NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-634

RUSAP–A Computer Program for the Calculation of Roll-up Solar Array Performance Characteristics

R. G. Ross, Jr.
J. V. Coyner, Jr.

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1973

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-634

RUSAP–A Computer Program for the Calculation of Roll-up Solar Array Performance Characteristics

R. G. Ross, Jr.
J. V. Coyner, Jr.

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1973

PREFACE

The work described in this report was performed by the Applied Mechanics Division of the Jet Propulsion Laboratory.

CONTENTS

TABLES

FIGURES

FIGURES (contd)

ABSTRACT

RUSAP is a FORTRAN IV computer program designed to determine the performance characteristics (power-to-weight ratio, blanket tension, structural member section dimensions, and resonant frequencies) of large-area, roll-up solar arrays of the single-boom, tensioned-substrate design. The program includes the determination of the size and weight of the base structure supporting the boom and blanket and the determination of the blanket tension and deployable boom stiffness needed to achieve the minimum-weight design for a specified frequency for the first mode of vibration. This report provides a complete listing of the program, a description of the theoretical background, and all information necessary to use the program.

I. INTRODUCTION

In recent years, considerable emphasis has been placed on the development of large-area solar arrays with high power-to-weight ratios and small packaging volumes. One of the concepts currently being developed is the single-boom roll-up array shown in Fig. 1 and described in Ref. 1.

The design consists of two flexible cell blankets tensioned between spacecraft-mounted storage drums and a leading-edge beam. The array is erected by a deployable boom that is connected between the leading-edge beam and its supporting structure on the spacecraft.

In studying potential applications for this design, one must predict the performance characteristics of arrays of widely varying sizes and natural frequencies. Because of the complex nature of the relationship between the size of the array, its first-mode natural frequency, and its structural parameters, it is difficult to predict the performance characteristics of arrays significantly different from the engineering prototype. To help solve this problem, a computer routine RUSAP was programmed to calculate the optimum size for structural members and the optimum blanket tension for an array that is to have a given area and width and a specified lowest deployed natural frequency. The program is composed of two basic parts: (1) an analysis of the deployable boom and array blanket, and (2) an analysis and sizing of the base structure supporting the boom and array blankets.

II. THEORETICAL BACKGROUND

A. Optimum Boom and Blanket Tension Analysis

If the power, area, and width of the array are specified, the length (and thus the size) of the array is fixed. Because the weight of the cell blanket and support structure is essentially fixed for an array of a given size, the relationship between total weight and deployed natural frequency is almost entirely controlled by the weight of the deployable boom. This implies that, when the size of the array is specified, the optimum array for a specific lowest deployed natural frequency can be defined as that with the lightest boom.

Because there are two possible first-vibration modes for a deployed array (Fig. 2), the cross-sectional size (and thus the weight) of the boom is determined by the following requirement: the boom must have sufficient stiffness to maintain the first-bending frequency equal to or greater than the required minimum frequency while loaded by sufficient blanket tension to maintain the first-torsion frequency at or above the minimum. Figure 3 shows the variation in first-bending and first-torsion frequencies as a function of the blanket tension for a single-boom array. As is shown in this figure, the optimum tension for a particular boom stiffness has been found to occur when the first symmetric and antisymmetric frequencies are equal. This tension yields the lightest boom for a particular lowest deployed natural frequency.

It can be seen in Fig. 3 that the first-mode frequency falls off relatively sharply below the maximum frequency point, but remains relatively flat for total blanket tensions up to approximately 50% of the boom-buckling load. Because of this phenomena, the blanket tension may be increased above the maximum frequency point, thus allowing the user to increase the antisymmetric first-mode frequency while only slightly decreasing the symmetric first mode. This tension increase has a minimal affect on the overall weight of the roll-up array and allows the user to operate on the flat part of the symmetric frequency curve, where variations in the blanket tension have little effect on the array frequency.

The program RUSAP has provisions for calculating the performance parameters for any selected boom and blanket tension, and, in addition, has provisions for calculating the optimum blanket tension. The optimum blanket tension is defined by the condition that symmetric and antisymmetric frequencies are equal. Each deployed array analysis requires the calculation of the natural frequencies of the combined tensioned-blanket/axial-loaded boom system that makes up the deployed array. In addition to the optimum tension analysis, the program can also be used to determine the boom stiffness that results in the lowest deployed frequency being equal to a desired minimum deployed natural frequency.

The modal analysis of the deployed solar array is based on a multiple-degree-of-freedom, finite-element representation of the boom, beam, and blanket components, as shown in Fig. 4. Each blanket is modeled by 10 rectangular finite elements, which describe the out-of-plane stiffness caused by the imposed blanket tension. The boom and beam components are similarly modeled by standard beam-column elements, which describe both the bending stiffness and the geometric stiffness caused by the axial preload. Although the blanket bending stiffness is neglected in the above analysis, more sophisticated 250-degree-of-freedom analyses, which include the bending stiffness, have shown that the simple model leads to less than 1% errors in the first-mode frequencies.

The stiffness and mass matrices for the overall array are developed by combining the element stiffness and consistent mass matrices for the elements described above. The generation procedure allows for the following arbitrary parameters:

(1) Array length.

(2) Array width.

(3) Blanket weight/unit area.

(4) Boom weight/unit length.

(5) Beam weight/unit length.

(6) Boom stiffness.

(7) Beam stiffness.

(8) Blanket tension.

After initialization of the above parameters by the root-finding routines, the natural frequencies of the array are determined by solution of the usual eigenvalue problem with a very fast eigenvalue algorithm.

B. Support Structure Analysis

The support structure analysis uses scaling equations to extrapolate the size and weight of the support-structure components of a proposed array from those of a prototype array. The configuration used as the baseline for this analysis is the 23 m^2 (250 ft^2) engineering prototype described in Ref. 1. Essentially, a dimensional-analysis approach was taken to determine the scale factors that are applied to the structural elements when design conditions change. The total weight W_1 of the reference array is broken down into 12 components:

$$W_1 = W_{bo1} + W_{s1} + W_{c1} + W_{b1} + W_{ss1} + W_{es1} + W_{cs1} + W_{be1} + W_{a1} + W_{ns1} + W_{bkt1} + W_{neg1} \tag{1}$$

where

W_{bo1} = weight of boom (determined by modal analysis program)

W_{s1} = weight of storage drum shell

W_{c1} = weight of end caps on storage drum shell

W_{b1} = weight of bearings

W_{ss1} = weight of support shaft

W_{es1} = weight of end supports

W_{cs1} = weight of center support

W_{be1} = weight of beam (leading-edge member)

W_{a1} = weight of boom actuator

W_{ns1} = weight of slip rings and harness

W_{bkt1} = weight of solar-array blanket

W_{neg1} = weight of NEG'ATOR spring-mechanism hardware (constant-force spring)

(Subscript 1 refers to the prototype reference array; the alphabetic subscripts refer to the structural components.)

The weight of a second array – differing from the reference array in geometry, material properties, and applied inertial loads – can be broken down in a similar manner:

$$W_2 = W_{bo2} + W_{s2} + W_{c2} + W_{b2} + W_{ss2} + W_{es2} + W_{cs2} + W_{be2} + W_{a2} + W_{ns2} + W_{bkt2} + W_{neg2} \tag{2}$$

The ratio $W_t = W_2/W_1$ can be written as

$$\begin{aligned} W_t = \frac{W_2}{W_1} = {} & \frac{W_{bo2}}{W_{bo1}}\frac{W_{bo1}}{W_1} + \frac{W_{s2}}{W_{s1}}\frac{W_{s1}}{W_1} + \frac{W_{c2}}{W_{c1}}\frac{W_{c1}}{W_1} + \frac{W_{b2}}{W_{b1}}\frac{W_{b1}}{W_1} + \frac{W_{ss2}}{W_{ss1}}\frac{W_{ss1}}{W_1} \\ & + \frac{W_{es2}}{W_{es1}}\frac{W_{es1}}{W_1} + \frac{W_{cs2}}{W_{cs1}}\frac{W_{cs1}}{W_1} + \frac{W_{be2}}{W_{be1}}\frac{W_{be1}}{W_1} + \frac{W_{a2}}{W_{a1}}\frac{W_{a1}}{W_1} + \frac{W_{ns2}}{W_{ns1}}\frac{W_{ns1}}{W_1} \\ & + \frac{W_{bkt2}}{W_{bkt1}}\frac{W_{bkt1}}{W_1} + \frac{W_{neg2}}{W_{neg1}}\frac{W_{neg1}}{W_1} \end{aligned} \tag{3}$$

where W_{bo1}/W_1, W_{s1}/W_1, W_{c1}/W_1, etc., are the fractions of the total weight of the reference array contributed by the individual components. A simplified notation is used to refer to the ratios of the weights of the components, $W_{bo2}/W_{bo1} = W_{bo}$, $W_{s2}/W_{s1} = W_s$, $W_{c2}/W_{c1} = W_c$, etc.

Geometric scale factors are introduced to define the change in size of the components, and relationships are then established between these geometric scale factors and the ratio of stresses in the components. The scale factors used are as follows:

λ^{so} = structural section overall dimension scale factor

λ^{st} = structural section material thickness scale factor

λ_h = blanket width scale factor

λ_h' = array width scale factor

λ_v = array length scale factor

λ_t = ratio of blanket tension of array 2 to that of array 1

λ_{acc} = ratio of launch-acceleration loading of array 2 to that of array 1

λ_{dia} = ratio of boom diameter of array 2 to that of array 1

λ_{ves} = ratio of total diameter of combined storage drum shell and rolled blanket of array 2 to that of array 1

Because only one acceleration load factor λ_{acc} relates the acceleration loadings in the three orthogonal directions (parallel to the storage drum, perpendicular to the storage drum and in the plane of the blanket, and perpendicular to the storage drum and normal to the plane of the blanket), this factor is an average of these three orthogonal acceleration factors.

By expressing the weight ratios in terms of material densities and volumes, and by expressing the volumes in terms of the scale factors, Eq. (3) becomes

$$
\begin{aligned}
W_t = {} & \frac{W_{bo1}}{W_1}\left[W_{bo}\right] + \frac{W_{s1}}{W_1}\left[\frac{\rho_{s2}}{\rho_{s1}}\left(\lambda_s^{so}\,\lambda_s^{st}\,\lambda_h\right)\right] + \frac{W_{c1}}{W_1}\left[\frac{\rho_{c2}}{\rho_{c1}}\left(\lambda_s^{so}\right)\left(\lambda_c^{st}\right)^2\right] \\
& + \frac{W_{b1}}{W_1}\left[\frac{\rho_{b2}}{\rho_{b1}}\left(\lambda_{ss}^{so}\right)^2\right] + \frac{W_{ss1}}{W_1}\left[\frac{\rho_{ss2}}{\rho_{ss1}}\left(\lambda_{ss}^{so}\,\lambda_{ss}^{st}\,\lambda_h\right)\right] \\
& + \frac{W_{es1}}{W_1}\left[\frac{\rho_{es2}}{\rho_{es1}}\left(\lambda_{es}^{so}\,\lambda_{es}^{st}\,\lambda_{ves}\right)\right] + \frac{W_{cs1}}{W_1}\left[\frac{\rho_{cs2}}{\rho_{cs1}}\left(\lambda_{dia}\,\lambda_{cs}^{st}\,\lambda_{ves}\right)\right] \\
& + \frac{W_{be1}}{W_1}\left[\frac{\rho_{be2}}{\rho_{be1}}\left(\lambda_{be}^{so}\,\lambda_{be}^{st}\,\lambda_h'\right)\right] + \frac{W_{a1}}{W_1}\left[\frac{\rho_{a2}}{\rho_{a1}}\left(\lambda_{dia}\,\lambda_a^{st}\,\lambda_v^{1/2}\right)\right] \\
& + \frac{W_{ns1}}{W_1}\left[\frac{\rho_{ns2}}{\rho_{ns1}}\left(\lambda_h'\lambda_v\right)\right] + \frac{W_{bkt1}}{W_1}\left[\frac{\rho_{bkt2}}{\rho_{bkt1}}\lambda_h\lambda_v\right] \\
& + \frac{W_{neg1}}{W_1}\left[\frac{\rho_{neg2}}{\rho_{neg1}}\left(\lambda_t\right)^{1/2}\right] \qquad (4)
\end{aligned}
$$

where ρ is the material density of the components, the subscripts on the geometric scale factors and densities indicate the structural components to which they refer, and the superscripts st and so indicate whether the scale factor refers to thickness or to overall size, respectively.

Relationships are then established between the geometric scale factors in Eq. (4) and the ratios of stresses of the components of the structure. This was done individually for each of the components sized by stress and load levels, and the results substituted into Eq. (4). Some components, however, are not sized by stress and load levels.

The ratio of the boom weights W_{bo} is determined by the modal analysis program. The NEG'ATOR spring-mechanism weight is a function of blanket tension. Blanket weight is a function of λ_h and λ_v. Slip ring and harness hardware is also assumed to be a function of λ_h and λ_v, and bearing size is a function of support-shaft size.

The derivation of the relationships for the storage drum shell will be considered as a typical example. For inertial loading, beam-bending moments are related by

$$\frac{M_2}{M_1} = \lambda_{acc} \lambda_h W_{bs} \tag{5}$$

where $W_{bs} = (W_{bkt2} + W_{s2})/(W_{bkt1} + W_{s1})$. The corresponding ratio of shell maximum bending stresses is

$$\frac{\sigma_{s2}}{\sigma_{s1}} = \frac{M_2 C_2 I_1}{M_1 C_1 I_2} \tag{6}$$

where the shell-section moments of inertia are related by

$$\frac{I_2}{I_1} = \left(\lambda_s^{so}\right)^3 \lambda_s^{st} \tag{7}$$

The ratio of the maximum bending stresses becomes

$$\frac{\sigma_{s2}}{\sigma_{s1}} = \frac{\lambda_{acc} \lambda_h W_{bs}}{\left(\lambda_s^{so}\right)^2 \left(\lambda_s^{st}\right)} \tag{8}$$

Critical buckling stresses for the shell are related by

$$\frac{\sigma_{sb2}}{\sigma_{sb1}} = \frac{\left(\lambda_s^{st}\right)^2 E_{s2}}{\left(\lambda_s^{so}\right)^2 E_{s1}} \tag{9}$$

Two additional relationships are assumed by the requirement that buckling and bending stress ratios be equally critical and that the bending-stress ratio be a function of the yield-strength ratio,

$$\frac{\sigma_{sb2}}{\sigma_{sb1}} = \frac{\sigma_{s2}}{\sigma_{s1}} \tag{10}$$

and

$$\frac{\sigma_{s2}}{\sigma_{s1}} = K_s \frac{\sigma_{ys2}}{\sigma_{ys1}} \tag{11}$$

where σ_{ys2} is the yield strength of the shell for array 2, σ_{ys1} is that for array 1, and K_s is the ratio of the factors of safety of the two arrays. These equations (5 through 11) are then solved for the unknowns λ_s^{st}, λ_s^{so}, and W_{bs}, and substituted into Eq. (4).

Table 1 lists the relationships established between the geometric scale factors in Eq. (4) and the ratios of the stresses for all of the components, where

$$W_{bscss} = \frac{W_{bkt2} + W_{s2} + W_{c2} + W_{ss2}}{W_{bkt1} + W_{s1} + W_{c1} + W_{ss1}}$$

$$W_{ba} = \frac{W_{bo2} + W_{a2}}{W_{bo1} + W_{a1}}$$

$$W_{bscssa} = \frac{W_{bkt2} + W_{s2} + W_{c2} + W_{ss2} + W_{a2} + W_{bo2} + W_{b2} + W_{neg2} + W_{ns2}}{W_{bkt1} + W_{s1} + W_{c1} + W_{ss1} + W_{a1} + W_{bo1} + W_{b1} + W_{neg1} + W_{ns1}}$$

$$\lambda_{ves} = \left\{\lambda_v + \left[\left(\lambda_s^{so}\right)^2 - \lambda_v\right] \frac{r_i^2}{r_o^2}\right\}^{1/2}$$

and

r_i = radius of shell of reference array

r_o = radius of combined shell and rolled blanket for reference array

The equations are then solved for the unknowns λ_s^{so}, λ_s^{st}, λ_c^{st}, λ_{ss}^{so}, λ_{ss}^{st}, λ_{es}^{so}, λ_{es}^{st}, λ_{cs}^{st}, λ_{be}^{so}, λ_{be}^{st}, λ_a^{st}, W_{bs}, W_{bscss}, and W_{ba}. The ratios of the weights of the individual components are then determined by direct substitution of these values into Eq. (4). Table 2 lists the solutions for the above scale factors.

To determine W_{bs}, W_{bscss} and W_{ba}, three additional equations (12 through 14) were written and solved for these three unknowns. To solve these equations, the scale-factor solutions must first be substituted for the scale factors so that the only unknown in each equation is either W_{bs}, W_{bscss}, or W_{ba}:

$$W_{bs} = \frac{W_{bkt1}}{W_{bs1}}\left(\lambda_h \lambda_v \frac{\rho_{bkt2}}{\rho_{bkt1}}\right) + \frac{W_{s1}}{W_{bs1}}\left[\frac{\rho_{s2}}{\rho_{s1}}\left(\lambda_s^{so}\ \lambda_s^{st}\ \lambda_h\right)\right] \tag{12}$$

where W_{bs1} is the weight of the blanket and storage drum shell of the reference array;

$$\begin{aligned} W_{bscss} &= \frac{W_{bs1}}{W_{bscss1}}(W_{bs}) + \frac{W_{c1}}{W_{bscss1}}\left[\frac{\rho_{c2}}{\rho_{c1}}\left(\lambda_s^{so}\right)\left(\lambda_c^{st}\right)^2\right] \\ &\quad + \frac{W_{ss1}}{W_{bscss1}}\left[\frac{\rho_{ss2}}{\rho_{ss1}}\left(\lambda_{ss}^{so}\right)\left(\lambda_{ss}^{st}\right)\lambda_h\right] \end{aligned} \tag{13}$$

where W_{bscss1} is the weight of the blanket, storage drum shell, end caps, and support shaft of the reference array; and

$$W_{ba} = \frac{W_{bo1}}{W_{ba1}}(W_{bo}) + \frac{W_{a1}}{W_{ba1}}\left(\frac{\rho_{a2}}{\rho_{a1}}\lambda_{dia}\lambda_a^{st}\lambda_v^{1/2}\right) \tag{14}$$

where W_{ba1} is the weight of the boom and the actuator of the reference array. Once these equations have been solved for the W's, and all scale factors have been determined, the final weight is calculated and the power-to-weight ratio is obtained.

This technique for analyzing the size and weight of a structure has inherent limitations. As in all parametric studies, arbitrary decisions have to be made as to how to describe the relationships between load, stress, and member size. Because of these limitations, the results obtained from the support-structure analysis are not intended as a substitute for a complete and detailed structural analysis, but as a good initial estimate of the sizing and weight of an array (given a required array area).

III. PROGRAM USAGE

A. Selection of Program Levels

RUSAP is designed to calculate the design parameters of roll-up solar arrays of the single-boom, two-drum, tensioned blanket configuration. The calculated parameters are power-to-weight ratio, blanket tension, boom stiffness, structural member weights, first-mode resonant frequencies, actuator width, and required array length and width.

The computer program has three levels of operation, which are listed below. Block diagrams for the three levels are shown in Figs. 5, 6, and 7.

Level 1: After the user inputs the desired first-mode frequency and all other required input parameters, the program calculates the optimum boom stiffness and optimum blanket tension that gives the required first-mode frequency. It then calculates weights and sizes of all base structural members.

Level 2: After the user inputs the desired boom stiffness and all other required input parameters, the program calculates the optimum blanket tension that maximizes the first-mode frequency of the deployed array. It then calculates weights and sizes of all base structural members.

Level 3: After the user inputs the desired boom stiffness, blanket tension, and all other required input parameters, the program calculates the deployed symmetric and antisymmetric first-mode frequencies. It then calculates weights and sizes of all base structural members.

There is a major difference between Level 1 and Levels 2 and 3. When operating Level 1, the user specifies the required frequency, and the program picks many boom stiffnesses in the process of establishing the optimum stiffness for the required natural frequency. Each time the program tries a new boom, it must also determine the optimum tension for that boom. Therefore, each boom stiffness iteration during Level 1 is equivalent to running one complete case at Level 2. If the user does not need an exact frequency, he can save computer time by running Level 2 after estimating boom stiffnesses that will bound his desired frequency. The user can determine these boom stiffness estimates by referring to Figs. A-9 through A-12 of Ref. 2. However, if the user does desire an exact natural frequency, he can save computer time when using Level 1 if he makes as accurate an estimate of the optimum boom stiffness as possible using the data in Ref. 2.

Using the UNIVAC 1108 Exec 8 computer system, RUSAP requires approximately 20 seconds to execute Level 1, 5 seconds to execute Level 2, and 1 second to execute Level 3.

B. Minimum Tension and Boom Buckling Considerations

When operating Level 1, the program establishes the smallest boom compatible with the natural frequency requirement when the first-mode frequency is maximized by using the optimum blanket tension. When the required natural frequency is very low, the optimum blanket tension may be lower than the minimum tension required to roll the blanket on the drum during retraction. The minimum tension for conventional substrates with bonded-on solar cells is defined approximately by the following empirical relationship:

$$\frac{\text{tension per blanket}}{\text{blanket width}} \geq 7.2 \text{ N/m } (0.5 \text{ lb/ft})$$

A warning message is printed in the program output whenever the above criterion is not met.

If a minimum-weight array is desired and the optimum tension is below the minimum tension, the user has a number of options. One is to develop a more flexible solar-cell blanket or to add a positive motor drive for array retraction. Another option is to set the blanket tension equal to the minimum and then to increase the size of the boom until the boom is not in danger of buckling. The critical buckling load for the boom is derived in Appendix C of Ref. 3 and is given by:

$$P_{\text{critical}} = \frac{\pi^2 EI}{L^2}$$

where the symbols and units are the usual ones for column buckling. Recall that a plot of first-mode frequency vs percent of buckling load is given earlier in Fig. 3. When a particular tension is selected, the user can obtain the array that operates at the desired percent of buckling by using the LEVEL = -3 option of the program with the PCTBKL parameter negative. In this case the program uses the following expression to define the boom EI:

$$\text{boom EI} = \frac{200 \times (\text{blanket tension}) \times (\text{array length})^2}{\pi^2 \times (\text{percent buckling})}$$

The LEVEL = -3 option is also useful when the user wishes to operate at a tension somewhat higher than the optimum tension to reduce the sensitivity of the first-mode frequency to small changes in blanket tension. This philosophy is discussed earlier in Subsection II-A. After running Level 1 to determine the optimum boom, the user can use LEVEL = -3 with the PCTBKL parameter positive to obtain the array with the tension given by:

$$\text{blanket tension} = \frac{\pi^2 \times (\text{boom EI}) \times (\text{percent buckling})}{200 \times (\text{array length})^2}$$

Note that the program always prints the boom buckling load and the percent of buckling operating point in the output listing of deployed array parameters.

C. Array Geometry Input Options

In all levels of the program there are two ways of inputing the geometric requirements of the solar array. The first allows the user to specify the total blanket area and array width, whereas the second allows the user to specify the width and length of the individual blankets. The two requirement formats are primarily associated with the stage at which the program is being applied in the design process.

The total blanket area and array width are suited for use early in the design process when the primary requirements are configuration constraints on the array width and desired power output, which implies a total blanket area. Later in the design process specific solar-cell module and bus-bar layouts lead to detailed specifications on the dimensions of the individual blankets. At this point the second format is more suitable.

With both of the geometric requirement input formats it is necessary for the program to determine some of the array dimensions. This is because the size of the boom affects the allowable spacing between the blankets. When the blanket area and array width are specified, the boom size determines the blanket width and the array length, whereas when the blanket length and blanket width are specified, the boom size determines the array width.

Since the actuator width, and therefore the width between blankets, is a function of the boom diameter, the program calculates an appropriate actuator width for each boom stiffness. The equation used is:

$$\text{actuator width} = \left(\frac{\text{proposed boom diameter}}{\text{prototype boom diameter}}\right) \times \left(\begin{matrix}\text{prototype}\\ \text{actuator width}\end{matrix}\right)$$

When the user inputs the total blanket area and the array width, the program calculates the blanket width and array length using

$$\text{blanket width} = \frac{(\text{array width - actuator width})}{2}$$

$$\text{array length} = \left[\frac{\text{blanket area}}{\text{array width - actuator width}}\right] + \text{blanket leader}$$

In the above expression the blanket leader is the total additional length of the substrate material extending beyond the celled blanket, and used to connect the blanket to the drum and leading edge member (Fig. 1).

When the user inputs the blanket width and blanket length, the equations used for calculating array width and array length are:

$$\text{array width} = 2 \times (\text{blanket width}) + \text{actuator width}$$

$$\text{array length} = \text{blanket length} + \text{blanket leader}$$

When choosing the length and width for a proposed array it should be noted that the ratio of length to width, or aspect ratio of the array, influences the array weight. In an earlier study (Ref. 2), the optimum aspect ratio was found to be primarily determined by the required lowest natural frequency of the array, and essentially independent of the array size.

In addition to the aspect ratio and first mode natural frequency there are many other parameters that affect the weight and cost of a proposed array. Many of these parameters may be varied by the user and are described in the next section.

IV. DATA PREPARATION

RUSAP is a main program designed to be used in batch mode. The input parameters that define the roll-up array's size, material properties, and desired first-mode natural frequency are read in from data cards using namelist names DATA1 and DATA2. Any number of array configurations can be generated in a single computer run by including one or more data sets in the run stream. A data set consists of a DATA1 group and its associated DATA2 group(s). A data group is the collection of cards that define the parameters associated with namelist name.

The FORTRAN name rule, by which any symbolic name that begins with I, J, K, L, M, or N is automatically an integer, is assumed.

A. NAMELIST/DATA1 General Parameters

NAMELIST/DATA1/ contains twenty parameters that may be assigned values in any DATA1 group. All of the parameters must be assigned a value in the first DATA1 group. Since the parameter assignments of previous data groups are used if not reassigned, DATA1 groups following the first need contain only those parameter assignments that are changes from the previous parameter assignments. However, it should be noted that the program automatically updates the parameters BKTL, BKTW, AREA, WIDTH, EIBOM1, and TENS at the end of each case by setting them equal to the "Deployed Array Parameters," which are printed in the output. These updated parameter values are used in the following case unless the user specifies new values.

The following parameters belong to NAMELIST/DATA1/ and may be included in any order. Data may be in either SI or English units as selected using the parameter NUNITS.

NSETS: an integer to be used in the first DATA1 group to indicate the number of cases to be run. RUSAP will read NSETS data sets including the first.

NUNITS: an integer used to define the input units: 0 for English units and 1 for SI units.

LEVEL: an integer used to set the level of operation of the program. There are three levels available to the user (1, 2, 3). The operation of the three levels is described in Subsection III-A. For Level 3 the sign of LEVEL is used to select between input parameter options. For LEVEL = 3 the parameters EIBOM1 and TENS are used to define the array. For LEVEL = -3 the array is defined by the parameters PCTBKL and either EIBOM1 or TENS depending on the sign of PCTBKL.

POWER: the required power output of the roll-up array; used to calculate the specific power in W/kg (W/lb). (Units are watts.)

FREQR: the required first-mode natural frequency. (Units are Hertz.) (Level 1 only.)

NTYPE: an integer used to select the array fixed dimensions. If NTYPE = 0, BKTL and BKTW are used. If NTYPE = 1, AREA and WIDTH are used.

BKTL: the required blanket length. (Units are meters or feet.) (Required when NTYPE = 0.)

BKTW: the required blanket width for each blanket. (Units are meters or feet.) (Used when NTYPE = 0.)

AREA: the total blanket area of the array. (Units are meters^2 or feet^2.) (Used when NTYPE = 1.)

WIDTH: the total width of the array. (Units are meters or feet.) (Used when NTYPE = 1.)

TLEADR: the total inboard plus outboard blanket leader length. Typical value is 0.3 m (1.0 ft). (Units are meters or feet.)

PCTBKL: a Level 3 parameter used to obtain an array that operates at a given percent of buckling. The absolute value of PCTBKL is the desired percent of buckling. The sign of PCTBKL is used to select between TENS and EIBOM1 as the other deployed array stiffness parameter. When PCTBKL is positive the array is defined by PCTBKL and EIBOM1; when negative, PCTBKL and TENS are used. (LEVEL = -3 only.)

TENS: the tension in each blanket. (Units are newtons or pounds.) (LEVEL = 3 or -3 only.)

EIBOM1: the stiffness (EI) of the deployable boom. For Level 1, this input boom stiffness is used as a starting value for the boom optimization. For Levels 2 and 3, the stiffness is used as the required boom stiffness. (Units are Newton-meters^2 or pound-feet^2.)

EBOOM: the deployable boom material modulus of elasticity (E). The value for steel is 0.2×10^{12} N/m^2 (0.4175×10^{10} lb/ft^2). (Units are Newtons/meter^2 or pounds/foot^2.)

RBOOM: the deployable boom material density. The value for steel is 7930 kg/m^3 (495 lb/ft^3). (Units are kilograms/meter^3 or pounds/foot^3.)

EIEFF: the structural efficiency of the deployable boom. Since the BI-STEM boom is split into two wrapped, open-tube sections, the effective section moment of inertia (I) is less than that of a closed tube with the same cross-section dimensions. Typical value is 0.80 efficiency. (Unitless quantity.)

DPT: the diameter of the deployable boom divided by the boom material thickness. The boom wall thickness is twice the material thickness. Typical value is 200. (Unitless quantity.)

ROARAY: the average weight per unit area of the array blankets. Typical value is 0.878 kg/m^2 (0.18 lb/ft^2). (Units are $kilograms/meter^2$ or $pounds/foot^2$.)

NDATA2: an integer that specifies how many DATA2 groups are included in the data set. 0 causes no base structure to be analyzed, 1 reads one DATA2 group, 2 reads two DATA2 groups, etc.

B. NAMELIST/DATA2 General Parameters

NAMELIST/DATA2/ contains the following 19 parameters that may be included in any DATA2 group in any order. They are all ratios of the array component properties in the proposed array to those in the prototype array. These parameters are all assigned nominal values of 1.00 prior to reading the first DATA2 group in the first data set. Since the parameter assignments of previous DATA2 groups are used if not reassigned, DATA2 groups following the first need contain only those parameter assignments that are changes from those in the previous DATA2 group. This is true even when the previous DATA2 group is in a previous data set. Note, however, that the number of DATA2 groups in each data set must equal NDATA2, even if the DATA2 parameters are a repeat of those in the previous data set.

RHOS: the ratio of the material density of the storage drum in the proposed array to that in the prototype array.

ES: the ratio of the modulus of elasticity of the storage drum in the proposed array to that in the prototype array.

SIGMS: the ratio of the yield strength stress of the storage drum in the proposed array to that in the prototype array.

RHOC: the ratio of the material density of the storage drum end caps in the proposed array to that in the prototype array.

EC: the ratio of the modulus of elasticity of the storage drum end caps in the proposed array to that in the prototype array.

SIGMC: the ratio of the yield strength stress of the storage drum end caps in the proposed array to that in the prototype array.

RHOSS: the ratio of the material density of the drum support shaft in the proposed array to that in the prototype array.

ESS: the ratio of the modulus of elasticity of the drum support shaft in the proposed array to that in the prototype array.

SIGMSS: the ratio of the yield strength stress of the drum support shaft in the proposed array to that in the prototype array.

RHOBE: the ratio of the material density of the leading edge beam in the proposed array to that in the prototype array.

EBE: the ratio of the modulus of elasticity of the leading edge beam in the proposed array to that in the prototype array.

SIGMBE: the ratio of the yield strength stress of the leading edge beam in the proposed array to that in the prototype array.

RHOES: the ratio of the material density of the drum end supports in the proposed array to that in the prototype array.

EES: the ratio of the modulus of elasticity of the drum end supports in the proposed array to that in the prototype array.

SIGMES: the ratio of the yield strength stress of the drum end supports in the proposed array to that in the prototype array.

RHOCS: the ratio of the material density of the center support in the proposed array to that in the prototype array.

ECS: the ratio of the modulus of elasticity of the center support in the proposed array to that in the prototype array.

SIGMCS: the ratio of the yield strength stress of the center support in the proposed array to that in the prototype array.

VLAMAC: a factor used to scale the launch vibration level from that set within the program. The factor λ_{acc}, the ratio of the launch acceleration loading of the proposed array to the loading used in the prototype array, is used in calculating array stresses and member sizes (Subsection II-B). The factor λ_{acc} is defined by:

$$\lambda_{acc} = \text{VLAMAC} \times \left(\frac{\text{Total weight of prototype array}}{\text{Total weight of proposed array}} \right)^{1/2}$$

The above square root factor implements a general rule of thumb that the launch acceleration loading for a subsystem on a given launch vehicle varies with the square root of the subsystem's mass. In other words, a small light subsystem sees a higher excitation level than a large heavy one on the same spacecraft. VLAMAC is used to change this relationship to account for different acceleration levels associated with different spacecraft/launch-vehicle combinations. The levels used on the reference array are described in Figs. 8 and 9.

The material properties of the individual components in the prototype array are listed in Table 3. They are to be used when calculating the parameter ratios previously described.

V. EXAMPLES AND SAMPLE OUTPUT

To demonstrate the program's use, three example problems are considered in Table 4. Note that Case 1 has two $DATA2 cards associated with two different base structure analyses.

The following data cards were used to input the above cases:

```
$DATA1  NSETS=3,NUNITS=1,LEVEL=1,POWER=2500.,FREQR=.077,NTYPE=1,AREA=23.225,
WIDTH=2.516,TLEADR=.305,EIBOM1=2000.,EBOOM=0.2E+12,RBOOM=7930.,EIEFF=.80,
DPT=200.,ROARAY=.878,NDATA2=2$
$DATA2  VLAMAC=.5$
$DATA2  VLAMAC=1.$
$DATA1  LEVEL=2,EIBOM1=1500.,NDATA2=1$
$DATA2  VLAMAC=1.$
$DATA1  NUNITS=0,LEVEL=-3,POWER=10000.,NTYPE=0,BKTL=81.46,BKTW=6.16,TLEADR=1.
PCTBKL=-50.,TENS=3.25,EBOOM=.418E+10,RBOOM=495.,ROARAY=.18$
$DATA2  VLAMAC=1.$
```

A. Description of Example Cases

Case 1 is an example of Level 1. The program determines the minimum weight design for the given size and natural frequency. Case 2 is an example of Level 2. The program uses the desired boom stiffness and determines the blanket tension that maximizes the frequency. Case 3 is an example of Level 3. The user specifies the PCTBKL and blanket tension, and the program calculated the boom stiffness and the first mode symmetric and antisymmetric frequency. This run is typical of a design with a configuration for which the optimum boom requires an unacceptably low blanket tension. In other words, the optimum blanket tension is too low to roll up the blanket during array retraction. A minimum weight design is obtained by setting the tension equal to the minimum recommended tension of 7.2 N/m (0.5 lb/ft) and increasing the boom stiffness until the boom is not in danger of buckling (Subsection III-B). In this case the boom is selected to operate at 50% of its buckling load.

B. Typical Level 1 Printout

```
****************************************************************************
****************************************************************************
*****     POWER= 2500. WATTS     AREA=  23.2 M**2      WIDTH= 2.516 M      *****
*****     REQUIRED FREQUENCY=  .077 HZ                                     *****
****************************************************************************
****************************************************************************


****INPUT PARAMETERS FOR DEPLOYED ROLL-UP SOLAR ARRAY****

REQUIRED POWER   =     2500. WATTS

BLANKET AREA     =     250.0 FT**2        (   23.22 M**2)

ARRAY WIDTH      =      8.25 FT           (   2.516 M)

LEADER LENGTH    =      1.00 FT           (    .305 M)

BLANKET UNIT WT. =      .180 LB/FT**2     (     .88 KG/M**2)

REQUIRED FREQ    =      .077 HZ

INITIAL BOOM (EI)=     4837. LB FT**2     (   2000. NM**2)

BOOM MODULUS (E) =  .4175+10 LB/FT**2     ( .200+12 N/M**2)

BOOM MATL DENSITY=     495.0 LB/FT**3     (   7930. KG/M**3)

BOOM EFFICIENCY  =       .80 EFFECTIVE (I)

DIA TO THICKNESS =      200. IN/IN

          ****PRINTOUT OF PROGRAM OPTIMIZATION ITERATIONS****
                     (ENGLISH UNITS-LB,FT,SEC,HZ)


    TENSION OPTIMIZATION ITERATIONS
     I       TENSION      BOOM (EI)      FREQ DIF       SYM FREQ      ASYM FREQ
     1     .41857+00     .48368+04     .29618-01     .62013-01     .32395-01
     2     .30247+01     .48368+04     .20164-01     .10710+00     .86940-01
     3     .30247+01     .48368+04     .20164-01     .10710+00     .86940-01
     4     .48027+01     .48368+04    -.41058-03     .10862+00     .10903+00
     5     .47672+01     .48368+04    -.30647-04     .10873+00     .10876+00
                                               END TENSION OPTIMIZATION
```

```
BOOM STIFFNESS OPTIMIZATION ITERATIONS
 I        TENSION       BOOM (EI)       FREQ DIF        FREQ          FREQ REQ
 0      .47672+01      .48368+04      .31743-01      .10874+00      .77000-01*

    TENSION OPTIMIZATION ITERATIONS
      I         TENSION        BOOM (EI)        FREQ DIF        SYM FREQ        ASYM FREQ
      1        .44867+00       .51995+04        .30371-01       .64153-01       .33782-01
      2        .32131+01       .51995+04        .21199-01       .11057+00       .89373-01
      3        .32131+01       .51995+04        .21199-01       .11057+00       .89373-01
      4        .51614+01       .51995+04       -.76944-03       .11215+00       .11292+00
      5        .50932+01       .51995+04        .13062-03       .11237+00       .11224+00
                                                          END TENSION OPTIMIZATION

    TENSION OPTIMIZATION ITERATIONS
      I         TENSION        BOOM (EI)        FREQ DIF        SYM FREQ        ASYM FREQ
      1        .10953+00       .12092+04        .15541-01       .32114-01       .16573-01
      2        .79294+00       .12092+04        .11377-01       .56571-01       .45194-01
      3        .79294+00       .12092+04        .11377-01       .56571-01       .45194-01
      4        .13099+01       .12092+04       -.23155-03       .57842-01       .58073-01
                                                          END TENSION OPTIMIZATION

BOOM STIFFNESS OPTIMIZATION ITERATIONS
 I        TENSION       BOOM (EI)       FREQ DIF        FREQ          FREQ REQ
 1      .13099+01      .12092+04     -.19043-01      .57957-01      .77000-01*

    TENSION OPTIMIZATION ITERATIONS
      I         TENSION        BOOM (EI)        FREQ DIF        SYM FREQ        ASYM FREQ
      1        .22751+00       .25694+04        .21986-01       .46005-01       .24019-01
      2        .16370+01       .25694+04        .15881-01       .80327-01       .64446-01
      3        .16370+01       .25694+04        .15881-01       .80327-01       .64446-01
      4        .26803+01       .25694+04       -.26172-03       .81979-01       .82241-01
                                                          END TENSION OPTIMIZATION

BOOM STIFFNESS OPTIMIZATION ITERATIONS
 I        TENSION       BOOM (EI)       FREQ DIF        FREQ          FREQ REQ
 2      .26803+01      .25694+04      .51103-02      .82110-01      .77000-01*

    TENSION OPTIMIZATION ITERATIONS
      I         TENSION        BOOM (EI)        FREQ DIF        SYM FREQ        ASYM FREQ
      1        .20282+00       .22816+04        .20571-01       .43481-01       .22910-01
      2        .14627+01       .22816+04        .15045-01       .76101-01       .61056-01
      3        .14627+01       .22816+04        .15045-01       .76101-01       .61056-01
      4        .23931+01       .22816+04       -.43306-06 ①     .77622-01       .77622-01
                                                          END TENSION OPTIMIZATION

BOOM STIFFNESS OPTIMIZATION ITERATIONS
 I        TENSION       BOOM (EI)       FREQ DIF        FREQ          FREQ REQ
 3      .23931+01      .22816+04      .62191-03 ②    .77622-01 ③   .77000-01*
```

There are many error messages that can appear in the above section. However, they may not affect the final answer. Without explaining the details of all the error messages, an easy check can be made to determine whether the program has run successfully. If the frequency differences marked ① and ② are less than approximately 5% of the calculated frequency ③ then both the sections have run successfully. If both of these sections have been completed successfully, the optimization has been successful, even if there are error messages. If the run is not successful, check your input data. If the data is correct, your configuration may have caused the eigenvalue routine to fail; and the only thing that can be done is to try a different boom stiffness or required frequency and see if that helps optimization.

```
**********DEPLOYED ARRAY PARAMETERS**********

REQUIRED POWER     =       2500. WATTS

BLANKET AREA       =       250.0 FT**2        (   23.22 M**2)

BLANKET WIDTH      =        3.87 FT           (   1.180 M)

BLANKET LENGTH     =       32.32 FT           (   9.858 M)

ARRAY WIDTH        =        8.25 FT           (   2.516 M)

ARRAY LENGTH       =       33.32 FT           (  10.163 M)

ACTUATOR WIDTH     =         .51 FT           (    .157 M)

LEADER LENGTH      =        1.00 FT           (    .305 M)

BLANKET UNIT WT. =          .180 LB/FT**2     (     .88 KG/M**2)

BLANKET TENSION    =        2.39 LB/BLANKET   (    10.6 N/BLANKET)

APPLIED BOOM LOAD=           .23  OF BOOM BUCKLING LOAD

BUCKLING LOAD      =       20.26 LB           (    90.2 N)

BOOM STIFFNESS EI=         2282. LB FT**2     (    943. NM**2)

BOOM DIAMETER      =        1.38 IN           (   .0350 M)

BOOM THICKNESS     =        .007 IN           (  .00018 M)

BOOM UNIT WEIGHT =           .19 LB/FT        (     .29 KG/M)

BOOM MODULUS (E) =   .4175+10 LB/FT**2        ( .200+12 N/M**2)

BOOM MATL DENSITY=         495.0 LB/FT**3     (   7930. KG/M**3)

BOOM EFFICIENCY    =         .80 EFFECTIVE (I)


FREQUENCY          =        .078 HZ


****BASE STRUCTURE INPUT PARAMETERS****

 RHOS     ES     SIGMS  RHOC     EC     SIGMC
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOSS    ESS    SIGMSS RHOBE    EBE    SIGMBE
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOES    EES    SIGMES RHOCS    ECS    SIGMCS
 1.000  1.000  1.000  1.000  1.000  1.000

 VLAMAC
  .500


 *******CALCULATED BASE STRUCTURE PARAMETERS*******
 ④ (LAUNCH ACCELERATION=   .50*BASELINE LEVEL)
TOTAL BLANKET WEIGHT          =    46.37 LBS (  21.05 KG)

DEPLOYED BOOM WEIGHT          =     6.45 LBS (   2.93 KG)

LEADING EDGE BEAM WEIGHT      =      .74 LBS (    .34 KG)

OUTBOARD END SUPPORT ASSY WT.=      1.89 LBS (    .86 KG)

   DRUM SHELL WT.      =   3.51 LBS (  1.59 KG)
   END CAP+GUIDE WT.   =   1.68 LBS (   .76 KG)
TOTAL DRUM ASSY WEIGHT        =     5.19 LBS (   2.36 KG)

   CENTER SUPPORT WT. =     .72 LBS (   .33 KG)
   BOOM ACTUATOR WT.  =    3.47 LBS (  1.58 KG)
   BEARING ASSY WT.   =    1.47 LBS (   .67 KG)
   NEGATOR ASSY WT.   =    1.68 LBS (   .76 KG)
   SLIPRING+HARNESS WT=    2.35 LBS (  1.07 KG)
   SUPPORT SHAFT WT.  =     .71 LBS (   .32 KG)
TOTAL CENTER SUPPORT ASSY WT =    10.40 LBS (   4.72 KG)

TOTAL SOLAR ARRAY WEIGHT      =    71.06 LBS (  32.26 KG)

POWER TO WEIGHT EFFICIENCY    =    35.18 WATTS/LB (  77.49 WATTS/KG)
```

If there is an error message at the point marked ④ in the program, it means one or more of the base structure equation solutions did not converge. If this happens, check all input data for errors. If there are no errors, the configuration may be too large or too small for the analysis to converge.

```
****BASE STRUCTURE INPUT PARAMETERS****

 RHOS    ES    SIGMS  RHOC    EC    SIGMC
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOSS   ESS   SIGMSS RHOBE   EBE   SIGMBE
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOES   EES   SIGMES RHOCS   ECS   SIGMCS
 1.000  1.000  1.000  1.000  1.000  1.000

 VLAMAC
 1.000


 *******CALCULATED BASE STRUCTURE PARAMETERS*******   ⑤
    (LAUNCH ACCELERATION= 1.00*BASELINE LEVEL)

TOTAL BLANKET WEIGHT           =     46.37 LBS (   21.05 KG)

DEPLOYED BOOM WEIGHT           =      6.45 LBS (    2.93 KG)

LEADING EDGE BEAM WEIGHT       =       .93 LBS (     .42 KG)

OUTBOARD END SUPPORT ASSY WT.=        4.08 LBS (    1.85 KG)

   DRUM SHELL WT.      =  5.74 LBS (  2.61 KG)
   END CAP+GUIDE WT.   =  4.49 LBS (  2.04 KG)
TOTAL DRUM ASSY WEIGHT         =    10.23 LBS (    4.65 KG)

   CENTER SUPPORT WT. =   1.35 LBS (   .61 KG)
   BOOM ACTUATOR WT.   =  5.35 LBS (  2.43 KG)
   BEARING ASSY WT.    =  2.49 LBS (  1.13 KG)
   NEGATOR ASSY WT.    =  1.68 LBS (   .76 KG)
   SLIPRING+HARNESS WT=   2.35 LBS (  1.07 KG)
   SUPPORT SHAFT WT.   =  1.21 LBS (   .55 KG)
TOTAL CENTER SUPPORT ASSY WT =      14.42 LBS (    6.55 KG)

TOTAL SOLAR ARRAY WEIGHT       =    82.49 LBS (  37.45 KG)

POWER TO WEIGHT EFFICIENCY     =    30.31 WATTS/LB (  66.76 WATTS/KG)


TIME FOR RUSAP         =  17.07  SEC.
```

This section (marked ⑤ above) is the second base structure analysis. The deployed array parameters remain the same and new base parameters are read.

C. Typical Level 2 Printout

```
******************************************************************************
******************************************************************************
*****     POWER= 2500. WATTS     AREA=  23.2 M**2      WIDTH= 2.516 M      *****
*****     BOOM STIFFNESS=  1500. N*M**2                                     *****
******************************************************************************
******************************************************************************



****INPUT PARAMETERS FOR DEPLOYED ROLL-UP SOLAR ARRAY****

REQUIRED POWER   =     2500. WATTS

BLANKET AREA     =     250.0 FT**2        (   23.22 M**2)

ARRAY WIDTH      =      8.25 FT           (   2.516 M)

LEADER LENGTH    =      1.00 FT           (    .305 M)

BLANKET UNIT WT. =      .180 LB/FT**2     (     .88 KG/M**2)

BOOM STIFFNESS EI=     3628. LB FT**2     (   1500. NM**2)

BOOM MODULUS (E) =  .4175+10 LB/FT**2     ( .200+12 N/M**2)

BOOM MATL DENSITY=     495.0 LB/FT**3     (   7930. KG/M**3)

BOOM EFFICIENCY  =       .80 EFFECTIVE (I)

DIA TO THICKNESS =      200. IN/IN

          ****PRINTOUT OF PROGRAM OPTIMIZATION ITERATIONS****
                    (ENGLISH UNITS-LB,FT,SEC,HZ)

    TENSION OPTIMIZATION ITERATIONS
      I       TENSION      BOOM (EI)      FREQ DIF       SYM FREQ      ASYM FREQ
      1     .31737+00     .36276+04     .25745-01     .54161-01     .28416-01
      2     .22758+01     .36276+04     .18272-01     .94009-01     .75737-01
      3     .22758+01     .36276+04     .18272-01     .94009-01     .75737-01
      4     .36868+01     .36276+04    -.17921-03     .95726-01     .95905-01
                                                 END TENSION OPTIMIZATION


**********DEPLOYED ARRAY PARAMETERS**********

REQUIRED POWER    =     2500. WATTS

BLANKET AREA      =     250.0 FT**2        (   23.22 M**2)

BLANKET WIDTH     =      3.84 FT           (   1.170 M)

BLANKET LENGTH    =     32.59 FT           (   9.939 M)

ARRAY WIDTH       =      8.25 FT           (   2.516 M)

ARRAY LENGTH      =     33.59 FT           (  10.244 M)

ACTUATOR WIDTH    =       .58 FT           (    .176 M)

LEADER LENGTH     =      1.00 FT           (    .305 M)

BLANKET UNIT WT.  =      .180 LB/FT**2     (     .88 KG/M**2)
```

```
 BLANKET TENSION  =      3.69 LB/BLANKET   (    16.4 N/BLANKET)

 APPLIED BOOM LOAD=      .23  OF BOOM BUCKLING LOAD

 BUCKLING LOAD    =     31.71 LB           (   141.1 N)

 BOOM STIFFNESS EI=     3628. LB FT**2     (   1500. NM**2)

 BOOM DIAMETER    =      1.55 IN           (   .0393 M)

 BOOM THICKNESS   =      .008 IN           (  .00020 M)

 BOOM UNIT WEIGHT =       .24 LB/FT        (     .36 KG/M)

 BOOM MODULUS (E) =  .4175+10 LB/FT**2     ( .200+12 N/M**2)

 BOOM MATL DENSITY=     495.0 LB/FT**3     (   7930. KG/M**3)

 BOOM EFFICIENCY  =       .80 EFFECTIVE (I)


 FREQUENCY        =      .096 HZ


****BASE STRUCTURE INPUT PARAMETERS****

 RHOS    ES    SIGMS  RHOC    EC    SIGMC
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOSS   ESS   SIGMSS RHOBE   EBE  SIGMBE
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOES   EES   SIGMES RHOCS   ECS  SIGMCS
 1.000  1.000  1.000  1.000  1.000  1.000

 VLAMAC
 1.000


 *******CALCULATED BASE STRUCTURE PARAMETERS*******
    (LAUNCH ACCELERATION= 1.00*BASELINE LEVEL)

TOTAL BLANKET WEIGHT          =    46.36 LBS (  21.05 KG)

DEPLOYED BOOM WEIGHT          =     8.20 LBS (   3.72 KG)

LEADING EDGE BEAM WEIGHT      =     1.17 LBS (    .53 KG)

OUTBOARD END SUPPORT ASSY WT.=      4.07 LBS (   1.85 KG)

   DRUM SHELL WT.      =  5.66 LBS (  2.57 KG)
   END CAP+GUIDE WT.   =  4.47 LBS (  2.03 KG)
TOTAL DRUM ASSY WEIGHT        =    10.13 LBS (   4.60 KG)

   CENTER SUPPORT WT. =   1.46 LBS (   .66 KG)
   BOOM ACTUATOR WT.  =   6.80 LBS (  3.09 KG)
   BEARING ASSY WT.   =   2.47 LBS (  1.12 KG)
   NEGATOR ASSY WT.   =   2.09 LBS (   .95 KG)
   SLIPRING+HARNESS WT=   2.37 LBS (  1.07 KG)
   SUPPORT SHAFT WT.  =   1.19 LBS (   .54 KG)
TOTAL CENTER SUPPORT ASSY WT =     16.37 LBS (   7.43 KG)

TOTAL SOLAR ARRAY WEIGHT      =    86.30 LBS (  39.18 KG)

POWER TO WEIGHT EFFICIENCY    =    28.97 WATTS/LB (  63.81 WATTS/KG)


TIME FOR RUSAP       =   3.33  SEC.
```

D. Typical Level 3 Printout

```
********************************************************************************
********************************************************************************
*****     POWER=10000. WATTS   BLANKET L= 81.46 FT   BLANKET W= 6.160 FT    *****
*****     BOOM STIFFNESS=  8965. LB*FT**2      TENSION=  3.25 LB            *****
********************************************************************************
********************************************************************************



****INPUT PARAMETERS FOR DEPLOYED ROLL-UP SOLAR ARRAY****

REQUIRED POWER   =     10000. WATTS

BLANKET AREA     =     1003.6 FT**2       (   93.23 M**2)

BLANKET WIDTH    =       6.16 FT          (   1.879 M)

BLANKET LENGTH   =      81.46 FT          (  24.845 M)

LEADER LENGTH    =       1.00 FT          (    .305 M)

BLANKET UNIT WT. =       .180 LB/FT**2    (     .88 KG/M**2)

BOOM STIFFNESS EI=      8965. LB FT**2    (   3707. NM**2)

BOOM MODULUS (E) =  .4180+10 LB/FT**2     ( .200+12 N/M**2)

BOOM MATL DENSITY=      495.0 LB/FT**3    (   7930. KG/M**3)

BOOM EFFICIENCY  =        .80 EFFECTIVE (I)

DIA TO THICKNESS =       200. IN/IN

PERCENT BUCKLING =     50.000

BLANKET TENSION  =       3.25 LB/BLANKET  (    14.5 N/BLANKET)


**********DEPLOYED ARRAY PARAMETERS**********

REQUIRED POWER   =     10000. WATTS

BLANKET AREA     =     1003.6 FT**2       (   93.23 M**2)

BLANKET WIDTH    =       6.16 FT          (   1.879 M)

BLANKET LENGTH   =      81.46 FT          (  24.845 M)

ARRAY WIDTH      =      13.04 FT          (   3.978 M)

ARRAY LENGTH     =      82.46 FT          (  25.150 M)

ACTUATOR WIDTH   =        .72 FT          (    .221 M)

LEADER LENGTH    =       1.00 FT          (    .305 M)

BLANKET UNIT WT. =       .180 LB/FT**2    (     .88 KG/M**2)

BLANKET TENSION  =       3.25 LB/BLANKET  (    14.5 N/BLANKET)

APPLIED BOOM LOAD=        .50  OF BOOM BUCKLING LOAD

BUCKLING LOAD    =      13.00 LB          (    57.8 N)

BOOM STIFFNESS EI=      8965. LB FT**2    (   3707. NM**2)
```

```
BOOM DIAMETER     =      1.94 IN          (  .0493 M)

BOOM THICKNESS    =      .010 IN          (  .00025 M)

BOOM UNIT WEIGHT =        .38 LB/FT       (    .57 KG/M)

BOOM MODULUS (E) =  .4180+10 LB/FT**2     ( .200+12 N/M**2)

BOOM MATL DENSITY=     495.0 LB/FT**3     (  7930. KG/M**3)

BOOM EFFICIENCY  =        .80 EFFECTIVE (I)


CALCULATED SYM. FREQ.=   .018 HZ.   CALCULATED ASYM. FREQ.=   .029 HZ.



****BASE STRUCTURE INPUT PARAMETERS****

 RHOS    ES     SIGMS  RHOC    EC     SIGMC
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOSS   ESS    SIGMSS RHOBE   EBE    SIGMBE
 1.000  1.000  1.000  1.000  1.000  1.000

 RHOES   EES    SIGMES RHOCS   ECS    SIGMCS
 1.000  1.000  1.000  1.000  1.000  1.000

 VLAMAC
 1.000



 *******CALCULATED BASE STRUCTURE PARAMETERS*******
    (LAUNCH ACCELERATION=  .50*BASELINE LEVEL)

TOTAL BLANKET WEIGHT          =   182.86 LBS (  83.02 KG)

DEPLOYED BOOM WEIGHT          =    31.64 LBS (  14.37 KG)

LEADING EDGE BEAM WEIGHT      =     2.42 LBS (   1.10 KG)

OUTBOARD END SUPPORT ASSY WT.=    12.37 LBS (   5.62 KG)

   DRUM SHELL WT.      =  19.50 LBS (  8.85 KG)
   END CAP+GUIDE WT.   =  12.84 LBS (  5.83 KG)
TOTAL DRUM ASSY WEIGHT        =    32.34 LBS (  14.68 KG)

   CENTER SUPPORT WT. =   4.03 LBS (  1.83 KG)
   BOOM ACTUATOR WT.  =  17.09 LBS (  7.76 KG)
   BEARING ASSY WT.   =   5.22 LBS (  2.37 KG)
   NEGATOR ASSY WT.   =   1.96 LBS (   .89 KG)
   SLIPRING+HARNESS WT=   9.19 LBS (  4.17 KG)
   SUPPORT SHAFT WT.  =   4.03 LBS (  1.83 KG)
TOTAL CENTER SUPPORT ASSY WT =    41.53 LBS (  18.85 KG)

TOTAL SOLAR ARRAY WEIGHT      =   303.16 LBS ( 137.63 KG)

POWER TO WEIGHT EFFICIENCY    =    32.99 WATTS/LB (  72.66 WATTS/KG)



TIME FOR RUSAP        =     .86  SEC.
```

REFERENCES

1. Hasbach, W. A., and Ross, R. G., Summary Report on the Development, Design and Test of a 66-W/kg (30-W/lb) Roll-Up Solar Array, Technical Report 32-1562. Jet Propulsion Laboratory, Pasadena, Calif., Sept. 15, 1972.

2. Coyner, J. V., and Ross, R. G., Parametric Study of the Performance Characteristics and Weight Variations of Large-Area Roll-Up Solar Arrays, Technical Report 32-1502. Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1970.

3. Feasibility Study 30 Watts Per Pound Roll-Up Solar Array, Final Report, Report No. 68SD4301, JPL Contract 915970. General Electric Company, Philadelphia, Pa., June 21, 1968.

Table 1. Geometric scale factor and stress-level relationships

Ratio	Storage shell	End caps	Support shaft	End supports	Center support	Beam	Actuator
$\frac{M_2}{M_1}$	$\lambda_{acc}\lambda_h W_{bs}$	$\lambda_{acc}(\lambda_s^{so}) W_{bs}$	$\lambda_{acc}\lambda_h W_{bscss}$	$\lambda_{acc}\lambda_{ves} W_{bscss}$	$\lambda_{acc}\lambda_{ves} W_{bscssa}$	$[W_{be}\lambda_{acc}(0.5) + \lambda_t(0.5)]\lambda_h'$	$\lambda_{acc}\lambda_v^{1/2} W_{ba}$
$\frac{\sigma_2}{\sigma_1}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$	$\frac{M_2 C_2 l_1}{M_1 C_1 l_2}$
$\frac{l_2}{l_1}$	$(\lambda_s^{so})^3(\lambda_s^{st})$	$(\lambda_s^{so})(\lambda_c^{st})^3$	$(\lambda_{ss}^{so})^3(\lambda_{ss}^{st})$	$(\lambda_{es}^{so})(\lambda_{es}^{st})^3$	$(\lambda_{cs}^{st})^3\lambda_{dia}$	$(\lambda_{be}^{so})^3(\lambda_{be}^{st})$	$\lambda_v(\lambda_a^{st})^2$
$\frac{\sigma_2}{\sigma_1}$	$\frac{\lambda_{acc}\lambda_h W_{bs}}{(\lambda_s^{so})^2(\lambda_s^{st})}$	$\frac{\lambda_{acc} W_{bs}}{(\lambda_c^{st})^2}$	$\frac{\lambda_{acc}\lambda_h W_{bscss}}{(\lambda_{ss}^{so})^2(\lambda_{ss}^{st})}$	$\frac{\lambda_{acc}(\lambda_{ves}) W_{bscss}}{(\lambda_{es}^{so})(\lambda_{es}^{st})^2}$	$\frac{\lambda_{acc}\lambda_{ves} W_{bscssa}}{(\lambda_{cs}^{st})^2\lambda_{dia}}$	$\frac{[W_{be}\lambda_{acc}(0.5) + \lambda_t(0.5)]\lambda_h'}{(\lambda_{be}^{so})^2(\lambda_{be}^{st})}$	$\frac{\lambda_{acc} W_{ba}}{(\lambda_a^{st})^2}$
$\frac{\sigma_{b2}}{\sigma_{b1}}$	$\frac{E_{s2}}{E_{s1}}\left(\frac{\lambda_s^{st}}{\lambda_s^{so}}\right)^2$	—	$\frac{E_{ss2}}{E_{ss1}}\left(\frac{\lambda_{ss}^{st}}{\lambda_{ss}^{so}}\right)^2$	$\frac{E_{es2}}{E_{es1}}\left(\frac{\lambda_{es}^{st}}{\lambda_{es}^{so}}\right)^2$	—	$\frac{E_{be2}}{E_{be1}}\left(\frac{\lambda_{be}^{st}}{\lambda_{be}^{so}}\right)^2$	—

Table 2. Scale-factor solutions for individual components

End caps	Storage shell	Support shaft	Actuator
$\lambda_c^{st} = \left[\frac{\lambda_{acc} W_{bs}}{K_c\left(\frac{\sigma_{yc2}}{\sigma_{yc1}}\right)}\right]^{1/2}$	$\lambda_s^{st} = \left[\lambda_{acc}\lambda_h W_{bs}\left(\frac{E_{s1}}{E_{s2}}\right)\right]^{1/3}$ $\lambda_s^{so} = \left[\frac{\lambda_{acc}\lambda_h W_{bs}}{(\lambda_s^{st}) K_s\left(\frac{\sigma_{ys2}}{\sigma_{ys1}}\right)}\right]^{1/2}$	$\lambda_{ss}^{st} = \left[\frac{\lambda_{acc}\lambda_h W_{bscss}}{\left(\frac{E_{ss2}}{E_{ss1}}\right)}\right]^{1/3}$ $\lambda_{ss}^{so} = \left[\frac{\lambda_{acc}\lambda_h W_{bscss}}{(\lambda_{ss}^{st}) K_{ss}\left(\frac{\sigma_{yss2}}{\sigma_{yss1}}\right)}\right]^{1/2}$	$\lambda_a^{st} = \left[\frac{\lambda_{acc} W_{ba}}{K_a\left(\frac{\sigma_{ya2}}{\sigma_{ya1}}\right)}\right]^{1/2}$

Beam	Center support	End supports
$\lambda_{be}^{st} = \left\{\frac{[W_{be}\lambda_{acc}(0.5) + \lambda_t(0.5)]\lambda_h'}{\left(\frac{E_{be2}}{E_{be1}}\right)}\right\}^{1/3}$ $\lambda_{be}^{so} = \left\{\frac{[W_{be}\lambda_{acc}(0.5) + \lambda_t(0.5)]\lambda_h'}{(\lambda_{be}^{st}) K_{be}\left(\frac{\sigma_{ybe2}}{\sigma_{ybe1}}\right)}\right\}^{1/2}$	$\lambda_{cs}^{st} = \left[\frac{\lambda_{acc}\lambda_{ves} W_{bscssa}}{\lambda_{dia} K_{cs}\left(\frac{\sigma_{ycs2}}{\sigma_{ycs1}}\right)}\right]^{1/2}$	$\lambda_{es}^{st} = \frac{(\lambda_{acc}\lambda_{ves} W_{bscss})^{1/3}}{\left[K_{es}\left(\frac{\sigma_{yes2}}{\sigma_{yes1}}\right)\left(\frac{E_{es2}}{E_{es1}}\right)\right]^{1/6}}$ $\lambda_{es}^{so} = \left[\frac{\lambda_{acc}\lambda_{ves} W_{bscss}}{(\lambda_{es}^{st})^2 K_{es}\left(\frac{\sigma_{yes2}}{\sigma_{yes1}}\right)}\right]$

Table 3. Component material properties

Components	Material density, kg/m^3 (lb/in.^{3})	Modulus of elasticity, N/m^2 (lb/in.^{2})	Tensile yield strength, N/m^2 (lb/in.^{2})
Storage drum	1800 (0.065)	4.5×10^{10} (6.5×10^6)	2.34×10^8 (34,000)
Storage drum end caps	1800 (0.065)	4.5×10^{10} (6.5×10^6)	2.34×10^8 (34,000)
Storage drum support shaft	1800 (0.065)	4.5×10^{10} (6.5×10^6)	2.34×10^8 (34,000)
Leading edge beam	2100 (0.076)	30.3×10^{10} (4×10^7)	4.14×10^8 (60,000)
Storage drum end supports	1800 (0.065)	4.5×10^{10} (6.5×10^6)	2.34×10^8 (34,000)
Center support	1800 (0.065)	4.5×10^{10} (6.5×10^6)	2.34×10^8 (34,000)

Table 4. Data defining three example cases

Parameter	Nominal values preset in program	Case 1	Case 2	Case 3
$DATA1				
NSETS	NO DATA 1 VALUES ARE PRESET	3	—[b]	—
NUNITS		1	—	0
LEVEL		1	2	-3
POWER		2500.	—	10000.
FREQR		0.077	*	*
NTYPE		1	—	0
BKTL		*[a]	*	81.46
BKTW		*	*	6.16
AREA		23.225	—	*
WIDTH		2.516	—	*
TLEADR		0.305	—	1.0
PCTBKL		*	*	-50.
TENS		*	*	3.25
EIBOM1		2000.	1500.	*
EBOOM		0.20E+12	—	0.418E+10
RBOOM		7930.	—	495.
EIEFF		0.80	—	—
DPT		200.	—	—
ROARAY		0.878	—	0.18
NDATA2		2	1	—
$DATA2				
RHOS	1.0	— —	—	—
ES	1.0	— —	—	—
SIGMS	1.0	— —	—	—
RHOC	1.0	— —	—	—
EC	1.0	— —	—	—
SIGMC	1.0	— —	—	—
RHOSS	1.0	— —	—	—
ESS	1.0	— —	—	—
SIGMSS	1.0	— —	—	—
RHOBE	1.0	— —	—	—
EBE	1.0	— —	—	—
SIGMBE	1.0	— —	—	—
RHOES	1.0	— —	—	—
EES	1.0	— —	—	—
SIGMES	1.0	— —	—	—
RHOCS	1.0	— —	—	—
ECS	1.0	— —	—	—
SIGMCS	1.0	— —	—	—
VLAMAC	1.0	0.5 1.0	—	—

[a] (*) indicates that no value need be assigned, since this parameter is not used for this case.

[b] (—) indicates the previous or preset values of the variables are to be used.

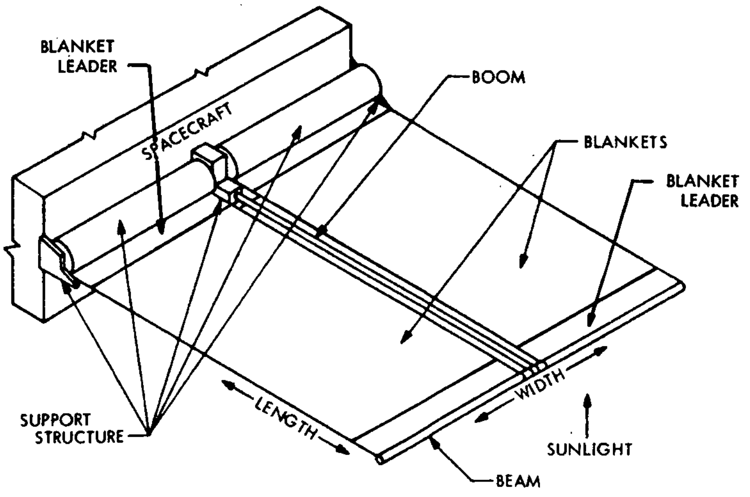

Fig. 1. Deployed array

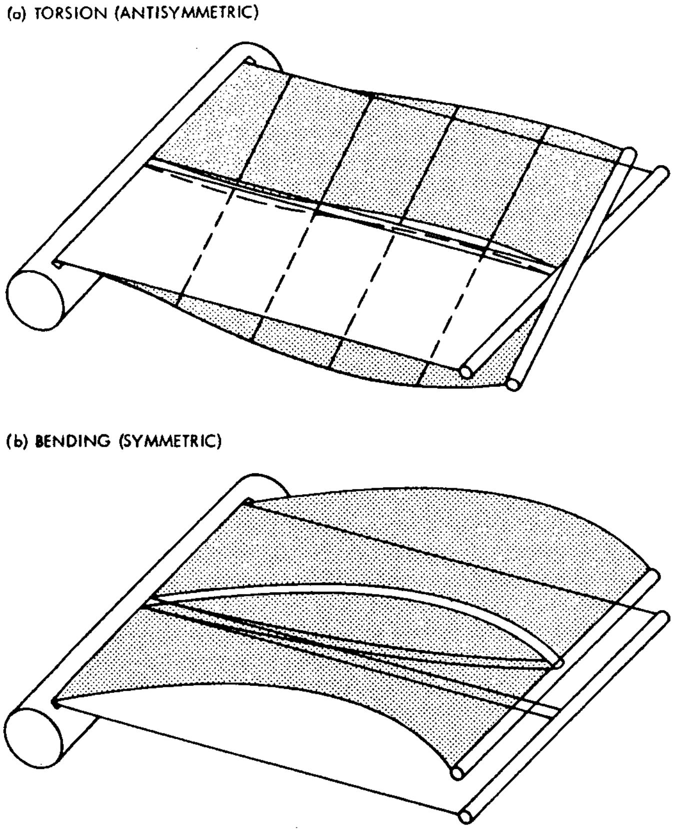

Fig. 2. Deployed array mode shapes

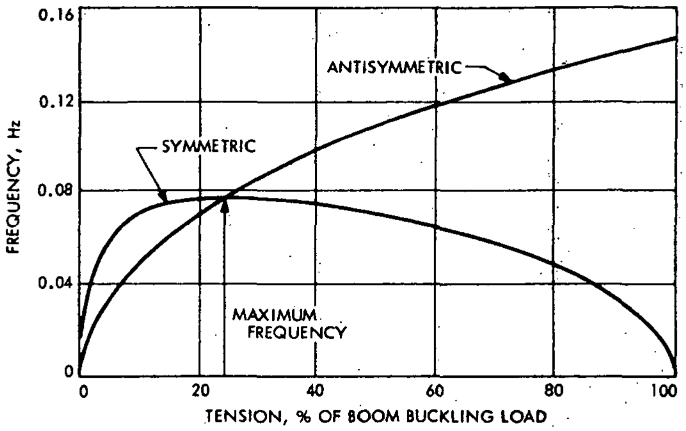

Fig. 3. Typical plot of frequency vs blanket tension

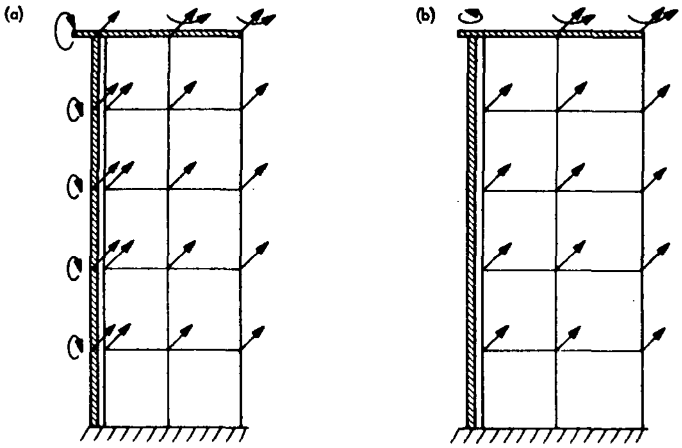

Fig. 4. Typical finite-element models: (a) symmetric; (b) antisymmetric

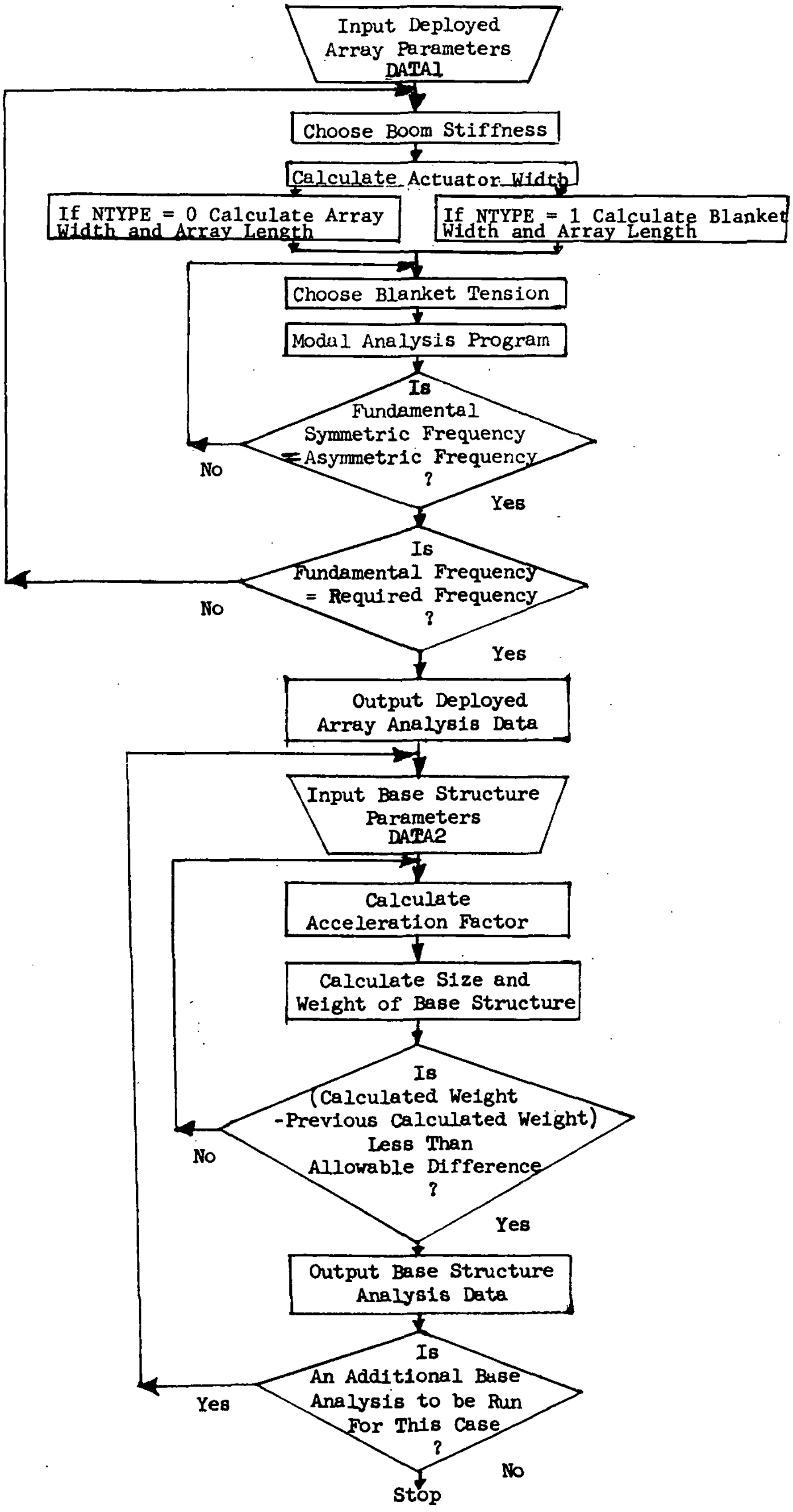

Fig. 5. Level 1 flow chart

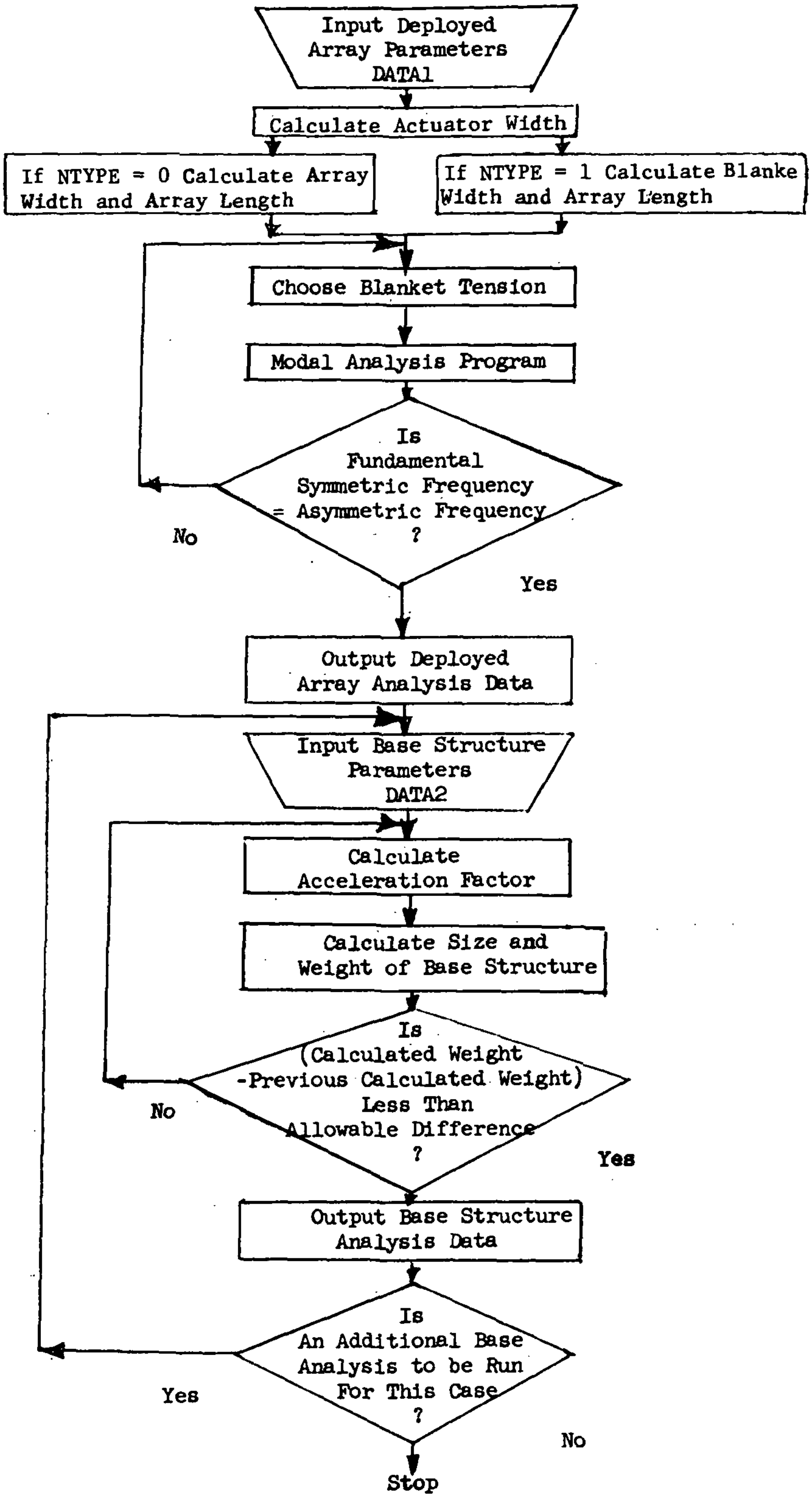

Fig. 6. Level 2 flow chart

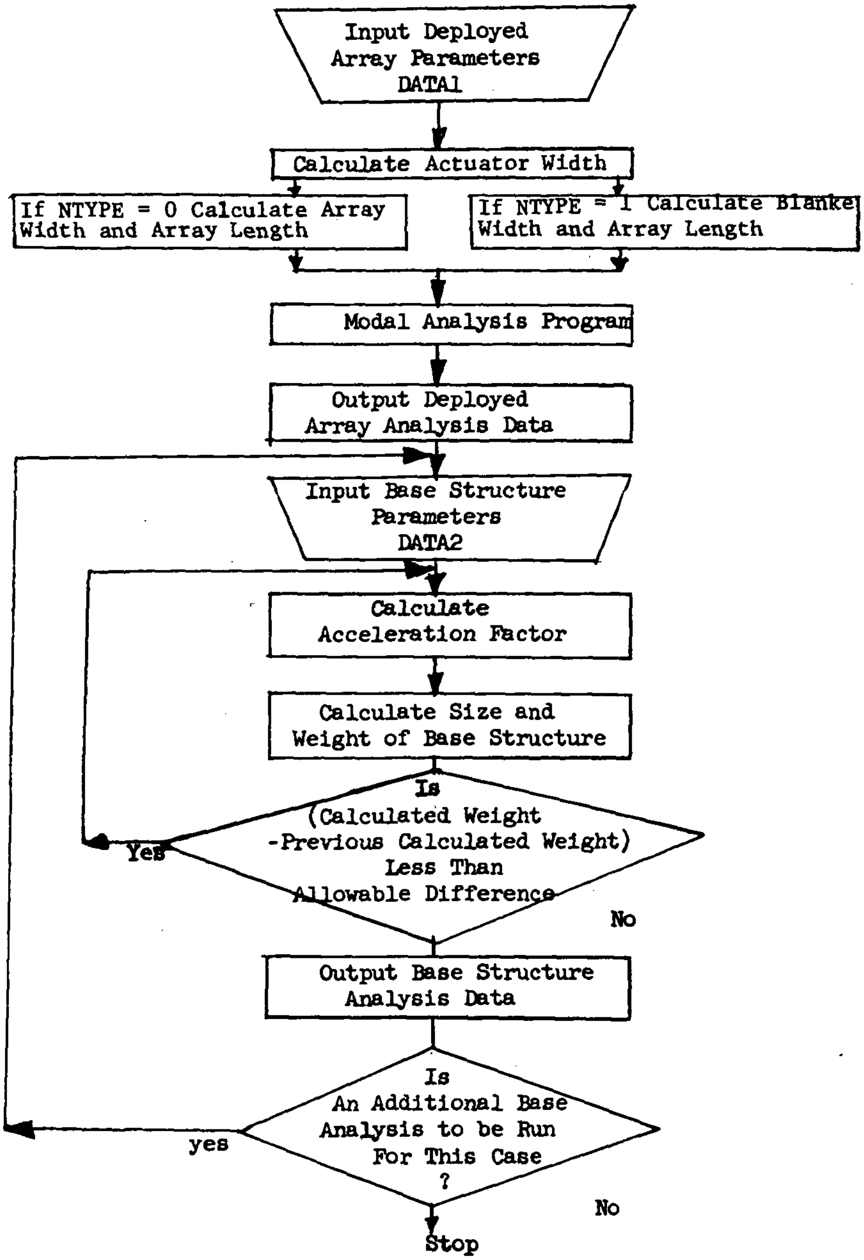

Fig. 7. Level 3 flow chart

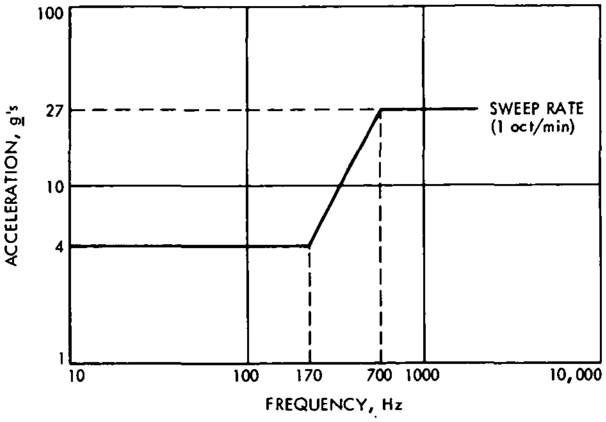

Fig. 8. Qualification sine vibration levels for the GE rollup array

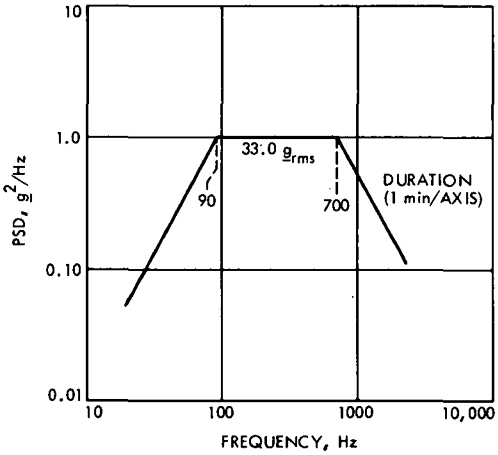

Fig. 9. Qualification random vibration levels for the GE rollup array

```
C     RUSAP (MAIN DRIVING PROGRAM)
C
C
      REAL L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDATA2,ROARAY,NW,NL,L,WIDTH,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,T
     2LEADR,BKTW,BKTL,NTYPE
      COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
      COMMON/JSP6/VLAMAC,VLAMT,VLAMH,RHOBE,AKBE,SIGMBE,EBE,VLMDIA
      COMMON/JSP7/PCTBKL
1     FORMAT(5F10.5,4I5)
2     FORMAT(4I10)
4     FORMAT(1H0,'BOOM STIFFNESS EI=',F10.0,' LB FT**2    (',F8.0,' NM**
     12)')
5     FORMAT(//1H0,'FREQUENCY        =',F10.3,' HZ')
6     FORMAT(1H0,'BLANKET TENSION  =',F10.2,' LB/BLANKET  (',F8.1,' N/BL
     1ANKET)')
7     FORMAT(1H0,'BUCKLING LOAD    =',F10.2,' LB          (',F8.1,' N)')
8     FORMAT(1H0,'APPLIED BOOM LOAD=',F10.2,' OF BOOM BUCKLING LOAD')
9     FORMAT(3F10.5,F11.5)
11    FORMAT(1H0,'REQUIRED POWER   =',F10.0,' WATTS')
12    FORMAT(1H0,'DIA TO THICKNESS =',F10.0,' IN/IN')
13    FORMAT(1H0,'BLANKET UNIT WT. =',F10.3,' LB/FT**2    (',F8.2,' KG/M
     1**2)')
17    FORMAT(1H0,'LEADER LENGTH    =',F10.2,' FT          (',F8.3,' M)')
20    FORMAT(1H0,'BLANKET AREA     =',F10.1,' FT**2       (',F8.2,' M**2
     1)')
21    FORMAT(1H0,'ARRAY LENGTH     =',F10.2,' FT          (',F8.3,' M)')
22    FORMAT(1H0,'ARRAY WIDTH      =',F10.2,' FT          (',F8.3,' M)')
23    FORMAT(1H0,'BOOM DIAMETER    =',F10.2,' IN          (',F8.4,' M)')
24    FORMAT(1H0,'BOOM THICKNESS   =',F10.3,' IN          (',F8.5,' M)')
25    FORMAT(1H0,'BOOM EFFICIENCY  =',F10.2,' EFFECTIVE (I)')
28    FORMAT(1H0,'INITIAL BOOM (EI)=',F10.0,' LB FT**2    (',F8.0,' NM**
     12)')
29    FORMAT(1H0,'PERCENT BUCKLING =',F10.3)
31    FORMAT(1H0,'REQUIRED FREQ    =',F10.3,' HZ')
32    FORMAT(1H0,'BOOM MATL DENSITY=',F10.1,' LB/FT**3    (',F8.0,' KG/M
     1**3)')
33    FORMAT(1H0,'BOOM MODULUS (E) =',E10.4,' LB/FT**2    (',E8.3,' N/M*
     1*2)')
35    FORMAT(1H0,'BOOM UNIT WEIGHT =',F10.2,' LB/FT       (',F8.2,' KG/M
     1)')
37    FORMAT(1H0,'ACTUATOR WIDTH   =',F10.2,' FT          (',F8.3,' M)')
40    FORMAT(//1H0,'CALCULATED SYM. FREQ.=',F7.3,' HZ.','   CALCULATED A
     1SYM. FREQ.=',F7.3,' HZ.')
41    FORMAT(1H0,'BLANKET WIDTH    =',F10.2,' FT          (',F8.3,' M)')
42    FORMAT(1H0,'BLANKET LENGTH   =',F10.2,' FT          (',F8.3,' M)')
49    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' FT**2
     1     WIDTH=',F6.3,' FT     *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' LB*FT**2     TENSION=',F6.2,' LB              *****')
50    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' FT**2
     1     WIDTH=',F6.3,' FT     *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' LB*FT**2                                *****')
51    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' FT**2
     1     WIDTH=',F6.3,' FT     *****'/1H ,'*****      REQUIRED FREQUENC
     2Y=',F6.3,' HZ                                   *****')
249   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' M**2
     1     WIDTH=',F6.3,' M      *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' N*M**2      TENSION=',F6.2,' N                *****')
250   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' M**2
     1     WIDTH=',F6.3,' M      *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' N*M**2                                  *****')
251   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS     AREA=',F6.1,' M**2
     1     WIDTH=',F6.3,' M      *****'/1H ,'*****      REQUIRED FREQUENC
     2Y=',F6.3,' HZ                                   *****')
59    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' FT
     1   BLANKET W=',F6.3,' FT   *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' LB*FT**2     TENSION=',F6.2,' LB              *****')
54    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' FT
     1   BLANKET W=',F6.3,' FT   *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' LB*FT**2                                *****')
61    FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' FT
     1   BLANKET W=',F6.3,' FT   *****'/1H ,'*****      REQUIRED FREQUENC
     2Y=',F6.3,' HZ                                   *****')
259   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' M
     1   BLANKET W=',F6.3,' M    *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' N*M**2      TENSION=',F6.2,' N                *****')
260   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' M
     1   BLANKET W=',F6.3,' M    *****'/1H ,'*****      BOOM STIFFNESS=',
     2F7.0,' N*M**2                                  *****')
261   FORMAT(1H ,'*****      POWER=',F6.0,' WATTS   BLANKET L=',F6.2,' M
     1   BLANKET W=',F6.3,' M    *****'/1H ,'*****      REQUIRED FREQUENC
     2Y=',F6.3,' HZ                                   *****')
52    FORMAT(////1H0,'****INPUT PARAMETERS FOR DEPLOYED ROLL-UP SOLAR AR
     1RAY****')
53    FORMAT(1H1,10X,'****PRINTOUT OF PROGRAM OPTIMIZATION ITERATIONS***
     1*'/22X,'(ENGLISH UNITS-LB,FT,SEC,HZ)')
56    FORMAT(1H ,'******************************************************
     1**************************')
57    FORMAT(1H1,'******************************************************
     1**************************')
60    FORMAT(1H1,'**********DEPLOYED ARRAY PARAMETERS**********')
62    FORMAT(//' MODAL ANALYSIS FAILURE'/' TOTAL TENSION IS LARGER THAN
     1THE BOOM BUCKLING LOAD OF',F7.2,' POUNDS')
63    FORMAT(/////1H ,'*WARNING* TENSION IS BELOW MINIMUM RECOMMENDED T
     1ENSION OF'/1H ,'           0.5 LBS/FT OF WIDTH REQUIRED TO ROLL UP
     2BLANKET')
      NAMELIST/DATA1/NSETS,LEVEL,POWER,AREA,EIBOM1,ROARAY,WIDTH,EIEFF,TL
     1EADR,FREQR,RBOOM,DPT,EBOOM,TENS,NDATA2,NUNITS,BKTW,BKTL,NTYPE,PCTB
     2KL
```

```
C     ALL VARIABLES STARTING WITH THE LETTERS GE ARE PROTOTYPE ARRAY VALUES
      DATA GEEIBE,GEROBE,DEGOP,GEDIAB,GEWIDC,/3040.,.11,20.,1.34,.5/
     1GEW,GET/8.25,2./
      MAXITR=15
      NW=2
      NL=5
      LOP=0

 100  LOP=LOP+1
      READ(5,DATA1)
      RPT=DPT
      BKLPCT=PCTBKL
      PCTBKL=ABS(PCTBKL)
      IF(NTYPE.EQ.0)AREA=BKTL*BKTW*2.
      IF(LEVEL.EQ.1)GO TO 111
      IF(LEVEL.EQ.2)GO TO 111
      IF(LEVEL.EQ.3)GO TO 111
      IF(NTYPE.NE.0)GO TO 102
      IF(BKLPCT.LT.0.0)GO TO 101
      TENS=3.14159**2*EIBOM1*PCTBKL/(200.*(BKTL+TLEADR)**2)
      GO TO 111
 101  EIBOM1=200.*TENS*(BKTL+TLEADR)**2/(3.14159**2*PCTBKL)
      GO TO 111
 102  IF(NUNITS.EQ.0)GO TO 104
      EIBOM1=EIBOM1/.41325331
      EBOOM=EBOOM/47.880259
      WIDTH=WIDTH/.3048
      TENS=TENS/4.4482216
      AREA=AREA/.09290304
      TLEADR=TLEADR/.3048
 104  IF(BKLPCT.LT.0.0)GO TO 103
      DIABOM=(2.*(EIBOM1*RPT/(EBOOM*EIEFF*4.*3.14159))**.25)*12.
      WIDCE=(DIABOM/GEDIAB)*GEWIDC
      L=AREA/(WIDTH-WIDCE)+TLEADR
      TENS=3.14159**2*EIBOM1*PCTBKL/(200.*L**2)
      GO TO 105
 103  CALL RSAZRO(EIBOM1,TENS,2)
 105  IF(NUNITS.EQ.0)GO TO 111
      AREA=AREA*.09290304
      TLEADR=TLEADR*.3048
      TENS=TENS*4.4482216
      WIDTH=WIDTH*.3048
      EBOOM=EBOOM*47.880259
      EIBOM1=EIBOM1*.41325331
 111  IF(NTYPE.EQ.0)GO TO 108
      WRITE(6,57)
      WRITE(6,56)
      IF(NUNITS.EQ.1)GO TO 106
      IF(LEVEL.EQ.1)WRITE(6,51)POWER,AREA,WIDTH,FREQR
      IF(LEVEL.EQ.2)WRITE(6,50)POWER,AREA,WIDTH,EIBOM1
      IF(IABS(LEVEL).EQ.3)WRITE(6,49)POWER,AREA,WIDTH,EIBOM1,TENS
      GO TO 107
 106  IF(LEVEL.EQ.1)WRITE(6,251)POWER,AREA,WIDTH,FREQR
      IF(LEVEL.EQ.2)WRITE(6,250)POWER,AREA,WIDTH,EIBOM1
      IF(IABS(LEVEL).EQ.3)WRITE(6,249)POWER,AREA,WIDTH,EIBOM1,TENS
 107  WRITE(6,56)
      WRITE(6,56)
      GO TO 114
 108  WRITE(6,57)
      WRITE(6,56)
      IF(NUNITS.EQ.1)GO TO 109
      IF(LEVEL.EQ.1)WRITE(6,61)POWER,BKTL,BKTW,FREQR
      IF(LEVEL.EQ.2)WRITE(6,54)POWER,BKTL,BKTW,EIBOM1
      IF(IABS(LEVEL).EQ.3)WRITE(6,59)POWER,BKTL,BKTW,EIBOM1,TENS
      GO TO 110
 109  IF(LEVEL.EQ.1)WRITE(6,261)POWER,BKTL,BKTW,FREQR
      IF(LEVEL.EQ.2)WRITE(6,260)POWER,BKTL,BKTW,EIBOM1
      IF(IABS(LEVEL).EQ.3)WRITE(6,259)POWER,BKTL,BKTW,EIBOM1,TENS
 110  WRITE(6,56)
      WRITE(6,56)
 114  IF(NUNITS.EQ.0)GO TO 115
C
C     SINCE ALL CALCULATIONS ARE PERFORMED IN ENGLISH UNITS,
C     THIS SECTION CONVERTS UNITS IF THE USER INPUTS SI UNITS.
C

      AREA=AREA/.09290304
      TLEADR=TLEADR/.3048
      TENS=TENS/4.4482216
      ROARAY=ROARAY/4.88242764
      WIDTH=WIDTH/.3048
      BKTL=BKTL/.3048
      BKTW=BKTW/.3048
      RBOOM=RBOOM/16.0184637
      EBOOM=EBOOM/47.880259
      EIBOM1=EIBOM1/.41325331
 115  WRITE(6,52)
      WRITE (6,11)POWER
      SIUNIT=AREA*.09290304
      WRITE(6,20)AREA,SIUNIT
      SIUNIT=WIDTH*.3048
      IF(NTYPE.EQ.1)WRITE(6,22)WIDTH,SIUNIT
      SIUNIT=BKTW*.3048
      IF(NTYPE.EQ.0)WRITE(6,41)BKTW,SIUNIT
      SIUNIT=BKTL*.3048
      IF(NTYPE.EQ.0)WRITE(6,42)BKTL,SIUNIT
      SIUNIT=TLEADR*.3048
      WRITE(6,17)TLEADR,SIUNIT
      SIUNIT=ROARAY*4.88242764
      WRITE (6,13)ROARAY,SIUNIT
      IF(IABS(LEVEL).LE.1)WRITE(6,31)FREQR
      SIUNIT=EIBOM1*.41325331
      IF(LEVEL.EQ.1)WRITE(6,28)EIBOM1,SIUNIT
      IF(IABS(LEVEL).GE.2)WRITE(6,4)EIBOM1,SIUNIT
      SIUNIT=EBOOM*47.880259
      WRITE (6,33)EBOOM,SIUNIT
      SIUNIT=RBOOM*16.0184637
      WRITE (6,32)RBOOM,SIUNIT
      WRITE (6,25)EIEFF
```

```
      WRITE (6,12)RPT
      IF(LEVEL.EQ.-3)WRITE(6,29)PCTBKL
      IF(IABS(LEVEL).NE.3)GO TO 205
      EIBOOM=EIBOM1
      DIABOM=(2.*(EIBOOM*RPT/(EBOOM*EIEFF*4.*3.14159))**.25)*12.
      THIBOM=DIABOM/RPT
      ROBOOM=4.*(3.14159*EIBOOM/(EBOOM*EIEFF*RPT))**.50*RBOOM*(360.-DEGO
     1P)/360.
      WIDCE=(DIABOM/GEDIAB)*GEWIDC
      IF(NTYPE.EQ.1)WPANEL=WIDTH-WIDCE
      IF(NTYPE.EQ.0)WPANEL=BKTW*2.
      IF(NTYPE.EQ.0)WIDTH=WPANEL+WIDCE
      IF(NTYPE.EQ.1)L=AREA/WPANEL+TLEADR
      IF(NTYPE.EQ.0)L=BKTL+TLEADR
      SIUNIT=TENS*4.4482216
      WRITE(6,6)TENS.SIUNIT
      T1=3.14159**2*EIBOM1/L**2
      IF(TENS.LT.0.5*T1) GOTO 150
      WRITE(6,62) T1
      GOTO 999
150   FDIF=RSAFX(TENS,EIBOOM,FREQA.FREQS)
      IF(FREQS.EQ.0.) GOTO 999
      GO TO 211
205   WRITE(6,53)
      IF(LEVEL.NE.2)GO TO 215
      EIBOOM=EIBOM1
      CALL RSAZRF(TENS,EIBOOM)
      GO TO 211
215   CALL RSAZER(EIBOOM.EIBOM1.TENS)
      DIABOM=(2.*(EIBOOM*RPT/(EBOOM*EIEFF*4.*3.14159))**.25)*12.
      THIBOM=DIABOM/RPT
      ROBOOM=4.*(3.14159*EIBOOM/(EBOOM*EIEFF*RPT))**.50*RBOOM*(360.-DEGO
     1P)/360.
      WIDCE=(DIABOM/GEDIAB)*GEWIDC
      IF(NTYPE.EQ.1)WPANEL=WIDTH-WIDCE
      IF(NTYPE.EQ.0)WPANEL=BKTW*2.
      IF(NTYPE.EQ.0)WIDTH=WPANEL+WIDCE
      IF(NTYPE.EQ.1)L=AREA/WPANEL+TLEADR
      IF(NTYPE.EQ.0)L=BKTL+TLEADR
211   WRITE(6,60)
      WRITE (6,11)POWER
      SIUNIT=AREA*.09290304
      WRITE(6,20)AREA,SIUNIT
      IF(NTYPE.EQ.1)BKTL=AREA/WPANEL
      IF(NTYPE.EQ.1)BKTW=WPANEL/2.
      SIUNIT=BKTW*.3048
      WRITE(6,41)BKTW,SIUNIT
      SIUNIT=BKTL*.3048
      WRITE(6,42)BKTL,SIUNIT
      SIUNIT=WIDTH*.3048
      WRITE (6,22)WIDTH,SIUNIT
      SIUNIT=L*.3048
      WRITE (6,21)L,SIUNIT
      SIUNIT=WIDCE*.3048
      WRITE(6,37)WIDCE,SIUNIT
      SIUNIT=TLEADR*.3048
      WRITE(6,17)TLEADR,SIUNIT
      SIUNIT=ROARAY*4.88242764
      WRITE (6,13)ROARAY.SIUNIT
      SIUNIT=TENS*4.4482216
      WRITE(6,6)TENS,SIUNIT
      BUCK=3.14159**2*EIBOOM/L**2
      BOOMLD=2.*TENS/BUCK
      WRITE(6,8)BOOMLD
      SIUNIT=BUCK*4.4482216
      WRITE(6,7)BUCK,SIUNIT
      SIUNIT=EIBOOM*.41325331
      WRITE (6,4)EIBOOM,SIUNIT
      SIUNIT=DIABOM*.0254
      WRITE (6,23)DIABOM,SIUNIT
      SIUNIT=THIBOM*.0254
      WRITE (6,24)THIBOM,SIUNIT
      SIUNIT=ROBOOM*1.4881639
      WRITE(6,35)ROBOOM,SIUNIT
      SIUNIT=EBOOM*47.880259
      WRITE (6,33)EBOOM,SIUNIT
      SIUNIT=RBOOM*16.0184637
      WRITE (6,32)RBOOM,SIUNIT
      WRITE (6,25)EIEFF
      IF(IABS(LEVEL).LE.2)WRITE(6,5)FREQ
      IF(IABS(LEVEL).GE.3)WRITE(6,40)FREQS,FREQA
      IF(2.*TENS/WPANEL.LT.0.5)WRITE(6,63)
      EIBOM1=EIBOOM
      IF (NDATA2.NE.0)CALL RSABAS(TENS,EIBOOM)
      PCTBKL=BKLPCT
      IF(NUNITS.EQ.0)GO TO 998
      AREA=AREA*.09290304
      TLEADR=TLEADR*.3048
      TENS=TENS*4.4482216
      ROARAY=ROARAY*4.88242764
      WIDTH=WIDTH*.3048
      BKTL=BKTL*.3048
      BKTW=BKTW*.3048
      RBOOM=RBOOM*16.0184637
      EBOOM=EBOOM*47.880259
      EIBOM1=EIBOM1*.41325331
998   IF(LOP.LT.NSETS)GO TO 100
999   STOP
      END
```

```
      SUBROUTINE RSAZER(X,X1,T)
      REAL X,X1,EPS,AER,RER,F,DF,OLDF,OLDDF,TOL,D,DX,OLDX,XX,A,B,C,FA,FB
     1,FC,MEAN,L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
     2,BKTW,BKTL,NTYPE
11    FORMAT(/1H0,'BOOM STIFFNESS OPTIMIZATION ITERATIONS'/'  I         TE
     1NSION      BOOM (EI)        FREQ DIF         FREQ            FREQ REQ')
12    FORMAT (1H ,I2,5E14.5,1H*)
32    FORMAT(45X,' BOOM STIFFNESS OPTIMIZATION DIDNT CONVERGE')
      ITER=0
      EPS=FREQR*.02
      AER=X1*.005
      RER=.010
      X= X1
      CALL RSAZRF(T,X)
      F=FREQ-FREQR
      FREQ1=FREQ
      WRITE(6,11)
      WRITE(6,12)ITER,T,X,F,FREQ1,FREQR
      IF(ABS(F).LE.EPS) RETURN
      ITER=1
      TOL=AER/RER
      D=(ABS(X*RER)+AER)*5.0
      CALL RSAZRF(T,X+D)
      DF=((FREQ-FREQR)-F)/D
      GO TO 20
15    WRITE(6,11)
      WRITE(6,12)ITER,T,X,F,FREQ1,FREQR
18    ITER=ITER+1
      D=(ABS(X*RER)+AER)*5.0
      CALL RSAZRF(T,X+D)
      DF=((FREQ-FREQR)-F)/D
      IF(DF*OLDDF.GT.0.) GO TO 20
      DX=X-OLDX
      D=2.
      GO TO 25
20    DX=-F/DF
      XX=DX/(SIGN(TOL,X)+X)
      D=AMAX1(0.5,XX,-2.*XX)
25    OLDX=X
      OLDF=F
      OLDFRQ=FREQ1
      OLDDF=DF
      X=OLDX+DX/D
      CALL RSAZRF(T,X)
      F=FREQ-FREQR
      FREQ1=FREQ
      IF(ABS(F).LE.EPS) GO TO 85
      IF(F * OLDF.LE.0.) GO TO 34
      IF(ITER.LT.MAXITR) GO TO 15
      PRINT 32
      ITER=-MAXITR
      GO TO 85
34    IF(ABS(DX/D).LE.ABS(X*RER)+AER) GO TO 85
      A=OLDX
      B=X
      FA=OLDF
      FB=F
      AFREQ=OLDFRQ
      BFREQ=FREQ1
      GO TO 60
40    WRITE(6,11)
      WRITE(6,12)ITER,T,B,FB,BFREQ,FREQR
      IF(ABS(X-B).LT.TOL) X =B+ SIGN(TOL,C-B)
      A=B
      FA=FB
      AFREQ=BFREQ
      IF((X-MEAN)*(B-X).GT.0.) GO TO 50
      B=MEAN
      GO TO 55
50    B=X
55    CALL RSAZRF(T,B)
      FB=FREQ-FREQR
      BFREQ=FREQ
      ITER=ITER+1
      IF(FC*FB.LT.0.) GO TO 65
60    C=A
      FC=FA
      CFREQ=AFREQ
65    IF(ABS(FB).LT.ABS(FC)) GO TO 70
      A=B
      FA=FB
      AFREQ=BFREQ
      B=C
      FB=FC
      BFREQ=CFREQ
      C=A
      FC=FA
      CFREQ=AFREQ
70    IF(ABS(FB).LT.EPS) GO TO 80
      MEAN =(B+C)/2.
      IF(FB.NE.FA) X = (A*FB-B*FA)/(FB-FA)
      IF(FB.EQ.FA) X = MEAN
      TOL = ABS(B*RER)+AER
      IF(ABS(MEAN-B).LT.TOL) GO TO 80
      IF(ITER.LT.MAXITR) GO TO 40
      PRINT 32
      ITER=-MAXITR
80    X=B
      F=FB
      FREQ=BFREQ
85    WRITE(6,11)
      WRITE(6,12) ITER,T,X,F,FREQ,FREQR
      RETURN
      END
```

```
      SUBROUTINE RSAZRF(X,EIBOOM)
      REAL X,X1,EPS,F,A,B,C,FA,FB,FC,MEAN,L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
     2,BKTW,BKTL,NTYPE
      COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
11    FORMAT(//5X,'TENSION OPTIMIZATION ITERATIONS'/5X'  I        TENSION
     1      BOOM (EI)        FREQ DIF        SYM FREQ      ASYM FREQ')
12    FORMAT(1H ,5X,I2,5E14.5)
24    FORMAT(50X,' END TENSION OPTIMIZATION')
32    FORMAT(50X,' TENSION OPTIMIZATION DIDNT CONVERGE')
39    FORMAT(50X,' NO ROOT IN BOUND')
      IF(EIBOOM.LE.0.)GO TO 100
      DIABOM=(2.*(EIBOOM*RPT/(EBOOM*EIEFF*4.*3.14159))**.25)*12.
      THIBOM=DIABOM/RPT
      ROBOOM=4.*(3.14159*EIBOOM/(EBOOM*EIEFF*RPT))**.50*RBOOM*(360.-DEGO
     1P)/360.
      WIDCE=(DIABOM/GEDIAB)*GEWIDC
      IF(NTYPE.EQ.1)WPANEL=W-WIDCE
      IF(NTYPE.EQ.0)WPANEL=BKTW*2.
      IF(NTYPE.EQ.0)W=WPANEL+WIDCE
      IF(NTYPE.EQ.1)L=AREA/WPANEL+TLEADR
      IF(NTYPE.EQ.0)L=BKTL+TLEADR
      X1=(.02*3.14159**2*EIBOOM/L**2)/2.
      X2=(.98*3.14159**2*EIBOOM/L**2)/2.
      EPS=.005*FREQR
      ITER=1
      A=X1
      B=X2
      FA=RSAFX(A,EIBOOM,FREQA,FREQS)
      AFREQA=FREQA
      AFREQS=FREQS
      FB=RSAFX(B,EIBOOM,FREQA,FREQS)
      BFREQA=FREQA
      BFREQS=FREQS
      IF(FB*FA.GT.0.) GO TO 36
      WRITE(6,11)
      GO TO 60
36    PRINT 39
      GO TO 101
40    WRITE(6,12)ITER,B,EIBOOM,FB,BFREQS,BFREQA
45    IF(ABS(X-B).LT.TOL) X =B+ SIGN(TOL,C-B)
      A=B
      FA=FB
      AFREQA=BFREQA
      AFREQS=BFREQS
      IF((X-MEAN)*(B-X).GT.0.) GO TO 50
      B=MEAN
      GO TO 55
50    B=X
55    FB=RSAFX(B,EIBOOM,FREQA,FREQS)
      BFREQA=FREQA
      BFREQS=FREQS
      ITER=ITER+1
      IF(FC*FB.LT.0.) GO TO 65
60    C=A
      FC=FA
      CFREQA=AFREQA
      CFREQS=AFREQS
65    IF(ABS(FB).LT.ABS(FC)) GO TO 70
      A=B
      FA=FB
      AFREQA=BFREQA
      AFREQS=BFREQS
      B=C
      FB=FC
      BFREQA=CFREQA
      BFREQS=CFREQS
      C=A
      FC=FA
      CFREQA=AFREQA
      CFREQS=AFREQS
70    IF( ABS(FB).LT.EPS) GO TO 80
      MEAN =(B+C)/2.
      IF(FB.NE.FA) X = (A*FB-B*FA)/(FB-FA)
      IF(FB.EQ.FA) X = MEAN
      TOL=ABS(B*.001)
      IF(ABS(MEAN-B).LT.TOL) GO TO 80
      IF(ITER.LT.MAXITR) GO TO 40
101   PRINT 32
      ITER=-MAXITR
80    X=B
      F=FB
      FREQS=BFREQS
      FREQA=BFREQA
      FREQ=(FREQS+FREQA)/2.
       WRITE(6,12) ITER,X,EIBOOM,F,FREQS,FREQA
      GO TO 26
100   FREQ=EIBOOM
26    WRITE(6,24)
      RETURN
      END
```

```
      FUNCTION RSAFX(T,EIBOOM,FREQA,FREQS)
      REAL M(4000),K(4000),VAL(100),L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
     2,BKTW,BKTL,NTYPE
      COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
      COMMON/JSP6/VLAMAC,VLAMT,VLAMH,RHOBE,AKBE,SIGMBE,EBE,VLMDIA
    3 FORMAT(' MODAL ANALYSIS FAILURE')
   21 FORMAT(2X,1H*,I5)
      W2=W/2.
      WPANE2=WPANEL/2.
      EIBEAN=GEEIBE*(0.5*T/GET+0.5*VLAMAC)*W/GEW
      ROBEAN=GEROBE*SQRT(EIBEAN/GEEIBE)
      T2=-2.*T
      ROARAY=ROARAY*(WPANE2/W2)
      CALL KRAY(K,N,EIBEAN,EIBOOM,T,T,T2,T2,NW,NL,L,W2,0)
      CALL MRAY(M,N,ROBEAN,ROBOOM,ROARAY,NW,NL,L,W2,0)
      CALL GENIGN(M,K,VAL,VAL,-N,0,OK)
      ROARAY=ROARAY*(W2/WPANE2)
      IF(OK.EQ.0.)GO TO 1
      DO 10 I=1,N
      A=VAL(I)
      IF (A.GT.0.) GO T) 15
   10 CONTINUE
   15 IF(I.GT.1)WRITE(6,21)I
      FREQS=SQRT(A)/6.2831853
      T=T*(W2/WPANE2)
      T2=-2.*T
      CALL KRAY(K,N,EIBEAN,EIBOOM,T,T,T2,T2,NW,NL,L,W2,1)
      CALL MRAY(M,N,ROBEAN,ROBOOM,ROARAY,NW,NL,L,W2,1)
      CALL GENIGN(M,K,VAL,VAL,-N,0,OK)
      T=T*(WPANE2/W2)
      T2=-2.*T
      IF(OK.EQ.0.)GO TO 1
      DO 11 I=1,N
      A=VAL(I)
      IF (A.GT.0.) GO T) 16
   11 CONTINUE
   16 IF(I.GT.4)WRITE(6,21)I
      FREQA=SQRT(A)/6.2831853
      RSAFX=FREQS-FREQA
      GO TO 2
    1 RSAFX=T*100.
      FREQS=0.
      WRITE(6,3)
    2 RETURN
      END
```

```
      SUBROUTINE RSABAS(T,EIBOOM)
      REAL L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
     2,BKTW,BKTL,NTYPE
      COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
      COMMON/JSP5/CONST,COEFF,EXP
      COMMON/JSP6/VLAMAC,VLAMT,VLAMH,RHOBE,AKBE,SIGMBE,EBE,VLMDIA
1     FORMAT(/' TOTAL BLANKET WEIGHT          =',F9.2,' LBS (',F7.2,' KG)
     1')
2     FORMAT('0   DRUM SHELL WT.      =',F7.2,' LBS (',F6.2,' KG)')
3     FORMAT('    END CAP+GUIDE WT.   =',F7.2,' LBS (',F6.2,' KG)')
4     FORMAT('    SUPPORT SHAFT WT.   =',F7.2,' LBS (',F6.2,' KG)')
5     FORMAT('    BEARING ASSY WT.    =',F7.2,' LBS (',F6.2,' KG)')
6     FORMAT(/' LEADING EDGE BEAM WEIGHT      =',F9.2,' LBS (',F7.2,' KG)
     1')
7     FORMAT(/' OUTBOARD END SUPPORT ASSY WT.=',F9.2,' LBS (',F7.2,' KG)
     1')
8     FORMAT('0   CENTER SUPPORT WT. =',F7.2,' LBS (',F6.2,' KG)')
9     FORMAT('    SLIPRING+HARNESS WT=',F7.2,' LBS (',F6.2,' KG)')
10    FORMAT(/' DEPLOYED BOOM WEIGHT          =',F9.2,' LBS (',F7.2,' KG)
     1')
11    FORMAT('    NEGATOR ASSY WT.    =',F7.2,' LBS (',F6.2,' KG)')
12    FORMAT('    BOOM ACTUATOR WT.   =',F7.2,' LBS (',F6.2,' KG)')
13    FORMAT(/' TOTAL SOLAR ARRAY WEIGHT      =',F9.2,' LBS (',F7.2,' KG)
     1')
14    FORMAT( ' TOTAL DRUM ASSY WEIGHT        =',F9.2,' LBS (',F7.2,' KG)
     1')
15    FORMAT( ' TOTAL CENTER SUPPORT ASSY WT =',F9.2,' LBS (',F7.2,' KG)
     1')
41    FORMAT(2I10)
42    FORMAT(9F8.4)
43    FORMAT(/' POWER TO WEIGHT EFFICIENCY    =',F9.2,' WATTS/LB (',F7.2,
     1' WATTS/KG)')
47    FORMAT(///1H0,' *******CALCULATED BASE STRUCTURE PARAMETERS*******
     1'/'      (LAUNCH ACCELERATION=',F5.2,'*BASELINE LEVEL)')
50    FORMAT(1H1,'****BASE STRUCTURE INPUT PARAMETERS****')
100   FORMAT(10F7.3)
101   FORMAT(1H0,' RHOS    ES    SIGMS  RHOC    EC    SIGMC')
102   FORMAT(1H0,' RHOSS   ESS   SIGMSS RHOBE   EBE  SIGMBE')
103   FORMAT(1H0,' RHOES   EES   SIGMES RHOCS   ECS  SIGMCS')
104   FORMAT(1H0,' VLAMAC')
      NAMELIST/DATA2/RHOS,ES,SIGMS,EC,SIGMC,RHOC,RHOSS,ESS,SIGMSS,EBE,SI
     1GMBE,RHOBE,SIGMES,EES,RHOES,SIGMCS,RHOCS,ECS,VLAMAC
      DATA RHOS,ES,SIGMS,AKS,EC,AKC,SIGMC,RHOC/1.,1.,1.,1.,1.,1.,1.,1./
     1RHOSS,ESS,AKSS,SIGMSS,RHOB,EBE,AKBE/1.,1.,1.,1.,1.,1.,1./
     2SIGMBE,RHOBE,AKES,SIGMES,EES,RHOES,AKCS,/1.,1.,1.,1.,1.,1.,1./
     3SIGMCS,RHOCS,ECS,RHOA,AKA,EA,SIGMA,RHONEG/1.,1.,1.,1.,1.,1.,1.,1./
```

```
     4VLAMAC/1./
      DATA GEWS,GEWC,GEWSS,GEWB,GEWBE,GEWES/5.78,4.53,1.21,2.50,.85,4.1/
     1GEWCS,GEWNS,GEWA,GEWNEG,GEBLKT/1.33,2.34,5.18,1.54,46.58/
     2GEBOOM,GETOTL,GERI,GERO,GEL/6.55,82.5,4.,5.,33.2/
      ACC=VLAMAC
      NBAS=0
      GEPANE=GEW-GEWIDC
   97 READ(5,DATA2)
      ACC=VLAMAC
   99 VLMDIA=DIABOM/GEDIAB
      VLAMT=T/GET
      VLAMH=WPANEL/GEPANE
      VLAMV=L/GEL
      VLAMHP=W/GEW
      WRITE(6,50)
      WRITE(6,101)
      WRITE(6,100)RHOS,ES,SIGMS,RHOC,EC,SIGMC
      WRITE(6,102)
      WRITE(6,100)RHOSS,ESS,SIGMSS,RHOBE,EBE,SIGMBE
      WRITE(6,103)
      WRITE(6,100)RHOES,EES,SIGMES,RHOCS,ECS,SIGMCS
      WRITE(6,104)
      WRITE(6,100)ACC
      WBOOM=ROBOOM*L
      WBLNKT=ROARAY*L*WPANEL
      WNS=GEWNS*VLAMHP*VLAMV
      WNEG=GEWNEG*RHONEG*SQRT(VLAMT)
      RATIO=SQRT(1./(VLAMH*VLAMV))
  300 VLAMAC=ACC*RATIO
      WST=GEBLKT+GEWS
      CONST=WBLNKT/WST
      RATIOW=GEWS/WST
      COEFF=RATIOW*(RHOS*VLAMH**(5./3.)*VLAMAC**(2./3.)*SQRT(1./(SIGMS*A
     1KS))/ES**(1./6.))
      EXP=2./3.
      CALL RSAZRO(WBSR,T,0)
      VSST=(VLAMAC*VLAMH*WBSR/ES)**(1./3.)
      VSSO=(VLAMAC*VLAMH*WBSR)**(1./3.)*SQRT(ES**(1./3.)/(SIGMS*AKS))
      WS=WBSR*WST-WBLNKT
      VCST=SQRT((VLAMAC*WBSR/(AKC*SIGMC)))
      WC=GEWC*RHOC*VSSO*VCST**2
      WST=GEBLKT+GEWS+GEWC+GEWSS
      CONST=((GEBLKT+GEWS)/WST)*WBSR+RHOC*VSSO*VCST**2*GEWC/WST
      RATIOW=GEWSS/WST
      COEFF=RATIOW*(RHOSS*VLAMH**(5./3.)*VLAMAC**(2./3.)*SQRT(1./(SIGMSS
     1*AKSS))/ESS**(1./6.))
      EXP=2./3.
      CALL RSAZRO(WBSCSR,T,0)
      VSSST=(VLAMAC*VLAMH*WBSCSR/ESS)**(1./3.)
      VSSSO=(VLAMAC*VLAMH*WBSCSR)**(1./3.)*SQRT(ESS**(1./3.)/(SIGMSS*AKS
     1S))
      WSS=WBSCSR*WST-WBLNKT-WS-WC
      WB=GEWB*RHOB*VSSSO**2
      CALL RSAZRO(WBER,T,1)
      WBE=WBER*GEWBE
      VBEST=((WBER*VLAMAC*.5+VLAMT*.5)*VLAMHP/EBE)**(1./3.)
      VBESO=SQRT((WBER*VLAMAC*.5+VLAMT*.5)*VLAMHP/(VBEST*AKBE*SIGMBE))
      VLMVES=SQRT(VLAMV+(VSSO**2-VLAMV)*GERI**2/GERO**2)
      VESST=(VLAMAC*VLMVES*WBSCSR)**(1./3.)/(AKES*SIGMES*EES)**(1./6.)
      VESSO=(VLAMAC*VLMVES*WBSCSR/(VESST**2*AKES*SIGMES))
      WES=GEWES*RHOES*VESST*VESSO*VLMVES
      WST=GEWA+GEBOOM
      CONST=WBOOM/WST
      RATIOW=GEWA/WST
      COEFF=RATIOW*RHOA*VLMDIA*VLAMV**(1./2.)*(VLAMAC/(AKA*SIGMA))**(1./
     12.)
      EXP=1./2.
      CALL RSAZRO(WBAR,T,0)
      WA=WBAR*WST-WBOOM
      VAST=(VLAMAC*WBAR/(AKA*SIGMA))**(1./2.)
      WST=GEBLKT+GEWS+GEWC+GEWSS+GEWA+GEBOOM+GEWB+GEWNEG+GEWNS
      WBSCSA=(WBLNKT+WS+WC+WSS+WA+WBOOM+WB+WNEG+WNS)/WST
      VCSST=SQRT(VLAMAC*VLMVES*WBSCSA/(VLMDIA*AKCS*SIGMCS))
      WCS=GEWCS*RHOCS*VLMDIA*VLMVES*VCSST
      WT=WBLNKT+WS+WC+WSS+WB+WBE+WES+WCS+WNS+WBOOM+WA+WNEG
      WDRM=WS+WC
      WBUS=WCS+WA+WB+WNEG+WNS+WSS
      WATPS=POWER/WT
      IF(ABS(RATIO-SQRT(GETOTL/WT)).LT.0.1*SQRT(GETOTL/WT))GO TO 301
      RATIO=SQRT(GETOTL/WT)
      GO TO 300
  301 WRITE(6,47)VLAMAC
      SIUNIT=.454*WBLNKT
      WRITE(6,1)WBLNKT,SIUNIT
      SIUNIT=.454*WBOOM
      WRITE(6,10)WBOOM,SIUNIT
      SIUNIT=.454*WBE
      WRITE(6,6)WBE,SIUNIT
      SIUNIT=.454*WES
      WRITE(6,7)WES,SIUNIT
      SIUNIT=.454*WS
      WRITE(6,2)WS,SIUNIT
      SIUNIT=.454*WC
      WRITE(6,3)WC,SIUNIT
      SIUNIT=.454*WDRM
      WRITE(6,14) WDRM,SIUNIT
      SIUNIT=.454*WCS
      WRITE(6,8)WCS,SIUNIT
      SIUNIT=.454*WA
```

```
      WRITE(6,12)WA,SIUNIT
      SIUNIT=.454*WB
      WRITE(6,5)WB,SIUNIT
      SIUNIT=.454*WNEG
      WRITE(6,11)WNEG,SIUNIT
      SIUNIT=.454*WNS
      WRITE(6,9)WNS,SIUNIT
      SIUNIT=.454*WSS
      WRITE(6,4)WSS,SIUNIT
      SIUNIT=.454*WBUS
      WRITE(6,15) WBUS,SIUNIT
      SIUNIT=.454*WT
      WRITE(6,13)WT,SIUNIT
      SIUNIT=WATPS/.454
      WRITE(6,43)WATPS,SIUNIT
  199 NBAS=NBAS+1
      VLAMAC=ACC
      IF(NDAT2.GT.NBAS)GO TO 97
  200 RETURN
      END


      SUBROUTINE RSAZRO(X,T,NCHECK)
      REAL MEAN,L
      COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
     1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
     2,BKTW,BKTL,NTYPE
      COMMON/JSP5/CONST,COEFF,EXP
      COMMON/JSP6/VLAMAC,VLAMT,VLAMH,RHOBE,AKBE,SIGMBE,EBE,VLMDIA
32    FORMAT('  ANALYSIS DIDNT CONVERGE')
39    FORMAT('  ANALYSIS,NO ROOT WITHIN BOUNDS')
      ITER=1
      RER=.001
      AER=.0001
      EPS=.0001
      IF(NCHECK.NE.2)GO TO 99
      A=10.
      B=500000.
      GO TO 100
99    A=.01
      B=100.
100   IF(NCHECK.NE.1)GO TO 104
      FA=RSAF2(A)
      FB=RSAF2(B)
      GO TO 109
104   IF(NCHECK.NE.0)GO TO 105
      FA=RSAF1(A)
      FB=RSAF1(B)
      GO TO 109
105   FA=RSAF3(A,T)
      FB=RSAF3(B,T)
109   IF(FB*FA.GT.0.) GO TO 38
      GO TO 60
38    PRINT 39
      ITER=0
      GO TO 85

45    IF(ABS(X-B).LT.TOL) X =B+ SIGN(TOL,C-B)
      A=B
      FA=FB
      IF((X-MEAN)*(B-X).GT.0.) GO TO 50
      B=MEAN
      GO TO 55
50    B=X
55    IF(NCHECK.NE.1)GO TO 106
      FB=RSAF2(B)
      GO TO 110
106   IF(NCHECK.NE.0)GO TO 107
      FB=RSAF1(B)
      GO TO 110
107   FB=RSAF3(B,T)
110   ITER=ITER+1
      IF(FC*FB.LT.0.) GO TO 65
60    C=A
      FC=FA
65    IF(ABS(FB).LT.ABS(FC)) GO TO 70
      A=B
      FA=FB
      B=C
      FB=FC
      C=A
      FC=FA
70    IF(ABS(FB).LT.EPS) GO TO 80
      MEAN =(B+C)/2.
      IF(FB.NE.FA) X = (A*FB-B*FA)/(FB-FA)
      IF(FB.EQ.FA) X = MEAN
      TOL = ABS(B*RER)+AER
      IF(ABS(MEAN-B).LT.TOL) GO TO 80
      IF(ITER.LT.MAXITR) GO TO 45
      PRINT 32
      ITER=-MAXITR
80    X=B
      F=FB
85    RETURN
      END
```

```
       FUNCTION RSAF3(EI,T)
       REAL L
       COMMON/JSP1/EIEFF,FREQ,DIABOM,THIBOM,ROBOOM,WIDCE,DEGOP,WPANEL,MAX
      1ITR,NDAT2,ROARAY,NW,NL,L,W,FREQR,RBOOM,EBOOM,RPT,POWER,AREA,TLEADR
      2,BKTW,BKTL,NTYPE
       COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
       COMMON/JSP7/PCTBKL
       RL1=SQRT(3.14159**2*EI*PCTBKL/(200.*T))
       RL2=AREA/(W-((2.*(EI*RPT/(EBOOM*EIEFF*4.*3.14159))**.25)*12.)*GEWI
      1DC/GEDIAB)+TLEADR
       RSAF3=RL1-RL2
       RETURN
       END


       FUNCTION RSAF1(W)
       COMMON/JSP5/CONST,COEFF,EXP
       IF(W.LT.0.0)GO TO 1
       RSAF1=W-COEFF*W**EXP-CONST
       GO TO 2
     1 RSAF1=50.*W
     2 RETURN
       END


       FUNCTION RSAF2(W)
       COMMON/JSP2/GEDIAB,GEWIDC,GEW,GET,GEEIBE,GEROBE
       COMMON/JSP6/VLAMAC,VLAMT,VLAMH,RHOBE,AKBE,SIGMBE,EBE,VLMDIA
       IF(W.LT.0.0)GO TO 1
       RSAF2=W-(W*VLAMAC*.5+VLAMT*.5)**(2./3.)*((VLAMH*(GEW-GEWIDC)+VLMDI
      1A*.5)/GEW)**(5./3.)*RHOBE/(SQRT(AKBE*SIGMBE)*EBE**(1./6.))
       GO TO 2
     1 RSAF2=50.*W
     2 RETURN
       END

       SUBROUTINE GENIGN(A,B,IGNVAL,IGNVEC,N,NUMVTR,DETB)
       INTEGER N,NUMVTR,M
       REAL A(1),B(1),IGNVAL(1),IGNVEC(1),DETB
       DOUBLE PRECISION DET
       DETB=0.
       M=IABS(N)
       CALL SYMDET(DET,B,B,M)
       IF(DET.EQ.0.D0) GOTO 1
       IF(DABS(DET).GT.1.D38) DETB=DSIGN(1.D38,DET)
       IF(DABS(DET).LT.1.D-38) DETB=DSIGN(1.D-38,DET)
       IF(DETB.EQ.0.) DETB=DET
       GOTO 5
1      PRINT 3
3      FORMAT(25H THUS GENIGN IS UNDEFINED)
       RETURN
5      CALL UINVRS(B,B,M)
       CALL UTRAU(B,A,A,M,IGNVAL)
       CALL SYMIGN(A,IGNVAL,IGNVEC,N,NUMVTR)
       IF(NUMVTR.EQ.0) RETURN
       CALL UXM(B,IGNVEC,IGNVEC,M,IABS(NUMVTR))
       RETURN
       END


       SUBROUTINE SYMIGN(A,IGNVAL,IGNVEC,N,NUMVTR)
       PARAMETER MXN=100
       INTEGER N,NUMVTR,NP1,I,P,M,NMM,IM1,MM1,K,KP1,LAST,L,F,J
       REAL A(1),IGNVAL(1),IGNVEC(1),NORM,OVRFLO,EPQ,LAMDA,DIA,SCALE,
      1      C,D,DD,FF,EPS,PRECS,S,T,XX,DI(MXN),CD(MXN),CDQ(MXN)
       LOGICAL FULL
       COMMON/SR1/DI/SR2/CD/SR3/CDQ/NRM/NORM
       CALL TRIDIA(A,DI,CD,CDQ,NORM,IABS(N))
       FULL=.TRUE.
       GOTO 10
       ENTRY TRIGN (IGNVAL,IGNVEC,N,NUMVTR)
       FULL=.FALSE.
10     PRECS=1.E-8
       OVRFLO=1.E38
       L=IABS(N)
       SCALE=SQRT(OVRFLO*PRECS)/AMAX1(1.,AMIN1(NORM,OVRFLO*PRECS))
       S=SCALE**2
       EPS=PRECS*NORM*SCALE/FLOAT(L)**2
       EPQ=AMAX1(EPS**2,1000./OVRFLO)
       NP1=L+1
       DO 15 I=1,L
         P=NP1-I
         IGNVAL(P)=DI(I)
         DI(I)=DI(I)*SCALE
15       CDQ(I)=CDQ(I)*S
       M=L
       GOTO 24
20     LAMDA=DI(M)/SCALE
         CDQ(M)=IGNVAL(NMM)
         I=NMM
21       IF(I.EQ.1) GOTO 22
           IM1=I-1
           IF (LAMDA.GE.IGNVAL(IM1)) GOTO 22
           IGNVAL(I)=IGNVAL(IM1)
           I=IM1
           GOTO 21
```

```
22      IGNVAL(I)=LAMDA
        IF(M.EQ.1) GOTO 49
        M=MM1
        CDQ(M)=0.
24      MM1=M-1
        NMM=L-MM1
        IF(M.EQ.1) GOTO 20
25      K=M
30        K=K-1
            IF(K.EQ.0) GOTO 35
            IF(CDQ(K).GT.EPQ) GOTO 30
          IF(K.EQ.MM1) GOTO 20
          CDQ(K)=0.
35        K=K+1
          T=DI(M)-DI(MM1)
          XX=CDQ(MM1)
          DD=T/2.
          IF(DD*DD.LE.ABS(PRECS*XX)) GOTO 37
          LAMDA=(XX/DD)/(1.+S3RT(1.+XX/DD**2))
          GOTO 40

37        LAMDA=SQRT(XX)
40        D=DI(M)+LAMDA
          DD=DI(MM1)-LAMDA
          IF(K.NE.MM1) GOTO 42
            DI(M)=D
            DI(MM1)=DD
            GOTO 20
42        LAMDA=D
          IF(ABS(T).LT.EPS.AND.ABS(DD).LT.ABS(D)) LAMDA=DD
          DIA=DI(K)-LAMDA
          IF(ABS(DIA).LT.EPS) DIA=DIA+SIGN(EPS,DIA)
          FF=DIA**2
          XX=CDQ(K)
          T=XX+FF
          GOTO 44
43          FF=DIA/C*DIA
            XX=CDQ(KP1)
            T=XX+FF
            CDQ(K)=S*T
            K=KP1
44          KP1=K+1
            C=FF/T
            S=XX/T
            XX=DIA
            DIA=C*(DI(KP1)-LAMDA)-S*XX
            DI(K)=(XX-DIA)+DI(KP1)
            IF(ABS(DIA).LT.EPS) DIA=DIA+SIGN(C*EPS,DIA)
            IF(K.LT.MM1) GOTO 43
          CDQ(MM1)=DIA/C*DIA*S
          DI(M)=DIA+LAMDA
        GOTO 25
49    LAST=NUMVTR
      IF(N.GT.0) GOTO 55
      IF(IGNVAL(1).LT.0..AND.IGNVAL(L).GT.0.) GOTO 50
      IF(ISIGN(1,LAST).GT.0) GOTO 56
      LAST=-LAST
      GOTO 57
50    F=1
      K=2
      IF(IGNVAL(2).GT.0.) GOTO 54
      K=3
51    IF(IGNVAL(K).GT.0.) GOTO 52
      K=K+1
      GOTO 51
52    J=K-1
      P=(K-F)/2+F-1
      DO 53 I=F,P
        DD=IGNVAL(I)
        IGNVAL(I)=IGNVAL(J)
        IGNVAL(J)=DD
53      J=J-1
      IF(K.GE.L) GOTO 55
54    F=K
      K=NP1
      GOTO 52
55    IF(ISIGN(1,LAST).GT.0) GOTO 57
      LAST=-LAST

56    F=1
      K=NP1
      GOTO 52
57    IF(NUMVTR.EQ.0) GOTO 70
      P=0
      DO 65 I=1,LAST
        CALL TRIVEC(CDQ,CD,DI,IGNVAL,I,NORM,L)
        IF(FULL) CALL TRIBAK(A,CD,DI,L)
        M=P
        DO 60 K=1,L
          M=M+1
60        IGNVEC(M)=DI(K)
65    P=P+L
70    IF(N.GT.0) RETURN
      DO 75 I=1,L
75      IGNVAL(I)=1./IGNVAL(I)
      RETURN
      END
```

```fortran
      SUBROUTINE MB(M,W,L,N1,N2,N3,N4,N)
      INTEGER N1,N2,N3,N4,N
      REAL M(N,1),W,L
      A=W/32.2/420.
      B=A*L
      C=B*L
      M(N1,N1)=M(N1,N1)+156.*A
      M(N1,N2)=M(N1,N2)+22.*B
      M(N1,N3)=M(N1,N3)+54.*A
      M(N1,N4)=M(N1,N4)-13.*B
      M(N2,N1)=M(N2,N1)+22.*B
      M(N2,N2)=M(N2,N2)+4.*C
      M(N2,N3)=M(N2,N3)+13.*B
      M(N2,N4)=M(N2,N4)-3.*C
      M(N3,N1)=M(N3,N1)+54.*A
      M(N3,N2)=M(N3,N2)+13.*B
      M(N3,N3)=M(N3,N3)+156.*A
      M(N3,N4)=M(N3,N4)-22.*B
      M(N4,N1)=M(N4,N1)-13.*B
      M(N4,N2)=M(N4,N2)-3.*C
      M(N4,N3)=M(N4,N3)-22.*B
      M(N4,N4)=M(N4,N4)+4.*C
      RETURN
      END


      SUBROUTINE SYMDET (DET,A,U,N)
      INTEGER N,C,CC,LCC,CP1,COL,RC,RCOL,DIAG,CCP1,DIAGM1
      REAL A(1),U(1),SUM
      DOUBLE PRECISION DET
      SUM=A(1)
      IF(SUM.LE.0.) GOTO 10
      U(1)=SQRT(SUM)
      DET=U(1)
      GOTO 12
10    PRINT 1
1     FORMAT(////47H SYMDET UNDEFINED, MATRIX NOT POSITIVE DEFINITE)
      IF(SUM.EQ.0.) PRINT 2
2     FORMAT(1H+,46X,17H, DETERMINANT=0.0)
      DET=0.D0
      RETURN
12    RC=2
      DO 15 C=2,N
        U(RC)=A(RC)/U(1)
15      RC=RC+C
      CC=1
      DO 35 C=2,N
        DIAG=CC+C
        CCP1=CC+1
        DIAGM1=DIAG-1
        SUM=0.
        DO 20 RC=CCP1,DIAGM1
20        SUM=SUM+U(RC)**2
        SUM=A(DIAG)-SUM
        IF(SUM.LE.0.) GOTO 10
        U(DIAG)=SQRT(SUM)
        DET=DBLE(U(DIAG))*DET
        IF(C.EQ.N) GOTO 35
        CP1=C+1
        LCC=DIAG+1
        DO 30 COL=CP1,N
          SUM=0.
          RCOL=LCC
          DO 25 RC=CCP1,DIAGM1
            SUM=SUM+U(RCOL)*U(RC)
25          RCOL=RCOL+1
          U(RCOL)=(A(RCOL)-SUM)/U(DIAG)
30      LCC=LCC+COL
35    CC=DIAG
      IF(DABS(DET).LT.1.D-153) GOTO 40
      IF(DABS(DET).GT.1.D+153) GOTO 45
      DET=DET**2
      RETURN
40    WRITE(6,41)
41    FORMAT(//' SYMDET UNDERFLOW, DETERMINATE SET TO 1.D-306 ')
      DET=1.D-306
      RETURN
45    WRITE(6,46)
46    FORMAT(//' SYMDET OVERFLOW, DETERMINATE SET TO 1.D+306')
      DET=1.D306
      RETURN
      END


      SUBROUTINE SYMINV (A,AI,N,DET)
      INTEGER NN,N,NM1,R,RP1,RI,I,IC,C,CM1,DIAG,ROW,ROWI,ROWR,FLAG
      REAL A(N),AI(N),SUM,DET
      DOUBLE PRECISION DTR
      CALL SYMDET(DTR,A,AI,N)
      FLAG=1
      DET=0.
      IF(DTR.EQ.0.D0) GOTO 1
      IF(DABS(DTR).GT.1.D38) DET =DSIGN(1.D38,DTR)
      IF(DABS(DTR).LT.1.D-38) DET =DSIGN(1.D-38,DTR)
      IF(DET.EQ.0.) DET =DTR
      GOTO 5
1     PRINT 3
3     FORMAT(25H THUS SYMINV IS UNDEFINED)
```

```
      RETURN
      ENTRY SINVRS (AI,N)
      FLAG=1
      GOTO 5
      ENTRY UINVRS (A,AI,N)
      FLAG=0
5     NN=(N**2+N)/2
      DIAG=0
      NM1=N-1
      DO 30 R=1,NM1
        DIAG=DIAG+R
        AI(DIAG)=1./AI(DIAG)
        RP1=R+1
        IC=DIAG+R
        DO 20 C=RP1,N
          RI=DIAG
          SUM=0.
          CM1=C-1
          DO 10 I=R,CM1
            SUM=SUM+AI(RI)*AI(IC)
            RI=RI+I
10          IC=IC+1
          AI(RI)=-SUM/AI(IC)
          IC=RI+C
20      CONTINUE
30    CONTINUE
      AI(NN)=1./AI(NN)
      IF(FLAG.EQ.0) RETURN
      DIAG=0
      DO 60 R=1,N
        ROWR=DIAG+1
        DIAG=DIAG+R
        DO 50 ROW=1,R
          SUM=0.
          RI=DIAG
          ROWI=ROWR
          DO 40 I=R,N
            SUM=SUM+AI(ROWI)*AI(RI)
            ROWI=ROWI+I
40          RI=RI+I
          AI(ROWR)=SUM
50      ROWR=ROWR+1
60    CONTINUE
      RETURN
      END


      SUBROUTINE KRAY(K,N,EIBEAM,EIBOOM,TOM,TIM,TOB,TIB,NW,NL,L,W,SYM)
      INTEGER N,NL,NN,NW,NZ,NWC,NBN,I,J,N1,SYM
      REAL K(1),EI,EIBOOM,EIBEAM,TOM,TIM,TOB,TIB,DTB,DTM,L,LL,LW,W,T
      EI=EIBOOM/2.
      NWC=NW+1
      NZ=NWC+NL*(2+NWC)
      NBN=NWC*(NL+1)
      LW=W/FLOAT(NW)
      LL=L/FLOAT(NL)
      DTM=(TOM-TIM)/FLOAT(NL*NW)
      DTB=(TIB-TOB)/FLOAT(NL)/2.
      T=TIM/FLOAT(NW)+DTM/2.
      NN=NBN+1
      N=NZ*NZ
      DO 5 I=1,N
5       K(I)=0.
      DO 10 I=1,NW
10      CALL KB(K,EIBEAM,LW,NN-I,I,NBN-I,I+1,NZ)
      DO 15 I=2,NWC
15      CALL KSM(K,LL,T,NZ,NW+I,NWC+I,NZ,NZ)
      N1=NWC+1
      NN=NBN-NWC
      DO 20 J=N1,NN,NWC
        T=T+DTM
        DO 20 I=1,NW
20        CALL KSM(K,LL,T,J+I-1,J+I+NW,J+I+NWC,J+I,NZ)
      IF(SYM.EG.1) GOTO 30
      N1=NBN+3
      T=(TOB-DTB)/2.
      DO 25 I=N1,NZ,2
        T=T+DTB
        CALL KSB(K,T,LL,I-1,I,I-3,I-2,NZ)
25      CALL KB(K,EI,LL,I-1,I,I-3,I-2,NZ)
      CALL KSB(K,T+DTB,LL,NZ,NZ,NZ-2,NZ-1,NZ)
      CALL KB(K,EI,LL,NZ,NZ,NZ-2,NZ-1,NZ)
      CALL PAKSUB(K,K,2,NZ-1,NZ)
      N=NZ-2
      RETURN
30    CALL PAKSUB(K,K,1,NBN-1,NZ)
      N=NBN-1
      RETURN
      END
```

```
      SUBROUTINE TRIDIA (A,DIA,CODI,CDSQ,NORM,N)
      INTEGER NN,N,NM1,I,IM1,COL,C,CP1,CCP1,LCC,ROW,R,
     1      RP1,RP2,RC,DIAG,DIAGPR
      REAL A(N),CODI(N),CDSQ(N),DIA(N),NORM,SUM,W,X
      NORM =0.
      NN=(N**2+N)/2
      NM1=N-1
      CCP1=1
      DO 20 C=1,NM1
        CP1=C+1
        DIAG=CCP1+C-1
        SUM=0.
        DO 10 RC=CCP1,DIAG
10        SUM=SUM+ABS( A(RC) )
        RC=DIAG+C
        DO 15 COL=CP1,N
          SUM=SUM+ABS( A(RC) )
15        RC=RC+COL
        IF(SUM.GT.NORM) NORM=SUM
        CCP1=CCP1+C
20    CONTINUE
      SUM=0.
      DO 25 RC=CCP1,NN
25      SUM=SUM+ABS( A(RC) )
      IF(SUM.GT.NORM) NORM=SUM
      DIAG=0.
30    DO 95 R=1,NM1
        RP1=R+1
        DIAG=DIAG+R
        DIA(R)=A(DIAG)
        DIAGPR=DIAG+R
        SUM=0.
        RC=DIAGPR
        DO 35 C=RP1,N
          SUM=SUM+A(RC)**2
35        RC=RC+C
        CDSQ(R)=SUM
        CODI(R)=SQRT(SUM)
        IF(NORM-CODI(R).NE.NORM) GOTO 40
        CODI(R)=0.
        CDSQ(R)=0.
        GOTO 95
40      IF(A(DIAGPR).GT.0.) CODI(R)=-CODI(R)
        A(DIAGPR)=A(DIAGPR)-CODI(R)
        W=A(DIAGPR)*CODI(R)
        RC=DIAGPR
        DO 50 C=RP1,N
          DIA(C)=A(RC)
50        RC=RC+C
        LCC=DIAGPR+1
        RC=LCC
        SUM=0.
        DO 55 C=RP1,N
          SUM=SUM+A(RC)*DIA(C)
55        RC=RC+C
        CODI(RP1)=SUM/W
        RP2=R+2
        LCC=LCC+RP1
        IF(R.EQ.NM1) GOTO 71
        DO 70 I=RP2,N
          IM1=I-1
          RC=LCC
          SUM=0.
          DO 60 ROW=RP1,IM1
            SUM=SUM+A(RC)*DIA(ROW)
60          RC=RC+1
          DO 65 C=I,N
            SUM=SUM+A(RC)*DIA(C)
65          RC=RC+C
          CODI(I)=SUM/W
          LCC=LCC+I
70      CONTINUE
71      SUM=0.
        DO 75 I=RP1,N
75        SUM=SUM+DIA(I)*CODI(I)
        X=SUM/W/2.
        DO 80 I=RP1,N
80        CODI(I)=DIA(I)*X+CODI(I)
        LCC=DIAGPR+1
        DO 90 C=RP1,N
          RC=LCC
          DO 85 ROW= RP1,C
            A(RC)=DIA(ROW)*CODI(C)+CODI(ROW)*DIA(C)+A(RC)
85          RC=RC+1
          LCC=LCC+C
90      CONTINUE
95    CONTINUE
      CODI(N)=0.
      CDSQ(N)=0.
      DIA(N)=A(NN)
      RETURN
      END
```

```
      SUBROUTINE TRIVEC(DI,CODI,V,IGNVAL,NUM,NORM,N)
      INTEGER N,NUM,I,IP1,J,NM1,ITR
      REAL DI(N),CODI(N),V(N),IGNVAL(N),X,X1,X2,Y,Z,SUM,C,E,S,
     1      P(100),Q(100),W(100),NORM,EPS,PRECS
      LOGICAL INTNM1,INT(100)
      COMMON/SR4/P/SR5/Q/SR6/INT/SR7/W
      PRECS=1.E-8
      X1=0.
      X2=SQRT(2.)
      NM1=N-1
      E=IGNVAL(NUM)
      IF(NUM.EQ.1) GOTO 10
      IF(ABS(E-IGNVAL(NUM-1)).LT.E*PRECS*100. ) GOTO 15
10    DO 12 I=1,N
12      V(I)=1.
      GOTO 25
15    DO 20 I=1,N
        X=AMOD(X1+X2,2.)
        X1=X2
        X2=X
20      V(I)=X-1.
25    EPS=PRECS*NORM/FLOAT(N)**2
      Y=1.
      X=(DI(1)-E)/NORM
      DO 35 I=1,NM1
        IP1=I+1
        S=CODI(I)/NORM
        C=(DI(IP1)-E)/NORM
        Y=S*Y
        IF(ABS(X).GE.ABS(S)) GOTO 30
        P(I)=S
        Q(I)=C
        INT(I)=.TRUE.
        Z=-X/S
        X=Y+Z*C
        Y=Z
        GOTO 35
30      IF(ABS(X).LT.EPS) X=EPS
        P(I)=X
        Q(I)=Y
        INT(I)=.FALSE.
        Z=-S/X
        X=C+Z*Y
        Y=1.
35    W(I)=Z
      INTNM1=INT(NM1)
      IF(ABS(X).LT.EPS) X=EPS
      ITR=0
40      ITR=ITR+1
        V(N)=V(N)/X
        INT(NM1)=.FALSE.
        SUM=V(N)**2
        DO 45 J=1,NM1
          I=N-J
          Y=V(I)-Q(I)*V(I+1)
          IF(INT(I)) Y=Y-CODI(I+1)*V(I+2)/NORM
          V(I)=Y/P(I)
45        SUM=SUM+V(I)**2
        S=SQRT(SUM)
        DO 50 I=1,N
50        V(I)=V(I)/S
        IF(ITR.GE.2) RETURN
        INT(NM1)=INTNM1
        DO 60 I=1,NM1
          IP1=I+1
          Z=W(I)
          IF(INT(I)) GOTO 55
          V(IP1)=V(IP1)+Z*V(I)
          GOTO 60
55        Y=V(I)
          V(I)=V(IP1)
          V(IP1)=Y+Z*V(I)
60      CONTINUE
      GOTO 40
      END


      SUBROUTINE CPYSUB(M,SUB,R1,R2,C1,C2,N)
      INTEGER R1,R2,C1,C2,R1C1,R1C2,R1C,R2C,RC,IJ,N,NN
      REAL M(1),SUB(1)
      R1C1=N*(C1-1)+R1
      R1C2=N*C2
      R2C=R1C1+R2-R1
      NN=N
      GOTO 10
      ENTRY PAKSUB(M,SUB,R1,R2,N)
      R1C1=N*(R1-1)+R1
      R1C2=N*R2
      R2C=R1C1
      NN=N+1
10    IJ=0
      DO 20 R1C=R1C1,R1C2,N
        DO 15 RC=R1C,R2C
          IJ=IJ+1
15        SUB(IJ)=M(RC)
20      R2C=R2C+NN
      RETURN
      END
```

```
      SUBROUTINE MRAY(M,N,ROBEAM,ROBOOM,ROARAY,NW,NL,L,W,SYM)
      INTEGER N,NL,NN,NW,NZ,NWC,NBN,I,J,N1,SYM
      REAL M(1),ROBEAM,ROBOOM,ROARAY,WO,WE,WA,L,LW,LL,W
      NWC=NW+1
      NZ=NWC+NL*(2+NWC)
      NBN=NWC*(NL+1)
      LW=W/FLOAT(NW)
      LL=L/FLOAT(NL)
      WO=ROBOOM*LL/2.
      WE=ROBEAM*LW
      WA=ROARAY*LL*LW
      NN=NBN+1
      N=NZ*NZ
      DO 5 I=1,N
5       M(I)=0.
      DO 10 I=1,NW
10      CALL MB(M,WE,LW,NN-I,I,NBN-I,I+1,NZ)
      DO 15 I=2,NWC
15      CALL MMEM(M,WA,NZ,NW+I,NWC+I,NZ,NZ)
      N1=NWC+1
      NN=NBN-NWC
      DO 20 J=N1,NN,NWC
        DO 20 I=1,NW
20        CALL MMEM(M,WA,J+I-1,J+I+NW,J+I+NWC,J+I,NZ)
      IF(SYM .EQ. 1) GOTO 30
      N1=NBN+3
      DO 25 I=N1,NZ,2
25      CALL MB(M,WO,LL,I-1,I,I-3,I-2,NZ)
      CALL MB(M,WO,LL,NZ,NZ,NZ-2,NZ-1,NZ)
      CALL PAKSUB(M,M,2,NZ-1,NZ)
      N=NZ-2
      RETURN
30    CALL PAKSUB(M,M,1,NBN-1,NZ)
      N=NBN-1
      RETURN
      END


      SUBROUTINE  UTRAU (U,A,UTAU,N,SRATCH)
      INTEGER N,C,CC,CCPR,LCC,COL,R,RC,RCPI,RI,I,IP1,DIAG
      REAL A(N),U(N),UTAU(N),SRATCH(N),SUM
      CC=(N**2+N)/2
      C=N
10    CC=CC-C
        DIAG=0
        DO 40 I=1,C
          LCC=DIAG
          DIAG=DIAG+I
          RI=LCC
          RC=CC
          SUM=0.
20        RI=RI+1
            RC=RC+1
            SUM=SUM+A(RI)*U(RC)
          IF(RI.LT.DIAG) GOTO 20
          IF(I.EQ.C) GOTO 35
          IP1=I+1
          RI=RI+I
          DO 30 COL=IP1,C
            RC=RC+1
            SUM=SUM+A(RI)*U(RC)
30          RI=RI+COL
35        SRATCH(I)=SUM
40      CONTINUE
        RC=CC+C
        R=C
45      RC=RC-R
          SUM=0.
          DO 50 I=1,R
            RCPI=RC+I
50          SUM=SUM+U(RCPI)*SRATCH(I)
          CCPR=CC+R
          UTAU(CCPR)=SUM
          R=R-1
        IF(R.GT.0) GOTO 45
        C=C-1
      IF(C.GT.1) GOTO 10
      UTAU(1)=A(1)*U(1)**2
      RETURN
      END


      SUBROUTINE UXM(U,M,UM,N,P)
      INTEGER N,P,NP,NM1,NC,OC,CC,RCC,RI,IC
      REAL U(N),M(N),UM(N),SUM
      NM1=N-1
      NP=N*P
      DO 30 OC=1,NP,N
        NC=OC+NM1
        RCC=1
        DO 20 CC=OC,NC
          RCC=RCC+CC-OC
          RI=RCC
          SUM=0.
          DO 10 IC=CC,NC
```

```
                RI=RI+IC-OC
10              SUM=SUM+U(RI)*M(IC)
             UM(CC)=SUM
20        CONTINUE
30      CONTINUE
        RETURN
        END


        SUBROUTINE TRIBAK (A,CODI,V,N)
        INTEGER N,NM1,C,I,R,RP1,RC,DIAGPR
        REAL A(N),CODI(N),V(N),SUM,F,X
        NM1=N-1
        DIAGPR=(N**2+N)/2-1
        R=N
        DO 20 I=1,NM1
          R=N-I
          RC=DIAGPR
          RP1=R+1
          IF(CODI(R).EQ.0.) GOTO 20
          X=A(RC)
          SUM=0.
          DO 10 C=RP1,N
            SUM=SUM+A(RC)*V(C)
10          RC=RC+C
          F=SUM/(X*CODI(R))
          RC=DIAGPR
          DO 15 C=RP1,N
            V(C)=A(RC)*F+V(C)
15          RC=RC+C
20        DIAGPR = DIAGPR-RP1
        RETURN
        END


        SUBROUTINE MMEM(M,W,N1,N2,N3,N4,N)
        INTEGER N1,N2,N3,N4,N
        REAL M(N,1),W
        A=W/36./32.2
        B=2.*A
        C=2.*B
        M(N1,N1)=M(N1,N1)+C
        M(N1,N2)=M(N1,N2)+B
        M(N1,N3)=M(N1,N3)+A
        M(N1,N4)=M(N1,N4)+B
        M(N2,N1)=M(N2,N1)+B
        M(N2,N2)=M(N2,N2)+C
        M(N2,N3)=M(N2,N3)+B
        M(N2,N4)=M(N2,N4)+A
        M(N3,N1)=M(N3,N1)+A
        M(N3,N2)=M(N3,N2)+B
        M(N3,N3)=M(N3,N3)+C
        M(N3,N4)=M(N3,N4)+B
        M(N4,N1)=M(N4,N1)+B
        M(N4,N2)=M(N4,N2)+A
        M(N4,N3)=M(N4,N3)+B
        M(N4,N4)=M(N4,N4)+C
        RETURN
        END

        SUBROUTINE KB(K,EI,L,N1,N2,N3,N4,N)
        INTEGER N1,N2,N3,N4,N
        REAL K(N,1),EI,L
        A=2.*EI/L
        B=3.*A/L
        C=2.*B/L
        K(N1,N1)=K(N1,N1)+C
        K(N1,N2)=K(N1,N2)+B
        K(N1,N3)=K(N1,N3)-C
        K(N1,N4)=K(N1,N4)+B
        K(N2,N1)=K(N2,N1)+B
        K(N2,N2)=K(N2,N2)+2.*A
        K(N2,N3)=K(N2,N3)-B
        K(N2,N4)=K(N2,N4)+A
        K(N3,N1)=K(N3,N1)-C
        K(N3,N2)=K(N3,N2)-B
        K(N3,N3)=K(N3,N3)+C
        K(N3,N4)=K(N3,N4)-B
        K(N4,N1)=K(N4,N1)+B
        K(N4,N2)=K(N4,N2)+A
        K(N4,N3)=K(N4,N3)-B
        K(N4,N4)=K(N4,N4)+2.*A
        RETURN
        END
```

```
SUBROUTINE KSM(K,L,T,N1,N2,N3,N4,N)
INTEGER N1,N2,N3,N4,N
REAL T,L,K(N,1)
A=T/6./L
B=2.*A
K(N1,N1)=K(N1,N1)+B
K(N1,N2)=K(N1,N2)-B
K(N1,N3)=K(N1,N3)-A
K(N1,N4)=K(N1,N4)+A
K(N2,N1)=K(N2,N1)-B
K(N2,N2)=K(N2,N2)+B
K(N2,N3)=K(N2,N3)+A
K(N2,N4)=K(N2,N4)-A
K(N3,N1)=K(N3,N1)-A
K(N3,N2)=K(N3,N2)+A
K(N3,N3)=K(N3,N3)+B
K(N3,N4)=K(N3,N4)-B
K(N4,N1)=K(N4,N1)+A
K(N4,N2)=K(N4,N2)-A
K(N4,N3)=K(N4,N3)-B
K(N4,N4)=K(N4,N4)+B
RETURN
END
```

```
SUBROUTINE KSB(K,T,L,N1,N2,N3,N4,N)
INTEGER N1,N2,N3,N4,N
REAL K(N,1),T,L
A=6.*T/5./L
B=T/10.
C=T*L/30.
K(N1,N1)=K(N1,N1)+A
K(N1,N2)=K(N1,N2)+B
K(N1,N3)=K(N1,N3)-A
K(N1,N4)=K(N1,N4)+B
K(N2,N1)=K(N2,N1)+B
K(N2,N2)=K(N2,N2)+4.*C
K(N2,N3)=K(N2,N3)-B
K(N2,N4)=K(N2,N4)-C
K(N3,N1)=K(N3,N1)-A
K(N3,N2)=K(N3,N2)-B
K(N3,N3)=K(N3,N3)+A
K(N3,N4)=K(N3,N4)-B
K(N4,N1)=K(N4,N1)+B
K(N4,N2)=K(N4,N2)-C
K(N4,N3)=K(N4,N3)-B
K(N4,N4)=K(N4,N4)+4.*C
RETURN
END
```

NASA — JPL — Coml., L.A., Calif.