EQUATIONS OF MOTION FOR A FLEXIBLE SPACECRAFT — LUMPED PARAMETER IDEALIZATION

by
Joel Storch
Stephen Gates

September 1982
1. Title of document: Equations of Motion for a Flexible Spacecraft - Lumped Parameter Idealizations

3. Work generated under Lab account No.: 92521

4. Classification: Unclassified

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

6. DOCUMENT TYPE

<table>
<thead>
<tr>
<th>Series</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Report (interim/final)</td>
<td>R</td>
</tr>
<tr>
<td>Specification</td>
<td>S</td>
</tr>
<tr>
<td>Technical Memorandum</td>
<td>C</td>
</tr>
<tr>
<td>Thesis</td>
<td>T</td>
</tr>
<tr>
<td>Technical Paper, slides, etc.</td>
<td>P</td>
</tr>
</tbody>
</table>

7. DISTRIBUTION

<table>
<thead>
<tr>
<th>Program</th>
<th>CSDL</th>
<th>Author(s)</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td></td>
<td>Author(s)</td>
<td>2</td>
</tr>
<tr>
<td>TIC/Library</td>
<td></td>
<td>Other external</td>
<td>3</td>
</tr>
</tbody>
</table>

8. PAPERS & PRESENTATIONS—public dissemination:

<table>
<thead>
<tr>
<th>Abstract, for submission/acceptance or oral presentation, requires a separate authorization form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript, for external publishing</td>
</tr>
<tr>
<td>Slides, vugraphs, etc. not included in a paper</td>
</tr>
<tr>
<td>Copies to be released at seminar/symposium</td>
</tr>
<tr>
<td>Oral presentation only</td>
</tr>
</tbody>
</table>

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 [ ]

11. Charge this effort to Lab account No.: 92521

AUTHORIZATION TO PREPARE - document described above

Signatures
Author's Supervisor [ ]
Program Manager [ ]
Chief Scientist [ ]

APPROVAL TO PUBLISH - camera-ready copy/manuscript

TCO/Publications: CSDL style guide and/or sponsor specs met? YES [ X ]

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

camera-ready copy: Joel Storch [ ]

technical approval: Program Manager/Technical Supervisor [ ]

overall approval: Department Head - Sponsor Interface [ ]

Department Head - Education IR&D [ ]

IF nonproject funded [ ]

Remarks: 

DOCUMENT CONTROL:

received copies, from signed [ ]

Distribution: Program copies, on Internal copies, on External copies, on
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.
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, 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 pre-processed 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.
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 \(\mathbf{R}\) 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 \(\mathbf{r}_i\) relative to \(0_g\) in the undeformed state. The elastic displacement of \(m_i\) is \(\mathbf{q}_i\), 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: $f^o(t)$, $\tau^o(t)$. In the domain of the elastic appendage we have an external force $f^i(t)$ acting upon the point mass $m_i$.

As an example, in the case of the Shuttle Remote Manipulator Arm, $f^o$ and $\tau^o$ 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

$$q = (q_1^x q_1^y q_1^z q_2^x q_2^y q_2^z \ldots q_n^x q_n^y q_n^z)^T$$

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

If $[K]$ is the stiffness matrix, then $-[K]q$ 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 $q$. Note that in generating $[K]$ the appendage is cantilevered to the rigid base.

![Diagram of a flexible appendage and a rigid body](image)

**Figure 1.** Idealized vehicle.
CHAPTER 3

EQUATIONS OF MOTION

3.1 Particle Translational Equations

\( \ddot{\mathbf{v}}^i \), the inertial velocity of the \( i^{th} \) particle, is given by

\[
\ddot{\mathbf{v}}^i = \frac{d}{dt} (\dot{\mathbf{v}}^i + \dot{\mathbf{r}}^i + \dot{\mathbf{q}}^i)
\]

Let \( \mathbf{u} \) and \( \mathbf{w} \) represent the (absolute) velocity and angular velocity of the body frame resolved in body axes, we then have

\[
\mathbf{v}^i = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{pmatrix} + \mathbf{w} \times (\mathbf{r}^i + \mathbf{q}^i) + \dot{\mathbf{q}}^i
\]

Differentiating this expression we arrive at the particle acceleration

\[
\frac{d}{dt} \mathbf{v}^i = \begin{pmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{pmatrix} - (\mathbf{r}^i + \mathbf{q}^i) \times \dot{\mathbf{w}} + \ddot{\mathbf{q}}^i + \mathbf{w} \times \left[ \begin{pmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{pmatrix} + 2\dot{\mathbf{q}}^i \right] + \mathbf{w} \times [\mathbf{w} \times (\mathbf{r}^i + \mathbf{q}^i)]
\]
Expanding the cross product in the last term and using the matrix-vector form for the cross product $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]^\wedge \mathbf{b}$ where $[\mathbf{a}]^\wedge = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$

the particle acceleration can be written as

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

(3-1)

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

$$f^i + f_e^i = m_i \frac{d}{dt} \mathbf{v}^i$$

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

Partitioning the stiffness matrix into (3×3) arrays

$$\begin{pmatrix} [K_{11}] & [K_{12}] & \cdots & [K_{1n}] \\ [K_{21}] & [K_{22}] & \cdots & [K_{2n}] \\ \vdots & \vdots & \ddots & \vdots \\ [K_{n1}] & [K_{n2}] & \cdots & [K_{nn}] \end{pmatrix}$$

$$f_e^i = - \sum_{j=1}^{n} [K_{ij}] q^j$$

Employing Eq. (3-1) for the particle acceleration, the translation equations for the appendage particles may be assembled as
\[
\begin{bmatrix}
m_1 \\
m_2 \\
\vdots \\
m_n \\
\end{bmatrix} \left( \begin{bmatrix}
\dot{u}_1 \\
\dot{v}_1 \\
\vdots \\
\dot{v}_n \\
\end{bmatrix} \right) - \begin{bmatrix}
m_1 (\dot{x}_1^2 + \dot{q}_1^2) \\
m_2 (\dot{x}_2^2 + \dot{q}_2^2) \\
\vdots \\
m_n (\dot{x}_n^2 + \dot{q}_n^2) \\
\end{bmatrix} \cdot \omega = \begin{bmatrix}
m_1 & 0 & 0 & \cdots & 0 \\
0 & m_2 & 0 & \cdots & 0 \\
0 & 0 & m_3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & m_n \\
\end{bmatrix} \cdot \dot{q} + [K]q = \begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\vdots \\
\dot{x}_n \\
\end{bmatrix} + \omega_v (3-2)
\]

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,...,n)

and the nonlinear kinematic term \( \omega_v \) is given by

\[
\omega_v = - \begin{bmatrix}
m_1 \dot{\omega}_1 \\
m_2 \dot{\omega}_2 \\
\vdots \\
m_n \dot{\omega}_n \\
\end{bmatrix} \cdot \begin{bmatrix}
\dot{u} \\
\dot{v} \\
\vdots \\
\end{bmatrix} - 2 \begin{bmatrix}
m_1 \dot{\omega}_1 \cdot \dot{q}_1^1 \\
m_2 \dot{\omega}_2 \cdot \dot{q}_2^2 \\
\vdots \\
m_n \dot{\omega}_n \cdot \dot{q}_n^n \\
\end{bmatrix} - \begin{bmatrix}
m_1 \omega \cdot (\dot{x}_1^1 + \dot{q}_1^1) \omega_1 \\
m_2 \omega \cdot (\dot{x}_2^2 + \dot{q}_2^2) \omega_2 \\
\vdots \\
m_n \omega \cdot (\dot{x}_n^n + \dot{q}_n^n) \omega_n \\
\end{bmatrix}
\]

\[
+ \left| \omega \right|^2 \begin{bmatrix}
m_1 (\dot{x}_1^1 + \dot{q}_1^1) \\
m_2 (\dot{x}_2^2 + \dot{q}_2^2) \\
\vdots \\
m_n (\dot{x}_n^n + \dot{q}_n^n) \\
\end{bmatrix} (3-3)
\]

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 \( \mathbf{s} \) is the vector from \( \mathbf{O}_g \) to the mass center of the rigid body (expressed in the body frame)

\[
mc = \sum_{i=1}^{n} m_i (\mathbf{r}_i + \mathbf{q}_i) + m_b \mathbf{s} \tag{3-4}
\]

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

\( \mathbf{c}(t) \) is the vector position of the instantaneous mass center relative to \( \mathbf{O}_g \).

The acceleration of the mass center is:

\[
\frac{d^2}{dt^2}(\mathbf{R} + \mathbf{c}) = \frac{d^2}{dt^2} \mathbf{R} + \begin{pmatrix} \omega \\ \omega \times \mathbf{v} \end{pmatrix} \tag{in body frame}
\]

\[
m \frac{d^2}{dt^2} \mathbf{c} = \sum_{i=1}^{n} m_i \mathbf{q}_i^i - mc \times \mathbf{\omega} + 2m \times \sum_{i=1}^{n} \mathbf{q}_i^i + \mathbf{\omega} \times (\mathbf{\omega} \times mc)
\]

Expressing this last term as: \( (\mathbf{\omega} \times mc) \mathbf{\omega} - ||\mathbf{\omega}||^2 mc \) the vehicle translational equation assumes the form

\[
\begin{bmatrix}
m & 0 & 0 \\
0 & m & 0 \\
0 & 0 & m
\end{bmatrix} \begin{pmatrix} & \dot{u} \\
& \dot{v} \\
& \dot{w}
\end{pmatrix} - [mc] \mathbf{\omega} + [m^1 \ m^2 \ \ldots \ m^n] \mathbf{\omega} = \sum_{i=0}^{n} \mathbf{r}_i^i + \mathbf{u}_t \tag{3-5}
\]
The nonlinear term $u_t$ is given by

$$ u_t = -m_\omega \times \left( \begin{pmatrix} u \\ v \\ w \end{pmatrix} \right) - 2\omega \times \sum_{i=1}^{n} m_i \dot{\mathbf{r}}^i - (\omega \cdot \mathbf{m}) \omega + ||\omega||^2 m_\omega $$

(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 $O_g$.

$$ [I_b] = \iiint (\lambda \cdot \lambda E - \lambda \Lambda^T) \, \text{d}m $$

$\lambda$ is the position vector of a mass element $\text{d}m$ 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:

$H_b$ - angular momentum of rigid base

$$ \sum_{i=1}^{n} H_i $$ - angular momentum of appendage particles

Let $\lambda_i$ and $\dot{\lambda}_i$ denote the position vectors from the system mass center to $m_i$ and a generic mass element in the rigid body, respectively.

$$ H_b = \iiint \dot{\lambda} \times \frac{d}{dt} \lambda \, dm $$

$$ H_i = \dot{\lambda}_i \times m_i \frac{d}{dt} \lambda_i $$
From Figure 2, \[ \mathbf{x}^i = \mathbf{r}^i + \mathbf{q}^i - \mathbf{c}, \mathbf{h} = \mathbf{s} + \mathbf{\lambda} - \mathbf{c}. \]

Inserting this expression for \( \mathbf{h} \) into the integral definition of \( H_B \) and recalling the definition of \([I_B]\) and the fact that \( \int \int \int \lambda \, dm = 0 \), we arrive at the following expression for \( H_B \)

\[
H_B = [I_B] \mathbf{w} + m_B (\mathbf{s} - \mathbf{c}) \times \frac{d}{dt} (\mathbf{s} - \mathbf{c})
\]

Thus

\[
\frac{d}{dt} H_B = [I_B] \mathbf{\dot{w}} + \mathbf{w} \times [I_B] \mathbf{\omega} + m_B (\mathbf{s} - \mathbf{c}) \times [\mathbf{\dot{w}} \times \mathbf{s} + \mathbf{\omega} \times (\mathbf{\dot{w}} \times \mathbf{s})]
\]

\[
- m_B (\mathbf{s} - \mathbf{c}) \times \frac{d^2}{dt^2} \mathbf{c}
\]
Turning to the angular momentum of the $i^{\text{th}}$ particle

$$\frac{d}{dt} \mathbf{H}^i = \mathbf{L}^i \times m_i \frac{d^2 \mathbf{L}^i}{dt^2}$$

Now

$$\frac{d^2 \mathbf{L}^i}{dt^2} = - (\mathbf{x}^i + \mathbf{q}^i) \times \dot{\omega} \mathbf{L}^i + \omega \mathbf{L}^i + 2\omega \times \mathbf{L}^i + [\omega \cdot (\mathbf{x}^i + \mathbf{q}^i)] \omega$$

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

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

$$\frac{d}{dt} \mathbf{H} = ([I_b] - m_b (\mathbf{e} - \mathbf{c}) \cdot \mathbf{s}) \dot{\omega} + \omega \times [I_b] \omega + \sum_{i=1}^n m_i [L^i] \cdot \dot{\omega}$$

$$+ m_b (\mathbf{e} - \mathbf{c}) \times [\omega \times (\mathbf{e} \times \mathbf{s})] - \sum_{i=1}^n m_i [L^i] \times (\mathbf{x}^i + \mathbf{q}^i) \cdot \dot{\omega}$$

$$+ 2 \sum_{i=1}^n m_i [L^i] \cdot \omega \mathbf{L}^i + \sum_{i=1}^n [\omega \cdot (\mathbf{x}^i + \mathbf{q}^i)] m_i \mathbf{L}^i \times \omega$$

$$- ||\omega||^2 \sum_{i=1}^n m_i \mathbf{L}^i \times (\mathbf{x}^i + \mathbf{q}^i)$$

(3-7)
The system rotational motion equation is

\[
\frac{d}{dt} \mathbf{H} = \mathbf{i}^0 - \mathbf{c} \times \mathbf{f}^0 + \sum_{i=1}^{n} (\mathbf{r}^i + \mathbf{g}^i - \mathbf{c}) \times \mathbf{f}^i
\]

\[
= \mathbf{i}^0 - \mathbf{c} \times \sum_{j=0}^{n} \mathbf{f}^j + \sum_{i=1}^{n} (\mathbf{r}^i + \mathbf{g}^i) \times \mathbf{f}^i
\]

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 \((u,v,w)\) explicitly. Since the translational equations (3-5) depend upon \(\dot{\omega}\) 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 \((u,v,w)\) by the following device.

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

\[
\frac{d}{dt} \mathbf{H} = \mathbf{i}^0 + \sum_{i=1}^{n} (\mathbf{r}^i + \mathbf{g}^i) \times \mathbf{f}^i - [mc]^- (\begin{pmatrix} u \\ v \\ w \end{pmatrix}) + [c]^- [mc]^- \dot{\omega}
\]

\[
- \mathbf{c} \times \sum_{i=1}^{n} m_i \mathbf{g}^i - [mc]^- \omega \begin{pmatrix} u \\ v \\ w \end{pmatrix} - 2[c]^- \omega \sum_{i=1}^{n} m_i \mathbf{g}^i
\]

\[
- (\omega \times mc) \omega \times \omega
\]

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

14
\[
\begin{align*}
\begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
&= \mathbf{[mc]}^{-1}
\begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
+ [\mathbf{I(t)}] \mathbf{\dot{\omega}}
+ \sum_{i=1}^{n} m_i (\mathbf{r}^i + \mathbf{q}^i) \mathbf{\dot{r}}^i + \sum_{i=1}^{n} m_i (\mathbf{r}^i + \mathbf{q}^i) \mathbf{\dot{q}}^i - [\mathbf{\bar{c}}] \mathbf{[mc]}^{-1} \\
&= \mathbf{I}^0 + \sum_{i=1}^{n} (\mathbf{r}^i + \mathbf{q}^i) \times \mathbf{\dot{r}}^i + \mathbf{u}_r
\end{align*}
\] (3-9)

Here

\[
[I(t)] = [I_b] - m_b (\mathbf{s} \cdot \mathbf{c}) \mathbf{\dot{s}} - \sum_{i=1}^{n} m_i (\mathbf{r}^i + \mathbf{q}^i) - [\mathbf{\bar{c}}] \mathbf{[mc]}^{-1}
\]

which can be simplified to

\[
[I(t)] = [I_b] - m_b (\mathbf{s} \cdot \mathbf{c}) \mathbf{\dot{s}} + \sum_{i=1}^{n} m_i (\mathbf{r}^i + \mathbf{q}^i)^2
\] (3-10)

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

The nonlinear rotation term \(\mathbf{u}_r\) is given by

\[
\begin{align*}
\mathbf{u}_r &= -(\mathbf{mc})^{-1} \begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
- 2 [\mathbf{c}] (\mathbf{\omega}) \begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
- \sum_{i=1}^{n} m_i \mathbf{\dot{r}}^i - (\mathbf{\omega} \cdot \mathbf{mc}) [\mathbf{c}] (\mathbf{\omega}) \\
&= \begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
- \begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
- \begin{pmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{w}
\end{pmatrix}
- \sum_{i=1}^{n} m_i \mathbf{\dot{r}}^i - (\mathbf{\omega} \cdot \mathbf{mc}) [\mathbf{c}] (\mathbf{\omega}) \\
&= \sum_{i=1}^{n} \mathbf{[\mathbf{\omega} \cdot (\mathbf{r}^i + \mathbf{q}^i)]} m_i \mathbf{\dot{r}}^i \times \mathbf{\omega} + |\mathbf{\omega}|^2 \sum_{i=1}^{n} m_i \mathbf{\dot{r}}^i \times (\mathbf{r}^i + \mathbf{q}^i)
\end{align*}
\] (3-11)
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 $q$.
2. Terms first order in $q$.
3. Terms second order in $q$.

The majority of coefficients are directly identifiable in this hierarchy. We have two coefficients which require additional attention: $m_c$ and $[I(t)]$.

From Eq. (3-4), $m_c = m_d s + \sum_{i=1}^{n} m_i \dot{x}_i + \sum_{i=1}^{n} m_i q_i(t)$. The first two terms are of category 1 and the sum will be denoted by $m_{c0}$. The time-dependent term will be denoted by $m_{c1}(t)$.

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

\[
[I_1] = [I_b] - m_b [\ddot{s}]^2 - \sum_{i=1}^{n} m_i [\ddot{z}_i]_t^2
\]

\[
[I_2(t)] = -\sum_{i=1}^{n} m_i ([\ddot{x}_i] [\ddot{z}_i] + [\ddot{z}_i] [\ddot{z}_i])
\]

\[
[I_3(t)] = -\sum_{i=1}^{n} m_i [\ddot{z}_i]^2
\]

4.2 Nonlinear Kinematic Terms

In this section, we concentrate upon the three nonlinear terms: \(U_t\), \(u_v\), and \(\omega\) 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 \(q, \dot{q}\)
2. Nonlinear terms first order in \(q, \dot{q}\)
3. Nonlinear terms second order in \(q, \dot{q}\)

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

\[
u^{(1)}_t = -m[\omega]^2 \left( \begin{array}{c} u \\ v \\ \gamma \end{array} \right) - (\omega \cdot m_{c_0}) \omega + ||\omega||^2 m_{c_0}
(4-1)
\]

\[
u^{(2)}_t = -2[\omega] \sum_{i=1}^{n} m_i \dddot{q}_i - (\omega \cdot m_{c_1}) \omega + ||\omega||^2 m_{c_1}
(4-2)
\]
In a similar manner from Eq. (3-3)

\[ u_v = u_v^{(1)} + u_v^{(2)} \]

\[
\begin{pmatrix}
\begin{bmatrix}
\frac{m_1 \omega}{r_1}
\frac{m_2 \omega}{r_2}
\vdots
\frac{m_n \omega}{r_n}
\end{bmatrix}
\end{pmatrix} - \begin{pmatrix}
\begin{bmatrix}
\frac{m_1 \omega \cdot q_1}{r_1}
\frac{m_2 \omega \cdot q_2}{r_2}
\vdots
\frac{m_n \omega \cdot q_n}{r_n}
\end{bmatrix}
\end{pmatrix} + ||\omega||^2
\]

\[ (4-3) \]

\[
\begin{pmatrix}
\begin{bmatrix}
\frac{m_1 \omega}{r_1}
\frac{m_2 \omega}{r_2}
\vdots
\frac{m_n \omega}{r_n}
\end{bmatrix}
\end{pmatrix} - \begin{pmatrix}
\begin{bmatrix}
\frac{m_1 \omega \cdot q_1}{r_1}
\frac{m_2 \omega \cdot q_2}{r_2}
\vdots
\frac{m_n \omega \cdot q_n}{r_n}
\end{bmatrix}
\end{pmatrix} + ||\omega||^2
\]

\[ (4-4) \]

Expansion of \( u_x \)

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

\[
-2 \sum_{i=1}^{n} m_i (r_i^x + q_i^x) (\omega)^{-\dot{q}^i} = -2 \sum_{i=1}^{n} m_i (r_i^x + q_i^x) (\omega)^{-\dot{q}^i}
\]

\[
= -2 \sum_{i=1}^{n} m_i [r_i^x]^{-\dot{q}^i} - 2 \sum_{i=1}^{n} m_i [q_i^x]^{-\dot{q}^i}
\]
The third term in $u$ can be expressed as

$$(w \cdot mc) \cdot w = (w \cdot mc_0) \cdot w + (w \cdot mc_1) \cdot w + (w \cdot mc_0) \cdot w$$

For the last two terms in $u$ the following expansions are useful

$$
[w \cdot (r^i + q^i)] \cdot m^i \cdot x \cdot w = (w \cdot r^i) \cdot m^i \cdot (r^i - c_0) \cdot w
\nonumber
$$

$$
+ (w \cdot q^i) \cdot m^i \cdot (q^i - c_0) \cdot w
\nonumber
$$

$$
+ (w \cdot r^i) \cdot m^i \cdot (q^i - c_1) \cdot w
\nonumber
$$

$$
+ (w \cdot q^i) \cdot m^i \cdot (q^i - c_1) \cdot w
\nonumber
$$

Collecting terms in $u$ independent of deformation

$u^{(1)}_r = -[mc_0]^w (u) - (w \cdot mc_0) \cdot w - w \cdot [I_b]^w
\nonumber
$$

$$
- m_b (s - c_0) \cdot [w \cdot (w \cdot s)] - \sum_{i=1}^n m^i (w \cdot r^i) \cdot (r^i - c_0) \cdot w
\nonumber
$$

$$
+ ||w||^2 \sum_{i=1}^n m^i \cdot r^i \cdot c_0
\nonumber
$$

The expression for $u^{(1)}_r$ can be further simplified by use of the following identities
\[ \sum_{i=1}^{n} m_i \mathbf{r}^i \times \mathbf{c}_0 = -m_b \mathbf{s} \times \mathbf{c}_0 \]

\[ \sum_{i=1}^{n} m_i (\mathbf{w} \cdot \mathbf{r}^i) (\mathbf{r}^i - \mathbf{c}_0) \times \mathbf{w} = -[\mathbf{w}] - \sum_{i=1}^{n} m_i \mathbf{r}^i \mathbf{r}^i \mathbf{w} \]

\[ + [\mathbf{w} \cdot (m_b \mathbf{s} - m_c \mathbf{c}_0)] \mathbf{c}_0 \times \mathbf{w} \]

\[-m_b (\mathbf{s} - \mathbf{c}_0) \times [\mathbf{w} \times (\mathbf{w} \times \mathbf{s})] = - (\mathbf{w} \cdot \mathbf{s}) m_b \mathbf{s} \times \mathbf{w} + (\mathbf{w} \cdot \mathbf{s}) m_b \mathbf{c}_0 \times \mathbf{w} \]

\[ - \|\mathbf{w}\|^2 m_b \mathbf{c}_0 \times \mathbf{s} \]

Incorporating these results we arrive at the final expression

\[ u^{(1)}_r = -[m_c \mathbf{c}_0]^T [\mathbf{w}] \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + [\mathbf{w}] \sum_{i=1}^{n} m_i \mathbf{r}^i \mathbf{r}^i \mathbf{w} - \mathbf{w} \times [I_b] \mathbf{w} \]

\[ - m_b (\mathbf{s} \cdot \mathbf{w}) \mathbf{s} \times \mathbf{w} \]

(4-5)

Collecting first order deformation dependent terms in \( u_r \)

\[ u^{(2)}_r = -[m_c \mathbf{c}_0]^T [\mathbf{w}] \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - 2 \sum_{i=1}^{n} m_i \mathbf{r}^i \mathbf{r}^i \mathbf{w} - (\mathbf{w} \cdot m_c \mathbf{c}_0) [c_1]^T \mathbf{w} \]

\[ - (\mathbf{w} \cdot m_c \mathbf{c}_0) \mathbf{c}_0 \times \mathbf{w} + m_b \mathbf{c}_0 \times [\mathbf{w} \times (\mathbf{w} \times \mathbf{s})] \]

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

\[ + \|\mathbf{w}\|^2 \sum_{i=1}^{n} m_i (\mathbf{r}^i \times \mathbf{c}_1 + \mathbf{q}^i \times \mathbf{c}_0) \]
The expression for $u_r^{(2)}$ can be further simplified by use of the following identities

$$
\sum_{i=1}^{n} m_i (r_i \times c_i + q_i \times c_0) = m_b c_1 \times s
$$

$$
- \sum_{i=1}^{n} \left( (\omega \cdot q_i) m_i (r_i - c_0) \times \omega + (\omega \cdot r_i) m_i (q_i - c_1) \times \omega \right) =
$$

$$
[\omega] \sum_{i=1}^{n} m_i (r_i \hat{q}^{iT} + q_i \hat{r}^{iT}) \omega + (\omega \cdot m_{e1}) c_0 \times \omega + [\omega] (m_{c0} - m_{c2}) c_1 \times \omega
$$

$$
m_b c_1 \times [\omega \times (\omega \times s)] = m_b (\omega \cdot s) c_1 \times \omega - m_b ||\omega||^2 c_1 \times s
$$

$$
u_r^{(2)} = -[m_{e1}]^{-1} [\omega]^{-1} \begin{vmatrix} u \\ v \\ w \end{vmatrix} - 2 \sum_{i=1}^{n} m_i [r_i]^{-1} [\omega]^{-1} q_i
$$

$$
+ [\omega]^{-1} \sum_{i=1}^{n} m_i (r_i \hat{q}^{iT} + q_i \hat{r}^{iT}) \omega
$$

(4-6)

Collecting second order deformation dependent terms in $u_r$ and simplifying

$$
u_r^{(3)} = -2 \sum_{i=1}^{n} m_i [q_i]^{-1} [\omega]^{-1} q_i + [\omega]^{-1} \sum_{i=1}^{n} m_i q_i \hat{q}^{iT} \omega
$$

(4-7)

$u_r = u_r^{(1)} + u_r^{(2)} + u_r^{(3)}$ 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, \( \omega = 0 \) and \( (\ddot{u}, \ddot{v}, \ddot{w}) = 0 \). The "constrained" appendage equation then assumes the familiar form

\[
[M]\dddot{\mathbf{q}} + [K]\mathbf{q} = \mathbf{0}
\]

where \([M] = \text{diag}(m^1, m^2, ..., m^n)\)

The natural frequencies, \( \omega_i \), and corresponding mode shapes, \( \mathbf{v}_i \), are determined from

\[
([K] - \omega^2[M])\mathbf{v} = \mathbf{0}
\]  \hspace{1cm} (5-1)

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 \( \mathbf{v}_i \) corresponding to positive eigenvalues \( \omega_i^2 \), even if there are multiple eigenvalues.
We assume that the eigenvectors are normalized such that \((v^i, [M]v^i) = 1\). It follows that \((v^i, [K]v^i) = \omega_i^2\). We can always create a mutually orthogonal set such that

\[
(v^i, [M]v^j) = 0 = (v^i, [K]v^j) \quad (i \neq j)
\]

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: \(W = [M]^{-1/2}v\). 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

\[
[K][M]^{-1/2}W = \omega^2 [M]^{1/2}W \quad \text{or} \quad [\tilde{\gamma}]W = \omega^2 W
\]

(5-2)

where \([\tilde{\gamma}]\) is the symmetric matrix: \([\tilde{\gamma}] = [M]^{-1/2}[K][M]^{-1/2}\)

It is easily verified that if the eigenvectors \(W^i\) \((i = 1, 2, \ldots, 3n)\) 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 \(v^1, v^2, \ldots, v^t\) \((t \leq 3n)\). We now make the transformation

\[
q = [\Phi]n
\]

(5-3)

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), pre-multiplying by $[\Phi]^T$ and recalling the orthogonality and normality conditions we arrive at

$$
\begin{bmatrix}
\dot{u}\\
\dot{v}\\
\dot{w}
\end{bmatrix} - [\Phi]^T \begin{bmatrix}
m_1 (\dot{x}^1 + \dot{q}^1) \\
m_2 (\dot{x}^2 + \dot{q}^2) \\
m_n (\dot{x}^n + \dot{q}^n)
\end{bmatrix} \cdot \omega + \bar{n} + [\Phi]^T \begin{bmatrix}
\omega_1 \\
\omega_2 \\
\cdot \\
0 \\
0 \\
\cdot
\end{bmatrix} n =
$$

$$
\begin{bmatrix}
m_1 \\
m_2 \\
m_n
\end{bmatrix}^T + [\Phi]^T u_v
$$

The vehicle translational equations (3-5) become

$$
\begin{bmatrix}
m & 0 & 0 \\
0 & m & 0 \\
0 & 0 & m
\end{bmatrix} \begin{bmatrix}
\ddot{u} \\
\ddot{v} \\
\ddot{w}
\end{bmatrix} - [mc]^{-1} \omega + [m^1 m^2 \ldots m^n] [\Phi] \ddot{n} = \sum_{i=0}^{n} \ddot{x}_i + u_t
$$

The vehicle rotational equations (3-9) become

$$
[mc]^{-1} \begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{w}
\end{bmatrix} + [I(t)] \dot{\omega} + \begin{bmatrix}
m_1 (\dot{r}^1 + \dot{q}^1) \\
m_2 (\dot{r}^2 + \dot{q}^2) \\
\cdot \\
m_n (\dot{r}^n + \dot{q}^n)
\end{bmatrix} [\Phi] \ddot{n} =
$$

$$
\begin{bmatrix}
0 \\
\cdot \\
\cdot \\
\ddot{x}_1 \\
\ddot{x}_2 \\
\cdot \\
\ddot{x}_n
\end{bmatrix} + \sum_{i=1}^{n} (\dot{r}^i + \dot{q}^i) \times \dot{x}_i + u_r
$$

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

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

\[ v_b = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \]

is the velocity of the mass center of the base

\[ v_i = \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \omega \times (r_i + q_i) \] is the velocity of the \( i \)th particle

Forming the inner products \((v_b, v_b); (v_i, v_i)\) and recalling Eq. (3-10) for \([I(t)]\) the kinetic energy can be written as

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

\[ - \frac{1}{2} (uvw)[mc]^{-1} \omega + \frac{1}{2} \omega [mc]^{-1} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{2} (uvw) \sum_{i=1}^{n} m_i q^i \]

\[ + \frac{1}{2} \sum_{i=1}^{n} m_i \{q^i\}^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{2} \omega^T \sum_{i=1}^{n} m_i (r_i + q_i) q^i - \frac{1}{2} \sum_{i=1}^{n} m_i \{q^i\}^T (r_i + q_i) \cdot \omega \]
We now rewrite those terms in \( T \) which depend upon \( q^i \) in terms of \( \dot{q}^i \)

\[
\sum_{i=1}^{n} \left( q^i \right)^T m_i \dot{q}^i = \dot{q}^T [M] \dot{q} = \dot{n}^T [\phi]^T [M] [\phi] \dot{n} = \dot{n}^T [E] \dot{n}
\]

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

\[
(\text{uvw}) \sum_{i=1}^{n} m_i \dot{q}^i = (\text{uvw}) \left[ \begin{array}{c} m_1^1 \dot{q}^1 \\ \vdots \\ m_n^1 \dot{q}^1 \end{array} \right] = (\text{uvw}) [m^1 \dot{q}^1 \ldots \dot{q}^n] [\phi] \dot{n}
\]

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

\[
U = (\text{uvw} \mid w^T \mid \dot{n} )^T
\]

Since \([I_b]\) is positive definite, an inspection of the initial expression for \( T \) reveals that \( T \geq 0 \) for all \( U \). If \( T = 0 \) then \( \dot{v}_b = w = v^i = 0 \) \((i = 1, 2, \ldots, n)\). But \( \dot{v}_b = 0 = w \) implies \((\text{uvw}) = 0 \) and \( v^i = 0 = (\text{uvw}) = w \) implies \( \dot{q} = 0 \). Hence \([\phi] \dot{n} = 0 \). Since the columns of \([\phi]\) are linearly independent we must have \( \dot{n} = 0 \) also. In other words, \( T = 0 \) if and only if \( U = 0 \). 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 $g = 0$ and $[I_b] = [0]$. We still have $T > 0$ but if $T = 0$ we can only argue that $(uvw) = 0$. We can have $T = 0$ for nonzero $\dot{u}$ and $\dot{\hat{n}}$ as long as $\dot{u} \times (r_i + q_i) + \dot{q}_i = 0 \ (i = 1, 2, \ldots, n)$. Thus for this later case $[A]$ is positive semi-definite. In particular $[A]$ 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.
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.

\[
R^{12} = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \theta_1 & -\sin \theta_1 \\
0 & \sin \theta_1 & \cos \theta_1
\end{pmatrix} \\
R^{23} = \begin{pmatrix}
\cos \theta_2 & 0 & \sin \theta_2 \\
0 & 1 & 0 \\
-\sin \theta_2 & 0 & \cos \theta_2
\end{pmatrix} \\
R^{34} = \begin{pmatrix}
\cos \theta_3 & -\sin \theta_3 & 0 \\
\sin \theta_3 & \cos \theta_3 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

\([R^4_j]\) is the transformation matrix from frame 'j' to frame 'i'.

Concatenating transformations, \([R^{14}] = [R^{12}][R^{23}][R^{34}]\)
We next derive the relationship between the body frame angular velocity $\omega$ (expressed in body coordinates) and the Euler rates $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$.

Let $\{i_p, j_p, k_p\}$ be the set of unit vectors along the axes of frame 'p' ($p = 1, 2, 3, 4$).

\[
\mathbf{\omega} = \dot{\theta}_1 i_1 + \dot{\theta}_2 j_2 + \dot{\theta}_3 k_3
\]

To express $\mathbf{\omega}$ 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

\[
\mathbf{\omega} = \begin{pmatrix}
\dot{\theta}_1 \cos \theta_2 \cos \theta_3 + \dot{\theta}_2 \sin \theta_3 \\
\dot{\theta}_2 \cos \theta_3 - \dot{\theta}_1 \cos \theta_2 \sin \theta_3 \\
\dot{\theta}_3 + \dot{\theta}_1 \sin \theta_2
\end{pmatrix}
\]

(in body frame)

This system can be inverted to yield

\[
\begin{pmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3
\end{pmatrix} = \begin{pmatrix}
\cos \theta_3 & -\sin \theta_3 & 0 \\
\cos \theta_2 & \cos \theta_2 & 0 \\
\sin \theta_3 & \cos \theta_3 & 0
\end{pmatrix} \mathbf{\omega}
\]

\[(\cos \theta_2 \neq 0)\]

\[
(7-2)
\]

\[
[R_{14}] = \begin{pmatrix}
\cos \theta_2 \cos \theta_3 & -\cos \theta_2 \sin \theta_3 & \sin \theta_2 \\
\cos \theta_1 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3 & \cos \theta_1 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3 & -\sin \theta_1 \cos \theta_2 \\
\sin \theta_1 \sin \theta_3 - \cos \theta_1 \sin \theta_2 \cos \theta_3 & \sin \theta_1 \cos \theta_3 + \cos \theta_1 \sin \theta_2 \sin \theta_3 & \cos \theta_1 \cos \theta_2
\end{pmatrix}
\]

(7-1)
CHAPTER 8
EQUATIONS OF MOTION — FIRST ORDER FORM

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

\[
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{w}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\mathbf{0}
\end{bmatrix}
- \mathbf{M}^{-1}
\begin{bmatrix}
\omega_1^2 n_1 \\
\omega_2^2 n_2 \\
\vdots \\
\omega_n^2 n_n
\end{bmatrix}
\begin{bmatrix}
\mathbf{F} + \mathbf{U} - \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{A}(t)
\end{bmatrix}
\]

(8-1)

where

\[
\mathbf{F} = \mathbf{f}^0 + \sum_{i=1}^{n} \left( \mathbf{f}^i \mathbf{a} + \mathbf{a} \mathbf{a} \right) \mathbf{f}^i
\]

(8-2)
Let \( \mathbf{R}_x', \mathbf{R}_y', \mathbf{R}_z' \) be the components of the inertial position vector of \( \mathbf{O}_g \) (origin of body frame) resolved along inertial axes and \([\Gamma]\) denote the matrix in Eq. (7-2). The kinematic relationships can now be written

\[
\left( \begin{array}{c}
\dot{\mathbf{R}}_x \\
\dot{\mathbf{R}}_y \\
\dot{\mathbf{R}}_z
\end{array} \right) = \mathbf{R}_{14}^{-1} \left( \begin{array}{c}
\dot{\mathbf{u}} \\
\dot{\mathbf{v}} \\
\dot{\mathbf{w}}
\end{array} \right), \quad \left( \begin{array}{c}
\dot{\mathbf{\theta}}_1 \\
\dot{\mathbf{\theta}}_2 \\
\dot{\mathbf{\theta}}_3
\end{array} \right) = [\Gamma] \omega
\]

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

\[
\mathbf{Y} = (\mathbf{R}_x \mathbf{R}_y \mathbf{R}_z \mathbf{R}_{12} \mathbf{R}_{13} \mathbf{R}_{23})^T \mathbf{u} \mathbf{v} \mathbf{w} \mathbf{\Omega}^T \mathbf{\Omega}_T
\]

The equations of motion written in first order form are

\[
\frac{d}{dt} \mathbf{Y} = \begin{bmatrix}
[R]^{14} & \begin{array}{c}
\dot{\mathbf{u}} \\
\dot{\mathbf{v}} \\
\dot{\mathbf{w}}
\end{array} \\
[\Gamma] \omega \\
\mathbf{\Omega}
\end{bmatrix}
\]

\[
\mathbf{A}^{-1} \mathbf{F} + \mathbf{U} - \begin{bmatrix} \mathbf{0} \\ [\Omega^2] \mathbf{\Omega}_T \end{bmatrix}
\]

This system of \((2t + 12)\) first order equations can be integrated numerically with appropriate initial conditions.
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 are used to perform certain standard computations. IMSL subroutine "EIGRS" is used for eigenvalue/eigenvector extraction and subroutine "LEQTIP" 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

\[
\begin{align*}
\{R^1, R^2, R^3, q^1, \ldots, q^n, uvw, \omega_1, \omega_2, \omega_3, q^1, \ldots, q^n\}
\end{align*}
\]

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: $F_\Psi$, $TAU_\Psi$, $FP$.

$F_\Psi$ — sum of external forces on rigid body

$TAU_\Psi$ — sum of external moments on rigid body taken about body frame origin.

$F_\Psi$ and $TAU_\Psi$ 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

* "0" 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 \( \mathbf{F} \) 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/Μ, N, MASS, RM, I;, S, NT; contains all mass and geometry data as well as the number of modes to be retained.

   Μ = mass of rigid body (real)
   N = number of particles (integer)
   MASS = masses of particles 1 through N (real N x 1 array)
   RM = position vectors of particles 1 through N prior to deformation, expressed in body frame (real 3 x N array)
   I; = 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 x 3 array)
   S = position vector from body frame origin to mass center of rigid body expressed in body frame (real 3 x 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 x 1 array)

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

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

THETA = initial 1-2-3 Euler angles of body frame with respect to inertial frame (real 3 x 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 76) 05/350 FORTRAN H EXTENDED DATE 02.27/12. 43.05 PAGE 1

REQUESTED OPTIONS: HOOD.TERM.,HOLDAP.,NAME(MAIN),AD(INDEF),OPT(0),.,FLAG(I),SIZE(360K),LC(160).

OPTIONS IN EFFECT: NAME(MAIN) HOOD(PREFIX LINE Chương(60)) SIZE(6368K) AUTOCLASS(65).

SOURCE EBCDIC MOLIST HOODDEF HOODAP NOHFILE COFIT(360K) NAMEDEF TEND(360K) IDEF FLAG(I)

C *********************************************************** 0000100
C * THIS PROGRAM SOLVES THE EQUATIONS OF MOTION OF A VEHICLE 00000300
C * CONSISTING OF A RIGID BASE WITH AN ATTACHED FLEXIBLE APPENDAGE. 00000400
C * THE APPENDAGE IS IDEALIZED AS A COLLECTION OF PARTICLES CONNECTED 00000500
C * BY HASSLESS ELASTIC STRUCTURE. IN ADDITION TO EXTERNAL FORCES ACTING00000600
C * UPON EACH OF THE PARTICLES, A FORCE AND TORQUE ARE ACCOMMODATED AT 00000700
C * THE SIMPLE FIXTURE CORRESPONDING TO THE ORIGIN OF BODY FRAME. 00000800
C * WRITTEN BY JOEL STOLCH & STEPHEN GATES C.S.D.L. BASED UPON 00000900
C * C.S.D.L. REPORT D1502 SEPT. 1982) 00001000
C *********************************************************** 00001100
C
C NOTE: ARRAYS ARE DIMENSIONED TO ACCOMMODATE A MAXIMUM OF 50 PARTICLES00001200
C
C 00015000
C 00016000
C 00017000
C 00018000
C 00019000
C
C 00020000
C 00021000
C 00022000
C
C 00023000
C 00024000
C 00025000
C 00026000
C 00027000
C 00028000
C 00029000
C 00030000
C 00031000
C 00032000
C 00033000
C 00034000
C 00035000
C 00036000
C 00037000
C 00038000
C 00039000
C 00040000
C 00041000
C 00042000
C 00043000
C 00044000
C 00045000
C 00046000
C 00047000
C 00048000
C 00049000
C 00050000
C 00051000
C 00052000
C 00053000
C 00054000
C 00055000
C 39
LEVEL 2.3.0 (JUNE 78) MAIN OS/360 FORTRAN H EXTENDED

00005400
00005500
00005600
00005700
00005800
00005900
00006000
00006100
00006200

*OPTIONS IN EFFECT=NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0384K) AUTOOBL(HOME)

*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOBJECT NOOB
SUBROUTINE SKEW(V,A)
C
C THIS SUBROUTINE CREATES THE SKewed SYMMETRIC MATRIX CORRESPONDING
C TO THE VECTOR "V".
C
REAL*8 V,A
DIMENSION V(3),A(3,3)

A(1,1)=0.0
A(1,2)=V(3)
A(1,3)=V(2)
A(2,1)=V(3)
A(2,2)=0.0
A(2,3)=V(1)
A(3,1)=V(2)
A(3,2)=V(1)
A(3,3)=0.0

RETURN
END
SUBROUTINE CROSS(A,B,C)

C THIS SUBROUTINE CALCULATES THE VECTOR CROSS PRODUCT

C A X B = C

REAL*8 A(3),B(3),C(3)

ISN 0003 C(1) = A(2)*B(3) - A(3)*B(2)

ISN 0005 C(2) = A(3)*B(1) - A(1)*B(3)

ISN 0006 C(3) = A(1)*B(2) - A(2)*B(1)

ISN 0007 RETURN

ISN 0008 END

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

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF Compilation *****
SUBROUTINE INIT

IMPLICIT REAL*8(A-H,O-Z)

REAL*8, ALLOCATABLE :: a, b, c, d

DIMENSION a(10), b(10)

COMMON /MODCO/ eta, eta0

COMMON /MODES/ ev, freq

DESCRIPTION OF NAMELIST VARIABLES

"MO" IS THE MASS OF THE RIGID BASE

"N" IS THE NUMBER OF PARTICLES THAT COM普RIZE THE FLEXIBLE APPENDAGE

"MASS" CONTAINS THE MASSES OF PARTICLES 1 TO N

"RM" CONTAINS THE POSITION VECTORS OF PARTICLES 1 THRU N IN THE UNDEFORMED STATE EXPRESSED IN THE BODY FRAME.

"I"G" IS THE INERTIA MATRIX OF THE RIGID BASE WITH RESPECT TO A FRAME LOCATED AT THE MASS CENTER OF THE BASE AND PARALLEL TO THE BODY FIXED AXIS SYSTEM.

"S" IS THE VECTOR FROM THE BODY FRAME ORIGIN TO THE MASS CENTER OF THE RIGID BODY.

"NW" NUMBER OF RETAINED MODES IN APPENDAGE VIBRATION

"UVV" IS THE INITIAL VELOCITY OF THE BODY FRAME ORIGIN EXPRESSED IN BODY COORDINATES.

"OMEGA" IS THE INITIAL ANGULAR VELOCITY OF THE BODY FRAME EXPRESSED IN BODY COORDINATES.
"R" IS THE INITIAL INERTIAL POSITION VECTOR OF THE BODY FRAME ORIGIN

"THETA" IS THE INITIAL SET OF ATTITUDE ANGLES FOR THE BODY FRAME

"DT" IS THE INTEGRATION TIME STEP

"TSTOP" IS THE TERMINAL TIME FOR THE SIMULATION

"DTI" IS THE TIME INTERVAL BETWEEN PRINTOUTS

"DTG" IS THE TIME INTERVAL BETWEEN PLOTTED POINTS

"IPILOT" IS AN ARRAY INDICATING VARIABLES TO BE PLOTTED.
NUMBERING CORRESPONDS TO LOCATION IN STATE VECTOR.
POINTS ARE PLOTTED EVERY "DTG" SECONDS.

READ(S,INPUT)
IF(INT .LE. 3NH) GO TO 5
WRITE(6,112) NT,N
STOP
DO 10 I=1,N
WRITE(6,100) R
10 WRITE(6,101) I, MASS(I), (RM(J,1),J=1,3)
1 'PARTICLE', T21, 'MASS', (SM(J,1)), 'POSITION', (FT.1),
WRITE(6,102) (ID[1],J), FM, (FT.3)
WRITE(6,103) 5, NT
WRITE(6,104) R, THETA
WRITE(6,105) 103, FORMAT(1HO,5X,'INITIAL VELOCITY=',SF7.2,3X,'FT/SEC',AX,
1 'INITIAL ANGULAR VELOCITY=',SF7.2,3X,'DEG/SEC')
WRITE(6,105) 104, UNI, OMEGA
WRITE(6,106) 102, FORMAT(1HO,5X,'INITIAL POSITION=',SF7.2,3X,'FT',4X,
1 'INITIAL ATTITUDE=',SF7.2,3X,'DEG')
READ(S,50)
WRITE(6,107) 50, DT, TSTOP, DTI, DTG
READ(S, PLOT)
DO 105 I=1,42
IF(IP(I).EQ. 0) GO TO 109
WRITE(6,108) I, IT, EQ, 109
109 IF(I.EQ. 0) GO TO 111
WRITE(6,110) (IP(I),I=1,II)
DO 110 I=1,II
WRITE(6,111) I, FORMAT(1HO,5X,'VARIABLES PLOTTED=',AX,4X,II1)
111 FORMAT(1HO,5X,'TIME STEP=',E12.4,3X,'SEC',3X,'TERMINATION TIME=',E12.4,3X,
1 'E12.4, SEC',3X,'PRINT INTERVAL=',E12.4,3X,'SEC',3X,'PLOT INTERVAL=',E12.4,3X,
1 'E12.4, SEC')
DO 112 I=1,II
FORMAT(1HO,5X,'MODES REQUESTED=',2X,II3, 'PARTICLES IN MODEL')
READ(S,50)
WRITE(6,109) 1, (T15,2E13.5)
WRITE(6,109) 107, FORMAT(1HO,5X,'S=',2E13.5, 'FT',3X,II,
1 'CONstrained APPENDAGE MODES'=I9, 200)
19 RETAINED'
WRITE(6,109) CHANGE ANGULAR VELOCITY & ATTITUDE TO RADIAN MEASURE

44
C

ISN 0054
111 DTR=DATA(1,000)/45.

ISN 0055
DO 15 I=1,3

ISN 0056
OMEGA(I)=DTR*OMEG(A(I))

ISN 0057
15 THETA(I)=DTR*THETA(I)

C

TM - TOTAL BODY MASS

ISN 0058
THMMO

ISN 0059
DO 20 I=1,N

ISN 0060
TMTH=MASSI(I)

ISN 0061
DO 30 J=1,I

ISN 0062
MOS(J)=MOS(I)+S(J)

ISN 0063
30 MCI(J)=0.9

ISN 0064
DO 40 I=1,N

ISN 0065
DO 45 J=1,3

ISN 0066
MRM(J,I)=MASS(I)*MRM(J,I)

ISN 0067
45 MCI(J)=MCI(J)+MRM(J,I)

ISN 0068
40 CONTINUE

ISN 0069
DO N=1,3

ISN 0070
42 MCI(I)=MCI(I)+MOS(I)

ISN 0071
HIS=3*N

ISN 0072
HPS=N3+6

ISN 0073
HTP=HT6

ISN 0074
HPS=HTP6

ISN 0075
H=0

C

READ IN STIFFNESS MATRIX

ISN 0076
READ(8) NDF,(I(K(I,J),J=1,NDOF),I=1,NDF)

ISN 0077
IF(NDF .EQ. NS) GO TO 400

ISN 0078
WRITE(6,102) N,NDF

ISN 0079
STOP

ISN 0080
CONTINUE

ISN 0081
400 FORMAT(HDG,10X,'INCONSISTENT DATA',EX,T3,' PARTICLES',EX,T3,
1 ' DEGREES OF FREEDOM IN STIFFNESS MATRIX')

ISN 0082
310 FORMAT(1HO,10X,'ERROR FROM IMSL ROUTINE "EIGS" ERROR CODE=',I4)

C

GET CONSTRAINED FREQUENCIES AND MODE SHAPES OF APPENDAGE

ISN 0083
L=1

ISN 0084
DO 300 J=1,N3

ISN 0085
LC=1/J3

ISN 0086
IF( (J=3*J3) .EQ. 0) LC=LC-1

ISN 0087
DO 300 I=1,J

ISN 0088
LR=1/J3

ISN 0089
IF( (1-3*J3) .EQ. 0) LR=LR-1

ISN 0090
AVL=(K(I,J)/DSQRT(MASS(LR)*MASS(LC)))

ISN 0091
L=L+1

ISN 0092
300 CONTINUE

ISN 0093
CALL EIGS(AV3,1,FREQ,EV,150,MRK,IER)

ISN 0094
IF(IER .EQ. 0) GO TO 310

ISN 0095
WRITE(6,301) IER

ISN 0096
301 FORMAT(HDG,10X,'ERROR FROM IMSL ROUTINE "EIGS" ERROR CODE=',I4)

ISN 0097
STOP

C

TRANSFORM EIGENVECTORS

ISN 0098

ISN 0099
310 DO 311 L=1,N
C LEVEL 2.3.0 (JUNE 70) INIT GS/360 FORTRAN II EXTENDED

ISN 0102
C 1+OSORT(MASS(I))

ISN 0103
DO 311 I=1,3

ISN 0104
IR=3*(I-1)*I

ISN 0105
311 EV(IR,I)=EV(IR,J)/C1

ISN 0106
DO 320 J=1,NS

ISN 0107
FHZ=DSQRT(FREQ(I)/2.083185)

ISN 0108
WRITE(6,321) I,FHZ(1:3),FREQUENCY,'(12.4,1X, 'N')',/IN

ISN 0109
321 FORMAT(31H3, 'MODE ' ,12,' FREQUENCY=' ,E12.4,1X, 'HZ')

ISN 0110
CONTINUE

ISN 0111
320 CONTINUE

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 CALCULATE TIME INVARIANT PART OF INERTIA MATRIX "INERT1"
C AND "CHAT"

ISN 0112
DO 50 I=1,3

ISN 0113
DO 50 J=1,3

ISN 0114
INERT(I,J)=0.0

ISN 0115
CMAT(I,J)=0.0

ISN 0116
50 CONTINUE

ISN 0117
DO 60 I=1,NS

ISN 0118
XS=RH(I,1)*M2

ISN 0119
YS=RH(I,2)*M2

ISN 0120
ZS=RH(I,3)*M2

ISN 0121
INERT(I,1)=INERT(1,1)+INERT(I,1)*MASS(I)*RH(I,3)

ISN 0122
INERT(I,2)=INERT(1,2)+INERT(I,2)*MASS(I)*RH(I,3)

ISN 0123
INERT(I,3)=INERT(1,3)+INERT(I,3)*MASS(I)*RH(I,3)

ISN 0124
INERT(2,1)=INERT(2,1)+INERT(2,1)*MASS(I)*RH(I,3)

ISN 0125
INERT(2,2)=INERT(2,2)+INERT(2,2)*MASS(I)*RH(I,3)

ISN 0126
INERT(2,3)=INERT(2,3)+INERT(2,3)*MASS(I)*RH(I,3)

ISN 0127
CHAT(I,1)=CHAT(I,1)+MASS(I)*XS

ISN 0128
CHAT(2,1)=CHAT(2,1)+MASS(I)*YS

ISN 0129
CHAT(3,1)=CHAT(3,1)+MASS(I)*ZS

ISN 0130
CONTINUE

ISN 0131
CHAT(I,2)=CHAT(I,2)+INERT(1,2)*M2

ISN 0132
CHAT(I,3)=CHAT(I,3)+INERT(1,3)*M2

ISN 0133
CHAT(2,1)=CHAT(2,1)+INERT(2,1)*M2

ISN 0134
CHAT(2,2)=CHAT(2,2)+INERT(2,2)*M2

ISN 0135
CHAT(2,3)=CHAT(2,3)+INERT(2,3)*M2

ISN 0136
CHAT(3,1)=CHAT(3,1)+INERT(3,1)*M2

ISN 0137
CHAT(3,2)=CHAT(3,2)+INERT(3,2)*M2

ISN 0138
CHAT(3,3)=CHAT(3,3)+INERT(3,3)*M2

ISN 0139
CONTINUE

ISN 0140
DO 70 I=1,3

ISN 0141
DO 70 J=1,3

ISN 0142
IF(I .LT. J) GO TO 70

ISN 0143
INERT(I,J)=INERT(J,I)

ISN 0144
CONTINUE

ISN 0145
70 CONTINUE

C CREATE TIME INVARIANT PORTIONS OF GENERALIZED MASS MATRIX "A"

ISN 0146

PAGE 4
LEVEL 2.3.0 (JUNE 78) INTIL 05/360 FORTRAN H EXTENDED DATE 82.24/12.21.47 PAGE 5

C
C (1,2) PARTITION "A12"
C
ISH 0147
CALL SKEW(MCO,A12)
ISH 0148
DO 80 I=1,3
ISH 0149
DO 80 J=1,3
ISH 0150
IF(I .EQ. J) GO TO 80
ISH 0152
A12(I,J)=-A12(I,J)
ISH 0153 80 CONTINUE
C
C (2,3) PARTITION "A23"
C
ISH 0154
DO 90 I=1,3
ISH 0155
L=3*I-2
ISH 0156 90 CALL SKEW(MMX(I,1),MMX(I,L))
ISH 0157
DO 94 I=1,3
ISH 0158
A23(I,J)=0.0
ISH 0159
DO 94 L=1,65
ISH 0161 94 A23(I,J)=A23(I,J)+MXM(I,L)*EV(I,J)
ISH 0162 92 CONTINUE
C
C STORE CONSTANT PARTITIONS OF "A"
C
ISH 0163
DO 200 I=1,NTP6
ISH 0164
DO 200 J=1,NTP6
ISH 0165
A(I,J)=0.0
ISH 0166 200 CONTINUE
C
C CREATE (1,1) PARTITION
C
ISH 0167
DO 210 I=1,3
ISH 0168
DO 210 J=1,3
ISH 0169 210 CONTINUE
C
C CREATE (1,3) PARTITION
C
ISH 0172
DO 225 I=1,3
ISH 0173
DO 225 J=1,65
ISH 0174 225 WRK(J,I)=0.0
ISH 0175
DO 230 L=1,165
ISH 0176 JS=3*J-2
ISH 0177
DO 230 L=1,165
ISH 0178 230 WRK(I,L)=MASS(L)
ISH 0179 JS=JS+1
ISH 0180 221 CONTINUE
ISH 0181 220 CONTINUE
ISH 0182
DO 283 I=1,3
ISH 0183
DO 283 J=1,165
ISH 0184
JNP=JS+6
ISH 0185
A(I,JNP)=0.0
ISH 0186
DO 281 L=1,165
ISH 0187 281 A(I,JNP)=A(I,JNP)+WRK(I,L)*EV(I,J)
ISH 0188 280 CONTINUE
ISH 0190 203 CONTINUE
C
C CREATE (3,3) PARTITION
C
ISH 0189  DO 230 L=1,HT
ISH 0190  230  A(L+6,L+6)=1.
C
C  SET INITIAL DEFORMATION AND RATE TO ZERO
C
ISH 0191  DO 250 I=1,N
ISH 0192  250  Q(J,I)=0.0
ISH 0193  DO 293 J=1,3
ISH 0194  293  QDOT(I,J)=0.0
ISH 0195  CONTINUE
ISH 0196  DO 293 I=1,N
ISH 0197  293  ETA(I)=0.0
ISH 0198  CONTINUE
ISH 0199  RETURN
ISH 0200  END
SUBROUTINE GHMMA

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 MASS,MCO,MRM,INERTI,MQ,MCI,ZNERT2,ZNERT3.
DIMENSION MASS(50),MCO(3),HRM(3,50),CMAT(3,3),A(156,156),
& A12(3,3),INERTI(3,3),A3X(3,150),A4X(3,50),MQ(3,50),MCI(3),
& Z2(3,3),INERTI(3,3),A3X(3,150),INERTI(3,3),DOTM(3,50),UM(3,3),
& OMEGA(3),R(3),THETA(3)
COMMON /CQNST/ TM,MASS,MCO,MCI,CMAT,N_H3,N3P6,HT_NTP6,NO
COMMON /HAT/ A,A12,INERTI,A3
COMMON /STATE/ R,THETA,Q,UVN,OMEGA,THETA1
COMMON /TDEPV/ MQ,MC1
COMMON /MODE$/ EV(150,150),FREQ(150)
DO 10 I=1,3
HC1(I)=0.0
DO 40 I=1,N
DO 45 J=1,3
MQ(J,Z)=MASS(I)(J-1)
HC1(J)=MCI(J)+MQ(J,Z)
CONTINUE
C
C Computes "A" neglecting time dependent (deformation dependent) terms
C
DO 50 J=1,3
JP3=J+3
00 50 Z=1,3
50 A(Z,JP3)=A12(Z,J)
DO 60 Z=1,3
ZP3=Z+3
60 A(IP3,JP3)=INERTI(Z,J)
DO 70 J=1,NT
JP6=J+6
DO 70 Z=1,3,
IP3=I+3
70 A(ZP3,JP6)=A23(Z,J)
C
ADD IN FIRST ORDER DEFORMATION TERMS
C
CALL SKEW(MCI,LM)
DO 80 J=1,3
JP3=J+3
DO 80 I=1,3
A12(JP3)=A12(I,J)
DO 80 J=1,3
JP6=J+6
DO 80 Z=1,3,
IP3=I+3
80 A(I,JP3)=A3X(I,J)
C
FIRST ORDER DEFORMATION TERMS IN INERTIA MATRIX - "INERT2"
C
DO 90 I=1,3
INERTI(I,100.0)
DO 90 L=1,3
DO 90 I=1,3
   SUM=0.0
   SUM=SUM+MRM(1,L)*Q(L,I)
   SUM=SUM+MRM(2,L)*Q(1,I)
   SUM=SUM+MRM(3,L)*Q(1,I)
   INERTZ1(1,L)=SUM
   INERTZ2(1,3)=SUM
   SUM=0.0
   SUM=SUM+MRM(L,L)*Q(L,I)
   SUM=SUM+MRM(1,L)*Q(1,I)
   SUM=SUM+MRM(2,L)*Q(1,I)
   INERTZ2(1,3)=SUM
   INERTZ2(2,3)=SUM
   INERTZ2(3,3)=SUM
   INERTZ3(1,1)=INERTZ2(1,1)
   INERTZ3(1,2)=INERTZ2(1,2)
   INERTZ3(1,3)=INERTZ2(1,3)
   INERTZ3(2,3)=INERTZ2(2,3)
   INERTZ3(3,3)=INERTZ2(3,3)
   INERTZ3(1,L)=INERTZ3(2,L)=INERTZ3(3,L)=SUM
   CONTINUE
DO 90 L=1,3
   SUM=0.0
   SUM=SUM+MRM(L,L)*Q(L,I)
   SUM=SUM+MRM(1,L)*Q(1,I)
   SUM=SUM+MRM(2,L)*Q(1,I)
   SUM=SUM+MRM(3,L)*Q(1,I)
   INERTZ3(1,1)=SUM
   INERTZ3(1,2)=SUM
   INERTZ3(1,3)=SUM
   INERTZ3(2,3)=SUM
   INERTZ3(3,3)=SUM
   INERTZ3(1,L)=INERTZ3(2,L)=INERTZ3(3,L)=SUM
   CONTINUE
C
C ADD IN SECOND ORDER DEFORMATION TERMS - "INERT3"
C
DO 90 L=1,3
   SUM=0.0
   SUM=SUM+MRM(L,L)*Q(L,I)
   SUM=SUM+MRM(1,L)*Q(1,I)
   SUM=SUM+MRM(2,L)*Q(1,I)
   SUM=SUM+MRM(3,L)*Q(1,I)
   INERTZ3(1,1)=SUM
   INERTZ3(1,2)=SUM
   INERTZ3(1,3)=SUM
   INERTZ3(2,3)=SUM
   INERTZ3(3,3)=SUM
   INERTZ3(1,L)=INERTZ3(2,L)=INERTZ3(3,L)=SUM
   CONTINUE
C
**LEVEL 2.3.0 (JUNE 78)**

**GNUASS**

**CS/360 FORTRAN H EXTENDED**

**DATE 02.24/12.21.50**

**PAGE 3**

```fortran
ISN 0098  LL=1
ISN 0099  220 CONTINUE
ISN 0100  260 CONTINUE
ISN 0101  SUM=0.0
ISN 0102  DO 240 I=1,N
ISN 0103  260 SUM=SUM+Q(1,I)*Q(2,I)
ISN 0104  INERT3(1,2)=SUM
ISN 0105  SUM=0.0
ISN 0106  DO 250 I=1,N
ISN 0107  250 SUM=SUM+Q(1,I)*Q(3,I)
ISN 0108  INERT3(1,3)=SUM
ISN 0109  SUM=0.0
ISN 0110  DO 260 I=1,N
ISN 0111  260 SUM=SUM+Q(2,I)*Q(3,I)
ISN 0112  INERT3(2,3)=SUM
ISN 0113  DO 270 J=1,N
ISN 0114  DO 270 I=1,J
ISN 0115  IF(I .LE. J) GO TO 270
ISN 0117  INERT3(I,J)=INERT3(J,I)
ISN 0119  DO 280 I=1,N
ISN 0120  IP3=I+3
ISN 0121  DO 280 J=1,N
ISN 0122  JP3=J+3
ISN 0124  302 FORMAT(1X,1H1,1H10X,'A MATRIX')
ISN 0125  301 FORMAT(1X,2X,15(F7.2,1X))
ISN 0126  RETURN
ISN 0127  END
```

---

**OPTIONS IN EFFECT**

- NAME(MAIN)
- NOOPTIMIZE
- LINECOUNT(60)
- SIZE(364K)
- AUTOOBL(NONE)

**OPTIONS IN SOURCE**

- EBCDIC
- NOLIST
- NODECK
- NOOBJECT
- NOSUBR
- NOSTMT
- C

**STATISTICS**

- SOURCE STATEMENTS = 126, PROGRAM SIZE = 8156, SUBPROGRAM NAME = GNUASS

***** END OF COMPILATION *****

256K BYTES OF CORE NOT USED
SUBROUTINE EXTFS(FD,TAUO,FP)

C THIS SUBROUTINE ASSEMBLES THE FORCE VECTOR "F" IN THE
C MOTION EQUATIONS AND IS PARTITIONED AS: FORCES FOR BODY
C TRANSLATION, FORCES FOR BODY ROTATION, AND FORCES FOR
C PARTICLE TRANSLATION.
C
C INPUT TO SUBROUTINE
C
C FD - EXTERNAL FORCE ON NS (AT ORIGIN OF BODY FRAME)
C TAUO - EXTERNAL TORQUE AT LOCATION OF BODY FRAME ORIGIN.
C FP - VECTOR OF EXTERNAL FORCES ON PARTICLES 1,2,....N.
C (ALL VECTORS EXPRESSED IN BODY FRAME)

IMPLICIT REAL(A-H,O-Z)
REAL*8 MASS,MCO,MRI,CMAT
DIMENSION FD(3),TAUO(3),FP(3,50),Q(3,50),RI(3,50),MRS(50),MCO(3),CMAT(3,50)
1 RI(3,50),Q(3,50),SUM(3),WV(3),WV(3),Q(3,50),UW(3),Q(3,50)
2 CMAT(3,50),THETA(3),PHI(3,50),QD(3,50),MV(3),MVI(3),DOT(3,50)
30 UJN(3),CMATDN,N3DNSP6,KTHTP6TMNS,TVI4_ONEGA,QDDT
COMMON /State/R,THETA,Q,UVI4_ONEGA,QDDT
COMMON /Const/MASS,MCO,RI,M,CMATDN,N3DNSP6,KTHTP6TMNS,TVI4_ONEGA,QDDT
COMMON /Force/F
COMMON /Modes/PHI,M

DO 10 J=1,3
10 F(J)=FD(J)

DO 20 J=1,3
20 F(J)=FD(J)

DO 30 J=1,3
30 SUM(J)=0.0

DO 40 I=1,3
40 MV(I)=RI(I,J)

CALL CROSS(MV(3),QD(3),RI(3,J))

DO 60 J=1,3
60 SLR(J)=SUM(J)

CONTINUE

DO 90 I=1,3
90 F(IP6)=0.0

DO 100 L=1,N3
100 F(IP6)=F(IP6)*PHI(L,I)*W2(L)

CONTINUE

CONTINUE

DO 130 J=1,3
130 F(IP6)=TAUO(J)+SUM(J)

DO 140 I=1,3
140 MV(I)=RI(I,J)

CONTINUE

CONTINUE

DO 160 I=1,3
160 F(IP6)=VA(I)*W2(I)

CONTINUE
SUBROUTINE MLKT

C THIS SUBROUTINE CALCULATES THE NON-LINEAR KINEMATIC TERMS IN THE C
  MOTION EQUATIONS. THESE TERMS ARE ASSEMBLED INTO THE VECTOR "UT". C

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 MASS,MC0,MRN,MC1,MC5,10,

DIMENSION MASS(50),MC0(3),MC1(3),MC5(3),MC6(3),MC1(3),
1 GD0T(3,50),UVW(3),OMEGA(3),UT(3),UV(150),UR(150),WV(3),
3 I0(3,3),I2(3),I3(50),PI(150,150),

COMMON /CONST/ TM,MASS,MC0,MRN,MC1,MC5,MC6,HT,HTP6,H6
COMMON /DEP/ MR,MC1
COMMON /STATE/ R,T,THETA,Q,UVI,IC,
COMMON /FIC/ U
COMMON /RIG/ MOS,HO,S
COMMON /ODES/ PHI,NS

EQUIVALENCE(UT(1),UV(1)),(UT(1),UR(1))

C CALCULATE DEFORMATION INDEPENDENT TERMS

CALL CROSS(UW,OMEGA,W1)
SUBW=OMEGA(1)+OMEGA(2)+OMEGA(3)

DO 20 J=1,3
20 UT(J)=TM*WV(J-1)+SUM*OMEGA(J)+I0(J,4)

CALL CROSS(MC0,WV1,W2)
DO 30 J=1,3
30 UT(J)=SUM*MC0(J)+UV(J-1)-IC(J)

CALL CROSS(OMEGA,WV3,W4)
DO 40 J=1,3
40 UT(J)=SUM*OMEGA(J)+IC(J)-UV(J-1)+IC(J)

CALL CROSS(MOS,OMEGA,WV2)
DO 50 J=1,3
50 UT(J)=SUM*OMEGA(J)+IC(J)-UV(J-1)+IC(J)

CALL CROSS(MC1,OMEGA,WV3)
DO 60 J=1,3
60 UT(J)=SUM*OMEGA(J)+IC(J)-UV(J-1)+IC(J)

CALL CROSS(MC5,OMEGA,WV4)
DO 70 J=1,3
70 UT(J)=SUM*OMEGA(J)+IC(J)-UV(J-1)+IC(J)

CONTINUE
DO 80 J=1,3
80 UT(J)=SUM*OMEGA(J)+I0(J,4)

DO 80 J=1,3
80 UT(J)=SUM*OMEGA(J)+I0(J,4)

CONTINUE
DO 80 J=1,3
80 UT(J)=SUM*OMEGA(J)+I0(J,4)

CONTINUE
DO 80 J=1,3
80 UT(J)=SUM*OMEGA(J)+I0(J,4)

CONTINUE
DO 80 J=1,3
80 UT(J)=SUM*OMEGA(J)+I0(J,4)
LEVEL 2.3.0 (JUNE 78) NLKT 05/360 FORTRAN H EXTENDED

```
00066400
00066500
00066600
00066700
00066800
00066900
00067000
00067100
00067200
00067300
00067400
00067500
00067600
00067700
00067800
00067900
00068000
00068100
00068200
00068300
00068400
00068500
00068600
00068700
00068800
00068900
00069000
00069100
00069200
00069300
00069400
00069500
00069600
00069700
00069800
00069900
00070000
```

CALCULATE FIRST ORDER DEFORMATION DEPENDENT TERMS

**DATE 82.246/12.21.54**
ISH 0101 320 NV4(J)=OMEGA(J,I) 00070010
ISH 0102 320 LL=I 00070020
ISH 0103 320 DO 330 J=1,3 00070030
ISH 0104 320 U(I,J)=0.0 00070040
ISH 0105 320 CONTINUE 00070050
ISH 0106 320 DO 307 L=1,3 00070060
ISH 0111 307 U(I,J)=U(I,J)+PHI(L,I)*UV(L,J) 00070070
C C CALCULATE SECOND ORDER DEFORMATION DEPENDENT TERMS C
ISH 0112 321 CONTINUE 00070080
ISH 0113 321 DO 340 I=1,3 00070090
ISH 0114 340 DO 311 J=1,3 00070100
ISH 0115 311 k,ll(I,J)=0.0 00070110
ISH 0116 311 CONTINUE 00070120
ISH 0117 350 NV3(J)=SUM*OMEGA(J) 00070130
ISH 0118 350 J=1,3 00070140
ISH 0119 350 CONTINUE 00070150
ISH 0120 DO 370 J=1,3 00070160
ISH 0121 370 RETURN 00070170
C OPTIONS IN EFFECT=NONE, AUTOMATIC NAME=NGM
C OPTIONS IN EFFECT=SOURCE EBCDIC, NOEDIT, NOFORMAT, NOHMS, NOSFIM, TERM IBM, IBM(1)
C STATISTICS= STATISTICAL GENERATION
C STATISTICS= NO, AUTOMATIC NAME=NLKT
****** END OF COMPILED PROGRAM *******
THZ$ SUBROUTINE ASSEMBLES THE EQUATIONS OF MOTION IN FIRST ORDER FORM AND SOLVES THE SET OF SIMULTANEOUS DIFFERENTIAL EQUATIONS (50 x 12)

THZ$ 6NH÷12

THZ$ REALNS(A-H,0-Z)

THZ$ REALNS MASS,MCO,HRM,ZHERTZ

THZ$ DIHESION MASS(50),MCO(3),HRM(3,50),CMT(3,3),A(156,156),AI2(3,3).

THZ$ 1 INERTI(3,3),A23(3,150),Q(3,50),UW4(3,3),GAM(3,3).

THZ$ 2 R(3),THETA(3),F(156,61,U156,R24(3,3),GAM(3,3),

THZ$ 3 B(156),YDOT(3,61),(3,61),N(156),N(156),

THZ$ 4 Q1500,QDOT(150),AV(1246)

THZ$ EQUIVALENCE (QV11,Q1111,QDOT1111)

THZ$ COMMON /MAAT/A,AL2,INERTI,A23

THZ$ COMMON /STATE/R,THETA,Q,UW4,OMEGA,QDOT

THZ$ COMMON /FORCE/F

THZ$ COMMON /ICFRC/U

THZ$ COMMON /TIME/ DT,TSTOP,DTP,DT8

THZ$ COMMON /MODES/ PH1(150,150),HRM(150)

THZ$ COMMON /MODEO/ ETA1(150),ETA1D(150)

THZ$ COMMON /FORCE/ ETA(150),FICFRC(150)

THZ$ COMMON /HODES/ PHZ(150,150),W$(150)

THZ$ COMMON /TME/ ETA(150),ETA1D(150)

DATA GATL/8_0,0/,ZPASS/O/

C CALCULATE R14 -TRANSFORMATION FROM BODY FRAME TO INERTIAL FRAME

C S=OSIN THETA(1)

C C=OSCIS THETA(1)

C S2=OSIN THETA(2)

C C2=OSCIS THETA(2)

C S3=OSIN THETA(3)

C C3=OSCIS THETA(3)

R1411=C=CM23

R1412=S=CM23

R1413=S=CM23

R1421=S=CM23

R1422=C=CM23

R1423=S=CM23

R1431=S=CM23

R1432=C=CM23

R1433=C=CM23

C CALCULATE "GAMA" - TRANSFORMS ANGULAR VELOCITY TO ATTITUDE RATES

C GAMA(1,1)=CM3/C2

C GAMA(1,2)=CM3/C2

C GAMA(1,3)=CM3/C2

C GAMA(2,1)=CM3/C2

C GAMA(2,2)=CM3/C2

C GAMA(2,3)=CM3/C2

C GAMA(3,1)=CM3/C2

C GAMA(3,2)=CM3/C2

C GAMA(3,3)=CM3/C2

DO 30 I=1,6

30 B(I)=F(I)*U(I)

C
LEVEL 2.3.0 (JUNE 78) SOLVE DS/360 FORTRAN H EXTENDED

ISN 0061
ISN 0044
ISN 0045
ISN 0046
ISN 0047
ISN 0048
ISN 0049
ISN 0050
ISN 0052
ISN 0053
ISN 0054
ISN 0055
ISN 0057
C
C
C
C
C
C
C
C
C
C
C
C
C
    L=1
    DO 300 J=1,HTP6
    DO 300 I=1,J
    AV(I)=A(I,J)
    L=I+1
    CALL LEGT(1,AV,1,HTP6,B,156,0,D1,D2,IER)
    IF (IER .EQ. 0) GO TO 60
    WRITE (6,61) IER
    STOP
    FORMAT (1HO,5X,’ERROR DETECTED BY IMSL LIBRARY ROUTINE ’LEGTP’)
1ERROR CODE=’,IER)
    IF (IPASS .EQ. 1) GO TO 105
    ZPASS=I
    SET NZTA VALUE OF "Y"
    O0 70 Z=1,3
    ZP3=I*3
    Y(IP3)=R(I)
    Y(L)=LNTW(Z)
    Y(LL)=OMEGA(Z)
    O0 100 Z=I,HT
    Y(L)=ETAD(Z)
    SET UP "YDOT"
    O0 110 I=1,3
    biV1(I)=R(I,1)vaJVI4(1)+R(I,2)wUVM(I)*R14(I,3)tHJ_4(3)
    MV2tI)=GAMA(I,I)*OMEGA(I)/GAM(3)6*OmgE(3)
    O0 120 I=1,3
    YDOT(I)=MY1( Z )
    IODO(I)=MY1( Z )
    O0 130 Z=I,HT
    YDOT(6*It=ETAD(Z)
    UPOATE VARIABLES IN STATE VECTOR
    CALL GOESLV(HO,Y,YDDOT,DT)
    O0 140 DO 150 Z=1,3
    R(I)=Y(I)
    O0 150 THETA(I)=Y(I)+S)
    O0 160 ETA(I)=Y(6+I)

    DO 300 1=1,HTP6
    DO 300 I=1,J
    AV(I)=A(I,J)
    L=I+1
    CALL LEGT(1,AV,1,HTP6,B,156,0,D1,D2,IER)
    IF (IER .EQ. 0) GO TO 60
    WRITE (6,61) IER
    STOP
    FORMAT (1HO,5X,’ERROR DETECTED BY IMSL LIBRARY ROUTINE ’LEGTP’)
1ERROR CODE=’,IER)
    IF (IPASS .EQ. 1) GO TO 105
    ZPASS=I
    SET NZTA VALUE OF "Y"
    O0 70 Z=1,3
    ZP3=I*3
    Y(IP3)=R(I)
    Y(L)=LNTW(Z)
    Y(LL)=OMEGA(Z)
    O0 100 Z=I,HT
    Y(L)=ETAD(Z)
    SET UP "YDOT"
    O0 110 I=1,3
    biV1(I)=R(I,1)vaJVI4(1)+R(I,2)wUVM(I)*R14(I,3)tHJ_4(3)
    MV2tI)=GAMA(I,I)*OMEGA(I)/GAM(3)6*OmgE(3)
    O0 120 I=1,3
    YDOT(I)=MY1( Z )
    IODO(I)=MY1( Z )
    O0 130 Z=I,HT
    YDOT(6*It=ETAD(Z)
    UPOATE VARIABLES IN STATE VECTOR
    CALL GOESLV(HO,Y,YDDOT,DT)
    O0 140 DO 150 Z=1,3
    R(I)=Y(I)
    O0 150 THETA(I)=Y(I)+S)
    O0 160 ETA(I)=Y(6+I)
LEVEL 2.3.0 (JUNE 78) SOLVE OS/360 FORTRAN H EXTENDED DATE 82.246/12.21.56 PAGE 3

ISH 0085 DO 170 I=1,3
ISH 0089 L=L+L+I
ISH 0090 L=L+L
ISH 0091 U0=I+U(L)
ISH 0092 170 OMEGA(I)=Y(L)
ISH 0093 DO 180 I=1,NT
ISH 0094 180 ETA(I)=Y(N+I+2*L)
C
ISH 0095 DO 200 I=1,NT
ISH 0096 QV(I)=0.0
ISH 0097 QDV(I)=0.0
ISH 0098 DD2S L=1,NT
ISH 0099 QV(I)+QV(I)+PHI(I,L)+ETA(I)
ISH 0100 QDV(I)=QDV(I)+PHI(I,L)+ETA(I)
ISH 0101 I=I CONTINUE
ISH 0102 RETURN
ISH 0103 END

C COMPUTE NEW VALUES FOR "Q" AND "QDV"

*OPTIONS IN EFFECT=NAME(=MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(6384K) AUTO DBL(NONE)

*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK NOOBJECT NOMAP NOFORMAT GOSTHT NOKREF NOFLAG terms IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 102, PROGRAM SIZE = 10788, SUBPROGRAM NAME = SOLVE

*STATISTICS* NO DIAGNOSTICS GENERATED

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

59
SUBROUTINE ODESIV(N,Y,DERIV,H)

DIMENSION DERIV(N),Y(N)

DATA IHTF/1/,C1/O.O/,C2/O.O/DCS/O./

C THIS SUBROUTINE INTEGRATES THE FIRST ORDER SYSTEM OF ORDINARY
DIFFERENTIAL EQUATIONS "DY/DT=DERIV" BY THE ADAMS METHOD
USING THIRD ORDER DIFFERENCES.

Y- SIZE OF SYSTEM
H- STEP SIZE
WITH THE NEW SOLUTION

IF(N .LE. 312) GO TO 10

IHTF:2
GO TO 5000

C FIRST CALL TO ROUTINE - Euler Integration
DO 10 I=1,N
10 Y(I) = Y(I) + H*DERIV(I)

C SECOND CALL TO ROUTINE - First Order Differences
DO 20 I=1,N
20 Y(I) = Y(I) + H*DERIV(I)

C THIRD CALL TO ROUTINE - Second Order Differences
DO 30 I=1,N
30 Y(I) = Y(I) + H*DERIV(I)

C ADAMS METHOD WITH 3RD ORDER DIFFERENCES.
DO 40 I=1,N
40 Y(I) = Y(I) + H*DERIV(I)

60
C UPDATE VECTOR 'Y'

5000 DO 60 I=1,N

60 Y(I)=Y(I)+H*(DERIV(I)+C1*BD1(I,2)+C2*BD2(I,2)+C3*BD3(I))

RETURN

END

*OPTIONS IN EFFECT(NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(5384K) AUTODBG(NONE)

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

*STATISTICS= SOURCE STATEMENTS = 43, PROGRAM SIZE = 16038, SUBPROGRAM NAME =ODESLV

*STATISTICS= NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

276K BYTES OF CORE NOT USED
SUBROUTINE PR2HT(T,F0,TAUO,F0)
SUBROUTINE PR2HT(T,F0,TAUO,F0)

REAL*8 (A-H,O-Z)
REAL*8 (A-H,O-Z)

REAL M0, MASS, MCO, MRM
REAL M0, MASS, MCO, MRM

REAL M4, TO,3|,OD(3), ROTAT
REAL M4, TO,3|,OD(3), ROTAT

Omega(3), QD(3), SP), UV(3|,OMEGA(3), QD(3), SP), UV(3|,OMEGA(3), QD(3), SP), UV(3|,OMEGA(3), QD(3), SP)

1 PCO(3), MR(3), CHAT(3,3), MASS(3,3), F0(3), TAUO(3), F0(3), TAUO(3), F0(3), TAUO(3)

2 IPLOT(42), PLTDA1(100,20)
2 IPLOT(42), PLTDA1(100,20)

COMMON /STATE/R,THETA,Q,UV_/STATE/R,THETA,Q,UV_
COMMON /STATE/R,THETA,Q,UV_/STATE/R,THETA,Q,UV_
COMMON /STATE/R,THETA,Q,UV_/STATE/R,THETA,Q,UV_
COMMON /STATE/R,THETA,Q,UV_/STATE/R,THETA,Q,UV_
COMMON /STATE/R,THETA,Q,UV_/STATE/R,THETA,Q,UV_

COMMON /CONST/ TH,MASS,CHO,MRM,CHAT,N3,N3P6,HT,NTP6,HO
COMMON /CONST/ TH,MASS,CHO,MRM,CHAT,N3,N3P6,HT,NTP6,HO
COMMON /CONST/ TH,MASS,CHO,MRM,CHAT,N3,N3P6,HT,NTP6,HO
COMMON /CONST/ TH,MASS,CHO,MRM,CHAT,N3,N3P6,HT,NTP6,HO
COMMON /CONST/ TH,MASS,CHO,MRM,CHAT,N3,N3P6,HT,NTP6,HO

COMMON /PLOT/ PLTDA1, IPLOT, HP
COMMON /PLOT/ PLTDA1, IPLOT, HP
COMMON /PLOT/ PLTDA1, IPLOT, HP
COMMON /PLOT/ PLTDA1, IPLOT, HP
COMMON /PLOT/ PLTDA1, IPLOT, HP

WRITE(4,100) T
WRITE(4,100) T
WRITE(4,100) T
WRITE(4,100) T
WRITE(4,100) T

WRITE(6,200) F0, TAU0
WRITE(6,200) F0, TAU0
WRITE(6,200) F0, TAU0
WRITE(6,200) F0, TAU0
WRITE(6,200) F0, TAU0

DO 20 IT =N, N
DO 20 IT =N, N
DO 20 IT =N, N
DO 20 IT =N, N
DO 20 IT =N, N

WRITE(6,120) I, (FP(J,1), J=1,3)
WRITE(6,120) I, (FP(J,1), J=1,3)
WRITE(6,120) I, (FP(J,1), J=1,3)
WRITE(6,120) I, (FP(J,1), J=1,3)
WRITE(6,120) I, (FP(J,1), J=1,3)

DO 10 I=1,3
DO 10 I=1,3
DO 10 I=1,3
DO 10 I=1,3
DO 10 I=1,3

T(D)=5T7.29578XTH[3TAU(L)
T(D)=5T7.29578XTH[3TAU(L)
T(D)=5T7.29578XTH[3TAU(L)
T(D)=5T7.29578XTH[3TAU(L)
T(D)=5T7.29578XTH[3TAU(L)

I00=57.29578*OMEGA(Z)
I00=57.29578*OMEGA(Z)
I00=57.29578*OMEGA(Z)
I00=57.29578*OMEGA(Z)
I00=57.29578*OMEGA(Z)

DO 20 I=1,N
DO 20 I=1,N
DO 20 I=1,N
DO 20 I=1,N
DO 20 I=1,N

MRITE(6,1101 R, UV(I4), O0)
MRITE(6,1101 R, UV(I4), O0)
MRITE(6,1101 R, UV(I4), O0)
MRITE(6,1101 R, UV(I4), O0)
MRITE(6,1101 R, UV(I4), O0)

10 FORMAT(ZHO,///,SX,'TIGHTH',FIO.3,' SEC';
10 FORMAT(ZHO,///,SX,'TIGHTH',FIO.3,' SEC';
10 FORMAT(ZHO,///,SX,'TIGHTH',FIO.3,' SEC';
10 FORMAT(ZHO,///,SX,'TIGHTH',FIO.3,' SEC';
10 FORMAT(ZHO,///,SX,'TIGHTH',FIO.3,' SEC';

110 FORIlAT(1H, SX,'R=*,3(X1PEll.4),* FT'/4X,'THETA=',
110 FORIlAT(1H, SX,'R=*,3(X1PEll.4),* FT'/4X,'THETA=',
110 FORIlAT(1H, SX,'R=*,3(X1PEll.4),* FT'/4X,'THETA=',
110 FORIlAT(1H, SX,'R=*,3(X1PEll.4),* FT'/4X,'THETA=',
110 FORIlAT(1H, SX,'R=*,3(X1PEll.4),* FT'/4X,'THETA=',

110 FORMAT(5X,'G',I3,*=',3(ZX,1PE11.4),' FT')
110 FORMAT(5X,'G',I3,*=',3(ZX,1PE11.4),' FT')
110 FORMAT(5X,'G',I3,*=',3(ZX,1PE11.4),' FT')
110 FORMAT(5X,'G',I3,*=',3(ZX,1PE11.4),' FT')
110 FORMAT(5X,'G',I3,*=',3(ZX,1PE11.4),' FT')

130 FORMAT(1H, 2X,'Q',I3,*=',3(ZX,1PE11.4),* FT')
130 FORMAT(1H, 2X,'Q',I3,*=',3(ZX,1PE11.4),* FT')
130 FORMAT(1H, 2X,'Q',I3,*=',3(ZX,1PE11.4),* FT')
130 FORMAT(1H, 2X,'Q',I3,*=',3(ZX,1PE11.4),* FT')
130 FORMAT(1H, 2X,'Q',I3,*=',3(ZX,1PE11.4),* FT')

200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/

202 FORMAT(1H, 4X,'F',I3,*=',3(ZX,1PE11.4),* FT')
202 FORMAT(1H, 4X,'F',I3,*=',3(ZX,1PE11.4),* FT')
202 FORMAT(1H, 4X,'F',I3,*=',3(ZX,1PE11.4),* FT')
202 FORMAT(1H, 4X,'F',I3,*=',3(ZX,1PE11.4),* FT')
202 FORMAT(1H, 4X,'F',I3,*=',3(ZX,1PE11.4),* FT')

200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/
200 FORMAT(1HO, SX,'FO=',3(X1PEZ1.4),' LS',/

RETURN
RETURN
RETURN
RETURN
RETURN
RETURN

ENTRY GRAF(T)
ENTRY GRAF(T)
ENTRY GRAF(T)
ENTRY GRAF(T)
ENTRY GRAF(T)
ENTRY GRAF(T)

IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203

C STORE VARIABLE$ FOR PLOTTZNG
C STORE VARIABLE$ FOR PLOTTZNG
C STORE VARIABLE$ FOR PLOTTZNG
C STORE VARIABLE$ FOR PLOTTZNG
C STORE VARIABLE$ FOR PLOTTZNG
C STORE VARIABLE$ FOR PLOTTZNG

C STORE INERTIAL POSITION
C STORE INERTIAL POSITION
C STORE INERTIAL POSITION
C STORE INERTIAL POSITION
C STORE INERTIAL POSITION
C STORE INERTIAL POSITION

C STORE ATTITUDE ANGLES IN DEGREES
C STORE ATTITUDE ANGLES IN DEGREES
C STORE ATTITUDE ANGLES IN DEGREES
C STORE ATTITUDE ANGLES IN DEGREES
C STORE ATTITUDE ANGLES IN DEGREES
C STORE ATTITUDE ANGLES IN DEGREES

DO 30 J=1,3
DO 30 J=1,3
DO 30 J=1,3
DO 30 J=1,3
DO 30 J=1,3
DO 30 J=1,3

IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203
IF(IPLOT(I) .EQ. O) GOTO 203

PLTDAT(NP+1)=F(NP)
PLTDAT(NP+1)=F(NP)
PLTDAT(NP+1)=F(NP)
PLTDAT(NP+1)=F(NP)
PLTDAT(NP+1)=F(NP)
PLTDAT(NP+1)=F(NP)

GO TO 300
GO TO 300
GO TO 300
GO TO 300
GO TO 300
GO TO 300
GO TO 300
ISTM 0050  
ISTM 0051  L=NY-4  
ISTM 0052  I1=1+L/3  
ISTM 0053  I2=I-L*(I1-1)  
ISTM 0054  IF(I2 .GT. 0) GO TO 321  
ISTM 0055  I1=I1-1  
ISTM 0056  I2=3  
ISTM 0057  321 PLTDAT(NP,I+1)=Q(I2,I1)  
ISTM 0058  GO TO 300  
ISTM 0059  330 IF(NY .GT. (N3P6+3)) GO TO 340  
ISTM 0060  C STORE TRANSLATIONAL VELOCITY  
ISTM 0061  PLTDAT(NP,I+1)=UV(NY-N3P6)  
ISTM 0062  GO TO 300  
ISTM 0063  340 IF(NY .GT. (N3P6+6)) GO TO 350  
ISTM 0064  C STORE ANGULAR VELOCITY IN DEG./SEC.  
ISTM 0065  PLTDAT(NP,I+1)=57.29578*OMEGA(NY-(N3+9))  
ISTM 0066  GO TO 300  
ISTM 0067  350 I=I1+L*(I2-I1)  
ISTM 0068  IF(I2 .NE. 0) GO TO 351  
ISTM 0069  I1=I1-1  
ISTM 0070  I2=3  
ISTM 0071  351 PLTDAT(NP,I+1)=QDOT(I2,I1)  
ISTM 0072  360 CONTINUE  
ISTM 0073  203 RETURN  
ISTM 0074  END  

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINESCOUNT(60) SIZE(1384K) AUTOOBL(HOME)  
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NOOBJECT NOFORMAT GOSTMT NOXREF NOALSF TERM IBM FLAG(I)  
*STATISTICS* SOURCE STATEMENTS = 76, PROGRAM SIZE = 2324, SUBPROGRAM NAME = PRINT  
***STATISTICS*** NO DIAGNOSTICS GENERATED  
****** END OF COMPILATION ****** 260K BYTES OF CORE NOT USED

LEVEL 2.3.0 (JUNE 78) PRINT Q5/360 FORTRAN H EXTENDED
SUBROUTINE PLOT

REAL RAH,PLDAT
DIMENSION IPUT(42),PLDAT(100,20),IT(144),RAH(4),IC(10),
I IMAG(512)

COMMON /PLOTT/ PLDAT,IPUT,HP
DATA IT(1)/0/,RAH/400/,IC(1)/100/,IMAG/100/,HP/I

C CALCULATE NUMBER OF VARIABLES TO BE PLOTTED
C

HV=0
DO 100 I=1,40
IF(IPL0T(I).EQ.0) GO TO 110
HV=HV+1
100)
IF(HV.EQ.0) RETURN
DO 120 IP=1,HP
CALL USPLT(PLDAT(I,1),PLDAT(I,IP+1),100,HP,1,IT,RAH,IC,1,
1 IMAG,IER)
120)
CONTINUE
RETURN
190 END

*OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(0364K) AUTODBL(NONE)
*OPTIONS IN EFFECT: SOURCE EBCDIC NOLIST NOOBJECT NMAP NFORMAT GOSTT NOXREF NOHALC NOAMSF TERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 10. PROGRAM SIZE = 21736. SUBPROGRAM NAME = PLOT
*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPIIIATION ******
*STATISTICS* NO DIAGNOSTICS THIS STEP

276K BYTES OF CORE NOT USED
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

\[ r_s = -\frac{b}{2}i_4 + \frac{h}{2}k_4 \]

\[ r^1 = -\frac{b}{2}i_4 + (b + L)j_4 + \frac{h}{2}k_4 \]

\[ r^2 = -\frac{b}{2}i_4 + (b + 2L)j_4 + \frac{h}{2}k_4 \]

\[ r^3 = -\frac{b}{2}i_4 - Lj_4 + \frac{h}{2}k_4 \]

\[ r^4 = -\frac{b}{2}i_4 - 2Lj_4 + \frac{h}{2}k_4 \]

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

\[ \bar{R} = R_x i_1 + R_y j_1 + R_z k_1 \]
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

\[ t_0 \]

\[ R_x = 1 \cdot 10^3 \text{ ft} \quad \theta_1 = 5 \text{ degrees} \]

\[ R_y = 2 \cdot 10^3 \text{ ft} \quad \theta_2 = 20 \text{ degrees} \]

\[ R_z = 3 \cdot 10^3 \text{ ft} \quad \theta_3 = 0 \text{ degrees} \]

\[ u = 0 \text{ ft/s} \quad \omega_1 = 0 \text{ deg/s} \]

\[ v = 5 \text{ ft/s} \quad \omega_2 = 10 \text{ deg/s} \]

\[ w = 0 \text{ ft/s} \quad \omega_3 = 0 \text{ deg/s} \]

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: \( R_y, \theta_2, q_x, q_y, q_z, v, \omega_2 \).
NAMELIST Input Data

&INPUT M@ = 5.0, N = 4, MASS = 1.0, 0.5, 1.0, 0.5,
RM = -0.5, 1.0, 1.0, -0.5, 21.0, 1.0, -0.5, -10.0, 1.0, -0.5,
-20.0, 1.0,
IQ = 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 ILOT = 2, 5, 16, 17, 18, 20, 23 &END
LIST OF REFERENCES


