NASA CONTRACTOR REPORT

NASA CR-1119

NASA CR-1119

DIGITAL PROGRAM FOR DYNAMICS OF NON-RIGID GRAVITY GRADIENT SATELLITES

by James L. Farrell, James K. Newton, and James J. Connelly

Prepared by
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.
for Goddard Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1968

NASA CR-1119

DIGITAL PROGRAM FOR DYNAMICS OF NON-RIGID GRAVITY GRADIENT SATELLITES

By James L. Farrell,

James K. Newton, and

James J. Connelly

Prepared under Contract No. NAS 5-9753-10 by
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.

for Goddard Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

ABSTRACT

A digital program has been written to determine the dynamic behavior of discretized models for gravity gradient satellite structures. Both passive (elastic reaction, damping) and active (controller) internal torques can be included in the computational model. The program can be utilized simply by observing straightforward directions given in the introductory section of this report, and a concrete example (hinged assembly model of the Radio Astronomy Explorer satellite) of program adaptation is described in detail. To facilitate application to other configurations a clear separation is made between 1) computations applicable to a general gravity gradient satellite, and 2) specific RAE computations.

The basis for this digital program is the Roberson-Wittenburg dynamical formalism, noted and referenced in the text. This formalism grew from the desire to systematize the rigorous dynamic analysis of structures with multiple interconnected members. In programming the formulation for the present problem, it was found possible to supply additional details applicable to a fairly general class of gravity gradient satellites. Thus the general portion of the program contains provisions for straightforward implementation of internal moments (passive spring and damper or active controller torques inherently provided as simple functions of integrated rates and attitude; constraint torques at locked hinge axes automatically accounted for by a simple indexing scheme), as well as solar radiation pressure and thermal (direct Earth and direct plus reflected solar heating) effects. The necessary astronomical and kinematical expressions are supplied in a standard form, with explicit relations valid for eccentricities up to one-tenth.

Practical implementation of the dynamical formalism calls for the following computational refinements: 1) artificial enlargement of small members, to hasten the integration of high frequency oscillations without materially affecting the overall (low frequency) excursions; 2) hinge interactions to enhance the accuracy of numerical differentiation, necessitated by moment characterization for discretized elastic members; 3) the use of weak restraining springs and dampers at locked joint axes, to counteract the double integration of small numerical errors incurred by the constraint torque formulation; 4) representation of torsionless biaxial bending by an orthogonal matrix with one vanishing eigenvector component; and 5) the use of kinetic or potential energy considerations in fitting segments to an elastic curve.

Through successful comparison with an independent Lagrangian model analysis, a three-segment model was deemed sufficient for each of the 750 ft. ½ inch dia. RAE antenna booms. Each orbit (approximately 4 hours) of the undisturbed satellite then requires roughly one hour of simulation time, and the machine time is approximately doubled by introduction of thermal effects (solar pressure has a less pronounced effect). The expensive nature of the program is attributed to modeling accuracy (e.g., full nonlinear coupling; the interaction between dynamical behavior and forcing functions; etc.) plus the large number of integrated variables, in comparison with the number of independent co-ordinates.

FOREWORD

This program was written for use by Goddard Space Flight Center, in the dynamic analysis of the Radio Astronomy Explorer satellite, under NASA Contract No. NAS5-9753-10. In combination with additional work performed under this contract (tasks 15 and 20), the results will provide (1) maximum insight into the three-dimensional flexible satellite motion, (2) comparison between this segmented model dynamics and another independent structural analysis (a Lagrangian modal analysis, documented separately), and (3) complete preparation for an operational program which provides statistical filtering of boom tip information (intermittently received by TV stations at fixed points on the Earth) in combination with attitude and damper position measurements.

TABLE OF CONTENTS

TABLE OF CONTENTS (con't)

INTRODUCTION

Accurate three-dimensional analysis of non-rigid assemblies has enjoyed only limited feasibility and flexibility in the past, due to the existence of unknown internal forces and moments which influence the relative motion between members. In many applications these relative motions interact with the rigid body degrees of freedom (e.g., the flexural behavior of a satellite boom changes the moments of inertia which in turn affect the attitude dynamics). Because of the resulting complexity, previous investigations have often employed analytical transformations whose detailed form depended heavily upon the specific configuration studied.

In order to provide a method applicable to a more general class of dynamical situations (e.g., flexural and torsional behavior of discretized structural models; attitude control of a hinged satellite assembly), a two-body Euler formulation devised by Fletcher, Rongved, and Yu[1] was extended to the N-body case by Hooker and Margulies;[2] the explicit development was then advanced by Roberson and Wittenberg.[3] These recent advances have been employed in a digital program capable of describing the rotational dynamics (attitude matrices and inertial angular rates) of multiple interconnected rigid bodies. The program is applicable to structural or attitude control problems subject to the conditions (1) existence of a unique path between any pair of bodies (the arrangement then conforms to the definition of a topological "tree"), and (2) characterization of interconnections by hinges which, for any pair of adjacent bodies, must be fixed in both members.

The present scope is restricted to gravity gradient satellites in a Keplerian orbit with eccentricity less than 0.1. In addition to the effects of ellipticity on gravity gradient torque, the program includes solar radiation pressure (at 100% reflectivity) and thermal bending of booms; heat flux sources are the sun and Earth (direct), plus the component of solar heat reflected by the Earth.

For a reasonably general class of satellites falling within the above scope, the program lends itself quite readily to implementation of accurate dynamic analysis. Although the pertinent mathematical developments (derivations in Refs. 2 and 3, augmented by the additions in Appendix A of this report) involve several arrays of variables, techniques for computer storage optimization have produced a practical computational arrangement for structures containing up to 26 members.* Aside from a few possible adjustments involving specific satellite geometry,** all that is necessary for immediate use of the program is a specification of the familiar satellite parameters, listed together with the corresponding Fortran designations in quotes below; (Appendix B demonstrates this specification procedure for an illustrative model of the Radio Astronomy Explorer (RAE) satellite;[4] the explicit thoroughness of the Roberson-Wittenburg[3] formulation is attested by the extreme simplicity of these parameters):

<u>System Parameters</u>

N ("N") The number of rigid bodies in the system (There are then N-1 hinges).

m ("EM") A vector (N x 1) defining the mass of each body.

---

*To exemplify the demand for machine capacity it is noted that, with 26 members, the augmented inertia matrix of Ref. 3 has $(26 \times 3)^2$, or over 6000, elements. This alone consumes about twenty per cent of the IBM 7094 core (the program is written <u>in single precision</u>).

** If solar pressure and/or thermal effects are to be taken into account ($A_E \neq 0$, $J'_E \neq 0$, or $J'_S \neq 0$), see Appendix C.
Also, if formulations of curvature (for hinge moments of discretized booms) require accurate numerical derivatives, the "interacting joint" technique exemplified in Appendix B must be used.

I ("A") A third order tensor (N x 3 x 3) containing the principal moments of inertia for each body. (The off-diagonal terms, though initially zero, will be appropriately augmented in accordance with the dynamics in the program).

R ("R") A third order tensor (N-1 x 3 x 3) of restraining torque coefficients which generate position feedback (for control problems) or elastic reaction (for structural analysis) at each hinge.

R' ("RP") A third order tensor (N-1 x 3 x 3) similar to R, but generating hinge torques proportional to relative angular rates between each pair of adjacent bodies.

S ("S") A matrix (N x N-1) of ones and zeros constructed simply as follows: Number the bodies (1 to N) and the hinges (1 to N-1). Set S_{ij} to zero for every combination of unconnected body (i) and hinge (j). For each pair of adjacent members (I and K) identify the one (body I) to which the coefficients R and R' are referenced.* Set S_{IJ} to +1 and S_{KJ} to -1 (where J is the hinge connecting the I and K members).

ρ ("RHO") A third order tensor (N-1 x 3 x 3) defining the orthogonal transformation between each pair of adjacent body principal axes in the undeformed or rest position. In the notation of the preceding item, $[\rho_J]$ is the transformation from K to I co-ordinates.

*One example requiring such an identification would be a skewed boom hinged to a satellite hub. The rotational degrees of freedom of the hinge would presumably be referenced to principal axes of the boom (bending and torsion), rather than the hub.

C ("C") A matrix (3N x 3N) generated from S as follows: All elements outside the upper left (3N x N-1) array are zero.* For each zero in S, place a (3 x 1) null vector at the corresponding position in the upper left (3N x N-1) corner of C. Choose a right-hand convention for positive rotations about principal axes of each body (consistent with the definition of A), and express each mass center-to-hinge vector in these principal co-ordinates. For each nonzero element of S, multiply the corresponding mass center-to-hinge vector by S_{ij} and enter this product in the corresponding location of C.

A_E ("AE") A vector (N x 1) defining the effective surface area of each body, assuming 100% reflectivity for solar radiation pressure.

N_c ("NC") The total number of locked hinge axes in the system. (To clarify this definition it is noted that the system has $3N-N_c$ rotational degrees of freedom). The present program capacity allows up to thirty-eight locked modes.

$\mathcal{M}$ ("MI") A vector $\{3(N-1) \times 1\}$ defined simply as follows: For every locked hinge axis identify the joint number (J) and the locked mode (α = 1, 2, 3 for x, y, z respectively); compute the argument $j = 3\,(J-1) + \alpha$. Number the locked modes 1, 2, ...N_c and, for each value of (j) representing a locked mode, set M_j equal to this index number. For all other values of (j), set M_j to zero.

*These locations are used at a later point, after C is no longer needed and its storage is utilized for other purposes.

J_E' ("XJE") Thermal bending constant computed as shown in Equation (A-19) from Earth heat flux (J_E; Equation A-15); linear thermal coefficient of expansion (e); boom segment length (ℓ), diameter (d), thickness (τ), earth heat absorptivity (a_E), and thermal conductivity (κ).

J_S' ("XJS") Thermal bending constant computed as shown in Equation (A-19) from the above parameters, with (J_E) and (a_E) replaced by the solar heat flux (J_S; Equations A-16 and A-17) and solar heat absorptivity (a_S), respectively.

Initial Conditions and Program Control

Θ ("TH") A third order tensor (N x 3 x 3) containing the direction cosine transformation from each set of body axes (defined for C above) to reference axes.* The reference axes are defined by the upward local vertical (+Z) and the orbit pole (+Y).

ω ("WV";"WM") A vector ("WV"; 3N x 1) equivalenced to a matrix ("WM"; 3 x N) containing the absolute angular rate for each body, expressed in its own (principal axis) co-ordinate frame.*

T ("ORBS") Total number of orbits to be simulated in one run.

N_R ("ENR") Number of readouts per orbit.

E_L ("ERL") A vector (12 N x 1) of allowable absolute error per integration step for angular rates (rad./sec.; 1 to 3N) and direction cosines (3N + 1 to 12N).

*It is thus seen that these addresses contain the desired information (satellite attitude, shape, angular rates) which can be read out at any time. To begin the computer run, the initial values are stored in these locations.

Astronomical Parameters

a_o ("AZ") Semi-major axis of orbit.

e_o ("EZ") Eccentricity of orbit. (The present program assumes $e_o \leq 0.1$, but an extension could readily be made).

t_o ("TZ") Time at periapsis, relative to the time (t = 0) at the start of the simulation run.

i_o ("EYZ") Orbital inclination.

Ω_o ("THZ") Longitude of the ascending node.

ω_o ("WZ") Argument of the perigee.

N_D ("ND") Launch date (e.g., $N_D = 1$ for January 1).

For most programs it will be convenient to compute many of the above parameters from other, more basic, inputs (e.g., length, modulus of elasticity, angles at connecting points, etc.). This portion of the program will therefore consist of (a) Part I; controllable (punched card) inputs, and (b) Part II; fixed and derived inputs. Again, reference is made to Appendix B for an illustration. It is noted that the present program setup calls for inputs in MKS units, and the above angle inputs should be expressed in degrees. Also, any of the above provisions (auxiliary variables, additional dimension statements, print-out directions, etc.) must also be added to suit the individual problem under consideration. In general, the desired readouts will be simple functions of the angular rates (ω-array) and attitude matrices (θ-array).

GENERAL PROGRAM COMPUTATIONS

The preceding introductory material contains the information required for program utilization. For those interested in the approach, the present discussion describes the fundamentals of the formulation (Refs. 1-3), and additional detail is included in the Appendix.*

To determine the behavior of coupled rigid body motion, the rotational dynamics are first expressed as a set of equations in the usual form,

$$[I]\,\dot{\underline{\omega}} + [\tilde{\omega}][I]\,\underline{\omega} = \underline{\mathcal{T}} \qquad (1)$$

where $[I]$, $\underline{\omega}$, $[\tilde{\omega}]$, and $\underline{\mathcal{T}}$ denote inertia tensor, angular rate vector, the operator ($\underline{\omega}\times$), and total torque vector, respectively. Since this Euler relation holds for each of the (N) members of the structure, Eq. (1) can be thought of as a (3N) dimensional equation; $\underline{\omega}$ therefore has (3N) components, representing the absolute angular rate of each member as previously defined. The total torque vector $\underline{\mathcal{T}}$ consists of (a) external torques, (b) internal torques, and (c) moments of internal forces. Since the internal forces are generally unknown and are not of primary interest in themselves, it is desirable to replace them by equivalent quantities obtained from Newton's laws. Consequently the internal forces are re-expressed in terms of external and d'Alembert forces; the motion of the composite structure mass center is then eliminated from the equations. As a result, the d'Alembert forces can be defined by second derivatives of position vectors relative to this composite mass center. Through the dynamical formalism, the moments of these d'Alembert forces are written in a convenient computational arrangement whereby

*This report describes only the details of <u>implementing</u> the dynamic computations; for details of the formalism itself the reader is referred to Ref. 3.

(1) part of the centripetal component is included as a constituent $\underline{Q}$ of the total torque $\underline{\mathcal{T}}$;

(2) the remainder of the centripetal component is taken into account through replacing $[I]$ in the second term of Eq. (1) by a constant augmented inertia matrix $[K]$;

(3) the tangential component (associated with $\dot{\underline{\omega}}$) is taken into account through replacing $[I]$ in the first term of Eq. (1) by the augmented inertia matrix $[K+\Psi]$, where $[\Psi]$ varies as a known function of the previously defined attitudes $[\theta]$.

With the external and internal torques, and the moments of the external forces denoted (as in Ref. 3) by $\underline{L}$, $e^T \circledcirc [S] \underline{\mathcal{X}}$, and $[P]\underline{F}$ respectively, Eq. (1) is rewritten as Eq. (15) of Ref. 3:

$$[K+\Psi]\dot{\underline{\omega}} + [\tilde{\omega}][K]\underline{\omega} = \underline{L} - [P]\underline{F} + e^T \circledcirc [S]\underline{\mathcal{X}} + \underline{Q} \qquad (2)$$

To adapt this formulation to the present program, the two terms in the gravity gradient expression (Eq. 19 of Ref. 3) are symbolized here as ($\underline{H} - \underline{G}'$) respectively; the transformed force vector $\underline{G}''$ is then substituted for $\underline{G}'$ to include solar pressure effects (Eqs. A-40 and A-41 in Appendix A of this report). The first two terms on the right of Eq. (2) can then be expressed as

$$\underline{L} - [P]\underline{F} = \underline{H} - \underline{G}'' \qquad (3)$$

The internal torque vector is then separated into two constituents as suggested in Ref. 2;

$$e^T \circledcirc [S]\underline{\mathcal{X}} = \underline{\mathcal{X}}' + [\xi]\underline{T}_c \qquad (4)$$

where $\underline{\mathcal{L}}'$ includes all spring and damper torques while $\underline{T}_c$ is a vector in which the ($i^{\underline{th}}$) component represents the constraint torque in the ($i^{\underline{th}}$) locked mode (see the definitions for N_c and $\mathcal{M}$ in the preceding section); $[\Upsilon]$ is the matrix defined by Eq. (A-28). The quantities on the left of Eq. (2) are written as

$$[\Gamma] \triangleq [K + \Psi] \quad ; \quad \underline{W} \triangleq [\tilde{\omega}][K]\underline{\omega} \tag{5}$$

so that

$$[\Gamma]\underline{\dot{\omega}} = \underline{E} + [\Upsilon]\underline{T}_c \tag{6}$$

where

$$\underline{E} = -\underline{W} + \underline{H} - \underline{G}'' + \underline{\mathcal{L}}' + \underline{Q} \tag{7}$$

In Appendix A it is shown that this leads to an expression of the form

$$[\Gamma]\underline{\dot{\omega}} = \underline{E} - [\Upsilon]\left\{[\Upsilon]^T[\Gamma]^{-1}[\Upsilon]\right\}^{-1}\left\{[\Upsilon]^T[\Gamma]^{-1}\underline{E} + \underline{\nu}\right\} \tag{A-31}$$

which is the actual equation solved through numerical integration in this program.

Just as the specific system portion of the program has been divided into (a) Part I - Controlled inputs, (b) Part II - Fixed and derived inputs, and (c) Readouts, the general program operations fall into three categories:

(a) Part 0 - Program setup (e.g., dimensions, physical constants, etc.)

(b) Part III - General system constants (e.g., barycentric vectors as defined in Ref. 3, etc.), and

(c) Part IV - Evaluation of derivatives and numerical integration.

The Fortran nomenclature and operation sequence were chosen to maintain reasonable storage requirements without incurring any appreciable loss in computation efficiency. The six largest arrays consume roughly twenty thousand storage locations, accommodating a maximum of 26 members and up to 38 locked modes for the complete satellite assembly. All steps of the general computation are identified in Appendix C.

APPENDIX

Most of the detailed theoretical background for the gravity gradient satellite program is contained in Ref. 3. In restricting the Roberson-Wittenburg approach to this application, however, it was found that additional aspects of the formulation (e.g., hinge moments, additional forces, etc.) could be defined more specifically with little further loss of generality. The various computations added to the general program are described in Appendix A.

In order to illustrate in a concrete manner how this program can be applied to an existing satellite, a model of the Radio Astronomy Explorer[4] is described in Appendix B. A detailed description of the computation then follows in Appendix C, in the form of an annotated Fortran listing.

It was found convenient to treat much of the notation in an individual sectional basis, with various quantities defined in the text of the derivations. The Roberson-Wittenburg notation[3], however, and its additions (e.g., augmentation of external force by the solar pressure, etc.) described in the body of this report, are retained. Components of torques, angular rates, and angular accelerations, for example are expressed in the co-ordinate frame of the appropriate structural member; it follows that vector equations are generally written in these body co-ordinates. The IJK index (previously defined in terms of the incidence matrix $[S]$ and the satellite shape $[\rho]$), and much of the additional notation defined in the body of the report, arises repeatedly throughout the analysis. For the model of gravity gradient booms, the cross-section is presumably circular (either solid or hollow); the length is chosen along the body x-axis, with (y) and (z) along the principal inertia axes of the cross-section. In all cases, the principal inertia axes are used for body co-ordinates, and standard right hand conventions are used for angle transformations; the matrix notations $[\ ]^T$; $tr[\ ]$; and

$[\beta]_u$ represent transpose; trace; and an orthogonal transformation obtained by a positive rotation of (β) radians about the u-axis, respectively. $\underline{1}_\alpha$ represents the (α) column of the 3×3 identity matrix $[I_{33}]$.

APPENDIX A

ANALYTICAL FORMULATIONS APPLICABLE TO THE GENERAL PROGRAM

In Ref. 3 the hinge moment computation was left open in order to maintain generality of scope for the dynamical formalism. For the gravity gradient satellite program it has been found that the torque at each joint can be characterized by a convenient formulation applicable to numerous hinge types. The method uses straightforward program logic based on the incidence matrix $[S]$, with the torque computed from the eigenvector and trace angle of the orthogonal transformation between adjacent members. When the "rest position" of one member relative to another is variable (e.g., due to thermal bending), the same basic formulation is augmented in a straightforward manner. The "locked mode" torque described in Ref. 2, for hinges with less than three degrees of freedom, has also been programmed.

In addition to providing explicit hinge moment computations, the program includes the force on each member due to solar radiation pressure. Finally there is a kinematical relation appropriate for satellites in low eccentricity orbits, and the position of the sun must also be defined in relation to satellite orientation. All of these items are covered by the analytical background material in this Appendix.

Hinge Torques

From the INTRODUCTION it is recalled that the undeformed shape of the satellite is defined in terms of the matrices $[\rho_J]$, where J ($\leq N-1$) is the hinge index number. The next section illustrates how a modified matrix $[\rho_J']$ performs this function when thermal effects are included. It follows that the reaction torque at hinge J is zero when the relative orientation $\{[V']$

$\hat{=} \; [\theta_I]^T [\theta_K] \}$ of the two members touching this hinge is equal to $[\rho_J']$; in general the hinge moment is a function of the deformation matrix,

$$[V] = [\rho_J'][V']^T \tag{A-1}$$

More specifically, the reaction torque is a function of the trace angle,

$$\lambda = \text{Arccos}\left\{\tfrac{1}{2}\, tr\,[V] - \tfrac{1}{2}\right\} \tag{A-2}$$

and the unit eigenvector $\underline{U}$ of $[V]$ which points along the positive axis of rotation. This vector satisfies the equation (denoting the 3 x 3 identity by I),

$$[V - I]\underline{U} = \underline{0} \tag{A-3}$$

and therefore can be computed from the cross product of any two nonvanishing* rows of $[V - I]$.

The product ($\lambda \underline{U}$) can be thought of as a displacement vector which generates a reaction torque. Immediately this suggests the form for the "position feedback" torque for a linear system having negligible delay: $\lambda [R_J] \underline{U}$, where $[R_J]$ is the restraining matrix for hinge J , described in Section 1.1. For attitude control problems the elements of $[R]$ are easily identified as the controller sensitivities; for structural applications $[R]$ consists of rigidity coefficients which are readily derived from the small angle** flexure and torsion formulas,

*e.g., when $\underline{U}$ is along the x-axis the first row of $[V - I]$ will vanish. Furthermore, when $\underline{U}$ is too close to the x-axis the use of the first row of $[V - I]$ in the computation would lead to numerical problems. Program logic avoids inaccuracy of this type.

**This restriction of course holds only for structural reaction torques. In contrast to restricting the scope of application, moreover, this can be viewed as a requirement imposed upon (N) since the angles can be made smaller by separating the model into a larger number of segments. Finally, it is noted that the hinge moment formulation could be modified to account for larger angles.

$$\text{Bending Moment} \doteq E\mathcal{J}\ d\theta_B/d\ell \tag{A-4}$$

$$\text{Torsion Moment} = G\mathcal{J}\ d\theta_T/d\ell \tag{A-5}$$

where E, G, $\mathcal{J}$, and $\mathcal{J}$ represent Young's modulus and the shear modulus of elasticity, the in-plane and polar area inertia moments of the structural member cross-section, respectively; ($d\theta_B/d\ell$) is a measure of bending curvature which can be approximated* by ($\lambda U_i/\ell$), where U_i is the component of $\underline{U}$ along the axis implied in Equation (A-4), and (ℓ) is the distance between centers of the members joined by hinge J ; a similar approximation is used for the torsional gradient ($d\theta_T/d\ell$). It follows that, for structural members with no inherent coupling between bending and torsion (such as that which would arise from displacement between mass and shear centers), $[R_J]$ is a diagonal matrix with elements ($G\mathcal{J}/\ell$), ($E\mathcal{J}_y/\ell$), and ($E\mathcal{J}_z/\ell$), where $\mathcal{J}_y$ and $\mathcal{J}_z$ correspond to principal axes of the cross-section.*

In addition to the above position feedback or elastic reaction, there may be a moment restraining relative angular rate between adjacent members. From the definition of $[R']$ it follows that this component of torque in the coordinates of body I is

$$[R'_J](\underline{\omega}'_K - \underline{\omega}_I) \quad ; \quad \underline{\omega}'_K \triangleq [V']\underline{\omega}_K \tag{A-6}$$

and (recalling from Section 1.1 that $[R]$ and $[R']$ are referenced to body I) the total hinge moment acting on body I is

$$\underline{\mathfrak{T}}'_I = \lambda[R_J]\underline{U} + [R'_J](\underline{\omega}'_K - \underline{\omega}_I) \tag{A-7}$$

*In many applications this first approximation will be inadequate; Appendix B illustrates a refined approximation method for ($d\theta_B/d\ell$) which was successfully applied to the RAE program.

and the hinge moment on body K is

$$\underline{\mathcal{L}}_K' = -[V']^T \underline{\mathcal{L}}_I' \tag{A-8}$$

This completes the discussion for this portion of the program. Before leaving this topic it is noted that (1) various nonlinear functions of the deformation ($\lambda \underline{U}$) and/or the relative rate ($\underline{\omega}_K' - \underline{\omega}_I$) could easily be programmed, to simulate nonlinear reaction torque characteristics encountered in practice; and (2) delayed feedback torque supplied from band-limited devices could be computed by standard convolution integral techniques.

Thermal Bending

Gravity gradient satellites often employ long narrow booms which are prone to nonuniform heating. A convenient way to take this into account is to replace the zero torque rest position matrix $[\rho_J]$ for each hinge by a new matrix $[\rho_J']$ which defines the reference shape under uneven heating conditions. This new matrix can be formed by a simple orthogonal transformation of its original value,

$$[\rho_J'] = [-\delta_Z]_Z [+\delta_Y]_Y [\rho_J] \tag{A-9}$$

where the bending angles (δ_Y) and (δ_Z) are identified by combining this expression with equation (A-1):

$$[V] = [-\delta_Z]_Z [+\delta_Y]_Y [\rho_J] [V']^T$$

Structural deformation is zero when $[V]$ is the identity matrix. Since $[V']^T$ is the transformation from actual I to K co-ordinates and $[\rho_J]$ is the transformation from K to original reference axes of I, it follows that the product $[-\delta_Z]_Z [+\delta_Y]_Y$ must be the transformation from the original (undeformed) to the new rest position (zero torque) axes of body I. This angular displacement of the reference orientation for each segment in the discrete model is obtained by inscribing a set of chords (n = number of segments per boom) inside the arc formed by thermal bending. The present description will begin with an example of planar bending caused by a single heat source, followed by extension to the general case.

In Fig. 1, $\mathbf{J}$ denotes the hinge connecting body I (represented by the chord $\mathbf{JJ'}$) to body K, which may be visualized on the left of the hinge.

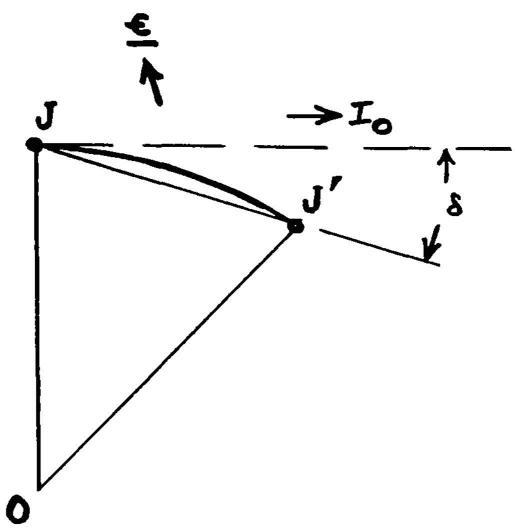

Fig. 1 Effect of Thermal Bending

The reference orientation of the I segment in the absence of heating effects will be taken along the direction of the arrow I_0. Thus δ is the bending angle due to a heat source in the direction $\underline{\epsilon}$, tentatively defined in the plane of the figure.

Under the above conditions the arc $\mathbf{JJ'}$ is essentially circular* with center at O and radius of curvature R_c meters. With (ℓ) again defined as the distance between centers of the members joined by hinge $\mathbf{J}$, it is seen that the average change in angle per unit length is ($\delta/\ell = 1/R_c$) and, combined with Eq. (4) of Ref. 5,

$$\delta = \frac{e\ell}{d}(\Delta T) \qquad \text{(A-10)}$$

*Just as in the preceding section, this small angle approximation is valid provided that the model contains a sufficient number of segments.

where e , d , and (ΔT) denote the linear thermal coefficient of expansion $(^\circ C)^{-1}$, boom diameter (meters), and diametric temperature differential (°C), respectively. With the unit vector along the segment x (length) axis JJ' denoted by $\underline{1}_1$ it is seen that a direction* for δ can be defined by the unit vector $\left\{ \underline{\epsilon} \times \underline{1}_1 / |\underline{\epsilon} \times \underline{1}_1| \right\}$. By reason of the small angle approximation for adjacent segments, then, the finite rotation can be treated as a vector,

$$\underline{\delta} = \frac{e\ell}{d}(\Delta T)\frac{\underline{\epsilon} \times \underline{1}_1}{|\underline{\epsilon} \times \underline{1}_1|} \tag{A-11}$$

Since $|\underline{\epsilon} \times \underline{1}_1|$ is equivalent to the cosine of the angle between $\underline{\epsilon}$ and the normal to the segment, Eq. (6) of Ref. 5 can be written here as

$$\frac{\Delta T}{|\underline{\epsilon} \times \underline{1}_1|} = \frac{d^2}{4K\tau}(aq) \tag{A-12}$$

and, therefore,

$$\underline{\delta} = \frac{e\ell d}{4K\tau}(aq)(\underline{\epsilon} \times \underline{1}_1) \tag{A-13}$$

in which K , τ , a , and q represents thermal conductivity (large calories/second-meter-°C), boom thickness (meters), and the absorptivity and heat radiation (large calories/second-meter^2) of the source, respectively. The convenience of this formulation is apparent when different heat sources are combined; with direct earth radiation and direct plus reflected solar radiation (at an albedo[6] of 0.4), the unit vectors in the source directions are denoted as

*The cross product conforms to the definition of $[\delta]$ as the transformation from the original to the deformed rest position of segment I.

$$\underline{\epsilon}_{Solar} = \begin{bmatrix} \sigma_{I1} \\ \sigma_{I2} \\ \sigma_{I3} \end{bmatrix} \quad ; \quad \underline{\epsilon}_{Earth} = \begin{bmatrix} -\theta_{I31} \\ -\theta_{I32} \\ -\theta_{I33} \end{bmatrix}$$

and the thermal deflection angles are computed from resultants thus: For the y-axis, the right of Eq. (A-13) is written with the substitutions,

I. Direct Earth

1) $a = a_E$, absorptivity for earth radiation.

2) $q = \mathcal{F} J_E$, where $\mathcal{F}$ is computed from the earth radius (R_E) and the Keplerian orbital distance (r) as[7]

$$\mathcal{F} = 2\left[1 - \sqrt{1 - (R_E/r)^2}\right] \qquad \text{(A-14)}$$

and J_E is the Stefan-Boltzmann constant ($5.67 \times 10^{-8}/4184$ large calories per sec per sq. meter per $°K^4$) multiplied by the fourth power of the effective[8] spherical blackbody Earth temperature (246°K):

$$J_E = \frac{5.67 \times 10^{-8}}{4184} (246)^4 \qquad \text{(A-15)}$$

3) $[\underline{\epsilon} \times \underline{1}_1] \cdot \underline{1}_2 = -\theta_{I33}$

II. Direct Solar

1) $a = a_S$, absorptivity for solar radiation.

2) $q = J_S (I_S - 1)$, where

$$I_S = \begin{cases} 2, & \text{sun not eclipsed} \\ 1, & \text{sun eclipsed} \end{cases}$$

and J_S is the product

$$J_S = P_S \, c/4184 \tag{A-16}$$

with c defined as the speed of light (3×10^8 m/sec) and

$$P_S = 4.5 \times 10^{-6} \; (\text{Newt.}/m^2) \tag{A-17}$$

3) $[\underline{\epsilon} \times \underline{1}_1] \cdot \underline{1}_2 = \sigma_{I3}$

III. Reflected Solar

1) $a = a_S$

2) $q = .4 \, \mathcal{F} J_S (I_S - 1)$, where all of these quentities are defined above. It is noted that $\mathcal{F}$ is not the true coefficient to be used for reflected radiation, but it provides an excellent approximation.[7]

3) $[\underline{\epsilon} \times \underline{1}_1] \cdot \underline{1}_2 = -\theta_{I33}$

The total rotation about the y-axis due to thermal bending is therefore

$$\delta_y = -\mathcal{F} J_E' \theta_{I33} + J_S' (\sigma_{I3} - .4 \mathcal{F} \theta_{I33})(I_S - 1) \tag{A-18}$$

where

$$J_E' = \frac{e \ell d a_E}{4 K \xi} J_E \; ; \qquad J_S' = \frac{e \ell d a_S}{4 K \xi} J_S \tag{A-19}$$

and, similarly, thermal bending about the z-axis is obtained by the negative $\{[\underline{\epsilon} \times \underline{1}_1] \cdot \underline{1}_3 = -\epsilon_2\}$ angle transformation in Eq. (A-9) with

$$\delta_z = -\mathcal{F} J_E' \theta_{I32} + J_S' (\sigma_{I2} - .4 \mathcal{F} \theta_{I32})(I_S - 1) \tag{A-20}$$

Thermal bending throughout the entire structure is accounted for by repeating these computations at every applicable hinge. For example, in the case of a nominally straight boom, the reference direction for the segment to the right of J' is the extended line JJ'; this segment is then denoted as I in computing the bending angles at J', while the chord JJ' takes the role of segment K. The preceding formulation then applies to the computation of angles at J', and likewise to all hinges where thermal bending can occur.

As a further refinement is is noted that the matrix $[-\delta_z]_z[+\delta_y]_y$ can be replaced by

$$[\delta] = \begin{bmatrix} \Upsilon & -\delta_z & -\delta_y \\ \delta_z & \dfrac{\delta_y^2 + \Upsilon\delta_z^2}{\delta_y^2+\delta_z^2} & \dfrac{\delta_y\delta_z(\Upsilon-1)}{\delta_y^2+\delta_z^2} \\ \delta_y & \dfrac{\delta_y\delta_z(\Upsilon-1)}{\delta_y^2+\delta_z^2} & \dfrac{\delta_z^2+\Upsilon\delta_y^2}{\delta_y^2+\delta_z^2} \end{bmatrix}; \; \Upsilon \triangleq \sqrt{1-\delta_y^2-\delta_z^2} \qquad \text{(A-21)}$$

It has been verified that this matrix is orthogonal and that the x-component of its real eigenvector is zero.

Constraint Torques at Locked Joints

In many instances there are hinges which are constructed to allow only one or two degrees of freedom, or it may be desirable to remove a degree of freedom from the computation model.* When a hinge constrains relative motion about the α-axis ($\alpha = 1, 2, 3$ for x, y, z respectively) of a given member (I) of the structure, the α-component of the relative angular rate between body I and K, expressed in I-coordinates, is set to zero:

*e.g., if the torsional natural frequencies of a gravity gradient boom are very high, it may be convenient to assume that only flexure is possible.

$$\underline{1}_\alpha^T \left\{ \underline{\omega}_I - [\theta_I]^T [\theta_K] \underline{\omega}_K \right\} = 0 \tag{A-22}$$

As noted in Ref. 2, the first step in deriving the constraint torque is to differentiate the above expression. It is shown in a later section of this Appendix that

$$d/dt \; [\theta_I] = [\theta_I][\Omega_I] - [N][\theta_I] \tag{A-23}$$

(where $[\Omega]$ and $[N]$ are skew-symmetric angular rate matrices), so that

$$\underline{1}_\alpha^T \dot{\underline{\omega}}_I - \underline{1}_\alpha^T \Big\{ \Big([\Omega_I]^T [\theta_I]^T - [\theta_I]^T [N]^T \Big) [\theta_K] \underline{\omega}_K + \tag{A-24}$$
$$[\theta_I]^T \Big([\theta_K][\Omega_K] - [N][\theta_K] \Big) \underline{\omega}_K + [\theta_I]^T [\theta_K] \dot{\underline{\omega}}_K \Big\} = 0$$

This is simplified by accounting for 1) the skew-symmetric character of $[\Omega]$ and $[N]$, and 2) the property $[\Omega_K] \underline{\omega}_K = \underline{0}$; introducing the previously defined notations $[V^r]$ and $\underline{\omega}_K^r$,

$$\underline{1}_\alpha^T \left\{ \dot{\underline{\omega}}_I - [V^r] \dot{\underline{\omega}}_K \right\} = - \underline{1}_\alpha^T [\Omega_I] \underline{\omega}_K^r \tag{A-25}$$

There will be a scalar equation of this form for every combination (J , α) which represents a locked joint constraint. It is therefore appropriate to define an identifying argument ($\dot{j}$) for each locked mode,

$$\dot{j} = 3(J - 1) + \alpha \tag{A-26}$$

plus a column vector $\underline{m}$ having $3(N-1)$ components such that $m_{\dot{j}}$ is zero for all unlocked modes and, for locked modes, $m_{\dot{j}}$ is an integer representing the ordered index of that mode ($1 \leq m_{\dot{j}} \leq N_C$, where N_C is the total number of locked modes). The set of equations (A-25) can then be written in matrix form

$$[\boldsymbol{\xi}]^T \underline{\dot{\omega}} = -\underline{\nu} \tag{A-27}$$

where $[\boldsymbol{\xi}]$ is a ($3N \times N_C$) matrix in which the only nonzero elements are the unit vector components

$$\underline{\xi}_{I,m_j} = \underline{1}_\alpha^T \; ; \; \underline{\xi}_{K,m_j} = -\underline{1}_\alpha^T [V'] \tag{A-28}$$

in the m_j row and the $I^{\underline{th}}$ and $K^{\underline{th}}$ triplet of columns respectively, I and K representing the bodies constrained by the m_j locked mode. The vector $\underline{\dot{\omega}}$ in (A-27) is the same angular acceleration vector appearing in the Roberson-Wittenburg equation; and $\underline{\nu}$ is a vector ($N_C \times 1$) whose m_j component is

$$\nu_{m_j} = \underline{1}_\alpha^T [\Omega_I] \underline{\omega}_K' \tag{A-29}$$

where (α, I, K) of course correspond to the locked mode under consideration.

At this point, Eq. (6) is premultiplied by $[\xi]^T[\Gamma]^{-1}$ and combined with (A-27):

$$\underline{T}_c = -\left\{[\xi]^T[\Gamma]^{-1}[\xi]\right\}^{-1}\left\{[\xi]^T[\Gamma]^{-1}\underline{E} + \underline{\nu}\right\} \tag{A-30}$$

so that the final differential equation is

$$[\Gamma]\underline{\dot{\omega}} = \underline{E} - [\xi]\left\{[\xi]^T[\Gamma]^{-1}[\xi]\right\}^{-1}\left\{[\xi]^T[\Gamma]^{-1}\underline{E} + \underline{\nu}\right\} \tag{A-31}$$

Radiation Pressure

The present formulation involves the effective solar radiation force on each member of the structure, expressed in the co-ordinates of that member. Although the pertinent theory is well-known, an example of a typical boom segment (again denoted here as "body I") is treated here to illustrate application to the problem at hand.

It is convenient to begin by considering a flat surface of area A subjected to a radiation pressure from a source along the unit sunline vector $\underline{\sigma}_I$ (see Eqs. A-50 —— A-54). With the force per unit area of A denoted by (p), the component of the effective force along $\underline{n}_A$ due to incident radiation has a magnitude $pA\left|\underline{\sigma}_I\cdot\underline{n}_A\right|$, where $\underline{n}_A$ is the unit normal to the surface A. With perfect reflection, the total force due to incident plus reflected radiation is directed along ($-\underline{n}_A$) and has a magnitude

$$\left|F_A\right| = 2pA\left|\underline{\sigma}_I\cdot\underline{n}_A\right| \tag{A-32}$$

While (p) is defined as the force per unit area of A, it is more convenient to work with the characteristics of the source itself; it is easily shown that

$$p = P_S\left|\underline{\sigma}_I\cdot\underline{n}_A\right| \tag{A-33}$$

where P_S is defined by Eq. (A-18). It follows that

$$\left|F_A\right| = 2P_S\left(\underline{\sigma}_I\cdot\underline{n}_A\right)^2 A \tag{A-34}$$

Application of this theory to nonplanar surfaces is straightforward in principle and, for regular geometry, often leads to simple solutions. Each

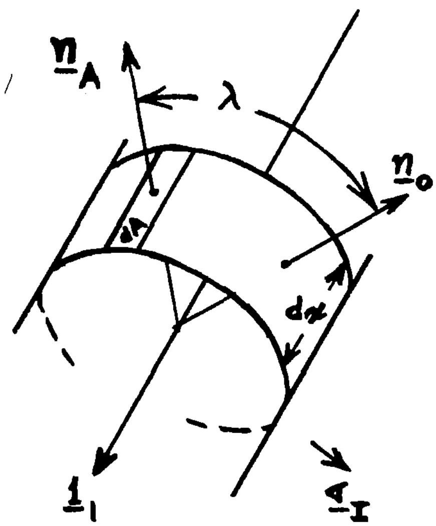

Fig. 2 Radiation Pressure on Section of Cylinder

section of a gravity gradient boom, for example, can be represented by a cylinder, centered about the boom torsion axis $\underline{1}_1$. Fig. 2 shows a small section of width (dx) with a principal normal (defined as the normal to the surface extending outward from the center of dx , and lying in the plane of $\underline{1}_1$ and $\underline{\sigma}_I$) along the direction $\underline{n}_o$. It is permissible to consider the unit sunline vector $\underline{\sigma}_I$ as originating from the intersection point of $\underline{1}_1$, $\underline{n}_o$, and $\underline{n}_A$, so that the spherical law of cosines can be invoked:

$$(\underline{\sigma}_I \cdot \underline{n}_A) = (\underline{\sigma}_I \cdot \underline{n}_o) \cos \lambda \tag{A-35}$$

where the significance of $\underline{n}_A$ of course lies in its orthogonality to the differential surface area

$$dA = \frac{d}{2}\, d\lambda\, dx \tag{A-36}$$

in which (d) represents boom diameter. Preparations are now complete for integrating the force over the cylindrical area. It is first noted that the component along $\underline{1}_1$ vanishes in the present problem because the resultant must be normal to the surface; furthermore the component along ($\underline{1}_1 \times \underline{n}_o$) vanishes by symmetry. For the differential area dA the component of force along ($-\underline{n}_o$) is ($dF_A \cos \lambda$); it is this quantity which must be integrated using

(A-34) through (A-36):

$$F_{cyl} = 2P_S(\underline{\sigma}_I\cdot\underline{n}_0)^2\left(\frac{d}{2}\right)\int_0^{l}\int_{-\pi/2}^{\pi/2}\cos^3\lambda\, d\lambda\, dx \qquad \text{(A-37)}$$

It can easily be shown that this is equivalent to $2P_S A_E(\underline{\sigma}_I\cdot\underline{n}_0)^2$ where A_E is the cylindrical effective area,

$$A_E = \frac{2}{3}\, l\, d \qquad \text{(A-38)}$$

Since the direction of the effective force is along the unit vector

$$-\underline{n}_0 = \frac{\underline{1}_1\times(\underline{1}_1\times\underline{\sigma}_I)}{\left|\underline{1}_1\times(\underline{1}_1\times\underline{\sigma}_I)\right|} = \frac{-1}{\sqrt{\sigma_{I2}^2+\sigma_{I3}^2}}\begin{bmatrix}0\\ \sigma_{I2}\\ \sigma_{I3}\end{bmatrix} \qquad \text{(A-39)}$$

the solar force vector expressed in I-co-ordinates is

$$\underline{F}_{cyl} = \begin{bmatrix}0\\ -\sigma_{I2}\\ -\sigma_{I3}\end{bmatrix}(2P_S)\sqrt{\sigma_{I2}^2+\sigma_{I3}^2}\; A_{E(I)} \qquad \text{(A-40)}$$

By a derivation along similar lines it can be shown that the effective radiation force vector for a sphere is

$$\underline{F}_{sph} = -\underline{\sigma}_I\,(2P_S)\,A_{E(sph)} \qquad \text{(A-41)}$$

where the effective area for a sphere of radius (l_1) is

$$A_{E(sph)} = \frac{1}{2}\pi l_1^2 \qquad \text{(A-42)}$$

Presence of these forces must of course be subject to the condition of no eclipse.

Kinematics

In this section the orthogonal matrices $[B]$; $[\theta_I]$; and $[D]$ will denote transformations from principal axes of a structural member (body I) to a set of inertial axes; from the body axes to a set of local axes; and from the local to the inertial co-ordinates, respectively. It follows immediately that

$$[B] = [D][\theta_I] \tag{A-43}$$

and it is well-known that

$$d/dt\ [B] = [B][\Omega_I] \tag{A-44}$$

where $[\Omega_I]$ is a skew-symmetric matrix of inertial angular rates ($\Omega_{I,12} = -\omega_{I3}$; $\Omega_{I,13} = +\omega_{I2}$; $\Omega_{I,23} = -\omega_{I1}$). Defining the local axes by ($+y$) along the orbit pole and ($+z$) along the upward local vertical,

$$d/dt\ [D] = [D][N] \tag{A-45}$$

where $[N]$ is a 3 x 3 matrix whose only nonzero elements appear in the positions ($N_{13} = -N_{31}$) representing the time derivative of the true anomaly; from the appropriate equation on page 262 of Ref. 9 it can be deduced that

$$N_{13} = \sqrt{\mu_E p_0}/r^2 \tag{A-46}$$

where $p_0 = a_0(1 - e_0^2)$; μ_E , r , a_0 , and e_0 are defined as the Earth gravitational constant; the Keplerian orbit position vector magnitude, semimajor axis, and eccentricity, respectively.

By differentiation of (A-43),

$$d/dt\ [\theta_I] = [D]^T\ d/dt\ [B] + \{d/dt\ [D]\}^T [B] \tag{A-47}$$

and, in combination with (A-43) through (A-45),

$$\frac{d}{dt}\left[\theta_I\right] = \left[\theta_I\right]\left[\Omega_I\right] + \left[N\right]^T\left[\theta_I\right] \tag{A-48}$$

To compute the angular rate for the matrix $[N]$ it is noted that gravity gradient satellites generally have low eccentricity (e.g., less than 0.1). The quantity ($\mathcal{n}$) in (A-46) can therefore be determined explicitly by a series approximation on page 153 of Ref. 9; using the notation (A_m) for the mean anomaly,

$$\mathcal{n} = a_0\Big[1 - e_o \cos A_m - \tfrac{1}{2} e_o^2 \left(\cos 2A_m - 1\right) - \tfrac{1}{8} e_o^3 \left(3 \cos 3A_m - 3 \cos A_m\right)\Big] \tag{A-49}$$

Astronomical Geometry

Solar position is determined using the celestial sphere model on page 9 of Ref. 9. A value of 23.5° is used as the ecliptic inclination and, on the ($N_D^{\underline{th}}$) day of the year, the sunline vector makes an angle of

$$\Psi_S = 2\pi\left(N_D - 80\right)/365 \tag{A-50}$$

with the vernal equinox. The sunline vector expressed in inertial (celestial sphere) co-ordinates is therefore

$$\underline{\sigma}'' = \begin{bmatrix} \cos \Psi_S \\ \cos 23.5^\circ \sin \Psi_S \\ \sin 23.5^\circ \sin \Psi_S \end{bmatrix} \tag{A-51}$$

To define this vector in local co-ordinates the inclination, longitude of the ascending node, and argument of the perigee for the satellite orbit are written as (i_o, Ω_o, ω_o) respectively, and the true anomaly is computed explicitly by the series approximation on page 154 of Ref. 9:

$$\nu = A_m + 2e_o \sin A_m + \frac{5}{4} e_o^2 \sin 2A_m + \frac{1}{12} e_o^3 (13 \sin 3A_m - 3 \sin A_m) \tag{A-52}$$

which is again accurate for most gravity gradient satellite applications ($e_o \leq 0.1$). The sunline in local co-ordinates is therefore

$$\underline{\sigma}' = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} [\omega_o + \nu]_z [i_o]_x [\Omega_o]_z \underline{\sigma}'' \tag{A-53}$$

The unit vector pointing toward the sun, expressed in body I-axes, is then obviously,

$$\underline{\sigma}_I = [\theta_I]^T \underline{\sigma}' \tag{A-54}$$

and it is this vector which is used for the appropriate thermal bending and radiation pressure computations.

This completes the description of analytical formulations used in the gravity gradient satellite program. The next Appendix describes an example configuration (RAE satellite[4]).

APPENDIX B

THE RADIO ASTRONOMY EXPLORER (RAE) SATELLITE CONFIGURATION AND PARAMETERS

The RAE satellite described in Ref. 4 is cruciform shaped, passively damped with a horizontal libration damper boom skewed out of the plane of the cruciform.[10] The typical RAE example configuration is shown in Fig. 3 with the flexible antenna booms approximated dynamically with 3 rigid segments per boom. The four 750 foot antenna booms in their undeformed state make an angle of 30 degrees with the $\underline{Z}_{HUB}$ axis in the cruciform plane. The damper boom (assumed rigid) is spring restrained to its reference position relative to the hub, and is skewed at an angle of 65 degrees from the cruciform plane. Each rigid body or member and each hinge is indexed as shown in figure 3. The hinge numbers are denoted by an underline. For a larger number of segments per boom, the same counter-clockwise numbering scheme would be used.

Each member has a body-fixed set of right-hand coordinates defined collinear with the principal inertia axes, and the origin is the mass center of each member. Only the direction of two axes need be given for the right hand coordinate frame to be well defined. The hub axes are shown in Fig. 3. THE FOLLOWING COORDINATE FRAMES OF THE REMAINING MEMBERS ARE DEFINED WITH THE SATELLITE IN ITS UNDEFORMED CONFIGURATION, and it should be remembered that the axes remain fixed in each member even after relative rotational displacement between them:

The damper boom $\underline{Z}$ and $\underline{X}$ axes are in the direction of $\underline{Z}_{HUB}$ and the outward directed damper boom centroidal axis respectively. In each antenna boom segment the $\underline{Y}$ and $\underline{X}$ axes are in the direction of $\underline{Y}_{HUB}$

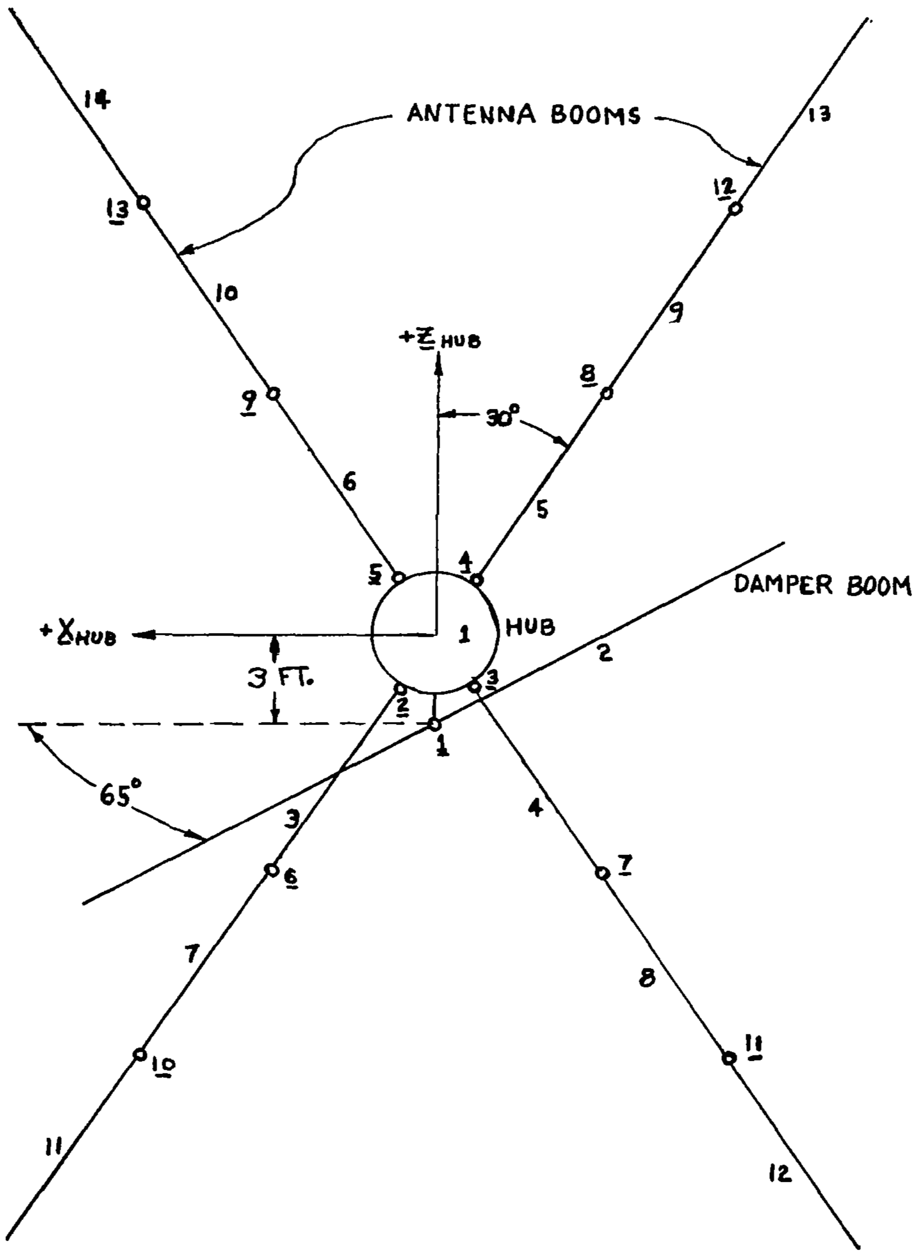

Fig. 3 RAE Configuration

and the outward directed antenna segment centroidal axis respectively. For radiation pressure effective area computations the hub is assumed spherical in shape, while the damper boom and all antenna segments are cylindrical. It is noted that the damper boom is free to rotate about its $\underline{Y}$ axis only.

All parameters in the RAE simulation are expressed in MKS units. However, since many RAE satellite properties are given in English units, appropriate conversion factors are listed below. Following this is a description of RAE variable inputs (Part I), fixed and derived inputs (Part II), and inputs to the general program (Parts III and IV) as defined in the INTRODUCTION of this report*. Included also are the formulations for initial conditions, magnetic hysteresis damping torque, and readouts.

<u>MKS Units</u>

Length - Meters (m)

Mass - Kilograms (Kg)

Temperature - Degrees Centigrade (°C)

Heat - Large Calories (K-cal; i.e., the amount of heat required to raise the temperature of 1 KG-H_2O by 1°C.)

Force - Newtons (Newt)

<u>Conversion Factors</u>

.3048m = 1 ft.

14.5939 Kg = 1 slug

1 Newt = .2248 lbs.

4184 Kcal = 1 Newt-m = 1 Joule

*Along with the mathematical symbol, the actual Fortran statement as used in the program is given in quotation.

RAE VARIABLE INPUTS (PART I)

System Parameters

n ("NPB") Number of segments per boom ($1 \leqslant n \leqslant 6$).

S_d ("SD1") Damper spring constant ratio. This ratio is defined as the damper hinge restraining spring constant $R_{(1,2,2)}$ divided by the corresponding "gravity gradient spring constant":

$$S_d \triangleq R_{(1,2,2)} \Big/ \left[(3+\sin^2\delta)\, I_{(2,2,2)} \left(\mu_E / a_0^3\right)\right] \tag{B-1}$$

where (μ_E) and (a_0) are defined after Equation (A-46); the angle (δ) is ($\frac{65\pi}{180} - \gamma$), where (γ) is the equilibrium yaw angle defined later in this Appendix. S_d should be greater than unity so that the damper boom will seek a horizontal rest position.

f_d ("FD") Linear damping or non-linear hysteresis* type damping option at the damper boom hinge. To simulate hysteresis damping, set $f_d = 0$. To simulate linear damping set f_d equal to the desired damping ratio, defined as

$$f_d \triangleq R'_{(1,2,2)} \Big/ \left[2\, I_{(2,2,2)} \sqrt{(3+\sin^2\delta)(S_d-1)\left(\mu_E / a_0^3\right)}\,\right] \tag{B-2}$$

It is noted that this conforms to the standard expression for a damping ratio, with the stiffness term defined as the combined effect of the spring restraint and gravity gradient.

*The hysteresis damper simulation is described later in this Appendix.

N_L ("NL") Locked mode option. Any or all three degrees of freedom at each hinge may be eliminated, so long as the total number of locked modes $N_C \leqslant 38$. However, only three locked mode configurations are included in the RAE simulation:

- $N_L = 1$ Eliminates the rotational degree of freedom about the x and z axes of the damper boom relative to the hub.
- $N_L = 2$ Includes $N_L = 1$ and also locks the torsional degree of freedom at hinges on all antenna boom segments.
- $N_L = 3$ Includes $N_L = 1$ and also locks the three degrees of freedom at the base of each antenna boom. (In this mode, $n = 1$; this simulates a rigid cruciform with a single degree of freedom damper boom).

I_T ("ITHERM") Thermal bending option. To include effects of thermal bending, set $I_T = 1$; otherwise set $I_T = 0$.

N_A ("NA") Solar pressure option. To include effects of solar pressure, set $N_A = 1$; otherwise set $N_A = 0$.

Initial and Final Conditions

X ("XINIT(I)" I= 1,...,12) Twelve initial amplitudes of fundamental RAE satellite librational and flexing modes as defined later in this Appendix.

T ("ORBS") Total number of orbits to be simulated.

N_R ("ENR") Number of readouts* printed per orbit.

Astronomical Parameters

e_o ("EZ") Eccentricity of orbit

i_o ("EYZ") Inclination angle between the normal to the orbital plane and the north geodetic pole of the celestial sphere.

Ω_o ("THZ") Longitude of the ascending node measured from the Vernal Equinox.

*Readout format is described later in this Appendix.

RAE FIXED AND DERIVED INPUTS (PART II)

Hub and Boom Parameters

m_1	("EM(1)")	Mass of hub (Kg.)
ℓ_1	("EL1")	Effective geometric radius of hub (m.)
$I_{1\alpha\alpha}$	("A(1,α,α)") $\alpha = 1,2,3$	Moments of Inertia about hub X, Y, and Z principal axes respectively (including two dipoles $\{$ 18.3 meter length $\}$ along hub X-axis)
m_B	("EMB")	Boom mass per unit length (Kg./m.)
ℓ_B	("ELB")	Total length of each antenna boom (m.)
ℓ	("SL")	Length of each segment, $\ell = \ell_B / n$
ℓ_d	("ELD")	Length of damper boom (m.)
d	("DIA")	Boom cross-section diameter (m.)
τ	("THK")	Boom wall thickness (m.)
ψ	("OLA")	Boom wall overlap half angle[11] (rad.)
ν	("POR")	Poisson's Ratio
E	("E")	Modulus of Elasticity (Newt./m.^{2})
f_α	("F(α)") $\alpha = 2,3$	Boom area moments of inertia about the diametral axes parallel to the boom segment Y and Z axis respectively*

$$f_2 = \frac{d^3 \tau}{8}\left[\pi + \psi + \sin\psi\cos\psi - \frac{2\sin^2\psi}{\pi+\psi}\right] \qquad \text{(B-3)}$$

$$f_3 = \frac{d^3 \tau}{8}\left[\pi + \psi - \sin\psi\cos\psi\right] \qquad \text{(B-4)}$$

*Boom principal axes are in the directions defined in the beginning of this Appendix. The values of f_2 and f_3 are given in Ref. 11 for open slit tubes, but are also valid for any tube with overlap.

f_1 ("F(1)") Multiplicative constant for torsional rigidity (locked slit tube):

$$f_1 = \frac{0.5}{1+\nu}\left(f_2 + f_3\right) \qquad \text{(B-5)}$$

K ("CK") Thermal conductivity of boom (Kcal/m.sec.°C)

e ("CTE") Temperature coefficient of linear expansion for boom $(°C)^{-1}$

A_W ("AW") Ratio of perforation area to total surface area of the booms.

J_E ("XJE") Earth heat flux density, as given in Eq. (A-15); (Kcal./sec.m^2).

J_S ("XJS") Solar heat flux density, as given by Eq. (A-16); (Kcal./sec.m^2).

a_E ———— Boom absorbtivity to Earth radiation. (For the RAE simulation set equal to 0.1 in equation A-19).

a_S ———— Boom absorbtivity to Solar radiation. (For the RAE simulation set equal to 0.05 in equation A-19).

Equilibrium Parameters

K_A, K_B ("QKA", "QBA") First cantilever mode antenna boom tip deflection in and out of the undeformed cruciform plane respectively (m.)

γ ("QGAM") Static yaw angle about hub vertical axis due to skewed damper boom.

These equilibrium parameters, explained in Ref. 12, are used in the computation of initial conditions and readouts described later in this Appendix.

Astronomical Parameters

a_o ("AZ") Orbital semi-major axis (m.)

ω_o ("WZ") Argument of perigee (deg.)

t_o ("TZ") Time at perigee of orbit (sec.). NOTE: $t \triangleq 0$ at start of simulation.

N_D ("ND") Vehicle launch date.

General Program Inputs

Following is a list of System Parameters plus Initial and Final Conditions, as defined in the INTRODUCTION of this report, applicable to the RAE satellite. The Astronomical Parameters, also called for in the INTRODUCTION, are the same as those just defined.

$\mathbf{N}$ ("N") Total number of rigid members, $N = 4n + 2$

$\mathbf{m}$ ("EM(I)") Mass vector with elements m_i , $1 \leq i \leq N$ (i[th] body).

m_1 = hub mass

$m_2 = \ell_d m_B$

$m_i = \ell m_B$ for $3 \leq i \leq N$

$\mathbf{I}$ ("A(I,α,β)") Inertia tensor with elements $I(i, \alpha, \beta)$, $1 \leq i \leq N$; $1 \leq \alpha, \beta \leq 3$:

$I(i, \alpha, \beta) = 0, \quad \alpha \neq \beta$

$I(1, \alpha, \alpha)$ = hub principal inertias*

$$I(i, 1, 1) = m_i \ell^2/4 \quad , \; 2 \leq i \leq N \tag{B-6}$$

$$I(2, \alpha, \alpha) = m_2 \ell_d^2/12 \quad , \; \alpha = 2, 3 \tag{B-7}$$

$$I(i, \alpha, \alpha) = m_i \ell^2/12 \quad , \; 3 \leq i \leq N \; ; \; \alpha = 2, 3 \tag{B-8}$$

$\mathbf{R}$ ("R(J,α,β)") Hinge spring constant tensor (See Eqs. A-7 and A-8):

$R(j, \alpha, \beta)$, $1 \leq j \leq (N-1)$; $1 \leq \alpha, \beta \leq 3$:

$R(j, \alpha, \beta) = 0 \; , \; \alpha \neq \beta$

$R(1,1,1) = R(1,3,3) = 0$

$$R(1,2,2) = (3 + \sin^2 \delta) S_d \left(\mu_E / a_0^3\right) I_{(2,2,2)} \tag{B-9}$$

$$R(j, \alpha, \alpha) = (1 - A_W) E f_\alpha / \ell \; , \; 2 \leq j \leq (N-1) ; 1 \leq \alpha \leq 3 \tag{B-10}$$

*A considerable saving in machine time was realized by increasing these values in the computational model. Ref. 12 shows that only the highest frequency (lowest amplitude) oscillations are affected.

At this point a brief digression is in order, to explain certain subtle aspects of structural discretization. First it is noted that high frequency torsional oscillations (which consume excessive machine time in simulation) can be circumvented through the previously defined input (N_L). When $N_L \neq 1$ all torsion axes are locked; thus (B-10) is applied for $\alpha = 2$ and 3 only. When $N_L = 3$ all antenna joints are locked and (B-10) is bypassed completely. Theoretically the rigidity constants for all locked modes should be zero, since Eq. (A-30) provides the necessary constraint torque. In practice, however, small computational imperfections in the value of this torque are doubly integrated with the dynamical equations. A weak spring and damper have been placed in the locked torsional joints, to counteract this cumulative effect.

In connection with bending moments, Eq. (A-4) was accompanied by a statement that the rate of change of slope $(d\theta_B/d\ell)$ cannot in general be adequately determined from a single hinge angle. The inherent accuracy limitations of numerical differentiation can, however, be minimized by using a properly weighted sum to compute derivatives at each point. For the RAE program this was accomplished through augmenting the standard internal torque computation* as follows: the vector $\{\lambda \underline{U}\}$ at every antenna boom hinge (J_S) is transformed into the co-ordinates of each interacting member (I_A); the result, multiplied by the appropriate rigidity matrix $[R]$ and weighting constant, is included in the total internal moment acting on that member. The members which may interact with

*The exact form of this refinement will vary with the particular structural topology, but the method exemplified here will be useful for a wide range of applications. It is noted that all weighting constants are set to zero in the Initial Program Setup, (Part 0). Therefore, in the absence of any subsequent introduction of weights in Part I or II, the internal torque modifications in Part IV of the General Program will not invalidate the standard formulation (Eqs. A-7 and A-8) applicable to truly isolated hinges.

any given hinge are the hub and the segments in the same quadrant as the hinge. The extent of interaction is determined from Newton's divided difference formula;[13] a three-segment planar model of an antenna boom will serve to illustrate the technique below.

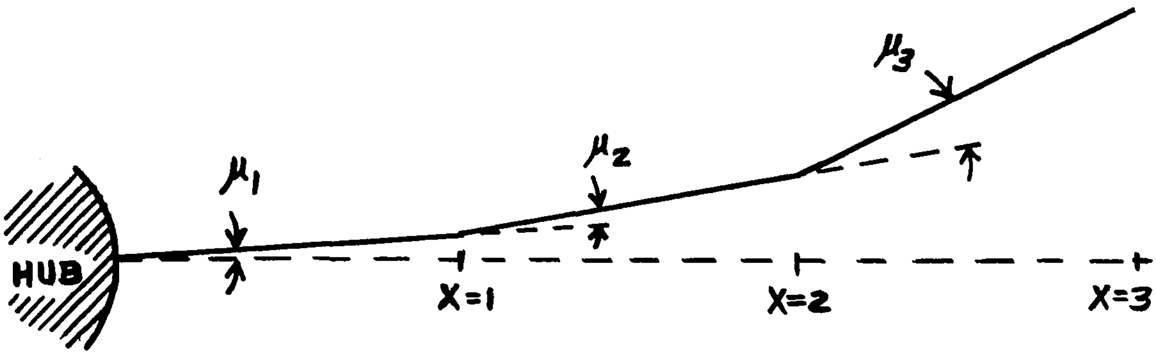

Fig. 4. Segmented Antenna Boom

Normalized Arc Length X	Angle off Base Tangent θ_B	1st Divided Difference $f(X_0, X_1)$	2nd Divided Difference $f(X_0, X_1, X_2)$	3rd Divided Difference $f(X_0, X_1, X_2, X_3)$
0	0			
		$2\mu_1$		
1/2	μ_1		$\frac{2}{3}(\mu_2 - 2\mu_1)$	
		μ_2		$\frac{1}{5}\mu_3 - \frac{7}{15}\mu_2 + \frac{8}{15}\mu_1$
3/2	$\mu_1 + \mu_2$		$\frac{1}{2}(\mu_3 - \mu_2)$	
		μ_3		$-\frac{7}{15}\mu_3 + \frac{1}{5}\mu_2$
5/2	$\mu_1 + \mu_2 + \mu_3$		$-\frac{2}{3}\mu_3$	
		0		
3	$\mu_1 + \mu_2 + \mu_3$			

In the accompanying table, attention is first drawn to the first two columns. The hinge angles (μ) can represent a Y or Z axis component of ($\lambda \underline{U}$) in the present problem. For a boom cantilevered to the hub (θ_B) is zero at the base and, since there should be no bending moment at the free end, (θ_B) should not change at $X = 3$. In addition to the known values of (θ_B) for each segment (presumably the centers), then, these two boundary conditions can be used to determine the derivative of (θ_B). Differentiating Eq. (17) of Ref. 13,

$$f'(X) = f(X_0, X_1) + [2X - (X_0 + X_1)]\, f(X_0, X_1, X_2) +$$
$$[3X^2 - 2(X_0 + X_1 + X_2)X + (X_0X_1 + X_0X_2 + X_1X_2)]\, f(X_0, X_1, X_2, X_3)$$

where (θ_B) is to be substituted for (f) and the divided differences obtained here conform to the definitions in Ref. 13. In solving this equation for θ_B' at the hinge points $X = 0$ and $X = 1$, the values (0, 1/2, 3/2, 5/2) are chosen for (X_0, X_1, X_2, X_3), respectively; at $X = 2$ the values (1/2, 3/2, 5/2, 3) are used. It is easily verified that

$$f_0' \quad (= \theta_B' \text{ @ } X = 0) \quad = \frac{46}{15}\mu_1 - \frac{41}{60}\mu_2 + \frac{3}{20}\mu_3$$

$$f_1' \quad (= \theta_B' \text{ @ } X = 1) \quad = \frac{67}{60}\mu_2 - \frac{2}{15}\mu_1 - \frac{1}{20}\mu_3$$

$$f_2' \quad (= \theta_B' \text{ @ } X = 2) \quad = \frac{67}{60}\mu_3 - \frac{1}{20}\mu_2$$

The internal moments acting on the hub and the inner, central, and outer segments would then be ($-f_0'$), ($f_0' - f_1'$), ($f_1' - f_2'$), and (f_2'), respectively, multiplied by the appropriate element of $[R]$. The amount by which this exceeds the corresponding component of $\{\lambda [R] \underline{U}\}$ is provided by the supplemental internal torque computations in Part IV and the weighting coefficients (Fortran designation "EPSIL") in Part II of the program.

The description of General Program inputs will now continue, with the damping tensor as the next item. Aside from the previously mentioned "weak dampers" in the locked modes, the only nonzero value for the present program is located at the first hinge, its value controlled by the normalized damping ratio (f_d) as indicated in Eq. (B-2):

$$R'_{(1,2,2)} = 2 f_d \, I_{(2,2,2)} \sqrt{(3+\sin^2\delta)(S_d-1)\,\mu_E/a_0^3} \qquad \text{(B-11)}$$

and a description of hysteresis damping (for the case f_d = 0) appears later in this Appendix.

S ("S(I,J)") Structure of the incidence matrix (Note Fig. 3):

$$[S] = \begin{bmatrix}
-1 & -1 & -1 & -1 & -1 & 0 & & \\
1 & & & & & 0 & & \\
 & 1 & & & & -1 & & \\
 & & 1 & & & & -1 & \\
 & & & \ddots & & & & \ddots \\
 & & & & & & & -1 \\
 & & & & \ddots & & & 0 \\
 & & & & & & & 0 \\
 & & & & & & & 0 \\
 & & & & & & & 1
\end{bmatrix}$$

(B-12)

ρ ("RHO(J,α,β)") Rest position rotation matrix between adjacent bodies with 3 x 3 sub-matrices $[\rho_j]$, $1 \le j \le (N-1)$:

$$[\rho_1] = [65\pi/180]_z$$

$$[\rho_2] = [\pi/3]_y$$

$$[\rho_3] = [2\pi/3]_y$$

$$[\rho_4] = [-2\pi/3]_y \qquad \text{(B-13)}$$

$$[\rho_5] = [-\pi/3]_y$$

$$[\rho_j] = [I_{33}], \quad 6 \le j \le (N-1)$$

C ("C(I,J)") Hinge connection matrix

$c_{31} = 3(0.3048)$; See Fig. 3

$$c_{12} = -c_{13} = -c_{14} = c_{15} = -\tfrac{1}{2}\ell_1$$

$$c_{32} = c_{33} = -c_{34} = -c_{35} = \frac{\sqrt{3}}{2}\ell_1 \qquad \text{(B-14)}$$

$$c_{ij} = -\tfrac{1}{2}\ell \begin{cases} 2 \le j \le (N-1),\ i = 3j+1 \ \text{AND}^* \\ 6 \le j \le (N-1),\ i = 3(j-4)+1 \end{cases}$$

All other elements of $[C]$ are zero.

*These values all have the same sign, since the reversal in direction from segment mass center to opposite hinges will be cancelled by the sign reversal in the pertinent incidence matrix elements.

A_E ("AE(I)") Effective areas for solar pressure forces:

If $N_A = 0$, all $A_{E(I)} = 0$; otherwise compute the values indicated (See Eqs. A-38 and A-42):

$$A_{E(1)} = \frac{1}{2}\pi \ell_1^2 \tag{B-15}$$

$$A_{E(2)} = \frac{2}{3} d \ell_d \tag{B-16}$$

$$A_{E(i)} = \frac{2}{3} d \ell \ , \quad 3 \leq i \leq N \tag{B-17}$$

$N_c, \mathcal{M}_j$ ("NC","MI(J)") Total number of locked hinge degrees of freedom and hinge axis indexing number respectively. Below is a table showing values of N_c and $\mathcal{M}_j$ for the three locked mode options, N_L .

N_L	N_c	$\mathcal{M}$
1	2	$\mathcal{M}_1 = 1; \mathcal{M}_3 = 2$
2	N	$\mathcal{M}_1 = 1; \mathcal{M}_3 = 2; \mathcal{M}_{\{3(J-1)+1\}} = J+1,\ 2 \leq J \leq N-1$
3	14	$\mathcal{M}_1 = 1; \mathcal{M}_3 = 2; \mathcal{M}_J = J-1,\ 4 \leq J \leq 15$

J_E' ("XJE") Thermal bending constant for Earth radiation as defined by Eq. (A-19).

J_S' ("XJS") Thermal bending constant for solar radiation as defined by Eq. (A-19).

This completes the RAE System Parameter specifications. The initial conditions need somewhat more detailed treatment here because of their relation to a Lagrangian formulation of the RAE satellite.[12]

Initial values for the angular position $\{$ direction cosine matrix $[\theta_I]\}$ and angular velocity vector $\underline{\omega}_I$ of each member I of the discrete RAE satellite model are derived from the initial conditions of an equivalent flexible continuous RAE satellite model.[12] First cantilever mode shape amplitudes of the booms are linearly transformed into a set of "satellite modes" ($X_5 \cdots X_{12}$). It is of interest to excite these "satellite modes" separately or in combination in the discrete model program. Also, nonzero initial values of the libration Euler angles (X_1, X_2, X_3) with respect to the local frame and of the single degree of freedom damper angle (X_4) from the continuous model will excite similar motion in the discrete model. Thus the initial values of the twelve quantities ($X_1 \cdots X_{12}$) are transformed to the initial attitude of each member in the discrete model. It is noted that there is no loss of generality in setting the initial derivatives ($\dot{X}_1 \cdots \dot{X}_{12}$) to zero, since motion is still excited by initial displacements from equilibrium.

The transformation from the twelve variables of the continuous model to each member's attitude in the discrete model is accomplished separately for the hub and the damper boom, and in combination for the antenna boom segments. First, the orthogonal transformation from hub to local axes is written as

$$[\theta_1] = [X_2]_Y [X_1]_X [X_3 + \gamma]_Z \tag{B-18}$$

where X_1, X_2, and X_3 are the roll, pitch, and yaw libration angles respectively and γ is a static yaw angle of the hub body axes in equilibrium due to the skewed damper boom. The orthogonal transformation from damper principal axes to the local frame is

$$[\theta_2] = [\theta_1][\rho_1]^T[X_4]_y \tag{B-19}$$

where $[\rho_1]$ represents the 65 degree hub-to-damper transformation about the yaw axis (see Fig. 3).

The conversion from the generalized coordinates describing the satellite deformation modes to any boom segment direction cosine matrix can be separated into three steps:

1. First, all in-plane and out of plane tip deflections can be expressed as a linear combination of the satellite flexing mode amplitudes. For example the in-plane and out of plane tip deflections of the lower left antenna boom are

$$W_z = -\frac{1}{2}\left[X_6 + (X_8 + 2K_A) - X_9 - X_{11}\right] \tag{B-20}$$

and

$$W_y = \frac{1}{2}\left[X_5 + (X_7 + 2K_B) - X_{10} - X_{12}\right] \tag{B-21}$$

where K_A and K_B are the in and out of plane static (equilibrium) tip deflections, respectively. All other boom tip deflections follow in a similar fashion from the transformation defined in Ref. 12.

2. In the second step, the elastic deformation slope is computed for each segment, making use of the first cantilever mode shape. A question immediately arises as to the method of fitting a finite number of segments to the cantilever curve. The segments could be inscribed or circumscribed, or their mass centers could be matched to the mode shape function; alternatively, the slope of each segment could be chosen to match the corresponding portion of strain energy in the continuous

elastic curve. Actually, the accompanying Fortran listing uses none of these methods. Instead, the first cantilevered mode function was approximated by a least squares fit, giving rise to proportionality constants ($\mathcal{A}$; Fortran designation "SLSQ") which fix the slopes of the kth segment as

$$\Delta_y = \mathcal{A}_k \, W_z/\ell_B \quad , \quad 1 \le k \le n \tag{B-22}$$

and

$$\Delta_z = \mathcal{A}_k \, W_y/\ell_B \quad , \quad 1 \le k \le n \tag{B-23}$$

for any values of in-plane (W_z) and transverse (W_y) tip deflection. It was then found that, for three* segments per boom, the results could be improved through small changes in the relative magnitudes of ($\mathcal{A}_k$). Chosen values for this case (i.e., n = 3) minimized the initial angular accelerations under equilibrium conditions. No further improvements were investigated for other segmented approximations, but curve fitting is recognized as a possible means of improving future discretized structural models of this type.

3. Finally, the transformation is computed for segment-to-local coordinates. Again using the lower left antenna quadrant as an example,

$$[\theta_7] = [\theta_1][\rho_2]^T[\Delta] \tag{B-24}$$

where $[\Delta]$ is computed exactly as in Eq. (A-21) using the angles in (B-22) and (B-23).

*A three-segment model was chosen for actual run trials, as a compromise between accuracy and economy of computation.

Initial angular rates arise solely from orbital motion since, as previously explained, the initial generalized co-ordinate derivatives $(\dot{X}_1, \dot{X}_2, \ldots. \dot{X}_{12})$ are zero. For any member, then, the initial angular rate vector is

$$\underline{\omega}_I = \dot{v} [\theta_I]^T \underline{1}_2 \quad , \qquad 1 \leq I \leq N \tag{B-25}$$

where ($\dot{v}$) follows readily from Eq. (A-52).

HYSTERESIS DAMPER SIMULATION

The hysteresis damping torque for the RAE satellite is taken from the model found in Ref. 14. The damping torque equation, repeated here, is

$$T_H = T_R + 2T_P\left[1 - \exp\left\{-\Sigma\left|\lambda - \lambda_R\right|\right\}\right]\operatorname{sgn}\left(\dot{\lambda}\right) \qquad \text{(B-26)}$$

where:

T_R = damping torque at the time when $\dot{\lambda}$ last changed sign,

T_P = peak (saturation) damper torque,

Σ = exponential rate constant,

λ = angle between damper axis and damper rest position (i.e., X_4 in the problem at hand),

λ_R = damper angle when $\dot{\lambda}$ last changed sign.

The damper hinge has only one rotational degree of freedom (y-axis of the damper boom). The other two degrees of freedom are eliminated by locked modes as described in Appendix A. When the input variable f_d is set to zero, this hysteresis torque replaces the usual (linear) computation for damping torques in part IV.

RAE READOUT DERIVATIONS AND FORMATS

The readouts consist of (1) constant parameters printed only once per computer run, and (2) variable parameters computed and printed out at multiple intervals during the simulated orbital period. The format will appear as written below.

Constant Readouts

"FAIRCHILD BOOMS OF" (n) "SEGMENTS PER BOOM"

"NL = " (N_L) "E = " (E)

"DAMPER SPRING CONSTANT" (S_d) "DAMPING RATIO" (f_d)

"ORBIT" "A" (a_o) "O" (Ω_o) "I" (i_o) "E" (e_o) "NR = " (N_R)

"SIMULATION TO LAST" (T) "ORBITS"

"INITIAL CONDITIONS"

"1" (X_1) "2" (X_2) .... etc., to (X_{12})

Sunline vector components in inertial coordinates:

"SUN" (σ_1'') (σ_2'') (σ_3'')

Variable Readouts

At integral multiples of T_o/N_R (where $T_o = 2\pi\sqrt{\mu_E/a_o^3}$ is the orbital period) the computations itemized below are performed and readouts printed.

(1) Time and satellite position $\{$ direction cosine elements of the transformation $[D]$ from local to inertial co-ordinates; inertial axes are defined by the Vernal Equinox ($+X$) and the north geodetic pole ($+Z$) $\}$:

"T = ________HRS" "SATELLITE" D_{13} D_{23} D_{33}

(2) When the sightline from the satellite to the sun is not obstructed by the Earth, this will be indicated by the readout "IN SUN"; the readout "SHADOW" appears during eclipse.

(3) Attitude of the reference axes of the composite satellite with respect to local axes,

$$[\theta'] = [\theta_1][-\gamma]_{\underline{z}} \tag{B-27}$$

where γ is the static offset angle of the $\underline{X}_{HUB}-\underline{Z}_{HUB}$ plane with the orbital plane:

"ATTITUDE"

θ'_{11}	θ'_{12}	θ'_{13}
θ'_{21}	θ'_{22}	θ'_{23}
θ'_{31}	θ'_{32}	θ'_{33}

(4) Attitude of the damper boom with respect to its reference position:

$$[V] = [\rho_1][\theta_1]^T[\theta_2] \tag{B-28}$$

"DAMPER" V_{23} V_{31} V_{12}

(5) Antenna boom deformation is characterised by lateral deflections, both in and out of the cruciform reference plane, for the tip of each segment. The in-plane and out-of-plane deflections are the 3rd and 2nd components respectively of a deflection vector $\underline{d}_i$, where (i) is the segment index number. Since the deflection at the tip of any segment includes the deflections

of all inner segments, $\underline{d}$ can be computed by a recursion formula:

$$\underline{d}_{4j+K-2} = \underline{d}_{4(j-1)+K-2} + \ell\,[\rho_{K+1}][\theta_1]^T \begin{bmatrix} \theta_{4j+K-2,11} \\ \theta_{4j+K-2,21} \\ \theta_{4j+K-2,31} \end{bmatrix} \quad \text{(B-29)}$$

$$2 \leq j \leq n\ ;\ 1 \leq K \leq 4$$

where j is the index number of the segment tip. To start the recursion, the values of $\underline{d}_3$, $\underline{d}_4$, $\underline{d}_5$, and $\underline{d}_6$ are computed from the last term of the same expression, (B-29). The relative twist angle (rad.) between adjacent members is computed from the trace angle (λ) and the X-axis component of the deformation eigenvector $\underline{U}$ for the hinge connecting those members; (λ) and $\underline{U}$ are defined in Eqs. (A-2) and (A-3), respectively.

$$d_{i,1} = \lambda U_1 \quad , 3 \leq i \leq N \quad \text{(B-30)}$$

The "DEFORMATION" format is as follows:

Boom number (K)	1	2	3	4
"IN-PLANE"	d_{33}	d_{43}	d_{53}	d_{63}
	d_{73}	d_{83}	d_{93}	$d_{10,3}$
	⋮	⋮	⋮	⋮
"OUT OF PLANE"	d_{32}	d_{42}	d_{52}	d_{62}
	d_{72}	d_{82}	d_{92}	$d_{10,2}$
	⋮	⋮	⋮	⋮

"TWIST" d_{31} d_{41} d_{51} d_{61}

d_{71} d_{81} d_{91} $d_{10,1}$

$\vdots$ $\vdots$ $\vdots$ $\vdots$

(6) The amplitudes of the satellite flexing modes[12] are computed from the antenna boom tip deflections and printed out. The computation is the exact inverse of the operation used to find tip deflections from flexing mode amplitudes, described earlier in this Appendix.

For example, the in-plane neutral mode amplitude is:

$$X_8 = -\frac{1}{2}\left(d_{4n+1,3} - d_{4n+2,3} + d_{4n-1,3} - d_{4n,3}\right) - 2K_A \qquad \text{(B-31)}$$

The "SATELLITE MODES" format is as follows:

"ROLL" (X_5) "PITCH" (X_6) "YAW" (X_7)

"LONGITUDINAL" (X_9) "LATERAL" (X_{10}) "VERTICAL" (X_{11})

"IN-PLANE NEUTRAL" (X_8) "OUT OF PLANE NEUTRAL" (X_{12})

APPENDIX C

PROGRAM LISTING

The RAE segmented model program presented here has been successfully run in Fortran IV single precision on the Univac 1108 and has been found in agreement with the independent Lagrangian analysis of Ref. 12. While many of the Fortran statements are self-explanatory or follow readily from previous discussion, understanding of the overall computational scheme is enhanced in several instances by the accompanying comments, referrals to equations, cross-references between different parts of the program, etc. It will be reiterated here that usage of the program for other satellite configurations will not require knowledge of the material in these Appendices. Program utilization for general purposes calls for 1) all inputs specified in the INTRODUCTION of this report, and 2) Parts 0, III, and IV of the present listing (with present FORMAT and WRITE statements replaced by desired readouts for the particular problem under consideration*), augmented by the accompanying subroutines ICE, INTEG, INVERT, and XSIMEQ, plus card Nos. 55-58 which provide necessary zero resets for each program run. It should be noted that the hinge interactions, solar pressure forces, thermal bending, and hysteresis damping present in Part IV, unless actuated by inputs in Parts I and II, are de-activated by the cards at the end of Part 0. If similar effects are to be included in a simulation for another satellite configuration, the following minor modifications are needed:

(1) Elastic coefficients derived on pages 38-40 must be co-ordinated with the appropriate satellite geometry, necessitating logic changes in cards 818-848.

*Note that deletion of the computations associated with readouts in the present program (e.g., calculations involving the Fortran designation "SD") is optional.

(2) The geometry of each member will determine its response to solar radiation pressure; card Nos. 646-652 will be replaced accordingly if solar pressure effects are to be taken into account.

(3) The thermal bending formulation was derived in terms of lengths between <u>centers</u> of adjacent members (See discussion preceding Eq. A-10). When this is not uniform throughout the structure, simple logic must be introduced into the computation (for the RAE this is done by card Nos. 722-724).

(4) Nonlinear damping and/or spring action can easily be simulated by modifying the internal torque ("EL") computation at any hinge.

```
C          PART 0
C
      INTEGER XSIMEQ
      DIMENSION MI(75),SQU(78,38),GAM(38),XT(38,38)
      DIMENSION JQ(78),XIB(78),XID(78),DVEC(3,3)
      DIMENSION S(26,25), PSI(78,78), PSIV(78,78),EM(26), A(26,3,3),
     1TH(26,3,3),R(25,3,3),RP(25,3,3),RHO(25,3,3),RHOP(25,3,3),AE(26)
     2,C(78,78)
      DIMENSION WV(78),WM(3,26),WD(78),VECT(78),H(78),
     1G(78),EL(78), Q(78), UU(78), V(3,3), VP(3,3), P(3,3),PQ(25,25),
     2PSQ(25,25),U(3),UP(3),WKP(3),Z(78,26),B(26,26,3)
      DIMENSION DUR(3,3),DDRD(3,3),EPSIL(6,7)
      DIMENSION Y(312),DY(312),ERL(312),DICE(2184),SIG(3,26),DELT(3),
     1SIGP(3),SIGDP(3)
      EQUIVALENCE(PSQ(1,1),PSI(1,1)),(PQ(1,1),Z(1,1)),(WM(1,1),WV(1))
     1 ,(B,Z),(C,PSIV)
      EQUIVALENCE (Y(1),WV(1)),(DY(1),WD(1))
      DATA PI /3.14159265/,FMU /0.398613E+15/,SPC /0.45E-05/,ERRD /0.638
     1E+07/
      IBUG = 0                  Auxiliary readout control.
      IORD = 4                  Inputs to subroutine ICE (See Card No. 583).
      KPRI = 0
      DO 12006 I=1,6
      DO 12006 J=1,7
12006 EPSIL(I,J) = 0.0          Deactivates interaction in hinge moment computations.
      DO 6781 I=1,26
 6781 AE(I) = 0.0               Deactivates solar pressure unless subsequently overridden.
      XJE = 0.0                 Deactivates thermal bending unless subsequently overridden.
      XJS = 0.0
      FD = 1.E-03               Deactivates hysteresis damping computations in Part IV for general
                                program utilization (See card Nos. 799 and 936).
C
C          PART I
C
      DIMENSION XINIT(12),ZIN(4),YIN(4),F(3),SHAD(2),SD(26,3)
      DIMENSION SLSQ(6)
      DATA (SHAD(I),I=1,2) /6HSHADOW,6HIN SUN/
      KBBC = 0
      KASE = 0                  Provisions for running more than one case.
 9999 CONTINUE
      KASE = KASE+1
      IF (KBBC .EQ. 0) GO TO 1970
      STP1 = T/KBBC
```

```
      KBBC = 0
      WRITE (6,1971) STP1 ———————— Readout for numerical integration step size.
 1971 FORMAT (1H1E18.5)
 1970 CONTINUE
      READ (5,190) NL,NA,ITHERM,NPB,THZ,EYZ,EZ,ENR,ORBS,FD,SD1
  190 FORMAT (4I1,7F10.0)
      READ (5,1901) XINIT
 1901 FORMAT (6F10.0)
      IF (NL .EQ. 3) NPB = 1 ———————— Antenna base lock is allowed only for the rigid cruciform.
C
C        PART II
C
      DO 1031 J=1,38
      GAM(J) = 0.0
      DO 1031 I=1,78
 1031 SQU(I,J) = 0.0
      DO 1032 I=1,26
      DO 1032 J=1,3
 1032 SD(I,J) = 0.0
      DO 1029 J = 1,6
 1029 SLSQ(J) = 0.
      GO TO (1038,1033,1034,1035,1036,1037),NPB ———— See pages 45-46.
 1033 SLSQ(1) = .595386
      SLSQ(2) = 1.40461
      GO TO 1038
 1034 CONTINUE
      SLSQ(1) = .5461  }
      SLSQ(2) = 1.1432 } ———— Obtained from minimum initial |dω/dt| at equilibrium;
      SLSQ(3) = 1.3107 }      slopes for 2, 4, 5 or 6 segments were obtained by least squares.
      GO TO 1038
 1035 SLSQ(1) = .321406
      SLSQ(2) = 1.01717
      SLSQ(3) = 1.28445
      SLSQ(4) = 1.37697
      GO TO 1038
 1036 SLSQ(1) = .261152
      SLSQ(2) = .865497
      SLSQ(3) = 1.1645
      SLSQ(4) = 1.33302
      SLSQ(5) = 1.37583
      GO TO 1038
 1037 SLSQ(1) = .21987
      SLSQ(2) = .750147
```

```
      SLSQ(3) = 1.04962
      SLSQ(4) = 1.25421
      SLSQ(5) = 1.34979
      SLSQ(6) = 1.37636
 1038 CONTINUE
      SIGH = 77.                              Damper exponential constant.
      TPH=.4506                               This normalized hysteresis saturation torque (equivalent to the net
                                              spring torque on an isolated damper, skewed at 65° and inclined
                                              at 0.4506 radian) when expressed in English units corresponds to
                                              1.05 x 10^-3 ft.lb. - See card No. 461.
      TRH = 0.0          } Hysteresis
      AMBDR = 0.0        } damper
      SDO = 1.0          } initialization
      AMDDT = 0.         }
      XJS = 3.0E+08*SPC/4184.                 Eq. A-16.
      XJE = 5.67E-08*246.0**4/4184.           Eq. A-15.
      DO 11000 I=1,6
      DO 11000 J=1,7
11000 EPSIL(I,J) = 0.0
      GO TO (11004,11008,11005,11006,11007,11009),NPB        See page 40.
11C08 EPSIL(1,1) = -13./6.
      EPSIL(1,2) = 7./3.
      EPSIL(1,3) = -1./6.
      EPSIL(2,1) = 5./6.
      EPSIL(2,2) = -1.
      EPSIL(2,3) = 1./6.
      GO TO 11004
11005 CONTINUE
      EPSIL(1,1) = -31./15.
      EPSIL(1,2) = 11./5.
      EPSIL(1,3) = -2./15.
      EPSIL(2,1) = 41./60.
      EPSIL(2,2) = -4./5.
      EPSIL(2,3) = 1./6.
      EPSIL(2,4) = -1./20.
      EPSIL(3,1) = -3./20.
      EPSIL(3,2) = 1./5.
      EPSIL(3,3) = -1./6.
      EPSIL(3,4) = 7./60.
      GO TO 11004
11006 CONTINUE
      EPSIL(1,1) = -31./15.
      EPSIL(1,2) = 11./5.
      EPSIL(1,3) = -2./15.
      EPSIL(2,1) = 41./60.
      EPSIL(2,2) = -4./5.
```

```
      EPSIL(2,3) =19./120.
      EPSIL(2,4) = -1./24.
      EPSIL(3,1) = -3./20.
      EPSIL(3,2) = 1./5.
      EPSIL(3,3) = -2./15.
      EPSIL(3,4) = 2./15.
      EPSIL(3,5) = -1./20.
      EPSIL(4,3) = 1./24.
      EPSIL(4,4) = -19./120.
      EPSIL(4,5) = 7./60.
      GO TO 11004
11007 CONTINUE
      EPSIL(1,1) = -31./15.
      EPSIL(1,2) = 11./5.
      EPSIL(1,3) = -2./15.
      EPSIL(2,1) = 41./60.
      EPSIL(2,2) = -4./5.
      EPSIL(2,3) = 19./120.
      EPSIL(2,4) = -1./24.
      EPSIL(3,1) = -3./20.
      EPSIL(3,2) = 1./5.
      EPSIL(3,3) = -2./15.
      EPSIL(3,4) = 1./8.
      EPSIL(3,5) = -1./24.
      EPSIL(4,3) = 1./24.
      EPSIL(4,4) = -1./8.
      EPSIL(4,5) = 2./15.
      EPSIL(4,6) = -1./20.
      EPSIL(5,4) = 1./24.
      EPSIL(5,5) = -19./120.
      EPSIL(5,6) = 7./60.
      GO TO 11004
11009 EPSIL(1,1) = -31./15.
      EPSIL(1,2) = 11./5.
      EPSIL(1,3) = -2./15.
      EPSIL(2,1) = 41./60.
      EPSIL(2,2) = -4./5.
      EPSIL(2,3) = 19./120.
      EPSIL(2,4) = -1./24.
      EPSIL(3,1) = -3./20.
      EPSIL(3,2) = 1./5.
      EPSIL(3,3) = -2./15.
      EPSIL(3,4) = 1./8.
```

```
172          EPSIL(3,5) = -1./24.
173          EPSIL(4,3) = 1./24.
174          EPSIL(4,4) = -1./8.
175          EPSIL(4,5) = 1./8.
176          EPSIL(4,6) = -1./24.
177          EPSIL(5,4) = 1./24.
178          EPSIL(5,5) = -1./8.
179          EPSIL(5,6) = 2./15.
180          EPSIL(5,7) = -1./20.
181          EPSIL(6,5) = 1./24.
182          EPSIL(6,6) = -19./120.
183          EPSIL(6,7) = 7./60.
184    11004 CONTINUE
185          TZ = 0.0
186          WZ = 0.0
187          AZ = 0.1238E+08 ———— Semimajor axis for 6000 Km. altitude.
188          ND = 80 ———— March 21.
189          ENPB = NPB
190    C
191    C     FAIRCHILD BOOM
192    C
193          ELB = 750.*.3048 ———— Antenna length in meters.
194      103 SL = ELB/ENPB ———— Segment length.
195          THK = .508E-04 ———— Boom wall thickness.
196          E = .117E+12
197     3418 AW = 0.0 ———— Value of E used in Eq. B-10 includes effects of perforations.
198          POR = 0.3 ———— Poisson's ratio.
199          EMB = 0.480E-03*14.5939/0.3048 ———— Linear mass density of booms (kg./m.).
200          CK = 0.031 ———— Thermal conductivity of booms (page 36).
201          DIA = 0.587/39.37
202          OLA = 0.1 ———— Small overlap angle due to interlocking at seam; not critical.
203          CTE = 0.0
204          IF (ITHERM .NE. 0) CTE = 1.87E-05 ———— coefficient of thermal expansion.
205          F(2) = .125*DIA**3*THK*(PI+OLA+SIN(OLA)*COS(OLA)-2.0*SIN(OLA)**2/
206         1(PI+OLA)) ———— Eq. B-3.
207          F(3) = .125*DIA**3*THK*(PI+OLA-SIN(OLA)*COS(OLA)) ———— Eq. B-4.
208          F(1) = 0.5/(1+POR)*(F(2)+F(3)) ———— Eq. B-5.
209          DO 104 I = 1,3
210      104 F(I) = E*F(I)*(1.0-AW)/SL ———— Eq. B-10; see card No. 368.
211          XJE = ELB/4.0/ENPB/CK/THK*0.1*CTE*DIA*XJE  }
212          XJS = ELB/4.0/ENPB/CK/THK*.05*CTE*DIA*XJS  } ———— Eq. A-19.
213          N = 4*NPB+2 ———— Segments, hub, and damper.
214          N3 = 3*N
```

```
      NM = N-1                          Allowable error per integration step for angular rates (rad./sec).
      EEE1=1.0E-06/2.0
      EEE2=1.0E-04/2.0                  Allowable error per integration step for direction cosines.
      DO 5555 I=1,N3
 5555 ERL(I) = EEE1
      N12 = 12*N
      NNNN = N3+1
      DO 5556 I=NNNN,N12
 5556 ERL(I) = EEE2
      DANG = 65.*PI/180.                Angle ΘD of Ref. 12; damper skew angle.
      QKA = 35.57
      QBA = .9830                       Equilibrium parameters (Ref. 12, Case 2).
      QGAM = .1131
      DO 7301 I=1,75
 7301 MI(I) = 0
      GO TO (7303,7304,7305),NL         See pages 34 and 43.
 7303 NC = 2
      MI(1) = 1
      MI(3) = 2
      GO TO 7401
 7304 NC = N
      MI(1) = 1
      MI(3) = 2
      DO 7307 I=2,NM
      M = 3*(I-1)+1
 7307 MI(M) = I+1
      GO TO 7401
 7305 NC = 14
      QKA = 0.0
      QBA = 0.0                         Values for rigid cruciform.
      QGAM = .0623134
      MI(1) = 1
      MI(3) = 2
      DO 7308 I=4,15
 7308 MI(I) = I-1                       Assumed hub dimension; not at all critical.
 7401 CONTINUE
      EL1 = 1.5*0.3048
      ELD = CBRT(12./EMB*1.0E+04*14.5939*.3048**2)   Damper length corresponding to
      EM(1) = 10.52*14.5939                          10^4 slug-ft.^2
      EM(2) = ELD*EMB
      DO 105 I = 3,N
  105 EM(I) = ELB*EMB/ENPB
      DO 106 I = 1,N
```

```
          DO 106 J = 1,3
          DO 106 K = 1,3
    106   A(I,J,K) = 0.0
          A(1,1,1) = 14.24
          A(1,2,2) = 90.8
          A(1,3,3) = 92.68
          A(1,1,1) = A(1,1,1)*1000.
          A(1,2,2) = A(1,2,2)*200.          See footnote on page 37.
          A(1,3,3) = A(1,3,3)*20.
   1065    CONTINUE
          DO 107 I = 1,3
    107   A(1,I,I) = A(1,I,I)*14.5939*0.3048**2
          A(2,1,1) = EM(2)*DIA**2/4.0          Eq. B-6.
          A(2,1,1) = A(2,1,1)*10000.
          DO 108 I = 2,3
    108   A(2,I,I) = 1./12.*EM(2)*ELD**2          Eq. B-7.
          DO 109 I = 3,N
          A(I,1,1) = EM(I)*DIA**2/4.0          Eq. B-6.
          A(I,1,1) = A(I,1,1)*10000.          Artificial enlargement of small inertias.
          DO 109 J = 2,3
    109   A(I,J,J) = 1./12.*EM(I)*SL**2          Eq. B-8.
          IF(NA) 110,111,110
    111   DO 112 I = 1,N
    112   AE(I) = 0.0
          GO TO 114
    110   AE(1) = 0.5*PI*EL1**2          Sphere; Eq. B-15.
          AE(2) = 2./3.*DIA*ELD          Cylinder; Eq. B-16.
          DO 113 I = 3,N
    113   AE(I) = 2./3.*DIA*SL          Cylinder; Eq. B-17.
    114   M = N-1
          DO 115 I = 1,N
          DO 115 J = 1,M
    115   S(I,J) = 0.0
          DO 116 I = 1,5
    116   S(1,I) = -1.0
          IF (NPB .EQ. 1) GO TO 5070
          DO 117 I = 6,M          Eq. B-12.
          J = I-3
    117   S(J,I) = -1.0
   5070   CONTINUE
          DO 118 I = 1,M
          J = I+1
    118   S(J,I) = 1.0
```

```
      M = 3*N
      DO 119 I = 1,M
      DO 119 J = 1,M
  119 C(I,J) = 0.0
      C(3,1) = 3.0*0.3048
      C(1,2) = -0.5*EL1
      C(1,5) = C(1,2)
      C(1,3) = -C(1,2)
      C(1,4) = C(1,3)
      C(3,4) = -EL1*SIN(PI/3.)
      C(3,5) = C(3,4)
      C(3,2) = -C(3,4)
      C(3,3) = C(3,2)
      M = N-1
      DO 120 I = 2,M
      K = 3*I+1
  120 C(K,I) = -SL/2.
      IF (NPB .EQ. 1) GO TO 5071
      DO 121 J = 6,M
      K = 3*(J-4)+1
  121 C(K,J) = -SL/2.
```

Eq. B-14. Note: Damper displacement C_{31} below hub mass center is not at all critical.

```
 5071 CONTINUE
      DO 122 I = 1,M
      DO 122 J = 1,3
      DO 122 K = 1,3
  122 RHO(I,J,K) = 0.0
      RHO(1,1,1) = COS(DANG)
      RHO(1,2,2) = COS(DANG)
      RHO(1,1,2) = SIN(DANG)
      RHO(1,2,1) = -RHO(1,1,2)
      RHO(1,3,3) = 1.
      RHO(2,2,2) = 1.
      RHO(3,2,2) = 1.
      RHO(4,2,2) = 1.
      RHO(5,2,2) = 1.
      RHO(2,1,1) = .5
      RHO(2,3,3) = .5
      RHO(5,1,1) = .5
      RHO(5,3,3) = .5
      RHO(3,1,1) = -.5
      RHO(3,3,3) = -.5
      RHO(4,1,1) = -.5
      RHO(4,3,3) = -.5
```

Eq. B-13.

```
344        RHO(2,3,1) = SIN(PI/3.)
345        RHO(3,3,1) = SIN(PI/3.)
346        RHO(4,1,3) = SIN(PI/3.)
347        RHO(5,1,3) = SIN(PI/3.)
348        RHO(2,1,3) = -SIN(PI/3.)
349        RHO(3,1,3) = -SIN(PI/3.)          Eq. B-13.
350        RHO(4,3,1) = -SIN(PI/3.)
351        RHO(5,3,1) = -SIN(PI/3.)
352        IF (NPB .EQ. 1) GO TO 5072
353        DO 123 I = 6,M
354        DO 123 K = 1,3
355   123  RHO(I,K,K) = 1.
356   5072 CONTINUE
357        M = N-1
358        DO 127 I = 1,M
359        DO 127 J = 1,3
360        DO 127 K = 1,3
361        R(I,J,K) = 0.0
362   127  RP(I,J,K) = 0.0
363        IF (NL .EQ. 3) GO TO 5073          Rigid Cruciform.
364        IQL = 1
365        IF (NL .EQ. 2) IQL = 2             Locked torsional modes.
366        DO 128 I=2,M
367        DO 128 J=IQL,3
368   128  R(I,J,J) = F(J)                    See card No. 210.
369        IF (NL .NE. 2) GO TO 1289
370   5073 CONTINUE
371        DO 1288 I=1,NM
372        R(I,1,1) = 10.*A(I+1,2,2)*FMU/AZ**3            Weak springs and dampers for locked torsiona
373        RP(I,1,1) = 5.*A(I+1,2,2)*SQRT(FMU/AZ**3)      modes.
374   1288 CONTINUE
375   1289 CONTINUE
376        R(1,2,2) = SD1*(3.0+SIN(DANG -QGAM)**2)*FMU/AZ**3*A(2,2,2)
                                          Eq. B-9 (at S_d = 1.298 thi provides a damper spring
                                          constant of 0.01015 ft.-lb./ radian).
377        RP(1,2,2) = 2.0*FD*A(2,2,2)*SQRT((3.0+SIN(DANG  -QGAM)**2)*
378       1(SD1-1.0)*FMU/AZ**3)           Eq. B-11.
379        DUR(2,2) = R(2,2,2)            Used for interacting hinges.
380        DUR(3,3) = R(2,3,3)
381        XC = COS(XINIT(1))
382        YC = COS(XINIT(2))
383        ZC = COS(QGAM+XINIT(3))
384        XF = SIN(XINIT(1))
385        YF = SIN(XINIT(2))
386        ZF = SIN(QGAM+XINIT(3))
```

```
      TH(1,1,1) = ZC*YC-XF*YF*ZF
      TH(1,1,2) = ZF*YC+XF*YF*ZC
      TH(1,1,3) = -XC*YF
      TH(1,2,1) = -XC*ZF
      TH(1,2,2) = XC*ZC                                Eq. B-18.
      TH(1,2,3) = XF
      TH(1,3,1) = ZC*YF+XF*YC*ZF
      TH(1,3,2) = YF*ZF-XF*YC*ZC
      TH(1,3,3) = XC*YC
      TH(2,1,1) = COS(XINIT(4))
      TH(2,1,2) = 0.0
      TH(2,1,3) = -SIN(XINIT(4))
      TH(2,2,1) = 0.0
      TH(2,2,2) = 1.0
      TH(2,2,3) = 0.0
      TH(2,3,1) = SIN(XINIT(4))
      TH(2,3,2) = 0.0
      TH(2,3,3) = COS(XINIT(4))
      DO 7480 I=1,3                                                 Eq. B-19.
      DO 7480 J=1,3
      DVEC(I,J) = 0.0
      DO 7480 K=1,3
 7480 DVEC(I,J) = DVEC(I,J)+RHO(1,K,I)*TH(2,K,J)
      DO 7481 I=1,3
      DO 7481 J=1,3
      TH(2,I,J) = 0.0
      DO 7481 K=1,3
 7481 TH(2,I,J) = TH(2,I,J)+TH(1,I,K)*DVEC(K,J)
      XINIT(7) = XINIT(7)+2.0*QBA
      XINIT(8) = XINIT(8)+2.0*QKA
      ZIN(1) = -0.5*(XINIT(6)+XINIT(8)-XINIT(9)-XINIT(11))     Eq. B-20.
      ZIN(2) = -0.5*(XINIT(6)-XINIT(8)-XINIT(9)+XINIT(11))
      ZIN(3) = -0.5*(XINIT(6)+XINIT(8)+XINIT(9)+XINIT(11))
      ZIN(4) =-0.5*(XINIT(6)-XINIT(8)+XINIT(9)-XINIT(11))
      YIN(1) = 0.5*(XINIT(5)+XINIT(7)-XINIT(10)-XINIT(12))     Eq. B-21.
      YIN(2) = -0.5*(-XINIT(5)+XINIT(7)+XINIT(10)-XINIT(12))
      YIN(3) = -0.5*(XINIT(5)+XINIT(7)+XINIT(10)+XINIT(12))
      YIN(4) = 0.5*(-XINIT(5)+XINIT(7)-XINIT(10)+XINIT(12))
      DO 1812 K=1,4
      DO 1812 J=1,NPB
      I = 4*J-2+K
      YYY = SLSQ(J)*ZIN(K)/ELB                   Eq. B-22.
      ZZZ = SLSQ(J)*YIN(K)/ELB                   Eq. B-23.
```

```
430           YYS = YYY**2
431           ZZS = ZZZ**2
432           SQUR = SQRT(1.0-YYS-ZZS)
433           TH(I,1,1) = SQUR
434           TH(I,1,2) = -ZZZ
435           TH(I,1,3) = -YYY
436           TH(I,2,1) = ZZZ
437           TH(I,2,2) =(YYS+ZZS*SQUR)/(YYS+ZZS)
438           TH(I,2,3) = YYY*ZZZ*(SQUR-1.0)/(YYS+ZZS)
439           TH(I,3,1) = YYY
440           TH(I,3,2) = TH(I,2,3)
441           TH(I,3,3) = (ZZS+YYS*SQUR)/(YYS+ZZS)
442           IF (YYS+ZZS .LT. 1.E-30) TH(I,2,2) = 1.0 }
443           IF (YYS+ZZS .LT. 1.E-30) TH(I,3,3) = 1.0 } ——— Resolution of possible singularity.
444           DO 1813 II=1,3
445           DO 1813 JJ=1,3
446           DVEC(II,JJ) = 0.0
447           DO 1813 KK=1,3
448      1813 DVEC(II,JJ) = DVEC(II,JJ)+RHO(K+1,KK,II)*TH(I,KK,JJ)
449           DO 1814 II=1,3
450           DO 1814 JJ=1,3
451           TH(I,II,JJ) = 0.0
452           DO 1814 KK=1,3
453      1814 TH(I,II,JJ) = TH(I,II,JJ)+TH(1,II,KK)*DVEC(KK,JJ)
454      1812 CONTINUE
455           DO 1815 I=1,N
456           DO 1815 J=1,3
457           CAM = -SQRT(FMU)*TZ/AZ**1.5
458      1815 WM(J,I) = TH(I,2,J)*SQRT(FMU)/AZ**1.5*(1.0+2.0*EZ*COS(CAM)
459          1 +2.5*EZ**2*COS(2.0*CAM)+1.0/12.0*EZ**3*(39.*COS(3.0*CAM)-3.0*
460          1 COS(CAM))) ——————————————————————————————————————— Eq. B-25.
461           TPH = TPH*(SD1-1.0)*(3.+SIN(DANG -QGAM)**2)*A(2,2,2)*FMU/AZ**3 ——— See card Nos. 92 and 376.
462           WRITE (6,4916)
463      4916 FORMAT (1H130X33HRAE SATELLITE DYNAMICS SIMULATION)
464           WRITE (6,4918) NPB,NL,E
465      4918 FORMAT (1H022X19HFAIRCHILD BOOMS OF I1,19H SEGMENTS PER BOOM. 5X
466          15HNL = I3,5X4HE = E10.4)
467           IF (ITHERM .NE. 0) WRITE (6,4919)
468      4919 FORMAT (1H034X24HTHERMAL EFFECTS INCLUDED)
469           IF (NA .NE. 0) WRITE (6,4920)
470      4920 FORMAT (1H029X31HSOLAR PRESSURE EFFECTS INCLUDED)
471           WRITE (6,4921) SD1,FD
472      4921 FORMAT (1H020X22HDAMPER SPRING CONSTANT E12.4,5X13HDAMPING RATIO
```

```
     1E12.4)
      WRITE (6,4922) AZ,THZ,EYZ,EZ,ENR
 4922 FORMAT (1H016X5HORBIT5X1HAE12.5,5X1HOE12.5,5X1HIE12.5,5X1HEE12.5,
     1 5X5HNR = F3.0)
      WRITE (6,4923)    ORBS
 4923 FORMAT (1H030X19HSIMULATION TO LAST F6.3,7H ORBITS)
      XINIT(7) = XINIT(7)-2.*QBA
      XINIT(8) = XINIT(8)-2.*QKA
      WRITE (6,4924) (I,XINIT(I),I=1,12)
 4924 FORMAT (1H035X18HINITIAL CONDITIONS//(I5,E10.4,I5,E10.4,I5,
     1E10.4,I5,E10.4,I5,E10.4,I5,E10.4))
      CG = COS(QGAM)
      SG = SIN(QGAM)
C
C        PART III
C
      IPART = 3
      CAPM = 0.0
      DO 3299 I=1,N
 3299 CAPM = CAPM+EM(I) ———————————— Total mass.
      ENZ = SQRT(FMU)/AZ**1.5
      PZ = AZ*(1.0-EZ**2) ————————————Used in Eq. A-46.
      WZ = WZ*PI/180.
      EYZ = EYZ*PI/180.
      THZ = THZ*PI/180.
      TO = 2.0*PI/ENZ
      CAPT = TO*ORBS ———————————————— Duration of run (seconds).
      XIF = SIN(EYZ)
      XIC = COS(EYZ)
      THF = SIN(THZ)
      THC = COS(THZ)
      PSIS = 2.*PI*FLOAT(ND-80)/365.——— Eq. A-50.
      XIS = 23.5*PI/180.
      SIGDP(1) = COS(PSIS)          }
      SIGDP(2) = COS(XIS)*SIN(PSIS) }—— Eq. A-51.
      SIGDP(3) = SIN(XIS)*SIN(PSIS) }
      WRITE (6,974) SIGDP
  974 FORMAT (1H05X3HSUN7X3E16.7)
      DO 420 I = 1,NM                  NOTE  Background for remainder of Part III is contained in Ref. 3.
      DO 410 J = 1,NM
      PSQ(I,J) = S(I+1,J)
  410 PQ(I,J) = 0.
  420 PQ(I,I) = 1.
```

```
      MQ = 25
      LQ = XSIMEQ(MQ,NM,NM,PSQ,PQ,DQ,JQ) ———————— Theorem 2 of Ref. 3.
      GO TO (425,900,910),LQ
  425 DO 440 I = 1,N3
      Z(I,1) = 0.
      DO 430 J = 1,NM
      Z(I,J+1) = 0.
      DO 430 K = 1,NM
  430 Z(I,J+1) = Z(I,J+1)+C(I,K)*PSQ(K,J) ————————Matrix [D] of Ref. 3.  NOTE:  Original [C]
                                                     is no longer needed; its storage can now be used
                                                     for other computations.
      DO 440 J = 1,N
      C(I,J) = 0.
      DO 440 K = 1,N
      DUM = -EM(K)/CAPM
      IF(K.EQ.J) DUM = DUM+1.
  440 C(I,J) = C(I,J)+Z(I,K)*DUM*EM(J) ———————— Product [D][μ] m of Ref. 3.
      DO 450 I=1,N3
      DO 450 J = 1,N3
      PSI(I,J) = 0.
      DO 450 K = 1,N
  450 PSI(I,J) = PSI(I,J)+C(I,K)*Z(J,K) ————————Matrix [J] of Ref. 3.
      WRITE (6,17249)
17249 FORMAT (1H )
      DO 4705 I=1,N3
      IALF = 1+MOD(I-1,3)
      ICAP = 1+(I-IALF)/3
      DO 470 J=1,N3
      IBET = 1+MOD(J-1,3)
      JCAP = 1+(J-IBET)/3
      IF (ICAP-JCAP) 470,3701,470
 3701 CONTINUE
      TAU = 0.
      IF (IALF-IBET) 4702,4703,4702
 4703 CONTINUE
      IF(IALF.EQ.IBET) TAU =
     1PSI(3*ICAP,3*ICAP)+PSI(3*ICAP-1,3*ICAP-1)+PSI(3*ICAP-2,3*ICAP-2)
 4702 CONTINUE
       TAU = TAU-PSI(I,J)
      A(ICAP,IALF,IBET) = A(ICAP,IALF,IBET)+TAU ————————Constant augmented inertia matrix.
  470 CONTINUE
 4705 CONTINUE
      DO 480 I = 1,N3
      IALF = 1+MOD(I-1,3)
      ICAP = 1+(I-IALF)/3
```

```
      DO 480 J=1,N
      PSI(I,J) = 0.
      DO 475 K=1,N
  475 PSI(I,J) = PSI(I,J)+Z(I,K)*FM(K)
      PSI(I,J)=-PSI(I,J)/CAPM+Z(I,J)
  480 CONTINUE
      DO 490 I = 1,N3
      IALF = 1+MOD(I-1,3)
      ICAP = 1+(I-IALF)/3
      DO 490 J=1,N
  490 B(ICAP,J,IALF) = PSI(I,J) ———— Barycentric vectors.
C
C        PART IV
C
      DO 495 I=1,N
      DO 495 L=1,3
      II = N3+(I-1)*9+(L-1)*3
      DO 495 K=1,3
  495 Y(II+K) = TH(I,K,L)                 } Preparation for numerical integration.
      TR = TO/ENR
      LICE = 4
      NV = 12*N
      T = 0.0
      GO TO 831
  800 CALL ICE (TR,T,TP,NV,Y,DY,DTCE,LICE,IND,IORD,KPRI,ERL) ——— Numerical integration.
      GO TO (810,820,830,840),LICE
C          BOX A
  810 DO 500 I = 1,N
      II = N3+(I-1)*9
      DO 500 K = 1,3
      DO 500 L = 1,3
  500 TH(I,K,L) = Y(II+3*L-3+K)
      KRUM = KRUM+1
      IPART = 4
      AM = ENZ*(T-TZ)
      COAM = COS(AM)
      RS = AZ*(1.-EZ*(COAM+.5*EZ*(COS(AM+AM)-1.+.75*EZ*(COS(3.*AM)-COAM)
     1))) ———————————————————————————————— Eq. A-49.
      SCRF = 2.0*(1.0-SQRT(1.0-(ERRD/RS)**2)) ————————————— Eq. A-14.
      RMU = 3.*FMU*RS**(-3)
      VV = AM+2.0*EZ*SIN(AM)+1.25*EZ**2*SIN(2.0*AM)
     1+1.0/12.0*EZ**3*(13.0*SIN(3.0*AM)-3.0*SIN(AM)) ————————— Eq. A-52.
      CV = COS(VV)
```

```
      SV = SIN(VV)
      SWZ = SIN(WZ)
      CWZ = COS(WZ)
      VC = CWZ*CV-SWZ*SV
      VF = SWZ*CV+CWZ*SV
      SIGP(1) = (-THC*VF-THF*XIC*VC)*SIGDP(1)+(-THF*VF+THC*XIC*VC)
     1*SIGDP(2)+XIF*VC*SIGDP(3)
      SIGP(2) = THF*XIF*SIGDP(1)-THC*XIF*SIGDP(2)+XIC*SIGDP(3)
      D13 = THC*VC-THF*XIC*VF                                  Eq. A-53.
      D23 = THF*VC+THC*XIC*VF
      D33 = XIF*VF
      SIGP(3) = D13*SIGDP(1)+D23*SIGDP(2)+D33*SIGDP(3)
      DO 530 I = 1,N
      ICON = 3*(I-1)
      DO 510 J = 1,3
      U(J) = 0.
      DO 510 K = 1,3
  510 U(J) = U(J)+A(I,J,K)*WM(K,I)                   Eq. 5, page 9.
      UU(ICON+1) = WM(2,I)*U(3)-WM(3,I)*U(2)
      UU(ICON+2) = WM(3,I)*U(1)-WM(1,I)*U(3)
      UU(ICON+3) = WM(1,I)*U(2)-WM(2,I)*U(1)
      DO 520 J = 1,3
      U(J) = 0.
      DO 520 K = 1,3
  520 U(J) = U(J)+A(I,J,K)*TH(I,3,K) ---------- First term of Eq. 19 of Ref. 3.
      H(ICON+1) = RMU*(TH(I,3,2)*U(3)-TH(I,3,3)*U(2))
      H(ICON+2) = RMU*(TH(I,3,3)*U(1)-TH(I,3,1)*U(3))
  530 H(ICON+3) = RMU*(TH(I,3,1)*U(2)-TH(I,3,2)*U(1))
      DO 550 I = 1,N3
      EL(I) = 0.
      Q(I) = 0.
      G(I) = 0.
      DO 550 J = 1,N3
  550 PSI(I,J) = 0.
      RUM = FMU*CAPM*RS**(-3)
      DUM = -SQRT(1.0-(ERRD/RS)**2)
      IS = 1
      IF(SIGP(3).GT.DUM) IS = 2 ---------------------- In sunlight.
      DO 730 I = 1,N
      IF (IS .EQ. 1 )          GO TO 140
  141 DO 143 JJ = 1,3
      SIG(JJ,I) = 0.0
      DO 143 K = 1,3
```

```
143  SIG(JJ,I) = SIG(JJ,I)+TH(I,K,JJ)*SIGP(K)          Eq. A-54.
     IF(I.NE.1) GO TO 144
     DO 145 JJ = 1,3
145  UP(JJ) = -SIG(JJ,I)*2.0*SPC*AE(I)                 Solar pressure force on sphere.
     GO TO 140
144  UP(1) = 0.0                                        Solar pressure force
     DO 146 JJ = 2,3                                    on cylinder.
146  UP(JJ) = -SIG(JJ,I)*2.0*SPC*SQRT(SIG(2,I)**2+SIG(3,I)**2)*AE(I)
140  CONTINUE
     ICON = 3*(I-1)
     DO 730 J = 1,N
     JCON = 3*(J-1)
     DO 670 JJ = 1,3
     DO 660 II = 1,3
     V(II,JJ) = 0.
     DO 660 K = 1,3
660  V(II,JJ) = V(II,JJ)+TH(I,K,II)*TH(J,K,JJ)
     P(1,JJ) = B(I,J,2)*V(3,JJ)-B(I,J,3)*V(2,JJ)
     P(2,JJ) = B(I,J,3)*V(1,JJ)-B(I,J,1)*V(3,JJ)        Eq. 12b of Ref. 3.
670  P(3,JJ) = B(I,J,1)*V(2,JJ)-B(I,J,2)*V(1,JJ)
     DO 690 K = 1,3
     II = ICON+K
     IF(I.EQ.J) GO TO 680
     PSI(II,JCON+1) = CAPM*(P(K,2)*B(J,I,3)-P(K,3)*B(J,I,2))
     PSI(II,JCON+2) = CAPM*(P(K,3)*B(J,I,1)-P(K,1)*B(J,I,3))     Eq. 14c of Ref. 3;
     PSI(II,JCON+3) = CAPM*(P(K,1)*B(J,I,2)-P(K,2)*B(J,I,1))     applicable for I ≠ J.
     GO TO 690
680  PSI(II,JCON+1) = A(I,K,1)
     PSI(II,JCON+2) = A(I,K,2)
     PSI(II,JCON+3) = A(I,K,3)
690  CONTINUE
     DO 700 K = 1,3
     U(K) = 0.
     DO 700 L = 1,3
     DUM = 3.*TH(J,3,K)*TH(J,3,L)
     IF(K.EQ.L) DUM = DUM-1.
700  U(K) = U(K)+DUM*B(J,I,L)
     IF (I .EQ. J) GO  TO 7210                          Second term of Eq. 19
     DO 710 K = 1,3                                     of Ref. 3.
710  G(ICON+K) = G(ICON+K)+RUM*(P(K,1)*U(1)+P(K,2)*U(2)+P(K,3)*U(3))
     U(1) = -B(J,I,1)*(WM(3,J)**2+WM(2,J)**2)+B(J,I,2)*
    1WM(1,J)*WM(2,J)+B(J,I,3)*WM(1,J)*WM(3,J)
     U(2) =  B(J,I,1)*WM(1,J)*WM(2,J)-B(J,I,2)*(WM(1,J)**2+
```

```
     1WM(3,J)**2)+B(J,I,3)*WM(2,J)*WM(3,J)
      U(3) = B(J,I,1)*WM(1,J)*WM(3,J)+B(J,I,2)*WM(2,J)*WM(3,J)
     1-B(J,I,3)*(WM(1,J)**2+WM(2,J)**2)
      DO 720 K = 1,3
  720 Q(ICON+K) = Q(ICON+K)+CAPM*(P(K,1)*U(1)+P(K,2)*U(2)+P(K,3)*U(3))  —— Eq. 14d of Ref. 3.
 7210 CONTINUE
      GO TO (147,148),IS
  148 DO 149 JJ = 1,3
      K = 3*(I-1)+JJ
  149 G(K) = G(K)+P(JJ,1)*UP(1)+P(JJ,2)*UP(2)+P(JJ,3)*UP(3)  —— Add moment of solar
                                                                  pressure force.
  147 CONTINUE
  730 CONTINUE
      DO 650 I = 1,N
      ICON = 3*(I-1)
      DO 6500 J=1,NM
      IF (S(I,J)-1.0) 6500,735,6500
  735 KAY = 0
      DO 560 K = 1,N
      IF(K.EQ.I) GO TO 560
      IF(S(K,J)) 555,560,555
  555 IF(KAY.NE.0) GO TO 920        —— Location of adjacent members for internal torques.
      KAY = K
  560 CONTINUE
      IF(KAY.LE.0) GO TO 920
      IF(S(KAY,J)+1.) 920,565,920
  565 TRACE = 0.
      ZIS = IS
      DO 150 II = 1,3
  150 DELT(II) = 0.0
      IF(J.EQ.1) GO TO 151   —— No thermal effects at damper hinge.
      DELT(2) = -SCRF*XJE*TH(I,3,3)+XJS*(SIG(3,I)-0.4*SCRF*TH(I,3,3))*
     1(ZIS-1.0)  —— Eq. A-18.
      DELT(3) = -SCRF*XJE*TH(I,3,2)+XJS*(SIG(2,I)-0.4*SCRF*TH(I,3,2))*
     1(ZIS-1.0)  —— Eq. A-20.
      IF (J .EQ. 1 .OR. J .GT. 5) GO TO 151
      DELT(2) = DELT(2)/2.          —— See discussion at beginning of Appendix C.
      DELT(3) = DELT(3)/2.
  151 CONTINUE
      YYY = DELT(2)
      ZZZ = DELT(3)
      YYS = YYY**2
      ZZS = ZZZ**2
      SQUR = SQRT(1.0-YYS-ZZS)
```

```
      DDRD(1,1) = SQUR
      DDRD(1,2) = -ZZZ
      DDRD(1,3) = -YYY
      DDRD(2,1) = ZZZ
      DDRD(2,2) = (YYS+ZZS*SQUR)/(YYS+ZZS)                     Eq. A-21.
      DDRD(2,3) = YYY*ZZZ*(SQUR-1.0)/(YYS+ZZS)
      DDRD(3,1) = YYY
      DDRD(3,2) = DDRD(2,3)
      DDRD(3,3) =(ZZS+YYS*SQUR)/(YYS+ZZS)
      IF (YYS+ZZS .LT. 1.E-30) DDRD(2,2) = 1.0      Elimination of possible indeterminacy
      IF (YYS+ZZS .LT. 1.E-30) DDRD(3,3) = 1.0      in Eq. A-21.
      DO 11119 II=1,3
      DO 11119 JJ=1,3
      RHOP(J,II,JJ) = 0.0
      DO 11119 KK=1,3
11119 RHOP(J,II,JJ) = RHOP(J,II,JJ)+DDRD(II,KK)*RHO(J,KK,JJ)   Eqs. A-9 and A-21.
  152 CONTINUE
      DO 570 K = 1,3
      DO 570 L = 1,3
      VP(K,L) = 0.
      DO 570 JJ = 1,3
  570 VP(K,L) = VP(K,L)+TH(I,JJ,K)*TH(KAY,JJ,L)
      DO 590 K = 1,3
      DO 580 L = 1,3
      V(K,L) = 0.
      DO 580 II = 1,3
  580 V(K,L) = V(K,L)+VP(L,II)*RHOP(J,K,II)                   Eq. A-1.
  590 TRACE = TRACE+V(K,K)
      IF (ABS(V(1,2))+ABS(V(1,3))+ABS(V(2,3)) .LT. .01) GO TO 603   Off-diagonal vector element
      CAMB = .5*(TRACE-1.)                                           for small angles.
      IF(ABS(CAMB).LT.1.) GO TO 595
      CAMB = SIGN(1.,CAMB)          Overrides small numerical error.
  595 AMBDA = ACOS(CAMB)                                       Eq. A-2.
C        EIGENVECTOR OF/V/                                     See Eq. A-3.
      IF(ABS(V(2,3)).GT.ABS(V(1,2))) GO TO 596
      IF(ABS(V(1,3)).GT.ABS(V(1,2))) GO TO 597
      U(1) = V(1,2)*V(2,3)-V(1,3)*(V(2,2)-1.)
      U(2) = V(1,3)*V(2,1)-V(2,3)*(V(1,1)-1.)     Cross product of 1st and 2nd rows of [V-I].
      U(3) = (V(1,1)-1.)*(V(2,2)-1.)-V(1,2)*V(2,1)
      GO TO 599
  596 IF(ABS(V(2,3)).GT.ABS(V(1,3))) GO TO 598
  597 U(1) = V(3,2)*V(1,3)-V(1,2)*(V(3,3)-1.)
      U(2) = (V(1,1)-1.)*(V(3,3)-1.)-V(1,3)*V(3,1)
```

```
      U(3) = V(3,1)*V(1,2)-V(3,2)*(V(1,1)-1.)
      GO TO 599
  598 U(1) = (V(2,2)-1.)*(V(3,3)-1.)-V(2,3)*V(3,2)
      U(2) = V(2,3)*V(3,1)-V(2,1)*(V(3,3)-1.)
      U(3) = V(2,1)*V(3,2)-(V(2,2)-1.)*V(3,1)
  599 UNRM = SQRT(U(1)**2+U(2)**2+U(3)**2)
      DO 600 K = 1,3
  600 U(K) = U(K)/UNRM
      U(1) = SIGN(U(1),V(2,3))
      U(2) = SIGN(U(2),V(3,1))
      U(3) = SIGN(U(3),V(1,2))
      GO TO 607
  603 AMBDA = SQRT(V(1,2)**2+V(2,3)**2+V(3,1)**2)
      U(1) = V(2,3)/AMBDA
      U(2) = V(3,1)/AMBDA
      U(3) = V(1,2)/AMBDA
C
  607 KON = 3*(KAY-1)
      DO 610 K=1,3
      WKP(K) = 0.
      DO 610 L=1,3
  610 WKP(K) = WKP(K)+VP(K,L)*WM(L,KAY)          Second half of Eq. A-6.
      DO 630 K = 1,3
      UP(K)=0.                                   Bypasses the hysteresis damper computation for
      DUM = 0.                                   general program utilization - See card No. 30.
      IF (FD .GT. 1.E-05) GO TO 4871
      IF (J .NE. 1 .OR. K .NE. 2) GO TO 4871
      SAMBDA = SIGN(AMBDA,V(3,1))
      TRYTH = TRH+SIGN(2.0*TPH*(1.0-EXP(-SIGH*ABS(SAMBDA-AMBDR))),AMDDT)   Eq. B-26.
      THH = SIGN(AMIN1(ABS(TRYTH),TPH),TRYTH)    Torque saturation.
      DO 4872 IJK = 1,3
 4872 UP(2) = UP(2)+R(J,K,IJK)*U(IJK)            Damper spring torque (Rigidity coefficient
      UP(2) = AMBDA*UP(2)+THH                    assumed fixed, i.e., no damper stops).
      WKPS = WKP(2)
      AMBDS = SAMBDA
      GO TO 630
 4871 CONTINUE
      DO 620 L = 1,3
      UP(K) = UP(K)+RP(J,K,L)*(WKP(L)-WM(L,I))   Viscous Damping.
  620 DUM = DUM+R(J,K,L)*U(L)
      UP(K) = UP(K)+AMBDA*DUM                    Damping plus spring torque.
  630 EL(ICON+K) = EL(ICON+K)+UP(K)              Internal torque.
      DO 640 K = 1,3
```

```
  640 EL(KON+K) = EL(KON+K)-VP(1,K)*UP(1)-VP(2,K)*UP(2)-VP(3,K)*UP(3) ——— Equal and opposite torque.
      IF (J .EQ. 1) GO TO 6501 ——— No interactions at damper hinge.
      DO 5546 II=1,3
      DO 5546 JJ=1,3
 5546 DVEC(II,JJ) = AMBDA*TH(I,II,JJ)
      DO 5547 II=1,3
      UP(II) = 0.0
      DO 5547 JJ=1,3
 5547 UP(II) = UP(II)+DVEC(II,JJ)*U(JJ)
      JS = 1+(J-2)/4
      DO 5550 II=1,3
      DVEC(1,II) = 0.0
      DO 5550 JJ=1,3
 5550 DVEC(1,II) = DVEC(1,II)+EPSIL(JS,1)*TH(1,JJ,II)*UP(JJ)
      DO 15551 II=1,3
      DVEC(2,II) = 0.0
      DO 5551 JJ=1,3
 5551 DVEC(2,II) = DVEC(2,II)+DUR(II,JJ)*DVEC(1,JJ)
15551 EL(II) = EL(II)+DVEC(2,II)                                          See pages 38-40.
      IQ = MOD(J-2,4)+3
      NP1 = NPB+1
      DO 5552 IP=2,NP1
      IA = IQ+4*(IP-2)
      DO 15553 II=1,3
      DO 15553 JJ=1,3
      DVEC(II,JJ) = 0.0
      DO 15553 KK=1,3
15553 DVEC(II,JJ) = DVEC(II,JJ)+DUR(II,KK)*TH(IA,JJ,KK)
      DO 5553 II=1,3
      IIA = II+3*(IA-1)
      DO 5553 IJ = 1,3
 5553 EL(IIA) = EL(IIA)+EPSIL(JS,IP)*DVEC(II,IJ)*UP(IJ)
 5552 CONTINUE
 6501 SD(J+1,1) = AMBDA*U(1) ——— Torsional displacement readout.
      DO 4601 K = 1,3
      KK = 3*(I-1)+K
      KL = 3*(J-1)+K
      IF (MI(KL) .EQ. 0) GO TO 4601 ——— Mode not locked.
      INDX = MI(KL)
      SQU(KK,INDX) = 1.0
      DO 4602 IB = 1,3                ——— Eq. A-28.
      III = 3*(KAY-1)+IB
 4602 SQU(III,INDX) = -VP(K,IB)
```

```
      GO TO (4603,4604,4605),K
4603  GAM(INDX) = WM(2,I)*WKP(3)-WM(3,I)*WKP(2)
      GO TO 4601
4604  GAM(INDX) = WM(3,I)*WKP(1)-WM(1,I)*WKP(3)      Eq. A-29.
      GO TO 4601
4605  GAM(INDX) = WM(1,I)*WKP(2)-WM(2,I)*WKP(1)
4601  CONTINUE
6500  CONTINUE
650   CONTINUE
      DO 750 J = 1,N3
750   VECT(J) = -UU(J)+H(J)+EL(J)-G(J)+Q(J)      Eq. 7, page 9.
      DO 5601 I=1,N3
      DO 5601 J=1,N3
5601  PSIV(I,J) = PSI(I,J)
7249  FORMAT (1H0/(9E11.4))
      CALL INVERT (PSIV,N3,78)
      IF (NC .EQ. 0) GO TO 14717
      DO 4639II = 1,NC
      DO 4639JJ = 1,NC
4639  XT(II,JJ) = 0.0
      DO 4700 II = 1,N3
      DO 4700 KK = 1,NC
      SUM = 0.0
      DO 4701 LL = 1,N3
4701  SUM = SUM+PSIV(II,LL)*SQU(LL,KK)
      DO 4700 JJ = 1,NC
4700  XT(JJ,KK)= XT(JJ,KK)+SQU(II,JJ)*SUM      Matrix [F]^T [Γ]^-1 [F] , Eq. A-31.
      CALL INVERT (XT,NC,38)
      DO 4712 I = 1,N3
      XID (I) = 0.0
      DO 4712 J = 1,N3
4712  XID(I) = XID(I)+PSIV(I,J)*VECT(J)
      DO 4713 I=1,NC
      XIB(I) = 0.0
      DO 4713 J = 1,N3
4713  XIB(I)=XIB(I)+SQU(J,I)*XID(J)      Vector [F]^T [Γ]^-1 E , Eq. A-31.
      DO 4714 I = 1,NC
      XID(I) = 0.0
      DO 4714 J = 1,NC
4714  XID(I) = XID(I)+XT(I,J)*XIB(J)
      DO 4715 I = 1,N3
      DO 4715 J = 1,NC
4715  VECT (I) = VECT(I)-SQU(I,J)*XID(J)
```

```
903          DO 4716 I = 1,NC
904          XID(I) = 0.0
905          DO 4716 J = 1,NC
906     4716 XID(I) = XID(I)+XT(I,J)*GAM(J)
907          DO 4717 I = 1,N3
908          DO 4717 J = 1,NC
909     4717 VECT(I) = VECT(I)-SQU(I,J)*XID(J)
910    14717 CONTINUE
911          DO 5610 I = 1,N3
912          WD(I) = 0.0
913          DO 5610 J = 1,N3
914     5610 WD(I) = WD(I)+VECT(J)*PSIV(I,J) ———— ω̇ from Eq. A-31.
915          CONS = SQRT(FMU*PZ)/RS**2 ———— Eq. A-46.
916          DO 780 I = 1,N
917          II = N3+(I-1)*9
918          DO 780 K = 1,3
919          CONK = (K-2)*CONS
920          DY(II+K) = TH(I,K,2)*WM(3,I)-TH(I,K,3)*WM(2,I)+CONK*TH(I,4-K,1)   }
921          DY(II+K+3) = TH(I,K,3)*WM(1,I)-TH(I,K,1)*WM(3,I)+CONK*TH(I,4-K,2) } —— Eq. A-48.
922     780  DY(II+K+6) = TH(I,K,1)*WM(2,I)-TH(I,K,2)*WM(1,I)+CONK*TH(I,4-K,3) }
923          IF (IBUG .NE. 0) WRITE (6,2060) T,RS,CONS
924          IF(IBUG.NE.0) WRITE(6,2200) (I,UU(I),H(I),EL(I),G(I),Q(I),WV(I), Extra readouts
925         1 WD(I),I=1,N3)
926     2060 FORMAT(1H0/10X16HPART  V - TIME = F8.2,19H,  ORBITAL RADIUS =       Y
927         1 F7.3,17H,  ORBITAL RATE = E8.3//)
928     2200 FORMAT(1H0,20X1HW,11X1HH,11X1HL,11X1HG,11X1HQ,7X5HOMEGA,
929         1 3X9HOMEGA DOT//(I10,7E12.4))
930          IF (IBUG .NE. 0 .AND. FD .LT. 1.E-05) WRITE (6,7654) AMBDS,AMBDR,
931         1 TRH,TRYTH,THH,AMDDT,WKPS,TPH
932     7654 FORMAT (1H05E18.5)
933          GO TO 800                       Bypasses the hysteresis damper computation for general
934     820  CONTINUE                        program utilization - See card No. 30.
935          KBBC = KBBC+1
936          IF (FD .GT. 1.E-05) GO TO 800
937          AMDDT = WKPS-WM(2,2) ———— Time derivative of damper angle -- See card No. 807.
938          SNN = SIGN(1.0,AMDDT)
939          IF (ABS(SNN+SOO) .GT. 1.5) GO TO 4887 ———— denotes no polarity change in hysteresis
940          TRH = THH      }                                   damper angle derivative.
941          AMBDR = AMBDS  } ——Reset values for Eq. B-26.
942     4887 SOO = SNN
943          GO TO 800
944     830  CONTINUE
945          THOUR = T/3600.
```

```
      WRITE (6,950) THOUR,D13,D23,D33,SHAD(IS)
  831 DVEC(1,1) = TH(1,1,1)*CG+TH(1,1,2)*SG
      DVEC(1,2) = -TH(1,1,1)*SG+TH(1,1,2)*CG
      DVEC(1,3) = TH(1,1,3)
      DVEC(2,1) = TH(1,2,1)*CG+TH(1,2,2)*SG
      DVEC(2,2) = -TH(1,2,1)*SG+TH(1,2,2)*CG
      DVEC(2,3) =TH(1,2,3)                          Eq. B-27.
      DVEC(3,1) = TH(1,3,1)*CG+TH(1,3,2)*SG
      DVEC(3,2) = -TH(1,3,1)*SG+TH(1,3,2)*CG
      DVEC(3,3) = TH(1,3,3)
      WRITE(6,951)((DVEC(I,J),J = 1,3),I = 1,3)
  951 FORMAT (1H040X8HATTITUDE //(20X3E16.7))
      DO 160 I = 1,3
      DO 160 J = 1,3
      DVEC(I,J) = 0.0
      DO 160 K = 1,3
  160 DVEC(I,J) = TH(1,K,I)*TH(2,K,J)+DVEC(I,J)
      DO 161 I = 1,3
      DO 161 J = 1,3
      V(I,J) = 0.0
      DO 161 K = 1,3
  161 V(I,J) = V(I,J)+RHO(1,I,K)*DVEC(K,J)          Eq. B-28.
      WRITE (6,952) V(2,3),V(3,1),V(1,2)
  952 FORMAT (1H010X6HDAMPER5X3E16.7)
      DO 162 II = 3,6
      DO 163 I = 1,3
      DO 163 J = 1,3
      DVEC(I,J) = 0.0
      DO 163 K = 1,3
  163 DVEC(I,J) = DVEC(I,J)+RHO(II-1,I,K)*TH(1,J,K)
      DO 164 I = 2,3
      SD(II,I) = 0.0
      DO 164 J = 1,3
  164 SD(II,I) = SD(II,I)+SL*DVEC(I,J)*TH(II,J,1)   Eq. B-29.
  162 CONTINUE
      DO 166 K=1,4
      DO 167 II = 1,3
      DO 167 JJ = 1,3
      DVEC(II,JJ) = 0.0
      DO 167 KK = 1,3
  167 DVEC(II,JJ) = DVEC(II,JJ)+SL*RHO(K+1,II,KK)* TH(1,JJ,KK)
      DO 165 J=2,NPB
      INDX = 4*J+K-2
```

```
      DO 168 II=2,3
      SD(INDX,II) = 0.0
      DO 168 JJ=1,3
  168 SD(INDX,II) = SD(INDX,II)+DVEC(II,JJ)*TH(INDX,JJ,1)
  165 CONTINUE                                                   Eq. B-29.
  166 CONTINUE
      DO 169 II=2,3
      DO 169 JJ=7,N
  169 SD(JJ,II) = SD(JJ,II)+SD(JJ-4,II)
      WRITE (6,953) (SD(I,3),I = 3,N)
      WRITE (6,954)(SD(I,2),I = 3,N)
      WRITE (6,955)(SD(I,1),I = 3,N)
  953 FORMAT (1H030X11HDEFORMATION/4X8HIN PLANE /(15X4E16.7))
  954 FORMAT (4X12HOUT OF PLANE /(15X4E16.7))
      DO 3330 I = 1,4
      J = 4*NPB-2+I
      ZIN(I)=SD(J,3)
 3330 YIN(I) = SD(J,2)
      XINIT(5) = -0.5*(YIN(3)+YIN(4)-YIN(1)-YIN(2))
      XINIT(6) = -0.5*(ZIN(3)+ZIN(4)+ZIN(1)+ZIN(2))
      XINIT(7) = -0.5*(YIN(3)-YIN(4)-YIN(1)+YIN(2))-2.0*QBA
      XINIT(8) = -0.5*(ZIN(3)-ZIN(4)+ZIN(1)-ZIN(2))-2.0*QKA      Eq. B-31.
      XINIT(9) = -0.5*(ZIN(3)+ZIN(4)-ZIN(1)-ZIN(2))
      XINIT(10) = -0.5*(YIN(3)+YIN(4)+YIN(1)+YIN(2))
      XINIT(11) = -0.5*(ZIN(3)-ZIN(4)-ZIN(1)+ZIN(2))
      XINIT(12) = -0.5*(YIN(3)-YIN(4)+YIN(1)-YIN(2))
      WRITE (6,9511) XINIT(5),XINIT(6),XINIT(7),XINIT(9),XINIT(10),
     1XINIT(11),XINIT(8),XINIT(12)
 9511 FORMAT (1H027X15HSATELLITE MODES//4X4HROLL9XE10.4,5X5HPITCH
     18XE10.4,5X3HYAW10XE10.4//4X12HLONGITUDINAL1XE10.4,5X7HLATERAL
     26XE10.4,5X8HVERTICAL5XE10.4//4X16HIN PLANE NEUTRAL5XE10.4,
     36X20HOUT OF PLANE NEUTRAL6XE10.4)
      IF (T .GT. CAPT) GO TO 9999                                Duration of program run.
      TP = T+TR
      GO TO 800
  840 CONTINUE
      GO TO 800
 2080 FORMAT(1H1,30X36HBEGIN INTEGRATION,  PRINT INTERVAL =F8.4///)
C           ERROR STOPS
  900 WRITE(6,3100) IPART
      GO TO 990
  910 WRITE(6,3110) IPART
      GO TO 990
```

```
920  WRITE(6,3120)(S(K,J),K=1,N)
990  WRITE(6,3000)
999  GO TO 9999
950  FORMAT (1H15X4HT = E16.7,6H HOURS5X9HSATELLITE3X3E16.7,4XA6)
955  FORMAT (4X5HTWIST /(15X4E16.7))
3000 FORMAT(1H0,15X,18H*** ERROR STOP ***)
3100 FORMAT(1H0,10X14HOVERFLOW, PART I3)
3110 FORMAT(1H0,10X,21HSINGULAR MATRIX, PART I3)
3120 FORMAT(1H0,10X,24HBAD COLUMN IN MATRIX /S///(15F6.1))
     END
```

Subroutines

Following are listings for standard Westinghouse subroutines ICE, INTEG, INVERT, and XSIMEQ. For theoretical descriptions of these computational schemes the reader is referred to "Standard Subroutines Used by Westinghouse RAE Operational Programs," April, 1968, Contract No. NAS5-9753-20.

```
      SUBROUTINE ICE (P,TT,TP,NN,Y,DY,F,L,INDEX,I,KPRI,ERL)
      DIMENSION ERL(1)
      DIMENSION DUMPR(312)
      DIMENSION Y(1),DY(1),F(1)                                         AAIC0030
C     DIMENSION Y(1),DY(1),F(1)                                         AAIC0040
      T = TT                                                            AAIC0050
      GO TO (100,200,300,400),L                                         AAIC0060
  100 IG=IG                                                             AAIC0070
      GO TO (101,102),IG                                                AAIC0080
  101 J = 1                                                             AAIC0090
      L = 2                                                             AAIC0100
      M = 0                                                             AAIC0110
      TS = T                                                            AAIC0120
      DO 106 K = 1,N                                                    AAIC0130
      K1 = K+3*N                                                        AAIC0140
      K2 = K1+N                                                         AAIC0150
      K3 = N + K                                                        AAIC0160
      F(K1) = Y(K)                                                      AAIC0170
      F(K3) = F(K1)                                                     AAIC0180
  106 F(K2) = DY(K)                                                     AAIC0190
      GO TO 402                                                         AAIC0200
  102 CALL INTEG(T,DT,   N,Y(1),DY(1),F(1),J,I)                         AAIC0210
      J = J+1                                                           AAIC0220
      IF(J-I   ) 103,103,104                                            AAIC0230
  103 L = 1                                                             AAIC0240
      GO TO 402                                                         AAIC0250
  104 M = M+1                                                           AAIC0260
  105 GO TO (110,120,130),M                                             AAIC0270
  110 DO 111 K = 1,N                                                    AAIC0280
      K1 = K+N+N                                                        AAIC0290
  111 F(K1) = Y(K)                                                      AAIC0300
  112 DO 113 K = 1,N                                                    AAIC0310
      K1 = K+3*N                                                        AAIC0320
      K2 = K1+N                                                         AAIC0330
      K3 = N + K                                                        AAIC0340
      Y(K) = F(K1)                                                      AAIC0350
      F(K3) = F(K1)                                                     AAIC0360
  113 DY(K) = F(K2)

      T = TS                                                            AAIC0380
      IF(T) 114,116,114                                                 AAIC0390
  114 IF(ABS(H/T)-.000001) 115,115,116                                  AAIC0400
```

```
      115  M = 0                                                        AAIC0410
           L = 4                                                        AAIC0420
           GO TO 402                                                    AAIC0430
      116  DT = .5*H                                                    AAIC0440
           M = 1                                                        AAIC0450
           J = 1                                                        AAIC0460
           GO TO 300                                                    AAIC0470
      120  DO 121 K = 1,N                                               AAIC0480
           K1 = K+N                                                     AAIC0490
      121  F(K1) = Y(K)                                                 AAIC0500
           M = 2                                                        AAIC0510
           J = 1                                                        AAIC0520
           IG = 2                                                       AAIC0530
           L = 1                                                        AAIC0540
           GO TO 402                                                    AAIC0550
      130  DO 131 K = 1,N                                               AAIC0560
           K1 = K+2*N                                                   AAIC0570
           F(K) = (Y(K)-F(K1))/(2.0**I-1.0)                             AAIC0580
           DUMPR(K) = Y(K)
           Y(K) = Y(K) +F(K)                                            AAIC0590
      131  CONTINUE                                                     AAIC0640
           IF (KPRI .EQ. 0) GO TO 1324
           WRITE (6,1325) T
      1325 FORMAT (1H15X18HATTEMPTED STEP AT  E10.4//10X2HY1,10X2HY210X2HYE10
          1X2HER//)
           DO 1326 IKZ = 1,N
           IAM = 2*N+IKZ
      1326 WRITE (6,1327) IKZ,F(IAM),DUMPR(IKZ),Y(IKZ),F(IKZ)
      1327 FORMAT (I5,4E12.4)
      1324 CONTINUE
           DO 1967 K=1,N
           IF (ABS(F(K)) .GT. ABS(ERL(K))/20.) GO TO 135
      1967 CONTINUE
      134  H = H+H                                                      AAIC0740
      1345 DT = H                                                       AAIC0750
           GO TO 401                                                    AAIC0760
      135  DO 1968 K=1,N
           IF (ABS(F(K)) .GT. ABS(ERL(K))) GO TO 136
      1968 CONTINUE
           GO TO 1345
      136  DO 137 K = 1,N                                               AAIC0780
           K1 = K+N                                                     AAIC0790
           K2 = K+N+N                                                   AAIC0800
```

```
137  F(K2) = F(K1)                                    AAIC0810
138  H = .5*H                                         AAIC0820
     MU=2
     GO TO 112                                        AAIC0830
200  MU=MU                                            AAIC0840
     GO TO (203,204),MU                               AAIC0850
203  H= AMAX1(AMAX1(H,H1),H2)                         AAIC0860
     MU = 2                                           AAIC0870
204  H1 = ABS(H)                                      AAIC0880
     IF(P) 205,206,206                                AAIC0890
205  H = -H1                                          AAIC0900
     GO TO 207                                        AAIC0910
206  H = H1                                           AAIC0920
207  IF(ABS(P)-H1) 208,209,209                        AAIC0930
208  H = P                                            AAIC0940
209  T2 = TP-T                                        AAIC0950
     IF(T2) 210,212,210                               AAIC0960
210  H2 = ABS(T2)                                     AAIC0970
     IF(TP) 211,213,211                               AAIC0980
211  IF(ABS(T2/TP)-.00001) 212,213,213                AAIC0990
212  T = TP                                           AAIC1000
     DT = H                                           AAIC1010
     L = 3                                            AAIC1020
     GO TO 402                                        AAIC1030
213  M = 0                                            AAIC1040
     J = 1                                            AAIC1050
     IF(H1-H2) 215,215,214                            AAIC1060
214  MU = 1                                           AAIC1070
     H = T2                                           AAIC1080
215  DT = H                                           AAIC1090
300  IG = 2                                           AAIC1100
     GO TO 102                                        AAIC1110
400  MU = 2                                           AAIC1120
     H = P                                            AAIC1130
     DT = P                                           AAIC1140
     N = NN                                           AAIC1150
401  IG = 1                                           AAIC1160
     L = 1                                            AAIC1170
402  TT = T                                           AAIC1180
     RETURN                                           AAIC1190
     END                                              AAIC1200
```

```
C           ICE INTEGRATION SUBROUTINE                                          0010
C               I = 2    SECOND ORDER RUNGE-KUTTA                               0020
C               I = 3    THIRD ORDER RUNGE-KUTTA                                0030
C               I = 4    FOURTH ORDER RUNGE-KUTTA                               0040
C           STORAGE    F1 = E    = Z1                                           0050
C                      F2 = YHAF1        TEMPORARY STORAGE REQUIRED=            0060
C                      F3 = YFULL        DIMENSION OF F ARRAY =                 0070
C                      F4 = YSAVE           N*(3+I)                             0080
C                      F5 = DYSAVE       WHERE N = NO OF DERIVATIVES            0090
C                      F6 = Z2           AND I = ORDER OF INTEGRATION           0100
C                      F7 = Z3                          PROCESS                 0110
      SUBROUTINE INTEG (T,DT,   N,Y,DY,F,J,I)                                   0120
      DIMENSION Y(1),DY(1),F(1)                                                 0130
C     DIMENSION Y(1),DY(1),F(1)                                                 0140
      DO 100 K = 1,N                                                            0150
      K1 = K                                                                    0160
      K2 = K+5*N                                                                0170
      K3 = K2+N                                                                 0180
      K4 = K + N                                                                0190
      GO TO (999,85,95,95),I                                                    0200
   85 GO TO (86,2,999,999),J                                                    0210
   86 F(K1) = DY(K)*DT                                                          0220
      Y(K) = F(K4) + F(K1)                                                      0230
      GO TO 100                                                                 0240
   95 GO TO (1,2,3,4),J                                                         0250
    1 F(K1) = DY(K)*DT                                                          0260
      Y(K) = F(K4)+.5*F(K1)                                                     0270
      GO TO 100                                                                 0280
    2 F(K2) = DY(K)*DT                                                          0290
      GO TO (999,22,23,24),I                                                    0300
    3 F(K3) = DY(K)*DT                                                          0310
      GO TO (999,33,33,34),I                                                    0320
    4 Y(K) = F(K4)+(F(K1)+2.0*(F(K2)+F(K3))+DY(K)*DT)/6.0                       0330
      GO TO 100                                                                 0340
   22 Y(K) = .5*(F(K1)+F(K2))                                                   0350
      GO TO 25                                                                  0360
   23 Y(K) = 2.0*F(K2)-F(K1)                                                    0370
      GO TO 25                                                                  0380
   24 Y(K) = .5*F(K2)                                                           0390
   25 Y(K) = Y(K)+F(K4)                                                         0400
      GO TO 100                                                                 0410
   33 Y(K) = F(K4)+(F(K1)+4.0*F(K2)+F(K3))/6.0                                  0420
```

```
      GO TO 100
   34 Y(K) = F(K4)+F(K3)
  100 CONTINUE
      GO TO (110,120,130,140),J
  110 GO TO (999,131,132,132),I
  120 GO TO (999,140,132,140),I
  130 GO TO (999,140,140,132),I
  131 T = DT + T
      GO TO 140
  132 T = T + .5*DT
  140 RETURN
  999 PAUSE
      GO TO 140
      END
```

```
C     MATRIX INVERSION BY GAUSS-JORDAN ELIMINATION
      SUBROUTINE INVERT(A,N,NN)
      DIMENSION A(NN,N),B(350),C(350),LZ(350)
      IF ( N.EQ.1) GO TO 300
      SUM=1.
      DO 5 I=1,N
    5 SUM=SUM*A(I,I)
C
      RAVG=10.**(-ALOG10(SUM)/N)
C
      DO 6 I=1,N
      DO 6 J=1,N
    6 A(I,J)=A(I,J)*RAVG
C
      DO 10 J = 1,N
   10 LZ(J) = J
      DO 20 I = 1,N
      K = I
      Y = A(I,I)
      L = I-1
      LP = I+1
      IF(N-LP) 14,11,11
   11 DO 13 J = LP,N
      W = A(I,J)
      IF(ABS(W)-ABS(Y)) 13,13,12
   12 K = J
      Y = W
   13 CONTINUE
   14 IF(Y.LT.1.E-8) GO TO 260
      DO 15 J = 1,N
      C(J) = A(J,K)
      A(J,K) = A(J,I)
      A(J,I) = -C(J)/Y
      A(I,J) = A(I,J)/Y
   15 B(J) = A(I,J)
      A(I,I) = 1./Y
      J = LZ(I)
      LZ(I) = LZ(K)
      LZ(K) = J
      DO 19 K = 1,N
      IF(I-K) 16,19,16
   16 DO 18 J = 1,N
```

```
      IF(I-J) 17,18,17
   17 A(K,J) = A(K,J)-B(J)*C(K)
   18 CONTINUE
   19 CONTINUE
   20 CONTINUE
      DO 200 I = 1,N
      IF(I-LZ(I)) 100,200,100
  100 K = I+1
      DO 500 J = K,N
      IF(I-LZ(J)) 500,600,500
  600 M =LZ(I)
      LZ(I) = LZ(J)
      LZ(J) = M
      DO 700 L = 1,N
      C(L) = A(I,L)
      A(I,L) = A(J,L)
  700 A(J,L) = C(L)
  500 CONTINUE
  200 CONTINUE
C
C
C        MAKE IT A SYMMETRIC MATRIX
C
       DO 250 I=1,N
       DO 250 J=I,N
      AVG=(A(I,J)+A(J,I))/2.*RAVG
      A(I,J)=AVG
      A(        J,I)=AVG
  250 CONTINUE
C
      RETURN
  300 IF(ABS  (A(1,1)).LT.1.E-10 )N=-IABS(N)
      A(1,1)=1./A(1,1)
      RETURN
C
  260 N=-IABS(N)
      RETURN
      END
```

```
      INTEGER FUNCTION XSIMEQ(IMAX,N,M,A,B,DET,IE)                      SIME0001
      DIMENSION A(IMAX,IMAX),B(IMAX,IMAX),IE(IMAX)                      SIME0002
      CALL OVERFL(JO)
      CALL D VCHK(JI)
      DO 43 I=1,N                                                       SIME0003
   43 IE(I) = I                                                         SIME0004
      DO 1 IN = 1,N                                                     SIME0005
      AMAX = A(IN,IN)                                                   SIME0006
      II = IN                                                           SIME0007
      JJ = IN                                                           SIME0008
      DO 11 I = IN,N                                                    SIME0009
      DO 11 J = IN,N                                                    SIME0010
      ZMT = ABS (A(I,J))                                                SIME0011
      IF(AMAX-ZMT)10,11,11                                              SIME0012
   10 AMAX = ZMT                                                        SIME0013
      II = I                                                            SIME0014
      JJ = J                                                            SIME0015
   11 CONTINUE                                                          SIME0016
      IF (A(II,JJ)) 69,33,69                                            SIME0017
   69 IF(II-IN) 13,17,13                                                SIME0018
   13 DO 15 J = 1,N                                                     SIME0019
      R = A(II,J)                                                       SIME0020
      A(II,J) = A(IN,J)                                                 SIME0021
   14 A(IN,J) = R                                                       SIME0022
      IF(J-M) 19,19,15                                                  SIME0023
   19 R = B(II,J)                                                       SIME0024
      B(II,J) = B(IN,J)                                                 SIME0025
      B(IN,J) = R                                                       SIME0026
   15 CONTINUE                                                          SIME0027
   17 IF(JJ-IN) 16,18,16                                                SIME0028
   16 DO 24 I = 1,N                                                     SIME0029
      R = A(I,JJ)                                                       SIME0030
      A(I,JJ) = A(I,IN)                                                 SIME0031
   24 A(I,IN) = R                                                       SIME0032
      IQ=IE(IN)                                                         SIME0033
      IE(IN)=IE(JJ)                                                     SIME0034
      IE(JJ) =IQ                                                        SIME0035
   18 DET = DET*AMAX    *(-1.)**((II-IN)+(JJ-IN))                       SIME0036
      KI = IN+1                                                         SIME0037
      IF (KI-N)   143,143,144                                           SIME0038
  143 DO 160  J = 1,M                                                   SIME0039
      B(IN,J) = B(IN,J)/A(IN,IN)                                        SIME0040
```

```
      DO 160 K = KI,N                                           SIME0041
  160 B(K,J) = B(K,J)-A(K,IN)*B(IN,J)                           SIME0042
      DO 80 J = KI,N                                            SIME0043
      A (IN,J) = A(IN,J)/A(IN,IN)                               SIME0044
      DO 80 K = KI,N                                            SIME0045
   80 A(K,J) = A(K,J)-A(K,IN)*A(IN,J)                           SIME0046
      DO 1 K = KI,N                                             SIME0047
    1 A(K,IN) = 0.                                              SIME0048
  145 A(IN,IN) = 1.                                             SIME0049
      NF = N-1                                                  SIME0050
      IF(NF.LE.0)GO TO 147
      DO 109 K = 1,NF                                           SIME0051
      I = N-K                                                   SIME0052
      NK = I+1                                                  SIME0053
      DO 109 L = 1,M                                            SIME0054
      SUM = 0.                                                  SIME0055
      DO 110 J = NK,N                                           SIME0056
  110 SUM = SUM+A(I,J)*B(J,L)                                   SIME0057
  109 B(I,L) = B(I,L)-SUM                                       SIME0058
  147 CONTINUE
      DO 111 K = 1,N                                            SIME0059
      DO 111 I = 1,N                                            SIME0060
      IF( IE(I)-K) 111,113,111                                  SIME0061
  111 CONTINUE                                                  SIME0062
      DO 118  I= 1,N                                            SIME0063
      DO 118  J= 1,M                                            SIME0064
  118 A(I,J) = B(I,J)                                           SIME0065
      CALL OVERFL(JO)                                           SIME0066
      CALL D VCHK(JI)                                           SIME0067
      GO TO (139,140), JO                                       SIME0068
  140 GO TO (139,141),JI                                        SIME0069
  141 XSIMEQ = 1                                                SIME0070
  189 RETURN                                                    SIME0071
   33 XSIMEQ = 3                                                SIME0072
      GO TO 189                                                 SIME0073
  144 DO 161 J = 1,M                                            SIME0074
  161 B(IN,J) = B(IN,J)/A(IN,IN)                                SIME0075
      GO TO 145                                                 SIME0076
  139 XSIMEQ = 2                                                SIME0077
      GO TO 189                                                 SIME0078
  113 DO 114 L=1,M                                              SIME0079
      Q = B(I,L)                                                SIME0080
      B(I,L) = B(K,L)                                           SIME0081
```

```
  114  B(K,L) = Q                                   SIME0082
       IQ = IE(K)                                   SIME0083
       IE(K) = IE(I)                                SIME0084
       IE(I) = IQ                                   SIME0085
       GO TO 111                                    SIME0086
       END                                          SIME0087
```

REFERENCES

1. Fletcher, H.J., L. Rongved, and E. Yu, "Dynamics Analysis of a Two-Body Gravitationally Oriented Satellite", BSTJ 42, 2239-2266 (1963).

2. Hooker, W. and G. Margulies, "The Dynamical Equations for an n-Body Satellite", J. Astro. Sci. XII, 123-128 (Winter, 1965).

3. Roberson, R.E. and J. Wittenburg, "A Dynamical Formalism for an Arbitrary Number of Interconnected Rigid Bodies, with Reference to the Problem of Satellite Attitude Control", IFAC Paper No. 46.D, London, 24 June 1966.

4. Stone, R.G., "RAE — 1500 Foot Antenna Satellite", Astronautics and Aeronautics, March 1965, p. 46.

5. Fixler, S.Z., "Effects of Solar Environment and Aerodynamic Drag on Structural Booms in Space", AIAA J. Spacecraft and Rockets 4, 315-321 (1967).

6. Levin, E., "Reflected Radiation Received by an Earth Satellite", ARSJ 32, 1328-1331 (1962).

7. Jensen, J. et. al., Design Guide to Orbital Flight, New York, McGraw-Hill, 1962, pp. 795 and 842.

8. Blanco, V.M. and S.W. McCuskey, Basic Physics of the Solar System, Reading Mass., Addison-Wesley, 1961, p. 53 and 286.

9. Danby, J.M.A., Fundamental of Celestial Mechanics, New York, Macmillan, 1964.

10. Tinling, B.E. and V.K. Merrick, "Exploitation of Inertial Coupling in Passive Gravity-Gradient-Stabilized Satellites", AIAA J. Spacecraft and Rockets 1, 381-387 (1964).

11. Rimrott, F.P.J., "Storable Tubular Extendible Member", Machine Design 37, 28, 156-165 (1965).

12. Newton, J.K. and J.L. Farrell, "Natural Frequencies of a Flexible Gravity Gradient Satellite", to appear in AIAA Journal of Spacecraft and Rockets.

13. Wylie, C.R., Advanced Engineering Mathematics, New York, McGraw-Hill, 1951, p. 509.

14. Dow, P.C. et.al., "Dynamic Stability of a Gravity Gradient Stabilized Satellite Having Long Flexible Antennas" presented at AIAA Guidance and Control Specialists Conference, Seattle, Washington, Aug. 15-17, 1966.

NASA-Langley, 1968 —— 30 CR-1119