I)**2
ISN 0119          YS=RM(2,I)**2
ISN 0120          ZS=RM(3,I)**2
ISN 0121          INERT1(1,1)=INERT1(1,1)+MASS(I)*(YS+ZS)
ISN 0122          INERT1(1,2)=INERT1(1,2)-MASS(I)*RM(1,I)*RM(2,I)
ISN 0123          INERT1(1,3)=INERT1(1,3)-MASS(I)*RM(1,I)*RM(3,I)
ISN 0124          INERT1(2,2)=INERT1(2,2)+MASS(I)*(XS+ZS)
ISN 0125          INERT1(2,3)=INERT1(2,3)-MASS(I)*RM(2,I)*RM(3,I)
ISN 0126          INERT1(3,3)=INERT1(3,3)+MASS(I)*(XS+YS)
ISN 0127          CMAT(1,1)=CMAT(1,1)+MASS(I)*XS
ISN 0128          CMAT(2,2)=CMAT(2,2)+MASS(I)*YS
ISN 0129          CMAT(3,3)=CMAT(3,3)+MASS(I)*ZS
ISN 0130     60   CONTINUE
ISN 0131          CMAT(1,2)=-INERT1(1,2)
ISN 0132          CMAT(1,3)=-INERT1(1,3)
ISN 0133          CMAT(2,3)=-INERT1(2,3)
ISN 0134          INERT1(1,1)=INERT1(1,1)+I0(1,1)+M0*(S(2)**2+S(3)**2)
ISN 0135          INERT1(1,2)=INERT1(1,2)+I0(1,2)-M0*S(1)*S(2)
ISN 0136          INERT1(1,3)=INERT1(1,3)+I0(1,3)-M0*S(1)*S(3)
ISN 0137          INERT1(2,2)=INERT1(2,2)+I0(2,2)+M0*(S(1)**2+S(3)**2)
ISN 0138          INERT1(2,3)=INERT1(2,3)+I0(2,3)-M0*S(2)*S(3)
ISN 0139          INERT1(3,3)=INERT1(3,3)+I0(3,3)+M0*(S(1)**2+S(2)**2)
ISN 0140          DO 70 I=1,3
ISN 0141          DO 70 J=1,3
ISN 0142          IF(I .LE. J) GO TO 70
ISN 0144          INERT1(I,J)=INERT1(J,I)
ISN 0145          CMAT(I,J)=CMAT(J,I)
ISN 0146     70   CONTINUE
            C
            C    CREATE TIME INVARIANT PORTIONS OF GENERALIZED MASS MATRIX "A"
```

```
                C
                C   (1,2) PARTITION "A12"
                C
ISN 0147              CALL SKEW(MCO,A12)
ISN 0148              DO 80 I=1,3
ISN 0149              DO 80 J=1,3
ISN 0150              IF(I .EQ. J) GO TO 80
ISN 0152              A12(I,J)=-A12(I,J)
ISN 0153        80    CONTINUE
                C
                C   (2,3) PARTITION "A23"
                C
ISN 0154              DO 90 I=1,N
ISN 0155              L=3*I-2
ISN 0156        90    CALL SKEW(MRM(1,I),WKM(1,L))
ISN 0157              DO 92 I=1,3
ISN 0158              DO 92 J=1,NT
ISN 0159              A23(I,J)=0.0
ISN 0160              DO 94 L=1,N3
ISN 0161        94    A23(I,J)=A23(I,J)+WKM(I,L)*EV(L,J)
ISN 0162        92    CONTINUE
                C
                C   STORE CONSTANT PARTITIONS OF "A"
                C
ISN 0163              DO 200 I=1,NTP6
ISN 0164              DO 200 J=1,NTP6
ISN 0165              A(I,J)=0.0
ISN 0166        200   CONTINUE
                C
                C   CREATE (1,1) PARTITION
                C
ISN 0167              DO 210 I=1,3
ISN 0168              DO 210 J=1,3
ISN 0169              IF(I .EQ. J) A(I,J)=TM
ISN 0171        210   CONTINUE
                C
                C   CREATE (1,3) PARTITION
                C
ISN 0172              DO 215 I=1,3
ISN 0173              DO 215 J=1,N3
ISN 0174        215   WKM(I,J)=0.0
ISN 0175              DO 220 L=1,N
ISN 0176              JS=3*L-2
ISN 0177              DO 221 I=1,3
ISN 0178              WKM(I,JS)=MASS(L)
ISN 0179              JS=JS+1
ISN 0180        221   CONTINUE
ISN 0181        220   CONTINUE
ISN 0182              DO 283 I=1,3
ISN 0183              DO 283 J=1,NT
ISN 0184              JP6=J+6
ISN 0185              A(I,JP6)=0.0
ISN 0186              DO 281 L=1,N3
ISN 0187        281   A(I,JP6)=A(I,JP6)+WKM(I,L)*EV(L,J)
ISN 0188        283   CONTINUE
                C
                C   CREATE (3,3) PARTITION
                C
```

```
                                                                         00037700
  ISN 0189           DO 230 L=1,NT                                       00037800
  ISN 0190     230   A(L+6,L+6)=1.                                       00037900
               C                                                         00038000
               C   SET INITIAL DEFORMATION AND RATE TO ZERO              00038100
               C                                                         00038200
  ISN 0191           DO 250 I=1,N                                        00038300
  ISN 0192           DO 250 J=1,3                                        00038400
  ISN 0193           Q(J,I)=0.0                                          00038500
  ISN 0194           QDOT(J,I)=0.0                                       00038600
  ISN 0195     250   CONTINUE                                            00038700
  ISN 0196           DO 293 I=1,NT                                       00038800
  ISN 0197           ETA(I)=0.0                                          00038900
  ISN 0198           ETAD(I)=0.0                                         00039000
  ISN 0199     293   CONTINUE                                            00039100
  ISN 0200           RETURN                                              00039200
  ISN 0201           END
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*     SOURCE STATEMENTS =      200, PROGRAM SIZE =    103026, SUBPROGRAM NAME = INITL
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                              232K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

  ISN 0002              SUBROUTINE GNMASS
                  C
                  C   THIS SUBROUTINE CALCULATES THE GENERALIZED MASS MATRIX "A"
                  C
  ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
  ISN 0004              REAL*8 MASS,MC0,MRM,INERT1,MQ,MC1,INERT2,INERT3
  ISN 0005              DIMENSION MASS(50),MC0(3),MRM(3,50),CMAT(3,3),A(156,156),
                       1 A12(3,3),INERT1(3,3),A23(3,150),Q(3,50),MQ(3,50),MC1(3),
                       2 WM(3,3),INERT2(3,3),A23Q(3,150),INERT3(3,3),QDOT(3,50),UVW(3),
                       3 OMEGA(3),R(3),THETA(3)
  ISN 0006              COMMON /CONST/ TM,MASS,MC0,MRM,CMAT,N,N3,N3P6,NT,NTP6,NO
  ISN 0007              COMMON /AMAT/ A,A12,INERT1,A23
  ISN 0008              COMMON /STATE/ R,THETA,Q,UVW,OMEGA,QDOT
  ISN 0009              COMMON /TDEPV/ MQ,MC1
  ISN 0010              COMMON /MODES/ EV(150,150),FREQ(150)
  ISN 0011              DO 10 I=1,3
  ISN 0012        10    MC1(I)=0.0
  ISN 0013              DO 40 I=1,N
  ISN 0014              DO 45 J=1,3
  ISN 0015              MQ(J,I)=MASS(I)*Q(J,I)
  ISN 0016        45    MC1(J)=MC1(J)+MQ(J,I)
  ISN 0017        40    CONTINUE
                  C
                  C   COMPUTE "A" NEGLECTING TIME DEPENDENT(DEFORMATION DEPENDENT) TERMS
                  C
  ISN 0018              DO 50 J=1,3
  ISN 0019              JP3=J+3
  ISN 0020              DO 50 I=1,3
  ISN 0021        50    A(I,JP3)=A12(I,J)
  ISN 0022              DO 60 I=1,3
  ISN 0023              IP3=I+3
  ISN 0024              DO 60 J=1,3
  ISN 0025              JP3=J+3
  ISN 0026        60    A(IP3,JP3)=INERT1(I,J)
  ISN 0027              DO 70 J=1,NT
  ISN 0028              JP6=J+6
  ISN 0029              DO 70 I=1,3
  ISN 0030              IP3=I+3
  ISN 0031        70    A(IP3,JP6)=A23(I,J)
                  C
                  C   ADD IN FIRST ORDER DEFORMATION TERMS
                  C
  ISN 0032              CALL SKEW(MC1,WM)
  ISN 0033              DO 80 J=1,3
  ISN 0034              JP3=J+3
  ISN 0035              DO 80 I=1,3
  ISN 0036        80    A(I,JP3)=A(I,JP3)-WM(I,J)
                  C
                  C   FIRST ORDER DEFORMATION TERMS IN INERTIA MATRIX - "INERT2"
                  C
  ISN 0037              DO 85 I=1,3
  ISN 0038              DO 85 J=1,3
  ISN 0039        85    INERT2(I,J)=0.0
```

```
ISN 0040          DO 90 L=1,3
ISN 0041          SUM=0.0
ISN 0042          DO 100 I=1,N
ISN 0043    100   SUM=SUM+MRM(L,I)*Q(L,I)
ISN 0044          SUM=2.0*SUM
ISN 0045          LL=L+1
ISN 0046          DO 110 II=1,2
ISN 0047          IF(LL .GT. 3) LL=1
ISN 0049          INERT2(LL,LL)=INERT2(LL,LL)+SUM
ISN 0050          LL=LL+1
ISN 0051    110   CONTINUE
ISN 0052    90    CONTINUE
ISN 0053          SUM=0.0
ISN 0054          DO 120 I=1,N
ISN 0055    120   SUM=SUM-(MRM(1,I)*Q(2,I)+MRM(2,I)*Q(1,I))
ISN 0056          INERT2(1,2)=SUM
ISN 0057          SUM=0.0
ISN 0058          DO 130 I=1,N
ISN 0059    130   SUM=SUM-(MRM(1,I)*Q(3,I)+MRM(3,I)*Q(1,I))
ISN 0060          INERT2(1,3)=SUM
ISN 0061          SUM=0.0
ISN 0062          DO 140 I=1,N
ISN 0063    140   SUM=SUM-(MRM(2,I)*Q(3,I)+MRM(3,I)*Q(2,I))
ISN 0064          INERT2(2,3)=SUM
ISN 0065          DO 150 I=1,3
ISN 0066          DO 150 J=1,3
ISN 0067          IF(I .LE. J) GO TO 150
ISN 0069          INERT2(I,J)=INERT2(J,I)
ISN 0070    150   CONTINUE
ISN 0071          DO 160 I=1,3
ISN 0072          IP3=I+3
ISN 0073          DO 160 J=1,3
ISN 0074          JP3=J+3
ISN 0075    160   A(IP3,JP3)=A(IP3,JP3)+INERT2(I,J)
ISN 0076          DO 170 I=1,N
ISN 0077          L=3*I-2
ISN 0078    170   CALL SKEW(MQ(1,I),A23Q(1,L))
ISN 0079          DO 180 I=1,3
ISN 0080          IP3=I+3
ISN 0081          DO 180 J=1,NT
ISN 0082          JP6=J+6
ISN 0083          DO 185 L=1,N3
ISN 0084    185   A(IP3,JP6)=A(IP3,JP6)+A23Q(I,L)*EV(L,J)
ISN 0085    180   CONTINUE
            C
            C  ADD IN SECOND ORDER DEFORMATION TERMS - "INERT3"
            C
ISN 0086          DO 190 I=1,3
ISN 0087          DO 190 J=1,3
ISN 0088    190   INERT3(I,J)=0.0
ISN 0089          DO 200 L=1,3
ISN 0090          SUM=0.0
ISN 0091          DO 210 I=1,N
ISN 0092    210   SUM=SUM+MQ(L,I)*Q(L,I)
ISN 0093          LL=L+1
ISN 0094          DO 220 II=1,2
ISN 0095          IF(LL .GT. 3) LL=1
ISN 0097          INERT3(LL,LL)=INERT3(LL,LL)+SUM
```

```
  ISN 0098          LL=LL+1                                              00050400
  ISN 0099    220   CONTINUE                                             00050500
  ISN 0100    200   CONTINUE                                             00050600
  ISN 0101          SUM=0.0                                              00050700
  ISN 0102          DO 240 I=1,N                                         00050800
  ISN 0103    240   SUM=SUM-MQ(1,I)*Q(2,I)                               00050900
  ISN 0104          INERT3(1,2)=SUM                                      00051000
  ISN 0105          SUM=0.0                                              00051100
  ISN 0106          DO 250 I=1,N                                         00051200
  ISN 0107    250   SUM=SUM-MQ(1,I)*Q(3,I)                               00051300
  ISN 0108          INERT3(1,3)=SUM                                      00051400
  ISN 0109          SUM=0.0                                              00051500
  ISN 0110          DO 260 I=1,N                                         00051600
  ISN 0111    260   SUM=SUM-MQ(2,I)*Q(3,I)                               00051700
  ISN 0112          INERT3(2,3)=SUM                                      00051800
  ISN 0113          DO 270 I=1,3                                         00051900
  ISN 0114          DO 270 J=1,3                                         00052000
  ISN 0115          IF(I .LE. J) GO TO 270                               00052100
  ISN 0117          INERT3(I,J)=INERT3(J,I)                              00052200
  ISN 0118    270   CONTINUE                                             00052300
  ISN 0119          DO 280 I=1,3                                         00052400
  ISN 0120          IP3=I+3                                              00052500
  ISN 0121          DO 280 J=1,3                                         00052600
  ISN 0122          JP3=J+3                                              00052700
  ISN 0123    280   A(IP3,JP3)=A(IP3,JP3)+INERT3(I,J)                    00052800
              C                                                          00052900
  ISN 0124    302   FORMAT(1H0,//,1H ,10X,'A MATRIX')                    00053000
  ISN 0125    301   FORMAT(1H0,2X,15(F7.2,1X))                           00053100
  ISN 0126          RETURN                                               00053200
  ISN 0127          END                                                  00053300

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =     126, PROGRAM SIZE =     8156, SUBPROGRAM NAME =GNMASS

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                          256K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

  ISN 0002             SUBROUTINE EXTF(F0,TAU0,FP)
                 C
                 C    THIS SUBROUTINE ASSEMBLES THE FORCE VECTOR  "F" IN THE
                 C    MOTION EQUATIONS AND IS PARTITIONED AS: FORCRES FOR BODY
                 C    TRANSLATION, FORCES FOR BODY ROTATION, AND FORCES FOR
                 C    PARTICLE TRANSLATION.
                 C
                 C    INPUT TO SUBROUTINE
                 C
                 C    F0 - EXTERNAL FORCE ON M0 (AT ORIGIN OF BODY FRAME)
                 C    TAU0 - EXTERNAL TORQUE AT LOCATION OF BODY FRAME ORIGIN.
                 C    FP - VECTOR OF EXTERNAL FORCES ON PARTICLES &,2,...,N.
                 C    (ALL VECTORS EXPRESSED IN BODY FRAME)
                 C
  ISN 0003             IMPLICIT REAL*8(A-H,O-Z)
  ISN 0004             REAL*8 MASS,MC0,MRM
  ISN 0005             DIMENSION F0(3),TAU0(3),FP(3,50),Q(3,50),RM(3,50),MASS(50),MC0(3),
                      1 MRM(3,50),CMAT(3,3),F(156),SUM(3),WV(3),WV1(3),QDOT(3,50),UVW(3),
                      2 OMEGA(3),R(3),THETA(3),WV2(150),PHI(150,150),WS(150)
  ISN 0006             COMMON /STATE/ R,THETA,Q,UVW,OMEGA,QDOT
  ISN 0007             COMMON /GEOM/ RM
  ISN 0008             COMMON /CONST/ TM,MASS,MC0,MRM,CMAT,N,N3,N3P6,NT,NTP6,NO
  ISN 0009             COMMON /FORCE/ F
  ISN 0010             COMMON /MODES/ PHI,WS
  ISN 0011             DO 10 J=1,3
  ISN 0012       10    F(J)=F0(J)
  ISN 0013             DO 20 I=1,N
  ISN 0014             DO 20 J=1,3
  ISN 0015       20    F(J)=F(J)+FP(J,I)
  ISN 0016             DO 30 J=1,3
  ISN 0017       30    SUM(J)=0.0
  ISN 0018             DO 40 I=1,N
  ISN 0019             DO 50 J=1,3
  ISN 0020       50    WV(J)=RM(J,I)+Q(J,I)
  ISN 0021             CALL CROSS(WV,FP(1,I),WV1)
  ISN 0022             DO 60 J=1,3
  ISN 0023       60    SUM(J)=SUM(J)+WV1(J)
  ISN 0024       40    CONTINUE
  ISN 0025             DO 70 J=1,3
  ISN 0026             JP3=J+3
  ISN 0027       70    F(JP3)=TAU0(J)+SUM(J)
  ISN 0028             L=1
  ISN 0029             DO 80 I=1,N
  ISN 0030             DO 85 J=1,3
  ISN 0031             WV2(L)=FP(J,I)
  ISN 0032             L=L+1
  ISN 0033       85    CONTINUE
  ISN 0034       80    CONTINUE
  ISN 0035             DO 90 I=1,NT
  ISN 0036             IP6=I+6
  ISN 0037             F(IP6)=0.0
  ISN 0038             DO 95 L=1,N3
  ISN 0039       95    F(IP6)=F(IP6)+PHI(L,I)*WV2(L)
```

```
  ISN 0040      90    CONTINUE                                                    00058700
  ISN 0041            RETURN                                                      00058800
  ISN 0042            END                                                         00058900

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      41, PROGRAM SIZE =      2734, SUBPROGRAM NAME =  EXTF

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  280K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002             SUBROUTINE NLKT
                C
                C   THIS SUBROUTINE CALCULATES THE NON-LINEAR KINEMATIC TERMS IN THE
                C   MOTION EQUATIONS. THESE TERMS ARE ASSEMBLED INTO THE VECTOR "U".
                C
ISN 0003             IMPLICIT REAL*8(A-H,O-Z)
ISN 0004             REAL*8 MASS,MC0,MRM,MQ,MC1,MOS,I0
ISN 0005             DIMENSION MASS(50),MC0(3),MRM(3,50),CMAT(3,3),MQ(3,50),MC1(3),
                    1 QDOT(3,50),UVW(3),OMEGA(3),UT(3),UR(3),UV(150),U(156),WV1(3),
                    2 WV2(3),WV3(3),WV4(3),WM1(3,3),Q(3,50),R(3),THETA(3),MOS(3),
                    3 I0(3,3),S(3),WS(150),PHI(150,150)
ISN 0006             COMMON /CONST/ TM,MASS,MC0,MRM,CMAT,N,N3,N3P6,NT,NTP6,NO
ISN 0007             COMMON /TDEPV/ MQ,MC1
ISN 0008             COMMON /STATE/ R,THETA,Q,UVW,OMEGA,QDOT
ISN 0009             COMMON /FICFRC/ U
ISN 0010             COMMON /RIGID/ MOS,I0,S
ISN 0011             COMMON /MODES/ PHI,WS
ISN 0012             EQUIVALENCE(U(1),UT(1)),(U(4),UR(1))
                C
                C   CALCULATE DEFORMATION INDEPENDENT TERMS
                C
ISN 0013             CALL CROSS(UVW,OMEGA,WV1)
ISN 0014             SUM=OMEGA(1)*MC0(1)+OMEGA(2)*MC0(2)+OMEGA(3)*MC0(3)
ISN 0015             OMS=OMEGA(1)**2+OMEGA(2)**2+OMEGA(3)**2
ISN 0016             DO 20 J=1,3
ISN 0017        20   UT(J)=TM*WV1(J)-SUM*OMEGA(J)+OMS*MC0(J)
ISN 0018             CALL CROSS(MC0,WV1,WV2)
ISN 0019             DO 30 J=1,3
ISN 0020        30   WV3(J)=CMAT(J,1)*OMEGA(1)+CMAT(J,2)*OMEGA(2)+CMAT(J,3)*OMEGA(3)
ISN 0021             CALL CROSS(OMEGA,WV3,WV4)
ISN 0022             DO 40 J=1,3
ISN 0023        40   UR(J)=WV2(J)+WV4(J)
ISN 0024             SUM=OMEGA(1)*S(1)+OMEGA(2)*S(2)+OMEGA(3)*S(3)
ISN 0025             CALL CROSS(MOS,OMEGA,WV2)
ISN 0026             DO 42 J=1,3
ISN 0027        42   WV3(J)=I0(J,1)*OMEGA(1)+I0(J,2)*OMEGA(2)+I0(J,3)*OMEGA(3)
ISN 0028             CALL CROSS(OMEGA,WV3,WV4)
ISN 0029             DO 44 J=1,3
ISN 0030        44   UR(J)=UR(J)-SUM*WV2(J)-WV4(J)
ISN 0031             DO 50 I=1,N
ISN 0032             L=3*I-2
ISN 0033             DO 55 J=1,3
ISN 0034        55   WV2(J)=MASS(I)*WV1(J)
ISN 0035             SUM=OMEGA(1)*MRM(1,I)+OMEGA(2)*MRM(2,I)+OMEGA(3)*MRM(3,I)
ISN 0036             DO 60 J=1,3
ISN 0037             WV3(J)=SUM*OMEGA(J)
ISN 0038        60   WV4(J)=OMS*MRM(J,I)
ISN 0039             LL=L
ISN 0040             DO 70 J=1,3
ISN 0041             UV(LL)=WV2(J)-WV3(J)+WV4(J)
ISN 0042        70   LL=LL+1
ISN 0043        50   CONTINUE
ISN 0044             DO 74 I=1,NT
```

```
ISN 0045            IP6=I+6
ISN 0046            U(IP6)=0.0
ISN 0047            DO 74 L=1,N3
ISN 0048       74   U(IP6)=U(IP6)+PHI(L,I)*UV(L)
               C
               C  CALCULATE FIRST ORDER DEFORMATION DEPENDENT TERMS
               C
ISN 0049            DO 80 J=1,3
ISN 0050       80   WV2(J)=0.0
ISN 0051            DO 90 I=1,N
ISN 0052            DO 95 J=1,3
ISN 0053       95   WV2(J)=WV2(J)+MASS(I)*QDOT(J,I)
ISN 0054       90   CONTINUE
ISN 0055            CALL CROSS(OMEGA,WV2,WV3)
ISN 0056            SUM=OMEGA(1)*MC1(1)+OMEGA(2)*MC1(2)+OMEGA(3)*MC1(3)
ISN 0057            DO 100 J=1,3
ISN 0058      100   UT(J)=UT(J)-2.0*WV3(J)-SUM*OMEGA(J)+OMS*MC1(J)
ISN 0059            CALL CROSS(MC1,WV1,WV2)
ISN 0060            DO 110 I=1,3
ISN 0061            DO 110 J=1,3
ISN 0062      110   WM1(I,J)=0.0
ISN 0063            DO 120 I=1,N
ISN 0064            DO 125 J=1,3
ISN 0065      125   WM1(J,J)=WM1(J,J)+2.0*MRM(J,I)*Q(J,I)
ISN 0066            WM1(1,2)=WM1(1,2)+MRM(1,I)*Q(2,I)+MRM(2,I)*Q(1,I)
ISN 0067            WM1(1,3)=WM1(1,3)+MRM(1,I)*Q(3,I)+MRM(3,I)*Q(1,I)
ISN 0068            WM1(2,3)=WM1(2,3)+MRM(2,I)*Q(3,I)+MRM(3,I)*Q(2,I)
ISN 0069      120   CONTINUE
ISN 0070            DO 130 I=1,3
ISN 0071            DO 130 J=1,3
ISN 0072            IF(I .LE. J) GO TO 130
ISN 0074            WM1(I,J)=WM1(J,I)
ISN 0075      130   CONTINUE
ISN 0076            DO 140 J=1,3
ISN 0077      140   WV3(J)=WM1(J,1)*OMEGA(1)+WM1(J,2)*OMEGA(2)+WM1(J,3)*OMEGA(3)
ISN 0078            CALL CROSS(OMEGA,WV3,WV4)
ISN 0079            SUM=0.0
ISN 0080            DO 150 I=1,N
ISN 0081      150   SUM=SUM+MRM(1,I)*QDOT(1,I)+MRM(2,I)*QDOT(2,I)+MRM(3,I)*QDOT(3,I)
ISN 0082            DO 160 J=1,3
ISN 0083      160   WV1(J)=SUM*OMEGA(J)
ISN 0084            DO 170 J=1,3
ISN 0085      170   WV3(J)=0.0
ISN 0086            DO 180 I=1,N
ISN 0087            SUM=MRM(1,I)*OMEGA(1)+MRM(2,I)*OMEGA(2)+MRM(3,I)*OMEGA(3)
ISN 0088            DO 190 J=1,3
ISN 0089      190   WV3(J)=WV3(J)+SUM*QDOT(J,I)
ISN 0090      180   CONTINUE
ISN 0091            DO 200 J=1,3
ISN 0092      200   UR(J)=UR(J)+WV2(J)+WV4(J)+2.*(WV3(J)-WV1(J))
ISN 0093            DO 300 I=1,N
ISN 0094            L=3*I-2
ISN 0095            CALL CROSS(OMEGA,QDOT(1,I),WV1)
ISN 0096            DO 310 J=1,3
ISN 0097      310   WV2(J)=MASS(I)*WV1(J)
ISN 0098            SUM=OMEGA(1)*MQ(1,I)+OMEGA(2)*MQ(2,I)+OMEGA(3)*MQ(3,I)
ISN 0099            DO 320 J=1,3
ISN 0100            WV3(J)=SUM*OMEGA(J)
```

```
                                                                                00070100
ISN 0101   320  WV4(J)=OMS*MQ(J,I)                                              00070200
ISN 0102        LL=L                                                            00070300
ISN 0103        DO 330 J=1,3                                                    00070400
ISN 0104        UV(LL)=UV(LL)-2.*WV2(J)-WV3(J)+WV4(J)                           00070500
ISN 0105   330  LL=LL+1                                                         00070600
ISN 0106   300  CONTINUE                                                        00070700
ISN 0107        DO 307 I=1,NT                                                   00070800
ISN 0108        IP6=I+6                                                         00070900
ISN 0109        U(IP6)=0.0                                                      00071000
ISN 0110        DO 307 L=1,N3                                                   00071100
ISN 0111   307  U(IP6)=U(IP6)+PHI(L,I)*UV(L)                                    00071200
           C                                                                    00071300
           C  CALCULATE SECOND ORDER DEFORMATION DEPENDENT TERMS                 00071400
           C                                                                    00071500
ISN 0112        DO 311 I=1,3                                                    00071600
ISN 0113        DO 311 J=1,3                                                    00071700
ISN 0114   311  WM1(I,J)=0.0                                                    00071800
ISN 0115        DO 321 I=1,N                                                    00071900
ISN 0116        DO 331 J=1,3                                                    00072000
ISN 0117   331  WM1(J,J)=WM1(J,J)+MQ(J,I)*Q(J,I)                                00072100
ISN 0118        WM1(1,2)=WM1(1,2)+MQ(1,I)*Q(2,I)                                00072200
ISN 0119        WM1(1,3)=WM1(1,3)+MQ(1,I)*Q(3,I)                                00072300
ISN 0120        WM1(2,3)=WM1(2,3)+MQ(2,I)*Q(3,I)                                00072400
ISN 0121   321  CONTINUE                                                        00072500
ISN 0122        DO 340 I=1,3                                                    00072600
ISN 0123        DO 340 J=1,3                                                    00072700
ISN 0124        IF(I .LE. J) GO TO 340                                          00072800
ISN 0126        WM1(I,J)=WM1(J,I)                                               00072900
ISN 0127   340  CONTINUE                                                        00073000
ISN 0128        DO 350 J=1,3                                                    00073100
ISN 0129   350  WV1(J)=WM1(J,1)*OMEGA(1)+WM1(J,2)*OMEGA(2)+WM1(J,3)*OMEGA(3)    00073200
ISN 0130        CALL CROSS(OMEGA,WV1,WV2)                                       00073300
ISN 0131        SUM=0.0                                                         00073400
ISN 0132        DO 360 I=1,N                                                    00073500
ISN 0133   360  SUM=SUM+MQ(1,I)*QDOT(1,I)+MQ(2,I)*QDOT(2,I)+MQ(3,I)*QDOT(3,I)   00073600
ISN 0134        DO 370 J=1,3                                                    00073700
ISN 0135        WV3(J)=SUM*OMEGA(J)                                             00073800
ISN 0136   370  WV4(J)=0.0                                                      00073900
ISN 0137        DO 380 I=1,N                                                    00074000
ISN 0138        SUM=OMEGA(1)*MQ(1,I)+OMEGA(2)*MQ(2,I)+OMEGA(3)*MQ(3,I)          00074100
ISN 0139        DO 385 J=1,3                                                    00074200
ISN 0140   385  WV4(J)=WV4(J)+SUM*QDOT(J,I)                                     00074300
ISN 0141   380  CONTINUE                                                        00074400
ISN 0142        DO 390 J=1,3                                                    00074500
ISN 0143   390  UR(J)=UR(J)+WV2(J)+2.*(WV4(J)-WV3(J))                           00074600
ISN 0144        RETURN                                                          00074700
ISN 0145        END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       144, PROGRAM SIZE =      8226, SUBPROGRAM NAME =  NLKT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                 244K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.246/12.21.56        PAGE   1

REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

 ISN 0002              SUBROUTINE SOLVE
                 C
                 C   THIS SUBROUTINE ASSEMBLES THE EQUATIONS OF MOTION IN FIRST ORDER
                 C   FORM AND SOLVES THE SET OF SIMULTANEOUS DIFFERENTIAL EQUATIONS
                 C                 (SIZE 6*N+12)
 ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
 ISN 0004              REAL*8 MASS,MCO,MRM,INERT1
 ISN 0005              DIMENSION MASS(50),MCO(3),MRM(3,50),CMAT(3,3),A(156,156),A12(3,3),
                     1 INERT1(3,3),A23(3,150),Q(3,50),QDOT(3,50),UVW(3),OMEGA(3),
                     2 R(3),THETA(3),F(156),U(156),R14(3,3),GAMA(3,3),
                     3 B(156),YDOT(312),Y(312),WV1(3),WV2(3),WV3(156),
                     4 QV(150),QDOTV(150),AV(12246)
 ISN 0006              EQUIVALENCE (QV(1),Q(1,1)),(QDOTV(1),QDOT(1,1))
 ISN 0007              COMMON /CONST/ TM,MASS,MCO,MRM,CMAT,N,N3,N3P6,NT,NTP6,NO
 ISN 0008              COMMON /AMAT/ A,A12,INERT1,A23
 ISN 0009              COMMON /STATE/ R,THETA,Q,UVW,OMEGA,QDOT
 ISN 0010              COMMON /FORCE/ F
 ISN 0011              COMMON /FICFRC/ U
 ISN 0012              COMMON /TIME/ DT,TSTOP,DTP,DTG
 ISN 0013              COMMON /MODES/ PHI(150,150),WS(150)
 ISN 0014              COMMON /MODCO/ ETA(150),ETAD(150)
 ISN 0015              DATA GAMA/8*0.0,1.0/,IPASS/0/
                 C
                 C   CALCULATE R14 -TRANSFORMATION FROM BODY FRAME TO INERTIAL FRAME
                 C
 ISN 0016              S1=DSIN(THETA(1))
 ISN 0017              C1=DCOS(THETA(1))
 ISN 0018              S2=DSIN(THETA(2))
 ISN 0019              C2=DCOS(THETA(2))
 ISN 0020              S3=DSIN(THETA(3))
 ISN 0021              C3=DCOS(THETA(3))
 ISN 0022              R14(1,1)=C2*C3
 ISN 0023              R14(1,2)=-C2*S3
 ISN 0024              R14(1,3)=S2
 ISN 0025              R14(2,1)=C1*S3+S1*S2*C3
 ISN 0026              R14(2,2)=C1*C3-S1*S2*S3
 ISN 0027              R14(2,3)=-S1*C2
 ISN 0028              R14(3,1)=S1*S3-C1*S2*C3
 ISN 0029              R14(3,2)=S1*C3+C1*S2*S3
 ISN 0030              R14(3,3)=C1*C2
                 C
                 C   CALCULATE "GAMA" - TRANSFORMS ANGULAR VELOCITY TO ATTITUDE RATES
                 C
 ISN 0031              GAMA(1,1)=C3/C2
 ISN 0032              GAMA(1,2)=-S3/C2
 ISN 0033              GAMA(2,1)=S3
 ISN 0034              GAMA(2,2)=C3
 ISN 0035              T2=S2/C2
 ISN 0036              GAMA(3,1)=-C3*T2
 ISN 0037              GAMA(3,2)=S3*T2
 ISN 0038              DO 30 I=1,6
 ISN 0039        30    B(I)=F(I)+U(I)
 ISN 0040              DO 40 I=1,NT
```

```
ISN 0041             IP6=I+6
ISN 0042      40     B(IP6)=F(IP6)+U(IP6)-WS(I)*ETA(I)
              C
              C   STORE UPPER TRIANGLE OF "A" IN "AV"
              C
ISN 0043             L=1
ISN 0044             DO 300 J=1,NTP6
ISN 0045             DO 300 I=1,J
ISN 0046             AV(L)=A(I,J)
ISN 0047             L=L+1
ISN 0048      300    CONTINUE
ISN 0049             CALL LEQT1P(AV,1,NTP6,B,156,0,D1,D2,IER)
ISN 0050             IF(IER .EQ. 0) GO TO 60
ISN 0052             WRITE(6,61) IER
ISN 0053             STOP
ISN 0054      61     FORMAT(1H0,5X,'ERROR DETECTED BY IMSL LIBRARY ROUTINE "LEQT1P"
                    1ERROR CODE=',I3)
ISN 0055      60     IF(IPASS .EQ. 1) GO TO 105
ISN 0057             IPASS=1
              C
              C   SET INITIAL VALUE OF "Y"
              C
ISN 0058             DO 70 I=1,3
ISN 0059             IP3=I+3
ISN 0060             Y(I)=R(I)
ISN 0061      70     Y(IP3)=THETA(I)
ISN 0062             DO 80 I=1,NT
ISN 0063      80     Y(I+6)=ETA(I)
ISN 0064             DO 90 I=1,3
ISN 0065             L=NTP6+I
ISN 0066             LL=L+3
ISN 0067             Y(L)=UVW(I)
ISN 0068      90     Y(LL)=OMEGA(I)
ISN 0069             DO 100 I=1,NT
ISN 0070             L=12+NT+I
ISN 0071      100    Y(L)=ETAD(I)
              C
              C   SET UP "YDOT"
              C
ISN 0072      105    DO 110 I=1,3
ISN 0073             WV1(I)=R14(I,1)*UVW(1)+R14(I,2)*UVW(2)+R14(I,3)*UVW(3)
ISN 0074      110    WV2(I)=GAMA(I,1)*OMEGA(1)+GAMA(I,2)*OMEGA(2)+GAMA(I,3)*OMEGA(3)
ISN 0075             DO 120 I=1,3
ISN 0076             YDOT(I)=WV1(I)
ISN 0077      120    YDOT(I+3)=WV2(I)
ISN 0078             DO 130 I=1,NT
ISN 0079      130    YDOT(6+I)=ETAD(I)
ISN 0080             DO 140 I=1,NTP6
ISN 0081      140    YDOT(NTP6+I)=B(I)
              C
              C   UPDATE VARIABLES IN STATE VECTOR
              C
ISN 0082             CALL ODESLV(NO,Y,YDOT,DT)
ISN 0083             DO 150 I=1,3
ISN 0084             R(I)=Y(I)
ISN 0085      150    THETA(I)=Y(I+3)
ISN 0086             DO 160 I=1,NT
ISN 0087      160    ETA(I)=Y(6+I)
```

```
ISN 0088          DO 170 I=1,3                                   00085900
ISN 0089          L=NTP6+I                                       00086000
ISN 0090          LL=L+3                                         00086100
ISN 0091          UVW(I)=Y(L)                                    00086200
ISN 0092    170   OMEGA(I)=Y(LL)                                 00086300
ISN 0093          DO 180 I=1,NT                                  00086400
ISN 0094    180   ETAD(I)=Y(NT+12+I)                             00086500
            C                                                    00086600
            C   COMPUTE NEW VALUES FOR "Q" AND "QDOT"            00086700
            C                                                    00086800
ISN 0095          DO 200 I=1,N3                                  00086900
ISN 0096          QV(I)=0.0                                      00087000
ISN 0097          QDOTV(I)=0.0                                   00087100
ISN 0098          DO 220 L=1,NT                                  00087200
ISN 0099          QV(I)=QV(I)+PHI(I,L)*ETA(L)                    00087300
ISN 0100    220   QDOTV(I)=QDOTV(I)+PHI(I,L)*ETAD(L)             00087400
ISN 0101    200   CONTINUE                                       00087500
ISN 0102          RETURN                                         00087600
ISN 0103          END                                            00087700
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      102, PROGRAM SIZE =    107808, SUBPROGRAM NAME = SOLVE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                          256K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002            SUBROUTINE ODESLV(N,Y,DERIV,H)
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            DIMENSION DERIV(N),Y(N),DERIVO(312),BD1(312,2),BD2(312,2),BD3(312)
ISN 0005            DATA INTF/1/,C1/0.0/,C2/0.0/,C3/0.0/
              C
              C   THIS SUBROUTINE INTEGRATES THE FIRST ORDER SYSTEM OF ORDINARY
              C   DIFFERENTIAL EQUATIONS  "DY/DT=DERIV" BY THE ADAMS METHOD
              C   USING THIRD ORDER DIFFERENCES.
              C     N- SIZE OF SYSTEM
              C     Y- VECTOR OF INITIAL VALUES ON INPUT. "Y" IS OVERWRITTEN
              C        WITH THE NEW SOLUTION
              C     H- STEP SIZE
              C
ISN 0006            IF(N .LE. 312) GO TO 10
ISN 0008            WRITE(6,12) N
ISN 0009            STOP
ISN 0010      12    FORMAT(1H0,5X,'ERROR IN SUBROUTINE **ODESLV** CALLED WITH STATE
                   1SIZE =',I3,'  EXCEEDS DIMENSION SIZE OF ARRAYS')
ISN 0011      10    GO TO(1000,2000,3000,4000),INTF
              C
              C   FIRST CALL TO ROUTINE  - EULER INTEGRATION
              C
ISN 0012      1000  DO 20 I=1,N
ISN 0013      20    DERIVO(I)=DERIV(I)
ISN 0014            INTF=2
ISN 0015            GO TO 5000
              C
              C   SECOND CALL TO ROUTINE - FIRST ORDER DIFFERENCES
              C
ISN 0016      2000  DO 30 I=1,N
ISN 0017            BD1(I,1)=DERIV(I)-DERIVO(I)
ISN 0018            BD1(I,2)=BD1(I,1)
ISN 0019      30    DERIVO(I)=DERIV(I)
ISN 0020            C1=.5
ISN 0021            INTF=3
ISN 0022            GO TO 5000
              C
              C   THIRD CALL TO ROUTINE - SECOND ORDER DIFFERENCES
              C
ISN 0023      3000  DO 40 I=1,N
ISN 0024            BD1(I,2)=DERIV(I)-DERIVO(I)
ISN 0025            BD2(I,1)=BD1(I,2)-BD1(I,1)
ISN 0026            BD2(I,2)=BD2(I,1)
ISN 0027            DERIVO(I)=DERIV(I)
ISN 0028      40    BD1(I,1)=BD1(I,2)
ISN 0029            INTF=4
ISN 0030            C2=5.0/12.0
ISN 0031            GO TO 5000
              C
              C   ADAMS METHOD WITH 3RD ORDER DIFFERENCES.
              C
ISN 0032      4000  DO 50 I=1,N
ISN 0033            BD1(I,2)=DERIV(I)-DERIVO(I)
```

```
ISN 0034            BD2(I,2)=BD1(I,2)-BD1(I,1)                                    00093100
ISN 0035            BD3(I)=BD2(I,2)-BD2(I,1)                                      00093200
ISN 0036            DERIVO(I)=DERIV(I)                                            00093300
ISN 0037            BD1(I,1)=BD1(I,2)                                             00093400
ISN 0038      50    BD2(I,1)=BD2(I,2)                                             00093500
ISN 0039            C3=3.0/8.0                                                    00093600
ISN 0040            GO TO 5000                                                    00093700
              C                                                                   00093800
              C  UPDATE VECTOR 'Y'                                                00093900
              C                                                                   00094000
ISN 0041      5000  DO 60 I=1,N                                                   00094100
ISN 0042      60    Y(I)=Y(I)+H*(DERIV(I)+C1*BD1(I,2)+C2*BD2(I,2)+C3*BD3(I))      00094200
ISN 0043            RETURN                                                        00094300
ISN 0044            END                                                           00094400
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      43, PROGRAM SIZE =     16838, SUBPROGRAM NAME =ODESLV

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                    276K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

  ISN 0002           SUBROUTINE PRINT(T,F0,TAU0,FP)
  ISN 0003           IMPLICIT REAL*8(A-H,O-Z)
  ISN 0004           REAL*8 MASS,MC0,MRM
  ISN 0005           REAL*4 TD(3),OD(3),PLTDAT
  ISN 0006           DIMENSION R(3),THETA(3),Q(3,50),UVW(3),OMEGA(3),QDOT(3,50),
                    1 MC0(3),MRM(3,50),CMAT(3,3),MASS(50),F0(3),TAU0(3),FP(3,50),
                    2 IPLOT(42),PLTDAT(100,20)
  ISN 0007           COMMON /STATE/R,THETA,Q,UVW,OMEGA,QDOT
  ISN 0008           COMMON /CONST/ TM,MASS,MC0,MRM,CMAT,N,N3,N3P6,NT,NTP6,NO
  ISN 0009           COMMON /PLOTT/ PLTDAT,IPLOT,NP
  ISN 0010           WRITE(6,100) T
  ISN 0011           WRITE(6,200) F0,TAU0
  ISN 0012           DO 201 I=1,N
  ISN 0013      201  WRITE(6,202) I,(FP(J,I),J=1,3)
  ISN 0014           DO 10 I=1,3
  ISN 0015           TD(I)=57.29578*THETA(I)
  ISN 0016      10   OD(I)=57.29578*OMEGA(I)
  ISN 0017           WRITE(6,110) R,TD,UVW,OD
  ISN 0018           DO 20 I=1,N
  ISN 0019      20     WRITE(6,120) I,(Q(J,I),J=1,3)
  ISN 0020           DO 30 I=1,N
  ISN 0021      30     WRITE(6,130) I,(QDOT(J,I),J=1,3)
  ISN 0022      100  FORMAT(1H0,///,5X,'TIME=',F10.3,' SEC')
  ISN 0023      110  FORMAT(1H ,8X,'R=',3(2X,1PE11.4),' FT',/,4X,'THETA=',
                    1 3(2X,1PE11.4),' DEG',/,6X,'UVW=',3(2X,1PE11.4),' FT/SEC',/,
                    2 4X,'OMEGA=',3(2X,1PE11.4),' DEG/SEC',/)
  ISN 0024      120  FORMAT(1H ,5X,'Q',I3,'=',3(2X,1PE11.4),' FT')
  ISN 0025      130  FORMAT(1H ,2X,'QDOT',I3,'=',3(2X,1PE11.4),' FT/SEC')
  ISN 0026      200  FORMAT(1H0,8X,'F0=',3(2X,1PE11.4),' LB',/,
                    1 6X,'TAU0=',3(2X,1PE11.4),' FT LB')
  ISN 0027      202  FORMAT(1H ,4X,'F',I3,'=',3(2X,1PE11.4),' LB')
  ISN 0028           RETURN
  ISN 0029           ENTRY GRAF(T)
  ISN 0030           IF(IPLOT(1) .EQ. 0) GO TO 203
                C
                C   STORE VARIABLES FOR PLOTTING
                C
  ISN 0032           NP=NP+1
  ISN 0033           IF(NP .GT. 100) GO TO 203
  ISN 0035           PLTDAT(NP,1)=T
  ISN 0036           DO 300 I=1,42
  ISN 0037           NY=IPLOT(I)
  ISN 0038           IF(NY .EQ. 0) GO TO 203
  ISN 0040           IF(NY .GT. 3) GO TO 310
                C
                C  STORE INERTIAL POSITION
                C
  ISN 0042           PLTDAT(NP,I+1)=R(NY)
  ISN 0043           GO TO 300
  ISN 0044      310  IF(NY .GT. 6) GO TO 320
                C
                C   STORE ATTITUDE ANGLES IN DEGREES
                C
```

```
 ISN 0046            PLTDAT(NP,I+1)=57.29578*THETA(NY-3)
 ISN 0047            GO TO 300
 ISN 0048      320   IF(NY .GT. N3P6) GO TO 330
               C
               C   STORE DEFORMATION COORDINATES
               C
 ISN 0050            L=NY-6
 ISN 0051            I1=1+L/3
 ISN 0052            I2=L-3*(I1-1)
 ISN 0053            IF(I2 .NE. 0) GO TO 321
 ISN 0055            I1=I1-1
 ISN 0056            I2=3
 ISN 0057      321   PLTDAT(NP,I+1)=Q(I2,I1)
 ISN 0058            GO TO 300
 ISN 0059      330   IF(NY .GT. (N3P6+3)) GO TO 340
               C
               C   STORE TRANSLATIONAL VELOCITY
               C
 ISN 0061            PLTDAT(NP,I+1)=UVW(NY-N3P6)
 ISN 0062            GO TO 300
 ISN 0063      340   IF(NY .GT. (N3P6+6)) GO TO 350
               C
               C   STORE ANGULAR VELOCITY IN DEG./SEC.
               C
 ISN 0065            PLTDAT(NP,I+1)=57.29578*OMEGA(NY-(N3+9))
 ISN 0066            GO TO 300
               C
               C   STORE DEFORMATION RATES
               C
 ISN 0067      350   L=NY-(N3+12)
 ISN 0068            I1=1+L/3
 ISN 0069            I2=L-3*(I1-1)
 ISN 0070            IF(I2 .NE. 0) GO TO 351
 ISN 0072            I1=I1-1
 ISN 0073            I2=3
 ISN 0074      351   PLTDAT(NP,I+1)=QDOT(I2,I1)
 ISN 0075      300   CONTINUE
 ISN 0076      203   RETURN
 ISN 0077            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       76, PROGRAM SIZE =      2324, SUBPROGRAM NAME = PRINT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                       268K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED           DATE 82.246/12.22.01        PAGE   1

REQUESTED OPTIONS: NOOBJ,TERM,,NOXREF,,NOMAP,,,NAME(MAIN),AD(NONE),OPT(0),,,FLAG(I),SIZE(384K),LC(60),

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

                                                                                  00103700
  ISN 0002            SUBROUTINE PLOT                                             00103800
  ISN 0003            IMPLICIT REAL*8(A-H,O-Z)                                    00103900
  ISN 0004            REAL*4 RAN,PLTDAT                                           00104000
  ISN 0005            DIMENSION IPLOT(42),PLTDAT(100,20),IT(144),RAN(4),IC(10),   00104100
                     1 IMAG4(5151)                                                00104200
  ISN 0006            COMMON /PLOTT/ PLTDAT,IPLOT,NP                              00104300
  ISN 0007            DATA IT(1)/0/,RAN/4*0./,IC(1)/1H*/                          00104400
                C                                                                 00104500
                C   CALCULATE NUMBER OF VARIABLES TO BE PLOTTED                   00104600
                C                                                                 00104700
  ISN 0008            NV=0                                                        00104800
  ISN 0009            DO 100 I=1,42                                               00104900
  ISN 0010            IF(IPLOT(I) .EQ. 0) GO TO 110                               00105000
  ISN 0012      100   NV=NV+1                                                     00105100
  ISN 0013      110   IF(NV .EQ. 0) RETURN                                        00105200
  ISN 0015            DO 120 IP=1,NV                                              00105300
  ISN 0016            CALL USPLT(PLTDAT(1,1),PLTDAT(1,IP+1),100,NP,1,1,IT,RAN,IC,1,  00105400
                     1 IMAG4,IER)                                                 00105500
  ISN 0017      120   CONTINUE                                                    00105600
  ISN 0018            RETURN                                                      00105700
  ISN 0019            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      18, PROGRAM SIZE =     21736, SUBPROGRAM NAME =  PLOT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      276K BYTES OF CORE NOT USED

*STATISTICS*  NO  DIAGNOSTICS THIS STEP
```

APPENDIX B

SAMPLE INPUT DATA

This appendix provides an illustrative example of the program NAMELIST input data corresponding to the vehicle in Figure 5. The vector geometry and inertia matrix for that vehicle are

$$\vec{S} = -\frac{b}{2}\,\underline{i}_4 + \frac{b}{2}\,\underline{j}_4 + \frac{h}{2}\,\underline{k}_4$$

$$\vec{r}^{1} = -\frac{b}{2}\,\underline{i}_4 + (b + L)\underline{j}_4 + \frac{h}{2}\,\underline{k}_4$$

$$\vec{r}^{2} = -\frac{b}{2}\,\underline{i}_4 + (b + 2L)\underline{j}_4 + \frac{h}{2}\,\underline{k}_4$$

$$\vec{r}^{3} = -\frac{b}{2}\,\underline{i}_4 - L\,\underline{j}_4 + \frac{h}{2}\,\underline{k}_4$$

$$\vec{r}^{4} = -\frac{b}{2}\,\underline{i}_4 - 2L\,\underline{j}_4 + \frac{h}{2}\,\underline{k}_4$$

$$[I_b] = \frac{m_b}{12}\begin{bmatrix} (b^2 + h^2) & 0 & 0 \\ 0 & (b^2 + h^2) & 0 \\ 0 & 0 & 2b^2 \end{bmatrix}$$

$$\vec{R} = R_x\,\underline{i}_1 + R_y\,\underline{j}_1 + R_z\,\underline{k}_1$$

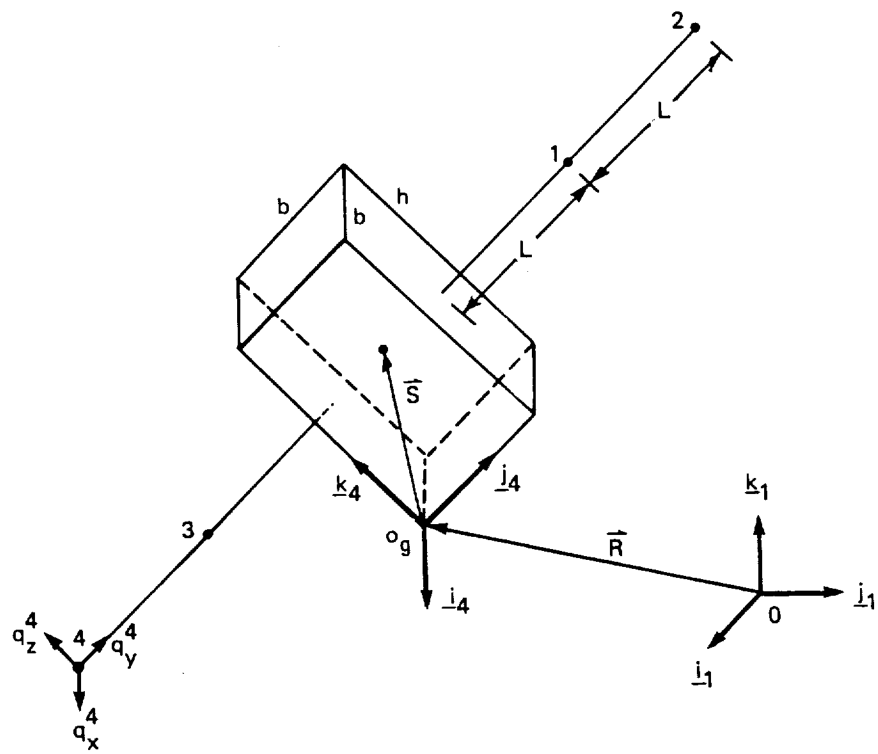


Figure 5. Example vehicle.

The namelist input items given below correspond to the following dimensions and masses

$$m_b = 5 \text{ slugs}$$

$$m_1 = m_3 = 1 \text{ slug}$$

$$m_2 = m_4 = 0.5 \text{ slug}$$

$$b = 1 \text{ ft}$$

$$h = 2 \text{ ft}$$

$$L = 10 \text{ ft}$$

Also two constrained modes are to be used in the simulation. The initial conditions on the kinematic variables are

@ t_0	$R_x = 1 \cdot 10^3$ ft	$\theta_1 = 5$ degrees
	$R_y = 2 \cdot 10^3$ ft	$\theta_2 = 20$ degrees
	$R_z = 3 \cdot 10^3$ ft	$\theta_3 = 0$ degrees
	$u = 0$ ft/s	$\omega_1 = 0$ deg/s
	$v = 5$ ft/s	$\omega_2 = 10$ deg/s
	$w = 0$ ft/s	$\omega_3 = 0$ deg/s

Note that the program in Appendix A sets the initial particle deflections, modal coordinates and the respective time derivatives to zero (see subroutine INITL).

The numerical integration is to proceed from time = 0 (set internally, see main program) to a final time of 60 seconds using an integration time step of 0.01 second. The print time step is to be 6 seconds and the plot time step 0.6 second.

The following variables are to be plotted versus time: R_y, θ_2, q_x^4, q_y^4, q_z^4, v, ω_2.

NAMELIST Input Data

```
&INPUT MØ = 5.0, N = 4, MASS = 1.0, 0.5, 1.0, 0.5,
RM = -0.5, 11.0, 1.0, -0.5, 21.0, 1.0, -0.5, -10.0, 1.0, -0.5,
     -20.0, 1.0,
IØ = 2.083, 0.0, 0.0, 0.0, 2.083, 0.0, 0.0, 0.0, 0.833,
S = -0.5, 0.5, 1.0, NT = 2 &END
&KIN R = 1.E3, 2.E3, 3.E3, THETA = 5.0, 20.0, 0.0,
UVW = 0.0, 5.0, 0.0, OMEGA = 0.0, 10.0, 0.0 &END
&RUN DT = 0.01, TSTOP = 60.0, DTP = 6.0, DTG = 0.6 &END
&PLT IPLOT = 2, 5, 16, 17, 18, 20, 23 &END
```

LIST OF REFERENCES

1. Hughes, P.C., "A Model for the Attitude Dynamics of CTS with Reference to Attitude Control System Design," Aerospace Engineering and Research Consultants Ltd., Downsview, Ont., AERCOL Report No. 75-14-4, 1975.

2. Likins, P.W., "Analytical Dynamics and Nonrigid Spacecraft Simulation," Technical Report 32-1593, Jet Propulsion Laboratory, Pasadena, CA 1974.

3. Bodley, C.S., A.D. Devers, A.C. Park, and H.P. Frisch, "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structures (DISCOS)," Volume I, NASA Technical Paper 1219, 1978.

4. Gates, S.S., "DISCOS Method for Incorporation of Finite-Element Model," Draper Intralab Memo DYN-82-1, January 1982.

5. Likins, P.W., "Dynamics and Control of Flexible Space Vehicles," Technical Report 32-1329, Revision 1, Jet Propulsion Laboratory, Pasadena, CA, 1970.

6. Likins, P.W., "Finite Element Appendage Equations for Hybrid Coordinate Dynamic Analysis," International Journal of Solids and Structures, Volume 8, pp. 709-731, 1972.

7. Storch, J., "Dynamic Equations for Arbitrary Motion of a Flexible Body - Finite Element Idealization," Draper Intralab Memo DYN-82-4, January 1982.

8. IMSL Library Reference Manual, Edition 8, International Mathematical and Statistical Libraries, Inc., Houston, TX, 1980.