N71-32690

NASA CONTRACTOR REPORT

NASA CR-111919

NASA CR-111919

MASTER AGREEMENT TASK ORDER FIVE

Analyses and Limited Evaluation of Payload and Legged Landing System Structures For The Survivable Soft Landing of Instrument Payloads

By O. R. Otto, R. M. Laurenson, R. A. Melliere, and R. L. Moore

Prepared by
MCDONNELL DOUGLAS ASTRONAUTICS COMPANY – EAST
St. Louis, Missouri 63166 (314) 232-0232
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D.C. • JULY 1971

NASA CR-111919

MASTER AGREEMENT
TASK ORDER FIVE

Analyses and Limited Evaluation of Payload and Legged Landing System Structures For The Survivable Soft Landing of Instrument Payloads

By O. R. Otto, R. M. Laurenson,
R. A. Melliere, and R. L. Moore

Prepared Under Contract No. NAS 1-8137 by

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST

Saint Louis, Missouri

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This report describes two computer programs developed for the investigation of legged planetary landers. Analytical methods incorporated in the programs, operating instructions, and examples of program utilization are described. One program developed is the Structural Analysis Program, containing a Center Body Option, which can be used to analyze a structure by utilizing a finite element idealization. It also contains a Landing Gear Option for determination of the energy absorption and load-stroke characteristics of either a cantilever or inverted tripod landing gear configuration. The second program is the Landing Loads and Motions Program, used to predict spatial landing dynamics of a legged lander containing up to five landing gears. This program also contains options for including the effects of a flexible center body and determining lander stability.

Several exemplary computer runs are discussed to aid in interpretation of the operating instructions and to illustrate various available program options. Analytical results for several of these cases are compared to test data obtained during model test programs conducted at NASA Langley Research Center.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

PAGE

TABLE OF CONTENTS (Continued)

APPENDIX

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF PAGES

i thru xiii

1 thru 479

THIS PAGE INTENTIONALLY LEFT BLANK

1. INTRODUCTION

This report describes two computer programs developed by McDonnell Douglas Astronautics Company - East under NASA Contract NAS-1-8137(U) for investigation of legged type planetary landers. They are the Structural Analysis Program and the Landing Loads and Motions Program. The Structural Analysis Program contains options for the determination of internal center body load distributions, center body modal information, and the large displacement behavior of either an inverted tripod or a cantilever gear configuration. The Landing Loads and Motions Program can be employed for the determination of the spatial landing motions of a lander idealized either as a rigid body or as a flexible body. In addition, options are available for the determination of overall landing loads and lander stability boundaries.

Results of analyses conducted on typical legged lander configurations are presented and compared with test results obtained at NASA Langley Research Center. Test results were obtained using the Task Order Three Lander(Reference (1)) shown in Figure 1-1. These analyses demonstrate the capabilities of the developed computer programs.

The programs developed during Task Order Five provide the capability of conducting a complete design study on a legged lander beginning at the preliminary design phase and continuing through a final evaluation of the lander's structural configuration. A flow diagram of this capability is presented in Figure 1-2. Program options permit detailed analyses to be conducted at any point during the evolution of a legged lander. For example, analytical investigations for boiler plate models, scale models, or final lander configurations can be conducted.

During the preliminary design phase, various landing gear configurations are evaluated with the Landing Gear Option of the Structural Analysis Program. At the same time the landing loads and stability of a lander idealized with a rigid center body are investigated with the Landing Loads and Motions Program. Results of these analyses are used to select a final landing gear configuration. In addition, the predicted rigid body landing loads are used in conjunction with the Center Body Option of the Structural Analysis Program to determine internal load distributions in the center body structure. Based on

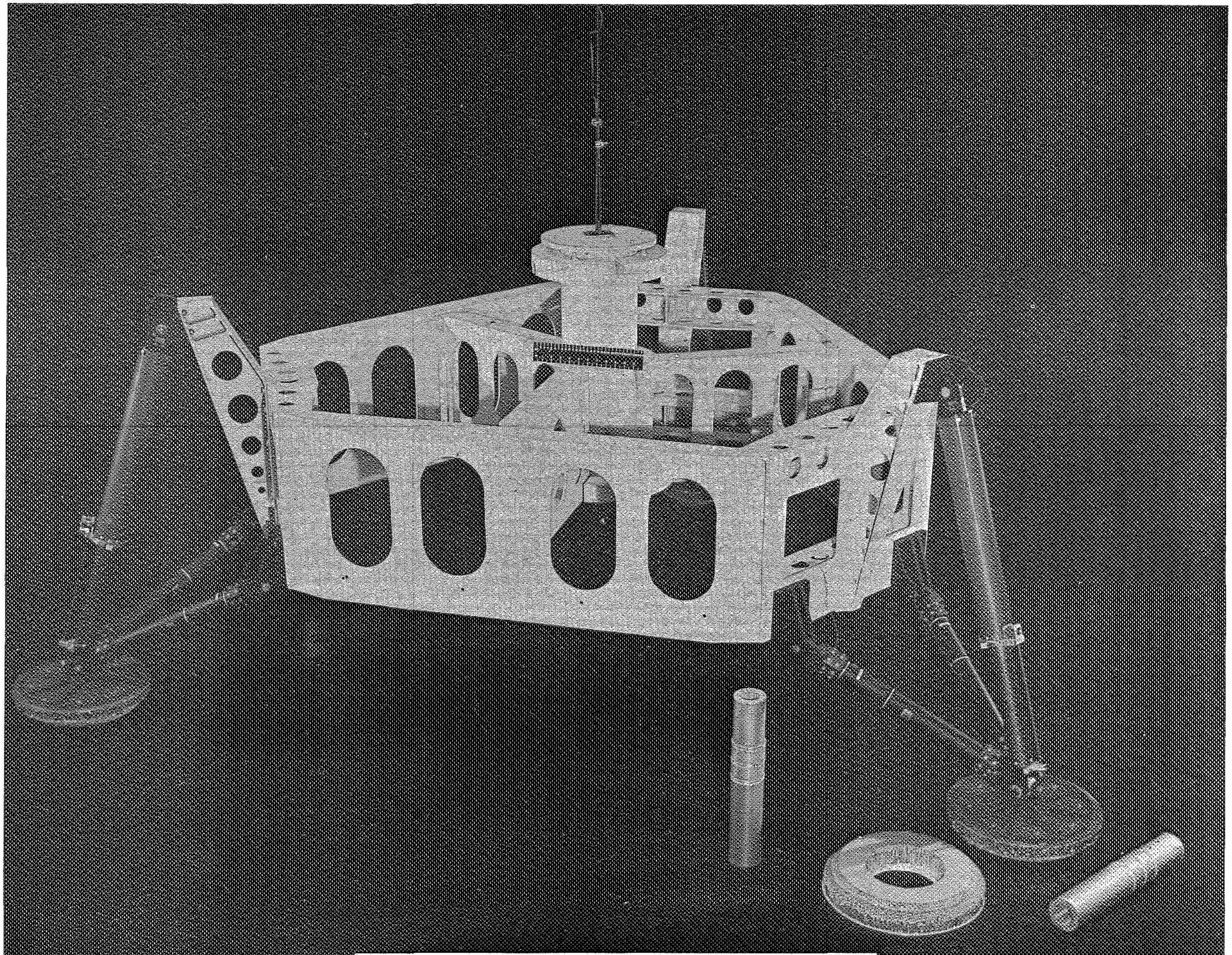
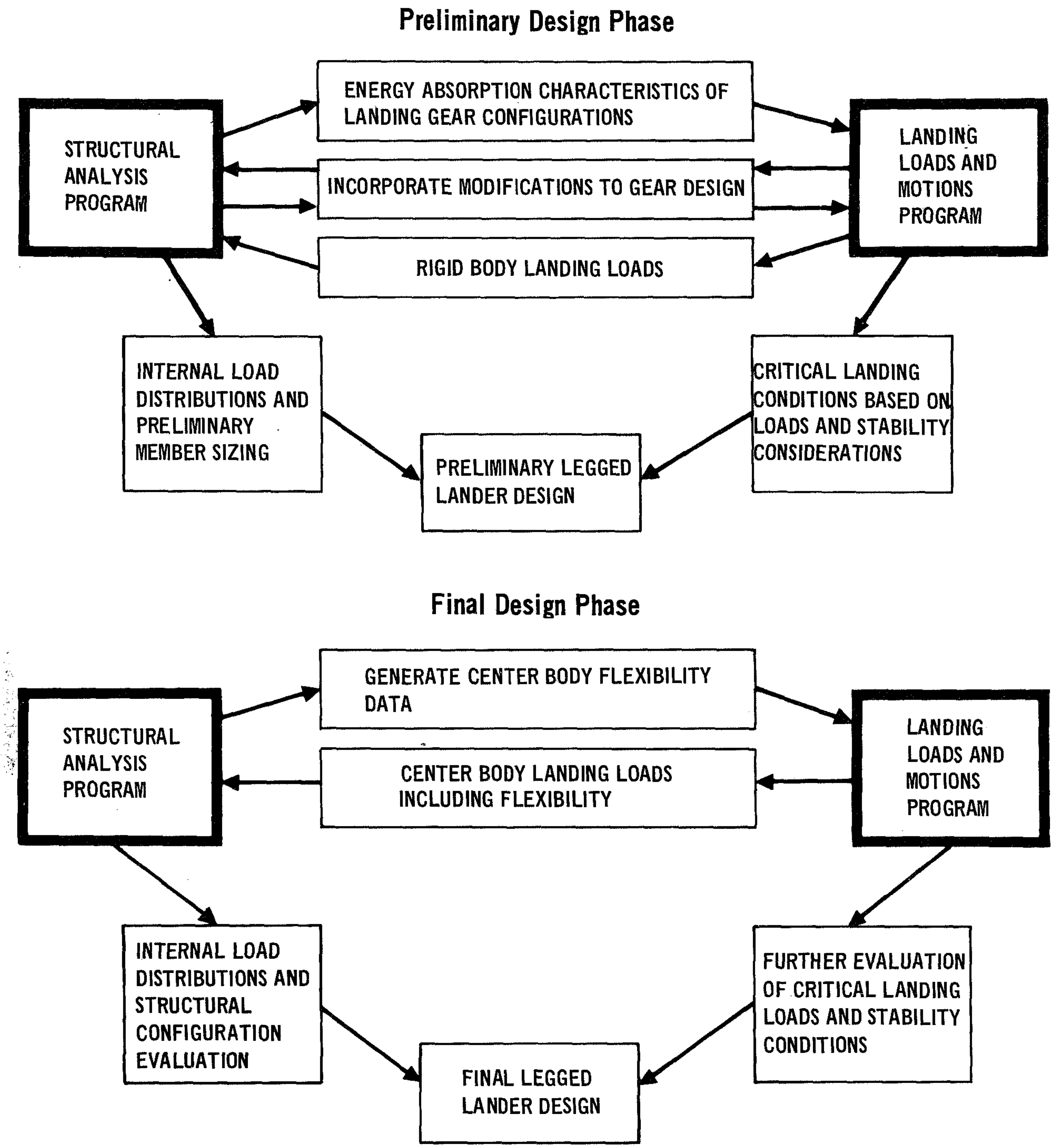


FIGURE 1–1 TASK ORDER THREE LANDER



Preliminary Design Phase

Final Design Phase

FIGURE 1–2 TASK ORDER FIVE PROGRAM USAGE DIAGRAM the results of this preliminary design phase, the lander's structural arrangement is selected. At this time, the programs are used to determine center body flexibility information, landing loads, and stability conditions for the flexible lander, and internal load distributions for the final center body structural configuration.

2. COMPUTER PROGRAM CAPABILITIES

The two computer programs developed during Task Order Five contain many options which permit analyses of a wide variety of legged lander configurations. Legged lander configurations considered and the associated program capabilities and options are discussed in the following sections.

2.1 Legged Lander Description - The general arrangement of a typical legged lander configuration is shown in Figure 2-1. The legged lander is composed of three basic components: (1) center body, (2) payload and auxiliary equipment, and (3) landing gear. Items included in each component are shown in Figure 2-2.

The two landing gear configurations reflected in the analysis are shown in Figure 2-3. They are the inverted tripod gear and the cantilever gear. The main strut of both gears incorporates an energy absorption system. Drag struts may also include an energy absorption system or for the inverted tripod gear, they may simply be frame members stabilizing the main strut.

2.2 Structural Analysis Program - Two major options are available in the Structural Analysis Program. The first of these is the Center Body Option for the small displacement, finite element analysis of a legged lander center body structure. In formulating the Center Body Option, the Structural Analysis Program written for Task Order Two, Reference (2), was employed as a nucleus. During Task Order Five, many improvements and additions were made to this baseline program. In addition, a Landing Gear Option was developed for determining the large displacement stroking behavior of an individual landing gear. This is a major addition to the capabilities of the Task Order Two version of the Structural Analysis Program.

2.2.1 Center Body Option - The Center Body Option of the Structural Analysis Program is formulated for solution of redundant structures based on the small displacement finite element stiffness method and utilizing iterative methods of solution. The program can determine internal and external load distributions and deflection patterns for any system of forces or deflections impressed on a network of structural bar members and shear webs. Plastic

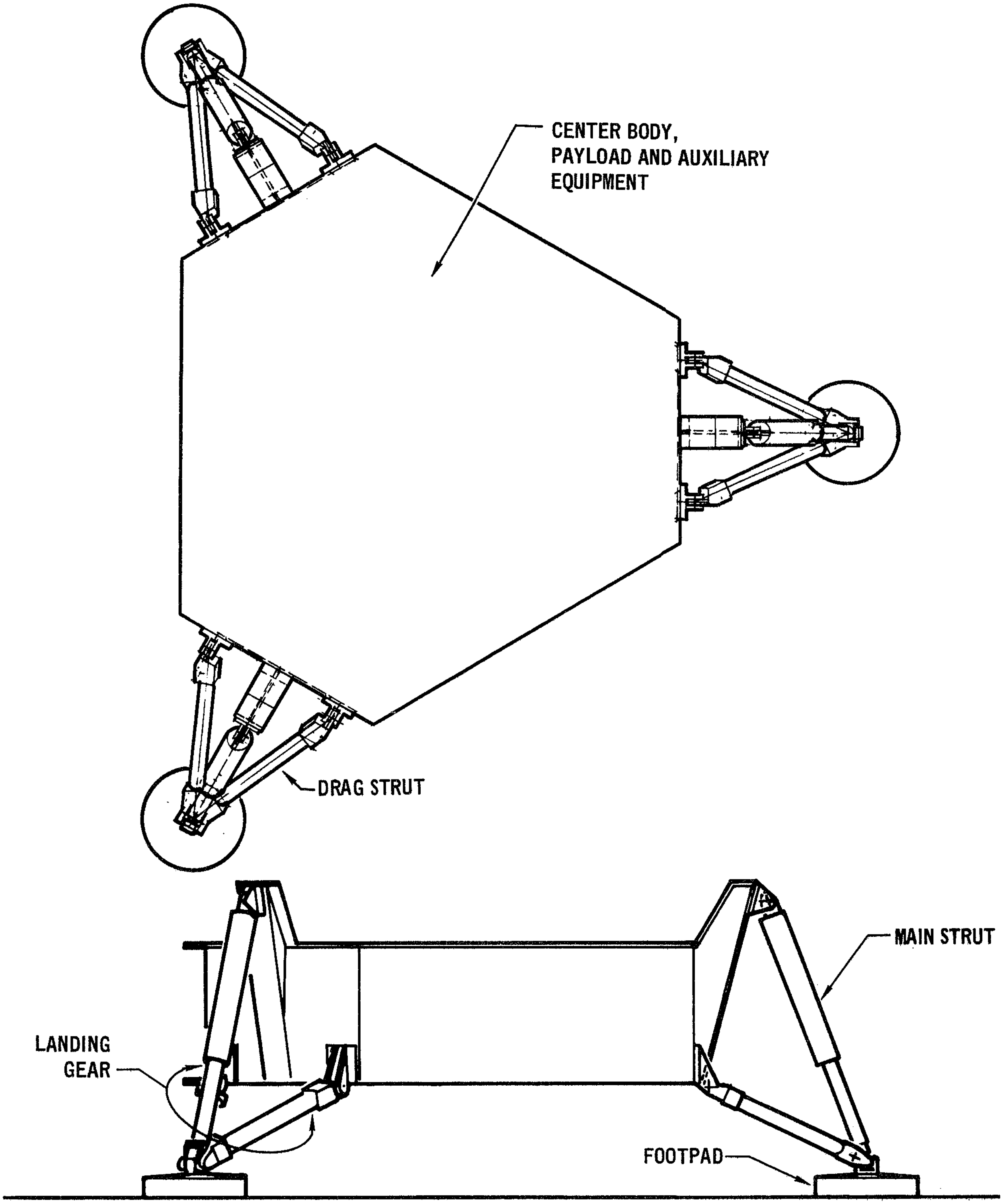
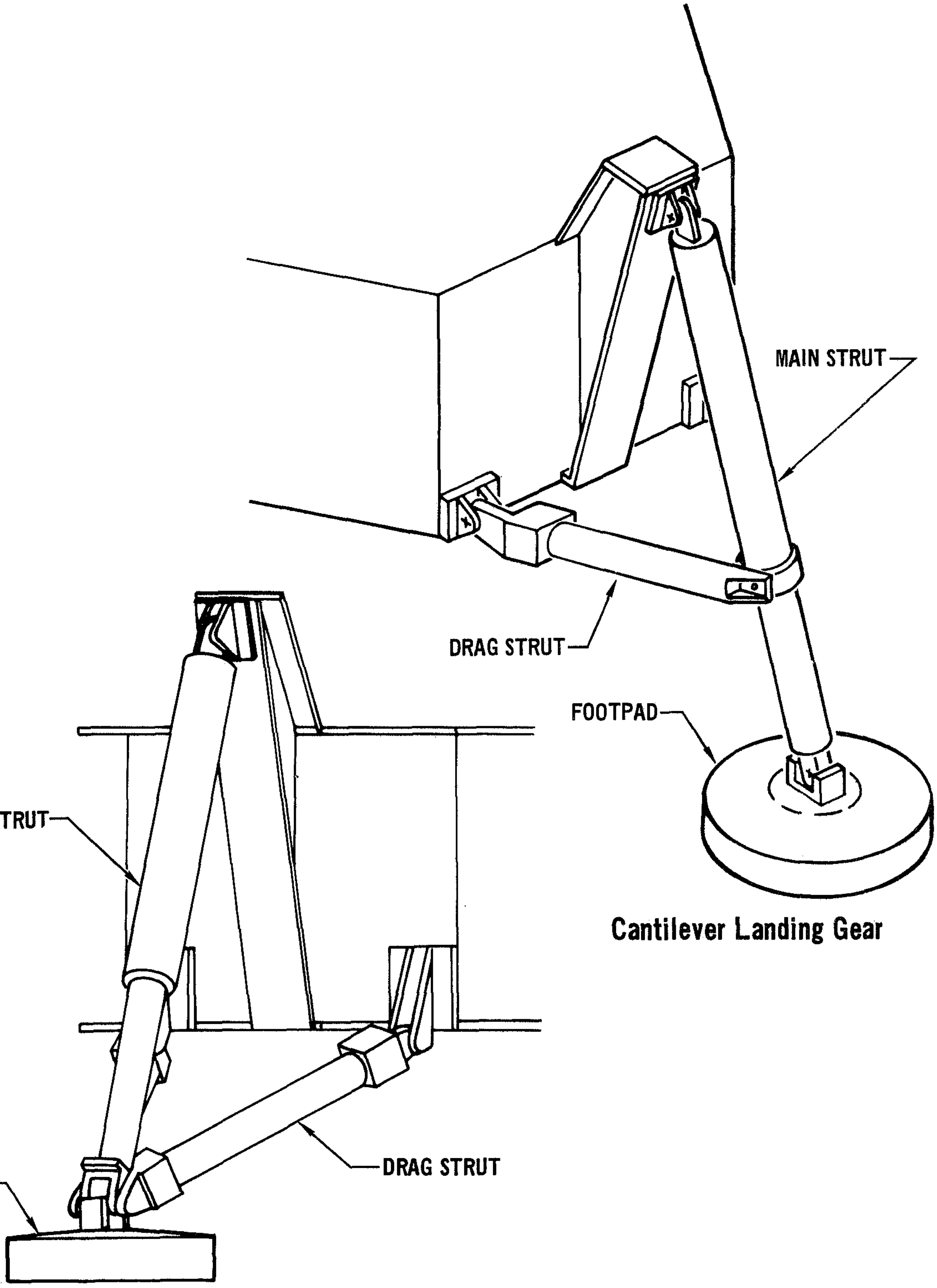


FIGURE 2–1 LEGGED LANDER

- CENTER BODY
  - LANDING GEAR CARRY-THROUGH STRUCTURE
  - PAYLOAD SUPPORT STRUCTURE
  - AUXILIARY EQUIPMENT SUPPORT STRUCTURE
- PAYLOAD AND AUXILIARY EQUIPMENT
  - FACSIMILE CAMERAS
  - ATMOSPHERIC SENSORS
  - SOIL ACQUISITION MECHANISM
  - SCIENCE INSTRUMENTS
  - COMMUNICATIONS EQUIPMENT
  - ELECTRONIC EQUIPMENT
  - POWER SUPPLY
  - TERMINAL PROPULSION SYSTEM
  - THERMAL CONTROL SYSTEM
- LANDING GEAR
  - FOOTPAD
  - MAIN STRUT
  - DRAG STRUTS

FIGURE 2-2 LEGGED LANDER COMPONENTS



Cantilever Landing Gear

Inverted Tripod Landing Gear

FIGURE 2–3 LANDING GEAR CONCEPTS

behavior of support structure can be simulated with the program. The program also allows simulation of cables, or any support members with restricted load carrying capability in certain directions. Mode shapes and natural frequencies may be obtained using this program. For modal analysis, a routine is incorporated to allow reduction in the size of the stiffness matrix for large structures, thus permitting a reduction in the required computer storage requirements and run time.

New capabilities available in this option of the Structural Analysis Program over the Task Order Two version follow. An optional iterative technique, the Conjugate Gradient Method, was added to increase program efficiency for solution of nodal deflections. A more efficient data input routine was programmed reducing core size required by this routine. The maximum number of bars which can be used to idealize a structure was increased from 74 to 130. A rectangular shear web element was added to expand idealization capability. An additional 2500 words of storage was provided for storing nonzero terms of the stiffness matrix. An improved matrix reduction routine was developed to increase efficiency when employing the modal analysis routine. The maximum number of lowest natural frequencies and associated mode shapes (excluding rigid body modes) that can be requested from the modal analysis routine was increased from 5 to 95. In addition, the capability for handling multiple data cases was added. These capabilities result in a program whose core storage requirements at execution utilize the maximum allowable storage (70K) specified for computer programs by NASA Langley Research Center.

The iterative method programmed for solution of nodal deflections was selected to minimize computer core storage allocated to the solution routine. It also permits minimization of computer time for a preliminary design problem, since accuracy requirements can be specified.

Both straight bar members and rectangular shear webs are available for idealization of the center body structure. As explained in Section 4.1.1.2, the program internally replaces a shear web by two diagonal bar elements. Hence the bar element is the basic element of the Center Body Option.

Each bar element is capable of carrying axial load, shear in two directions, bending in two directions, and torsion. Bar properties are assumed to be constant between nodes (symmetry about one of the principal axes is assumed). The bar's longitudinal axis is assumed to pass through the centroidal axis and the bending neutral axes are assumed to align with centroidal axes. Torsional shear center of the bar is assumed to be on the bar's longitudinal axis.

Cables, which have one end attached to a support and the other to a structural joint, may be idealized by restricting the compression load carrying capability to zero, thus exercising the plastic analysis program logic.

Plastic behavior of support structure, such as crushable attenuators, may be simulated by inputting the force causing plastic deformation of each support member. An upper and lower limit must be input on the particular force or moment being constrained.

The program is based on small deflection theory with Hooke's Law applying, except with regard to the plastic support option of the program. Buckling of members is not considered, and coupling effects, such as occur with beam-columns, are neglected. Bars are assumed to be rigidly connected to each other, unless otherwise specified. Bars pinned at both ends may be simulated by setting the appropriate moments of inertia equal to zero. Bars pinned at one end and fixed at the other may be idealized through the use of special input indicators. Loads are applied to the joints as concentrated forces in global coordinates.

Computer core storage restrictions necessitated a number of limitations on the problem size which may be considered with the Center Body Option. As described in the operating instructions, this storage can be increased to accommodate larger problems. Within the current limitations, the allowable problem size is governed by the following considerations: up to 74 joints with six degrees of freedom at each joint, or a maximum assembled stiffness matrix 444 by 444; 130 bars maximum; 30 shear webs maximum, each of which reduces the allowable number of bars by two; 26 reference points maximum; 300 support constraints maximum; 88 plastic force constraints maximum (88 upper and 88 lower limits); and 88 nonzero input deflections maximum. Mode shapes and natural frequencies can be determined for a maximum assembled stiffness matrix size of 102 by 102 which corresponds to 17 joints with six degrees of freedom or many joints with fewer degrees of freedom. A reduction routine permits modal analysis of large, complex structures because the associated large stiffness matrix (444 x 444 maximum) may be reduced to the allowable size (102 by 102) by selectively eliminating degrees of freedom. The program utilizes the mode shapes of the reduced system to generate mode shapes for all degrees of freedom of the original large stiffness matrix. The number of lowest natural frequencies and associated mode shapes requested from the modal analysis routine for any structure with free-free support must be a multiple of five and less than the order of the reduced stiffness matrix minus six.

2.2.2 Landing Gear Option - The Landing Gear Option of the Structural Analysis Program provides the capability for investigating the large displacement stroking behavior, energy absorption characteristics, and internal load distribution in an inverted tripod or cantilever landing gear. These characteristics are obtained by solving the displacement equations of equilibrium at specified steps of applied footpad joint parameters. These parameters can be specified in either of two ways: (1) a displacement normal to an arbitrarily oriented landing surface and the friction coefficient of the surface can be specified or (2) all three orthogonal displacement components in the Surface Coordinate System can be specified. The number of steps to be used for investigating the displaced state must be specified. The landing surface can be arbitrarily oriented with respect to the lander by specifying three Euler angles which orient the Surface Coordinate System relative to the Lander Coordinate System. A least squares solution technique is employed to solve the equations of equilibrium for the displacements at each step. The gear geometry and strut properties are arbitrary input quantities.

The program determines the external forces and displacements at the node points of the gear in the Surface Coordinate System, the internal loads in all struts, and the total energy components absorbed by the gear in both the Surface and Lander Coordinate Systems. The energy associated with crushing of footpad attenuation material can also be included for a footpad with up to three honeycomb crush levels.

The inverted tripod landing gear is idealized in the program with four nodes and three pin-ended elements representing the main strut and two drag struts. These elements have only axial load-carrying capability and may contain attenuation material with up to five crush levels in both tension and compression.

The cantilever landing gear is idealized in the program with five nodes and four elements. The main strut is idealized with two elements both of which have bending and axial load-carrying capability. These members are assumed to be rigidly connected at the center junction of the main strut. The junctions of the main strut with the center body and the footpad are assumed to be pinned. The drag struts are pin-ended with axial load-carrying capability only. The drag struts and the lower element of the main strut contain attenuation material with up to five crush levels in both tension and compression.

For both the inverted tripod and cantilever gear idealizations, all element properties are assumed constant between nodes. None of the elements of either gear is assumed to be capable of carrying torsional loads. The bending moment of inertia about any axis normal to the longitudinal axis of the main strut of the cantilever gear is assumed to be constant and beam-column effects are not included. Each strut of either gear may be of a different material.

2.3 Landing Loads and Motions Program - The Landing Loads and Motions Program predicts the landing dynamics of a legged lander. The lander is idealized as a center body structure to which the landing gears are attached. A small footpad which contacts the landing surface is located at the base of each landing gear. Program options are available for obtaining landing motions, landing loads, and stability information for planar or spatial landings on many different types of landing surfaces.

The lander center body may be idealized as either a rigid body or the effects of a flexible structure may be included. For a rigid center body up to six rigid body degrees of freedom may be included in the analysis. To conserve computer run time, any combination of center body rigid degrees of freedom may be suppressed when running planar landing cases. The flexible center body is represented by the superposition of a number of free-free vibratory modes on the rigid body motion. From one to five modes may be included in the analysis. Flexible center body information may be obtained from either the Center Body Option of the Structural Analysis Program or some other eigenvalue program.

For a given legged lander configuration, up to five gears may be considered which may be either inverted tripod or cantilever gears. Each gear consists of a main strut and two drag struts which have pinned ends; thus, no moments or torques may be introduced at their ends. Both the main strut and drag struts are capable of carrying tension and compression loads and may possess velocity dependent force characteristics, elastic-plastic load-stroke characteristics, or a combination of the two. Five plastic load levels are available in both tension and compression for all of the landing gear struts. The load-stroke characteristics of all main struts in a given lander configuration are the same. Likewise, these characteristics for all the drag struts are the same, however, they may be different than the main struts. For a cantilever gear, the effect of main strut bending is included by modifying the elastic portion of the drag strut load-stroke relationship. Any combination of a constant magnitude Coulomb friction force, or a damping force dependent on the magnitude of the strut stroking velocity may be included in either the main struts, drag struts, or both. These friction forces are directed opposite to the strut stroking velocity. Relative motion between the center body and each footpad is employed in determining the magnitudes and directions of the landing gear strut loads.

Each footpad is represented as a single mass with three rigid body translational degrees of freedom. One degree of freedom is normal to the landing surface and the other two are in the plane of the landing surface. On an optional basis, a plastic load attenuation material, with up to three crush levels, may be located on the bottom of each footpad. For footpads whose equations of motion are not being integrated, the associated gears are assumed to be extensions of the center body structure and their inertia effects are included in the center body equations of motion.

Two soil mechanics routines are available for studying the footpad-soil interaction phenomenon. One method is similar to the footpad-soil interaction analysis developed during the Lunar Module Soil Mechanics Study. In this case, the soil is represented in terms of a number of semiempirical relationships. The second method determines the soil force through a simple elastic-plastic relationship between soil pressure and depth of soil penetration in conjunction with a coefficient of friction.

In addition to the standard program output defining the time histories of the footpad and center body motions and the load-stroke time histories in each landing gear strut, a number of additional output options are available. Accelerations at as many as ten points on the center body may be determined. This option allows the determination of landing accelerations for equipment items throughout the center body or correlation with test data obtained from accelerometers located at points other than the center of gravity.

The time history of the lander stability angle and pitching velocity of the lander in the direction of minimum stability may be obtained. This option, in conjunction with the spatial capabilities of the program and the two soil mechanics routines, allows a comprehensive study of the lander's stability characteristics to be made.

On an optional basis, all of the time history quantities required for defining the landing loads and acceleration patterns throughout the center body structure may be output on a magnetic tape. Points in time corresponding to possible high center body landing loads are determined by evaluating the printed output data obtained from the Landing Loads and Motions Program. The data on the magnetic tape may then be retrieved and the total landing load distribution throughout the center body structure determined at these specific time points of interest. The resulting loads are a combination of the inertia loads, gravity loads, and landing gear loads acting on the center body structure. These landing loads can then be input to the Center Body Option of the Structural Analysis Program and the internal loads in the individual structural members obtained.

Two numerical integration methods are incorporated in the program. These consist of a constant step Runge-Kutta method and a variable step Runge-Kutta method. The constant step method allows definite user control over integration step size. The variable step method often results in less computer time for long runs such as required for stability cases.

THIS PAGE INTENTIONALLY LEFT BLANK

3. STRUCTURAL DESIGN CRITERIA

In the design of a legged lander configuration, specific constraints and criteria are imposed on the design dependent on the mission to be accomplished by the vehicle. Presented in the following sections are typical design constraints and factors of safety to be used during the design process of a legged lander. These were used as guidelines in determining the capabilities and options to be incorporated in the two computer programs developed during Task Order Five. However, it is emphasized that these do not represent limitations on the computer programs.

3.1 Design Constraints - The following factors must be considered in landing system and payload structure design: simplicity, reliability, stowability, structural compatibility, environmental compatibility, weight and sterilizability. Methods must be provided for accomplishing postlanding payload exposure to permit operations of experiments such as bioscience and imagery; measurements of wind velocity and direction, ambient pressure, temperature, and humidity; determination of soil composition; and operation of systems such as power, communication, and thermal control. In addition, the following is a typical list of specific constraints which should be considered.

(1) Mass of the landed vehicle shall be 584 kilograms.

(2) The landed vehicle (center body structure and landing gear structure) shall be compatible with an 3.35 meter base diameter, 120° blunted cone entry vehicle.

(3) Touchdown shall occur at a vertical velocity of 6.1 meters per second (relative to the gravity vector), and a maximum horizontal velocity of 3.7 meters per second.

(4) The total mass of the scientific payload shall be a minimum of 175 kilograms and the payload packing density shall not exceed a maximum of 943 dynes/cm^3.

(5) The landing vehicle shall not be restricted in orientation about the roll axis.

(6) Pitch and yaw attitudes at touchdown may vary as much as $\pm 10°$ from a plane normal to the gravity vector.

(7) The landing vehicle shall have as a goal the capability of successfully landing on slopes of 30° to the local horizontal.

(8) The landing system shall have the capability of performing satisfactorily when landing on surfaces containing particles varying in size from sand to 12.7 centimeter diameter rocks.

(9) The atmospheric pressure at the surface shall be assumed to be nine mb.

(10) The drag force on the footpad shall vary with penetration and applied normal forces.

(11) The landing surface shall be assumed to have an average crushing stress of 41×10^8 dynes/m^2 for penetrations to depths of 15.2 centimeters, a constant density of 1414 dynes/cm^3 for penetrations to depths of 15.2 centimeters and an angle of internal friction of 39 degrees.

(12) The coefficient of sliding friction between the surface and the footpad shall be assumed to be 0.3.

(13) Payload deceleration at any point in the payload shall be limited to a maximum of 20 earth g-units and landing deceleration of the footpads shall be limited to a maximum of 250 earth g-units.

(14) Methods shall be available for accomplishing post landing payload exposure to permit operation of experiments such as bioscience and imagery; measurements of wind velocity and direction, ambient pressure, temperature, and humidity; determination of soil composition; and operation of systems such as power, communication, and thermal control.

(15) Post landing orientation required by certain individual equipment items will be accomplished by aligning the item's positive X axis within ±5 degrees to an axis perpendicular to the local surface slope.

(16) Materials considered for use in the structures shall be compatible with space environment and a maximum temperature of 500°K and shall have minimum outgassing characteristics when exposed to a vacuum. Organic materials shall not be used in areas which may be subject to abrasion with the landing surface or to fragmentation with subsequent scattering of fragments on the landing surface.

(17) Surface gravitational acceleration shall be assumed to be 375 cm/sec^2.

3.2 Factors of Safety - The following factors shall be applied to the maximum loads (limit loads) encountered in planetary landing within the constraints specified above.

Energy Absorbing Material for Landing Gear Struts	1.00
All Other Structures	1.25

The load obtained by multiplying limit load by the appropriate factor of safety is the ultimate load used in sizing the structure.

The landing gear system shall have capability for stroke greater than that required for landings within constraints defined herein. This additional stroke shall be available to provide clearance between the bottom of the center body and a rock lying on the surface.

THIS PAGE INTENTIONALLY LEFT BLANK

4. STRUCTURAL ANALYSIS PROGRAM

The Structural Analysis Program contains a Center Body Option and a Landing Gear Option. The Center Body Option determines internal load distributions in the center body structure, and generates modal data for use in the Landing Loads and Motions Program. Energy absorption capability of a single landing gear undergoing large displacement stroking motion is investigated with the Landing Gear Option.

In the Center Body Option, internal and external load distributions and deflection patterns are determined for any system of forces or deflections impressed on an elastic network of structural bar members and rectangular shear webs. Problems for which the support structure behaves plastically can be solved. In addition, mode shapes and natural frequencies can be obtained for a center body with free-free support.

The Landing Gear Option of the program is used to analyze the large displacement stroking behavior of a single inverted tripod or cantilever landing gear. Energy absorption characteristics of the gear and internal strut loads are obtained for any number of applied footpad displacement steps. The footpad can be required to stroke a given distance along any arbitrarily oriented straight line, or a given distance normal to an arbitrarily oriented frictional plane. In the latter case, the coefficient of friction between the footpad and the frictional plane determines the sliding path of the footpad in the plane.

Analytical methods employed in the linear small displacement and nonlinear large displacement finite element portions of the Structural Analysis Program are presented in Section 4.1. A discussion of the normal mode method for obtaining modal data is also included in this section. Organization of the program is described in Section 4.2 and information necessary to operate the program is presented in Section 4.3. A listing of the program is contained in Appendix H.

4.1 Analytical Methods - The landing system and payload structures are highly redundant space frames consisting of a network of members possessing extensional, flexural, and torsional stiffness. Several methods for solving complex structural problems are in use today that effectively utilize the computer. Two methods considered for the Structural Analysis Program were the finite difference method and the finite element method. The finite element method was selected because of simplicity in dealing with nonhomogeneous, anisotropic structural applications. In addition, the elements can be changed easily in shape and size to follow complex boundary conditions or to allow for regions of rapid changes in stress or deflection.

A fundamental part of the finite element method is the technique used to obtain displacements at junctions of elements. Two approaches considered for the Structural Analysis Program were the force (flexibility) method and the stiffness (displacement) method. The stiffness method was selected primarily because the stiffness matrix developed may be used directly to generate the centerbody normal modes required to include centerbody flexibility in the Landing Loads and Motions Program. Also, the stiffness method lends itself readily to the large displacement technique necessary for landing gear analysis.

4.1.1 Center Body Option - The Center Body Option of the Structural Analysis Program can be used to analyze a structure idealized with an elastic network of structural bar elements and rectangular shear webs. The program internally replaces a rectangular shear web by two diagonal bar elements whose stiffness characteristics are equivalent to the shear web. Hence, the basic element of the program is the bar element.

4.1.1.1 Coordinate System - Two coordinate systems employed in the Center Body Option are the Local Coordinate System and the Global Coordinate System. Each bar member has its set of local coordinates as shown in Figure 4-1. Displacement notation of a general bar element capable of carrying axial load, shear in two directions, bending in two directions, and torsion is also indicated in Figure 4-1 for the Local Coordinate System. Subscript 1 refers to a displacement due to axial load, subscripts 2 and 3 are for displacement due to shear loads, subscript 4 is for a rotation due to torque, and subscripts 5 and 6 are for rotations due to moments. The Local Coordinate System origin for each bar is located at joint "p," with the X_ℓ axis aligned along the member axis. Positive X_ℓ is on the side of joint "p" towards joint "q." The Y_ℓ axis is perpendicular to X_ℓ and is located in the "pqr" plane. Positive Y_ℓ is on the side of X_ℓ towards point "r." The Z_ℓ

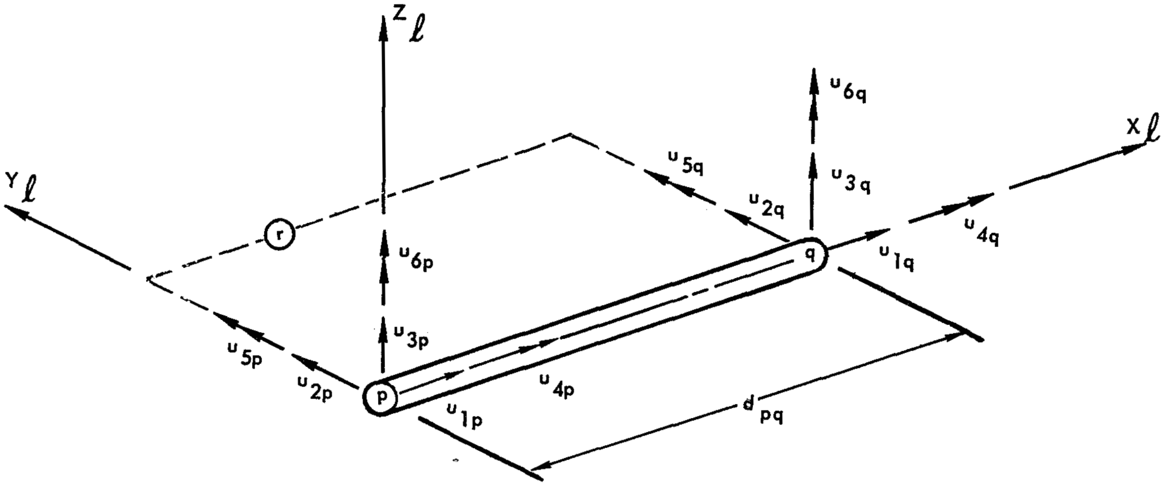


FIGURE 4–1 DISPLACEMENTS OF GENERAL BAR ELEMENT IN LOCAL COORDINATE SYSTEM

axis is then established using the right hand rule. Local Coordinate Systems are established by identifying each bar's origin (joint "p"), end (joint "q"), and orientation of the bending axis Y_ℓ (defined by joint "r"). Moment of inertia I_T about the Y_ℓ axis, moment of inertia I_N about the Z_ℓ axis, cross-sectional area A, torsional constant J, modulus of elasticity E, and shear modulus G are specified for each bar relative to its Local Coordinate System.

A common coordinate system (global) for all structural elements must be established so that element forces and displacements may be related to a common frame of reference. This Global Coordinate System may be any convenient orthogonal right hand system. Displacement notation of the bar element in the Global Coordinate System is shown in Figure 4-2. Joint locations, external load distributions, and joint deflections are all specified in the Global Coordinate System.

4.1.1.2 Stiffness Matrices - A stiffness matrix for each bar element is generated in its Local Coordinate System based on small deflection theory. This is done by applying a unit displacement or rotation to one end of the bar (while restraining all other rotations and displacements) and determining the induced forces and moments. Displacements and rotations are applied sequentially until all degrees of freedom at each end of the bar element have been included. Displacement notation of a general bar element capable of carrying axial load, shear in two directions, bending in two directions, and torsion was shown in Figure 4-1.

Force-displacement relationships for the three unit displacements and three unit rotations possible at each end of the bar element are shown in Figure 4-3. Forces and displacements in the Local Coordinate System are related in matrix form by

$$\{F\} = [K]\,\{\delta_L\} \tag{4-1}$$

In Equation (4-1), [K] is the matrix of stiffness coefficients; {F} is a column matrix of applied forces at the joints, and $\{\delta_L\}$ is the column matrix of joint displacements. Application of Maxwell's Reciprocal Law allows formulation of a symmetric stiffness matrix when an orthogonal coordinate system is employed. The element stiffness matrix for a general bar is given in

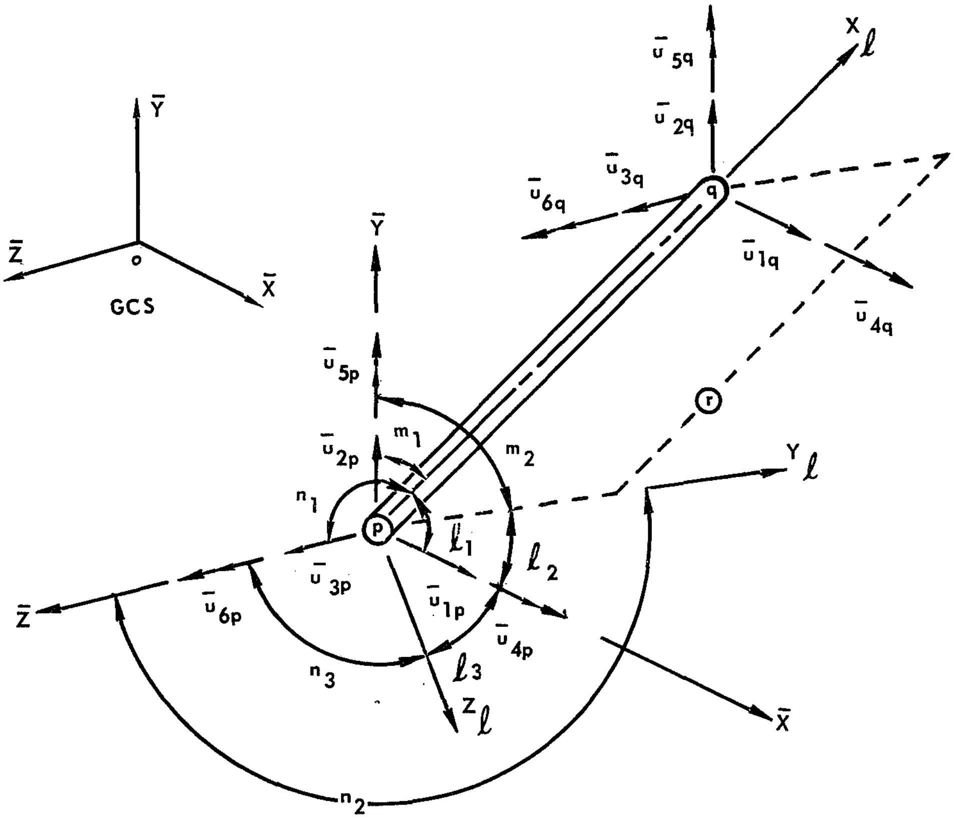


FIGURE 4–2 DISPLACEMENTS OF GENERAL BAR ELEMENT IN GLOBAL COORDINATE SYSTEM

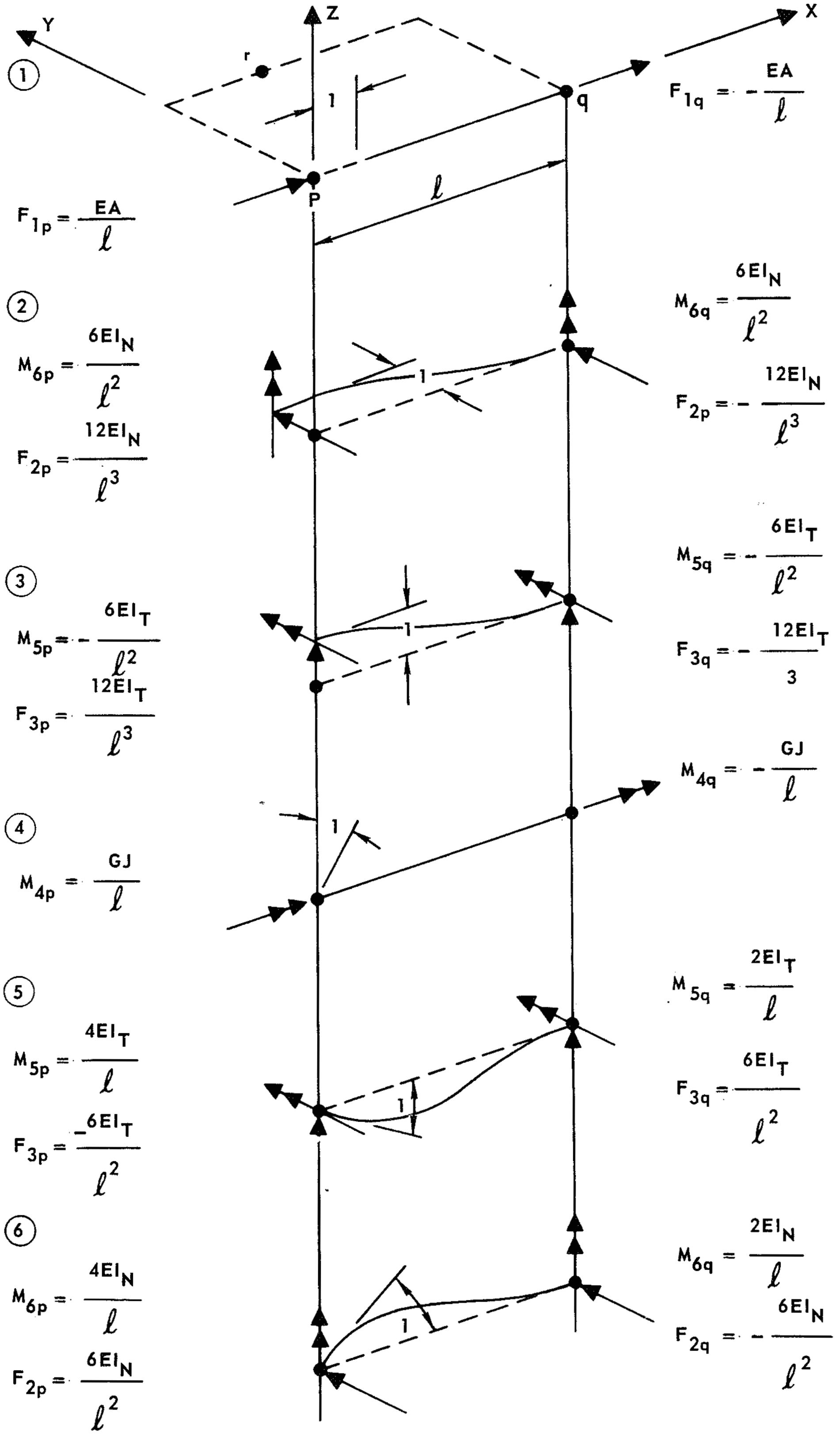


FIGURE 4–3 FORCE – DISPLACEMENT RELATIONS FOR GENERAL BAR ELEMENT

Figure 4-4. Terms in this matrix are the stiffness coefficients given in Figure 4-3.

If it is desired to include the effects of shearing strain on elemental beam deflection, the program applies the stiffness factors given in Figure 4-5 to the corresponding terms in Figure 4-4. Input shear form factors K_N and K_T are determined by dividing the total cross-sectional area of the bar by the area effective in carrying shear loads. The area used in determining K_N should be the area effective in carrying shear in the local Y_ℓ direction (bending about the Z_ℓ axis), while K_T is determined based on effective area for shear in the Z_ℓ direction. An example calculation of shear form factors K_N and K_T is given in Figure 4-6.

In addition, a rectangular shear web element is incorporated in the Center Body Option. The shear web idealization used by the program is indicated in Figure 4-7. Here the shear web is replaced by two diagonal bar elements whose stiffness characteristics are equivalent to the shear web. Cross-sectional properties of diagonals are derived by requiring equivalent strain energies in shear web and diagonals for statically equivalent internal loads.

Both ends of a member may be pinned (allowed to freely rotate) in a given direction by setting the appropriate moment of inertia equal to zero. If it is desired to pin one end of a member, while the other end remains fixed, the program utilizes the terms shown in Figures 4-8 through 4-11 in place of the corresponding terms in Figure 4-5. Terms used depend on the joint (p or q) pinned and the bending axis (Y_ℓ or Z_ℓ) about which the end is allowed to rotate. For example, if it is desired to pin the q end of a member with respect to bending about the Y_ℓ axis, terms in Figure 4-11 would be used in place of the corresponding terms in Figure 4-5.

To determine the stiffness matrix for the completely assembled structure, all element forces and displacements are related to the Global Coordinate System. Displacement notation of the bar element in Global Coordinate System is defined in Figure 4-2.

Transformation matrices are needed to change the frame of reference of each element from the Local to the Global Coordinate System. This transformation is expressed by the linear matrix equation

$$\{\delta_L\} = [\lambda]\{\delta_G\} \qquad (4\text{-}2)$$

$$K = \begin{array}{c|cccccccccccc|}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1p} & \frac{EA}{\ell} & & & -\frac{EA}{\ell} & & & & & & & & \\
F_{2p} & & \frac{12EI_N}{\ell^3} & & & -\frac{12EI_N}{\ell^3} & & & & \frac{6EI_N}{\ell^2} & & & \frac{6EI_N}{\ell^2} \\
F_{3p} & & & \frac{12EI_T}{\ell^3} & & & -\frac{12EI_T}{\ell^3} & & -\frac{6EI_T}{\ell^2} & & & -\frac{6EI_T}{\ell^2} & \\
F_{1q} & -\frac{EA}{\ell} & & & \frac{EA}{\ell} & & & & & & & & \\
F_{2q} & & -\frac{12EI_N}{\ell^3} & & & \frac{12EI_N}{\ell^3} & & & & -\frac{6EI_N}{\ell^2} & & & -\frac{6EI_N}{\ell^2} \\
F_{3q} & & & -\frac{12EI_T}{\ell^3} & & & \frac{12EI_T}{\ell^3} & & \frac{6EI_T}{\ell^2} & & & \frac{6EI_T}{\ell^2} & \\
M_{4p} & & & & & & & \frac{GJ}{\ell} & & & -\frac{GJ}{\ell} & & \\
M_{5p} & & & -\frac{6EI_T}{\ell^2} & & & \frac{6EI_T}{\ell^2} & & \frac{4EI_T}{\ell} & & & \frac{2EI_T}{\ell} & \\
M_{6p} & & \frac{6EI_N}{\ell^2} & & & -\frac{6EI_N}{\ell^2} & & & & \frac{4EI_N}{\ell} & & & \frac{2EI_N}{\ell} \\
M_{4q} & & & & & & & -\frac{GJ}{\ell} & & & \frac{GJ}{\ell} & & \\
M_{5q} & & & -\frac{6EI_T}{\ell^2} & & & \frac{6EI_T}{\ell^2} & & \frac{2EI_T}{\ell} & & & \frac{4EI_T}{\ell} & \\
M_{6q} & & \frac{6EI_N}{\ell^2} & & & -\frac{6EI_N}{\ell^2} & & & & \frac{2EI_N}{\ell} & & & \frac{4EI_N}{\ell}
\end{array}$$

FIGURE 4–4 STIFFNESS MATRIX FOR GENERAL BAR ELEMENT

$$
K_S = \begin{array}{c|cccccccccccc|}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1q} & 1 & & & 1 & & & & & & & & \\
F_{2p} & & C_{1N} & & & C_{1N} & & & & C_{1N} & & & C_{1N} \\
F_{3p} & & & C_{1T} & & & C_{1T} & & C_{1T} & & & C_{1T} & \\
F_{1q} & 1 & & & 1 & & & & & & & & \\
F_{2q} & & C_{1N} & & & C_{1N} & & & & C_{1N} & & & C_{1N} \\
F_{3q} & & & C_{1T} & & & C_{1T} & & C_{1T} & & & C_{1T} & \\
M_{4p} & & & & & & & 1 & & & 1 & & \\
M_{5p} & & & C_{1T} & & & C_{1T} & & C_{2T} & & & C_{3T} & \\
M_{6p} & & C_{1N} & & & C_{1N} & & & & C_{2N} & & & C_{3N} \\
M_{4q} & & & & & & & 1 & & & 1 & & \\
M_{5q} & & & C_{1T} & & & C_{1T} & & C_{3T} & & & C_{2T} & \\
M_{6q} & & C_{1N} & & & C_{1N} & & & & C_{3N} & & & C_{2N} \\
\end{array}
$$

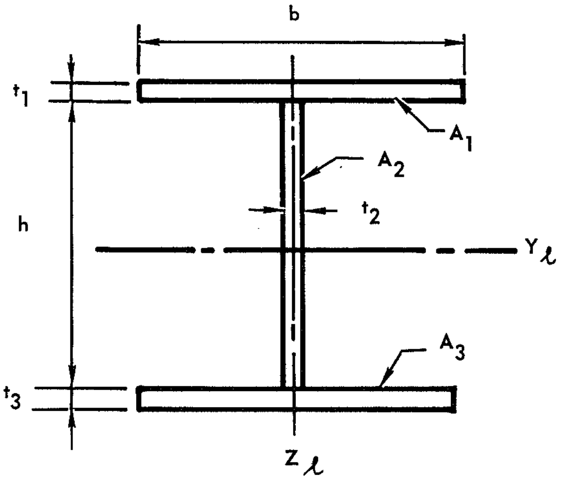
$$C_{1N} = \frac{1}{1 + \alpha_N} \qquad C_{1T} = \frac{1}{1 + \alpha_T}$$

$$C_{2N} = 1 - \frac{3}{4(1 + 1/\alpha_N)} \qquad C_{2T} = 1 - \frac{3}{4(1 + 1/\alpha_T)}$$

$$C_{3N} = 1 - \frac{3}{2\,(1 + 1/\alpha_N)} \qquad C_{3T} = 1 - \frac{3}{2(1 + 1/\alpha_T)}$$

$$\alpha_N = \frac{12EI_N/\ell^3}{GA\,/\ell} \cdot K_N \qquad \alpha_T = \frac{12EI_T/\ell^3}{GA\,/\ell} \cdot K_T$$

FIGURE 4–5 STIFFNESS FACTORS APPLIED TO ORIGINAL MATRIX TO ACCOUNT FOR ELEMENTAL BAR SHEAR STRAIN

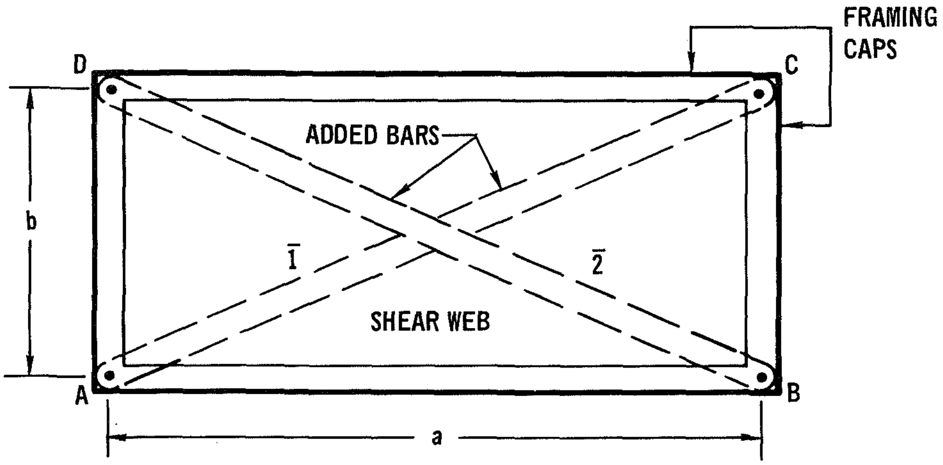


$$A_1 = bt_1$$
$$A_2 = ht_2$$
$$A_3 = bt_3$$
$$A_{TOTAL} = A_1 + A_2 + A_3$$

$$K_N = \frac{A_{TOTAL}}{A_1 + A_3}$$

$$K_T = \frac{A_{TOTAL}}{A_2}$$

FIGURE 4–6 EXAMPLE CALCULATION OF SHEAR FORM FACTOR



GIVEN SHEAR WEB:

MODULUS OF ELASTICITY E, POISSON'S RATIO ν, AND THICKNESS t.
CORNER JOINTS A, B, C, D NUMBERED COUNTERCLOCKWISE.

DIAGONAL REPLACEMENT OF WEB BY PROGRAM:

ADD 2 BARS AS SHOWN. BARS ARE PINNED AT BOTH ENDS AND NOT ATTACHED TO EACH OTHER.
BAR PROPERTIES ARE AS FOLLOWS:

AREA = $t. (a^2 + b^2)^{3/2}/4ab\ (1 + \nu)$.
$E = E_{web}$
OTHER PROPERTIES = 0
WHERE a = LENGTH AB
b = LENGTH AD

INTERPRETATION OF PROGRAM OUTPUT:

SHEAR FLOW IN WEB; $q = (P_{\bar{2}} - P_{\bar{1}})/\sqrt{a^2 + b^2}$
WHERE $P_{\bar{2}}$ = AXIAL LOAD IN BAR $\bar{2}$.
$P_{\bar{1}}$ = AXIAL LOAD IN BAR $\bar{1}$.
SIGN CONVENTION; + q CORRESPONDS TO SHEAR FLOW ON SIDE AB FROM A TO B.
+ P CORRESPONDS TO TENSION IN BAR.

FIGURE 4-7 SHEAR WEB REPLACEMENT WITH DIAGONALS

$$K_{6p} = \begin{array}{c|cccccccccccc|}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1p} & & & & & & & & & & & & \\
F_{2p} & & C_{6p} & & & C_{6p} & & & & 0 & & & 2C_{6p} \\
F_{3p} & & & & & & & & & & & & \\
F_{1q} & & & & & & & & & & & & \\
F_{2q} & & C_{6p} & & & C_{6p} & & & & 0 & & & 2C_{6p} \\
F_{3q} & & & & & & & & & & & & \\
M_{4p} & & & & & & & & & & & & \\
M_{5p} & & & & & & & & & & & & \\
M_{6p} & & 0 & & & 0 & & & & 0 & & & 0 \\
M_{4q} & & & & & & & & & & & & \\
M_{5q} & & & & & & & & & & & & \\
M_{6q} & & 2C_{6p} & & & 2C_{6p} & & & & 0 & & & 3C_{6p} \\
\hline
\end{array}$$

$$C_{6p} = \frac{1}{4 + \alpha_N}$$

$$\alpha_N = \frac{12\,E\,I_N/\ell^3}{GA/\ell}\,K_N$$

FIGURE 4–8 SUBSTITUTE FACTORS IN K_S MATRIX TO ALLOW ELIMINATION OF ROTATIONAL RESTRAINT IN Z DIRECTION AT POINT P

$$K_{6q} = \begin{array}{c|cccccccccccc}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1p} & & & & & & & & & & & & \\
F_{2p} & & C_{6q} & & & C_{6q} & & & & 2C_{6q} & & & 0 \\
F_{3p} & & & & & & & & & & & & \\
F_{1q} & & & & & & & & & & & & \\
F_{2q} & & C_{6q} & & & C_{6q} & & & & 2C_{6q} & & & 0 \\
F_{3q} & & & & & & & & & & & & \\
M_{4p} & & & & & & & & & & & & \\
M_{5p} & & & & & & & & & & & & \\
M_{6p} & & 2C_{6q} & & & 2C_{6q} & & & & 3C_{6q} & & & 0 \\
M_{4q} & & & & & & & & & & & & \\
M_{5q} & & & & & & & & & & & & \\
M_{6q} & & 0 & & & 0 & & & & 0 & & & 0
\end{array}$$

$$C_{6q} = \frac{1}{4 + \alpha_N}$$

$$\alpha_N = \frac{12\,E\,I_N/\ell^3}{GA/\ell}\,K_N$$

FIGURE 4–9 SUBSTITUTE FACTORS IN K_S MATRIX TO ALLOW ELIMINATION OF ROTATIONAL RESTRAINT IN Z DIRECTION AT POINT Q

$$
K_{5_p} = \begin{array}{c|cccccccccccc|}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1p} & & & & & & & & & & & & \\
F_{2p} & & & & & & & & & & & & \\
F_{3p} & & & C_{5p} & & & C_{5p} & & 0 & & & 2C_{5p} & \\
F_{1q} & & & & & & & & & & & & \\
F_{2q} & & & & & & & & & & & & \\
F_{3q} & & & C_{5p} & & & C_{5p} & & 0 & & & 2C_{5p} & \\
M_{4p} & & & & & & & & & & & & \\
M_{5p} & & & 0 & & & 0 & & 0 & & & 0 & \\
M_{6p} & & & & & & & & & & & & \\
M_{4q} & & & & & & & & & & & & \\
M_{5q} & & & 2C_{5p} & & & 2C_{5p} & & 0 & & & 3C_{5p} & \\
M_{6q} & & & & & & & & & & & & \\
\hline
\end{array}
$$

$$C_{5p} = \frac{1}{4 + \alpha_T}$$

$$\alpha_T = \frac{12 E I_T / \ell^3}{GA/\ell} K_T$$

FIGURE 4–10 SUBSTITUTE FACTORS IN K_S MATRIX ALLOWING ELIMINATION OF ROTATIONAL RESTRAINT IN Y DIRECTION AT POINT P

$$K_{5q} = \left[\begin{array}{c|cccccccccccc}
 & u_{1p} & u_{2p} & u_{3p} & u_{1q} & u_{2q} & u_{3q} & u_{4p} & u_{5p} & u_{6p} & u_{4q} & u_{5q} & u_{6q} \\
\hline
F_{1p} & & & & & & & & & & & & \\
F_{2p} & & & & & & & & & & & & \\
F_{3p} & & & C_{5q} & & & C_{5q} & & 2C_{5q} & & & 0 & \\
F_{1q} & & & & & & & & & & & & \\
F_{2q} & & & & & & & & & & & & \\
F_{3q} & & & C_{5q} & & & C_{5q} & & 2C_{5q} & & & 0 & \\
M_{4p} & & & & & & & & & & & & \\
M_{5p} & & & 2C_{5q} & & & 2C_{5q} & & 3C_{5q} & & & 0 & \\
M_{6p} & & & & & & & & & & & & \\
M_{4q} & & & & & & & & & & & & \\
M_{5q} & & & 0 & & & 0 & & 0 & & & 0 & \\
M_{6q} & & & & & & & & & & & &
\end{array}\right]$$

$$C_{5q} = \frac{1}{4 + \alpha_T}$$

$$\alpha_T = \frac{12E\, I_T / \ell^3}{GA/\ell}\; K_T$$

FIGURE 4–11 SUBSTITUTE FACTORS IN K_S MATRIX ALLOWING ELIMINATION OF ROTATIONAL RESTRAINT IN Y DIRECTION AT POINT Q where $[\lambda]$ is a matrix of direction cosines (cosines of angles between Local and Global Coordinate Systems) obtained by resolving global displacements in the direction of local coordinates. For the general bar, $[\lambda]$ is given in Equation (4-3).

$$\lambda = \begin{bmatrix} \bar{\lambda} & 0 & 0 & 0 \\ 0 & \bar{\lambda} & 0 & 0 \\ 0 & 0 & \bar{\lambda} & 0 \\ 0 & 0 & 0 & \bar{\lambda} \end{bmatrix} \qquad (4\text{-}3)$$

In Equation (4-3), each $\bar{\lambda}$ is a 3 by 3 matrix of the direction cosines of the local coordinate axes relative to the global system, as shown in Equation (4-4).

$$\bar{\lambda} = \begin{bmatrix} \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \\ \ell_3 & m_3 & n_3 \end{bmatrix} \qquad (4\text{-}4)$$

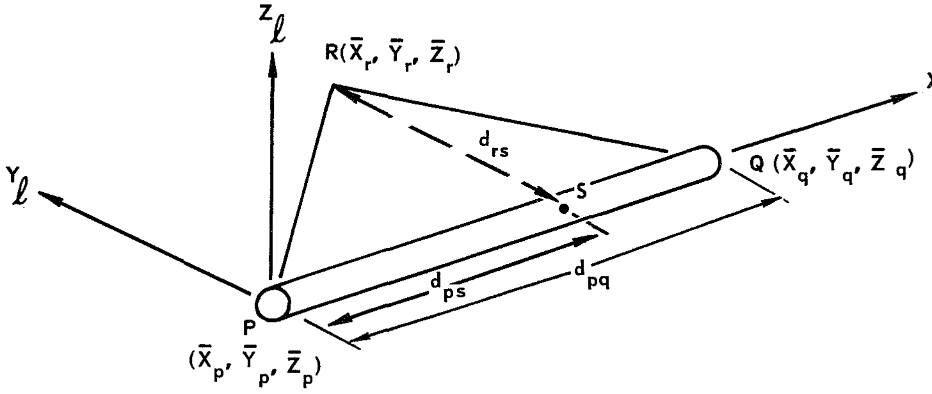
The row format of λ corresponds to the sequence of displacements in the Local Coordinate System specified for the stiffness matrix in Figure 4-4. The column format of λ corresponds to a similar sequence in the Global Coordinate System (i.e., $\bar{u}_{1_p}$, $\bar{u}_{2_p}$, $\ldots\bar{u}_{6_p}$). Values of the direction cosines relating the Local Coordinate System to the Global Coordinate System are determined as indicated in Figure 4-12.

Each element stiffness matrix is transformed from the Local to the Global Coordinate System by Equation (4-5)

$$[\bar{K}] = [\lambda]^T[K][\lambda] \qquad (4\text{-}5)$$

where $[\bar{K}]$ is the element stiffness matrix transformed to Global Coordinate System and $[\lambda]^T$ is the transpose of transformation matrix $[\lambda]$.

The total stiffness matrix (K_A) for the assembled structure (in the Global Coordinate System) as shown in Figure 4-13, is generated by systematically adding the transformed element stiffness matrices. Nodal points on



$$d_{pq} = \sqrt{(\bar{X}_q - \bar{X}_p)^2 + (\bar{Y}_q - \bar{Y}_p)^2 + (\bar{Z}_q - \bar{Z}_p)^2}$$

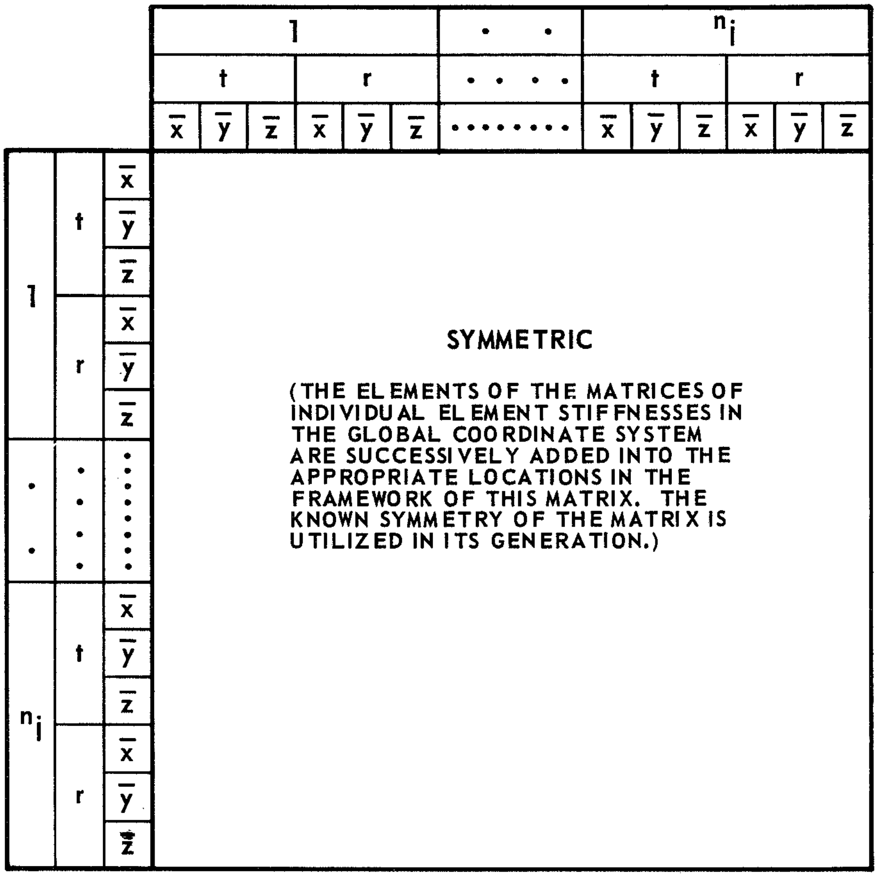
$$d_{rs} = \sqrt{(\bar{X}_r - \bar{X}_p)^2 + (\bar{Y}_r - \bar{Y}_p)^2 + \bar{Z}_r - \bar{Z}_p)^2 - d_{ps}^2}$$

$$d_{ps} = \ell_1 (\bar{X}_r - \bar{X}_p) + m_1 (\bar{Y}_r - \bar{Y}_p) + n_1 (\bar{Z}_r - \bar{Z}_p)$$

DIRECTION COSINE	COSINE OF ANGLE BETWEEN		EQUATION FOR DIRECTION COSINE
	LOCAL AXIS	GLOBAL AXIS	
ℓ_1	X_ℓ	$\bar{X}$	$(\bar{X}_q - \bar{X}_p)/d_{pq}$
m_1	X_ℓ	$\bar{Y}$	$(\bar{Y}_q - \bar{Y}_p)\ d_{pq}$
n_1	X_ℓ	$\bar{Z}$	$(\bar{Z}_q - \bar{Z}_p)\ d_{pq}$
ℓ_2	Y_ℓ	$\bar{X}$	$[(\bar{X}_r - \bar{X}_p) - \ell_1 d_{ps}]/d_{rs}$
m_2	Y_ℓ	$\bar{Y}$	$[(\bar{Y}_r - \bar{Y}_p) - m_1 d_{ps}]/d_{rs}$
n_2	Y_ℓ	$\bar{Z}$	$[(\bar{Z}_r - \bar{Z}_p) - n_1 d_{ps}]/d_{rs}$
ℓ_3	Z_ℓ	$\bar{X}$	$m_1 n_2 - m_2 n_1$
m_3	Z_ℓ	$\bar{Y}$	$-\ell_1 n_2 + \ell_2 n_1$
n_3	Z_ℓ	$\bar{Z}$	$\ell_1 m_2 - \ell_2 m_1$

NOTE: POINT r IS AN ARBITRARILY SELECTED POINT LYING IN THE $X\ell - Y\ell$ PLANE. IT IS USED TO IDENTIFY THE ORIENTATION OF MEMBER BENDING AXES $Z\ell$ AND $Y\ell$ RELATIVE TO THE GLOBAL COORDINATE SYSTEM. (I_N IS THE MOMENT OF INERTIA ABOUT THE $Z\ell$ AXIS AND I_T IS THE MOMENT OF INERTIA ABOUT THE $Y\ell$ AXIS.)

FIGURE 4-12 DIRECTION COSINES OF LOCAL COORDINATE SYSTEM RELATIVE TO GLOBAL COORDINATE SYSTEM



SYMMETRIC

(THE ELEMENTS OF THE MATRICES OF INDIVIDUAL ELEMENT STIFFNESSES IN THE GLOBAL COORDINATE SYSTEM ARE SUCCESSIVELY ADDED INTO THE APPROPRIATE LOCATIONS IN THE FRAMEWORK OF THIS MATRIX. THE KNOWN SYMMETRY OF THE MATRIX IS UTILIZED IN ITS GENERATION.)

SYMBOLS t AND r INDICATE TRANSLATIONAL AND ROTATIONAL DISPLACEMENTS RESPECTIVELY.

FIGURE 4–13 ASSEMBLED STIFFNESS MATRIX FORMAT the idealized structure are numbered consecutively from 1 to n_J. The stiffness matrix K_A is assembled with a row and column format corresponding to the three translational followed by the three rotational degrees of freedom in the global system at each node in sequence. In this general case, the size of the stiffness matrix is $6n_J \times 6n_J$.

4.1.1.3 Elastic Analysis - The assembled stiffness matrix $[K_A]$ is related to the column matrices of global forces and displacements at each node by Equation (4-6) which represents a combination of Equation (4-1) through (4-5). The word "forces" in this discussion implies both forces and moments, and the word "displacements" implies both deflections and rotations.

$$\{F\} = [K_A]\{\delta_G\} \tag{4-6}$$

The stiffness matrix is singular. That is, its determinant vanishes and its inverse does not exist. Boundary conditions (supports) must be defined to prevent rigid body motion. Once $[K_A]$ has been determined, a solution can be obtained for any set of support conditions.

As described in Reference (3), the unknown nodal displacements and support reactions are normally obtained by partitioning Equation (4-6) according to the location and orientation of supports as indicated in Equation (4-7).

$$\left\{\begin{array}{c} F_{(m-n)} \\ \hdashline F_n \end{array}\right\} = \left[\begin{array}{c:c} A_{(m-n) \times (m-n)} & B_{(m-n) \times n} \\ \hdashline B'_{n \times (m-n)} & D_{n \times n} \end{array}\right] \left\{\begin{array}{c} \delta_{(m-n)} \\ \hdashline \delta_n \end{array}\right\} \tag{4-7}$$

Subscripts n and m are

n = number of support boundary conditions (i.e., $\delta_n = 0$),

m = order of stiffness matrix.

This partitioning results in two sets of equations: Equation (4-8) relates unknown nodal displacements to known applied forces, and Equation (4-9) relates unknown support reactions to unknown nodal displacements.

$$\{F_{(m-n)}\} = [A]\{\delta_{(m-n)}\} \tag{4-8}$$

$$\{F_n\} = [B'] \{\delta_{(m-n)}\} \tag{4-9}$$

The inverse of matrix [A] in Equation (4-8) is the flexibility matrix of the structure. Equation (4-8) may be rewritten to give unknown nodal displacements in terms of the flexibility matrix and known applied forces as shown in Equation (4-10).

$$\{\delta_{(m-n)}\} = [A]^{-1} \{F_{(m-n)}\} \tag{4-10}$$

The unknown reactions may be obtained by combining Equations (4-9) and (4-10) as shown in Equation (4-11).

$$\{F_n\} = [B'] [A]^{-1} \{F_{m-n}\} \tag{4-11}$$

The preceding discussion outlines the method normally used to obtain nodal displacements and reactions. Due to computer core limitations and the desire to minimize program running time, an iterative method of determining nodal deflections and reactions, once the stiffness matrix was established, was programmed. An iterative method eliminates the need for matrix inversion.

The requirement for plastic supports calls for multiple solutions of sets of equations very nearly the same. These sets of simultaneous equations are represented by matrix Equation (4-8), relating unknown nodal deflections to known applied forces. This type of problem is best handled by an iterative solution, since the required nodal deflections are approximately known after the initial elastic solution is determined, thus minimizing computing time. Four iterative techniques were programmed; Gauss-Siedel, Gauss-Siedel with Aitkens Delta Squared improvements, the Overrelaxation method, and the Conjugate Gradient technique. Reference (4) describes the first three methods and Reference (5) the latter. Any one of these techniques may be best suited for a specific structural problem. However, experience dictates that for highly redundant space frames, such as the legged lander, the Conjugate Gradient technique appears to be the best suited of the above solution methods.

After nodal displacements have been determined using the iteration procedure, the unknown support reactions are obtained by substitution of these nodal displacements into matrix Equation (4-9). Forces on elements in the Local Coordinate System are found by transforming the nodal displacements into the Local Coordinate System using Equation (4-2) and applying the appropriate force-displacement relationships using Equation (4-1).

The program also is capable of solving problems wherein the nodal displacements are known, as in problems where some supports settle. Combinations of known applied forces and known nodal displacements may be input, and all forces and displacements will be determined. Boundary conditions imposed on the problem must be sufficient to prevent rigid body motion. Partitioning of Equation (4-7) for this case results in Equation (4-12), relating unknown nodal displacements to known applied forces and known nodal displacements, and in Equation (4-13), which relates unknown forces and reactions to the now known (determined in Equation (4-12)) nodal displacements. Subscript n for this case implies either zero or nonzero known boundary conditions.

$$\{F_{(m-n)}\} = [A]\{\delta_{(m-n)}\} + [B]\{\delta_n\} \tag{4-12}$$

$$\{F_n\} = [B']\{\delta_{(m-n)}\} + [D]\{\delta_n\} \tag{4-13}$$

This solution is simply a more general case of the initial partitioning (Equation (4-8) and (4-9)), where the only known nodal displacements were zero and terms involving δ_n were therefore not included.

4.1.1.4 Plastic Support - When analyzing a space frame it may be desirable to idealize support members with restricted load-carrying capability. An upper (positive) and lower (negative) limit can be set on the magnitude of any force or moment component in such a support member. Initial solution of the space frame redundancy assumes elastic deformation, but the magnitude of the load components at the support are compared with input limits to ascertain if these have been exceeded. Those load components (reactions) which exceed the input limits are then assumed to be at the input cutoff values. The new set of boundary conditions (new column matrix of deflections and forces) are then employed with the stiffness matrix to obtain a new solution. The load components at the supports and the limits are again compared. If the limits are exceeded, the process is repeated until no limits are exceeded and equilibrium is achieved.

A plastic attenuator support can be modeled with this feature by setting the appropriate input limit to the force causing plastic deformation. Cables with one end attached to a support and the other to a structural joint can be idealized by making one of the limits on each reaction force component zero, allowing the cable to carry only tension and no compression. Determination of the proper limit (positive or negative) to use on a particular force component depends on whether the desired limit is in the positive or negative global coordinate direction. If a desired tension force in a cable results in a force component in the positive global direction at the support, the lower (negative) limit is set equal to zero. If the desired tension force component is in the negative global direction, the upper (positive) limit is set equal to zero.

4.1.1.5 Modal Analysis - The frequencies and mode shapes for the free-free center body structure are determined once the unrestrained stiffness matrix of the center body (Section 4.1.1.2) has been obtained. This modal analysis is performed in the optional Modal Analysis Routine in the Center Body Option of the Structural Analysis Program.

Free vibrations of the center body structure are defined by Equation (4-14)

$$[M]\{\ddot{q}(s, t)\} + [K]\{q(s, t)\} = 0 \tag{4-14}$$

where

$[M]$ = mass matrix of the center body.
$[K]$ = stiffness matrix of the center body.
$\{\ddot{q}\}, \{q\}$ = accelerations and displacements describing the motion of the control points throughout the center body.
s = space coordinates in the center body.
t = time.

The above representation of the stiffness and inertia characteristics of the center body are input data for the Modal Analysis Routine. An eigenvalue routine is used to obtain the vibratory free-free mode shapes and corresponding frequencies.

The large order center body stiffness matrix results in an eigenvalue problem which is too large to solve practically. The computer run time required to obtain the frequencies and mode shapes would be excessive. In addition, the eigenvalue problem would exceed the allotted computer core storage requirements. For these reasons, the size of the center body's structural stiffness matrix is reduced before solving the eigenvalue problem. This reduction technique is discussed in Reference (6).

In the reduction procedure, a number of degrees of freedom, corresponding to various displacements and rotations at the center body joints, are removed. To remove these, it is assumed that the inertia forces and/or moments associated with these degrees of freedom are negligible. In this procedure, all of the strain energy associated with the removed degrees of freedom is retained. This reduction procedure is outlined as follows.

$$\begin{Bmatrix} P \\ \hdashline 0 \end{Bmatrix} = \left[\begin{array}{c:c} K_{11} & K_{12} \\ \hdashline K_{21} & K_{22} \end{array}\right] \begin{Bmatrix} q_1 \\ \hdashline q_2 \end{Bmatrix} \tag{4-15}$$

where

K_{11}, K_{12}, K_{21}, K_{22} = segments of total center body stiffness matrix.

q_1 = degrees of freedom on which forces and moments (P's) exist.

q_2 = degrees of freedom on which negligible forces and moments exist.

In the above, the total stiffness matrix has been reordered such that the elements associated with the degrees of freedom to be retained appear first.

Equation (4-15) is equivalent to the two following expressions

$$\{P\} = [K_{11}]\{q_1\} + [K_{12}]\{q_2\} \tag{4-16}$$

$$\{0\} = [K_{21}]\{q_1\} + [K_{22}]\{q_2\} \tag{4-17}$$

Solving the second of these for $\{q_2\}$ gives

$$\{q_2\} = -[K_{22}]^{-1}[K_{21}]\{q_1\} \tag{4-18}$$

Substituting this into Equation (4-16) results in the reduced form

$$\{P\} = \Big[[K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}]\Big] \{q_1\} \tag{4-19}$$

from which the reduced stiffness matrix, [K*], is defined as

$$[K^*] = [K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}] \tag{4-20}$$

The eigenvalue problem associated with the reduced system is

$$[M^*]\{\ddot{q}_1\} + [K^*]\{q_1\} = 0 \tag{4-21}$$

where [M*] is a diagonal mass matrix whose elements represent the distribution of the center body mass at the degrees of freedom to be retained. Frequencies and mode shapes of the total elastic center body structure are obtained using Equations (4-18) and (4-21). Reference (7) summarizes the Householder-Ortega-Wilkinson Method used to determine mode shapes and natural frequencies. An example of the use of this routine is given in Section 4.3.1.3.

4.1.2 Landing Gear Option - The Landing Gear Option of the Structural Analysis Program can be used to investigate the energy absorption characteristics and internal loads in inverted tripod or cantilever landing gears. The finite element stiffness method employed in this option of the program is based on large displacement (nonlinear) finite element theory.

In linear finite element theory, changes in the stiffness matrix due to displacements are assumed to be negligible. Landing gear struts experience large rotational and extensional displacements during stroking of the gear. These displacements require that the finite element idealization of the struts include "rotational" and "extensional" nonlinearities. To illustrate the nonlinear nature of the problem, consider a planar gear subjected to the displacements shown in Figure 4-14. Stiffness of the gear changes appreciably during stroking from the original to the displaced position. This change is due to the following displacement nonlinearities:

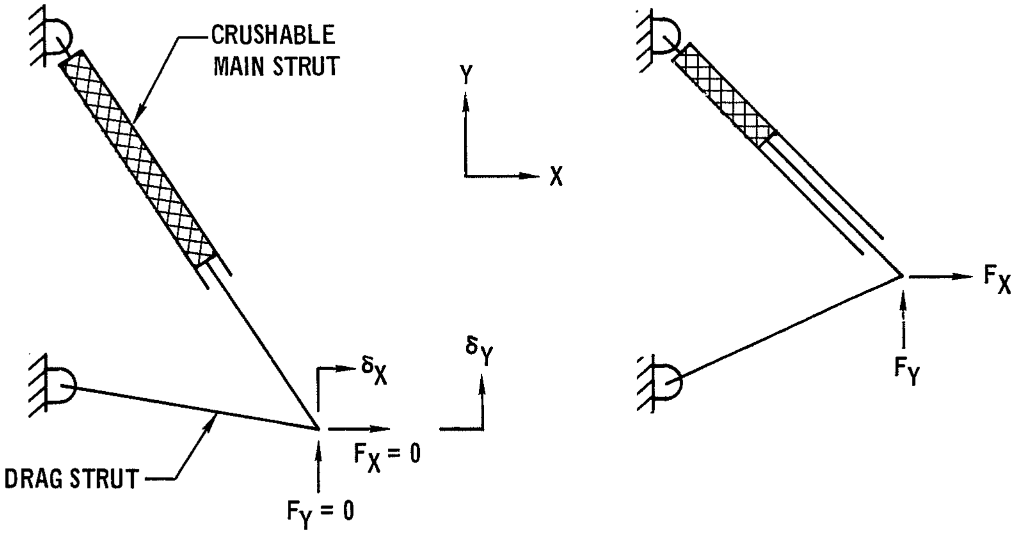
(1) rotational nonlinearities - large rotations cause significant changes in strut orientation and the resulting stiffness;

(2) extensional nonlinearities - compressive (or tensile) strut crushing changes the strut length and slope of the axial load-stroke curve (referred to as axial stiffness) both of which alter the stiffness.

For these reasons linear theory cannot be used in the analysis of landing gears since changes in the stiffness matrix due to displacements are not negligible.

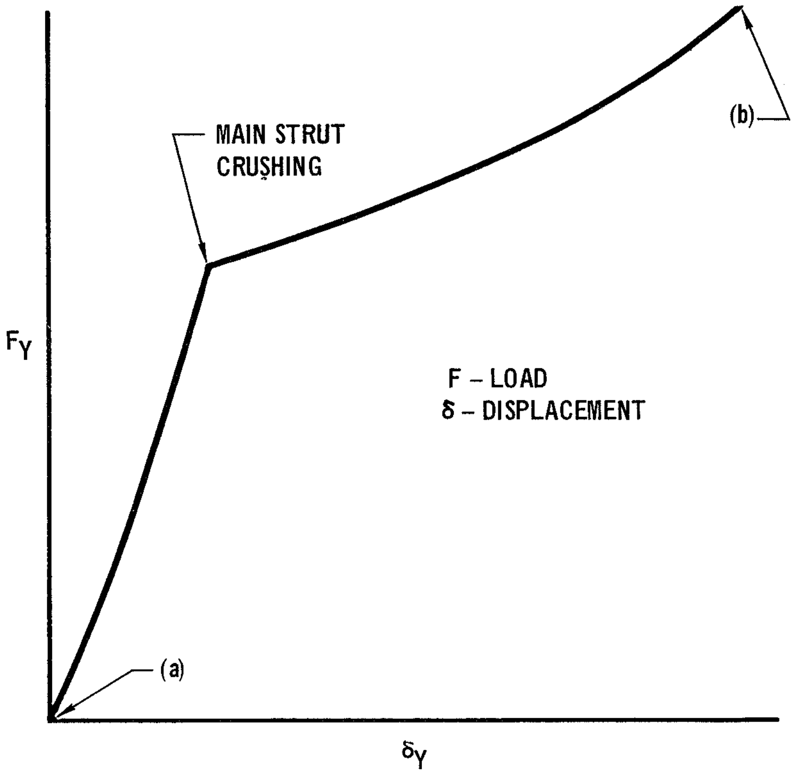
4.1.2.1 Modified Incremental Stiffness Method - To account for the nonlinearities associated with landing gear analysis, the incremental stiffness finite element method is employed. A modification of this method is made to insure that load unbalances do not result in the gear due to the linear approximations made in each step of the solution technique.

In the incremental stiffness method, nonlinear behavior is approximated through a sequence of linear solutions. The loading is divided into a number of incremental steps. For each step, an increment of the external load is applied and incremental displacements determined. These displacements, when added to the structural positions at the conclusion of the previous step, define updated geometry. The incremental stiffness matrix, used to determine these incremental displacements, is updated at the conclusion of each step to reflect changes due to displacement nonlinearities. Because small increments



(a) ORIGINAL GEAR POSITION

(b) DISPLACED GEAR POSITION



(c) TYPICAL NONLINEAR LOAD–STROKE CURVE

FIGURE 4–14 LARGE DISPLACEMENTS OF A TYPICAL TWO–DIMENSIONAL GEAR are used, a linear small displacement problem is solved at each step. The method is illustrated in Figure 4-15 for a typical nonlinear load-stroke curve. For each step n, the incremental displacement ($\delta_n - \delta_{n-1}$) corresponding to the applied incremental load ($F_n - F_{n-1}$) is determined from the equation

$$\delta_n = \delta_{n-1} + (F_n - F_{n-1})/S_n, \quad n = 1, \ldots, m \tag{4-22}$$

where m is the total number of incremental steps. In this equation, S_n is the incremental stiffness of the structure corresponding to the displaced state, δ_{n-1}. For example, at the conclusion of step 1 (point A_1 in Figure 4-15), the incremental stiffness S_2 (at point P_1) is determined for the displaced state, δ_1. This stiffness and the next incremental load ($F_2 - F_1$) are then used in Equation (4-22) to determine δ_2 (defining point A_2),

$$\delta_2 = \delta_1 + (F_2 - F_1)/S_2 \tag{4-23}$$

This process is repeated until all load increments have been applied. For the gear of Figure 4-14 the above method would be applied simultaneously to the nonlinear load-stroke curves in the X and Y directions.

Due to the linear approximation at each step of the incremental stiffness method, error in the solution accumulates as a function of step size. This can be seen in Figure 4-15 where at the conclusion of step 4 the difference between points A_4 and P_4 is significant. This error, the difference between the applied load (F_4) and the true load corresponding to the displaced state (the force at point P_4), is defined as the load unbalance. To eliminate this unbalance, the incremental stiffness method is combined with an iteration procedure to assure convergence to the correct solution. For each incremental step, iteration is employed to insure that the internal loads, corresponding to the displaced state, are in equilibrium with the applied loads.

The iteration approach as applied to the nth incremental step of Figure 4-15 is illustrated in Figure 4-16. The incremental displacement $\delta_n - \delta_{n-1}$, corresponding to the incremental applied load $F_n - F_{n-1}$, is sought. Knowing the incremental stiffness S_1 at the beginning of the nth incremental step, the first estimate of the incremental displacement, $\Delta\delta_1$, is found from

$$\Delta\delta_1 = (F_n - F_{n-1})/S_1 \tag{4-24}$$

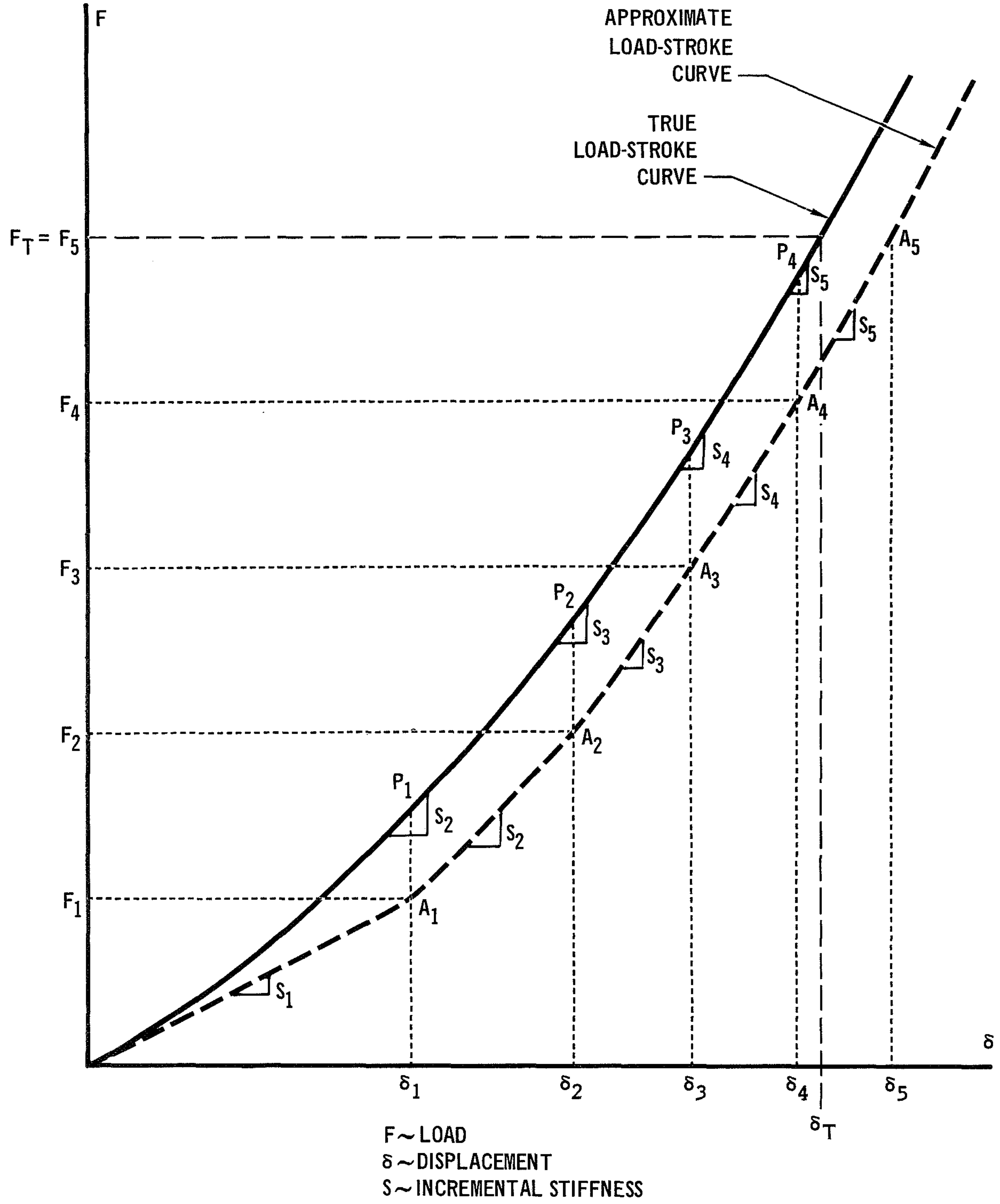
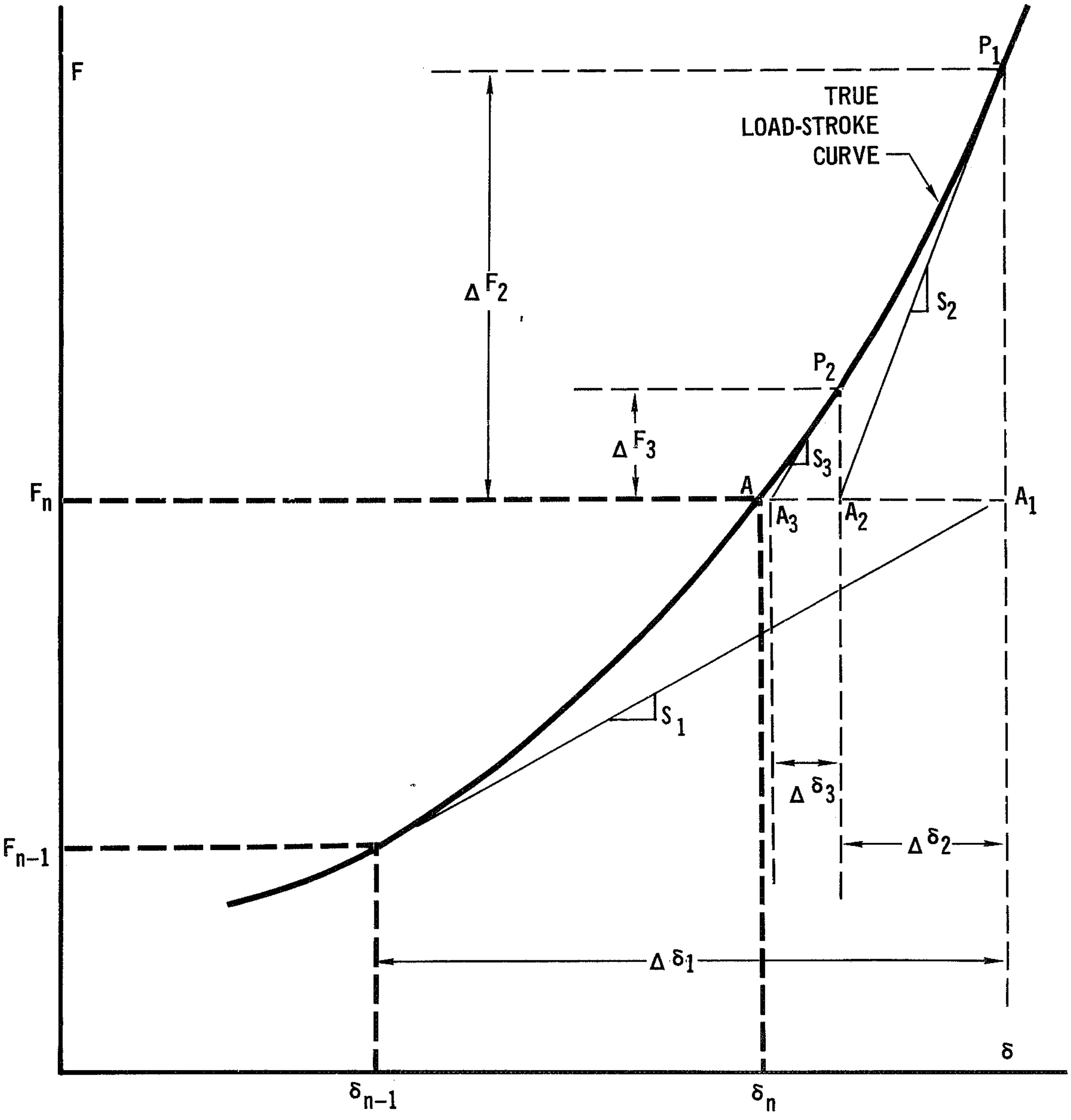


FIGURE 4–15 INCREMENTAL STIFFNESS METHOD APPLIED TO A TYPICAL NONLINEAR LOAD-STROKE CURVE



F ~ LOAD
δ ~ DISPLACEMENT
S ~ INCREMENTAL STIFFNESS

FIGURE 4–16 TYPICAL INCREMENTAL STEP OF MODIFIED INCREMENTAL STIFFNESS METHOD

This yields point A_1. The true incremental stiffness S_2 and the resultant strut reaction load $F_n + \Delta F_2$ corresponding to the displacement state $\delta_{n-1} + \Delta\delta_1$ are defined by point P_1. The force unbalance ΔF_2, the difference between the total applied load, F_n, and the reaction load $F_n + \Delta F_2$, is applied to the structure whose incremental stiffness is S_2. Thus, the new estimate of the incremental displacement, $\Delta\delta_1 + \Delta\delta_2$, is determined where $\Delta\delta_2$ is defined by

$$\Delta\delta_2 = \Delta F_2/S_2 \tag{4-25}$$

This process is repeated and a new estimate of the incremental displacement, $\sum_{i=1}^{\ell} \Delta\delta_i$, is determined using the relation

$$\Delta\delta_\ell = \Delta F_\ell/S_\ell, \quad \ell = 3, \ldots. \tag{4-26}$$

until ΔF_ℓ is arbitrarily small (convergence at point A). When this occurs, the applied load is balanced by the reaction load and equilibrium has been attained.

For the gear of Figure 4-14, the above method would be applied simultaneously to the nonlinear load-stroke curves in the X and Y directions. For each of these directions, the force unbalance would be the difference between the applied load in that direction and the sum of the internal drag strut and main strut loads in that direction.

The modified incremental stiffness method is well suited to the analysis of landing gears. When employing this method, errors will not be introduced when structural stiffness properties change abruptly as is common in landing gear members containing attenuation. The error at each step of the process is known since the system is in equilibrium within a predetermined tolerance. In addition, the method provides a displacement-load history for any desired number of increments of applied gear stroke. This is essential for determining energy absorption characteristics of landing gears.

The modified incremental stiffness method, as applied to three dimensional inverted tripod or cantilever gears, requires the solution of the matrix equation

$$\Delta\{F\} = [S]\,\Delta\{\delta\} \tag{4-27}$$

Equation (4-27) is solved several times for each step of the loading. In this equation, $\Delta\{F\}$ is a column matrix of incremental forces applied at the nodes of the gear; $\Delta\{\delta\}$ is a column matrix of incremental nodal displacements; and [S] is the instantaneous incremental stiffness of the gear. For landing gear analysis, the independent variables in Equation (4-27) for each step of the process will be the applied displacements. The method of employing Equation (4-27) for each step of the process is as follows:

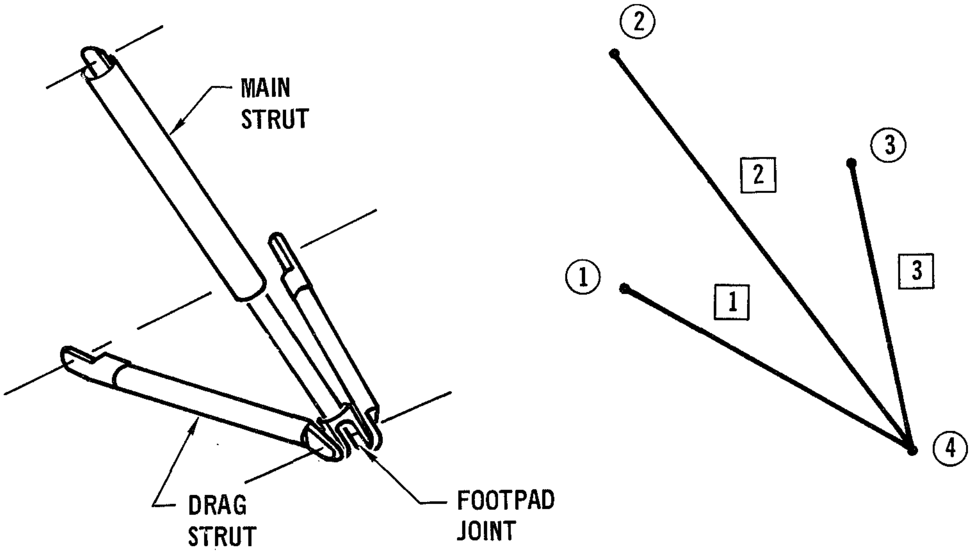
(i) Equation (4-27) is solved once for the case where $\Delta\{\delta\}$ is the total applied displacement vector divided by the number of incremental steps. The incremental stiffness matrix [S] of the gear corresponds to the displaced conditions existing at the conclusion of the previous step. This resulting equation is analogous to Equation (4-22). The nodal locations are then updated.

(ii) Equation (4-27) is then solved repeatedly (iteration is employed) until the largest load unbalance component is less than a predetermined tolerance. At the conclusion of each iteration, nodal locations are updated. For the first of these iterations, [S] corresponds to the displaced condition existing at the conclusion of (i). For all succeeding iterations, [S] corresponds to the displaced condition existing at the conclusion of the previous iteration. $\Delta\{F\}$ is a vector representing the difference between the applied load components and the components of internal load corresponding to the displaced condition existing at the conclusion of the previous iteration. Hence, $\Delta\{F\}$ represents the load unbalances. Each of the above equations is analogous to Equation (4-26).

4.1.2.2 Structural Idealization - The Landing Gear Option employs a fixed idealization for inverted tripod and cantilever landing gears. This idealization (node point and element numbering) must be adhered to when employing this option of the program.

The inverted tripod landing gear is idealized in the program with four nodes and three pin-ended elements (the main strut and two drag struts) as shown in Figure 4-17(a). These elements are capable of carrying axial loads only and may contain honeycomb attenuation for both tension and compression. Each element may be made of a different material. Each node is assumed to be

(a) INVERTED TRIPOD GEAR



NOTE:

② JOINT NUMBER

$\boxed{4}$ ELEMENT NUMBER

(b) CANTILEVER GEAR

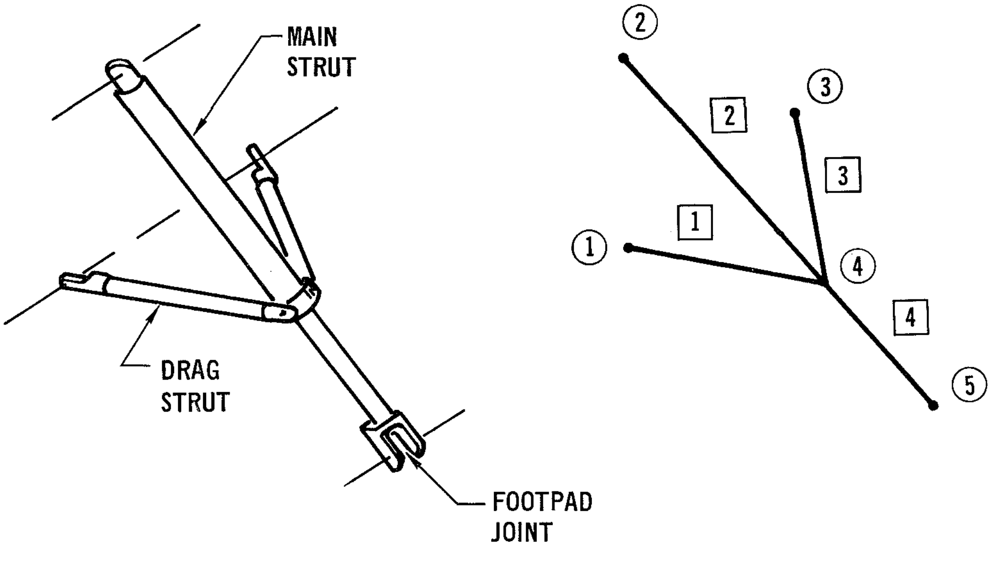


FIGURE 4–17 LANDING GEAR IDEALIZATIONS

a pin which can support three force components but no moments. Each element of a gear has associated with it a set of three nodes (p, q, and r) which defines the Local Coordinate System, as explained in Section 4.1.2.3, for that element. The fixed idealization of the inverted tripod gear employs the following nodal point numbering system for the elements:

Element No.	p	q	r
1	1	4	2
2	2	4	3
3	3	4	2

The cantilever landing gear is idealized with five nodes and four elements as shown in Figure 4-17(b). The main strut is idealized with two elements, both of which are capable of carrying bending as well as axial loads. These members (elements 2 and 4 in Figure 4-17(b)) are assumed to be rigidly connected at node 4. This provides moment continuity along the main strut. For elements 2 and 4 of the main strut, the moment of inertia about any axis normal to the element is assumed to be constant. However, the moment of inertia for element 2 may be different than that for element 4. For each of these elements, the modulus of elasticity for bending displacements may be different than the modulus for axial displacements. The junctions of the main strut with the center body (node 2) and the footpad (node 5) are assumed to be pinned. The drag struts (elements 1 and 3) are pin-ended with axial load-carrying capability only. Thus, a drag strut cannot carry bending moments at either end. Both drag struts (elements 1 and 3) and the lower element of the main strut (element 4) may contain honeycomb attenuation for both tension and compression. Each strut of the cantilever gear may be made of a different material. The fixed idealization of the cantilever gear employs the following nodal point numbering system for the elements:

Element No.	p	q	r
1	1	4	2
2	2	4	RP
3	3	4	2
4	5	4	RP

Point r (RP) for elements 2 and 4 is a floating reference point whose coordinates are continually changing. Initially, this point is selected as node 1. As the main strut bends, the floating reference point is located as described in Section 4.1.2.5.

For either of the above gears, struts which can carry axial loads only (all elements of the inverted tripod gear and elements 1 and 3 of the cantilever gear) are defined as "axial struts." Struts which are capable of carrying bending as well as axial loads (elements 2 and 4 of the cantilever gear) are defined as "bending struts."

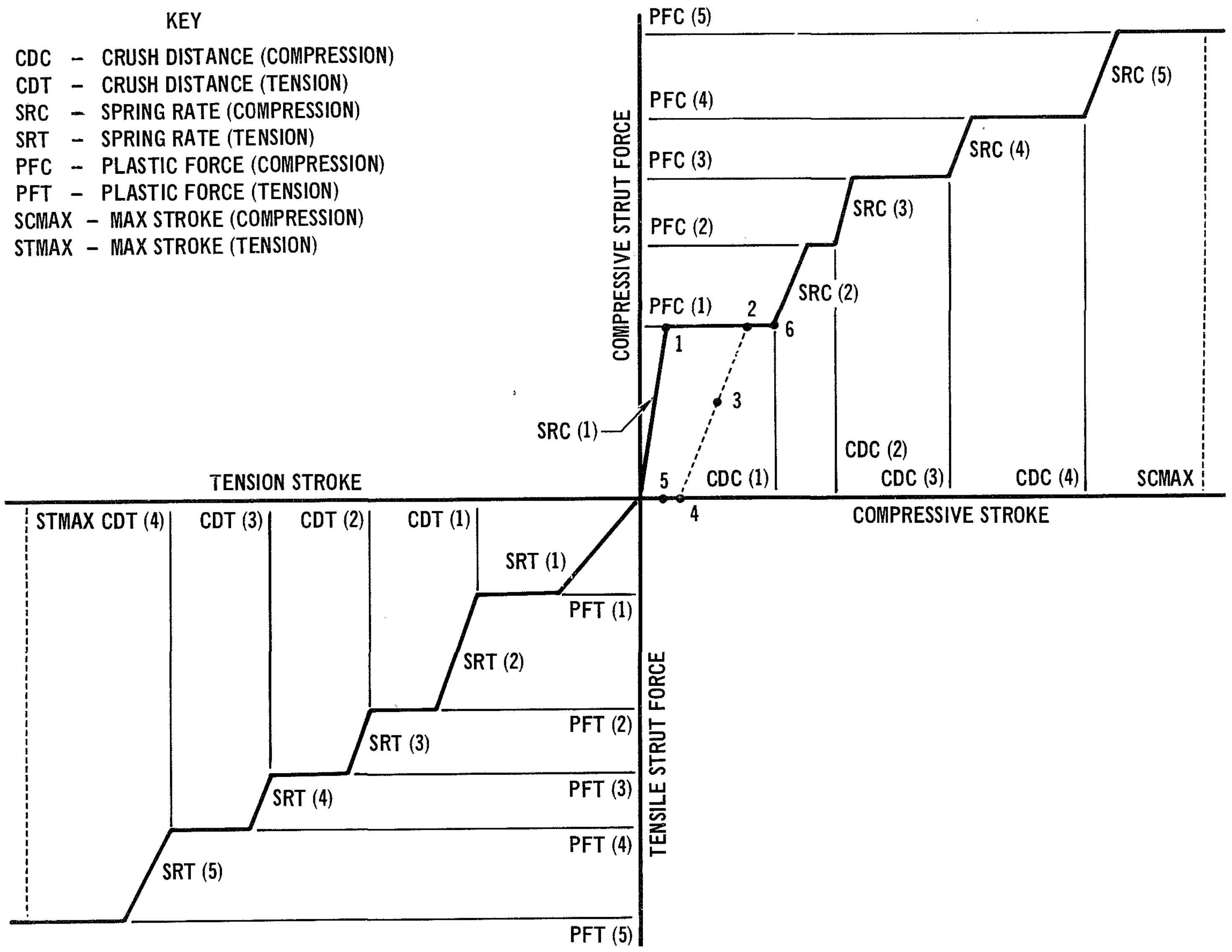
For a strut which contains honeycomb attenuation, the material defining the strut properties is assumed to have a load-stroke curve for axial displacements similar to that shown in Figure 4-18. This curve represents the stroking characteristics of stacked honeycomb cartridges housed within a landing gear strut. Each cartridge is assumed to crush at constant load as the strut is stroked. Cartridges possessing different crushing strengths may be stacked in series to form a desired load-stroke characteristic. Up to five cartridges can be used to attenuate compression loads and up to five to attenuate tension loads.

For the typical load-stroke curve shown in Figure 4-18, determination of the axial load corresponding to stroking causing compression in the strut is explained in the following discussion. The axial load corresponding to tensile stroking of the strut would be determined in a similar manner.

As the strut initially begins to stroke, the strut load increases linearly with stroke to point 1 where the first crush load is reached. The load then remains constant with stroke until either the stroke reverses direction or a second elastic portion is reached. If the direction of stroke reverses, point 2, one of the following load-stroke sequences is possible:

(1) Elastic unloading to an intermediate point between 2 and 4, such as point 3, at which time the compressive stroke again increases. This results in the load increasing elastically to point 2 and then following the original load-stroke curve.

(2) Elastic unloading through point 3 to point 4. A continued decrease in stroke to point 5 occurs at a zero strut load. With a reversal of stroke the strut will compress with zero load until point 4 is



KEY

CDC – CRUSH DISTANCE (COMPRESSION)
CDT – CRUSH DISTANCE (TENSION)
SRC – SPRING RATE (COMPRESSION)
SRT – SPRING RATE (TENSION)
PFC – PLASTIC FORCE (COMPRESSION)
PFT – PLASTIC FORCE (TENSION)
SCMAX – MAX STROKE (COMPRESSION)
STMAX – MAX STROKE (TENSION)

FIGURE 4–18 TYPICAL STRUT LOAD-STROKE CURVE again reached. The load will then increase linearly to point 2 and continue to follow the original load-stroke curve.

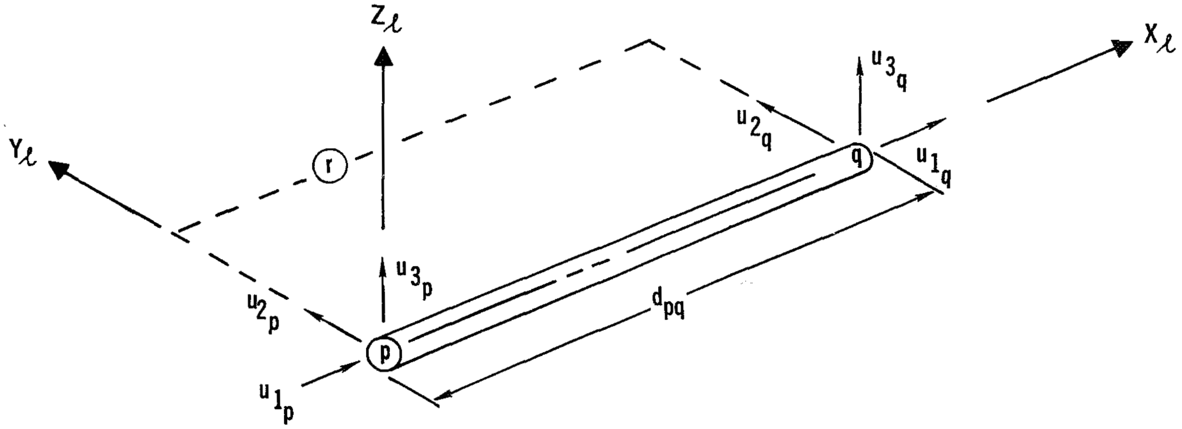
(3) Elastic unloading through points 3, 4, and 5, followed by a continued decrease in stroke until the strut goes into tension. At this point, the strut load is governed by its tension load-stroke characteristics.

If at point 2 the stroke had not reversed direction, the load would remain constant until point 6 was reached. The force would then increase linearly with stroke and continue to follow the load-stroke curve until unloading took place. When the compressive stroke exceeds the maximum allowable stroke (SCMAX) the strut is assumed to have bottomed out.

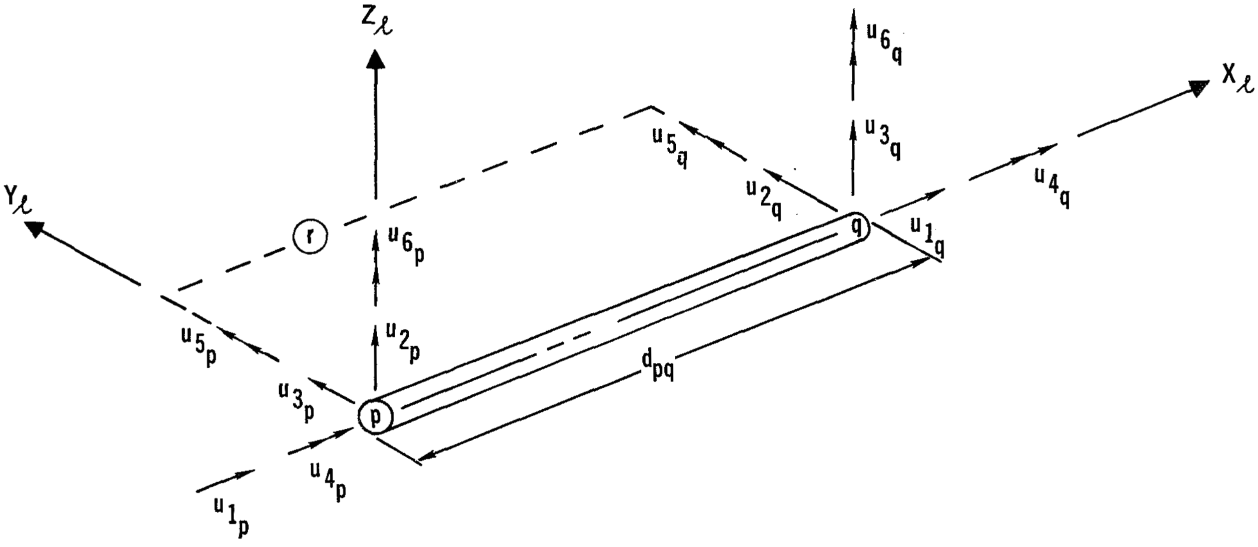
Two options are available for selecting the slope of the strut's elastic unloading characteristic. This slope may be either the slope of the next elastic portion of the load-stroke curve, or it may be some input value. The first option is governed by the assumption that at any point in the stroking of a strut, the elastic slope of the load-stroke curve is determined by the elastic properties of the uncrushed cartridges acting in series. Thus, when a cartridge crushes, its elastic characteristic is no longer reflected in the strut load-stroke curve.

For a bending strut which may or may not contain honeycomb attenuation, the bending properties (modulus of elasticity for bending displacements and the moment of inertia) are assumed to remain constant.

4.1.2.3 Coordinate Systems - Two types of right-hand orthogonal coordinate systems employed in the Landing Gear Option are the Local Coordinate System and the Global Coordinate System. Each axial strut or bending strut of either gear has its set of local coordinates as illustrated in Figure 4-19. The Local Coordinate System origin for each strut is located at node "p," with the X_ℓ axis aligned along the member axis. Positive X_ℓ is on the side of node "p" towards node "q." The Y_ℓ axis is perpendicular to X_ℓ and is located in the "pqr" plane. Positive Y_ℓ is on the side of X_ℓ towards "r." The Z_ℓ axis is then established using the right hand rule. Nodes p, q, and r, which establish each strut's local axis system, were preselected for all struts of the inverted tripod and cantilever gears.



(a) Axial Struts



(b) Bending Struts

FIGURE 4–19 STRUT DISPLACEMENTS IN LOCAL COORDINATE SYSTEM

Unless noted otherwise, in the following the terms "displacement" and "load" refer to incremental displacement or incremental load occurring during a step of the modified incremental stiffness method. The symbol "Δ" signifying an incremental quantity (see Equation (4-27)) is dropped for convenience.

Displacement notation of a typical axial strut capable of carrying axial load only is indicated in Figure 4-19(a) for the Local Coordinate System. Subscript 1 refers to displacement due to axial load while subscripts 2 and 3 are for displacements due to shear loads. Although axial struts can carry axial load only, displacements due to shear loads must be included in large displacement analysis as will be seen in the following sections. Cross-sectional area A, and modulus of elasticity E_A for axial displacements, must be specified for each axial strut relative to its Local Coordinate System. For an axial strut which contains honeycomb attenuation, this information is supplied in the form of the load-stroke curve shown in Figure 4-18.

Displacement notation of a typical bending strut capable of carrying axial load, shear in two directions, and bending in two directions is indicated in Figure 4-19(b) for the Local Coordinate System. Subscript 1 refers to a displacement due to axial load, subscripts 2 and 3 are for displacements due to shear loads, subscript 4 is for rotation due to torque, and subscripts 5 and 6 are for rotations due to moments. Although bending struts are not capable of carrying torsional loads, it is advantageous to include the torsional degrees of freedom since matrix transformation to a common coordinate system will be accomplished later and in this system all nodes will have six degrees of freedom. Cross-sectional area A, and modulus of elasticity E_A for axial displacements, must be specified for each bending strut relative to its Local Coordinate System. For a bending strut which contains honeycomb attenuation, this information is supplied in the form of the load-stroke curve shown in Figure 4-18. Moment of inertia I, and the modulus of elasticity E_B for bending displacements, are specified for each bending strut relative to its Local Coordinate System. The moment of inertia is assumed to be constant about any axis normal to X_ℓ.

Two Global Coordinate Systems employed in the Landing Gear Option are the Lander Coordinate System and the Surface Coordinate System as shown in

Figure 4-20. The origins of these coordinate systems are assumed coincident. Coordinates of all nodes of the gear are input in the Lander Coordinate System.

The Surface Coordinate System must be chosen such that the axis normal to the surface, X_S in Figure 4-20, is pointed outward. Three Euler angles defined as yaw (ψ), pitch (θ), and roll (ϕ), are used to orient the Surface Coordinate System with respect to the Lander Coordinate System. That is, the orientation of the Surface Coordinate System can be found by rotations in the order ψ, θ, and ϕ about the rotated Z_S, Y_S and X_S surface axes, respectively. The transformation matrix $[T_G]$ relating vector components in the Surface Coordinate System (VX_S, VY_S, VZ_S) to vector components in the Lander Coordinate System (VX_L. VY_L, VZ_L) is defined by Equation (4-28).

$$\begin{Bmatrix} VX_S \\ VY_S \\ VZ_S \end{Bmatrix} = [T_G] \begin{Bmatrix} VX_L \\ VY_L \\ VZ_L \end{Bmatrix} \tag{4-28}$$

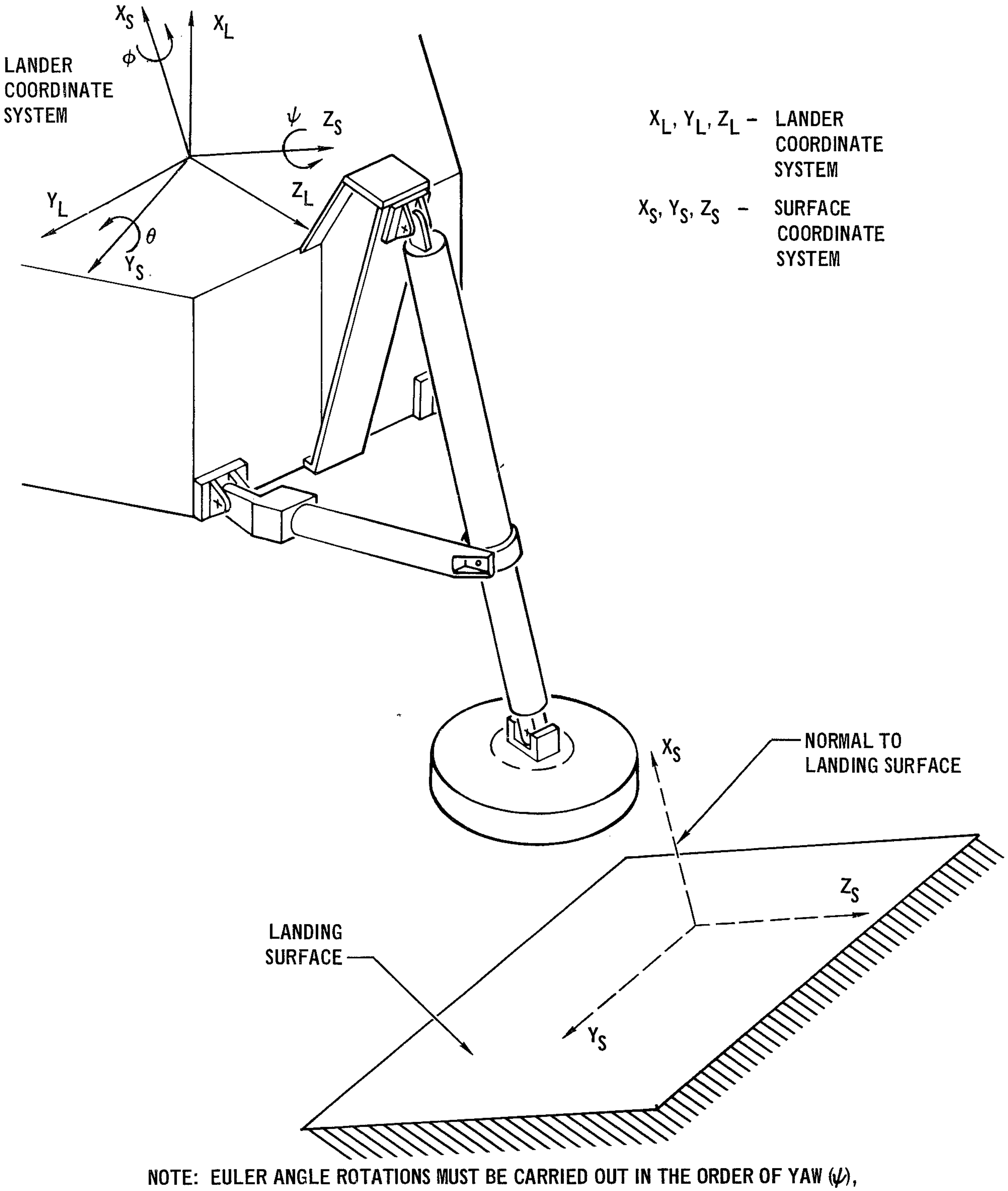
Terms in the transformation matrix $[T_G]$ are defined in Equation (4-29).

$$[T_G] = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \tag{4-29}$$

$$\begin{aligned}
D_{11} &= \text{Cos } \theta \text{ Cos } \psi \\
D_{12} &= \text{Cos } \theta \text{ Sin } \psi \\
D_{13} &= -\text{Sin } \theta \\
D_{21} &= \text{Sin } \phi \text{ Sin } \theta \text{ Cos } \psi - \text{Cos } \phi \text{ Sin } \psi \\
D_{22} &= \text{Sin } \phi \text{ Sin } \theta \text{ Sin } \psi + \text{Cos } \phi \text{ Cos } \psi \\
D_{23} &= \text{Sin } \phi \text{ Cos } \theta \\
D_{31} &= \text{Cos } \phi \text{ Sin } \theta \text{ Cos } \psi + \text{Sin } \phi \text{ Sin } \psi \\
D_{32} &= \text{Cos } \phi \text{ Sin } \theta \text{ Sin } \psi - \text{Sin } \phi \text{ Cos } \psi \\
D_{33} &= \text{Cos } \phi \text{ Cos } \theta
\end{aligned}$$

Coordinates of all node points of the gear are transformed to surface coordinate orientation using Equation (4-28).

The Surface Coordinate System is established as the common system for all elements of a gear so that element total and incremental loads and displacements may be related to a common frame of reference. The term "total",when applied to loads or displacements,refers to the sum of incremental quantities which have occurred during the incremental steps which have been taken to the



NOTE: EULER ANGLE ROTATIONS MUST BE CARRIED OUT IN THE ORDER OF YAW (ψ), PITCH (θ), AND ROLL (ϕ) ABOUT SURFACE COORDINATE AXES.

FIGURE 4-20 LANDING GEAR COORDINATE SYSTEMS current point of interest. Displacement notation for the axial strut and bending strut in the Surface Coordinate System is shown in Figure 4-21. In this figure and in the following discussion, the Surface Coordinate System X_S, Y_S, Z_S will be referred to as $\overline{X}$, $\overline{Y}$, $\overline{Z}$. Footpad loading conditions (see Section 4.1.2.7) are specified in this system. Matrix assembly and solution of equations for each step of the modified incremental stiffness method (see Section 4.1.2.6) are also accomplished in the surface system. That is, external loads and displacements at all nodes of the gear, as well as the total energy absorbed by the gear, are determined in surface coordinate components.

4.1.2.4 Incremental Stiffness Matrix - The incremental stiffness matrix, employed in the modified incremental stiffness method, is generated for both an "axial" strut and "bending" strut in the Local Coordinate System. This stiffness matrix will be shown to be the sum of the standard small displacement stiffness and the large displacement geometric stiffness. The geometric stiffness matrix is derived in Reference (8) using basic nonlinear theory in conjunction with the incremental stiffness method.

For each strut of a landing gear, incremental loads and displacements in the Local Coordinate System are related in matrix form by

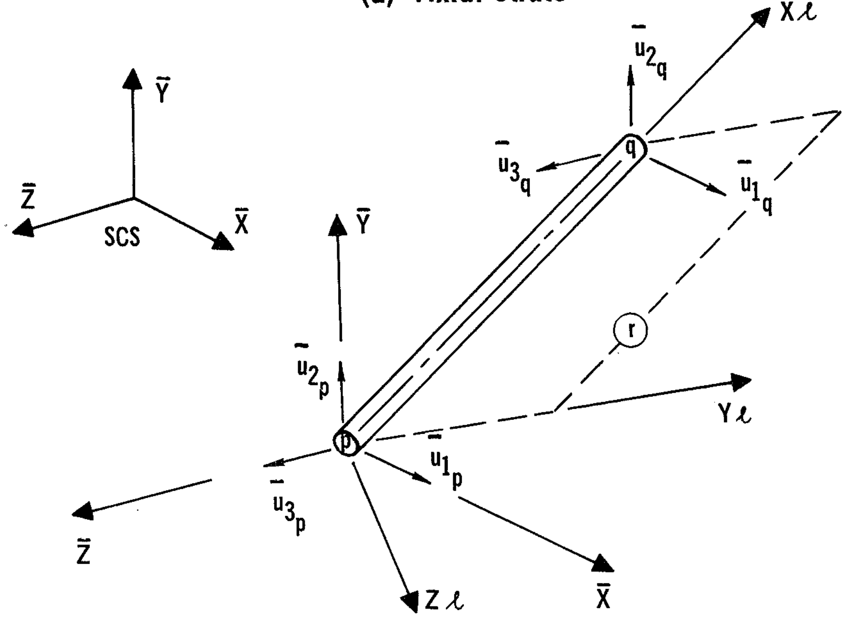
$$\Delta\{F_\ell\} = [S_\ell]\,\Delta\{\delta_\ell\} \tag{4-30}$$

In Equation (4-30), $[S_\ell]$ is the incremental stiffness matrix; $\Delta\{F_\ell\}$ is a column matrix of applied incremental loads at the nodes; and $\Delta\{\delta_\ell\}$ is the column matrix of incremental displacements at the nodes. In the following, for convenience, the Δ symbol will be dropped. Accordingly, unless otherwise noted, the terms displacement and load refer to incremental quantities.

A. Axial Struts - As defined in Section 4.1.2.2, all elements of the inverted tripod gear and elements 1 and 3 of the cantilever gear (Figure 4-17(b)) are "axial" struts with load-carrying capability in the axial direction only. Displacement notation in the Local Coordinate System for a typical axial strut was shown in Figure 4-19(a).

The incremental stiffness matrix $[S_\ell]$ of an axial strut (in local coordinates) is the sum of the small displacement stiffness matrix shown in Figure 4-22 and the geometric stiffness matrix shown in Figure 4-23. The small displacement stiffness matrix of an axial strut is equivalent to the stiffness

(a) Axial Struts



(b) Bending Struts

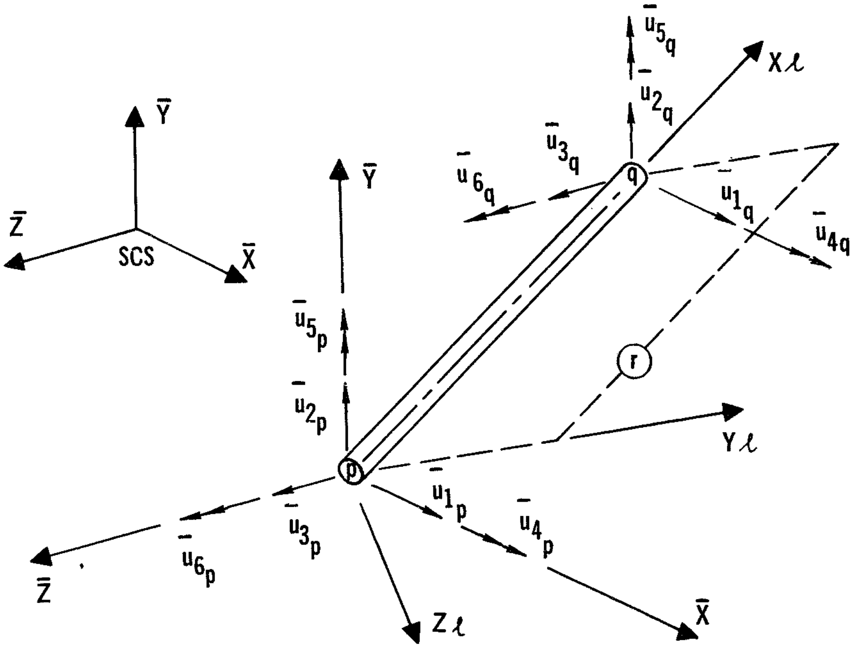


FIGURE 4–21 STRUT DISPLACEMENTS IN SURFACE COORDINATE SYSTEM

$$\begin{array}{c|cccccc|}
 & u_{1_p} & u_{2_p} & u_{3_p} & u_{1_q} & u_{2_q} & u_{3_q} \\
\hline
F_{1_p} & \frac{AE_A}{d_{pq}} & & & \frac{-AE_A}{d_{pq}} & & \\
F_{2_p} & & & & & & \\
F_{3_p} & & & & & & \\
F_{1_q} & \frac{-AE_A}{d_{pq}} & & & \frac{AE_A}{d_{pq}} & & \\
F_{2_q} & & & & & & \\
F_{3_q} & & & & & & \\
\hline
\end{array}$$

FIGURE 4–22 SMALL DISPLACEMENT STIFFNESS MATRIX OF AN AXIAL STRUT

$$
\begin{array}{c|cccccc|}
 & u_{1_p} & u_{2_p} & u_{3_p} & u_{1_q} & u_{2_q} & u_{3_q} \\
\hline
F_{1_p} & & & & & & \\
\hline
F_{2_p} & & \frac{P_0}{d_{pq}} & & & \frac{-P_0}{d_{pq}} & \\
\hline
F_{3_p} & & & \frac{P_0}{d_{pq}} & & & \frac{-P_0}{d_{pq}} \\
\hline
F_{1_q} & & & & & & \\
\hline
F_{2_q} & & \frac{-P_0}{d_{pq}} & & & \frac{P_0}{d_{pq}} & \\
\hline
F_{3_q} & & & \frac{-P_0}{d_{pq}} & & & \frac{P_0}{d_{pq}} \\
\hline
\end{array}
$$

FIGURE 4–23 GEOMETRIC STIFFNESS MATRIX OF AN AXIAL STRUT matrix of the general bar element in the Center Body Option (see Figure 4-4) with $I_N = I_T = J = 0$ and only the translational degrees of freedom included. In Figure 4-22, A is the cross-sectional area; E_A the modulus of elasticity for axial displacements; and dpq the current length of the strut. For an axial strut composed of honeycomb attenuation, the terms ($A\ E_A$/dpq) in the small displacement stiffness matrix are replaced by the slope of the load-stroke curve (see Figure 4-18) corresponding to the current axial stroke of the strut.

The geometric stiffness matrix, shown in Figure 4-23, includes the effect of "total" internal axial loads on the equilibrium equations in the presence of large rotational displacements. In the stiffness terms, P_o is the total internal axial load in the strut corresponding to the current total axial displacement (stroke) of the strut. Positive values for P_o signify that the strut is in tension.

B. Bending Struts - As defined in Section 4.1.2.2, the main strut of the cantilever gear (elements 2 and 4 in Figure 4-17(b)) is composed of "bending" struts with bending capability as well as axial capability. Displacement notation in the Local Coordinate System for a typical bending strut was shown in Figure 4-19(b).

The incremental stiffness matrix $[S_\ell]$ of a bending strut (in local coordinates) is the sum of the small displacement stiffness matrix shown in Figure 4-24 and the geometric stiffness matrix shown in Figure 4-25. The small displacement stiffness matrix is equivalent to the stiffness matrix of the general bar element in the Center Body Option (see Figure 4-4) with $J = 0$ and $I_N = I_T = I$. In Figure 4-24, A is the cross-sectional area; E_A the modulus of elasticity for axial displacements; E_B the modulus of elasticity for bending displacements; I the moment of inertia about any axis normal to the strut; and dpq the current length of the strut. For a bending strut whose axial capability is defined by honeycomb attenuation, the terms ($A\ E_A$/dpq) in the small displacement stiffness matrix are replaced by the slope of the load-stroke curve (see Figure 4-18) corresponding to the current axial stroke of the strut.

The geometric stiffness matrix, shown in Figure 4-25, includes the effect of total internal axial loads on the equilibrium equations in the presence of large rotational displacements. In Figure 4-25, P_o is the total internal axial load in the strut corresponding to the current total axial displacement (stroke). Positive values for P_o signify tension.

$$
\begin{array}{c|cccccccccccc|}
 & u_{1_p} & u_{2_p} & u_{3_p} & u_{1_q} & u_{2_q} & u_{3_q} & u_{4_p} & u_{5_p} & u_{6_p} & u_{4_q} & u_{5_q} & u_{6_q} \\
\hline
F_{1_p} & \frac{AE_A}{d_{pq}} & & & \frac{-AE_A}{d_{pq}} & & & & & & & & \\
F_{2_p} & & \frac{12\,E_B I}{(d_{pq})^3} & & & \frac{-12\,E_B I}{(d_{pq})^3} & & & & \frac{6\,E_B I}{(d_{pq})^2} & & & \frac{6\,E_B I}{(d_{pq})^2} \\
F_{3_p} & & & \frac{12\,E_B I}{(d_{pq})^3} & & & \frac{-12\,E_B I}{(d_{pq})^3} & & \frac{-6\,E_B I}{(d_{pq})^2} & & & \frac{-6\,E_B I}{(d_{pq})^2} & \\
F_{1_q} & \frac{-A\,E_A}{d_{pq}} & & & \frac{A\,E_A}{d_{pq}} & & & & & & & & \\
F_{2_q} & & \frac{-12\,E_B I}{(d_{pq})^3} & & & \frac{12\,E_B I}{(d_{pq})^3} & & & & \frac{-6\,E_B I}{(d_{pq})^2} & & & \frac{-6\,E_B I}{(d_{pq})^2} \\
F_{3_q} & & & \frac{-12\,E_B I}{(d_{pq})^3} & & & \frac{12\,E_B I}{(d_{pq})^3} & & \frac{6\,E_B I}{(d_{pq})^2} & & & \frac{6\,E_B I}{(d_{pq})^2} & \\
M_{4_p} & & & & & & & & & & & & \\
M_{5_p} & & & \frac{-6\,E_B I}{(d_{pq})^2} & & & \frac{6\,E_B I}{(d_{pq})^2} & & \frac{4\,E_B I}{d_{pq}} & & & \frac{2\,E_B I}{d_{pq}} & \\
M_{6_p} & & \frac{6\,E_B I}{(d_{pq})^2} & & & \frac{-6\,E_B I}{(d_{pq})^2} & & & & \frac{4\,E_B I}{d_{pq}} & & & \frac{2\,E_B I}{d_{pq}} \\
M_{4_q} & & & & & & & & & & & & \\
M_{5_q} & & & \frac{-6\,E_B I}{(d_{pq})^2} & & & \frac{6\,E_B I}{(d_{pq})^2} & & \frac{2\,E_B I}{d_{pq}} & & & \frac{4\,E_B I}{d_{pq}} & \\
M_{6_q} & & \frac{6\,E_B I}{(d_{pq})^2} & & & \frac{-6\,E_B I}{(d_{pq})^2} & & & & \frac{2\,E_B I}{d_{pq}} & & & \frac{4\,E_B I}{d_{pq}} \\
\hline
\end{array}
$$

FIGURE 4–24 SMALL DISPLACEMENT STIFFNESS MATRIX OF A BENDING STRUT

$$
\begin{array}{c|cccccccccccc|}
 & u_{1_p} & u_{2_p} & u_{3_p} & u_{1_q} & u_{2_q} & u_{3_q} & u_{4_p} & u_{5_p} & u_{6_p} & u_{4_q} & u_{5_q} & u_{6_q} \\
\hline
F_{1_p} & & & & & & & & & & & & \\
F_{2_p} & & \frac{6\,P_o}{5\,d_{pq}} & & & \frac{-6\,P_o}{5\,d_{pq}} & & & & \frac{P_o}{10} & & & \frac{P_o}{10} \\
F_{3_p} & & & \frac{6\,P_o}{5\,d_{pq}} & & & \frac{-6\,P_o}{5\,d_{pq}} & & \frac{-P_o}{10} & & & \frac{-P_o}{10} & \\
F_{1_q} & & & & & & & & & & & & \\
F_{2_q} & & \frac{-6\,P_o}{5\,d_{pq}} & & & \frac{6\,P_o}{5\,d_{pq}} & & & & \frac{-P_o}{10} & & & \frac{-P_o}{10} \\
F_{3_q} & & & \frac{-6\,P_o}{5\,d_{pq}} & & & \frac{6\,P_o}{5\,d_{pq}} & & \frac{P_o}{10} & & & \frac{P_o}{10} & \\
M_{4_p} & & & & & & & & & & & & \\
M_{5_p} & & & \frac{-P_o}{10} & & & \frac{P_o}{10} & & \frac{2\,P_o d_{pq}}{15} & & & \frac{-P_o d_{pq}}{30} & \\
M_{6_p} & & \frac{P_o}{10} & & & \frac{-P_o}{10} & & & & \frac{2\,P_o d_{pq}}{15} & & & \frac{-P_o d_{pq}}{30} \\
M_{4_q} & & & & & & & & & & & & \\
M_{5_q} & & & \frac{-P_o}{10} & & & \frac{P_o}{10} & & \frac{-P_o d_{pq}}{30} & & & \frac{2\,P_o d_{pq}}{15} & \\
M_{6_q} & & \frac{P_o}{10} & & & \frac{-P_o}{10} & & & & \frac{-P_o d_{pq}}{30} & & & \frac{2\,P_o d_{pq}}{15}
\end{array}
$$

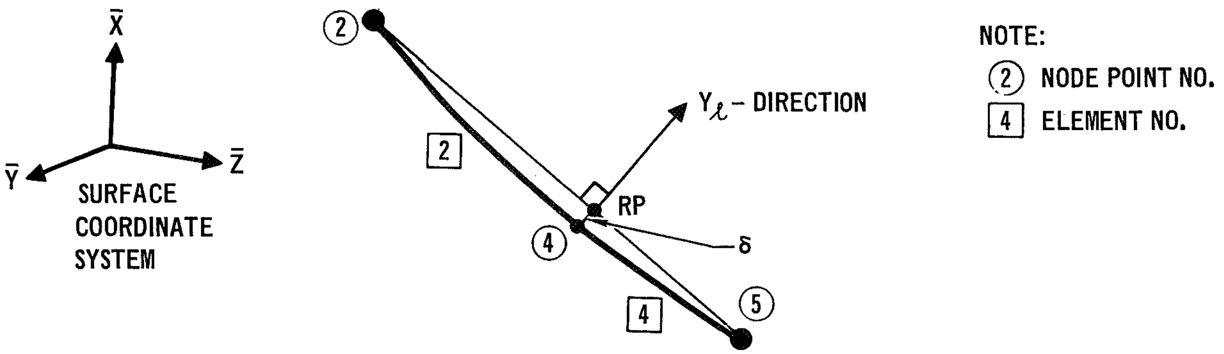
FIGURE 4–25 GEOMETRIC STIFFNESS MATRIX OF A BENDING STRUT

4.1.2.5 Internal Loads - For each iteration at each step of the modified incremental stiffness method, the total internal loads in each strut must be determined for the current displaced state of the strut. The total internal loads must be calculated so that load unbalances can be determined (see Section 4.1.2.1).

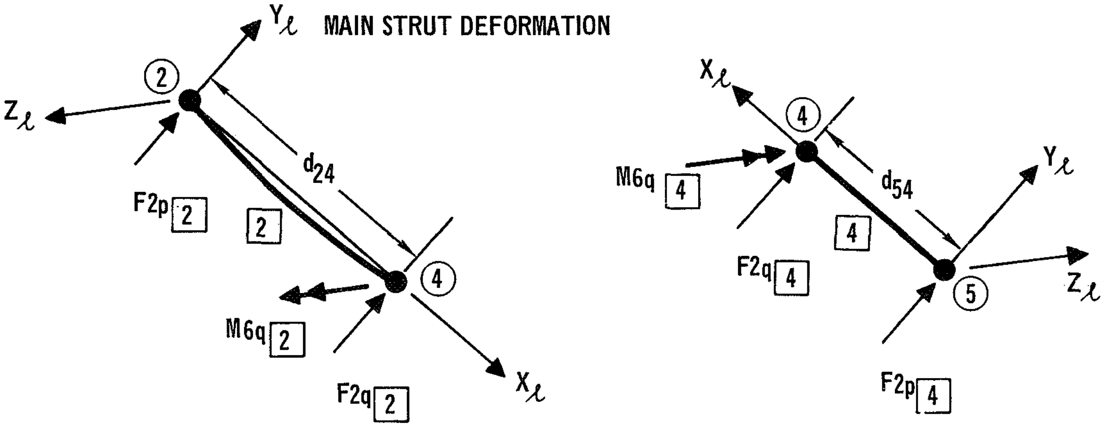
For axial struts, the total internal loads are simply the axial tension or compression forces which would have to be applied to the ends of the strut to extend or compress it to its present state. The extension or compression of a strut (the stroke) is the difference between the current length and the original length. Total axial force is the product of stroke and the small displacement stiffness term ($A\ E_A/dpq$) defined in Section 4.1.2.4. A positive stroke corresponds to extension of the strut and yields a positive (tensile) total axial force. For an axial strut composed of honeycomb attenuation, the total axial force is determined by the load-stroke curve (see Figure 4-18).

The total internal loads for bending struts consist of axial forces, shear forces, and bending moments. The axial forces are calculated in the same way as for axial struts. Calculation of bending loads (shear forces and bending moments) in the deflected main strut of the cantilever gear (elements 2 and 4 in Figure 4-17) is shown in Figure 4-26. Because the main strut is pinned at both ends, loads causing bending are shear loads ($F2p_{\boxed{2}}$ and $F2p_{\boxed{4}}$) applied at the ends in the plane of the deformed strut and directed along the Y_ℓ axis. The Y_ℓ direction is defined by a line passing through node 4 perpendicular to the line connecting nodes 2 and 5 and intersecting it at reference point RP. Since there is bending continuity at node 4, internal loads in elements 2 and 4 are shears ($F2q_{\boxed{2}}$ and $F2q_{\boxed{4}}$) and equal bending moments ($M6q_{\boxed{2}}$ and $M6q_{\boxed{4}}$). Coordinates of the floating reference point (RP) for elements 2 and 4 are also calculated in Figure 4-26.

4.1.2.6 Matrix Assembly and Solution of Equations - As explained in Section 4.1.2.3, the Surface Coordinate System is used for all elements of a gear to relate total and incremental loads and displacements to a common frame of reference. For each element of the gear, the incremental stiffness matrix is transformed from Local to Surface Coordinate System as described in Section 4.1.1.2.



MAIN STRUT DEFORMATION



INTERNAL LOADS IN ELEMENTS 2 AND 4

DEFINITIONS:

$a = \bar{X}_5 - \bar{X}_2$

$b = \bar{Y}_5 - \bar{Y}_2$

$c = \bar{Z}_5 - \bar{Z}_2$

$e = \bar{X}_5 - \bar{X}_4$

$f = \bar{Y}_5 - \bar{Y}_4$

$g = \bar{Z}_5 - \bar{Z}_4$

$d_{25} = d_{24} + d_{54}$

E_{B2} = BENDING MODULUS OF ELASTICITY (ELEMENT 2)

E_{B4} = BENDING MODULUS OF ELASTICITY (ELEMENT 4)

I_2 = MOMENT OF INERTIA (ELEMENT 2)

I_4 = MOMENT OF INERTIA (ELEMENT 4)

$$\delta = \frac{1}{d_{25}} \sqrt{(bg - cf)^2 + (ce - ag)^2 + (af - be)^2}$$

$$T = \frac{\delta}{\left[\dfrac{(d_{24})^3 d_{54}}{3} - \dfrac{(d_{24})^4 d_{54}}{3\,d_{25}}\right] / E_{B_2} I_2 + \left[\dfrac{(d_{24})^2 (d_{25})^2}{3} - (d_{24})^3 d_{25} + (d_{24})^4 - \dfrac{(d_{24})^5}{3\,d_{25}}\right] / E_{B_4} I_4}$$

REFERENCE POINT COORDINATES:

$\bar{X}_{RP} = \bar{X}_4 + (c^2e - acg - abf + b^2e)/(d_{25})^2$

$\bar{Y}_{RP} = \bar{Y}_4 + (a^2f - abe - bcg + c^2f)/(d_{25})^2$

$\bar{Z}_{RP} = \bar{Z}_4 + (b^2g - bcf - ace + a^2g)/(d_{25})^2$

TOTAL INTERNAL LOADS:

$F2p_{\boxed{2}} = d_{54}\,T = -F2q_{\boxed{2}}$

$F2p_{\boxed{4}} = d_{24}\,T = -F2q_{\boxed{4}}$

$M6q_{\boxed{2}} = M6q_{\boxed{4}} = d_{24}\,d_{54}\,T$

FIGURE 4-26 INTERNAL BENDING LOADS IN MAIN STRUT OF CANTILEVER GEAR

Before matrix assembly can be accomplished, the incremental stiffness matrix in the Surface Coordinate System for all elements of the gear must be of the same order. All elements of the inverted tripod gear are axial struts and hence have incremental stiffness matrices of the same order. For the cantilever gear, the incremental stiffness matrices of the axial struts (elements 1 and 3 in Figure 4-17) must be expanded to the row and column format of the bending strut stiffness matrices of elements 2 and 4. This is accomplished by adding rows and columns of zeros corresponding to the six rotational degrees of freedom.

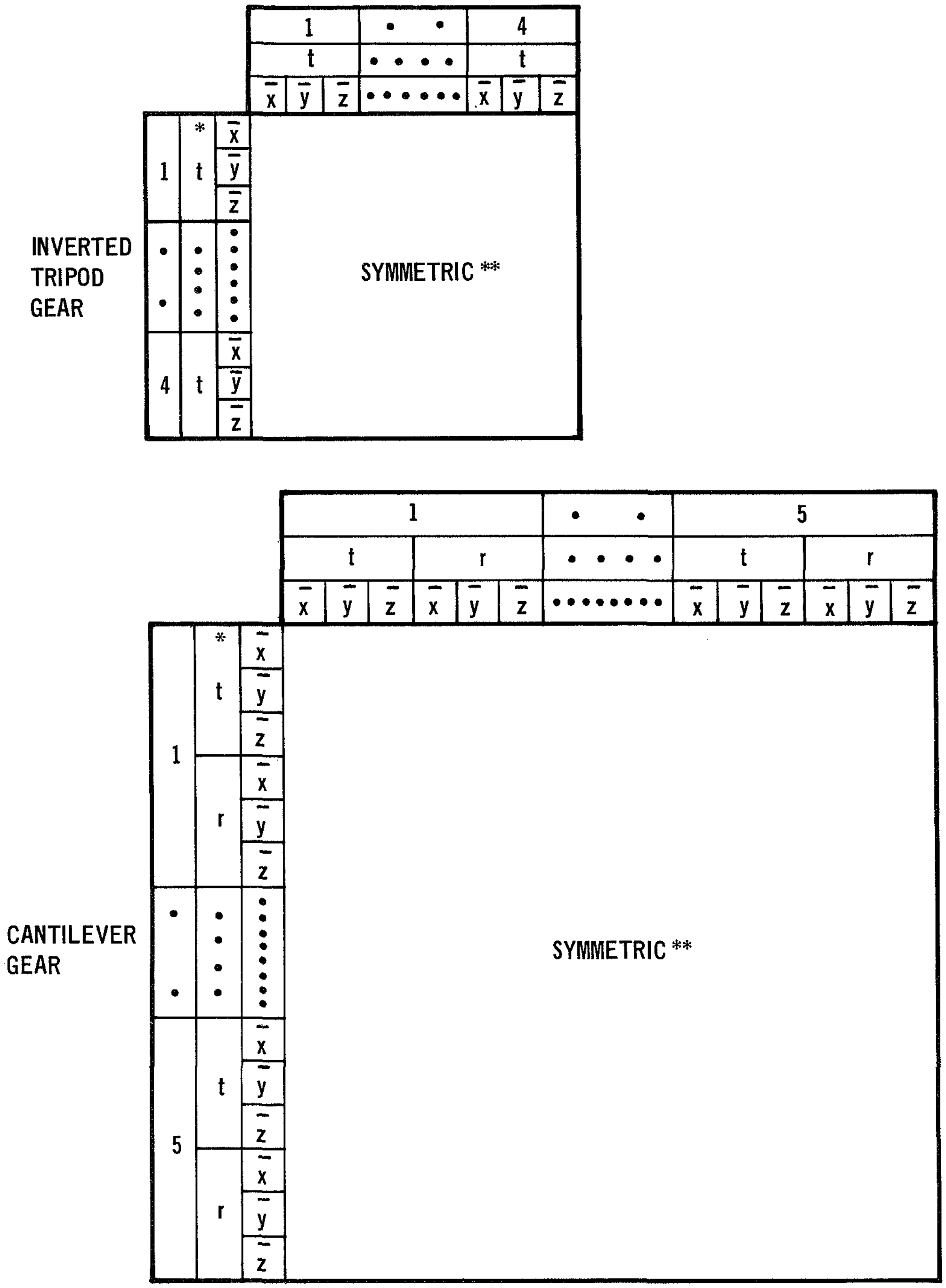
The total incremental stiffness matrix $[\overline{S}]$ (in the Surface Coordinate System) for the assembled gear is generated by systematically adding the transformed element stiffness matrices (according to the p and q nodes of the element) as shown in Figure 4-27. Node points on the inverted tripod gear are numbered consecutively from 1 to 4 (see Figure 4-17). The incremental stiffness matrix $[\overline{S}]$ of this gear is assembled with a row and column format corresponding to the three translational degrees of freedom in the Surface Coordinate System at each node in sequence. In this case, the size of the stiffness matrix is 12 by 12. For the cantilever gear, node points are numbered consecutively from 1 to 5. The incremental stiffness matrix $[\overline{S}]$ of this gear is assembled with a row and column format corresponding to the three translational followed by the three rotational degrees of freedom in the Surface Coordinate System at each node in sequence. In this case,the size of the stiffness matrix is 30 by 30.

The assembled incremental stiffness matrix $[\overline{S}]$ is related to the column matrices of incremental surface coordinate loads and displacements at each node by Equation (4-31).

$$\Delta\{\overline{F}\} = [\overline{S}]\ \Delta\{\overline{\delta}\} \tag{4-31}$$

For the fixed idealization of the inverted tripod gear (see Figure 4-17), nodes 1, 2, and 3 are pinned and thus have zero translational displacements. Since only translational displacements are included in the stiffness formulation for this gear, the degrees of freedom of interest are the translations of node 4. Therefore, for the inverted tripod, Equation (4-31) reduces to

$$\Delta\{\overline{F}'\} = [\overline{S}']\ \Delta\{\overline{\delta}'\} \tag{4-32}$$



* SYMBOLS t AND r INDICATE TRANSLATIONAL AND ROTATIONAL INCREMENTAL DISPLACEMENTS, RESPECTIVELY.

** TERMS IN THE MATRICES OF INDIVIDUAL ELEMENT INCREMENTAL STIFFNESSES IN THE SURFACE COORDINATE SYSTEM ARE SUCCESSIVELY ADDED INTO THE APPROPRIATE LOCATIONS IN THE FRAMEWORK OF THIS MATRIX.

FIGURE 4–27 ASSEMBLED INCREMENTAL STIFFNESS MATRIX FORMAT

In this equation, $\Delta\{\overline{F}'\}$ is a column matrix of three incremental force components at node 4; $\Delta\{\overline{\delta}'\}$ a column matrix of three incremental translations at node 4; and $[\overline{S}']$ a 3 by 3 matrix of stiffness terms, extracted from $[\overline{S}]$, corresponding to the degrees of freedom retained at node 4.

For the fixed idealization of the cantilever gear (see Figure 4-17), nodes 1, 2, and 3 are pinned and hence have zero translational displacements. Accordingly, equations associated with the 3 translational degrees of freedom at each of these nodes are not of interest. Elements 1 and 3 have zero bending capability. Therefore, the six equations associated with the rotational degrees of freedom at nodes 1 and 3 are not of interest. The remaining degrees of freedom for a cantilever gear are three rotations at node 2, three translations and three rotations at node 4, and three translations and three rotations at node 5. Thus, for the cantilever gear Equation (4-31) reduces to

$$\Delta\{\overline{F}''\} = [\overline{S}''] \, \Delta\{\overline{\delta}''\} \qquad (4\text{-}33)$$

In Equation (4-33), $\Delta\{\overline{F}''\}$ is a column matrix containing three incremental moment components at node 2, three incremental force components and three incremental moment components at node 4, and three incremental force components and three incremental moment components at node 5; $\Delta\{\overline{\delta}''\}$ a column matrix of the fifteen incremental displacements described above; and $[\overline{S}'']$ a 15 by 15 matrix of stiffness terms, extracted from $[\overline{S}]$, corresponding to the 15 degrees of freedom retained.

Equation (4-32) or (4-33) is solved repeatedly at each step of the modified incremental stiffness method (see Section 4.1.2.1). A least squares solution technique is programmed for solution of these equations. The independent variables for each of these solutions are discussed in Section 4.1.2.7. After the incremental nodal displacements have been determined, the updated nodal locations can be found as the sum of the previous nodal coordinates and the incremental nodal displacements. For the updated nodal locations, the total internal loads can then be determined as described in Section 4.1.2.5.

4.1.2.7 Applied Displacement Options - For each step of the modified incremental stiffness method, the governing matrix equation, Equation (4-32) for an inverted tripod gear or Equation (4-33) for a cantilever gear, is solved as outlined below. Both of these equations are analogous to Equation

(4-27) and the solution technique is the same as discussed in Section 4.1.2.1.

(i) Equation (4-32) (or Equation (4-33)) is solved once for the case where $\Delta\{\bar{\delta}'\}$ (or $\Delta\{\bar{\delta}''\}$), the total applied displacement vector divided by the number of incremental steps, is the (known) independent variable.

(ii) Equation (4-32) (or Equation (4-33)) is solved repeatedly for the case where $\Delta\{\bar{F}'\}$ (or $\Delta\{\bar{F}''\}$), the load unbalance vector, is the (known) independent variable. The load unbalance vector is calculated as the difference between applied external nodal loads and total internal nodal loads. Calculation of internal loads is described in Section 4.1.2.5.

Two options are available for specifying the total applied displacement vector discussed in (i) above. In the first option, total displacement of the footpad joint (node 4 for the inverted tripod gear and node 5 for cantilever gear) normal to a landing surface, and the coefficient of friction of the surface are specified. For example, suppose it is desired to investigate the displaced state of the gear at each of N steps for an applied normal displacement value of D units. For this case, the program determines N equilibrium configurations for the gear. For each of these configurations, the footpad joint is located in one of N different planes spaced D/N units apart. Whether the footpad slides on the landing surface as the footpad joint is displaced from the (n-1)st plane to the nth plane (n = 1, ..., N) depends on the friction coefficient. As explained in Section 4.1.2.3, the landing surface can be arbitrarily oriented with respect to the Lander Coordinate System.

In the second option available for specifying the applied displacement vector, three components of footpad joint displacement are specified in the Surface Coordinate System ($\bar{X}$, $\bar{Y}$, $\bar{Z}$). For example, if it is desired to investigate the displaced state of the gear at each of N steps when the applied footpad joint displacement components are $\bar{X}_f$, $\bar{Y}_f$, and $\bar{Z}_f$, the program determines N equilibrium configurations for the gear. In each of these configurations, the footpad joint is located at one of the positions $(\bar{X}_o + n\,\bar{X}_f/N,\ \bar{Y}_o + n\,\bar{Y}_f/N,\ \bar{Z}_o + n\,\bar{Z}_f/N)$ where n = 1, ..., N. The original position of the footpad joint (see Figure 4-17) is assumed to be ($\bar{X}_o$, $\bar{Y}_o$, $\bar{Z}_o$).

4.1.2.8 Energy Absorption - For each step of the landing gear displacement history, energy absorbed by the gear during the step is calculated in both Surface and Lander Coordinate System components.

Although an element representing the footpad is not included in the landing gear idealization, energy associated with crushing of footpad honeycomb can be included in the analysis. Footpad honeycomb attenuation is idealized with up to three levels of crushing in compression as shown in Figure 4-28.

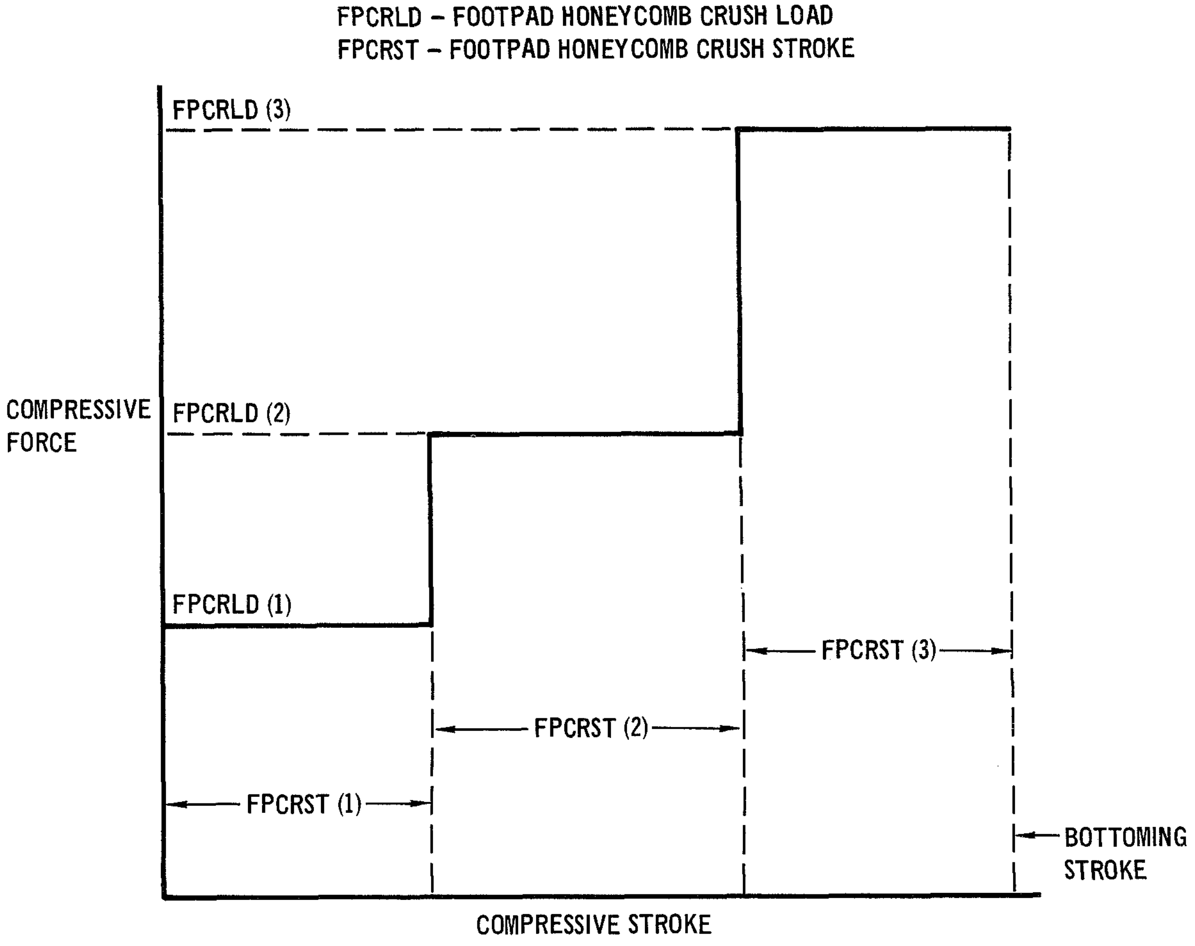


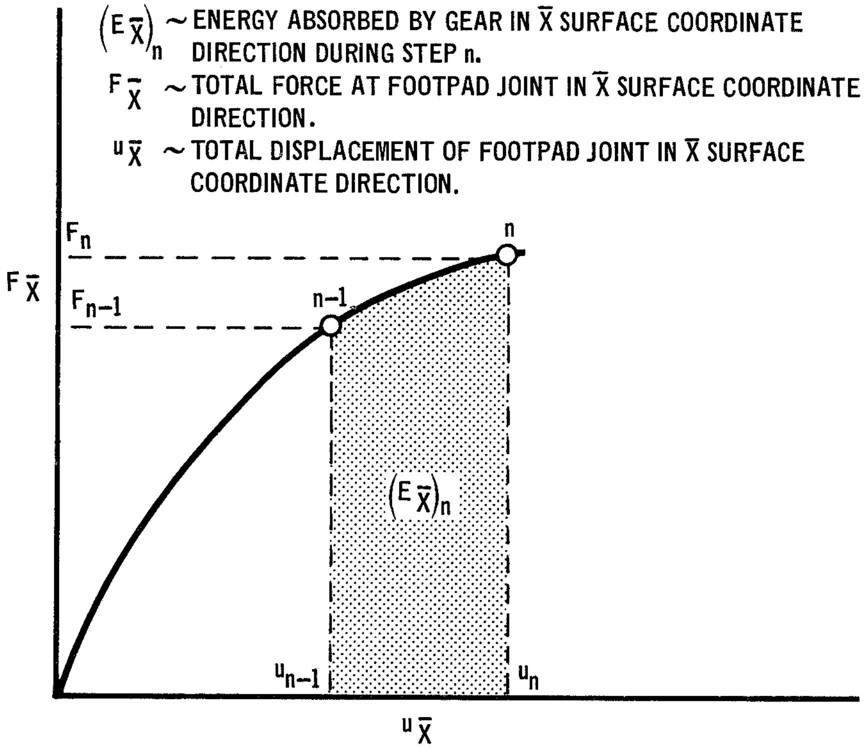
FIGURE 4-28 ASSUMED LOAD-STROKE CURVE FOR FOOTPAD HONEYCOMB

Each surface coordinate energy component absorbed at the footpad joint during a step is the sum of strut crush energy and footpad crush energy. Strut crush energy in a given surface coordinate direction is calculated as the product of the average force component (in that direction) over the step and the incremental displacement component (in the same direction) which occurred during the step. Figure 4-29 presents an example calculation of strut crush energy component absorbed in the $\overline{X}$ surface coordinate direction during the n^{th} incremental step. For a gear which has footpad attenuation with up to three levels of crush force, footpad crush energy may be added into the surface coordinate energy component in the direction normal to the landing surface. For a given incremental step, footpad crush energy is added into this energy component if the force component in the normal direction (see Figure 4-20) at the footpad joint exceeds one of the footpad honeycomb crush forces (Figure 4-28) for the first time. When this occurs, the associated level of honeycomb crush is assumed to have bottomed out during the step. Footpad crush energy component associated with this crushing is the product of crush load (FPCRLD) and the crush stroke (FPCRST). For example, the surface coordinate direction normal to the landing surface is $\overline{X}$ (Figure 4-20). If during incremental step n, the force in the $\overline{X}$ direction at the footpad joint exceeds the first crush force FPCRLD(1) in Figure 4-28, and this has not occurred for any previous step, the energy term FPCRLD(1) times FPCRST(1) will be added to the $\overline{X}$ strut crush energy component for this step.

Components of energy absorption during a given incremental step are obtained similarly in Lander Coordinate System components. In both coordinate systems, total energy absorption components are determined after n incremental steps as the sum of the energy components obtained in each of the n steps taken.

4.2 Program Description - Organization of the Structural Analysis Program is described in this section. Included are flow diagrams and discussions of subroutine functions for both the Center Body Option and Landing Gear Option.

4.2.1 Subroutines - The Structural Analysis Program is divided into nine Fortran OVERLAY segments. OVERLAY organization and description of the function of each subroutine are shown in Figure 4-30. This organization is required to stay within the allotted core storage requirements. The main OVERLAY (0, 0) acts as the principal executive routine and determines the order in which primary OVERLAYs (1, 0) through (5, 0) are called. OVERLAY (1, 0) processes the data set header cards and handles the data sets and initialization relating to the Center Body Option. OVERLAY (2, 0) generates the total stiffness matrix for the Center Body Option. OVERLAY (3, 0) determines the displacements and rotations for the Center Body Option and prints the results. The modal analysis portion of the Center Body Option is handled in OVERLAY (4, 0). OVERLAY (5, 0) contains the routines for the Landing Gear Option and has three supporting secondary overlays; OVERLAY (5, 1) which reads the landing gear data cases, OVERLAY (5, 2) which contains the executive routine for the inverted tripod gear, and OVERLAY (5, 3) which contains the executive routine for the cantilever gear.



FOR STEP n:

$(E_{\bar{X}})_n$ = AREA UNDER CURVE BETWEEN POINTS n-1 AND n

$$= \frac{(F_n + F_{n-1})}{2} (u_n - u_{n-1})$$

FIGURE 4-29 EXAMPLE CALCULATION OF STRUT CRUSH ENERGY

OVERLAY	SUBROUTINE	FUNCTION
(0, 0)	MAIN	PROGRAM EXECUTIVE ROUTINE
	ERPNT1	ERROR PROCESSING ROUTINE FOR ALL OVERLAYS
	ERPNT2	SECOND ENTRY POINT TO ERPNT1
	WRSTRK	SPARSE MATRIX PRINT ROUTINE
(1, 0)	INITAL	EXECUTIVE ROUTINE
	DATSET	HEADER CARD PROCESSOR AND CENTER BODY OPTION DATA CASE READER
	RDDATA	CENTER BODY OPTION DATA CASE PROCESSOR
	SHRPAN	CONVERT SHEAR PANELS TO EQUIVALENT DIAGONAL BAR ELEMENTS
	CROSSX	CALCULATE THE CROSS PRODUCT OR DOT PRODUCT OF TWO VECTORS
(2, 0)	STIFF	EXECUTIVE ROUTINE
	SETSTF	INITIALIZE STRUCTURAL STIFFNESS MATRIX STORAGE ARRAY IN A SPARSE BLOCKED FORMAT
	STFTRN	CALCULATE ELEMENT STIFFNESS AND TRANSFORMATION MATRICES
	STORSM	CONSTRUCT ASSEMBLED STIFFNESS MATRIX
	TRASMK	TRANSFORM ELEMENT STIFFNESS MATRICES FROM LOCAL TO GLOBAL COORDINATE SYSTEM
	WRBDAT	PRINT ELEMENT STIFFNESS AND TRANSFORMATION MATRICES
	BRSTRA	SECOND ENTRY POINT TO WRBDAT, PRINTS TRANSFORMED ELEMENT STIFFNESS MATRICES
	WRSTDK	WRITE ELEMENT STIFFNESS AND TRANSFORMATION MATRICES ON FILE 2
(3, 0)	FINIAL	EXECUTIVE ROUTINE
	PANDTK	PRINT AND STORE ON FILE 9 THE ASSEMBLED STIFFNESS MATRIX SET-UP FOR SOLUTION
	SOLVE	DISPLACEMENTS AND ROTATIONS DETERMINED USING ITERATIVE METHOD. GLOBAL FORCES AND MOMENTS ON JOINTS CALCULATED
	PNTFMV	PRINT DISPLACEMENTS, ROTATIONS, AND GLOBAL FORCES AND MOMENTS
	FMBARS	ELEMENT BAR FORCES AND MOMENTS CALCULATED USING RESULTS OF SOLVE
(4, 0)	NLMDAL	EXECUTIVE ROUTINE
	PUNPACK	DATA HANDLING ROUTINE FOR SUBROUTINE REDUCE
	REDUCE	PRINT ASSEMBLED STIFFNESS MATRIX AND REDUCE TO DESIRED SIZE (REDUCED MATRIX 102 x 102 OR LESS)
	STFMAS	CREATES MATRIX SYSTEM, USING INPUT DIAGONAL MASS MATRIX AND REDUCED STIFFNESS MATRIX, FOR WHICH EIGENVALUES CAN BE FOUND
	FNALEV	EIGENVECTORS OF REDUCED SYSTEM TRANSFORMED INTO EIGENVECTORS OF FULL SYSTEM
	EEPNT	WRITE EIGENVALUES AND EIGENVECTORS
	TRIDIA	MATRIX SYSTEM IS TRI-DIAGONALIZED USING HOUSEHOLDER'S METHOD
	EIGVAL	SPECIFIED NUMBER OF SMALLEST EIGENVALUES OF MATRIX SYSTEM ARE CALCULATED USING ORTEGA'S METHOD (PLUS SIX RIGID BODY NATURAL FREQUENCIES OF ZERO)
	EIGVEC	CALCULATES EIGENVECTORS ASSOCIATED WITH LOWEST EIGENVALUES USING WILKINSON'S METHOD
	PRNT1	MODAL DATA OUTPUT ROUTINE FOR REDUCED SYSTEM
	PRNT2	SECOND ENTRY POINT OF PRNT1 TO OUTPUT MODAL DATA OF FULL SYSTEM
	P1A721	PRINT SUPPORT ROUTINE FOR PRNT1 AND PRNT2
(5, 0)	GEAREX	EXECUTIVE ROUTINE
	OUTPUT	LANDING GEAR OPTION OUTPUT
	ENERGY	COMPUTE AND SUM ENERGY QUANTITIES
	TRNFSM	TRANSFORM ELEMENT STIFFNESS MATRIX
	TRALMG	TRANSFORM FORCE-MOMENTS FROM LOCAL TO GLOBAL COORDINATES
	STFMIT	COMPUTE NON-BENDING ELEMENT STIFFNESS MATRIX
	TRANSM	COMPUTE ELEMENT LENGTH AND TRANSFORMATION MATRIX
	DMFSS	DETERMINE THE RANK OF A SYMMETRIC POSITIVE SEMI-DEFINITE MATRIX
	DMLSS	CALCULATE THE LEAST SQUARES SOLUTION OF A SYSTEM OF SIMULTANEOUS EQUATIONS WITH SYMMETRIC POSITIVE SEMI-DEFINITE COEFFICIENT MATRIX WHOSE RANK IS KNOWN
	STRUT	COMPUTE ELEMENT AXIAL LOAD AND STIFFNESS
(5, 1)	INPUT	READ LANDING GEAR OPTION DATA CASES
	GEOM	COMPUTE LANDER COORDINATE TO SURFACE COORDINATE TRANSFORMATION MATRIX FROM EULER ANGLES
(5, 2)	INVTRP	EXECUTE INVERTED TRIPOD DATA COMMANDS
(5, 3)	CANTIL	EXECUTE CANTILEVER DATA COMMANDS
	BNDLDS	COMPUTE BENDING LOADS FOR CANTILEVER GEAR
	STFMCF	COMPUTE BENDING ELEMENT STIFFNESS MATRIX

FIGURE 4–30 STRUCTURAL ANALYSIS PROGRAM SUBROUTINES

4.2.2 Flow Diagram - The Structural Analysis Program is shown schematically in Figure 4-31. While detailed steps, such as those required in the iteration loop of the Center Body Option, are not presented, the basic sequence of events is shown for both options of the program. A listing of the program is given in Appendix H.

4.3 Program Operation - Information necessary to operate both options of the Structural Analysis Program is contained in this section.

4.3.1 Center Body Option - This section includes definition of input requirements and format, and output interpretation for the Center Body Option of the program. Examples of input and output data for a typical problem are contained in Appendices A and B.

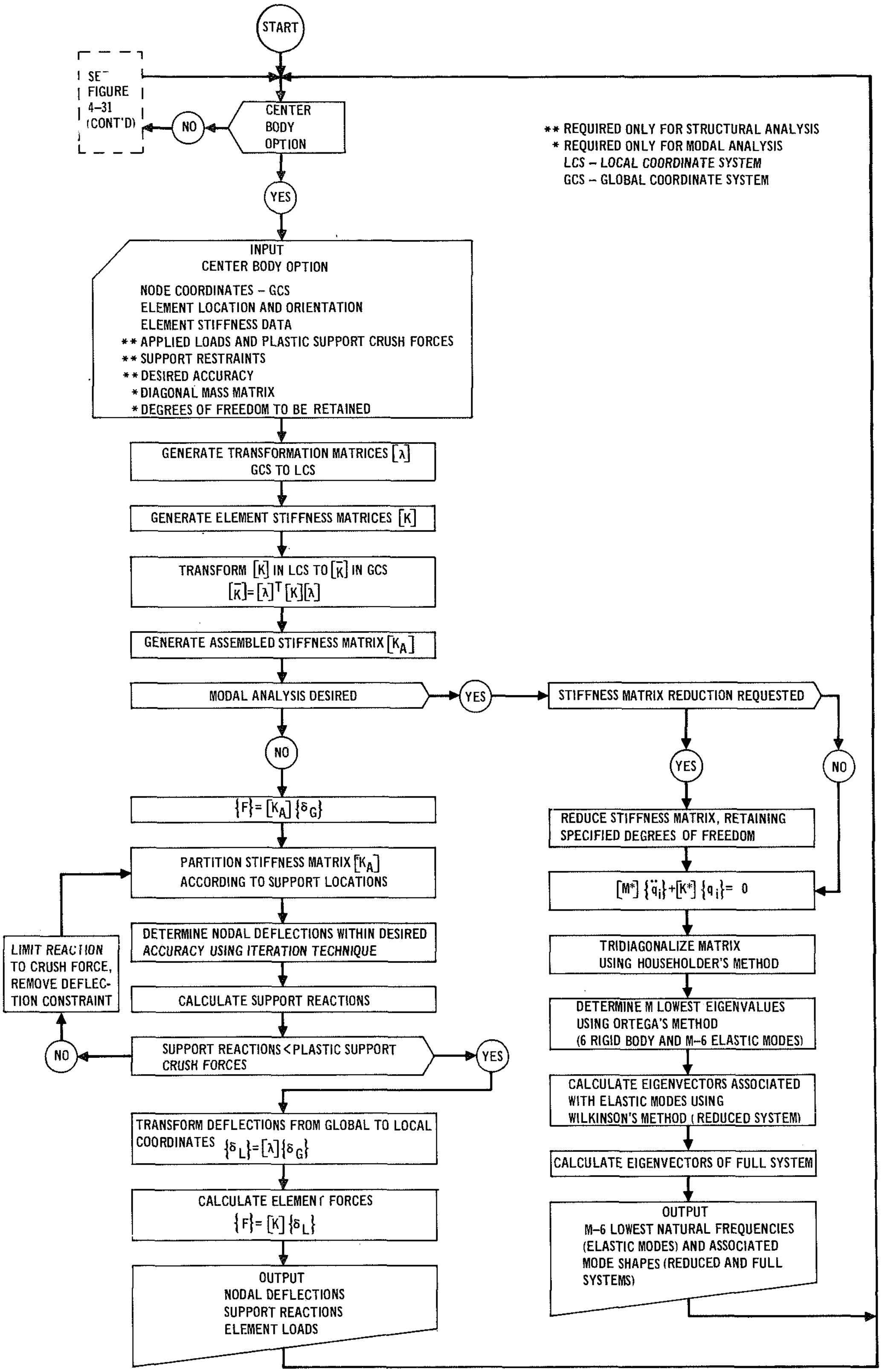
4.3.1.1 Input Data - For each Center Body Option data set, a data case header card and information describing geometry, supports, applied loads and displacements, member properties, and indicators needed to control program operation are required as input data. Multiple data cases may be run by stacking data sets. Each data set must consist of the following:

Header Card
Data Cards
Data Option Indicator (NAMELIST) Card(s)

A. Header Card - The first card of a data set must be a header card. This card indicates that the Center Body Option will be employed and must contain the characters STRUCTURE in columns 1 through 9 as indicated in Figure 4-32.

B. Data Cards - The input format for data cards employs a system of code numbers, located in column 1, to identify the type of data being read. The nine code numbers and corresponding type of information being input are as follows:

(0) Comment cards
(1) Joint information cards
(2) Reference point information cards
(3) Force and moment limits or specified displacement and rotation cards
(4) Specified force vector cards
(5) Specified moment vector cards
(6) Bar information cards
(7) Rectangular shear web information cards
(8) Data terminator card



** REQUIRED ONLY FOR STRUCTURAL ANALYSIS
* REQUIRED ONLY FOR MODAL ANALYSIS
LCS – LOCAL COORDINATE SYSTEM
GCS – GLOBAL COORDINATE SYSTEM

FIGURE 4–31 FLOW DIAGRAM STRUCTURAL ANALYSIS PROGRAM

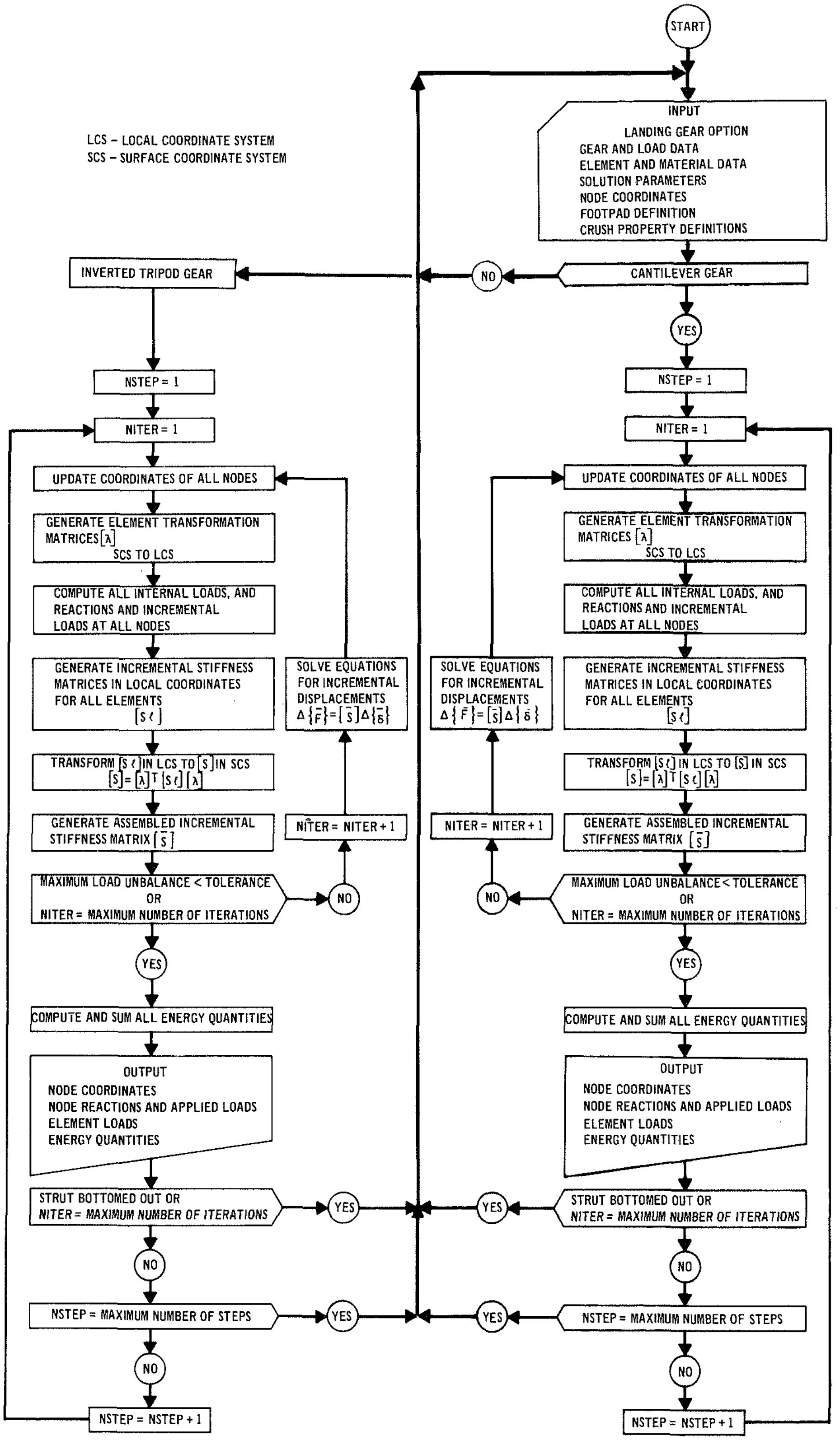


FIGURE 4–31 FLOW DIAGRAM STRUCTURAL ANALYSIS PROGRAM (Continued)

B.0 Comment Cards - As many comment cards as desired may be input by leaving columns 1 through 6 blank and entering the comments in columns 7 through 80. These comment cards may be located anywhere within the data cards as long as they precede the data terminator.

B.1 Joint Information Cards - Joint information is input on data cards in which a 1 is placed in column 1. Location of joint data in the fields of the card is indicated in Figure 4-32. Joints must be numbered sequentially (right-justified in columns 2 through 5) from 1 through the total number of joints (maximum of 74). The global coordinates, $\overline{X}$, $\overline{Y}$, and $\overline{Z}$, may be any right hand orthogonal coordinate system (see Figure 4-2). If the structure is planar, it will be simpler to choose a Global Coordinate System such that two of the axes lie in the plane of the structure. Advantage should be taken of any structural symmetry which may exist in selecting global axes.

Six data fields located between columns 36 and 53 are used to indicate various constraints at a particular joint. These constraints may be combinations of specified displacements (global deflections and/or rotations) and limits on reaction loads (global forces and/or moments) at a joint. Leaving these columns blank causes the program to assume that there are no displacement constraints at a joint; however, there are applied loads at the joint. These applied loads are discussed in the next paragraph. Constraints are defined by inserting nonzero integer identifying indicators (right-justified) in the fields of this region. The absolute value of these identifying indicators corresponds to the limit number on the code 3 data card defining the magnitudes of the particular constraint. When positive identifying indicators are used, the program assumes that upper and lower plastic limits are placed on the appropriate loads at a support and that the corresponding displacement or rotation is zero as defined on the related code 3 data card. For negative identifying indicators, the program assumes that no restraints are placed on the loads, but the displacements are specified on the related code 3 data card. The same code 3 data card may be used to define identical constraints at a number of joints. The maximum number of plastic loads and specified displacements is 88.

Applied forces are indicated by placing a separate set of integer identifying indicators in columns 54 through 57 (right-justified). The program looks for the three global components of force on cards with code number 4 in

(Header Card and Card Codes 0, 1, and 2)

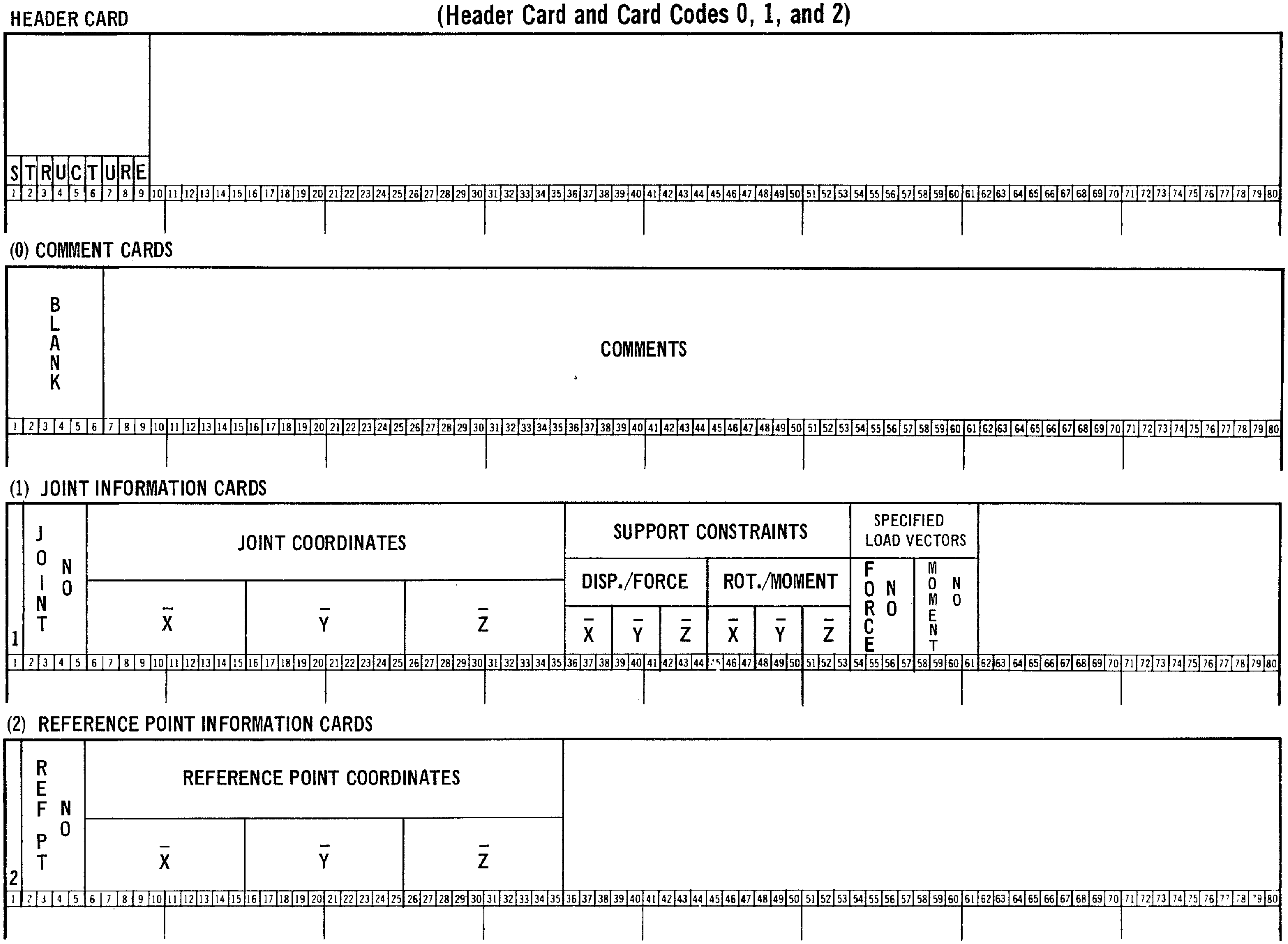


FIGURE 4-32 INPUT DATA FORMAT – CENTER BODY OPTION

column 1, and the corresponding identifying indicator in columns 2 through 5. Identifying force indicators must be sequential with a maximum of 74. Applied moments are numbered separately, with this set of identifying indicators being placed in columns 58 through 61, and the three corresponding global moment components being located on cards with a 5 in column 1. A total of 74 identifying moment indicators may be used. If either of these identifying indicators is zero (or blank), the appropriate applied load is zero and no corresponding code 4 or 5 data card is required. If displacements and/or rotations have already been restrained in specific directions, forces and/or moments may not be applied in these same directions.

Since free-free modes are obtained with the modal analysis routine, no constraints or applied loads should be specified when employing this program option. Thus, for modal analysis data cases, columns 36 through 61 should be left blank on all code 1 data cards.

B.2 Reference Point Information Cards - A code number 2 in column 1 indicates the data represents global coordinates and identifying joint number of a reference point. Reference points are provided to permit orientation of bar local coordinate (bending) axes, where structural joints will not suffice. Numbering of these reference points begins with one greater than the number of joints and is sequential through the total number of reference points. For example if there are 36 structural joints, the first reference point must be 37.

B.3 Force and Moment Limit Cards - A 3 in column 1 indicates data on the card is either a known displacement or rotation (columns 26 through 35), or upper and lower plastic force or moment constraints (columns 6 through 15 for an upper bound and 16 through 25 for a lower bound). The integer identifying number (limit number) in columns 2 through 5 corresponds to an identifying indicator or indicators on code 1 data cards, previously discussed. These limit numbers must be sequential in order. The sign of the identifying indicator on the code 1 data card determines what data are being read: A plastic force constraint and zero displacement (positive sign) or known displacement (negative sign). The known displacement may be nonzero or zero, simulating a conventional support. If the force is to be limited in the positive global direction, use the upper limit. A lower limit indicates a constraint on the

(Data Codes 3 Thru 6)

(3) FORCE/MOMENT LIMITS AND SPECIFIED DISPLACEMENT/ROTATION CARDS

		LIMITS ON FORCE/MOMENT		SPECIFIED DISP/ROT VALUE	
3	LIMIT NO	UPPER LIMIT	LOWER LIMIT		
1	2–5	6–15	16–25	26–35	36–80

(4) SPECIFIED FORCE VECTOR CARDS

		SPECIFIED FORCE COMPONENTS			
4	FORCE NO	$F_{\bar{X}}$	$F_{\bar{Y}}$	$F_{\bar{Z}}$	
1	2–5	6–15	16–25	26–35	36–80

(5) SPECIFIED MOMENT VECTOR CARDS

		SPECIFIED MOMENT COMPONENTS			
5	MOMENT NO	$M_{\bar{X}}$	$M_{\bar{Y}}$	$M_{\bar{Z}}$	
1	2–5	6–15	16–25	26–35	36–80

(6) BAR INFORMATION CARDS

		BAR LOCAL AXIS LOCATORS			OPTION	INDICES		BAR CROSS-SECTIONAL AREA	BAR MOMENTS OF INERTIA			BAR MATERIAL MODULI		SHEAR FORM FACTORS	
6	BAR NO.	P	Q	R	K	I_T	I_N	A	I_N	I_T	J	E	G	K_N	K_T
1	2–5	6–8	9–11	12–14	15	16	17	18–25	26–34	35–43	44–52	53–61	62–70	71–75	76–80

FIGURE 4–33 INPUT DATA FORMAT – CENTER BODY OPTION

force in the negative global direction.

For modal analysis data cases, code 3 data cards should not be input.

B.4 Specified Force Vector Cards - Applied forces at a joint are input as three or less global force components on cards with a 4 in column 1. The identifying number (force number), right-justified in columns 2 through 5, corresponds to an identifying indicator on code 1 data cards thereby defining the joint or joints where forces will be applied.

For modal analysis data cases, code 4 data cards should not be input.

B.5 Specified Moment Vector Cards - Applied moments at a joint are input as three or less global moment components on cards with a 5 in column 1. The identifying number (moment number), right-justified in columns 2 through 5, corresponds to an identifying indicator on code 1 data cards to define the joint or joints where the moments will be applied.

Code 5 data cards should not be input for modal analysis data cases.

B.6 Bar Information Cards - A card with a 6 in column 1 indicates bar data are being input. Bar number is entered in columns 2 through 5, right-justified, with numbering being sequential from one through the total number of bars (130 maximum). Bar origin (point "p"), end (point "q"), and direction of the Y_ℓ bending axis (point "r"), locating the bar Local Coordinate System (Figure 4-1), are determined by specifying joint numbers in the appropriate locations in columns 6 through 14 (all right-justified). A reference point may be used for point "r," but points "p" and "q" must be nodes (joints) on the structure. Point "r" must not be on the line "pq" or its extension.

Bar cross-sectional area perpendicular to the "pq" (X_ℓ) axis, A; moment of inertia about the Z_ℓ axis, I_N; moment of inertia about the Y_ℓ axis, I_T; and torsional constant, J, are entered in the appropriate columns as indicated in Figure 4-33. Members pinned at both ends for bending about the Z_ℓ axis can be simulated by making the moment of inertia I_N equal to zero. Similarly making I_T equal to zero simulates a member pinned at both ends for bending about the Y_ℓ axis. If A is set equal to zero, no axial load will be carried by the member, and setting J equal to zero prevents the member from carrying torsion. Any combination of these may be utilized. Modulus of elasticity, E, and shear modulus, G, for the bar material are entered in columns 53 through 61 and 62 through 70, respectively. If E-format is used it must be right-justified.

Caution must be exercised when using pinned members or members with A, I_T, I_N, or J equal to zero, because a joint may be left with no load carrying capability in one of the three global directions or with no rotational restraint about one of the three global axes. An example of this would be two bars in the same plane pinned together. An artificial restraint (support) must be placed on the joint in the global direction in which there is no load carrying capability, even though there will be no force induced in this support direction. This instruction is input on code 1 data cards. The program will automatically reject any problem in which load carrying capability (translational or rotational) does not exist at any joint.

The program has the ability to account for the effect of bar shear strain on bending deflections. If a zero (blank) is placed in column 15, shear strain is not accounted for. A 1 in column 15 causes the program to read the values of shear form factors, K_N and K_T, and account for the effect of shear strain. The factor K_N is the total cross-sectional area divided by the area effective in carrying shear in the Y_ℓ direction, while K_T is the total area divided by the area effective in carrying shear in the Z_ℓ direction (see Figure 4-6).

The program can also idealize a bar pinned at one end and rigidly attached at the other. To exercise this option a zero must be placed in column 15. If the rotational release is to be for bending about the Y_ℓ axis (I_T), a 1 or 2 is placed in column 16. A 1 indicates the "p" end is pinned while a 2 indicates the "q" end is pinned. The same number code (1 - "p" end pinned, 2 - "q" end pinned) is used in column 17 if the rotational release is for bending about the Z_ℓ axis (I_N). Values inserted for K_N and K_T for these cases will be automatically utilized.

B.7 <u>Rectangular Shear Web Information Cards</u> - Shear web information is input on data cards in which a 7 is placed in column 1. Location of shear web data in the fields of the card is illustrated in Figure 4-34. Shear webs must be numbered sequentially from 1 through the total number of shear webs (maximum of 30). The integer representing the shear web number is right-justified in columns 2 through 5.

(Card Codes 7 and 8 and Data Option Indicator Card(s))

(7) RECTANGULAR SHEAR WEB INFORMATION CARDS

			SHEAR WEB JOINTS				SHEAR WEB MATERIAL PROPERTIES			
7	SHEAR	WEB NO	A	B	C	D	ν	E	t	
1	2 3	4 5	6 7 8	9 10 11	12 13 14	15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32 33 34 35 36 37	38 39 40 41 42 43 44 45 46 47	48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

(8) DATA TERMINATOR CARD

8	BLANK	COMMENTS
1	2 3 4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

($INDATA) DATA OPTION INDICATOR CARD(S)

			DATA OPTION INDICATORS (SEE FIGURE 4–35)		
BLANK	$INDATA	BLANK	NAME 1 = XXX, NAME 2 = YYY, ...	$END	
1	2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68	69 70 71 72	73 74 75 76 77 78 79 80

FIGURE 4–34 INPUT DATA FORMAT – CENTER BODY OPTION

Four data fields located between columns 6 and 17 are used to specify the joints A, B, C, and D, numbered counterclockwise (see Figure 4-7), which represent the corners of the rectangular web. These integers must be right-justified in their respective fields. Care must be taken to insure that the joints A, B, C, and D define a plane and form a rectangle. The program will reject any problem in which this is not true.

Three data fields located between columns 18 and 47 are used to specify the material properties and thickness of the web. The Poisson's ratio, ν, modulus of elasticity, E, and thickness, t, of the web are input in these fields as indicated in Figure 4-34. These values must be right-justified in their fields if E-Format is used.

B.8 Data Terminator Card - After all data needed on code 1 through 7 data cards and the desired comment cards are input, a data terminator card is required to signify the end of the data. This card must contain an 8 in column 1 and may contain any desired comments in columns 7 through 80.

C. Data Option Indicator (NAMELIST) Card(s) - Following the previously described input data, a NAMELIST system of input indicated by $INDATA is used to define option indicator cards for selection of output options, iteration method, modal analysis routine, program termination, and tape options. Figure 4-34 presents the card format to be used. $INDATA is entered in columns 2 through 8 of the first data option card. Columns 10 through 68 on the first and any succeeding cards are used to specify optional data indicator values. $END must be entered in columns 69 through 72 of the last data option card.

Data option indicators and their nominal and optional values are defined in Figure 4-35. If nominal values are acceptable for all NAMELIST items, $INDATA is entered in columns 2 through 8, $END is entered in columns 69 through 72, and the rest of the card left blank. By listing any of the option data indicators, such as INDNMA = 1, the nominal control is overridden, as in this case where normal mode analysis would be utilized. A data option indicator need be listed only if a value other than the nominal value is desired.

INDICATOR	NOMINAL VALUE	OPTIONAL VALUE	CONTROL FUNCTION
INDSFL	1		DO NOT } WRITE BAR LOCAL STIFFNESS AND TRANSFORMATION MATRICES.
		0	DO
INDSFG	1		DO NOT } WRITE BAR TRANSFORMED STIFFNESS MATRICES
		0	DO
IND1SL	0		WILL NOT } INPUT AN INITIAL SOLUTION OF DISPLACEMENTS AND ROTATIONS IN "SOLVEC",
		1	WILL
INDITR	3X(NO. OF ROWS IN STRUCTURAL STIFFNESS MATRIX)	K (ANY INTEGER)	THE ITERATIVE SOLUTION OF THE SIMULTANEOUS EQUATIONS WILL BE STOPPED AFTER ("INDITR" + 1) ITERATIONS
ERRTOL	.0001	A (ANY REAL NUMBER)	THE ABSOLUTE VALUE OF "ERRTOL" IS USED AS THE ERROR TOLERANCE TO TERMINATE THE ITERATIVE SOLUTION FOR DISPLACEMENTS AND ROTATIONS.
SOLVEC	$0._{(i)}$		THE INITIAL SOLUTION OF DISPLACEMENTS AND ROTATIONS IS ZERO.
		$A_{(i)}$	THE A i's (i = 1, 6X(NO. JOINTS)) ARE INPUT VALUES (SEPARATED BY COMMAS) DEFINING AN INITIAL SOLUTION OF DISPLACEMENTS AND ROTATIONS. (SOLVEC IS DIMENSIONED 444) ie. $A_{(1)} = \bar{X}$ DISPLACEMENT OF JOINT 1 . . $A_{(4)} = \bar{X}$ ROTATION OF JOINT 1 . . $A((j-1)X6+1) = \bar{X}$ DISPLACEMENT OF JOINT j . . $A((j-1)X6+4) = \bar{X}$ ROTATION OF JOINT j.

FIGURE 4–35 NAMELIST ($INDATA) DATA OPTION INDICATORS

INDICATOR	NOMINAL VALUE	OPTIONAL VALUE	CONTROL FUNCTION
RELAXF	1.		EMPLOY GAUSS-SIEDEL SOLUTION METHOD
		A (1.0 $\leq$ A $\leq$ 2.0)	THE RELAXATION FACTOR EMPLOYED IF THE OVERRELAXATION METHOD OF SOLUTION IS USED. RECOMMENDED OPTIONAL VALUE = 1.2
INDRLX	2		EMPLOY CONJUGATE GRADIENT SOLUTION TECHNIQUE.
		1	EMPLOY GAUSS-SIEDEL METHOD OF SOLUTION WITH NOMINAL RELAXATION FACTOR OF 1., OR USE OVERRELAXATION SOLUTION METHOD WITH OPTIONAL RELAXATION FACTOR DEFINED IN "RELAXF."
		0	EMPLOY GAUSS-SIEDEL SOLUTION METHOD WITH AITKEN'S Δ^2 IMPROVEMENTS.
INDRKT	0		DO NOT — READ STRUCTURAL STIFFNESS MATRIX, AND BAR LOCAL STIFFNESS AND TRANSFORMATION MATRICES FROM PHYSICAL TAPE 9.
		1	DO
INDWKT	0		DO NOT — WRITE STRUCTURAL STIFFNESS MATRIX AND BAR LOCAL STIFFNESS AND TRANSFORMATION MATRICES ON PHYSICAL TAPE 9.
		1	DO
INDPLS	0		DO NOT — CONSIDER PLASTICITY THROUGH USE OF LIMITS ON FORCES/MOMENTS.
		1	DO
MINRST	6	K (ANY INTEGER)	LESS THAN "MINRST" RESTRAINTS WILL CAUSE PROGRAM TERMINATION DURING SOLUTION OF SIMULTANEOUS EQUATIONS.
INDNMA	0		DO NOT — RUN NORMAL MODE ANALYSIS.
		1	DO

FIGURE 4–35 NAMELIST ($INDATA) DATA OPTION INDICATORS (Continued)

INDICATOR	NOMINAL VALUE	OPTIONAL VALUE	CONTROL FUNCTION
IRWKP	$0_{(i)}$	K(i) (INTEGERS)	ROW NUMBERS IN ASCENDING ORDER (SEPARATED BY COMMA'S) TO BE KEPT IN REDUCED STIFFNESS MATRIX WHEN RUNNING NORMAL MODE ANALYSIS ($i = 1, 2 \ldots , n$ WHERE $n \leq 102$; "IRWKP" IS DIMENSIONED 103).
AMASS	$0._{(i)}$	A_i	DIAGONAL MASS MATRIX ELEMENTS (SEPARATED BY COMMAS) IN ASCENDING ORDER, INPUT WHEN RUNNING NORMAL MODE ANALYSIS ($i = 1, 2, \ldots , n$ WHERE $n \leq 102$; "AMASS" IS DIMENSIONED 102).
IREDTO	ORDER OF STIFFNESS MATRIX	K ($K \leq 102$)	THE REQUIRED ORDER OF REDUCED STIFFNESS MATRIX WHEN RUNNING NORMAL MODE ANALYSIS.
INDWNM	0	1	DO NOT DO WRITE NORMAL MODE DATA ON TAPE FOR USE IN TASK ORDER TWO LANDING LOADS AND MOTIONS PROGRAM OR THE CENTER BODY LOADS PROGRAM.
TMAX	9999.	A (ANY REAL NUMBER)	NUMBER OF SECONDS AFTER WHICH MACHINE WILL PRINT PRESENT SOLUTION AND STOP, IF CONVERGENCE TO PRESCRIBED TOLERANCE HAS NOT BEEN REACHED (RECOMMENDED OPTIONAL VALUE = .9X (CP TIME)).
ISFDIM	12,904	K (INTEGER)	THE NUMBER OF WORDS OF STORAGE RESERVED TO STORE THE TOTAL STIFFNESS MATRIX. THE LENGTH OF LABELED COMMON, "CMAIN", MUST BE RE-DEFINED TO ALLOW STORAGE OF A STIFFNESS MATRIX WHICH REQUIRES "K" WORDS OF STORAGE SPACE, WHERE $K > 12904$.
INDWTS	0	1	DO NOT DO PRINT THE NONZERO ELEMENTS OF THE TOTAL STIFFNESS MATRIX.
NEIGVL	5	M ($5 \leq M \leq$ IREDTO-6)	NUMBER OF NON-RIGID BODY MODES TO BE CALCULATED IF MODAL ANALYSIS IS EMPLOYED, "NEIGVL" MUST BE A MULTIPLE OF 5.

FIGURE 4–35 NAMELIST ($INDATA) DATA OPTION INDICATORS (Continued)

The ERRTOL control is important, since it affects computer time. The value specified, multiplied by 100, is the maximum acceptable percent error in the solution. The error for the Gauss-Siedel method, the Gauss-Siedel method with Aitken's Delta Squared improvements, and the overrelaxation method is defined as the largest relative difference between consecutive displacement and rotation solutions. This relative difference is calculated by taking the difference between given solution values for two successive iterations and dividing by the last value of the solution.

The error for the Conjugate Gradient Method is defined as the largest term found when the residual vector (see Reference (5)) is divided by the largest element in the force/moment vector. The program will run until it achieves the accuracy prescribed by ERRTOL or exceeds the maximum number of iterations (INDITR).

An initial set of nodal deflections may be input using the indicator SOLVEC. This option allows the program to start with these values for deflections and achieve convergence more rapidly. This option is also useful if minor changes are made in a structure after an initial solution has been obtained.

The maximum number of iterations allowed (INDITR), nominally three times the number of rows in the stiffness matrix, should be selected based on the size and sparsity of the stiffness matrix and desired length of computer run time. The nominal iteration method used is the Conjugate Gradient Method. If INDRLX is set equal to 0 the Gauss-Siedel method with Aitken's Delta Squared Improvements is employed. If INDRLX is set equal to 1, the Overrelaxation Method is employed and a value of RELAXF (the relaxation factor) between 1.0 and 2.0 should then be input. A nominal relaxation factor of 1.0 is set in the program, which is equivalent to a standard Gauss-Siedel solution (without Aitken's Delta Squared Improvements). For employing the Overrelaxation Method, a RELAXF equal to 1.2 has been found to result in minimum machine time and rapid convergence for a number of typical problems. Experience dictates that for highly redundant space frames, such as the legged lander, the nominal solution technique (the Conjugate Gradient Method) is the best suited of the above methods and is recommended.

The MINRST option is selected based on the minimum number of restraints (supports) which must exist for stability (nominally six). If, at any time, less than this number of supports exist, the program will automatically terminate with an error message.

If it is desired to perform a normal mode analysis, INDNMA is set equal to 1. If reduction is to be employed, the required order of the reduced stiffness matrix is input in IREDTO. The row numbers to be retained in the reduced stiffness matrix must be listed in ascending order in IRWKP if reduction is employed. Values for the diagonal mass matrix terms associated with the degrees of freedom retained in the reduced stiffness matrix, are input in AMASS. If all six degrees of freedom at a particular joint are retained, the first three numbers in AMASS would represent the mass associated with this joint. The next three numbers would represent the mass moment of inertia associated with this joint relative to the directions of the global $\bar{X}$, $\bar{Y}$ and $\bar{Z}$ axes. If degrees of freedom at a joint are removed in the reduced stiffness matrix, the corresponding mass or inertia items should not be input in AMASS. If reduction is not required, no data is required for IRWKP and IREDTO and the number of terms in AMASS must equal the order of the system stiffness matrix.

The number of nonrigid body modes to be calculated is input in NEIGVL. NEIGVL,which has a nominal value of 5, can optionally be any multiple of five, and must be less than or equal to IREDTO less six.

Care must be exercised in the use of the reduction routine. It is possible to eliminate so many rows from the stiffness matrix that a number of the remaining degrees of freedom are no longer independent. This has the effect of eliminating one or more of the system's six rigid body modes. To insure that this does not occur for a general space frame, at least two translational degrees of freedom in each of three global coordinate directions should remain following reduction. For the special case of a straight beam, two torsional rotations should remain in addition to these translations.

Error messages are printed if the user accidentally specifies a reduction which leads to the elimination of too many rows from the stiffness matrix.

Should more than 12904 words be required to store the stiffness matrix in a given problem, ISFDIM must be set equal to the required number of words and labeled COMMON must be redimensioned accordingly.

4.3.1.2 Output Data - Center Body Option output includes all input data, NAMELIST indicator values used, nodal displacements and rotations, number of iterations required for convergence, maximum error, nodal forces and moments, bar forces and moments, shear flows in shear webs, and a CPU time summary. Modal analysis output includes all input data, NAMELIST indicator values used, natural frequencies, mode shapes, generalized inertia properties, and a CPU time summary. All input data is printed out in a block format. Classification of input data is by code number in column 1, as described in Section 4.3.1.1.

For problems not requiring modal analysis, nodal displacements and rotations are printed out. The number of iterations required for the solution as well as the maximum error in the solution are then printed. Global forces and moments acting on the joints are also output. Forces and moments acting on both ends of all bars are printed. The p, q, and r joint numbers used for each bar are also listed to aid in interpretation of the direction of these forces and moments (positive local sign convention is indicated in Figure 4-1). Shear flows and identifying joints are printed for all shear webs. Positive local sign convention for establishing directions of shear flows is given in Figure 4-7.

When plastic force constraints are violated, the program prints out the elastic solution, which constraints were violated, and the plastic solution (nodal deflections and rotations, and nodal forces and moments). Forces and moments acting on both ends of all bars as well as shear flows in all shear webs are printed out for the final solution. The last output item is a CPU time summary showing a breakdown of CPU time usage.

When employing the modal analysis routine, output data include: input data, the specified number of lowest natural frequencies (excluding rigid body modes), corresponding mode shapes (normalized) for both the reduced and complete systems, generalized inertia properties, and a CPU time summary.

On an optional basis, this modal analysis output may be placed on magnetic tape. This information is required by the Center Body Landing Loads Program, as discussed in Appendix E. In addition, the data on this tape is compatible with the Task Order Two Landing Loads and Motion Program, Reference (2). In this case, the Task Order Five modal analysis could be employed for a footpad structure of a platform lander. Thus the improved matrix reduction routine could be taken advantage of.

4.3.1.3 Example of Program Operation - Examples of typical structural and modal analysis problems are included to illustrate interpreting input instructions and output data.

An example of obtaining deflections and internal loads for a center body structure subject to external loads is presented in Section 6.2.4 for the Task Order Three lander center body subject to typical landing loads. Input data and output listing for this problem are presented in Appendix A.

Input data and resulting output data when employing the modal analysis routine of the Center Body Option is given in Appendix B. This analysis was conducted on the center body structure of the Task Order Three lander. The idealization of the center body structure and plots of a number of the mode shapes are presented in Section 6.2.1. This modal data was input to the Landing Loads and Motions Program when studying the effects of center body flexibility on correlation with drop test data for this lander. This landing analysis is discussed in Section 6.2.2.

The data set shown in Appendix B is for a structural idealization with 33 joints and 53 bar members. This results in a stiffness matrix of order 198. The matrix reduction routine was employed to reduce this to order 99. This was accomplished by removing the rotations at all of the joints. Thus, the input mass matrix contained terms associated with each of the three translations at each joint. Twenty modes were requested from the modal analysis routine. Only the mode shapes for the complete system are shown in Appendix B.

4.3.2 Landing Gear Option - This section includes definition of input requirements and format, and output interpretation for the Landing Gear Option of the Structural Analysis Program. Examples of input and output data for a typical problem are contained in Appendix C.

4.3.2.1 Input Data - For each Landing Gear Option data set a data case header card and information describing landing gear type, loading information, strut properties, gear geometry, footpad information, energy absorption requirements, and indicators needed to control program operation are required as input data. Multiple data cases may be run by stacking data sets. Each data set must consist of a header card followed by the appropriate data cards.

A. Header Card - A header card is required as the first card of a data set to indicate that the Landing Gear Option will be employed. This card must contain the characters GEAR in columns 1 through 4 as indicated in Figure 4-36.

B. Data Cards - The input format for data cards employs a system of code numbers, right-justified in columns 1 and 2, to identify the type of data being read. The seventeen code numbers and corresponding type of input information are as follows:

(0) Comment cards
(1) Gear and load card
(2) Friction card
(3) Applied displacement card
(4) Strut material cards
(5) Solution parameter card
(6) Nodal point cards
(7) Footpad card
(8) Material parameter cards
(9) Material crush cards
(10) Compression crush distance cards
(11) Tension crush distance cards
(12) Compression spring rate cards
(13) Tension spring rate cards
(14) Compression plastic force cards
(15) Tension plastic force cards
(16) Data terminator card

B.0 Comment Cards - As many comment cards as desired may be input by leaving columns 1 through 6 blank and entering comments in columns 7 through 80 as shown in Figure 4-36. These comment cards may be located anywhere within the data cards and must precede the data terminator card.

(Header Card and Card Codes 0, 1, and 2)

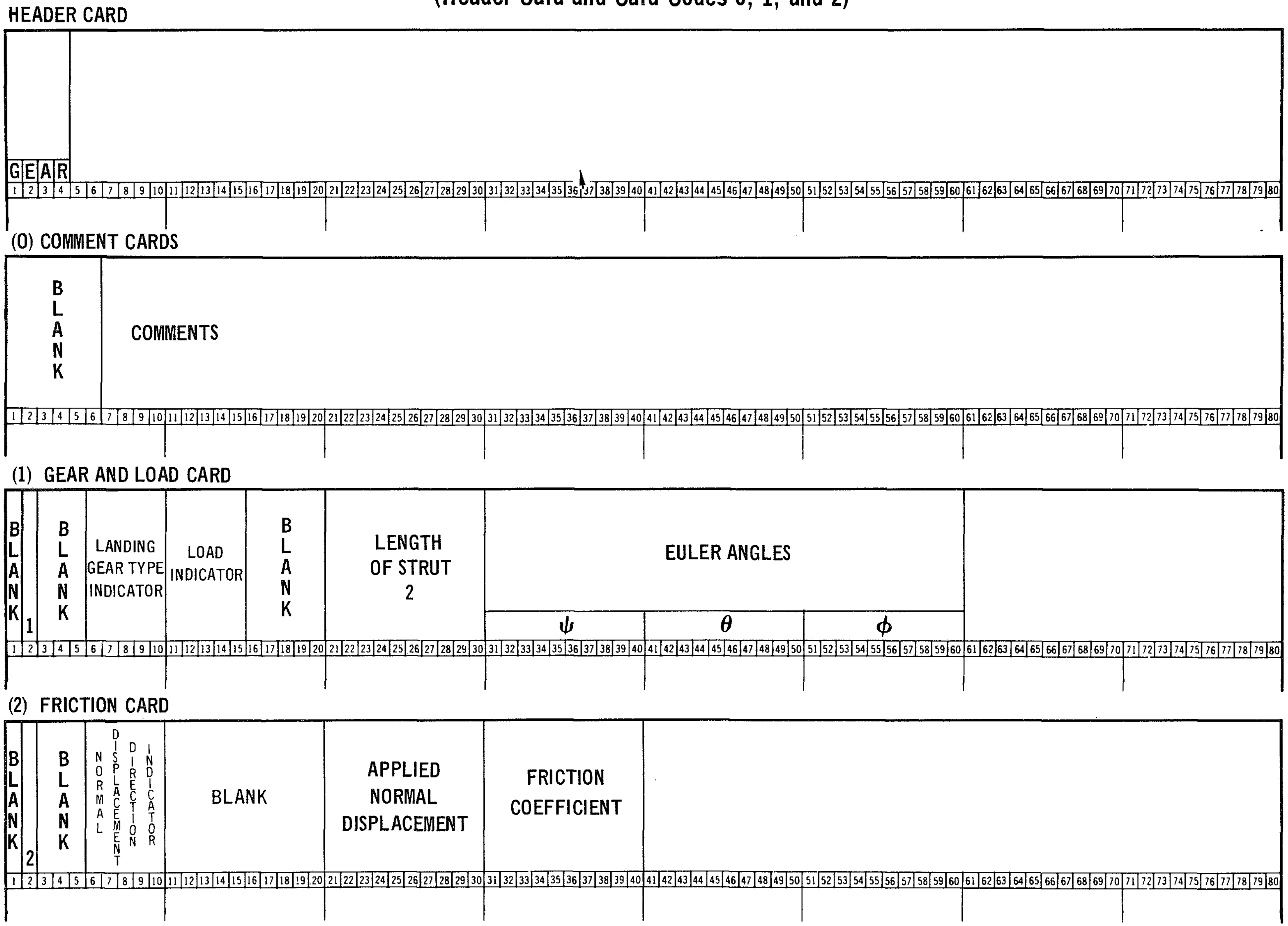


FIGURE 4–36 INPUT DATA FORMAT – LANDING GEAR OPTION

B.1 <u>Gear and Load Card</u> - Information describing landing gear type, type of applied loading, strut parameter for the cantilever gear, and the orientation of the landing surface is input on a data card in which a 1 is placed in column 2. Location of data fields on this card is illustrated in Figure 4-36. A 1 right-justified in columns 6 through 10 signifies that the gear being analyzed is an inverted tripod. Similarly, a 2 in this field indicates a cantilever gear.

A 1 right-justified in columns 11 through 15 indicates that the applied loading is of the frictional type in which the displacement normal to a landing surface is specified. For this type of loading, a code 2 data card is required as input data to specify the applied normal displacement and the friction coefficient. A 2 right-justified in columns 11 through 15 indicates that three applied displacement components are specified. For this case an associated code 3 data card must be input. As explained in Section B.2, a code 2 data card must also be input to specify the direction normal to the landing surface if the footpad has attenuation.

For the cantilever gear only, the length of strut 2 (see Figure 4-17) must be input in columns 21 through 30. This information is required to locate node 4 on the cantilever gear. For the inverted tripod gear this data field is left blank.

Three data fields located between columns 31 and 60 are used to specify the Euler angles ψ, θ, and ϕ which orient the Surface Coordinate System relative to the Lander Coordinate System. These Euler angles must be input in degrees. A definition of these angles and the associated sign convention is given in Figure 4-20.

B.2 <u>Friction Card</u> - Information describing the direction and magnitude of applied displacement (normal to the landing surface) at the footpad joint and the coefficient of friction of the surface is input on a data card in which a 2 is placed in column 2. Location of data fields on this card is illustrated in Figure 4-36. A 1, 2, or 3 right-justified in columns 6 through 10 of this card signifies that the positive direction normal to the landing surface is the X, Y, or Z surface coordinate direction, respectively. This information as well as the information supplied in the remaining data fields of this card must be specified if loading of the friction type is selected on the code 1 data card. Should the three applied displacement components case be specified on the code 1 data card and the footpad have at least 1 level of attenuation, the normal displacement direction indicator must be supplied on the code 2 data card to indicate the surface coordinate direction to check for footpad crushing. For this case, the remaining information on the data card is not used. Care must be exercised when selecting the coordinate system to insure that the normal surface coordinate direction is out of the surface.

The magnitude of the applied normal displacement in the surface coordinate direction indicated in columns 6 through 10 is supplied in columns 21 through 30. The coefficient of friction of the landing surface is entered in columns 31 through 40.

B.3 Applied Displacement Card - For the case in which three displacement components at the footpad joint are specified, the three surface coordinate components of applied displacement are entered on a data card with a 3 in column 2. Location of data fields on this card is illustrated in Figure 4-37. Three data fields located between columns 21 and 50 are used to enter the three components of applied displacement at the footpad joint. These displacement components must be given in the Surface Coordinate System.

B.4 Strut Material Cards - Each strut of a gear must be assigned a material identification number to indicate which material, input on code 8 through 15 data cards, makes up the strut. The material identification number of each element (strut) is input on data cards with a 4 in column 2. Location of data fields on this card is illustrated in Figure 4-37. One card must be input for each element of the gear, i.e., three for the inverted tripod and four for the cantilever.

The element number must be right-justified in columns 6 through 10. The number identifying the material of which this element is made is right-justified in columns 11 through 15. More than one element may have the same material identification number. This is convenient in the case of a gear which has identical drag struts since then only one set of material properties has to be defined for these struts.

(Card Codes 3 Thru 6)

(3) APPLIED DISPLACEMENT CARD

BLANK	3	BLANK	APPLIED DISPLACEMENT COMPONENTS IN SURFACE COORDINATES			
			X	Y	Z	
1	2	3–20	21–30	31–40	41–50	51–80

(4) STRUT MATERIAL CARDS

BLANK	4	BLANK	ELEMENT NO	MATERIAL NO	
1	2	3–5	6–10	11–15	16–80

(5) SOLUTION PARAMETER CARD

BLANK	5	BLANK	NO. OF LOADING STEPS	NO. OF ITERATIONS	NO. OF DIFFERENT MATERIALS	ENERGY CHECK INDICATOR	BLANK	ENERGY CUTOFF	SOLUTION TOLERANCE	
1	2	3–5	6–10	11–15	16–20	21–25	26–30	31–40	41–50	51–80

(6) NODAL POINT CARDS

BLANK	6	BLANK	NODE NO.	BLANK	NODE COORDINATES			
					X	Y	Z	
1	2	3–5	6–10	11–20	21–30	31–40	41–50	51–80

FIGURE 4–37 INPUT DATA FORMAT – LANDING GEAR OPTION

B.5 Solution Parameter Card - The number of steps for investigating the applied loading, the number of iterations to be allowed, the number of different materials used in the gear idealization, an indicator to specify in which direction to check the energy absorbed, the energy absorption cutoff, and the solution tolerance are input on a data card with a 5 in column 2. Location of data fields on this card is illustrated in Figure 4-37.

Four data fields located between columns 6 and 25 are used to specify the number of steps to divide the applied loading, the number of iterations allowed before the solution is assumed to be nonconvergent, the number of different materials employed in idealizing the struts of the gear, and an indicator to specify in which surface coordinate direction to check the energy absorption associated with the loads acting on the footpad. These integers must be right-justified in their respective fields. The energy check indicator is set to 1, 2, or 3 if the energy absorption is to be checked in the X, Y, or Z surface coordinate direction, respectively.

The energy cutoff used to make the energy absorption check is input in columns 31 through 40. In assigning a sign to this cutoff it must be remembered that a plus sign signifies that the force and corresponding displacement components of interest are in the same direction while a minus sign signifies that they are opposite in direction. The tolerance used to check for solution convergence is input in columns 41 through 50. For input data employing units of centimeters and dynes, the solution would be assumed converged to a tolerance specified as 4.0×10^4 when the magnitude of the largest unbalance, between any internal and external load component at any joint of the gear, was less than 4.0×10^4 dynes or 4.0×10^4 dyne-cm as the case may be. For each loading step,the program will run until the solution converges to the tolerance of the maximum number of iterations is exceeded.

B.6 Nodal Point Cards - For each node point (joint) of the gear, except for node 4 of the cantilever gear, the initial node coordinates in the Lander Coordinate System are input on cards with a 6 in column 2. The location of data fields on these cards is illustrated in Figure 4-37. Four of these cards must be input for both the inverted tripod and cantilever gears. On each card, the node point number must be right-justified in columns 6 through 10. Three data fields located between columns 21 and 50 are used to input the X, Y, and Z coordinates of the node in the Lander Coordinate System.

B.7 Footpad Card - Information regarding attenuation on the bottom of the footpad is input on a card with a 7 in column 2. Location of data fields on this card is illustrated in Figure 4-38. An attenuation indicator to signify whether or not the footpad has attenuation is right-justified in columns 6 through 10. A 1 signifies that the footpad has attenuation while a 2 indicates that there is none. The number of attenuation crush levels is entered, right-justified, in columns 11 through 15.

Three data fields located between columns 21 and 50 are used to input the magnitudes of up to three attenuation crush forces (see Figure 4-28). These crush forces must be entered in the order of increasing magnitude. The crush stroke magnitudes, associated with up to three crush levels, are entered in three data fields located between columns 51 and 80. Each stroke entry represents the stroke of the footpad honeycomb during crushing of the level of interest and does not include the stroking that may have occurred due to crushing of lower levels. For a footpad without attenuation the information in columns 21 through 80 will be ignored.

B.8 Material Parameter Cards - For each different material, the material identification number, plasticity indicator for axial displacements, moduli of elasticity, and cross-sectional properties are input on a card with an 8 in column 2. Location of data fields on this card is illustrated in Figure 4-38. Material identification number is entered, right-justified, in columns 6 through 10. A plasticity indicator for this material is entered, right-justified, in columns 11 through 15. A plasticity indicator of 1 signifies that for axial displacements the material behaves elastically. A plasticity indicator of 2 indicates that for axial displacements the material behaves plastically. That is, the axial characteristics of this material are those of honeycomb attenuation.

Two data fields located between columns 21 and 40 are used to input the moduli of elasticity for bending and axial displacements of the strut composed of the material identified in columns 6 through 10. Two data fields located between columns 41 and 60 are used to input the moment of inertia for bending displacements and the cross-sectional area of the strut composed of the material identified in columns 6 through 10.

For a material with plastic axial capability (a plasticity indicator of 2), the modulus of elasticity for bending displacements and the moment of

(Card Codes 7 Thru 10)

(7) FOOTPAD CARD

BLANK		BLANK	ATTENUATION INDICATOR	NO. CRUSH LEVELS	BLANK	ATTENUATION CRUSH FORCES			ATTENUATION CRUSH STROKES		
	7					1ST	2ND	3RD	1ST	2ND	3RD
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

(8) MATERIAL PARAMETER CARDS

BLANK		BLANK	MATERIAL NO	PLASTICITY INDICATOR	BLANK	MODULI OF ELASTICITY		CROSS SECTIONAL PROPERTIES		
						BENDING	AXIAL	MOMENT OF INERTIA	AREA	
	8					E_B	E_A	I	A	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

(9) MATERIAL CRUSH CARDS

BLANK		BLANK	PLASTIC MATERIAL NO	UNLOADING INDICATOR	INITIAL STROKE INDICATOR	ALLOWABLE STROKE		UNLOADING SPRING RATE		
	9					COMP	TENS	COMP	TENS	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

(10) COMPRESSION CRUSH DISTANCE CARDS

		BLANK	PLASTIC MATERIAL NO	BLANK	CRUSH DISTANCES IN COMPRESSION					
1	0				1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

FIGURE 4-38 INPUT DATA FORMAT – LANDING GEAR OPTION

inertia must be entered if the strut composed of this material can carry bending loads. These parameters should not be entered if the strut cannot carry bending loads. For either of the above materials, the modulus of elasticity for axial displacements and the cross-sectional area will be ignored since this information regarding axial capability is input in the crush property cards (card codes 9 through 15) for a material with plastic axial capability. The material for the lower member of the main strut (element 4 in Figure 4-17(b)) of the cantilever gear is an example of a material with plastic axial capability for which the bending properties must be input. The material for the main strut (element 2 in Figure 4-17(a)) of the inverted tripod gear is an example of a material with plastic axial capability for which bending properties would not be input.

For a material with elastic axial capability (a plasticity indicator of 1), the modulus of elasticity for axial displacements and the cross-sectional area must be entered. The modulus of elasticity for bending displacements and the moment of inertia are input only if the associated strut has bending capability. The material for the upper member of the main strut (element 2 in Figure 4-17(b)) of the cantilever gear is an example of a material with elastic axial capability for which bending properties must be input. For an inverted tripod gear with rigid drag struts, the material for the drag struts (elements 1 and 3 in Figure 4-17(a) is an example of a material with elastic axial capability for which bending properties would not be input.

B.9 Material Crush Cards - For each material with plastic axial capability (a material which has a plasticity indicator of 2 on the code 8 data card), the material identification number, crush indicators, allowable strokes, and unloading spring rates must be input on a data card with a 9 in column 2. The location of data fields on this card is illustrated in Figure 4-38. The material identification number is right-justified in columns 6 through 10. An unloading indicator is right-justified in columns 11 through 15. A value of 0 for this indicator signifies that unloading of this material will be along a slope specified by the spring rates input in columns 41 through 60 of this card. An unloading indicator of 1 signifies that the material will unload along a slope specified by the next spring rate associated with the load-stroke curve. An initial stroke indicator is input, right-justified, in columns 16 through 20.

A value of 1 for this indicator signifies that the strut which is composed of this material will initially stroke in tension. A value of -1 indicates that this strut will initially stroke in compression. This information is needed for a crushable material to indicate whether the stiffness of the strut at the start of the loading should reflect the first spring rate in tension or that in compression.

Two data fields located between columns 21 and 40 are used to input the magnitudes of allowable (bottoming) strokes in compression and tension. For the material of a strut which can stroke in one direction only, the appropriate bottoming stroke should be input as 1.E-20. Two data fields located between columns 41 and 60 are used to input the unloading spring rate magnitudes in compression and tension. These values should be input only if the unloading indicator is input as 0 in columns 11 through 15.

B.10 <u>Compression Crush Distance Cards</u> - For each material with plastic axial capability, the crush distances in compression (see Figure 4-18) must be input on a data card with a 10 in columns 1 and 2. Data fields on this card are defined in Figure 4-38. The material identification number is right-justified in columns 6 through 10.

Five data fields, located between columns 21 and 70, are used to define the magnitudes of up to five crush distances in compression. Any number of crush distances between one and five may be input. The last crush distance input must be greater than the allowable compression stroke input on a code 9 data card for this material. This crush distance should have a fictitious value of 1.E+20 to insure that it will not be exceeded during program iteration prior to final convergence. For the material of a strut which can stroke only in tension, one compressive crush distance should be input at a value of 1.E+20.

B.11 <u>Tension Crush Distance Cards</u> - For each material with plastic axial capability, the crush distances in tension (see Figure 4-18) must be input on a data card with an 11 in columns 1 and 2. Data fields on this card are defined in Figure 4-39. The material identification number is right-justified in columns 6 through 10.

Five data fields, located between columns 21 and 70, are used to define the magnitudes of up to five crush distances in tension. Any number of crush distances between one and five may be input. The last crush distance input

(Card Codes 11 thru 14)

(11) TENSION CRUSH DISTANCE CARDS

1	1	BLANK	PLASTIC MATERIAL NO	BLANK	CRUSH DISTANCES IN TENSION					
					1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

(12) COMPRESSION SPRING RATE CARDS

1	2	BLANK	PLASTIC MATERIAL NO	BLANK	SPRING RATES IN COMPRESSION					
					1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

(13) TENSION SPRING RATE CARDS

1	3	BLANK	PLASTIC MATERIAL NO	BLANK	SPRING RATES IN TENSION					
					1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

(14) COMPRESSION PLASTIC FORCE CARDS

1	4	BLANK	PLASTIC MATERIAL NO	BLANK	PLASTIC FORCES IN COMPRESSION					
					1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

FIGURE 4–39 INPUT DATA FORMAT – LANDING GEAR OPTION

must be greater than the allowable tension stroke input on a code 9 data card for this material. This crush distance should have a fictitious value of 1.E20 to insure that it will not be exceeded during program iteration prior to final convergence. For the material of a strut which can stroke only in compression, one tension crush distance should be input at a value of 1.E+20.

B.12 Compression Spring Rate Cards - For each material with plastic axial capability, the spring rates in compression (see Figure 4-18) must be input on a card with a 12 in columns 1 and 2. Data fields on this card are defined in Figure 4-39. The material identification number is right-justified in columns 6 through 10.

Five data fields located between columns 21 and 70 are used to define the magnitudes of up to five spring rates in compression. Any number of spring rates between 1 and 5 may be input. For the material of a strut which can only stroke in tension, one spring rate in compression should be input at a value equal to the bottoming spring rate.

B.13 Tension Spring Rate Cards - For each material with plastic axial capability, the spring rates in tension (see Figure 4-18) must be input on a data card with a 13 in columns 1 and 2. Data fields on this card are defined in Figure 4-39. The material identification number is right-justified in columns 6 through 10.

Five data fields located between columns 21 and 70 are used to define the magnitude of up to five spring rates in tension. Any number of spring rates between 1 and 5 may be input. For the material of a strut which can only stroke in compression, one spring rate in tension should be input at a value equal to the bottoming spring rate.

B.14 Compression Plastic Force Cards - For each material with plastic axial capability, the plastic forces in compression (see Figure 4-18) must be input on a data card with a 14 in columns 1 and 2. Data fields on this card are defined in Figure 4-39. Material identification number is entered, right-justified, in columns 6 through 10.

Five data fields located between columns 21 and 70 are used to define the magnitudes of up to five plastic forces in compression. Any number of plastic forces between 1 and 5 may be input. For the material of a strut which can only stroke in tension, one plastic force in compression should be input at a value slightly less than the product of SRC(1) and CDC(1).

B.15 Tension Plastic Force Cards - For each material with plastic axial capability, the plastic forces in tension (see Figure 4-18) must be input on a data card with a 15 in columns 1 and 2. Data fields on this card are defined in Figure 4-40. Material identification number is right-justified in columns 6 through 10.

Five data fields located between columns 21 and 70 are used to define the magnitudes of up to five plastic forces in tension. Any number of plastic forces between 1 and 5 may be input. For the material of a strut which can only stroke in compression, one plastic force in tension should be input at a value slightly less than the product of SRT(1) and CDT(1).

B.16 Data Terminator Card - After all data needed on code 1 through 15 data cards and the desired comment cards are input, a data terminator card is required to signify the end of the data set. This card must contain a 16 in columns 1 and 2.

4.3.2.2 Output Data - Output data from the Landing Gear Option includes all input data, initial conditions, displaced conditions for the gear at each step of the specified loading, and a CPU time summary.

The initial conditions printed for the gear include the X, Y, and Z coordinates of all nodes in the Surface Coordinate System.

The displaced conditions, printed for the gear at each step of loading, include the following: step of loading; number of iterations required to reach convergence; coordinates of all nodes in the Surface Coordinate System; external forces and moments at the nodes in the Surface Coordinate System; the internal axial force, stroke, and axial stiffness of each element; the internal shear force and moment at each end of all elements; the total energy absorbed at the footpad in both surface and lander coordinate components; the number of remaining (noncrushed) footpad attenuation levels; and surface and lander coordinate components of the energy absorbed by footpad attenuation during the loading step concerned.

Should a particular loading step fail to converge, the solution existing in the program, after the maximum number of iterations had been employed, would be printed out and program termination would occur. Should the allowable stroke in any of the members be exceeded or the load normal to the footpad be

(Card Codes 15 and 16)

(15) TENSION PLASTIC FORCE CARDS

1	5	BLANK	PLASTIC MATERIAL NO	BLANK	PLASTIC FORCES IN TENSION					
					1ST	2ND	3RD	4TH	5TH	
1	2	3 4 5	6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80

(16) DATA TERMINATOR CARD

1	6	
1	2	3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

FIGURE 4-40 INPUT DATA FORMAT – LANDING GEAR OPTION

tensile for any loading step, the current solution and an appropriate error message would be printed, and program termination would occur. The program will continue through all loading steps unless one of the above errors occurs causing termination. If at any loading step the energy cutoff in the appropriate surface coordinate direction is exceeded, a message will be printed to indicate this fact and the program will continue.

4.3.2.3 Example of Program Operation - To illustrate interpretation of input instructions and output data, a typical example of the Landing Gear Option is included. An example problem where the inverted tripod gear of the Task Order Three lander was analyzed is discussed in Section 6.1.1. Input data and output listing for this example are presented in Appendix C.

5. LANDING LOADS AND MOTIONS PROGRAM

The Landing Loads and Motions Program provides the capability of predicting the landing dynamics of a legged lander. Program output presents the time histories of the landing gear loads and the spatial positions, velocities, and accelerations of the lander. Options are available for determining the distribution of landing loads throughout the lander structure and lander stability information.

The lander is comprised of up to five landing gears connecting the center body to the footpads which make contact with the landing surface. The center body may be treated as either a rigid structure with up to six rigid body degrees of freedom, or the effects of a flexible structure may be superimposed on these rigid body motions. Inverted tripod or cantilever landing gears having elastic-plastic load-stroke characteristics, velocity dependent energy absorption characteristics, or a combination of the two may be considered. Two soil mechanics routines are available and, on an optional basis, footpad attenuation material may be located on the bottom of each footpad.

Analytical methods developed for this program are presented in Section 5.1. Organization of the computer program is presented in Section 5.2 and operating instructions are discussed in Section 5.3. A program listing is given in Appendix I.

5.1 Analytical Methods - Analytical methods associated with the Landing Loads and Motions Program are presented below. Included are discussions of coordinate systems, equations of motion, structural idealization, soil mechanics routines, and footpad attenuation system.

5.1.1 Coordinate Systems - Four types of coordinate systems, as shown in Figure 5-1, are used to locate the lander as a function of time. These consist of two axis systems fixed relative to the planet, and two systems moving with the lander. These coordinate systems are all right-handed, and each has three orthogonal axes.

The coordinate systems are defined as follows:

Surface Coordinate System (X_s, Y_s, Z_s) - A coordinate system fixed in the planet and oriented with respect to the slope of the local

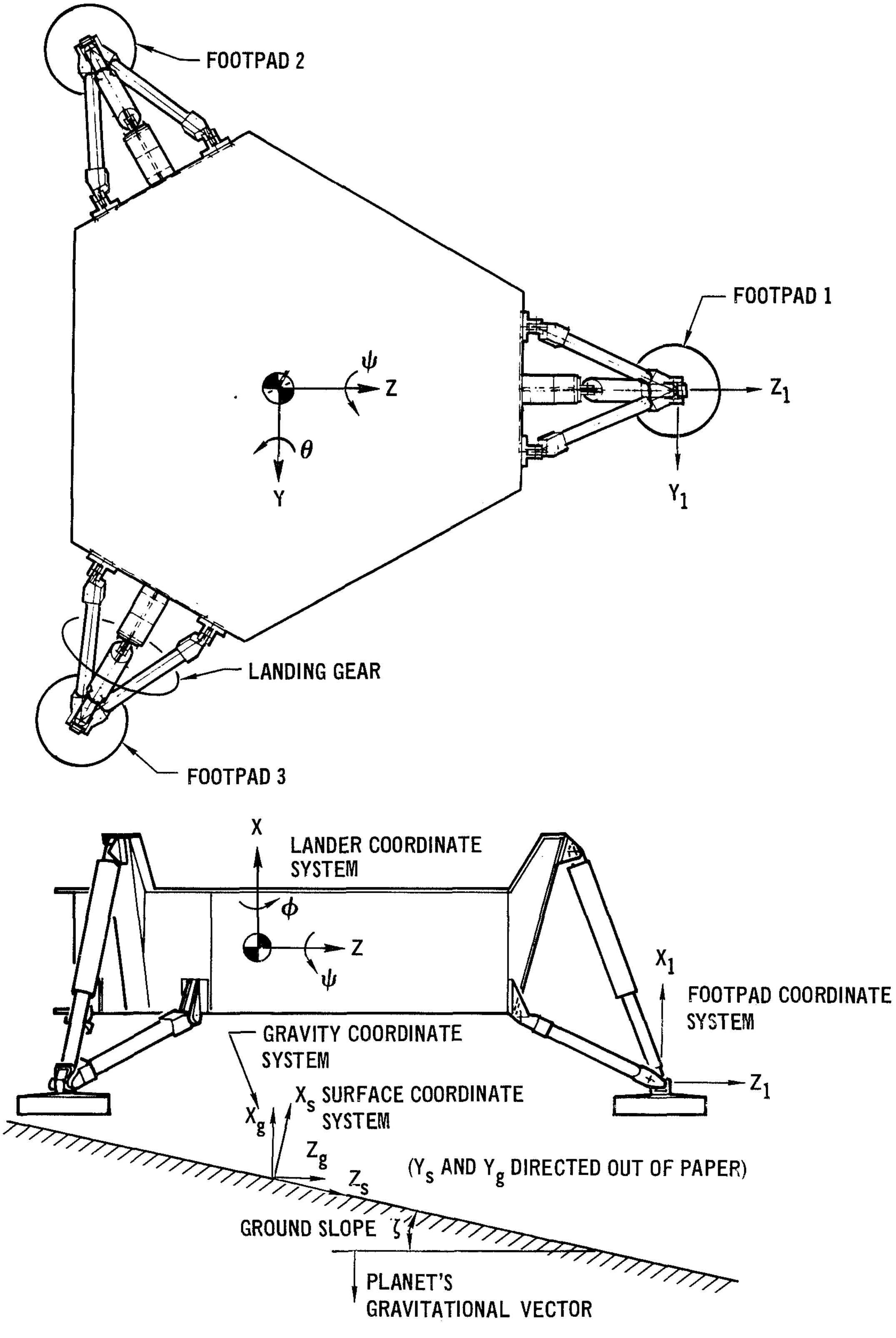


FIGURE 5-1 COORDINATE SYSTEMS

surface. The X axis of this system is perpendicular to the ground surface. The Z axis corresponds to the principle direction along the ground slope and the Y axis is 90 degrees across the slope.

Gravity Coordinate System (X_g, Y_g, Z_g) - A coordinate system fixed relative to the planet's local acceleration of gravity vector. The X_g axis is directed opposite to the local gravitational vector and the Y_g axis is parallel to the Surface Coordinate System Y_s axis. The Z_g axis is perpendicular to the gravitational vector and in the same general direction as the Surface Coordinate System Z_s axis.

Lander Coordinate System (X, Y, Z) - A coordinate system moving with the lander and fixed at the center of gravity of the center body. The lander's angular positions are defined in terms of the three Euler angles ψ, θ, and ϕ. Definition of the lander's angular position relative to the landing surface is based on a specific order for these Euler angle rotations. The order required for these rotations consists of an initial rotation, ψ, about the Z_s axis, followed by a rotation, θ, about the displaced Y axis and finally by a rotation, ϕ, about the direction of the displaced X axis resulting from the previous two rotations. Note, that when all three of these Euler angles are zero, the Lander Coordinate System is aligned with the Surface Coordinate System.

Footpad Coordinate System (X_i, Y_i, Z_i) - A coordinate system moving with the ith footpad and fixed at the footpad-main strut pivot point. The Y_i -Z_i plane remains parallel to the bottom of the footpad as the footpad impacts the landing surface. Footpad 1 may be located anywhere relative to the Lander Coordinate System. The remaining footpads are numbered consecutively in a positive angular direction about the X axis.

The Lander Coordinate System is related to the Surface Coordinate System by the following expression:

$$\begin{Bmatrix} X_s \\ Y_s \\ Z_s \end{Bmatrix} = \begin{bmatrix} DC11 & DC12 & DC13 \\ DC21 & DC22 & DC23 \\ DC31 & DC32 & DC33 \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \qquad (5\text{-}1)$$

The elements in the lander's direction cosine matrix, [DC] are given as follows:

$$
\begin{aligned}
DC11 &= \text{Cos } \theta \text{ Cos } \psi \\
DC12 &= \text{Sin } \phi \text{ Sin } \theta \text{ Cos } \psi - \text{Cos } \phi \text{ Sin } \psi \\
DC13 &= \text{Cos } \phi \text{ Sin } \theta \text{ Cos } \psi + \text{Sin } \phi \text{ Sin } \psi \\
DC21 &= \text{Cos } \theta \text{ Sin } \psi \\
DC22 &= \text{Sin } \phi \text{ Sin } \theta \text{ Sin } \psi + \text{Cos } \phi \text{ Cos } \psi \\
DC23 &= \text{Cos } \phi \text{ Sin } \theta \text{ Sin } \psi - \text{Sin } \phi \text{ Cos } \psi \\
DC31 &= -\text{Sin } \theta \\
DC32 &= \text{Sin } \phi \text{ Cos } \theta \\
DC33 &= \text{Cos } \phi \text{ Cos } \theta
\end{aligned}
\tag{5-2}
$$

In addition to the above transformations, relationships between the time derivatives of the Euler angles and the angular velocity components about the lander coordinate axes are required. Integration of these define the lander's angular position as a function of time. The required relationships are expressed as

$$
\begin{aligned}
\dot{\phi} &= \dot{\Omega}_x + \text{Tan } \theta(\dot{\Omega}_y \text{Sin } \phi + \dot{\Omega}_z \text{Cos } \phi) \\
\dot{\theta} &= \dot{\Omega}_y \text{Cos } \phi - \dot{\Omega}_z \text{Sin } \phi \\
\dot{\psi} &= (\dot{\Omega}_z \text{Cos } \phi + \dot{\Omega}_y \text{Sin } \phi)/\text{Cos } \theta
\end{aligned}
\tag{5-3}
$$

where $\dot{\Omega}_x$, $\dot{\Omega}_y$, and $\dot{\Omega}_z$ are the components of the lander's angular velocity in the Lander Coordinate System.

5.1.2 <u>Equations of Motion</u> - The lander's equations of motion are discussed in two parts. The first presents the rigid body equations for the footpads considered to be in contact with the landing surface. The second part presents the development of the center body equations of motion, including the effects of a flexible center body structure.

5.1.2.1 <u>Contacting Footpads</u> - A footpad is termed "contacting" for two conditions. In the first, a footpad is considered to be contacting only after the footpad actually makes contact with the landing surface. Footpads not considered in contact with the landing surface are included in the center body equations as discussed in Section 5.1.2.2.

Secondly, on an optional basis, all of the lander's footpads may be treated as contacting footpads even if some of the footpads have not yet impacted the landing surface. In this case, inertia loading of the landing gear struts, due to the footpad mass, is simulated.

Each footpad considered to be in contact with the landing surface has three rigid body translation degrees of freedom. Two of these are in the plane of the landing surface and the third is normal to the landing surface. Equations of motion for each contacting footpad are written in the form

$$\begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \begin{Bmatrix} \ddot{X}_i \\ \ddot{Y}_i \\ \ddot{Z}_i \end{Bmatrix} = \begin{Bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \end{Bmatrix} + \begin{Bmatrix} -m_i \; g \; \text{Cos} \; \zeta \\ 0 \\ m_i \; g \; \text{Sin} \; \zeta \end{Bmatrix} \tag{5-4}$$

Terms in Equation (5-4) are defined as follows:

m_i = mass of ith footpad.

$\ddot{X}_i$, $\ddot{Y}_i$, $\ddot{Z}_i$ = translational accelerations of ith footpad in Surface Coordinate System.

F_{x_i}, F_{y_i}, F_{z_i} = sum of forces acting on ith footpad in Surface Coordinate System.

g = local acceleration of gravity.

ζ = local surface slope.

Two sets of forces are acting on each contacting footpad. These are the loads due to the stroking of the landing gear struts and the loads resulting from the interaction between the soil and the footpad attenuation system. These soil loads are zero when the footpad is clear of the landing surface.

5.1.2.2 Center Body - The effects of a flexible center body structure on the center body's motion have been included in the analysis. This is an optional feature of the Landing Loads and Motions Program and may be suppressed through input data. If suppressed, the center body is treated as a rigid body with up to six degrees of freedom.

To completely describe the dynamic motion of the elastic center body, a continuous elastic body must be considered. However, for the analysis of complex structures, the structure is often idealized as a network of finite elements. Motions of the idealized structure are determined at a finite number of arbitrarily selected control points distributed throughout the body. The motion of each control point on this body is expressed in terms of three translational and three rotational displacements. This idealization results in a total number of equations of motion equal to six times the number of control points selected.

In order to reduce the number of equations to be solved in the Landing Loads and Motions Program, the normal mode method, References (9) and (10), was employed. In this method the motion of the center body is approximated by the combination of a limited number of vibratory modes plus the six rigid body modes. By selecting the vibratory modes which will be excited for a given landing condition, the behavior of the flexible center body structure is obtained.

For the analysis of the legged lander, the center body's free-free (unrestrained) modes were chosen as the vibratory modes. The rigid body modes were assumed to be the three translational displacements defining the position of the center body center of gravity in the Surface Coordinate System and three angular displacements defined in the Lander Coordinate System.

In developing the center body equations of motion, expressions defining the motion of a point on the center body were obtained. These were used to evaluate the kinetic and potential energy of the center body. The final form of the center body's equations of motion were obtained by applying the Lagrangian equations to these energy expressions.

The total displacement of a point on the center body, point j in Figure 5-2, is defined as

$$\bar{r}_j = \bar{R} + \bar{\rho}_j \qquad (5\text{-}5)$$

$\bar{r}_j$ = position vector of point j relative to the Surface Coordinate System.

$\bar{R}$ = position vector of center body center of gravity relative to Surface Coordinate System.

$\bar{\rho}_j$ = position vector of point j relative to the Lander Coordinate System.

Because the center body is experiencing angular velocities and accelerations, the velocity and acceleration of point j are expressed as

$$\dot{\bar{r}}_j = \dot{\bar{R}} + \dot{\bar{\rho}}_j + (\dot{\bar{\Omega}} \times \bar{\rho}_j)$$

$$\ddot{\bar{r}}_j = \ddot{\bar{R}} + \ddot{\bar{\rho}}_j + (\ddot{\bar{\Omega}} \times \bar{\rho}_j) + \dot{\bar{\Omega}} \times (\dot{\bar{\Omega}} \times \bar{\rho}_j) + 2(\dot{\bar{\Omega}} \times \dot{\bar{\rho}}_j) \qquad (5\text{-}6)$$

In the above, the following definitions apply

$\dot{\bar{\Omega}}$ = angular velocity of center body in the Lander Coordinate System.
$\ddot{\bar{\Omega}}$ = angular acceleration of center body in the Lander Coordinate System.
$\dot{\bar{r}}_j$ = velocity vector of point j relative to the Surface Coordinate System.
$\ddot{\bar{r}}_j$ = acceleration vector of point j relative to the Surface Coordinate System.
$\dot{\bar{R}}$ = velocity vector of center body center of gravity relative to the Surface Coordinate System.
$\ddot{\bar{R}}$ = acceleration vector of center body center of gravity relative to the Surface Coordinate System.
$\dot{\bar{\rho}}_j$ = velocity vector of point j relative to the Lander Coordinate System.
$\ddot{\bar{\rho}}_j$ = acceleration vector of point j relative to the Lander Coordinate System.

It is assumed that the position vector locating the point on the center body can be separated into a term which varies with time and a term which remains constant with time. Thus, the location of point j relative to the Lander Coordinate System is

$$\bar{\rho}_j(s_j, t) = \bar{\rho}_{oj}(s_j) + \bar{\rho}_{ej}(s_j, t) \qquad (5\text{-}7)$$

where

s_j = coordinates of point j in Lander Coordinate System ($s_j = X_j$, Y_j, and Z_j).
$\bar{\rho}_{oj}$ = undeformed position of point j in Lander Coordinate System.
$\bar{\rho}_{ej}$ = deformed position of point j in Lander Coordinate System measured from the undeformed position of that point.

These position vectors are shown in Figure 5-3.

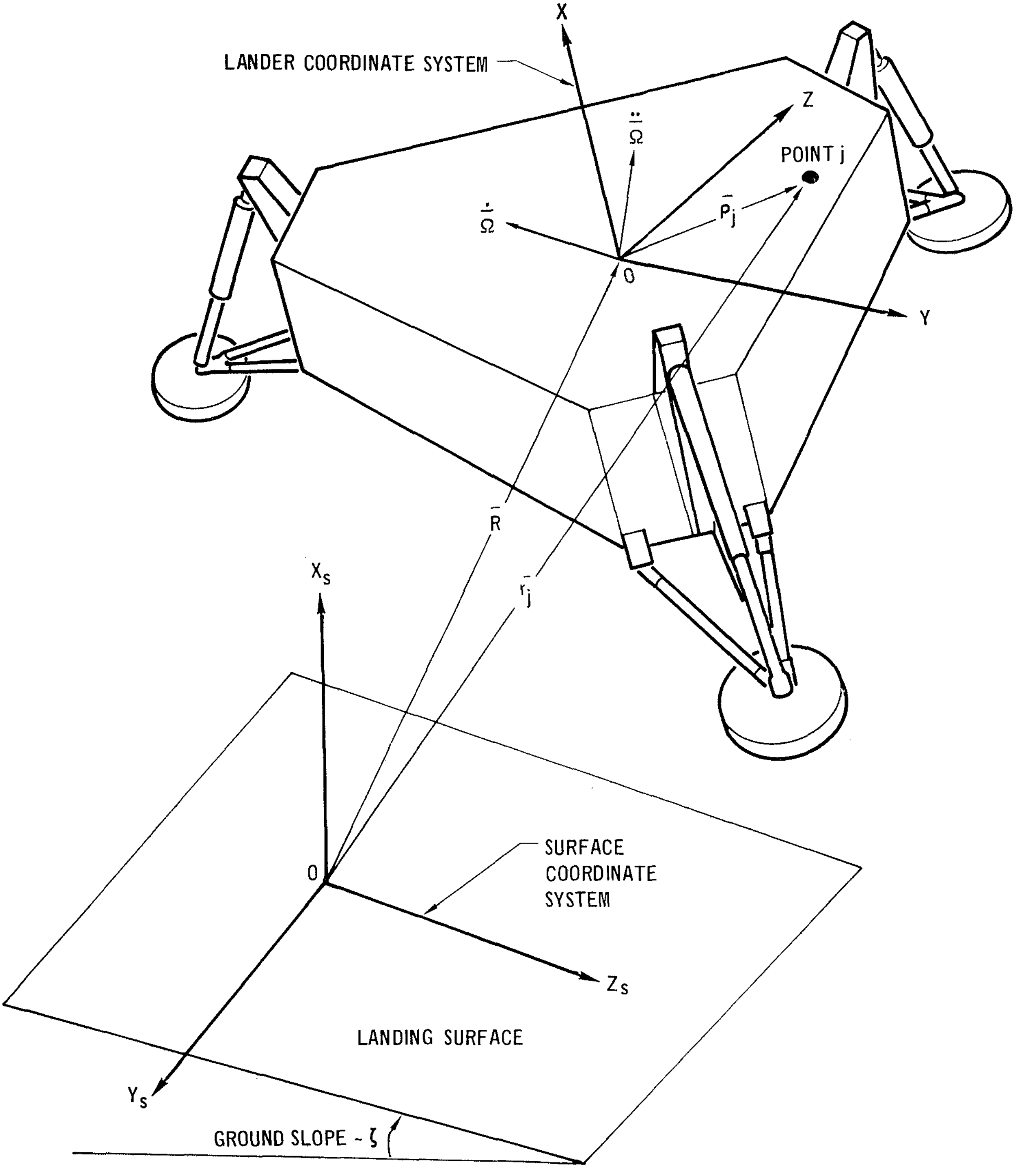


FIGURE 5–2 POSITION OF POINT ON CENTER BODY IN SURFACE COORDINATE SYSTEM

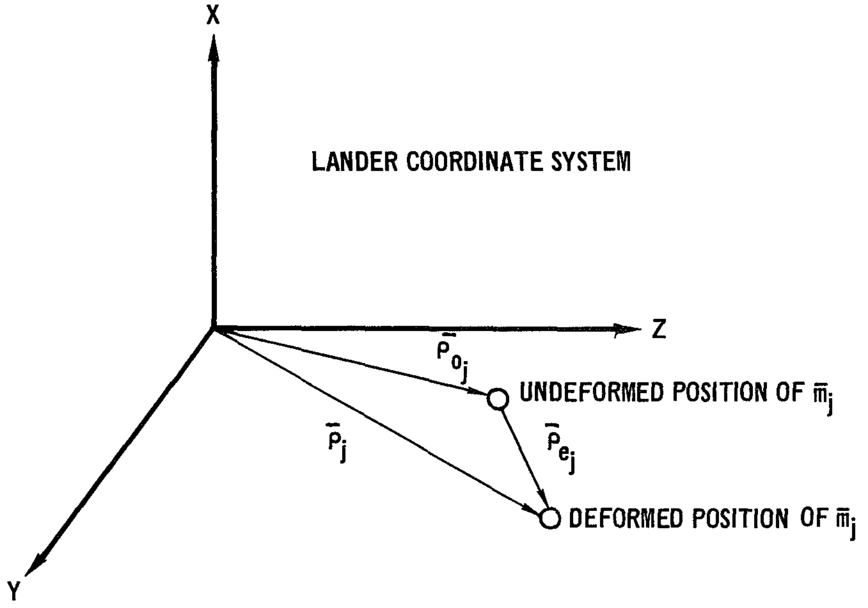


FIGURE 5-3 POSITION OF POINT ON CENTER BODY IN LANDER COORDINATE SYSTEM

Employing the assumption that the elastic deformation is represented by the superposition of a limited number of vibratory modes, the deformed position of point j is

$$\bar{\rho}_{ej} = \sum_{n}^{N} \bar{\phi}_{nj} q_n(t) \tag{5-8}$$

where

N = number of modes included.

$\bar{\phi}_{nj}$ = magnitude of nth elastic mode shape at point j. These are a function of the coordinates s_j.

q_n = generalized coordinate associated with nth mode. These are a function of time.

Expressing the deformed position of point j as components in the three axes of the Lander Coordinate System results in the following

$$\bar{\rho}_{ej} = \rho^j_{ex}\bar{i} + \rho^j_{ey}\bar{j} + \rho^j_{ez}\bar{k} \tag{5-9}$$

In the above, $\bar{i}$, $\bar{j}$, and $\bar{k}$ are the unit normal vectors in the Lander Coordinate

System. Combining Equations (5-8) and (5-9) results in the following expression for the deformed position of point j.

$$\bar{\rho}_{ej} = \sum_{n}^{N} (\phi^{j}_{xn}\bar{i} + \phi^{j}_{yn}\bar{j} + \phi^{j}_{zn}\bar{k})q_{n} \tag{5-10}$$

The terms, ϕ^{j}_{xn}, ϕ^{j}_{yn}, and ϕ^{j}_{zn} are the components of the nth mode shape at point j in the three lander axes directions.

The total velocity of point j is obtained by combining Equations (5-6) and (5-7).

$$\dot{\bar{r}}_{j} = \dot{\bar{R}} + \dot{\bar{\rho}}_{ej} + \dot{\bar{\Omega}} \times (\bar{\rho}_{oj} + \bar{\rho}_{ej}) \tag{5-11}$$

The total kinetic energy of the center body, T, is obtained by summing the kinetic energy of all control points on the center body each having a mass $\bar{m}_{j}$

$$T = \frac{1}{2} \sum_{j}^{J} \bar{m}_{j}\dot{\bar{r}}_{j} \cdot \dot{\bar{r}}_{j} \tag{5-12}$$

where J equals the total number of mass points on the center body. Combining Equation (5-12) with the definitions given in Equations (5-10)and (5-11), results in the following kinetic energy expression

$$\begin{aligned} T = {} & \frac{1}{2} M(\dot{q}_{x}^{2} + \dot{q}_{y}^{2} + \dot{q}_{z}^{2}) + \frac{1}{2} (I_{xx}\dot{q}_{rx}^{2} + I_{yy}\dot{q}_{ry}^{2} + I_{zz}\dot{q}_{rz}^{2}) \\ & - (I_{xy}\dot{q}_{rx}\dot{q}_{ry} + I_{xz}\dot{q}_{rx}\dot{q}_{rz} + I_{yz}\dot{q}_{ry}\dot{q}_{rz}) \\ & + \frac{1}{2} \dot{q}_{rx}^{2} \sum_{n}^{N} [(P_{yn} + P_{zn})q_{n} + \frac{1}{2} N_{xn}q_{n}^{2}] \\ & + \frac{1}{2} \dot{q}_{ry}^{2} \sum_{n}^{N} [(P_{xn} + P_{zn})q_{n} + \frac{1}{2} N_{yn}q_{n}^{2}] \\ & + \frac{1}{2} \dot{q}_{rz}^{2} \sum_{n}^{N} [(P_{xn} + P_{yn})q_{n} + \frac{1}{2} N_{zn}q_{n}^{2}] \\ & + \frac{1}{2} \sum_{n}^{N} m_{n}q_{n}^{2} \end{aligned} \tag{5-13}$$

Terms in Equation (5-13) are defined as follows, where the q_k's and q_{rk}'s are the generalized coordinates of the center body's rigid body modes.

M = center body mass.

I_{xx}, I_{yy}, I_{zz} = center body moments of inertia

I_{xy}, I_{xz}, I_{yz} = center body products of inertia.

$q_k, \dot{q}_k, \ddot{q}_k$ = rigid body translational displacement, velocity, and acceleration of center body center of gravity in the Surface Coordinate System (for $k = X_s, Y_s$, or Z_s axes).

$q_{rk}, \dot{q}_{rk}, \ddot{q}_{rk}$ = rigid body angular displacement, velocity, and acceleration relative to Lander Coordinate System (for $k = X, Y$, or Z).

m_n = generalized mass of nth elastic mode,

$$= \sum_j^J \bar{m}_j \bar{\phi}_{nj} \cdot \bar{\phi}_{nj} .$$

In addition, the following center body generalized inertia properties are expressed as

$$P_{xn} = \sum_j^J X_j \bar{m}_j \phi^j_{xn}$$

$$P_{yn} = \sum_j^J Y_j \bar{m}_j \phi^j_{yn}$$

$$P_{zn} = \sum_j^J Z_j \bar{m}_j \phi^j_{zn}$$

$$N_{xn} = \sum_j^J \bar{m}_j ({\phi^j_{yn}}^2 + {\phi^j_{zn}}^2)$$

$$N_{yn} = \sum_j^J \bar{m}_j ({\phi^j_{xn}}^2 + {\phi^j_{zn}}^2)$$

$$N_{zn} = \sum_j^J \bar{m}_j ({\phi^j_{xn}}^2 + {\phi^j_{yn}}^2)$$

The potential energy of the center body consists of the potential due to the planet's gravity field and the strain energy due to the center body's elastic deformation.

The potential due to the gravity field is

$$U_g = gM(\text{Cos}\ \zeta\ q_x - \text{Sin}\ \zeta\ q_z) \tag{5-14}$$

and the center body's strain energy is

$$U_s = \frac{1}{2}\sum_n^N m_n\omega_n^2 q_n^2 \tag{5-15}$$

where ω_n is the natural frequency at the nth free-free mode. Combining these expressions, the total potential energy of the center body is

$$U = gM(\text{Cos}\ \zeta\ q_x - \text{Sin}\ \zeta\ q_z) + \frac{1}{2}\sum_n^N m_n\omega_n^2 q_n^2 \tag{5-16}$$

The final form of the center body's equations of motion were obtained by applying the Lagrangian equations to the energy terms of Equations (5-13) and (5-16). The Lagrangian equations are expressed as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_c}\right) - \frac{\partial T}{\partial q_c} + \frac{\partial U}{\partial q_c} = Q_c. \tag{5-17}$$

where

q_c, $\dot{q}_c$ = cth generalized coordinate and generalized velocity (either rigid body or elastic modes).

T = center body's kinetic energy.

U = center body's potential energy.

Q_c = generalized force or moment in cth mode.

The generalized forces of the rigid body translational modes are the combination of the inertia effects of noncontacting footpads and the sum of the landing gear strut forces in the respective axis directions. The generalized moments of the rigid body rotational modes are the sum of the moments at the center body center of gravity, about the center body's axes. These consist of the inertia effects of the noncontacting footpads and moments due to the landing gear strut loads. For the elastic modes, the generalized force in the nth mode is

$$Q_n = \sum_{p}^{P} (F_x^p \phi_{nx}^{\ p} + F_y^p \phi_{ny}^{\ p} + F_z^p \phi_{nz}^{\ p}) \tag{5-18}$$

In the above, p refers to the pth point on the center body where a force is applied. There are a total of P forces, each of which has been resolved into components in the Lander Coordinate System axes.

To evaluate the inertia effects of a noncontacting footpad, consider the position vectors shown in Figure 5-4. The position of a footpad in question is given as

$$\bar{r}_{f_i} = \bar{R} + \bar{f}_i \tag{5-19}$$

where

$\bar{r}_{f_i}$ = position vector of ith noncontacting footpad relative to Surface Coordinate System.

$\bar{R}$ = position vector of center body center of gravity relative to Surface Coordinate System.

$\bar{f}_i$ = position vector of ith noncontacting footpad relative to Lander Coordinate System.

With the assumption that the noncontacting footpad is on an extension of the center body structure, its acceleration, $\ddot{r}_{f_i}$, is expressed as

$$\ddot{\bar{r}}_{f_i} = \ddot{\bar{R}} + (\ddot{\bar{\Omega}} \times \bar{f}) + \dot{\bar{\Omega}} \times (\dot{\bar{\Omega}} \times \bar{f}) \tag{5-20}$$

In the above, the terms $\ddot{\bar{R}}$, $\ddot{\bar{\Omega}}$, and $\dot{\bar{\Omega}}$ are defined in Figure 5-2 and Equation (5-6). Thus, the total inertia loads of all the noncontacting footpads at the center body center of gravity are

$$\bar{F}_f = - \sum_{i}^{I} m_i \ddot{\bar{r}}_{f_i}$$

and (5-21)

$$\bar{T}_f = - \sum_{i}^{I} m_i (\bar{f}_i \times \ddot{\bar{r}}_{f_i})$$

where I is the total number of noncontacting footpads, each with a mass m_i.

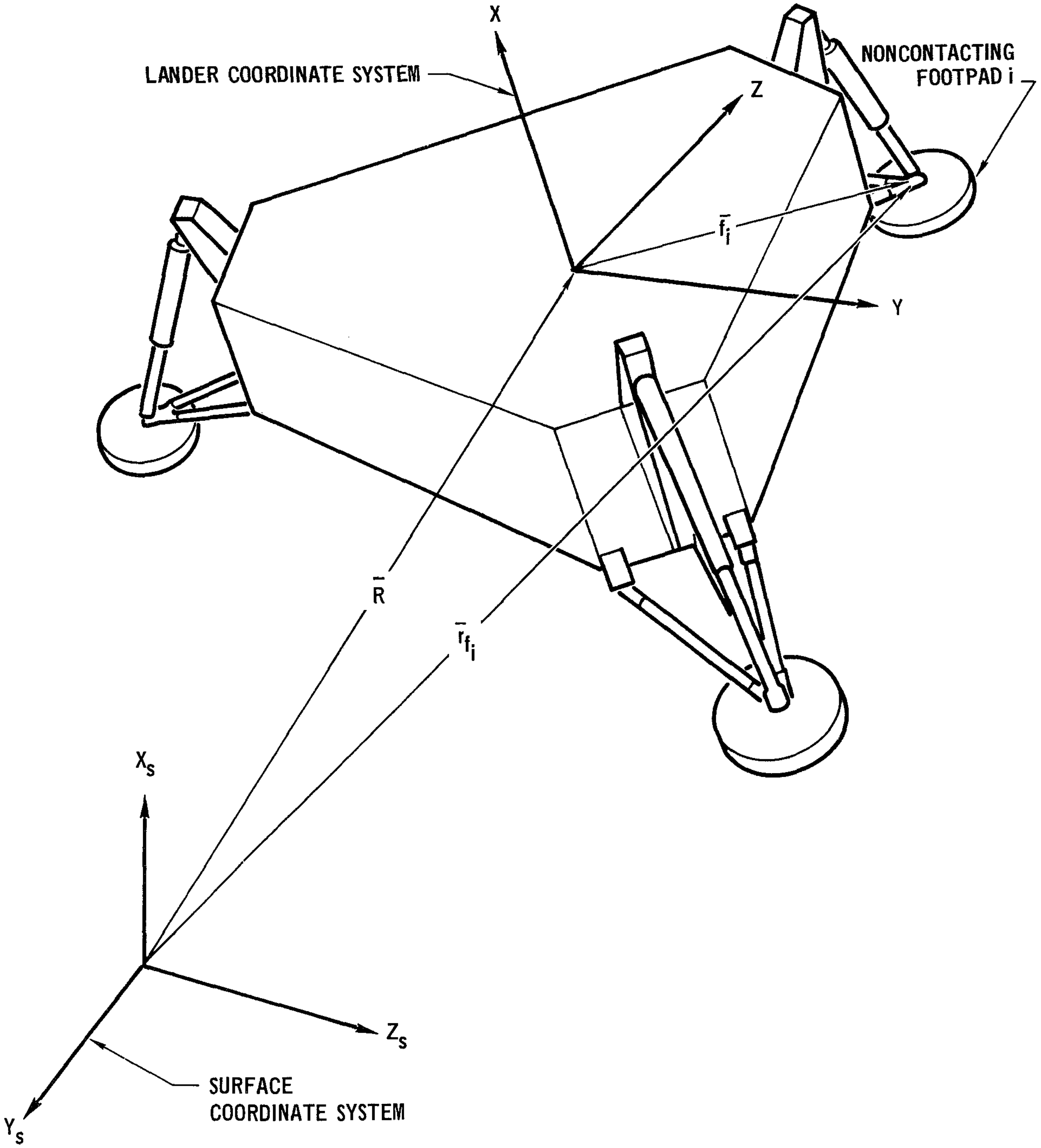


FIGURE 5–4 POSITION OF NONCONTACTING FOOTPAD IN SURFACE COORDINATE SYSTEM

Equations (5-21) may be written as

$$\begin{Bmatrix} F_{xf} \\ F_{yf} \\ F_{zf} \end{Bmatrix} = -\sum_{i}^{I} m_i \begin{Bmatrix} \ddot{q}_x \\ \ddot{q}_y \\ \ddot{q}_z \end{Bmatrix} - \sum_{i}^{I} m_i \, [DC] \begin{bmatrix} 0 & -f_{z_i} & f_{y_i} \\ f_{z_i} & 0 & -f_{x_i} \\ -f_{y_i} & f_{x_i} & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q}_{rx} \\ \ddot{q}_{ry} \\ \ddot{q}_{rz} \end{Bmatrix}$$

(5-22)

$$-\sum_{i}^{I} m_i \, [DC] \begin{Bmatrix} \dot{q}_{ry} (\dot{q}_{rx} f_{y_i} - \dot{q}_{ry} f_{x_i}) - \dot{q}_{rz} (\dot{q}_{rz} f_{x_i} - \dot{q}_{rx} f_{z_i}) \\ \dot{q}_{rz} (\dot{q}_{ry} f_{z_i} - \dot{q}_{rz} f_{y_i}) - \dot{q}_{rx} (\dot{q}_{rx} f_{y_i} - \dot{q}_{ry} f_{x_i}) \\ \dot{q}_{rx} (\dot{q}_{rz} f_{x_i} - \dot{q}_{rx} f_{z_i}) - \dot{q}_{ry} (\dot{q}_{ry} f_{z_i} - \dot{q}_{rz} f_{y_i}) \end{Bmatrix}$$

and

$$\begin{Bmatrix} T_{xf} \\ T_{yf} \\ T_{zf} \end{Bmatrix} = -\sum_{i}^{I} m_i \begin{bmatrix} (f_{y_i}{}^2 + f_{z_i}{}^2) & -f_{x_i} f_{y_i} & -f_{x_i} f_{z_i} \\ -f_{x_i} f_{y_i} & (f_{x_i}{}^2 + f_{z_i}{}^2) & -f_{y_i} f_{z_i} \\ -f_{x_i} f_{z_i} & -f_{y_i} f_{z_i} & (f_{x_i}{}^2 + f_{y_i}{}^2) \end{bmatrix} \begin{Bmatrix} \ddot{q}_{rx} \\ \ddot{q}_{ry} \\ \ddot{q}_{rz} \end{Bmatrix}$$

$$-\sum_{i}^{I} m_i \begin{bmatrix} 0 & -\tilde{f}_{z_i} & \tilde{f}_{y_i} \\ \tilde{f}_{z_i} & 0 & -\tilde{f}_{x_i} \\ -\tilde{f}_{y_i} & \tilde{f}_{x_i} & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q}_x \\ \ddot{q}_y \\ \ddot{q}_z \end{Bmatrix}$$

(5-22) (Continued)

$$-\sum_{i}^{I} m_i \begin{Bmatrix} f_{x_i} \dot{q}_{rx} (f_{y_i} \dot{q}_{rz} - f_{z_i} \dot{q}_{ry}) + (f_{y_i}{}^2 - f_{z_i}{}^2) \dot{q}_{ry} \dot{q}_{rz} + f_{y_i} f_{z_i} (\dot{q}_{rz}{}^2 - \dot{q}_{ry}{}^2) \\ f_{y_i} \dot{q}_{ry} (f_{z_i} \dot{q}_{rx} - f_{x_i} \dot{q}_{rz}) + (f_{z_i}{}^2 - f_{x_i}{}^2) \dot{q}_{rx} \dot{q}_{rz} + f_{x_i} f_{z_i} (\dot{q}_{rx}{}^2 - \dot{q}_{rz}{}^2) \\ f_{z_i} \dot{q}_{rz} (f_{x_i} \dot{q}_{ry} - f_{y_i} \dot{q}_{rx}) + (f_{x_i}{}^2 - f_{y_i}{}^2) \dot{q}_{rx} \dot{q}_{ry} + f_{x_i} f_{y_i} (\dot{q}_{ry}{}^2 - \dot{q}_{rx}{}^2) \end{Bmatrix}$$

where

$$\begin{Bmatrix} \tilde{f}_{x_i} \\ \tilde{f}_{y_i} \\ \tilde{f}_{z_i} \end{Bmatrix} = [DC] \begin{Bmatrix} f_{x_i} \\ f_{y_i} \\ f_{z_i} \end{Bmatrix}$$

are the components of the ith footpad location, from the center body center of gravity, in the Surface Coordinate System and

f_{x_i}, f_{y_i}, f_{z_i} = components of position vector locating ith noncontacting footpad in the Lander Coordinate System

[DC] = direction cosine matrix, Equation (5-1), relating Lander Coordinate System to Surface Coordinate System.

F_{xf}, F_{yf}, F_{zf} = inertia forces due to noncontacting footpads, relative to Surface Coordinate System.

T_{xf}, T_{yf}, T_{zf} = inertia moments due to noncontacting footpad, relative to Lander Coordinate System.

Using the center body energy expressions, Equations (5-13) and (5-16), and the noncontacting footpad inertia loads, Equation (5-22), in conjunction with the Lagrangian equation leads to the center body equations of motion given in Equation (5-23).

5.1.3 Landing Gear Strut Idealization - The two landing gear configuration options available in the Landing Loads and Motions Program are applicable to the inverted tripod gear and the cantilever gear. Each landing gear consists of a main strut and two drag struts. It is assumed that these are pinned struts and thus moments or torsion are not introduced at their ends. Both the main strut and the drag struts are capable of carrying tension and compression loads and may possess either velocity dependent energy absorption characteristics, stroke dependent characteristics, or a combination of the two. The energy absorption characteristics of all main struts in a given lander are the same. Similarly, characteristics of all the drag struts are the same, however they may be different than those of the main struts. For a cantilever gear, the effect of bending in the main strut is included by altering the energy absorption properties of the drag struts.

Relative motion between each footpad and the center body is used to determine the stroke and velocity of stroke in each strut. This information is used to determine the load in a landing gear strut employing the subroutine STRUT.

5.1.3.1 Stroke Dependent Attenuation - The primary type of energy absorption mechanism for the landing gear system consists of crushable

$$\left[\begin{array}{ccc|ccc}
M+\sum_{i}^{I} m_i & 0 & 0 & \multicolumn{3}{c}{[\,DC\,]\begin{bmatrix} 0 & -\sum_{i}^{I} m_i f_{z_i} & \sum_{i}^{I} m_i f_{y_i} \\ \sum_{i}^{I} m_i f_{z_i} & 0 & -\sum_{i}^{I} m_i f x_i \\ -\sum_{i}^{I} m_i f_{y_i} & \sum_{i}^{I} m_i f_{x_i} & 0 \end{bmatrix}} \\
0 & M+\sum_{i}^{I} m_i & 0 & & & \\
0 & 0 & M+\sum_{i}^{I} m_i & & & \\
\hline
0 & -\sum_{i}^{I} m_i \tilde{f}_{z_i} & \sum_{i}^{I} m_i \tilde{f}_{y_i} & Ixx+\sum_{i}^{I} m_i (f_{y_i}{}^2+f_{z_i}{}^2)+\sum_{n}^{N} |\,2\,(P_{yn}+P_{zn})+N_{xn}\,q_n\,|\,q_n & -Ixy-\sum_{i}^{I} m_i f_{x_i} f_{y_i} & -Ixz-\sum_{i}^{I} m_i f_{x_i} f_{z_i} \\
\sum_{i}^{I} m_i \tilde{f}_{z_i} & 0 & -\sum_{i}^{I} m_i \tilde{f}_{x_i} & -\,Ixy-\sum_{i}^{I} m_i f_{x_i} f_{y_i} & Iyy+\sum_{i}^{I} m_i (f_{x_i}{}^2+f_{z_i}{}^2)+\sum_{n}^{N} |\,2\,(P_{xn}+P_{zn})+N_{yn}\,q_n\,|\,q_n & -Iyz-\sum_{i}^{I} m_i f_{y_i} f_{z_i} \\
-\sum_{i}^{I} m_i \tilde{f}_{y_i} & \sum_{i}^{I} m_i \tilde{f}_{x_i} & 0 & -\,Ixz-\sum_{i}^{I} m_i f_{x_i} f_{z_i} & -Iyz-\sum_{i}^{I} m_i f_{y_i} f_{z_i} & Izz+\sum_{i}^{I} m_i (f_{x_i}{}^2+f_{y_i}{}^2)+\sum_{n}^{N} |\,2\,(P_{xn}+P_{yn})+N_{zn}\,q_n\,|\,q_n
\end{array}\right]
\left\{\begin{array}{c} \ddot{q}_x \\ \ddot{q}_y \\ \ddot{q}_z \\ \hline \ddot{q}_{rx} \\ \ddot{q}_{ry} \\ \ddot{q}_{rz} \end{array}\right\}
=
\left\{\begin{array}{c} F_x - gM\ \mathrm{COS}\,\zeta \\ F_y \\ F_z + gM\ \mathrm{SIN}\,\zeta \\ \hline T_x \\ T_y \\ T_z \end{array}\right\}$$

$$-\left\{\begin{array}{c}
-[\,DC\,]\left\{\begin{array}{l}
\sum_{i}^{I} m_i |\,\dot{q}_{ry}\,(\dot{q}_{rx} f_{y_i}-\dot{q}_{ry} f_{x_i})-\dot{q}_{rz}\,(\dot{q}_{rz} f_{x_i}-\dot{q}_{rx} f_{z_i})\,| \\
\sum_{i}^{I} m_i |\,\dot{q}_{rz}\,(\dot{q}_{ry} f_{z_i}-\dot{q}_{rz} f_{y_i})-\dot{q}_{rx}\,(\dot{q}_{rx} f_{y_i}-\dot{q}_{ry} f_{x_i})\,| \\
\sum_{i}^{I} m_i |\,\dot{q}_{rx}\,(\dot{q}_{rz} f_{x_i}-\dot{q}_{rx} f_{z_i})-\dot{q}_{ry}\,(\dot{q}_{ry} f_{z_i}-\dot{q}_{rz} f_{y_i})\,|
\end{array}\right\} \\
\hline
\dot{q}_{rx}\dot{q}_{rz}\,(Ixy+\sum_{i}^{I} m_i f_{x_i} f_{y_i})-\dot{q}_{rx}\dot{q}_{ry}\,(Ixz+\sum_{i}^{I} m_i f_{x_i} f_{z_i})+(\dot{q}_{rz}{}^2-\dot{q}_{ry}{}^2)(Iyz+\sum_{i}^{I} m_i f_{y_i} f_{z_i})+\dot{q}_{ry}\dot{q}_{rz}\,|\,Izz-Iyy+\sum_{i}^{I} m_i\,(f_{y_i}{}^2-f_{z_i}{}^2)\,|+2\,\dot{q}_{rx}\sum_{n}^{N} |\,(P_{yn}+P_{zn})+N_{xn}\,q_n|\,\dot{q}_n+\dot{q}_{ry}\dot{q}_{rz}\sum_{n}^{N} |\,2\,(P_{yn}-P_{zn})+(N_{zn}-N_{yn})q_n|\,q_n \\
\dot{q}_{rx}\dot{q}_{ry}\,(I_{yz}+\sum_{i}^{I} m_i f_{y_i} f_{z_i})-\dot{q}_{ry}\dot{q}_{rz}\,(Ixy+\sum_{i}^{I} m_i f_{x_i} f_{y_i})+(\dot{q}_{rx}-\dot{q}_{rz}{}^2)(\,Ixz+\sum_{i}^{I} m_i f_{x_i} f_{z_i})+\dot{q}_{rx}\dot{q}_{rz}\,|\,Ixx-Izz+\sum_{i}^{I} m_i\,(f_{z_i}{}^2-f_{x_i}{}^2)\,|+2\,\dot{q}_{ry}\sum_{n}^{N} |\,(P_{xn}+P_{zn})+N_{yn}\,q_n|\,\dot{q}_n+\dot{q}_{rx}\dot{q}_{rz}\sum_{n}^{N} |\,2\,(P_{zn}-P_{xn})+(N_{xn}-N_{zn})\,q_n|\,q_n \\
\dot{q}_{ry}\dot{q}_{rz}\,(Iyz+\sum_{i}^{I} m_i f_{x_i} f_{z_i})-\dot{q}_{rx}\dot{q}_{rz}\,(Iyz+\sum_{i}^{I} m_i f_{y_i} f_{z_i})+(\dot{q}_{ry}{}^2-\dot{q}_{rx}{}^2)\,(\,Ixy+\sum_{i}^{I} m_i f_{x_i} f_{y_i})+\dot{q}_{rx}\dot{q}_{ry}\,|\,Iyy-Ixx+\sum_{i}^{I} m_i\,(f_{x_i}{}^2-f_{y_i}{}^2)|+2\,\dot{q}_{rz}\sum_{n}^{N} |\,P_{xn}+P_{yn})+N_{zn}\,q_n|\,\dot{q}_n+\dot{q}_{rx}\dot{q}_{ry}\sum_{n}^{N} |\,2\,(P_{xn}-P_{yn})+(N_{yn}-N_{xn})q_n|\,q_n
\end{array}\right\} \qquad (5\text{–}23)$$

AND

$$m_n\,\ddot{q}_n+\omega_n{}^2\,m_n\,q_n=Q_n+\dot{q}_{rx}{}^2\,|\,(P_{yn}+P_{zn})+N_{xn}\,q_n|+\dot{q}_{ry}{}^2\,|\,(P_{xn}+P_{zn})+N_{yn}\,q_n\,|+\dot{q}_{rz}{}^2\,|\,(P_{xn}+P_{yn})+N_{zn}\,q_n\,|\quad \mathrm{FOR}\ n=1,2,\ldots\ N$$

cartridges housed inside the landing gear struts. Each cartridge crushes at a constant load when the landing gear strut is stroked. Several cartridges, possessing different crushing strengths, may be stacked in series to form a desired load-stroke characteristic. In the present study, all struts may contain a maximum of five cartridges to attenuate compression loads and five to attenuate tension loads.

The idealization of the crushable landing gear struts in the Landing Loads and Motions Program is the same as is employed in the Landing Gear Option of the Structural Analysis Program. A discussion explaining the procedure for determining strut loads during strut stroking is presented in Section 4.1.2.2. In addition, a typical load-stroke relationship for a strut is shown in Figure 4-18.

5.1.3.2 Main Strut Bending - For a cantilever gear, the drag strut loads acting normal to the main strut in combination with the lateral loads on the footpad cause lateral deflections of the main strut. The assumption is made that the effect of this main strut deflection can be approximated through a modification of the drag strut load-stroke relationship.

The main strut is idealized as a simply supported beam whose elastic axis is defined by simple beam theory. Neglecting the effect of axial loads on the lateral deflection of the main strut, the deflected main strut is shown in Figure 5-5. The force F_n is the component of the drag strut force normal to the main strut axis. This force is defined as

$$F_n \bar{n} = F(\bar{s} \cdot \bar{n}) \bar{n} \qquad (5\text{-}24)$$

where F is the force in the drag strut, $\bar{n}$ is a unit vector normal to the main strut, and $\bar{s}$ is a unit vector in the direction of the drag strut. The flexural stiffness of the main strut on either side of the drag strut attach point is defined as EI_1 and EI_2.

The lateral deflection of the main strut due to the load F_n is expressed as

$$\Delta \bar{n} = F_n \left[\frac{1}{EI_1} \left(\frac{a^3 b}{3\ell} - \frac{a^4 b}{3\ell^2} \right) + \frac{1}{EI_2} \left(\frac{a^4}{\ell} - \frac{a^5}{3\ell^2} + \frac{a^2 \ell}{3} - a^3 \right) \right] \bar{n} \qquad (5\text{-}25)$$

The magnitude of this deflection, λ , in the direction of the drag strut axis is

$$\lambda\bar{s} = \Delta(\bar{s}\cdot\bar{n})\bar{s} \tag{5-26}$$

A spring constant reflecting the bending deflection of the main strut is defined as

$$K_B = \frac{F}{\lambda} \tag{5-27}$$

This expression is evaluated with the aid of Equations (5-24), (5-25) and (5-26) and results in

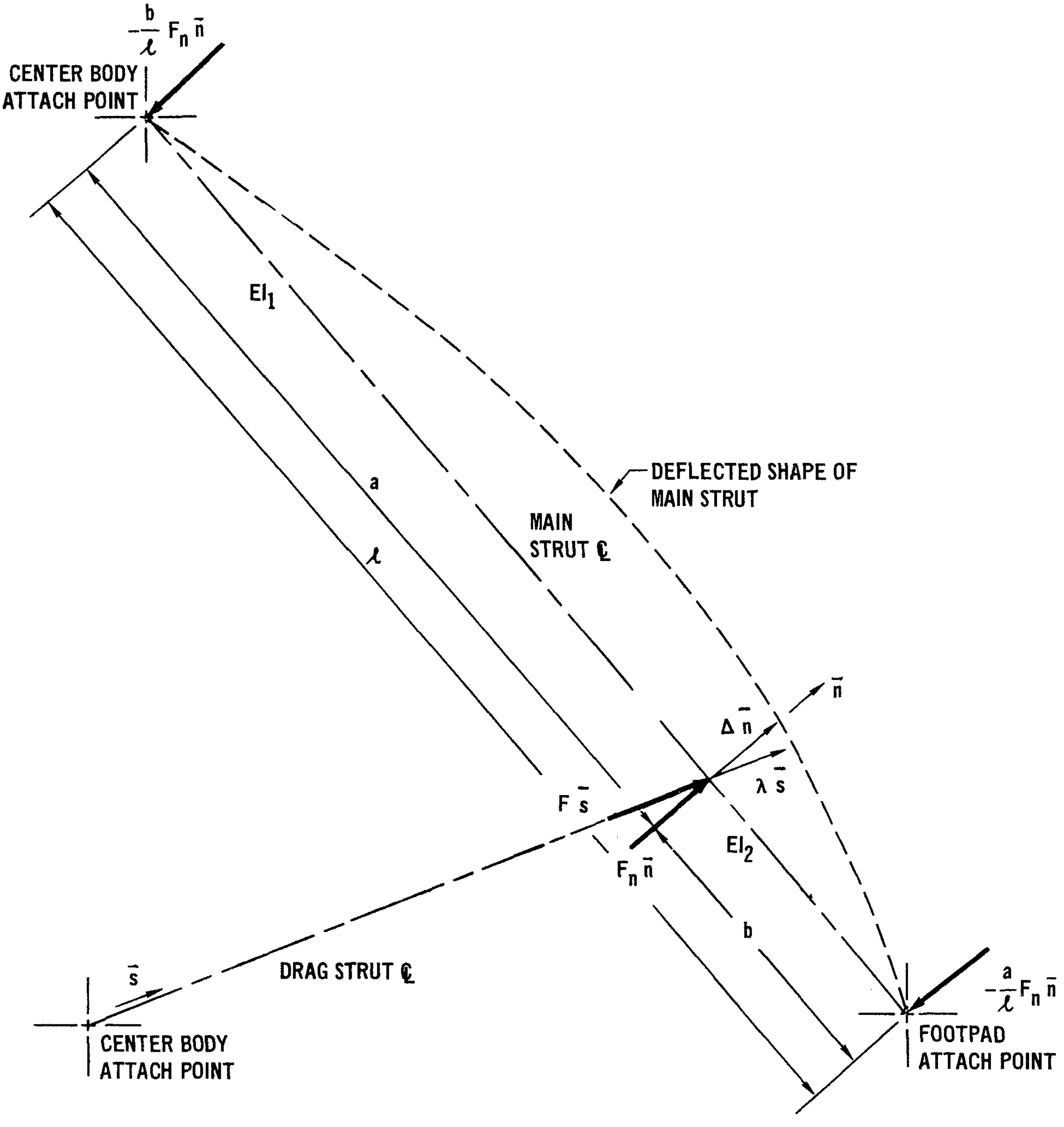
$$K_B = \frac{1}{(\bar{s}\cdot\bar{n})^2\left[\frac{1}{EI_1}\left(\frac{a^3b}{3\ell} - \frac{a^4b}{3\ell^2}\right) + \frac{1}{EI_2}\left(\frac{a^4}{\ell} - \frac{a^5}{3\ell^2} + \frac{a^2\ell}{3} - a^3\right)\right]} \tag{5-28}$$

This linear spring is assumed to be in series with the spring defining each elastic portion of the load-stroke diagram of the drag strut. A modified spring constant for these elastic portions of the load-stroke curve is calculated to reflect the main strut bending. This modified spring constant is used to define a modified load-stroke relationship for the drag strut. A typical drag strut modified load-stroke curve is shown in Figure 5-6. This modified curve is then employed to define the loads in the drag strut.

During stroking of the main strut, its bending characteristics change due to the change in strut length. These changes in bending characteristics are incorporated by continually modifying the spring constant of the drag strut load-stroke curve. As a result the drag strut load-stroke relationship is continually updated to reflect these changes in main strut bending characteristics.

5.1.3.3 Velocity Dependent Attenuation - Provisions are available for the inclusion of a constant friction force and a velocity dependent damping force in each strut. Both of these are applied in a direction opposite to the velocity of the stroking motion in the strut. The combination of these force terms is expressed as

$$F_v = -\frac{\dot{S}}{|\dot{S}|}\left[F_r + C\gamma\,|\dot{S}|^{\gamma}\right] \tag{5-29}$$

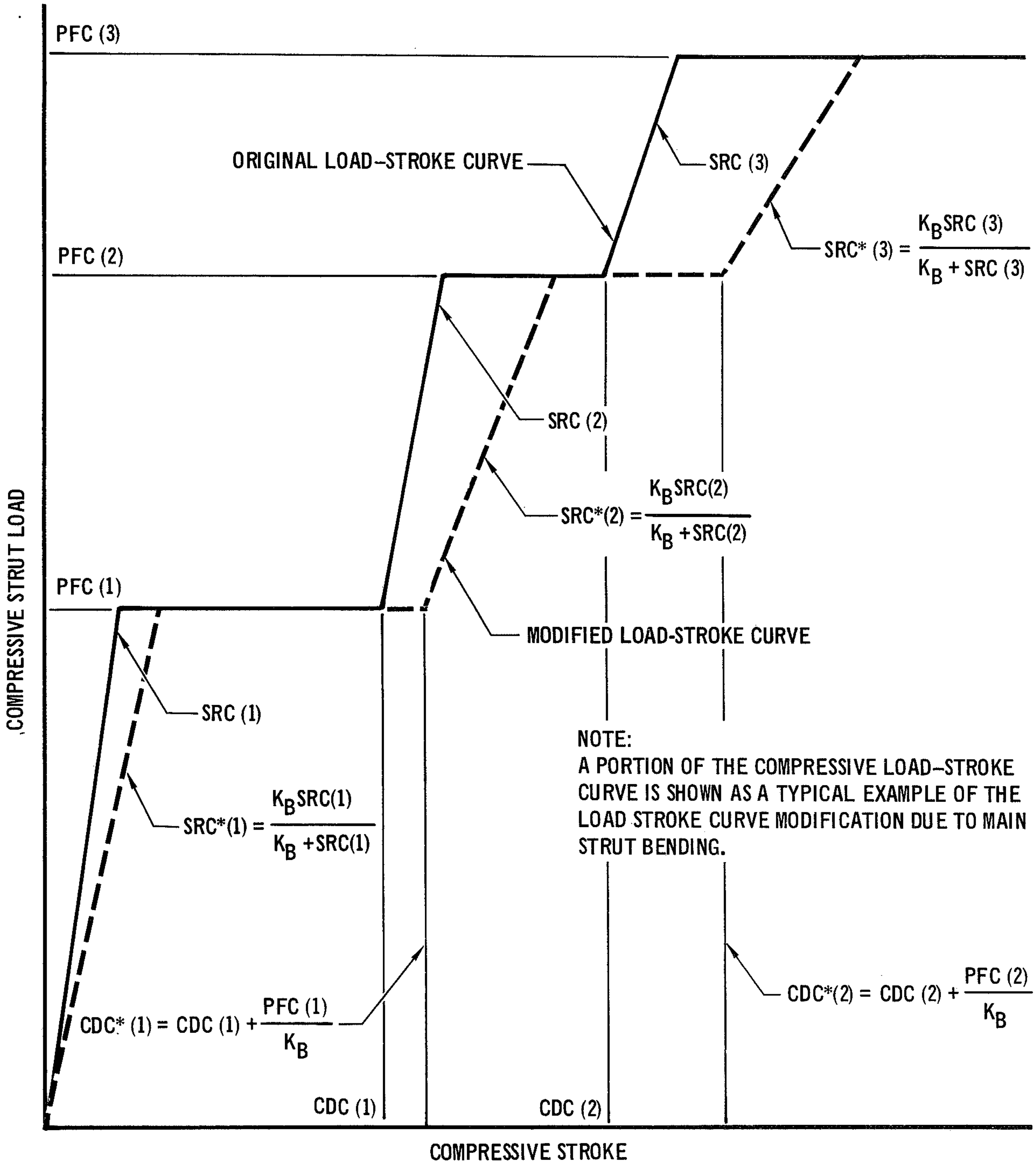


FOLLOWING RELATIONSHIPS HOLD:

$F_n \bar{n} = F(\bar{s} \cdot \bar{n}) \bar{n}$

$\lambda \bar{s} = \Delta (\bar{n} \cdot \bar{s}) \bar{s}$

FIGURE 5–5 DEFLECTION OF CANTILEVER GEAR MAIN STRUT



NOTE:
A PORTION OF THE COMPRESSIVE LOAD-STROKE CURVE IS SHOWN AS A TYPICAL EXAMPLE OF THE LOAD STROKE CURVE MODIFICATION DUE TO MAIN STRUT BENDING.

FIGURE 5-6 MODIFIED DRAG STRUT LOAD-STROKE CURVE where

F_v = total velocity dependent force along axis of strut.

$\dot{S}$ = stroking velocity of strut.

F_r = magnitude of constant friction force (input quantity).

$C\gamma$ = coefficient of velocity proportional force (input quantity).

γ = power of velocity in velocity proportional force (input quantity).

This force is superimposed on the crushing force or may be included by itself.

5.1.4 Soil Mechanics - Two methods of representing the footpad-soil interaction are incorporated in the Landing Loads and Motions Program. The first of these, referred to as the Primary Soil Mechanics Method, is a modification of the soil mechanics analyses developed during the Lunar Module (LM) soil mechanics studies. This modification is similar to that employed during the Surveyor analysis. An alternate soil mechanics method, referred to as the Secondary Soil Mechanics Method, determines the soil force through a simple elastic-plastic relationship between soil pressure and depth of soil penetration. On an optional basis, a crushable footpad attenuation system located on the bottom of the footpads may be included with either of these soil mechanics routines. The footpad attenuation system is discussed in Section 5.1.5. Symbols employed in the soil mechanics routines are defined in Figure 5-7. The soil forces are determined in the subroutine SOIL.

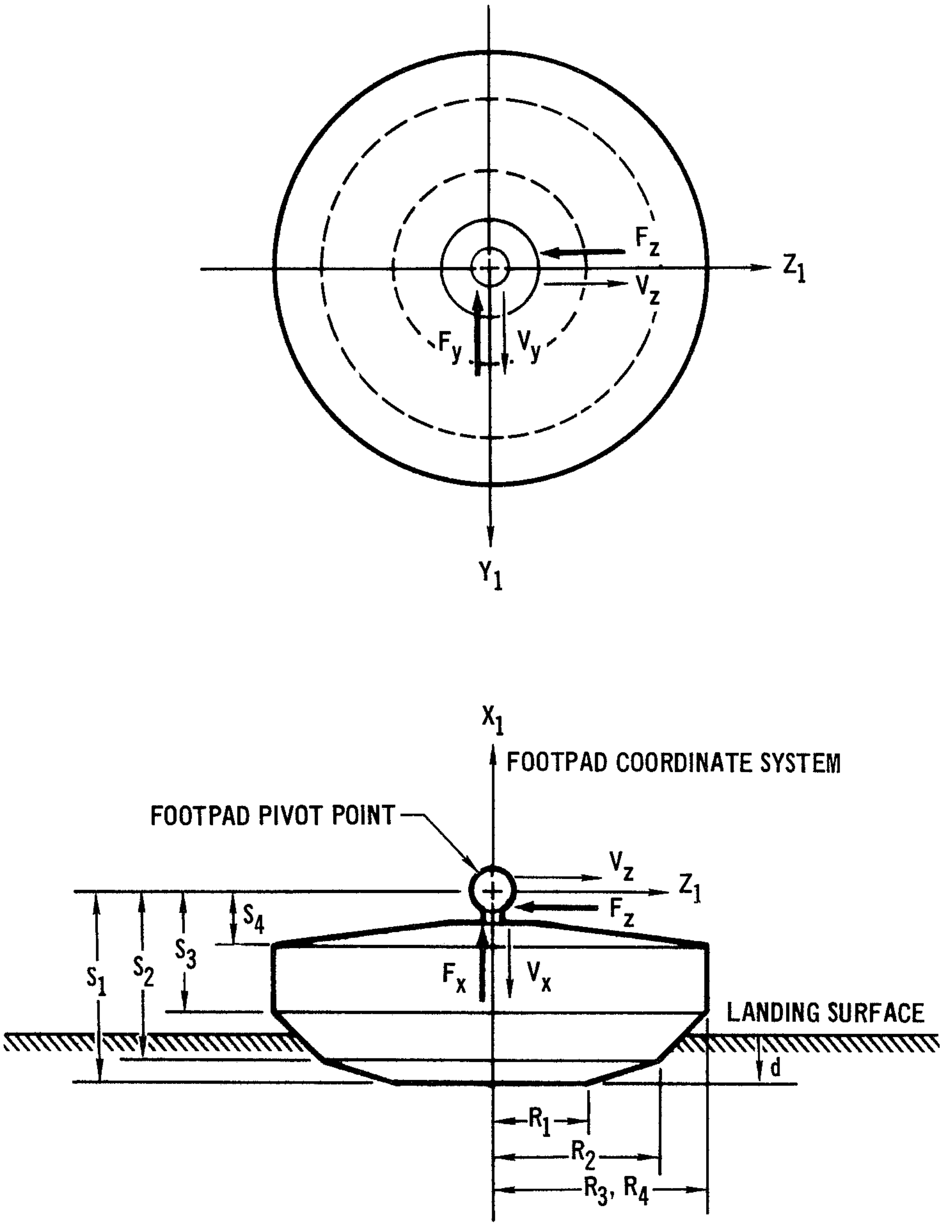
The surface of each footpad is represented by a number of concentric conical and/or cylindrical segments as shown in Figure 5-8. It is assumed that the footpad is always aligned with the plane of the landing surface. Upon entering the soil mechanics routine, the footpad velocities and soil penetration indicated in Figure 5-8 are available. The three components of the soil force in the Footpad Coordinate System are returned from the soil mechanics routine.

5.1.4.1 Primary Soil Mechanics - The footpad-soil interaction method developed in Reference (11) for the LM shaped footpad utilized principles which are fundamental to the interaction phenomenon occurring during soil penetration. Applicability of this basic method to a different footpad shape was demonstrated by the good agreement obtained between telemetered Surveyor lunar impact data and predicted landing dynamics as reported in Reference (12).

SYMBOL	DEFINITION	UNITS *
A	FOOTPAD AREA PROJECTED ON LANDING SURFACE	L^2
A_p	FOOTPAD AREA PROJECTED ON PLANE PARALLEL TO PLANE DEFINED BY SURFACE NORMAL AND VELOCITY V_{ap}	L^2
A_Θ	FOOTPAD AREA PROJECTED ON PLANE NORMAL TO VELOCITY V_{ap}	L^2
C_{ms}	SOIL DYNAMIC MECHANICAL STRENGTH COEFFICIENT	--
C_d	SOIL DRAG COEFFICIENT	--
d	DEPTH OF SOIL PENETRATION	L
dr/dd	CHANGE IN FOOTPAD RADIUS WITH RESPECT TO SOIL PENETRATION	--
D_r	RELATIVE DENSITY OF SOIL (NO COMPACTION, $D_r = 0$; MAXIMUM COMPACTION, $D_r = 1$)	--
F_{ap}	SOIL FORCE PARALLEL TO FOOTPAD VELOCITY	F
F_{np}	SOIL FORCE NORMAL TO FOOTPAD VELOCITY	F
F_x, F_y, F_z	COMPONENTS OF SOIL FORCE IN FOOTPAD COORDINATE SYSTEM	F
g	LOCAL ACCELERATION OF GRAVITY	L/T^2
g_e	EARTH ACCELERATION OF GRAVITY	L/T^2
r	FOOTPAD RADIUS AT FOOTPAD–LANDING SURFACE INTERSECTION	L
r_m	MAXIMUM FOOTPAD RADIUS	L
R_1, R_2, R_3, R_4	PARAMETERS DEFINING FOOTPAD SHAPE (SEE FIGURE 5–8)	L
S_1, S_2, S_3, S_4	PARAMETERS DEFINING FOOTPAD SHAPE (SEE FIGURE 5–8)	L
V_{ap}	TOTAL VELOCITY OF FOOTPAD	L/T
V_x, V_y, V_z	COMPONENTS OF FOOTPAD VELOCITY IN FOOTPAD COORDINATE SYSTEM	L/T
γ	SOIL UNIT WEIGHT	F/L^3
η	WEDGE SHAPE FACTOR FOR MOVING SOIL MASS	--
Θ	ANGLE DEFINING DIRECTION OF VELOCITY V_{ap} RELATIVE TO SURFACE NORMAL	A
λ	RATIO OF AXIAL TO NORMAL SOIL FORCE	--
ρ	BULK MASS DENSITY OF SOIL	FT^2/L^4
Φ	INTERNAL FRICTION ANGLE OF SOIL	A

* UNITS:
L – LENGTH
F – FORCE
T – TIME
A – ANGLE

FIGURE 5–7 SYMBOLS FOR SOIL MECHANICS ROUTINE



d = DEPTH OF SOIL PENETRATION

V_x, V_y, V_z = VELOCITIES OF FOOTPAD IN FOOTPAD COORDINATE SYSTEM

F_x, F_y, F_z = COMPONENTS OF SOIL FORCE IN FOOTPAD COORDINATE SYSTEM

FIGURE 5–8 FOOTPAD REPRESENTATION

In Reference (11) a theory of soil elasto-plastic deformation is used to define the force between the footpad and the deformed soil surface. The soil mass, displaced by the moving footpad, is considered as a degree of freedom independent of the lander system. A spring, representing the soil elasticity, is placed between the footpad and the soil mass. Additional external forces applied to the soil mass represent a momentum transfer force and a force due to the soil strength.

It was shown in the Surveyor simulation that sufficient accuracy can be obtained by neglecting soil elasticity and assuming that the moving soil mass is attached rigidly to the footpad. This simplification results in the removal of the soil mass differential equation from the analysis. Therefore, the soil force acting on the footpad is considered to be the sum of a soil strength term, a soil drag term, and a term approximating the effect of the changing soil mass.

The Primary Soil Mechanics Method employs the empirical relationships discussed in Reference (12). Forces acting on each footpad consist of an axial force, F_{ap}, parallel to the velocity vector of the footpad and a force, F_{np}, normal to the velocity vector.

The axial force is the sum of the forces due to the soil strength, soil drag, and effect of the changing soil mass and is expressed as

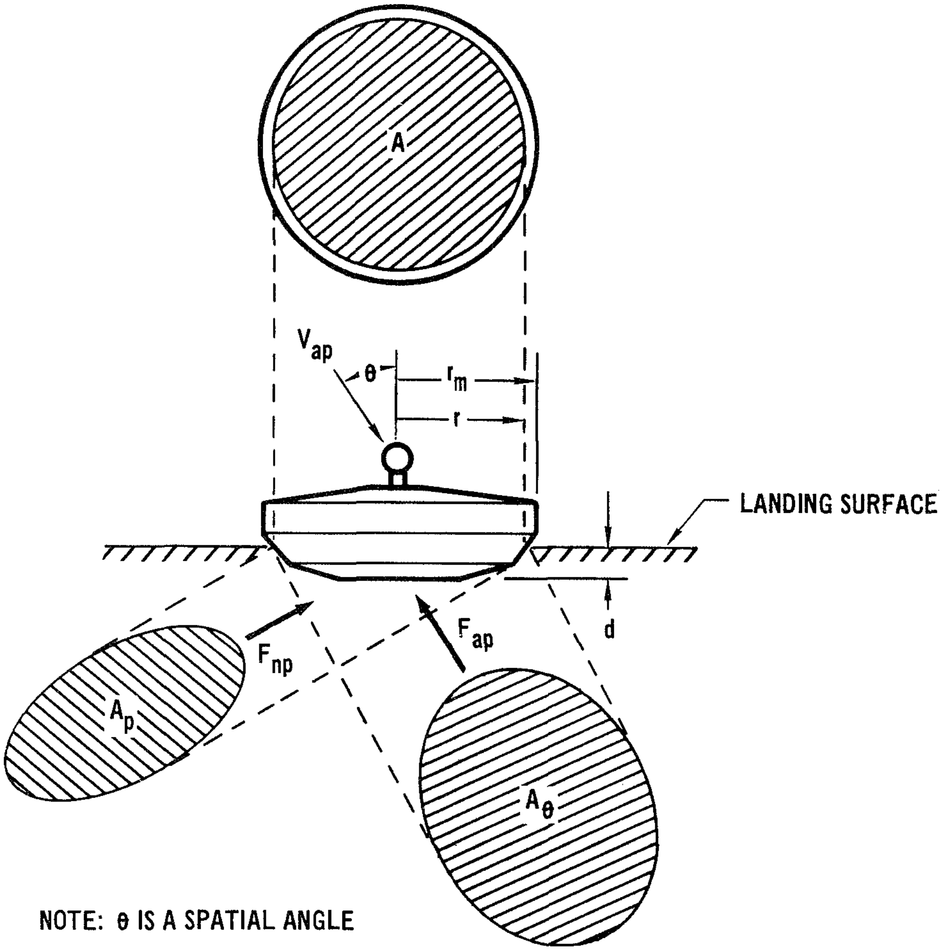
$$F_{ap} = C_{ms}\rho g d A_\theta + C_d \rho A_\theta V_{ap}^2 + 3\eta\rho r^2 \frac{dr}{dd}\left(\frac{A_\theta}{A}\right)^{3/2} V_{ap}^2 \operatorname{Cos}\theta \qquad (5\text{-}30)$$

Coefficients C_{ms} and C_d, representing the soil dynamic mechanical strength and drag coefficients respectively, are empirical factors determined from test and discussed further in later paragraphs. The wedge shape factor for moving soil mass, η, is a function of angle of internal friction of the soil, and is also discussed further in the section on empirical relationships. The term $(A_\theta/A)^{3/2}$ accounts for the effect of the angle θ. Change in footpad radius with respect to depth of the footpad, dr/dd, is defined by footpad geometry at the current depth of soil penetration.

The force acting on a footpad normal to the velocity vector is

$$F_{np} = \lambda F_{ap} \qquad (5\text{-}31)$$

This force is always directed out of the soil and is in the plane defined by the surface normal and the velocity vector. The quantity λ is discussed in the paragraphs defining the empirical relationships. Figure 5-9 presents a diagramatic representation of these two soil forces.



NOTE: θ IS A SPATIAL ANGLE

FIGURE 5-9 SOIL FORCES PRIMARY SOIL MECHANICS METHOD

Empirical relationships are determined in terms of the soil properties from impact and drag tests conducted during the LM study. Properties of twelve soils are described in Figure 5-10. The most significant of these properties are: unit weight, γ (bulk mass density, ρ, times g_e); relative density, D_r; the angle of internal friction, ϕ; and, to a lesser degree, the elastic modulus of the soil, E. Based on results reported in Reference (12), the properties γ, D_r and ϕ are adequate to describe the soil for landing dynamics and are used in the present study. In the soil mechanics subroutine, the empirical relationships for C_{ms}, C_d, λ, and η used for the Surveyor footpad, are employed during the determination of the soil forces acting on each footpad. These relationships are presented in Figure 5-11.

NO.	BENDIX DESIGNATION	DESCRIPTION	RELATIVE DENSITY D_r	UNIT * WEIGHT γ dynes/cm^3	FRICTION ANGLE ϕ deg.	ELASTIC MODULUS E dynes/cm^2
1	RS LOOSE	RED CRUSHED VOLCANIC SCORIA (NARROWLY GRADED)	0	650	40	379×10^6
2	PS LOOSE	WHITE CRUSHED PUMICE (NARROWLY GRADED)	0	374	43	310×10^6
3	RS INTER	RED CRUSHED VOLCANIC SCORIA (NARROWLY GRADED)	.45	721	44.5	593×10^6
4	RS DENSE	RED CRUSHED VOLCANIC SCORIA(NARROWLY GRADED)	.80	795	47.3	752×10^6
5	RSM-a LOOSE	MIXTURE OF RS AND CRUSHED MARBLE (MS) (NARROWLY GRADED)	0	914	37	538×10^6
6	RC2 LOOSE	RED CRUSHED VOLCANIC SCORIA(BROADLY GRADED)	0	965	43	552×10^6
7	SS LOOSE	WHITE SILICA SAND (WEDRON 40 40–NARROWLY GRADED)	0	1488	29	690×10^6
8	SS INTER.	WHITE SILICA SAND (WEDRON 40 40–NARROWLY GRADED)	.53	1634	36.8	1241×10^6
9	SS DENSE	WHITE SILICA SAND (WEDRON 40 40–NARROWLY GRADED)	.69	1681	39	1655×10^6
10	LSM INTER.	MIXTURE OF RC AND AIR-FLOATED CLAY	.50	1217	42	276×10^6
11	LSM DENSE	MIXTURE OF RC AND AIR-FLOATED CLAY	.70	1288	42	414×10^6
12	RSM–b DENSE	MIXTURE OF RS AND CRUSHED MARBLE (MS) (NARROWLY GRADED)	.75	1335	48	1380×10^6

*TO OBTAIN BULK MASS DENSITY OF SOIL, DIVIDE THIS QUANTITY BY EARTH ACCELERATION OF GRAVITY.
THIS INFORMATION OBTAINED FROM REFERENCE (11).

FIGURE 5–10 PROPERTIES OF SOILS

WEDGE SHAPE FACTOR FOR MOVING SOIL MASS:

$$\eta = \frac{\pi}{3} \; \frac{1 + \mathrm{TAN}\,\phi/2}{1 - \mathrm{TAN}\,\phi/2}$$

SOIL DYNAMIC MECHANICAL STRENGTH COEFFICIENT:

$$C_{ms} = 29\,e^{1.4\,D_r}\,\mathrm{TAN}\,\phi$$

SOIL DRAG COEFFICIENT:

$$C_d = 0.8 + \left(\frac{g}{g_e}\right)(4 + 80\,D_r)\left(\frac{r}{r_m}\right)^2 f(\theta)\,\mathrm{TAN}\,\phi \quad \mathrm{FOR}\ D_r < 0.5$$

$$C_d = 0.8 + 4\left(\frac{g}{g_e}\right) e^{4.83\,D_r}\left(\frac{r}{r_m}\right)^2 f(\theta)\,\mathrm{TAN}\,\phi \quad \mathrm{FOR}\ D_r \geq 0.5$$

WHERE:

$$f(\theta) = 1 - \frac{2\theta}{\pi} \quad \mathrm{FOR}\ 0 \leq \theta < 45^0$$

$$f(\theta) = 0 \quad \mathrm{FOR}\ \theta \geq 45^0$$

RATIO OF AXIAL TO NORMAL SOIL FORCE:

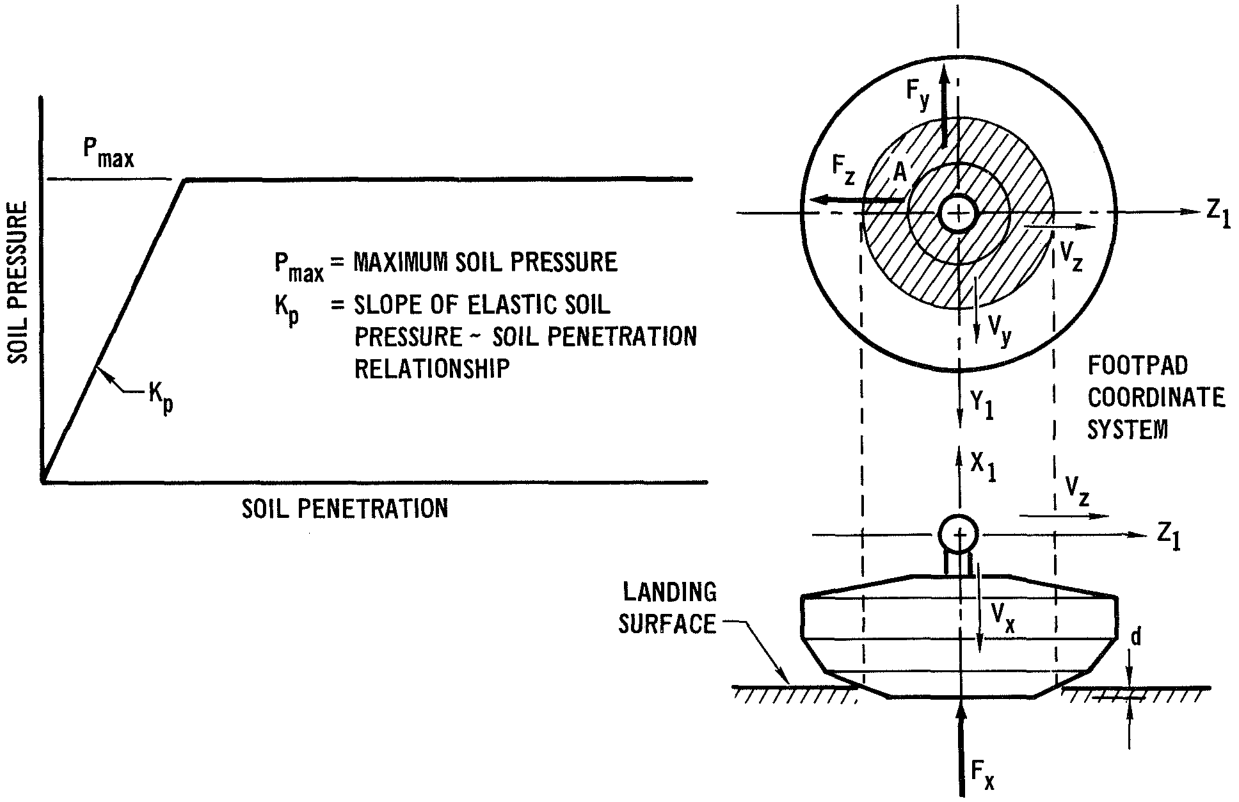
$$\lambda = 0.25\left(\frac{A_p}{A_\theta}\right)(1 - e^{-50\theta})(1 + \mathrm{SIN}\,\theta) \quad \mathrm{FOR}\ 0 \leq \theta < 90^0$$

$$\lambda = 0.5\left(\frac{A_p}{A_\theta}\right) \quad \mathrm{FOR}\ 90^0 \leq \theta < 135^0$$

$$\lambda = 0 \quad \mathrm{FOR}\ \theta \geq 135^0$$

FIGURE 5-11 EMPIRICAL SOIL RELATIONSHIPS

5.1.4.2 Secondary Soil Mechanics - An alternate soil mechanics routine is available in the Landing Loads and Motions Program. This method determines the pressure acting on a footpad in terms of depth of penetration of the footpad. The pressure-penetration relationship is defined as shown in Figure 5-12. Initially the soil pressure increases linearly from zero at zero penetration to a selected pressure at a specified cutoff depth. Beyond this depth, the pressure remains constant. The normal soil force is the product of the pressure determined from the relationship shown in Figure 5-12 and the area of the footpad projected on the landing surface.



THE SOIL FORCE NORMAL TO THE LANDING SURFACE OBTAINED BY THE SECONDARY SOIL MECHANICS METHOD IS GIVEN AS

$F_x = A\,d\,K_p$ OR $F_x = A\,P_{max}$

WHICHEVER IS LESS.

FIGURE 5-12 NORMAL SOIL FORCE SECONDARY SOIL MECHANICS METHOD

During the integration of the footpad's equations of motion, the normal soil force is compared to the force required to bring the footpad to rest during the next integration time step. This critical soil force is determined by

$$F_{cr} = m_i\left(g \ \text{Cos}\,\zeta - \frac{V_x}{\Delta t}\right) - S_x \tag{5-32}$$

where

F_{cr} = critical soil force.
m_i = mass of footpad.
g = local acceleration of gravity.
ζ = ground slope.
V_x = velocity of footpad into landing surface.
Δt = integration step size.
S_x = component of landing gear strut loads normal to landing surface.

The normal soil force applied to the footpad is never allowed to be greater than the magnitude of the force determine by Equation (5-32).

A force, acting in the plane of the landing surface, is obtained by multiplying the **normal** force by the selected value for the coefficient of friction. This force is applied in a direction opposite to the footpad's velocity in the plane of the landing surface. The components of this inplane soil force are expressed as

$$\begin{aligned} F_y &= - \frac{V_y}{\sqrt{V_y^{\,2} + V_z^{\,2}}}\ \mu F_x \\ F_z &= - \frac{V_z}{\sqrt{V_y^{\,2} + V_z^{\,2}}}\ \mu F_x \end{aligned} \tag{5-33}$$

where

V_y and V_z = inplane footpad velocities as shown in Figure 5-12.
F_y and F_z = components of inplane soil force.
μ = coefficient of friction.
F_x = normal soil force.

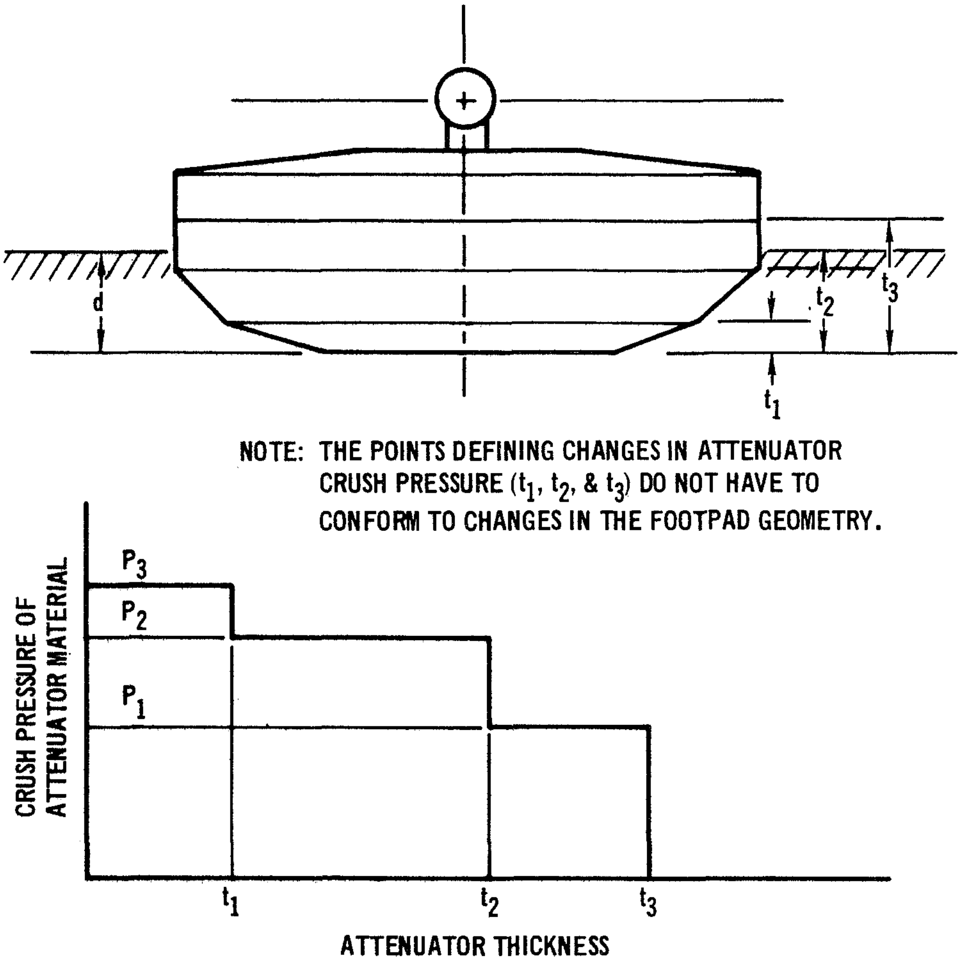
It should be noted that the force, F_x, employed in Equation (5-33), is the magnitude of the normal soil force obtained after the force resulting from the relationships of Figure 5-12 is compared with the critical soil force given by Equation (5-32).

5.1.5 Footpad Attenuation System - An additional attenuation system may be located on the bottom of the footpads to limit the landing loads of the footpads. Provisions have been made for including a crushable material on any or all of the conical segments used to represent the footpad shape. The amount of attenuation material crushing for each footpad is determined by a balance between the attenuator force and the soil force. The footpad attenuation system may be included with either the Primary or Secondary Soil Mechanics Methods. Forces associated with the footpad attenuation system are determined in the soil mechanics subroutine.

The footpad attenuation system may have three crush levels and the pressure-stroke relationship for the attenuator material is shown in Figure 5-13. It is assumed that the attenuator is crushed in a direction normal to the plane of the landing surface. The depth of soil penetration and amount of attenuator crushing, are determined by comparing the crush pressure with the soil pressure. With a soft soil and a stiff footpad attenuation material, most of the deformation will take place in the soil. For a very hard soil, a majority of the deformation will occur in the attenuator material. Intermediate values result in deformation of both the soil and attenuator materials.

At the end of a time interval, the attenuator thickness of each footpad which experiences crushing is adjusted to reflect this deformation. This is done by subtracting the crush distance from the coordinate locating the bottom of the footpad. Thus, if the lander rebounds, the crushed shape of the footpad attenuator is retained for the next impact. When the attenuator on a footpad is completely crushed the footpad attenuation portion of the analysis is bypassed, and the soil forces are applied directly to the footpad.

5.1.6 Footpad Sliding Motion - The two forces determined by the soil mechanics in the plane of the landing surface (F_y and F_z) are components of the inplane soil force F_f which is in a direction opposite to the sliding velocity of the footpad. This force tends to retard the sliding motion of the footpad along the landing surface.



NOTE: THE POINTS DEFINING CHANGES IN ATTENUATOR CRUSH PRESSURE (t_1, t_2, & t_3) DO NOT HAVE TO CONFORM TO CHANGES IN THE FOOTPAD GEOMETRY.

FIGURE 5-13 FOOTPAD ATTENUATION SYSTEM

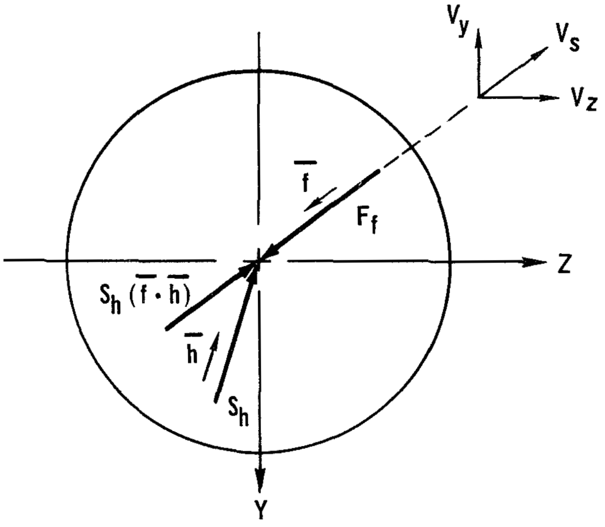
Referring to Figure 5-14, the equation of motion for the footpad in the plane of the landing surface and in the direction of the sliding motion is

$$m_i \frac{dV_s}{dt} = - F_f - S_h \, (\bar{f}\cdot\bar{h}) \qquad (5\text{-}34)$$

Therefore, the velocity of the footpad at the end of an integration step is

$$V_s \Big|_{t_n+\Delta t} = - \frac{F_f + S_h(\bar{f}\cdot\bar{h})}{m_i} \Delta t + V_s \Big|_{t_n} \qquad (5\text{-}35)$$

The critical value of F_f, which will just bring the footpad to rest is obtained from Equation (5-35) by setting the velocity at the end of the integration step ($V_s \big|_{t_n+\Delta t}$) equal to zero. Thus,



V_S = SLIDING VELOCITY OF FOOTPAD
S_h = COMPONENT OF LANDING GEAR STRUT LOAD PARALLEL TO LANDING SURFACE
$\bar{h}$ = UNIT VECTOR IN DIRECTION OF S_h
F_f = COMPONENT OF SOIL FORCE IN PLANE OF LANDING SURFACE
$\bar{f}$ = UNIT VECTOR IN DIRECTION OF F_f
m_i = MASS OF FOOTPAD
Δt = INTEGRATION STEP SIZE

WHEN THE FORCES F_f AND S_h OPPOSE EACH OTHER, THE FOLLOWING HOLDS

$$(\bar{f}\cdot\bar{h}) < 0$$

WHILE

$$(\bar{f}\cdot\bar{h}) > 0$$

WHEN THESE FORCES ACT IN THE SAME DIRECTION.

FIGURE 5–14 EVALUATION OF IN-PLANE SOIL FORCE MAGNITUDE

$$F_{fc} = \left.\frac{m_i V_s}{\Delta t}\right|_{t_n} - S_h\,(\bar{f}\cdot\bar{h}) \qquad (5\text{-}36)$$

Equation (5-36), in conjunction with Equation (5-37), is employed to maintain the magnitude of F_f at a value equal to or less than that which is required to bring the footpad to rest during the next integration step. When the relationship of Equation (5-37) holds, the magnitude of F_f determined by the soil mechanics routine is not changed.

$$F_f \leq F_{fc} \tag{5-37}$$

If the condition of Equation (5-37) is not met, the magnitude of F_f is recalculated by Equation (5-38) when F_f and $S_h(\bar{f}\cdot\bar{h})$ are in opposite directions and

$$F_f = \left.\frac{m_i V_s}{\Delta t}\right|_{t_n} - S_h\,(\bar{f}\cdot\bar{h}) \tag{5-38}$$

Equation (5-39) governs the magnitude of F_f when $S_h(\bar{f}\cdot\bar{h})$ and F_f are in the same direction.

$$F_f = \left.\frac{m_i V_s}{\Delta t}\right|_{t_n} \tag{5-39}$$

5.1.7 Lander Stability - To determine the stability of a legged lander configuration, the "plane of lander motion," as shown in Figure 5-15, is defined. This plane is defined by the gravity vector $\bar{g}$, and the translational velocity $\dot{\bar{R}}$, of the lander's center of gravity. The lander is considered unstable when the gravity vector passes outside of the area bounded by the lander footpads. If all the lander footpads are on the same side of the plane of lander motion, the lander is said to be experiencing yaw instability.

The more common case is when two footpads are astride the plane of lander motion, as shown in Figure 5-15. In this case, the vector $\bar{L}$, extending from the lander center of gravity to the intersection point of a line between these two footpads and the plane of lander motion, is obtained. The stability angle is then defined as

$$S = \text{Cos}^{-1}\left\{\frac{(\bar{L}\cdot\bar{g})}{|\bar{L}|\;|\bar{g}|}\right\} \tag{5-40}$$

As long as S is positive, the lander is considered to be stable, When S passes through zero the lander is said to be experiencing pitch instability.

To aid in the evaluation of the lander stability, the pitching velocity is also determined. This quantity is the component of the lander's total angular velocity in the plane of lander motion.

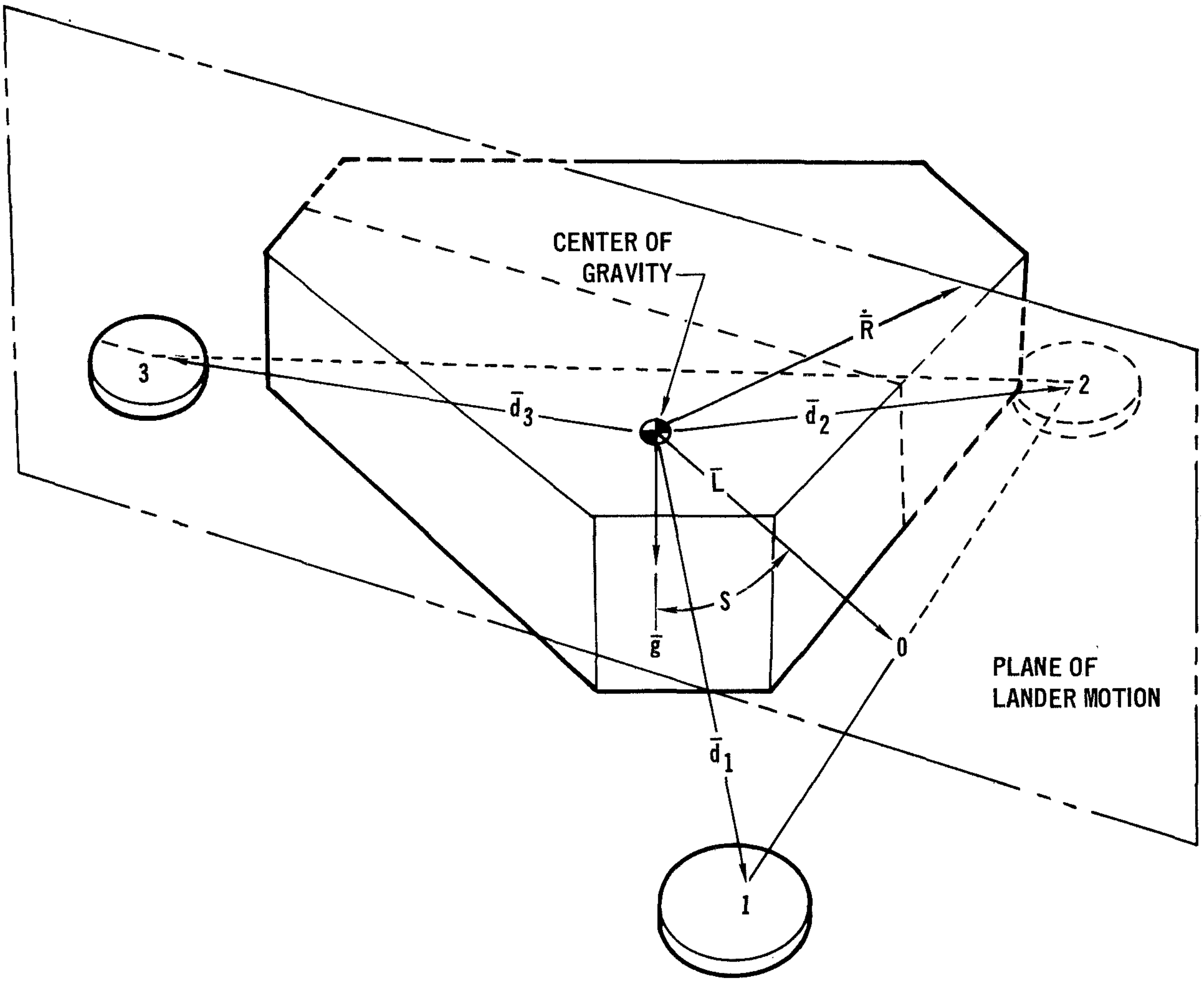


FIGURE 5-15 LANDER STABILITY ANGLE

This stability determination is made in the subroutine STABLE. It is an optional routine of the Landing Loads and Motions Program and is performed if the input indicator JCKSAB is set equal to 1. If the lander becomes unstable, the run is terminated with a printed message stating which type of instability was experienced.

5.2 Program Description - The Landing Loads and Motions Program is best described by defining functions of the program subroutines and examining program organization, as presented in a flow diagram. A listing of the program is given in Appendix I. All programming is in FORTRAN 2.0 for machine computation on CDC 6600 computers.

5.2.1 Subroutines - The Landing Loads and Motions Program is divided into three OVERLAY segments. Each segment consists of an executive subprogram and a number of subroutines as shown in Figure 5-16. This organization has been used to minimize the core storage requirements of the program. Several subroutines have multiple entry points, as indicated in Figure 5-16. The function of each subroutine, depending on the point of entry, is defined in Figure 5-16.

OVERLAY(LLMPT5,0,0) consists of the executive subprogram LLMP. LLMP calls the other two overlays in the proper order and contains all of the COMMON blocks.

READIT is the executive subprogram in OVERLAY(LLMPT5,1,0). This segment of the program reads and prints the input data and initializes all the routines before integration of the equations of motion.

LLMPEX is the executive subprogram in OVERLAY(LLMPT5,2,0). It controls the solution of the equations of motion. Subroutines in this portion of the program obtain the forces in the landing gear struts and determine the soil forces. These forces are summed on the center body and contacting footpads, resulting accelerations determined, equations of motion integrated, and time history quantities printed. At the completion of a time history, program control is returned to OVERLAY(LLMPT5,0,0) for the possible consideration of an additional data set.

Two numerical integration methods are incorporated in the subroutine RKCUT. These consist of a constant step Runge-Kutta method and a variable step Runge-Kutta method. A description of these integration techniques is presented in Appendix F.

SUBPROGRAM NAME	SUBROUTINE NAME	ENTRY POINT	OVERLAY LOCATION	SUBPROGRAM OR SUBROUTINE OPERATIONS
LLMP		LLMP	OVERLAY (LLMPT5,0,0)	MAIN EXECUTIVE SUBPROGRAM. CONTROLS THE EXECUTION OF THE TOTAL PROGRAM.
READIT		READIT	OVERLAY (LLMPT5,1,0)	EXECUTIVE SUBPROGRAM IN OVERLAY (LLMPT5,1,0). ALSO READS THE INPUT DATA AND DETERMINES IF THE AMOUNT OF DATA IS CONSISTENT WITH THE INPUT CONTROL INDICATORS. IF NOT, THE PROGRAM IS TERMINATED. INITIALIZES THE PROGRAM.
	DATAOT	DATAOT		PRINTS INPUT DATA.
LLMPEX		LLMPEX	OVERLAY (LLMPT5,2,0)	EXECUTIVE SUBPROGRAM IN OVERLAY (LLMPT5,2,0). CONTROLS INTEGRATION OF EQUATIONS OF MOTION.
	FTPAD	FTPAD		CONTROLS DETERMINATION OF LOADS IN LANDING GEAR STRUTS AND SOIL FORCES.
	GEOM	GEOM		DETERMINES DIRECTION COSINE MATRIX AND TIME DERIVATIVES OF EULER ANGLES.
	STRUT	STRUT		DETERMINES LOAD IN LANDING GEAR STRUTS.
	SOIL	SOIL		DETERMINES SOIL FORCES.
	INITUP			NUMERICAL INTEGRATION INITIALIZATION SUBROUTINE.
		LOC		DEFINES AND STORES CUTOFF VARIABLES.
		INUPD		SETS UP STORAGE FOR INTEGRATED VARIABLES.
	RKCUT			RUNGE-KUTTA NUMERICAL INTEGRATION SUBROUTINE.
		SETUP		INITIALIZATION OF RKCUT ROUTINE.
		INTEG		INTEGRATION OF EQUATIONS OF MOTION.
		UPDATE		UPDATES INTEGRATION VARIABLES AND MODIFIES INTEGRATION INTERVAL.
		CUT		CHECKS CUTOFF LIMITS.
	OUTPUT	OUTPUT		PRINTS TIME HISTORY INFORMATION.
		SUMMRY		PRINTS SUMMARY INFORMATION AT THE END OF A DATA CASE.
	STABLE	STABLE		CHECKS STABILITY OF LANDER.
	MATINV	MATINV		MATRIX INVERSION SUBROUTINE.
	GMPRD	GMPRD		MATRIX MULTIPLICATION SUBROUTINE.

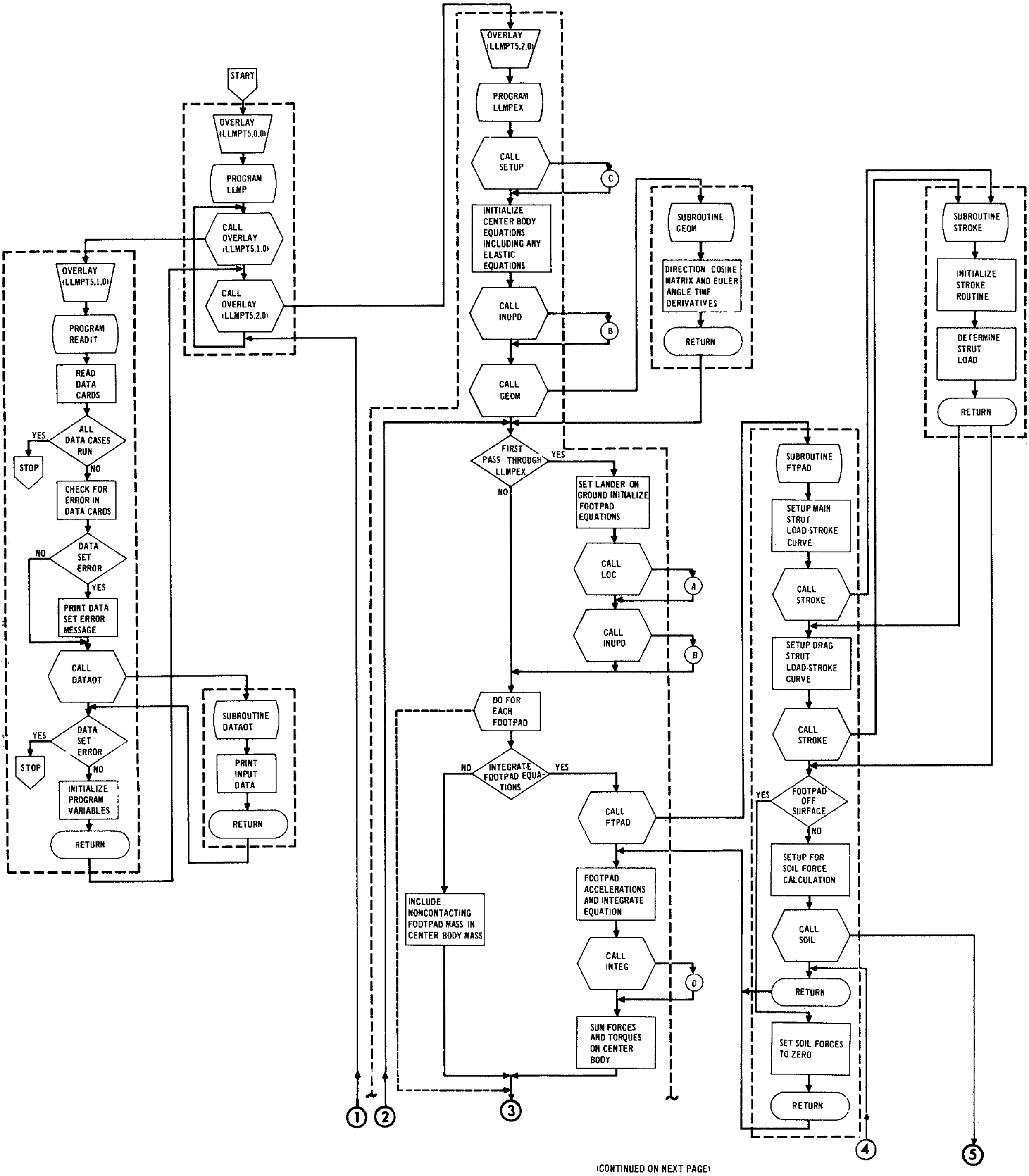
FIGURE 5-16 LANDING LOADS AND MOTIONS PROGRAM SUBROUTINES AND SUBPROGRAMS

5.2.2 Flow Diagram - A flow diagram showing the general operation of the Landing Loads and Motions Program is presented in Figure 5-17. The three OVERLAY segments are shown in addition to the various subprograms and subroutines which are located in each OVERLAY. This diagram is not intended to be a comprehensive programming chart, but shows the general flow of the program logic and indicates the order of operations within each subroutine. A complete listing of the Landing Loads and Motions Program is given in Appendix I.

5.3 Program Operation - Successful operation of the Landing Loads and Motions Program depends on proper input of data and correct interpretation of output data. These considerations are discussed in the following paragraphs. Examples of required input data and resulting output data for a typical landing condition are given in Appendix D.

5.3.1 Input Data - Required as input data is information describing the geometric and inertia properties of the specific lander to be studied; initial lander attitudes; linear and rotational velocities; surface conditions such as ground slope and soil properties; and the indicators needed to control the program's operation. This section defines the format of the input data cards and contains instructions for properly supplying input data to the program.

Figure 5-18 shows the required format for the input data cards. Columns 6 through 9 contain a card number, which must be right justified. Input data is placed in floating point form in columns 11-20, 21-30, 31-40, 41-50, and 51-60. Data input in a F format need not be right justified, but data in an E format must be right justified. In either case, the data must be contained entirely within the field of 10 columns provided. Columns 61 and on may be used for sequence numbers, identification statements, or comments. Following the last card of a data set, a card with NEXT in columns 1 through 4, must appear. Multiple cases may be run by stacking the data sets. Note that the input indicator IDSETN(Card 8) must be equal to 1 if another data set follows the current data set. The data cards for any additional data sets follow the first data set, and each of these are terminated with a NEXT in columns 1 through 4. A card with STOP in columns 1 through 4 signals the end of all the data sets.



(CONTINUED ON NEXT PAGE)

FIGURE 5-17 FLOW DIAGRAM LANDING LOADS AND MOTIONS PROGRAM

(CONTINUED FROM PREVIOUS PAGE)

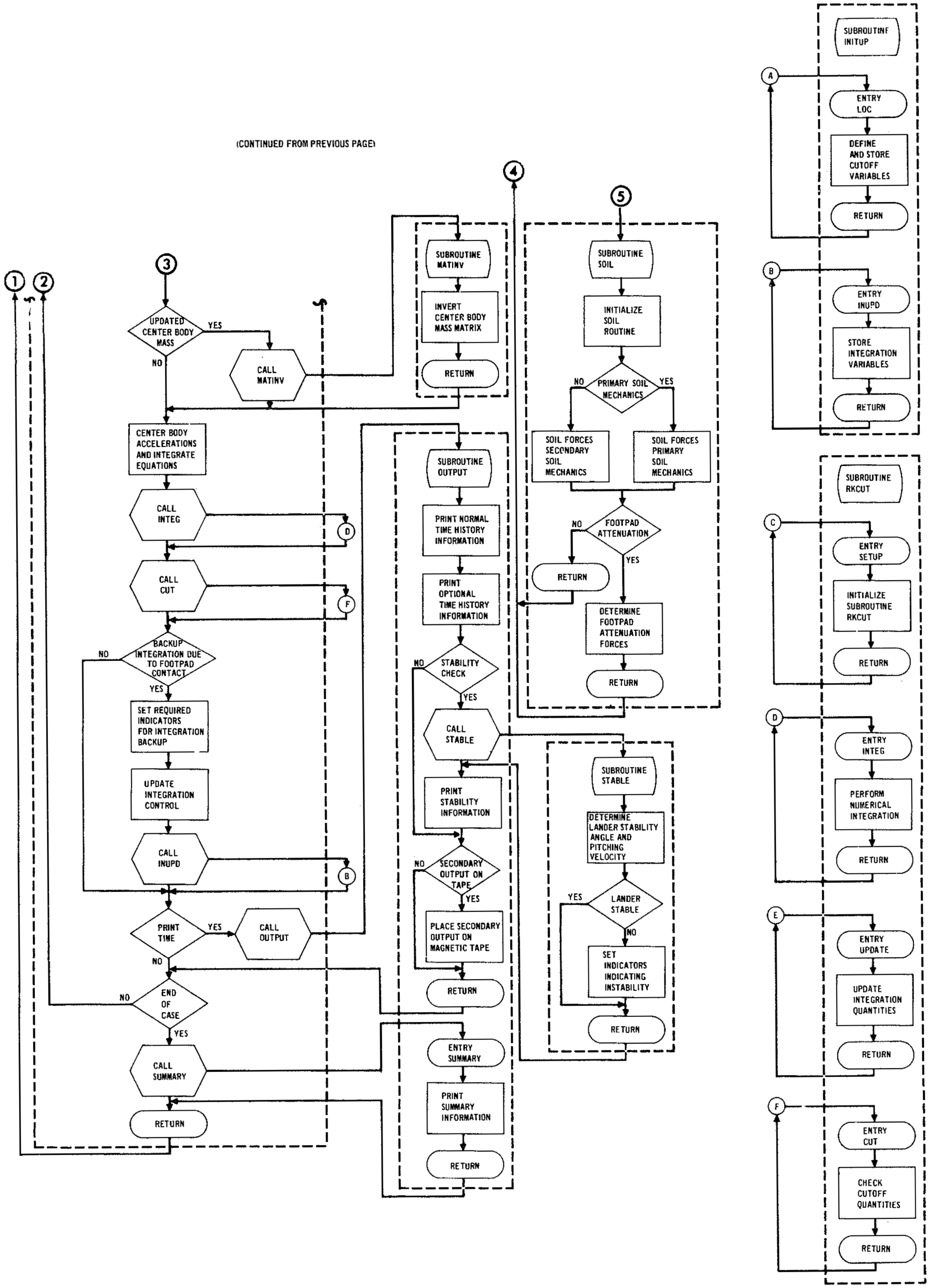


FIGURE 5-17 FLOW DIAGRAM LANDING LOADS AND MOTIONS PROGRAM (Continued)

There are a number of indicators and counters which check the input data as it is read to ensure that the correct amount of information has been input. If the number of data cards is incorrect, the run will be terminated, and error messages printed to indicate where the data error occurred. All of the indicators governing these input options are discussed in Figure 5-19.

Data cards for the first data set must contain all of the information required to completely initialize the first case. Only optional data, consistent with the input control indicators, may be left out of the first data set. For following data sets, only the information which is to be different (or in addition to that of the preceding case) needs to be changed on the appropriate cards.

There is no specific order in which the low number data cards, Cards 1 through 42, must appear in a data set. However, the order of the higher numbered cards (Cards 100 through 1700) is of importance. All of these cards with higher numbers define variables which are subscripted in the Landing Loads and Motions Program. Therefore, all the cards with the same card number must be input in the order in which the user requires this information stored in the program.

All input parameters and their associated data card numbers are defined in Figure 5-20. Most of these parameters are adequately explained in this figure, and Figure 5-19, but a number require additional comments.

There is no specific system of units associated with the input information, except for the angular quantities which must be expressed in degrees. All other parameters may be expressed in any consistent set of units, either English or Metric (inches or centimeters, pounds or dynes).

Care must be exercised to ensure that lander velocity, position, and attitude are correctly initialized. All lander initial conditions are referenced to the Gravity Coordinate System, Figure 5-1. The three components of velocity VELX, VELY, and VELZ are parallel to the axes of the Gravity Coordinate System. Note that a negative VELX is directed into the landing surface.

The desired ground slope, relating the Surface Coordinate System to the Gravity Coordinate System, must be established. This is accomplished by rotating the Surface Coordinate System about the Gravity Coordinate System Y_g axis an amount ζ equal to the ground slope. A positive slope corresponds to a rotation in the positive sense about Y_g.

The lander's initial angular orientation is obtained by three successive rotations about the three lander axes. The first is a rotation, ANGX, of the Lander Coordinate System about the Gravity Coordinate System X_g axis. This is followed by a rotation, ANGY, about the displaced position of the Lander Coordinate System Y axis. Final angular orientation is obtained by the rotation, ANGZ, about the direction of the Z axis resulting from the first two rotations. When these three rotations are zero, the initial attitude of the Lander Coordinate System is aligned with the Gravity Coordinate System.

The input quantity CUTERR (Card 7) defines a tolerance band, above the landing surface, which governs the placement of a footpad's equations of motion into the integration routine. When a footpad enters this band, the inertia effects of the footpad are removed from the center body equations of motion, initial conditions for the footpad are determined by center body motions, and integration of footpad equations of motion begun. This tolerance parameter has meaning only when INLEG (Card 2) is 0.

When including the effects of an elastic center body structure, the indicator MODEIN (Card 42) governs the number of center body modes which are input. NMODES (Card 3) governs the number of modes actually included in the analysis, and may be less than or equal to MODEIN.

A discussion of the input data required for a typical landing condition is presented in Sections 5.3.3 and 6.2.2 and the computer output for this case is given in Appendix D.

5.3.2 Output Data - At specified times during the integration of the equations of motion, various time varying quantities defining the position, velocities, and accelerations of the lander center body and footpads are printed. At the completion of a run, a summary page listing the maximum strokes in the landing gear struts, the reason for termination of the run, and

1	CASENO				
2	TIMAX	IPTCNT	INLEG	IFPPRT	
3	NMODES	NOOUT			
4	INDFXD	INDFYD	INDFZD		
5	INDFXR	INDFYR	INDFZR		
6	HMAX	HMIN	EMAX	EMIN	IP
7	IVARH	IMTH	CUTERR		
8	NFORC	IQUOUT	JCKSAB	IDSETN	
9	ZETA	GRAV	GRAVE		
10	ANGX	ANGY	ANGZ		
11	WX	WY	WZ		
12	VELX	VELY	VELZ		
13	CBMASS				
14	CBIXX	CBIYY	CBIZZ		
15	CBIXY	CBIXZ	CBIYZ		
16	FPMASS				
17	RAD(1)	RAD(2)	RAD(3)	RAD(4)	
18	SS(1)	SS(2)	SS(3)	SS(4)	
19	ATTHCK(1)	ATTHCK(2)	ATTHCK(3)		
20	ATTPRS(1)	ATTPRS(2)	ATTPRS(3)		
21	NTYPE				
22	SOILP(1)	SOILP(2)	SOILP(3)		
23	NOLEG	ILEG	DRAGST		
24	PFCMS(1)	PFCMS(2)	PFCMS(3)	PFCMS(4)	PFCMS(5)
25	PFTMS(1)	PFTMS(2)	PFTMS(3)	PFTMS(4)	PFTMS(5)
26	CDCMS(1)	CDCMS(2)	CDCMS(3)	CDCMS(4)	CDCMS(5)
27	CDTMS(1)	CDTMS(2)	CDTMS(3)	CDTMS(4)	CDTMS(5)

FIGURE 5–18 INPUT DATA FORMAT LANDING LOADS AND MOTIONS PROGRAM

```
       28 SRCMS(1)  SRCMS(2)  SRCMS(3)  SRCMS(4)  SRCMS(5)
       29 SRTMS(1)  SRTMS(2)  SRTMS(3)  SRTMS(4)  SRTMS(5)
       30 SCMXMS    STMXMS    SRUCMS    SRUTMS
       31 IRETMS
       32 FRICMS    COEFMS    GAMMS     AEI1      AEI2
       33 PFCDS(1)  PFCDS(2)  PFCDS(3)  PFCDS(4)  PFCDS(5)
       34 PFTDS(1)  PFTDS(2)  PFTDS(3)  PFTDS(4)  PFTDS(5)
       35 CDCDS(1)  CDCDS(2)  CDCDS(3)  CDCDS(4)  CDCDS(5)
       36 CDTDS(1)  CDTDS(2)  CDTDS(3)  CDTDS(4)  CDTDS(5)
       37 SRCDS(1)  SRCDS(2)  SRCDS(3)  SRCDS(4)  SRCDS(5)
       38 SRTDS(1)  SRTDS(2)  SRTDS(3)  SRTDS(4)  SRTDS(5)
       39 SCMXDS    STMXDS    SRUCDS    SRUTDS
       40 IRETDS
       41 FRICDS    COEFDS    GAMDS
       42 MODEIN
      100 XOUT(I)   YOUT(I)   ZOUT(I)
      200 XFP(I)    YFP(I)    ZFP(I)
      300 XMSCB(I)  YMSCB(I)  ZMSCB(I)
      400 XDSCB(I)  YDSCB(I)  ZDSCB(I)
      500 GM(I)     OMEGA(I)
      600 WNX(I)    WNY(I)    WNZ(I)
      700 PX(I)     PY(I)     PZ(I)
      800 PMSX(I,1) PMSX(I,2) PMSX(I,3) PMSX(I,4) PMSX(I,5)
      900 PMSY(I,1) PMSY(I,2) PMSY(I,3) PMSY(I,4) PMSY(I,5)
     1000 PMSZ(I,1) PMSZ(I,2) PMSZ(I,3) PMSZ(I,4) PMSZ(I,5)
     1100 PDSX(I,1) PDSX(I,2) PDSX(I,3) PDSX(I,4) PDSX(I,5)
     1200 PDSY(I,1) PDSY(I,2) PDSY(I,3) PDSY(I,4) PDSY(I,5)
```

FIGURE 5–18 INPUT DATA FORMAT LANDING LOADS AND MOTIONS PROGRAM (Continued)

```
     1300 PDSZ(I,1) PDSZ(I,2) PDSZ(I,3) PDSZ(I,4) PDSZ(I,5)
     1400 POUTX(I,1)POUTX(I,2)POUTX(I,3)POUTX(I,4)POUTX(I,5)
     1500 POUTY(I,1)POUTY(I,2)POUTY(I,3)POUTY(I,4)POUTY(I,5)
     1600 POUTZ(I,1)POUTZ(I,2)POUTZ(I,3)POUTZ(I,4)POUTZ(I,5)
     1700 PCGX(I)   PCGY(I)   PCGZ(I)
NEXT
STOP
```

FIGURE 5–18 INPUT DATA FORMAT LANDING LOADS AND MOTIONS PROGRAM (Continued)

CARD 2:		
	IPTCNT	INDICATOR WHICH DEFINES NUMBER OF INTEGRATION STEPS BETWEEN PRINT TIMES.
	INLEG = 0	FOOTPAD EQUATIONS OF MOTION INTEGRATED ONLY AFTER ACTUAL IMPACT WITH LANDING SURFACE.
	INLEG = 1	FOOTPAD EQUATIONS OF MOTION INTEGRATED AT ALL TIMES.
	IFPPRT > 0	NUMBER OF INTEGRATION TIME STEPS TO BE PRINTED FOLLOWING IMPACT OF A FOOTPAD.
	IFPPRT = 0	IPTCNT (CARD 2) CONTROLS PRINT INTERVAL AT ALL TIMES.
CARD 3:		
	NMODES = 0	RIGID CENTER BODY ASSUMED.
	NMODES > 0	NUMBER OF CENTER BODY MODES INCLUDED IN ANALYSIS (5 MAXIMUM).
	NOOUT = 0	NO SECONDARY ACCELERATION OUTPUT POINTS INCLUDED.
	NOOUT > 0	NUMBER OF SECONDARY ACCELERATION OUTPUT POINTS (10 MAXIMUM).
CARD 8:		
	NFORC = 0	NO SECONDARY TIME HISTORY OUTPUT OBTAINED.
	NFORC = 1	SECONDARY TIME HISTORY PLACED ON MAGNETIC TAPE CORRESPONDING TO TIMES OF PRINTED OUTPUT.
	IQUOUT = 0	NO SECONDARY INTEGRATION VARIABLE OUTPUT.
	IQUOUT = 1	SECONDARY INTEGRATION VARIABLE OUTPUT OBTAINED WITH NORMAL OUTPUT. THIS OPTION IS OVERRIDDEN WHEN NMODES = 0 SINCE THE NORMAL OUTPUT LISTS ALL THE INTEGRATED VARIABLES IN THIS CASE.
	JCKSAB = 0	NO LANDER STABILITY CHECK.
	JCKSAB = 1	LANDER STABILITY CHECK PERFORMED.
	IDSETN = 0	INDICATES ANOTHER DATA CASE DOES NOT FOLLOW THE CURRENT DATA.
	IDSETN = 1	INDICATES ANOTHER DATA CASE DOES FOLLOW THE CURRENT DATA.
CARD 21:		
	NTYPE = 0	PRIMARY SOIL MECHANICS ROUTINE.
	NTYPE = 1	SECONDARY SOIL MECHANICS ROUTINE.
CARD 23:		
	NOLEG	INDICATES NUMBER OF LEGS (5 MAXIMUM).
	ILEG = 0	INVERTED TRIPOD GEAR.
	ILEG = 1	CANTILEVER GEAR.
CARD 42:		
	MODEIN = 0	NO MODAL DATA INPUT.
	MODEIN > 0	NUMBER OF CENTER BODY MODES TO BE INPUT (5 MAXIMUM).
CARD 100:		NUMBER OF INPUT CARDS 100 MUST EQUAL NOOUT (CARD 3).
CARD 200:		NUMBER OF INPUT CARDS 200 MUST EQUAL NOLEG (CARD 23).
CARD 300:		NUMBER OF INPUT CARDS 300 MUST EQUAL NOLEG (CARD 23).
CARD 400:		NUMBER OF INPUT CARDS 400 MUST EQUAL TWO TIMES NOLEG (CARD 23).
CARD 500:		NUMBER OF INPUT CARDS 500 MUST EQUAL MODEIN (CARD 42).
CARD 600:		NUMBER OF INPUT CARDS 600 MUST EQUAL MODEIN (CARD 42).
CARD 700:		NUMBER OF INPUT CARDS 700 MUST EQUAL MODEIN (CARD 42).
CARD 800:		NUMBER OF INPUT CARDS 800 MUST EQUAL NOLEG (CARD 23).
CARD 900:		NUMBER OF INPUT CARDS 900 MUST EQUAL NOLEG (CARD 23).
CARD 1000:		NUMBER OF INPUT CARDS 1000 MUST EQUAL NOLEG (CARD 23).
CARD 1100:		NUMBER OF INPUT CARDS 1100 MUST EQUAL TWO TIMES NOLEG (CARD 23).
CARD 1200:		NUMBER OF INPUT CARDS 1200 MUST EQUAL TWO TIMES NOLEG (CARD 23).
CARD 1300:		NUMBER OF INPUT CARDS 1300 MUST EQUAL TWO TIMES NOLEG (CARD 23).
CARD 1400:		NUMBER OF INPUT CARDS 1400 MUST EQUAL NOOUT (CARD 3).
CARD 1500:		NUMBER OF INPUT CARDS 1500 MUST EQUAL NOOUT (CARD 3).
CARD 1600:		NUMBER OF INPUT CARDS 1600 MUST EQUAL NOOUT (CARD 3).
CARD 1700:		NUMBER OF INPUT CARDS 1700 MUST EQUAL MODEIN (CARD 42).

FIGURE 5–19 INPUT DATA CONTROL INDICATORS LANDING LOADS AND MOTIONS PROGRAM

CARD NO.	INPUT VARIABLE	COORDINATE SYSTEM*	VARIABLE DEFINITION
1	CASENO		CASE NUMBER.
2	TIMAX		MAXIMUM RUN TIME.
2	IPTCNT		TIME HISTORY PRINT CONTROL. NUMBER OF Δt's BETWEEN PRINTS.
2	INLEG		FOOTPAD INTEGRATION CONTROL INDICATOR. = 0 FOOTPAD EQUATIONS INTEGRATED ONLY AFTER FOOTPAD IMPACT = 1 FOOTPAD EQUATIONS INTEGRATED AT ALL TIMES
2	IFPPRT		PRINT CONTROL INDICATOR FOR FOOTPAD IMPACT = 0 CONTROL OF PRINT INTERVAL ALWAYS GOVERNED BY IPTCNT > 0 NUMBER INTEGRATION TIME STEPS TO BE PRINTED FOLLOWING FOOTPAD IMPACT
3	NMODES		NUMBER OF ELASTIC MODES INCLUDED IN ANALYSIS.
3	NOOUT		NUMBER OF SECONDARY ACCELERATION OUTPUT POINTS.
4	INDFXD		INDICATOR TO SUPPRESS CENTER BODY X_S DEGREE OF FREEDOM. = 0 ALLOW DEGREE OF FREEDOM $\neq$ 0 SUPPRESS DEGREE OF FREEDOM
4	INDFYD		INDICATOR TO SUPPRESS CENTER BODY Y_S DEGREE OF FREEDOM (SEE INDFXD).
4	INDFZD		INDICATOR TO SUPPRESS CENTER BODY Z_S DEGREE OF FREEDOM (SEE INDFXD).
5	INDFXR		INDICATOR TO SUPPRESS CENTER BODY ROTATION ABOUT X AXIS (SEE INDFXD).
5	INDFYR		INDICATOR TO SUPPRESS CENTER BODY ROTATION ABOUT Y AXIS (SEE INDFXD).
5	INDFZR		INDICATOR TO SUPPRESS CENTER BODY ROTATION ABOUT Z AXIS (SEE INDFXD).
6	HMAX		MAXIMUM INTEGRATION TIME INTERVAL.
6	HMIN		MINIMUM INTEGRATION TIME INTERVAL (VARIABLE STEP RUNGE-KUTTA).
6	EMAX		MAXIMUM INTEGRATION ACCURACY (VARIABLE STEP RUNGE-KUTTA).
6	EMIN		MINIMUM INTEGRATION ACCURACY (VARIABLE STEP RUNGE-KUTTA).
6	IP		QUANTITY USED TO SET INITIAL INTEGRATION STEP SIZE. $\Delta t = \Delta t_{max}/(2^{IP})$.
7	IVARH		INTEGRATION INDICATOR IVARH = 0 VARIABLE STEP INTEGRATION. IVARH = 1 CONSTANT STEP INTEGRATION.
7	IMTH		DUMMY VARIABLE. SET EQUAL TO 0.
7	CUTERR		TOLERANCE ON CONTROL OF FOOTPAD IMPACT
8	NFORC		SECONDARY TIME HISTORY OUTPUT INDICATOR. NFORC = 1 SECONDARY OUTPUT ON TAPE 3. NFORC = 0 NO SECONDARY OUTPUT.
8	IQUOUT		INTEGRATED VARIABLE OUTPUT INDICATOR. IQUOUT = 1 PRINT ALL INTEGRATED VARIABLES. IQUOUT = 0 NO INTEGRATED VARIABLE OUTPUT.
8	JCKSAB		STABILITY CHECK INDICATOR. JCKSAB = 0 – DO NOT CHECK STABILITY. JCKSAB = 1 – CHECK LANDER STABILITY.
8	IDSETN		MULTIPLE DATA CASE INDICATOR. IDSETN = 1 – ANOTHER DATA SET FOLLOWING CURRENT DATA SET. IDSETN = 0 – NO ADDITIONAL DATA SETS.

* NOTE: THE FOLLOWING ABBREVIATIONS ARE USED TO DEFINE THE COORDINATE SYSTEMS.
SCS – SURFACE COORDINATE SYSTEM.
LCS – LANDER COORDINATE SYSTEM.
GCS – GRAVITY COORDINATE SYSTEM

FIGURE 5–20 INPUT DATA LANDING LOADS AND MOTIONS PROGRAM

CARD NO.	INPUT VARIABLE	COORDINATE SYSTEM*	VARIABLE DEFINITION
9	ZETA	GCS	GROUND SLOPE.
9	GRAV		ACCELERATION OF GRAVITY ON PLANET.
9	GRAVE		ACCELERATION OF GRAVITY ON EARTH.
10	ANGX	LCS	INITIAL ANGULAR ROTATION OF LCS ABOUT X_g.
10	ANGY	LCS	ANGULAR ROTATION OF LCS ABOUT DIRECTION OF Y FOLLOWING ANGX.
10	ANGZ	LCS	ANGULAR ROTATION OF LCS ABOUT DIRECTION OF Z FOLLOWING ANGX AND ANGY.
11	WX	LCS	INITIAL LANDER ANGULAR VELOCITY ABOUT X AXIS.
11	WY	LCS	INITIAL LANDER ANGULAR VELOCITY ABOUT Y AXIS.
11	WZ	LCS	INITIAL LANDER ANGULAR VELOCITY ABOUT Z AXIS.
12	VELX	GCS	INITIAL LANDER VELOCITY ALONG X_g AXIS.
12	VELY	GCS	INITIAL LANDER VELOCITY ALONG Y_g AXIS.
12	VELZ	GCS	INITIAL LANDER VELOCITY ALONG Z_g AXIS.
13	CBMASS		CENTER BODY MASS.
14	CBIXX		CENTER BODY MASS MOMENT OF INERTIA – I_{xx}.
14	CBIYY		CENTER BODY MASS MOMENT OF INERTIA – I_{yy}.
14	CBIZZ		CENTER BODY MASS MOMENT OF INERTIA – I_{zz}.
15	CBIXY		CENTER BODY PRODUCT OF INERTIA – I_{xy}.
15	CBIXZ		CENTER BODY PRODUCT OF INERTIA – I_{xz}.
15	CBIYZ		CENTER BODY PRODUCT OF INERTIA – I_{yz}.
16	FPMASS		FOOTPAD MASS.
17	RAD(I)		RADIUS OF ITH FOOTPAD SEGMENT (SEE FIGURE 5–8).
18	SS(I)		DISTANCE FROM ITH FOOTPAD SEGMENT TO FOOTPAD PIVOT POINT (SEE FIGURE 5–8).
19	ATTHCK(I)		THICKNESS OF ITH SEGMENT OF FOOTPAD ATTENUATION MATERIAL RELATIVE TO BOTTOM OF FOOTPAD (SEE FIGURE 5–13).
20	ATTPRS(I)		CRUSH PRESSURE OF ITH SEGMENT OF FOOTPAD ATTENUATION MATERIAL (SEE FIGURE 5–13).
21	NTYPE		SOIL MECHANICS INDICATOR. NTYPE = 0 – PRIMARY SOIL MECHANICS. NTYPE = 1 – SECONDARY SOIL MECHANICS.
22	SOILP(1)		FOR NTYPE = 0 – SOIL INTERNAL FRICTION ANGLE. FOR NTYPE = 1 – COEFFICIENT OF FRICTION.
22	SOILP(2)		FOR NTYPE = 0 – SOIL UNIT WEIGHT. FOR NTYPE = 1 – SLOPE OF ELASTIC SOIL PENETRATION RELATIONSHIP.
22	SOILP(3)		FOR NTYPE = 0 – SOIL RELATIVE DENSITY. FOR NTYPE = 1 – MAXIMUM SOIL PRESSURE.
23	NOLEG		NUMBER OF LEGS.
23	ILEG		TYPE GEAR INDICATOR. ILEG = 0 – INVERTED TRIPOD. ILEG = 1 – CANTILEVER.
23	DRAGST		DISTANCE FROM CENTER BODY ATTACH POINT TO DRAG STRUT ATTACH POINT ALONG AXIS OF MAIN STRUT FOR CANTILEVER GEAR.
24	PFCMS(I)		MAIN STRUT ITH COMPRESSION PLASTIC LOAD LEVEL (SEE FIGURE 4–18)
25	PFTMS(I)		MAIN STRUT ITH TENSION PLASTIC LOAD LEVEL (SEE FIGURE 4–18).

*NOTE: THE FOLLOWING ABBREVIATIONS ARE USED TO DEFINE THE COORDINATE SYSTEMS.

SCS – SURFACE COORDINATE SYSTEM.
LCS – LANDER COORDINATE SYSTEM.
GCS – GRAVITY COORDINATE SYSTEM.

FIGURE 5–20 INPUT DATA LANDING LOADS AND MOTIONS PROGRAM (Continued)

CARD NO.	INPUT VARIABLE	COORDINATE SYSTEM*	VARIABLE DEFINITION
26	CDCMS(I)		MAIN STRUT ITH COMPRESSION PLASTIC STROKE LIMIT (SEE FIGURE 4–18).
27	CDTMS(I)		MAIN STRUT ITH TENSION PLASTIC STROKE LIMIT (SEE FIGURE 4–18).
28	SRCMS(I)		MAIN STRUT ITH COMPRESSION SPRING RATE (SEE FIGURE 4–18).
29	SRTMS(I)		MAIN STRUT ITH TENSION SPRING RATE (SEE FIGURE 4–18).
30	SCMXMS		MAXIMUM ALLOWABLE MAIN STRUT COMPRESSION STROKE.
30	STMXMS		MAXIMUM ALLOWABLE MAIN STRUT TENSION STROKE.
30	SRUCMS		MAIN STRUT COMPRESSION UNLOADING SPRING RATE.
30	SRUTMS		MAIN STRUT TENSION UNLOADING SPRING RATE.
31	IRETMS		MAIN STRUT UNLOADING INDICATOR. IRETMS = 0 – UNLOAD ALONG SRUCMS OR SRUTMS. IRETMS = 1 – UNLOADING SPRING RATE OBTAINED FROM LOAD-STROKE CURVE.
32	FRICMS		MAIN STRUT SLIDING FRICTION FORCE.
32	COEFMS		COEFFICIENT OF MAIN STRUT VELOCITY DEPENDENT FRICTION.
32	GAMMS		POWER OF MAIN STRUT VELOCITY DEPENDENT FRICTION.
32	AEI1		BENDING STIFFNESS (EI) OF UPPER SECTION OF CANTILEVER GEAR MAIN STRUT. (SEE FIGURE 5–5)
32	AEI2		BENDING STIFFNESS (EI) OF LOWER SECTION OF CANTILEVER GEAR MAIN STRUT (SEE FIGURE 5–5).
33	PFCDS(I)		DRAG STRUT ITH COMPRESSION PLASTIC LOAD LEVEL (SEE FIG. 4–18)
34	PFTDS(I)		DRAG STRUT ITH TENSION PLASTIC LOAD LEVEL (SEE FIG. 4–18)
35	CDCDS(I)		DRAG STRUT ITH COMPRESSION PLASTIC STROKE LIMIT (SEE FIGURE 4–18).
36	CDTDS(I)		DRAG STRUT ITH TENSION PLASTIC STROKE LIMIT (SEE FIGURE 4–18).
37	SRCDS(I)		DRAG STRUT ITH COMPRESSION SPRING RATE (SEE FIGURE 4–18).
38	SRTDS(I)		DRAG STRUT ITH TENSION SPRING RATE (SEE FIGURE 4–18)
39	SCMXDS		MAXIMUM ALLOWABLE DRAG STRUT COMPRESSION STROKE.
39	STMXDS		MAXIMUM ALLOWABLE DRAG STRUT TENSION STROKE.
39	SRUCDS		DRAG STRUT COMPRESSION UNLOADING SPRING RATE.
39	SRUTDS		DRAG STRUT TENSION UNLOADING SPRING RATE.
40	IRETDS		DRAG STRUT UNLOADING INDICATOR. IRETDS = 0 – UNLOAD ALONG SRUCDS OR SRUTDS. IRETDS = 1 – UNLOADING SPRING RATE OBTAINED FROM LOAD-STROKE CURVE.
41	FRICDS		DRAG STRUT SLIDING FRICTION FORCE.
41	COEFDS		COEFFICIENT OF DRAG STRUT VELOCITY DEPENDENT FRICTION.
41	GAMDS		POWER OF DRAG STRUT VELOCITY DEPENDENT FRICTION.
42	MODEIN		NUMBER OF ELASTIC MODES INPUT.
100	XOUT(I)	LCS	X COORDINATE OF I TH ACCELERATION OUTPUT POINT.
100	YOUT(I)	LCS	Y COORDINATE OF I TH ACCELERATION OUTPUT POINT.
100	ZOUT(I)	LCS	Z COORDINATE OF I TH ACCELERATION OUTPUT POINT.

*NOTE: THE FOLLOWING ABBREVIATIONS ARE USED TO DEFINE THE COORDINATE SYSTEMS.
SCS – SURFACE COORDINATE SYSTEM.
LCS – LANDER COORDINATE SYSTEM.
GCS – GRAVITY COORDINATE SYSTEM

FIGURE 5–20 INPUT DATA LANDING LOADS AND MOTIONS PROGRAM (Continued)

CARD NO.	INPUT VARIABLE	COORDINATE SYSTEM*	VARIABLE DEFINITION
200	XFP(I)	LCS	X COORDINATE OF ITH FOOTPAD.
200	YFP(I)	LCS	Y COORDINATE OF ITH FOOTPAD.
200	ZFP(I)	LCS	Z COORDINATE OF ITH FOOTPAD.
300	XMSCB(I)	LCS	X COORDINATE OF CENTER BODY END OF ITH MAIN STRUT.
300	YMSCB(I)	LCS	Y COORDINATE OF CENTER BODY END OF ITH MAIN STRUT.
300	ZMSCB(I)	LCS	Z COORDINATE OF CENTER BODY END OF ITH MAIN STRUT.
400	XDSCB(I)	LCS	X COORDINATE OF CENTER BODY END OF ITH DRAG STRUT.
400	YDSCB(I)	LCS	Y COORDINATE OF CENTER BODY END OF ITH DRAG STRUT.
400	ZDSCB(I)	LCS	Z COORDINATE OF CENTER BODY END OF ITH DRAG STRUT.
500	GM(I)		GENERALIZED MASS OF ITH ELASTIC MODE. (EQUATION (5–13)
500	OMEGA(I)		FREQUENCY OF ITH ELASTIC MODE (Hz).
600	WNX(I)		GENERALIZED INERTIA PROPERTY N_{xn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
600	WNY(I)		GENERALIZED INERTIA PROPERTY N_{yn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
600	WNZ(I)		GENERALIZED INERTIA PROPERTY N_{zn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
700	PX(I)		GENERALIZED INERTIA PROPERTY P_{xn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
700	PY(I)		GENERALIZED INERTIA PROPERTY P_{yn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
700	PZ(I)		GENERALIZED INERTIA PROPERTY P_{zn} FOR ITH ELASTIC MODE (EQUATION (5–13)).
800	PMSX(I,J)	LCS	X MODE SHAPE FOR ITH MAIN STRUT POINT IN JTH MODE.
900	PMSY(I,J)	LCS	Y MODE SHAPE FOR ITH MAIN STRUT POINT IN JTH MODE.
1000	PMSZ(I,J)	LCS	Z MODE SHAPE FOR ITH MAIN STRUT POINT IN JTH MODE.
1100	PDSX(I,J)	LCS	X MODE SHAPE FOR ITH DRAG STRUT POINT IN JTH MODE.
1200	PDSY(I,J)	LCS	Y MODE SHAPE FOR ITH DRAG STRUT POINT IN JTH MODE.
1300	PDSZ(I,J)	LCS	Z MODE SHAPE FOR ITH DRAG STRUT POINT IN JTH MODE.
1400	POUTX(I,J)	LCS	X MODE SHAPE FOR ITH ACCELERATION POINT IN JTH MODE.
1500	POUTY(I,J)	LCS	Y MODE SHAPE FOR ITH ACCELERATION POINT IN JTH MODE.
1600	POUTZ(I,J)	LCS	Z MODE SHAPE FOR ITH ACCELERATION POINT IN JTH MODE.
1700	PCGX(I)	LCS	X MODE SHAPE AT CENTER OF GRAVITY IN ITH MODE.
1700	PCGY(I)	LCS	Y MODE SHAPE AT CENTER OF GRAVITY IN ITH MODE.
1700	PCGZ(I)	LCS	Z MODE SHAPE AT CENTER OF GRAVITY IN ITH MODE.

*NOTE: THE FOLLOWING ABBREVIATIONS ARE USED TO DEFINE THE COORDINATE SYSTEMS.
SCS – SURFACE COORDINATE SYSTEM.
LCS – LANDER COORDINATE SYSTEM.
GCS – GRAVITY COORDINATE SYSTEM.

FIGURE 5–20 INPUT DATA LANDING LOADS AND MOTIONS PROGRAM (Continued) case run time are presented. In addition, optional output data, as requested through input indicators, may be obtained. These consist of lander stability angle and pitching velocity, all of the individual integrated quantities when considering a flexible center body, and accelerations at points other than the center body center of gravity. In this latter case, the coordinates of the points at which the accelerations are desired are included as input data. The individual integrated quantities are of interest when considering a flexible structure because the normal output presents the combination of the rigid body and elastic motions. Examples of the output data obtained from the Landing Loads and Motions Program are presented in Appendix D.

The units of the output data are consistent with the system of units used in the input data. Only the angular quantities are output in the specific units of degrees.

An additional output option is available in the Landing Loads and Motions Program. This option allows the output of the landing gear strut forces and all of the time history variables required to determine the inertia loading throughout the center body structure. With this information, the internal loads of the center body structural members may be determined with the Center Body Option of the Structural Analysis Program. This data is output at points in time corresponding to the times when printed output is generated by the program. This option results when the input indicator NFORC (Card 8) is equal to 1 and the data is placed on magnetic tape for future use. The Center Body Landing Loads Program, which is used to retrieve this data from the tape and generate input information for the Structural Analysis Program, is discussed in Appendix E.

5.3.3 Example of Program Operation - A listing of the input data, and the resulting printed output obtained from the Landing Loads and Motions Program is presented in Appendix D. The output pages in this appendix present the input data, the beginning of the time history, and the last pages of the time history including the summary information. This data set corresponds to a computer run demonstrating correlation with the third drop test conducted at NASA Langley Research Center on the Task Order Three lander.

For this example case, the fixed step, Runge-Kutta integration routine was used with output printed every 10 integration steps. An integration time step of 0.0001 sec with a total real time of 0.1 sec was requested. In addition to the normal printed output, the center body time histories and landing gear loads were output on magnetic tape for later use with the Structural Analysis Program. The retrieval of this information from the tape is discussed in Appendix E. Flexible center body input data was determined with the Structural Analysis Program, as discussed in Sections 4.3.1.3 and 6.2.1.

The lander had three inverted tripod gears with crushable main struts and elastic drag struts. The Secondary Soil Mechanics routine was used in conjunction with a footpad attenuation system. Since the landing case considered was planar in nature, the unnecessary rigid body degrees of freedom were suppressed to conserve computer run time. The equations of motion for all the footpads were integrated during the complete run.

This run, including a flexible center body structure represented with three free-free modes, and the real time of 0.1 sec required 53 sec of CDC 6600 CP time. Computer runs for this length of real time required approximately 34 CP sec to run a case with a rigid center body.

Plots showing various time histories obtained from this run are given in Section 6.2.2. Also shown is a comparison of the results of this run with those obtained with a rigid center body.

6. DEMONSTRATION OF PROGRAM CAPABILITIES

Examples of the capabilities of the various developed computer programs are presented in the following sections. The Task Order Three lander configuration, shown in Figure 1-1, was chosen for many of these analyses since experimental data was available with which to compare the computer results. In general, the correlation between experimental data and the analytical studies was quite good. In addition to the planar landings of the Task Order Three Lander, results for spatial landings on different types of soil are discussed.

6.1 Large Displacement Analysis of Landing Gears - The Landing Gear Option of the Structural Analysis Program was employed to analyze typical inverted tripod and cantilever landing gear configurations. An analysis of the Task Order Three inverted tripod gear to determine drag strut loads versus stroke normal to a landing surface is presented. Predicted drag strut loads are correlated with loads obtained in the sixth drop test on the Task Order Three landing gear performed at NASA Langley Research Center. In addition, drag strut loads as a function of stroke are predicted for an arbitrarily oriented friction plane to indicate the influence on these loads of a non-symmetrically loaded gear.

Analysis of the one-sixth scale LM cantilever gear to determine maximum energy absorption capability for various orientations of the landing surface is discussed in Section 6.1.2. Friction coefficient of the surface was varied to show the way in which friction affects energy absorption.

The fixed finite element idealization of inverted tripod and cantilever landing gears, employed in the Structural Analysis Program Landing Gear Option, was shown in Figure 4-17.

6.1.1 Inverted Tripod Gear - The inverted tripod gear of the Task Order Three Lander was analyzed to determine drag strut loads as a function of gear stroke. A listing of input information and output data for this problem is presented in Appendix C. Only selected pages of computer output are included due to the volume of output. Header card (see Figure 4-36) for this data case does not appear in the output listing although it was the first card of the data set.

Main strut honeycomb properties used for the gear are given in Figure 6-1. The Surface Coordinate System and Lander Coordinate System were assumed to be coincident, with the X_L and X_S axes normal to (and pointing out of) the landing surface (see Figure 4-20). Hence, the Euler angles ψ, θ, and ϕ are equal to 0. This condition corresponds to a "straight-in" drop of a lander such that the drag struts of a gear are loaded symmetrically. Behavior of the gear was investigated in 60 steps for an applied displacement of 15.24 cm normal to the landing surface. Friction coefficient of the surface was assumed to be .8. Footpad honeycomb crush forces of 1.102×10^9 dynes and 4.18×10^9 dynes and corresponding crush strokes of 1.27 cm and 4.45 cm were assumed.

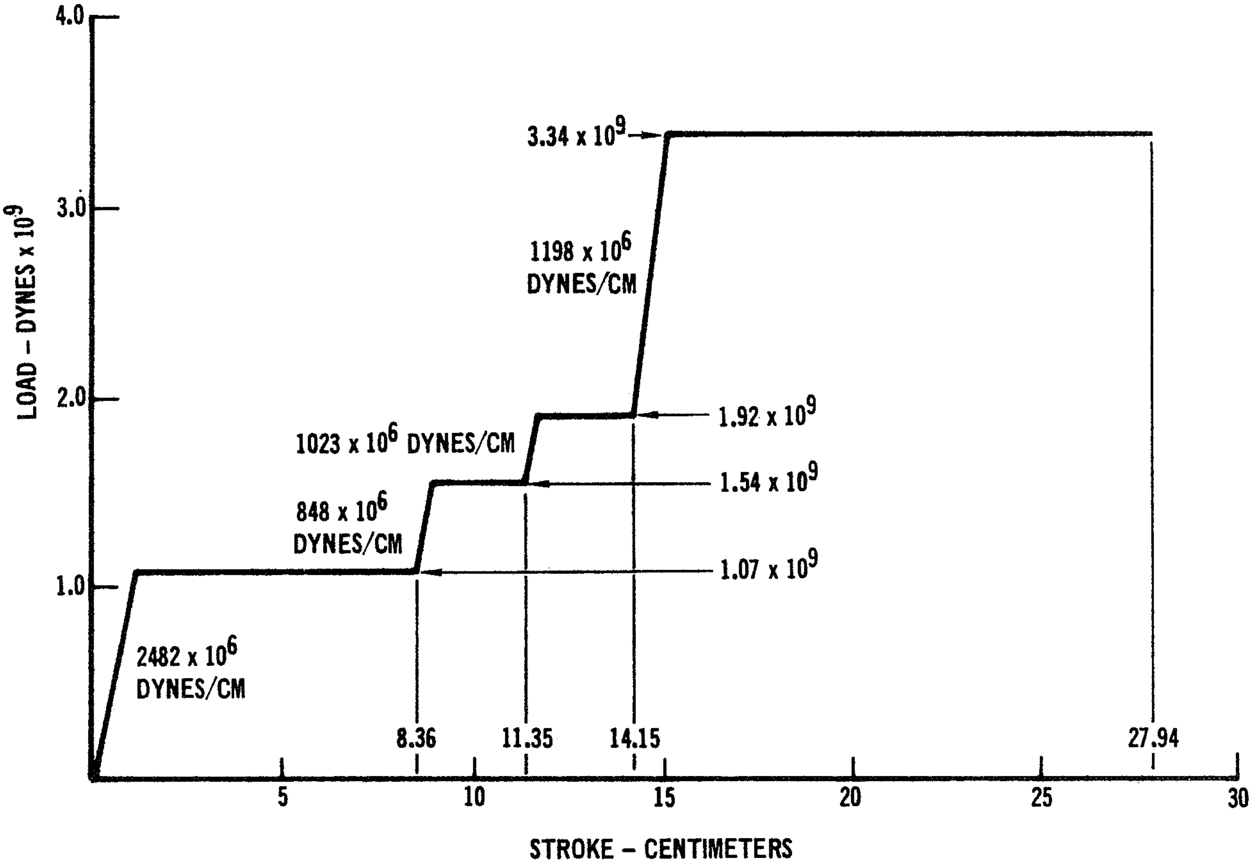


FIGURE 6-1 LOAD - STROKE CURVE FOR MAIN STRUT OF INVERTED TRIPOD GEAR - TASK ORDER THREE LANDER

A correlation between drag strut loads obtained in the computer run (steps 1, 2, 9, 10, 59 and 60 shown in Appendix C) and loads obtained for this gear during the sixth leg drop test performed at NASA Langley Research Center is depicted in Figure 6-2. As illustrated, at the onset of stroking the experimental drag strut loads appear to build up instantaneously. This is due to large footpad inertia forces acting during initial footpad impact; thus, correlation of drag strut loads at the onset of stroking is poor since the program assumes that the struts are initially unloaded. Footpad inertia forces are negligible after a normal (X_S) stroke of about 1 cm and from this point on correlation is reasonably good. In the figure, only one curve is shown for predicted drag strut loads since the drag struts were loaded equally because of symmetry. This example indicates that the static analysis employed in the Landing Gear Option can be used to obtain an estimate of dynamic drag strut loads.

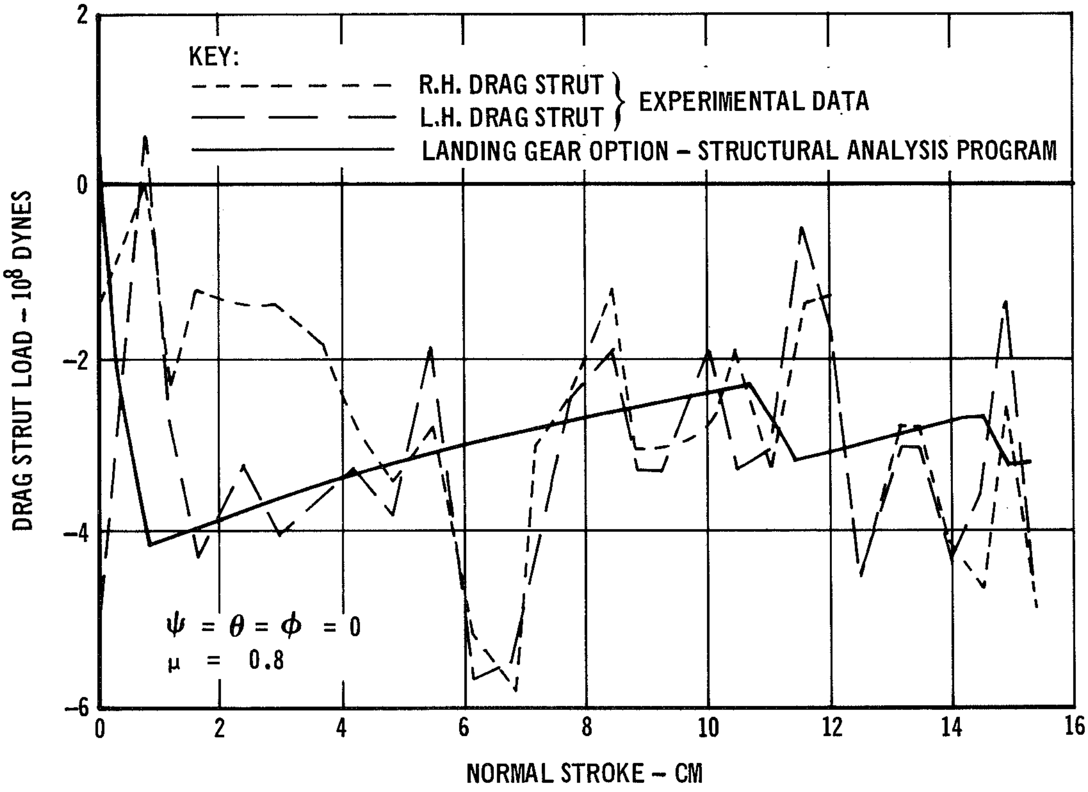


FIGURE 6-2 DRAG STRUT LOAD CORRELATION – INVERTED TRIPOD GEAR

For a nonsymmetrically loaded gear, the drag strut loads may be considerably higher than for the symmetrically loaded case. To emphasize this fact, the inverted tripod gear described above was analyzed with the program using the same input data with the exception that the landing surface was oriented such that $\psi = 25°$ and $\theta = \phi = 0$. This condition can be visualized by referring to Figure 4-20 and imagining that the landing surface is rotated 25° about the Z_S direction. For this run, drag strut loads as a function of stroke normal to the rotated landing surface are shown in Figure 6-3. As shown, one drag strut is in tension while the other is in compression. The maximum drag strut load of 4.52×10^9 dynes is considerably higher than the drag strut loads shown in Figure 6-2 for the symmetrically loaded gear.

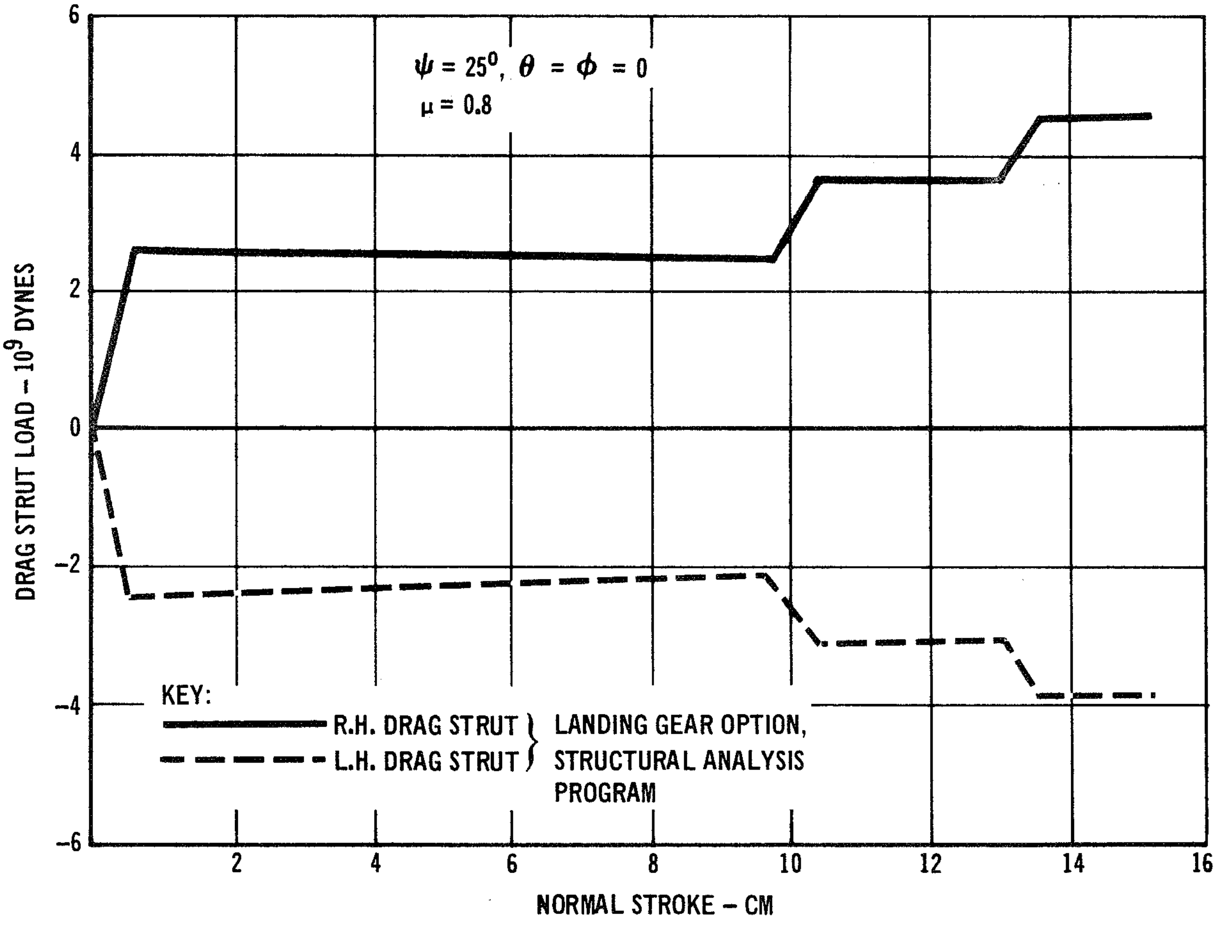


FIGURE 6-3 ANTICIPATED DRAG STRUT LOADS FOR INVERTED TRIPOD GEAR

6.1.2 Cantilever Gear - The one-sixth scale LM cantilever gear was analyzed to determine maximum energy absorption capability as a function of landing surface orientation and friction coefficient.

Honeycomb crush properties assumed for the main strut and drag struts of the gear are shown in Figures 6-4 and 6-5. Various orientations of the landing surface (see Figure 4-20) were considered, each of which provided symmetrical loading of the drag struts. These orientations are described by the Euler angle combinations $\psi = 0°$, $\phi = 0°$, and θ varied from 0° to 40°. The maximum value of 40° is the sum of the maximum ground slope relative to the local horizontal (30°) and the maximum pitch-up of the lander (10°) specified in typical design constraints, Section 3.1. Behavior of the gear was investigated by applying a displacement of 30.48 cm, normal to the landing surface, in 100 steps. This large value of normal displacement was selected to insure that the gear bottomed out before all steps were taken. For each coefficient of friction considered, a series of computer runs, were made in which θ was varied from 0° to 40°. Friction coefficients of 0.2, 0.8, and 1.0 were considered.

Maximum energy absorbed by the gear in a direction normal to the landing surface (X_S direction) is presented in Figure 6-6. Each point in the figure represents the total energy absorbed by the gear before the main strut bottomed out. Curves for friction coefficients of 0.8 and 1.0 are the same because in each case the friction was sufficient to prevent the footpad from sliding. For the friction coefficient of 0.2, the curve includes the effect of footpad sliding on the landing surface. As evidenced in this figure, energy absorption capability of the gear is minimum for values of θ between 20° and 30°.

6.2 Analysis of Task Order Three Lander - A complete analysis of a typical legged lander configuration is discussed in the following sections. This study consisted of performing a modal analysis on the center body structure employing the Center Body Option of the Structural Analysis Program. This modal data was then incorporated in the Landing Loads and Motions Program to account for the effects of a flexible center body on the landing response of the vehicle. Output from the landing program was used, in conjunction with the Center Body Landing Loads Program, to determine the distribution of inertia, gravity, and landing gear strut loads throughout the center body. These loads were then input to the Center Body Option of the Structural Analysis Program to determine internal member load distributions.

The legged lander considered was the 3/8 mass version of the Task Order Three lander. Figure 6-7 shows the structural configuration of this vehicle. Included in following analysis of this lander are examples of correlation between predicted results obtained with the Landing Loads and Motions Program and the results of the third drop test conducted at NASA Langley Research Center.

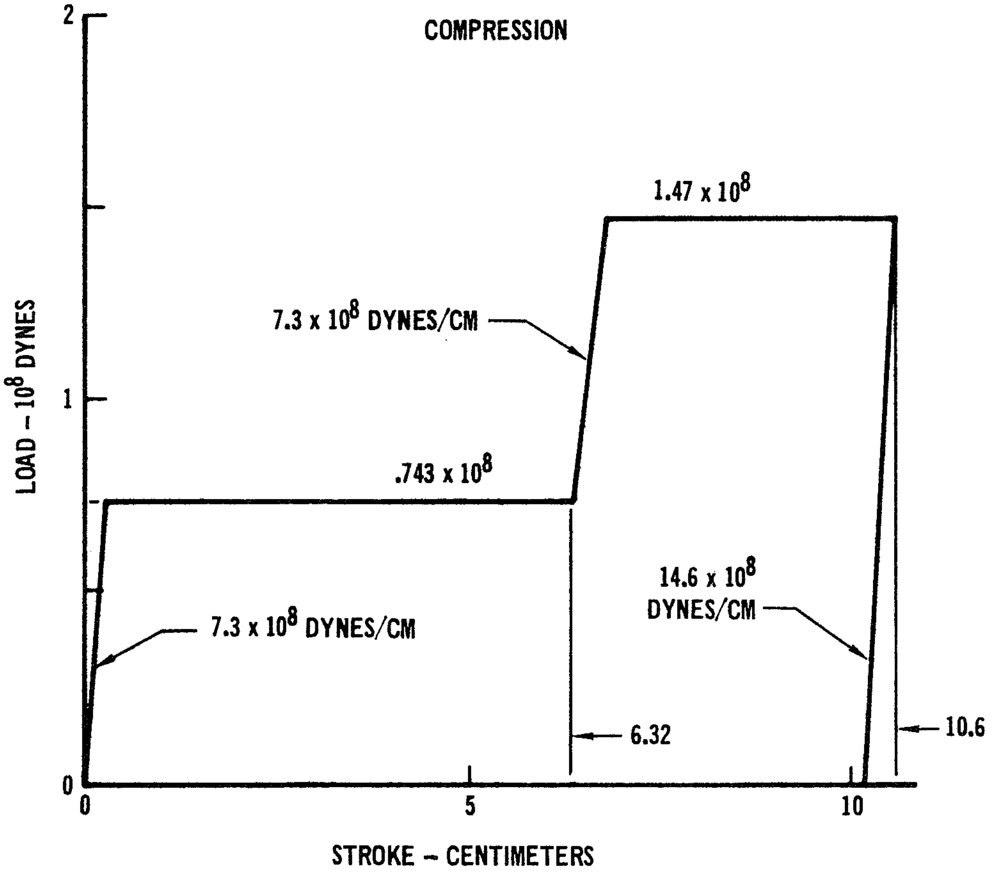


FIGURE 6-4 LOAD-STROKE CURVE FOR MAIN STRUT OF ONE-SIXTH SCALE LM CANTILEVER GEAR

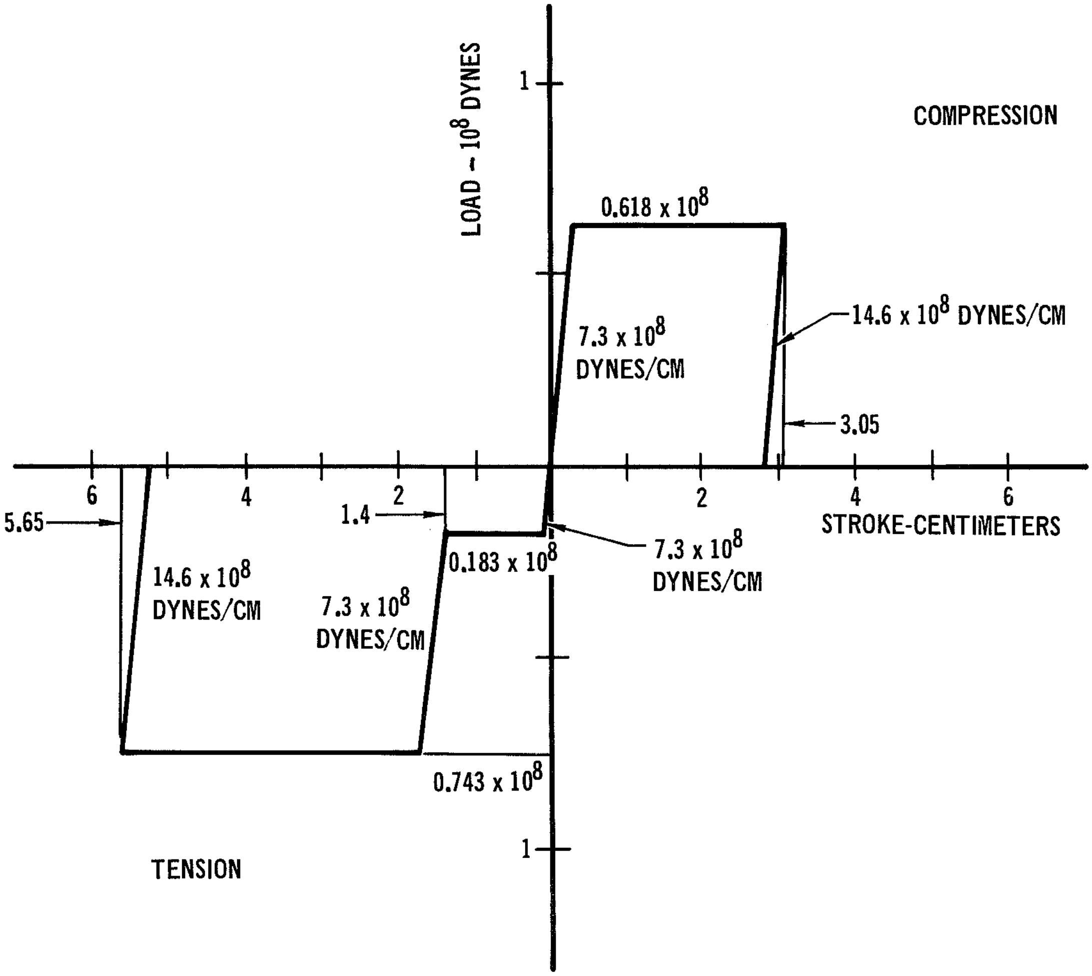


FIGURE 6–5 LOAD-STROKE CURVE FOR DRAG STRUT OF ONE-SIXTH SCALE LM CANTILEVER GEAR

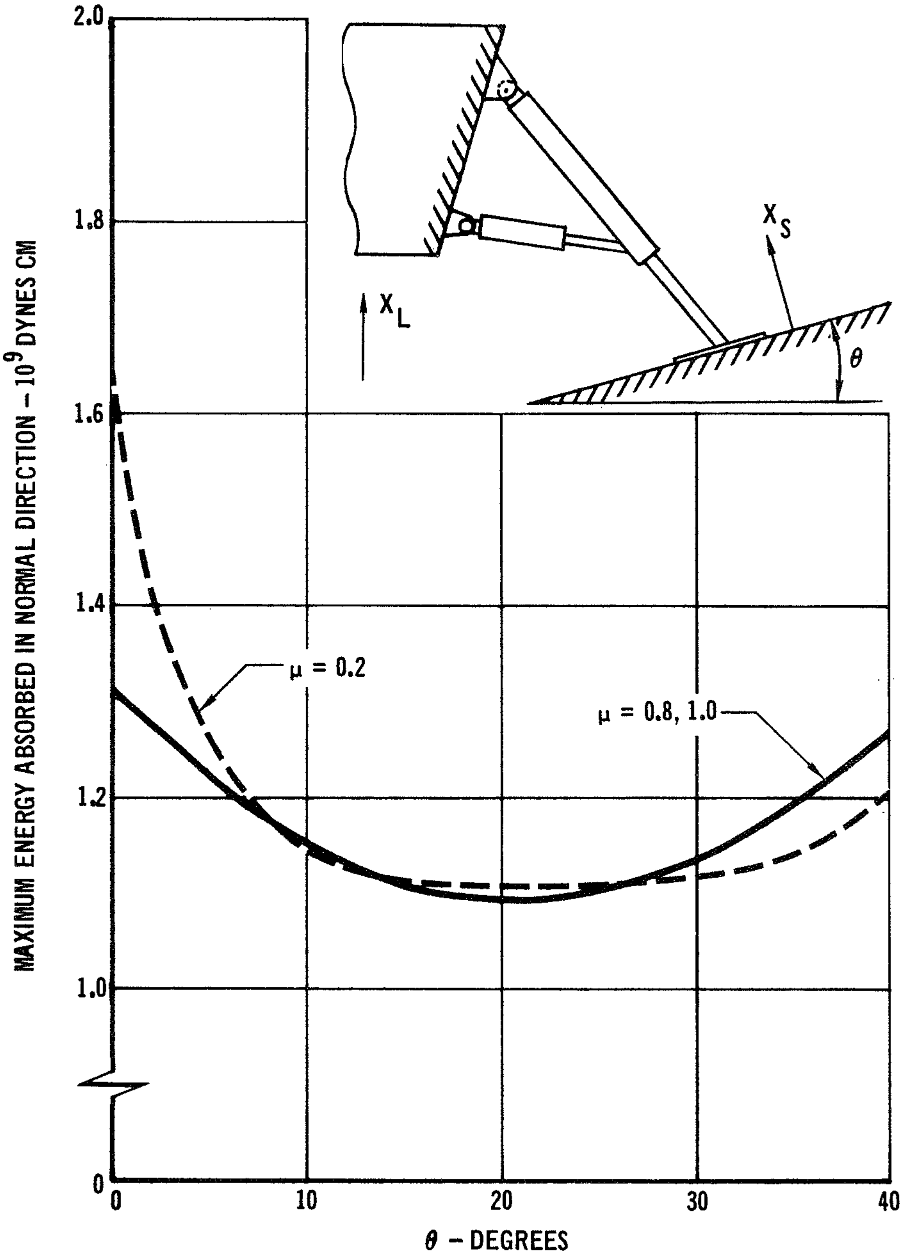


FIGURE 6–6 MAXIMUM ENERGY ABSORPTION FOR ONE-SIXTH SCALE LM CANTILEVER GEAR

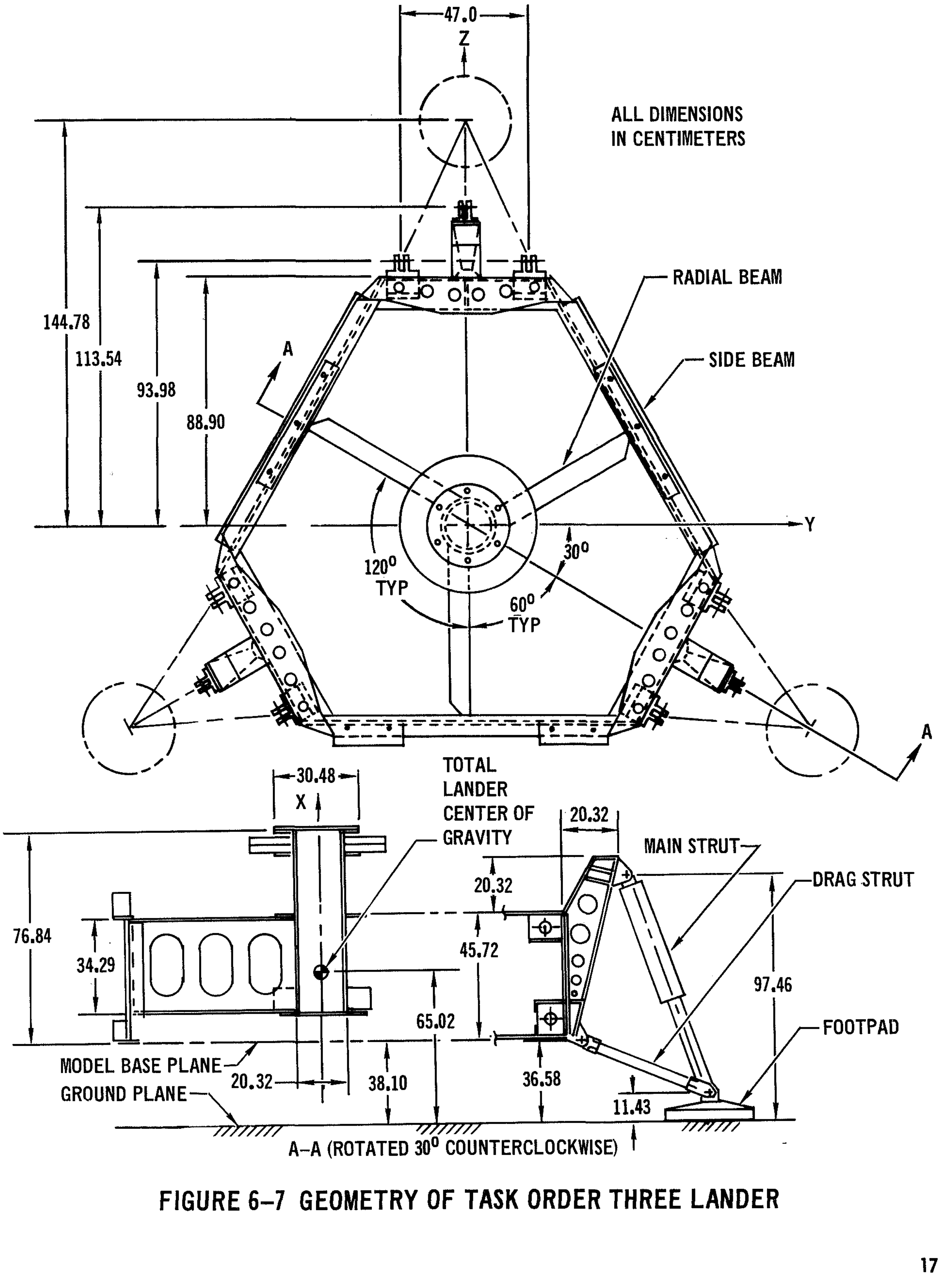


FIGURE 6–7 GEOMETRY OF TASK ORDER THREE LANDER

6.2.1 Modal Analysis of Center Body Structure - The finite element idealization of the center body structure is shown in Figure 6-8. The member stiffness properties for each pair of bars representing a side beam (for example, members 10 and 11) and a radial beam (members 48 and 49) are such that the total stiffness of these deep beams is maintained. The moment carrying capabilities of members 46, 47, 48, 49, 50, and 51 about the lander X axis, where these members attach to the side beams, were removed. This was done to simulate the high flexibility associated with these attach points.

This idealization resulted in a stiffness matrix containing 33 joints with 198 degrees of freedom. Before obtaining the structure's frequencies and mode shapes, the stiffness matrix was reduced to 99 degrees of freedom by removing the three rotational degrees of freedom at each joint. Thus, the input mass matrix contained translational masses associated with all of the joints shown in Figure 6-8.

Twenty elastic modes were requested and the program output for this modal analysis is shown in Appendix B. Three modes, with frequencies of 34.6 Hz, 48.2 Hz, and 101 Hz were obtained whose modal deformation patterns were predominately in the lander Y-Z plane and are the type of modes that would be excited by a landing such as the third drop test of the Task Order Three lander. These are the modes which were included in the data for the Landing Loads and Motions Program, Section 6.2.2, when considering the effects of a flexible center body structure. Figures 6-9 through 6-11 show the mode shapes associated with these three frequencies.

6.2.2 Correlation with Third Drop Test - The predicted landing response for the initial conditions corresponding to NASA Langley's third drop test were obtained considering both a rigid and flexible center body structure. The initial conditions for this drop test condition are shown in Figure 6-12. A portion of the output from the Landing Loads and Motions Program for the flexible body is presented in Appendix D.

The main strut load-stroke relationship shown in Figure 6-1 was used for these correlation runs. A sketch of the actual footpad geometry, including footpad attenuation, and the footpad idealization employed in the landing analyses are shown in Figure 6-13. The Secondary Soil Mechanics routine was employed with the following soil parameters:

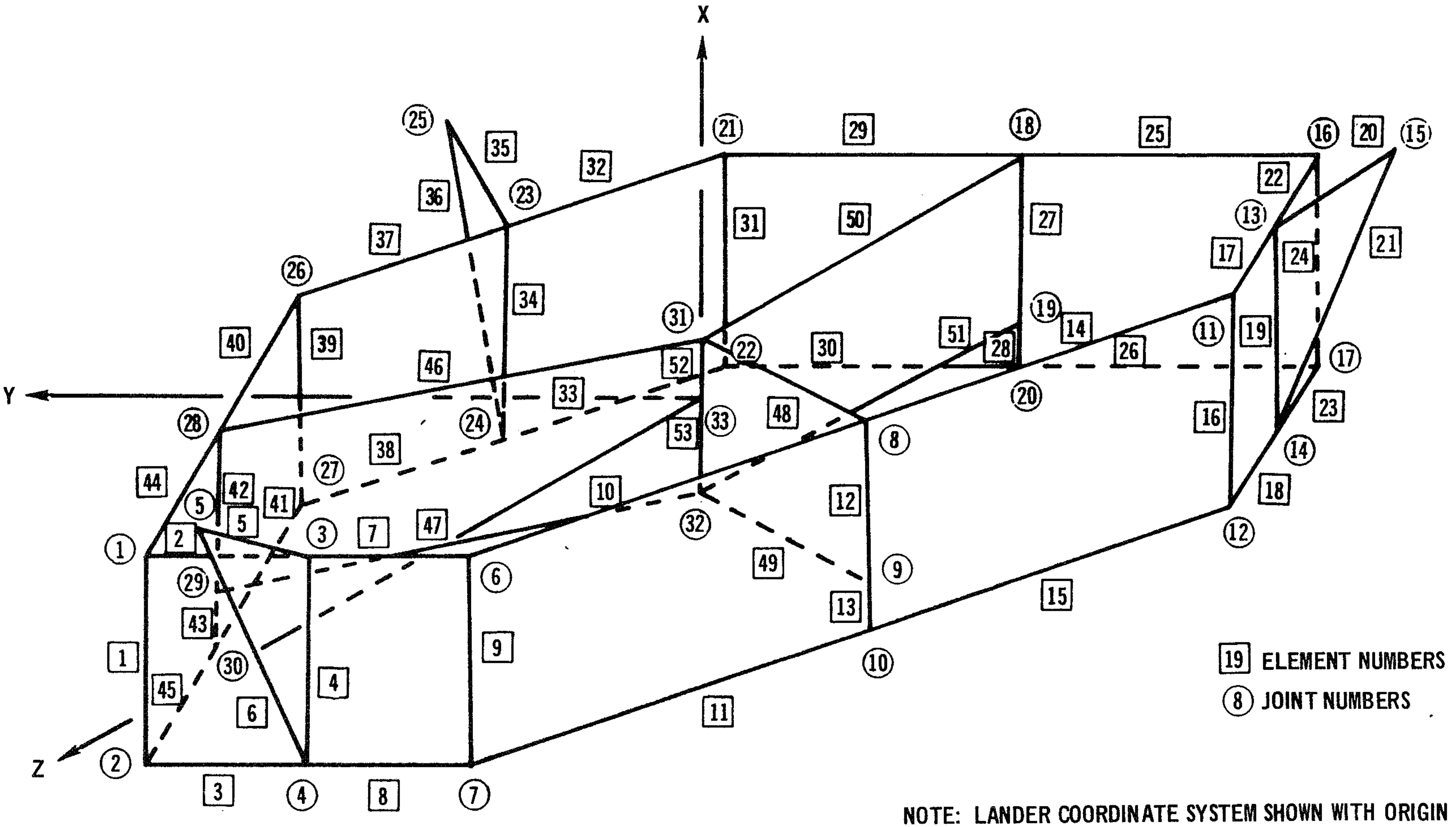
Soil elasticity constant = 1.4 x 10^7 dynes/cm^3
Coefficient of friction = 0.9

The distribution of center body mass and inertia and footpad masses used for the 3/8 mass Task Order Three lander in these runs is summarized in Figure 6-14. This breakdown was obtained by requiring that the combination of the center body mass moments of inertia and the footpad masses, multiplied by the square of their respective moment arms, approximated the moments of inertia of the total 3/8 mass Task Order Three lander. In addition, the center of gravity location for the total lander was maintained with this mass distribution.

Predicted center of gravity acceleration time histories and loads in the outside drag strut of leg 2 for a rigid center body are compared with the experimental data in Figures 6-15 and 6-16. The high frequency oscillations present in both the predicted load and acceleration results are the spring-mass effect of the footpad mass in conjunction with the landing gear struts. In addition, oscillation in the drag strut load near the end of the time history (when the footpad is off the ground) is the gear-mass system experiencing free vibration.

Comparison between measured and predicted center of gravity accelerations, main strut loads, main strut strokes, and drag strut loads when center body flexibility was included, is shown in Figures 6-17 through 6-22. Correlation between analytical and experimental results is good in this case. The inclusion of center body flexibility results in a less rapid buildup of center of gravity accelerations upon initial impact, especially in the Z direction. In addition, the inclusion of structural flexibility can readily be seen by the low frequency oscillation present in the center of gravity Z acceleration.

An interesting point is indicated in Figure 6-22. It can be seen that there is an initial oscillatory load in this drag strut before the footpad impacts the landing surface. This is the result of the inertia loading of the footpad mass on the end of the elastic landing gear struts. This inertia loading effect of the noncontacting footpad is also indicated in Figure 6-18 by the buildup of main strut load in leg 1 before footpad impact.



NOTE: LANDER COORDINATE SYSTEM SHOWN WITH ORIGIN AT CENTER OF GRAVITY OF CENTER BODY.

FIGURE 6–8 CENTER BODY STRUCTURAL IDEALIZATION

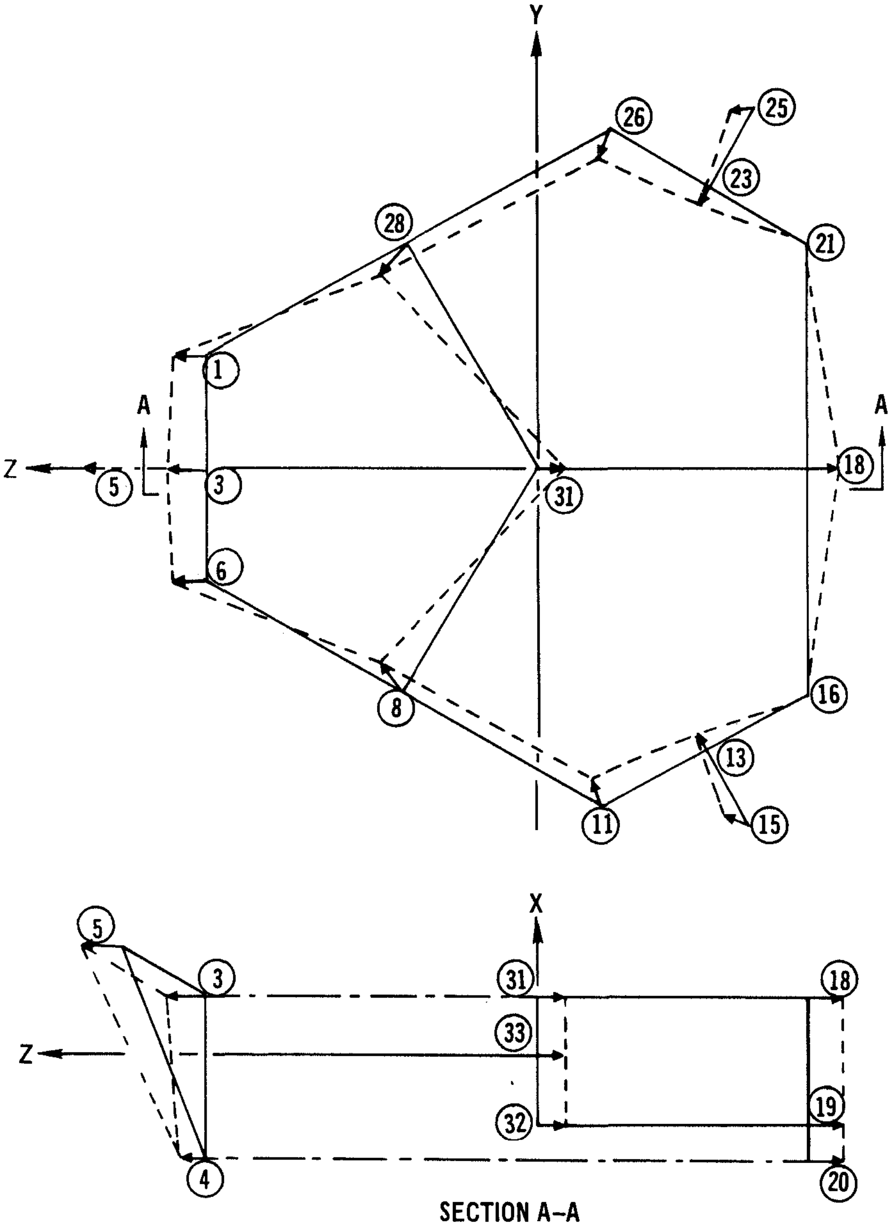


FIGURE 6–9 CENTER BODY MODE SHAPE ASSOCIATED WITH 34.6 H_Z MODE

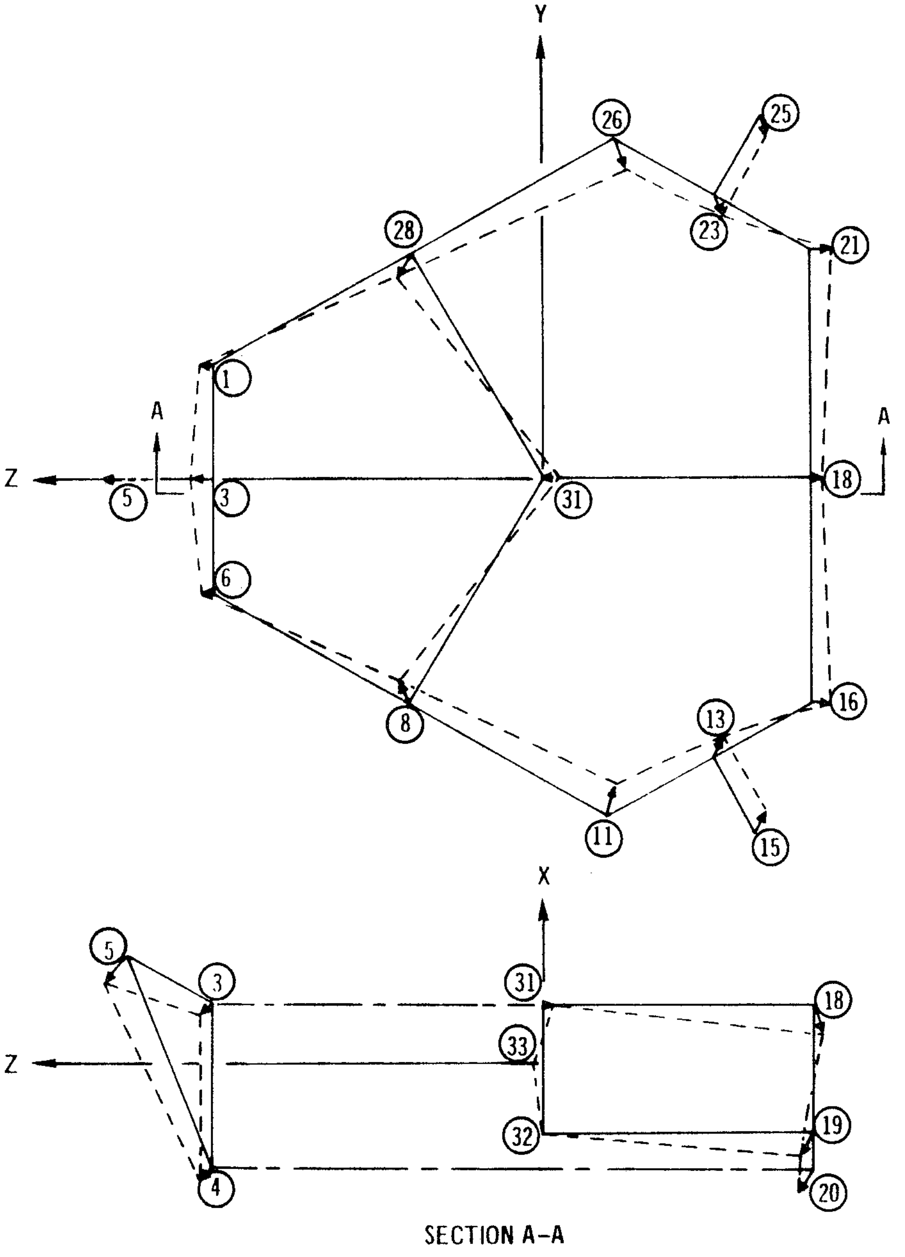


FIGURE 6–10 CENTER BODY MODE SHAPE ASSOCIATED WITH 48.2 H_Z MODE

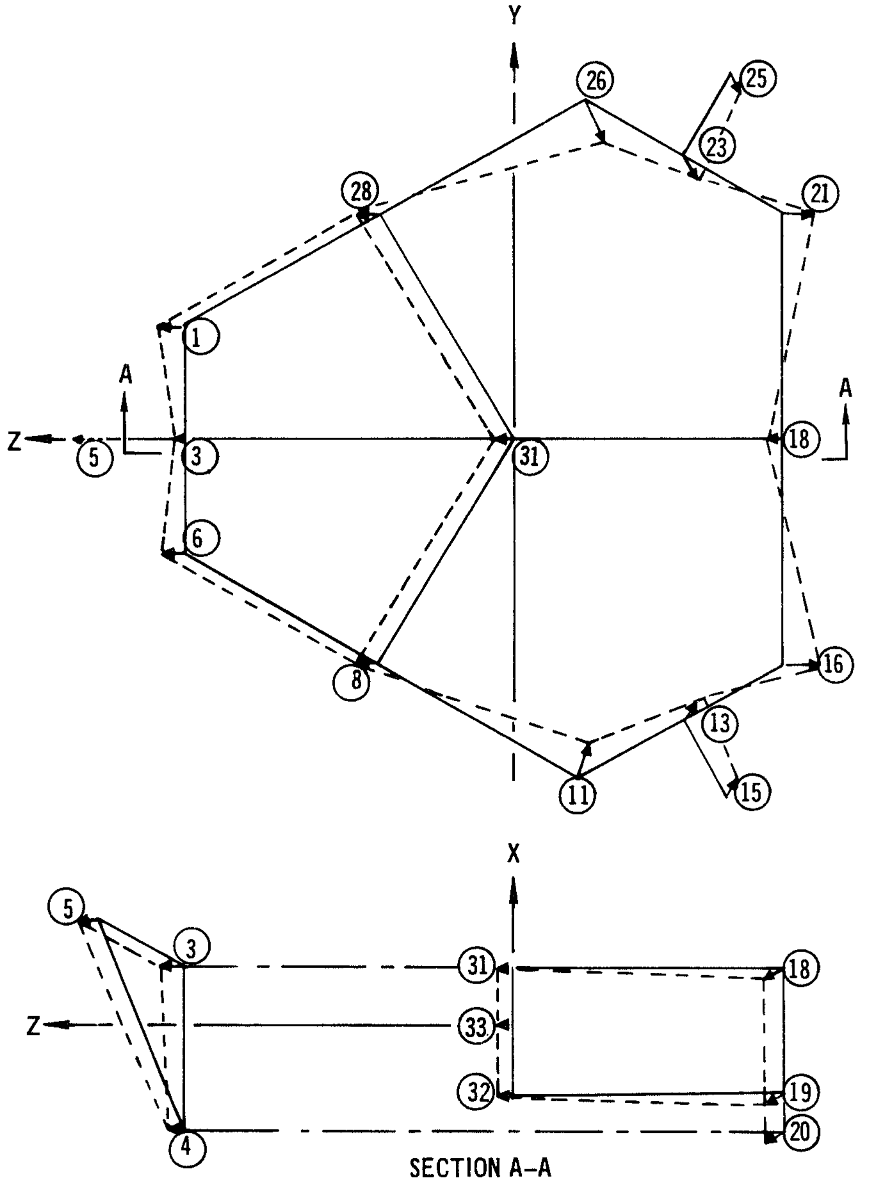


FIGURE 6–11 CENTER BODY MODE SHAPE ASSOCIATED WITH 101 H_Z MODE

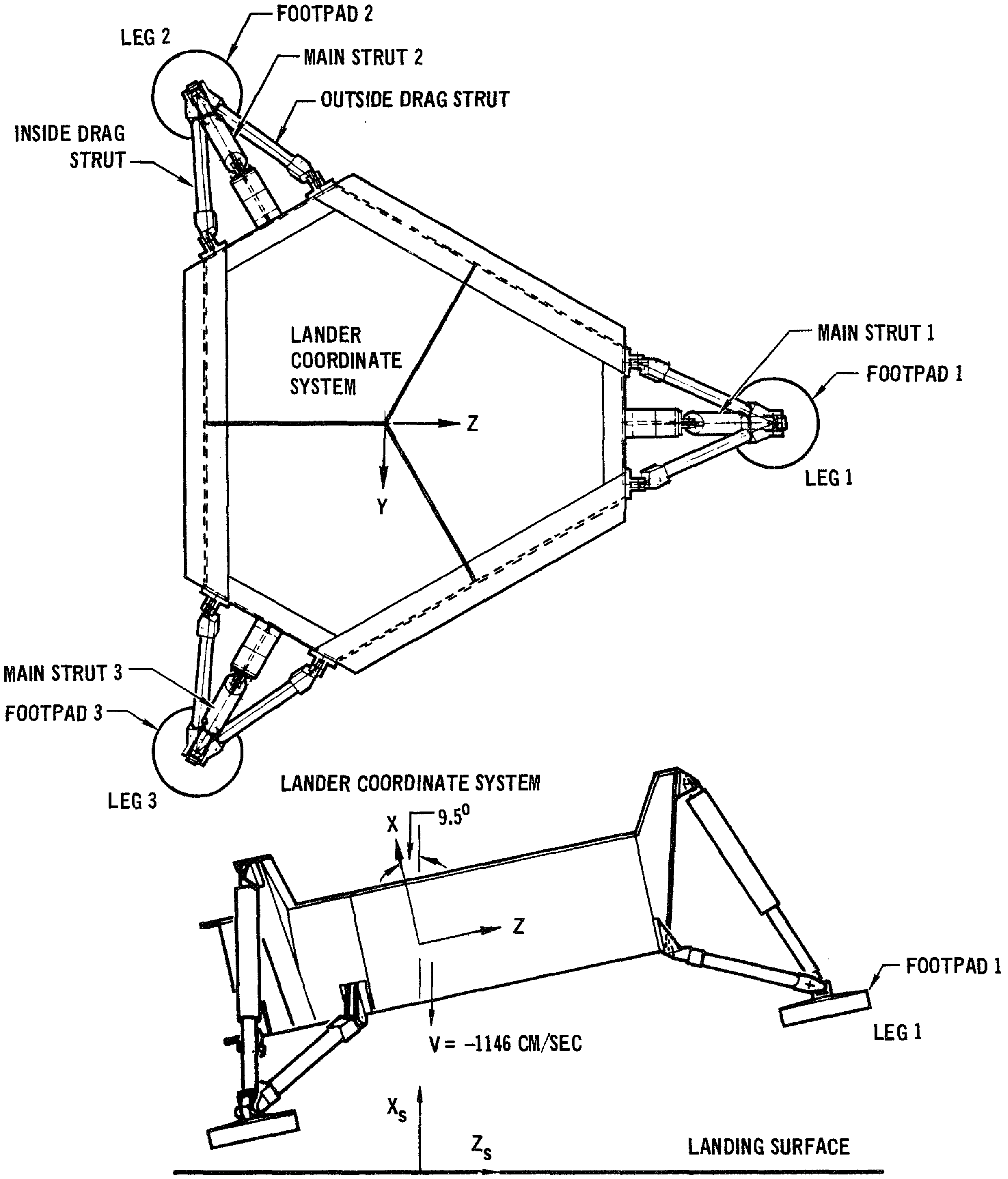
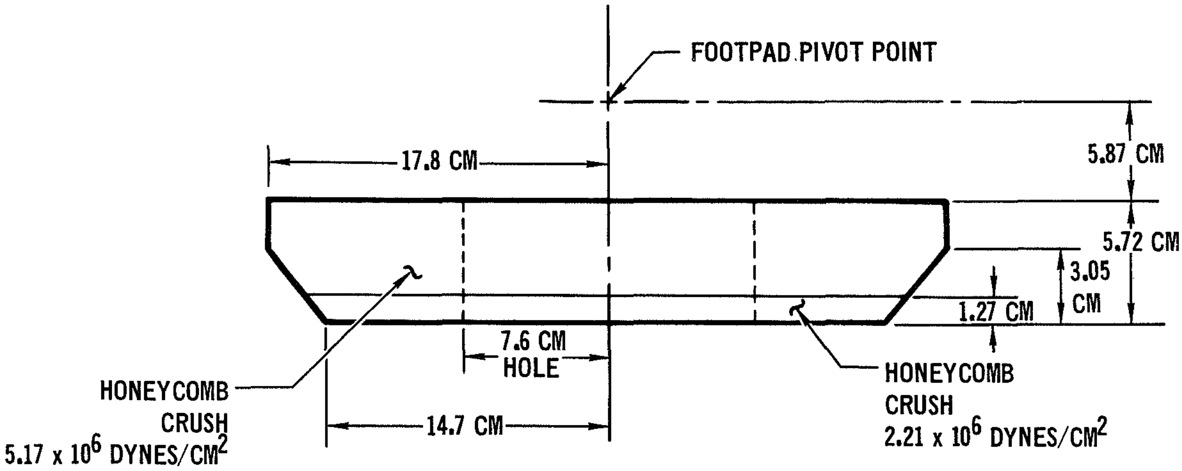
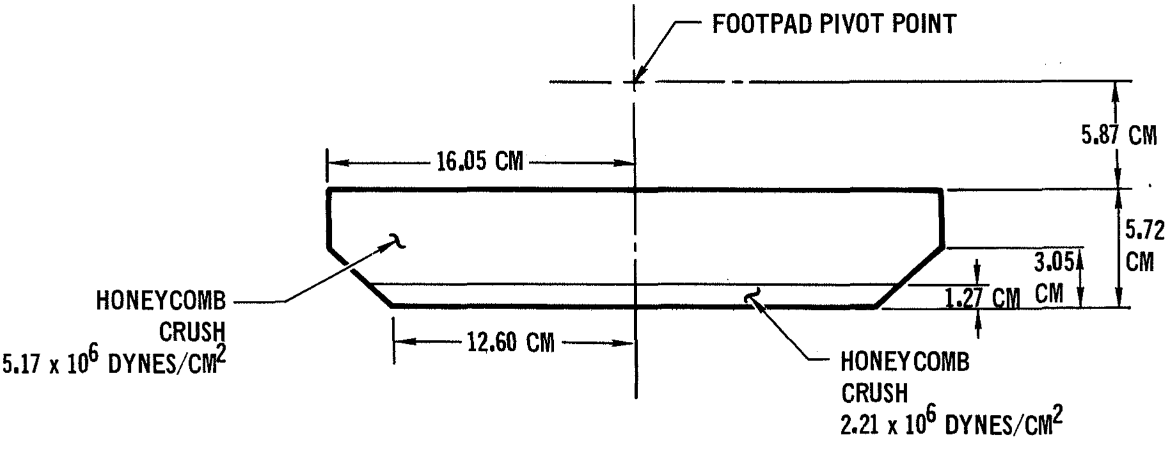


FIGURE 6–12 INITIAL CONDITIONS FOR DROP TEST THREE



ACTUAL FOOTPAD



IDEALIZED FOOTPAD

FIGURE 6-13 FOOTPAD IDEALIZATION

ITEM	MASS (GRAMS)	X* (CM)	Y (CM)	Z (CM)	I_{xx} (GRAM-CM^2)	I_{yy} (GRAM-CM^2)	I_{zz} (GRAM-CM^2)
CENTER BODY	180,500	67.97	0	0	8.031×10^8	4.992×10^8	4.968×10^8
FOOTPAD 1	3,497	11.43	0	144.78	–	–	–
FOOTPAD 2	3,497	11.43	–125.37	– 72.39	–	–	–
FOOTPAD 3	3,497	11.43	125.37	– 72.39	–	–	–

* RELATIVE TO BOTTOM OF FOOTPAD

FIGURE 6–14 3/8 MASS TASK ORDER THREE LANDER MASS BREAKDOWN

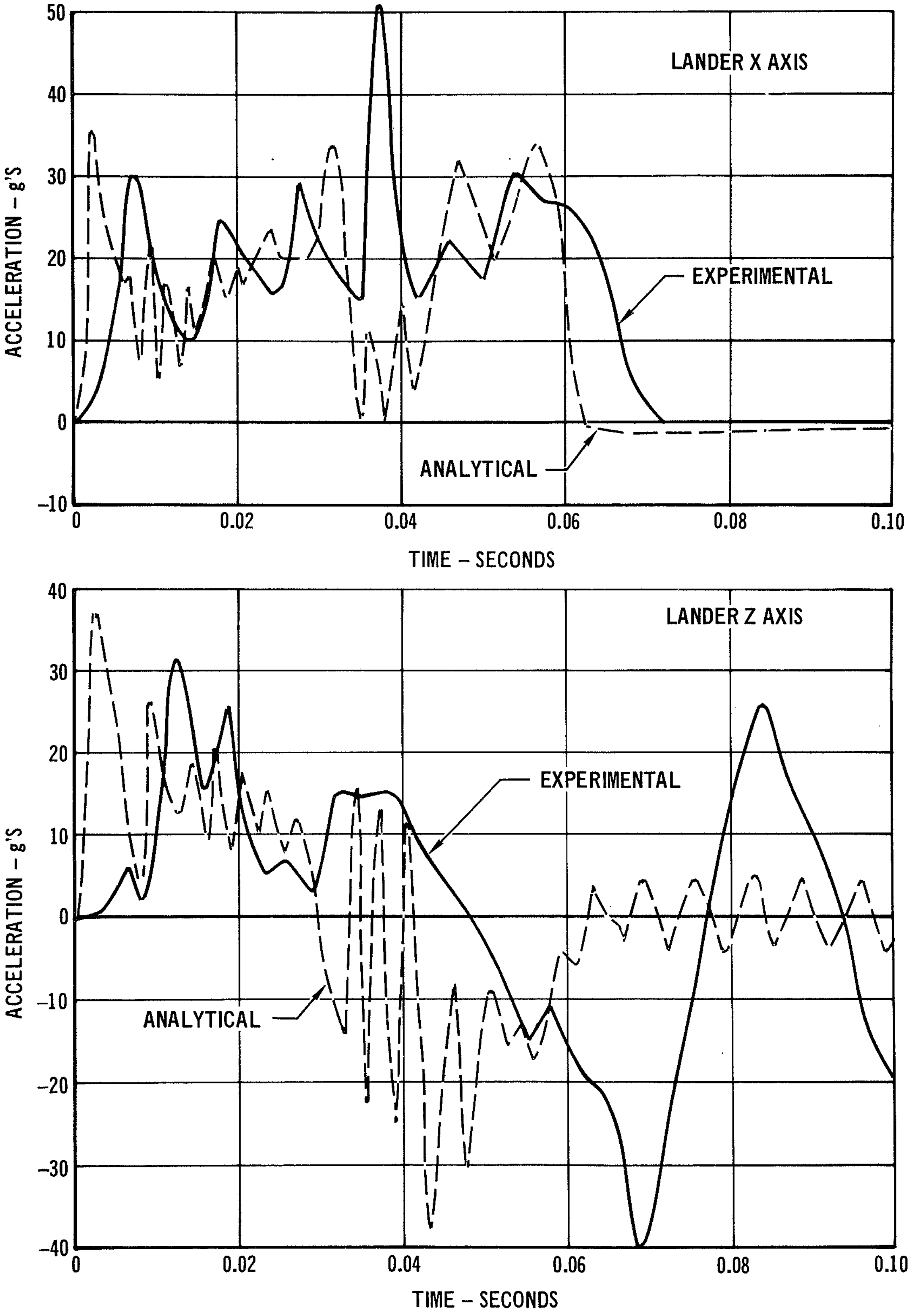


FIGURE 6–15 CENTER OF GRAVITY ACCELERATIONS WITH RIGID CENTER BODY

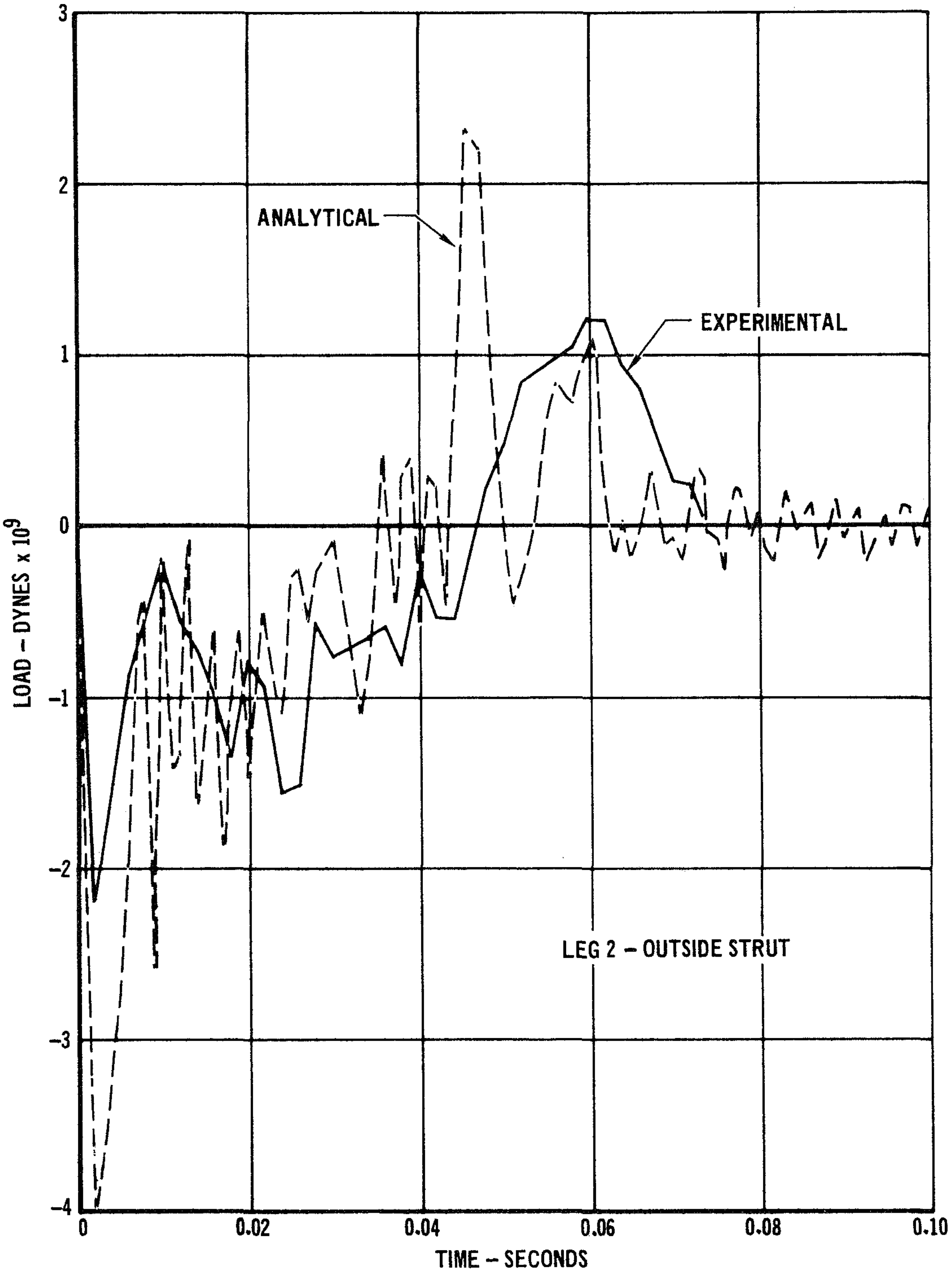


FIGURE 6–16 OUTSIDE DRAG STRUT LOAD WITH RIGID CENTER BODY

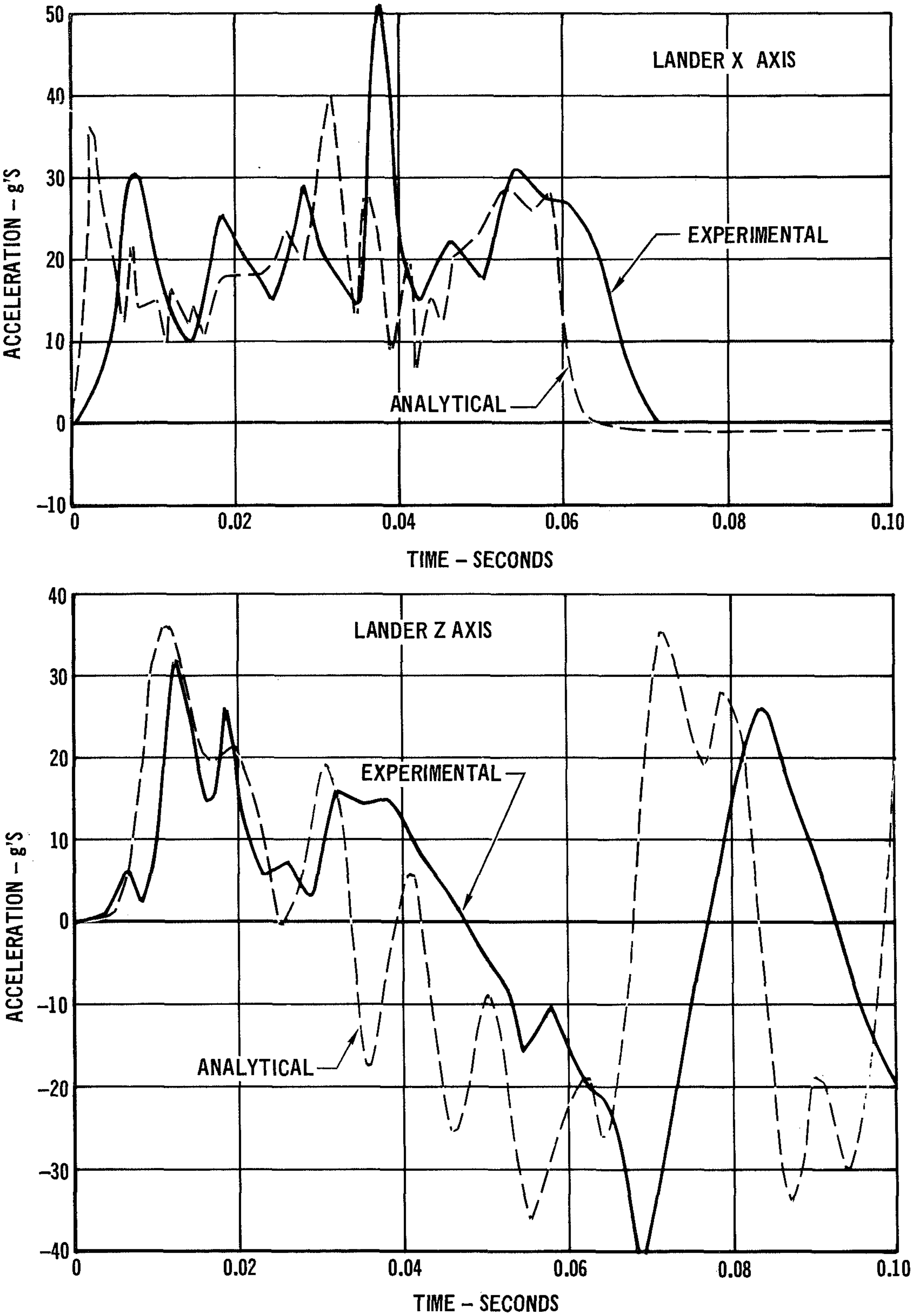
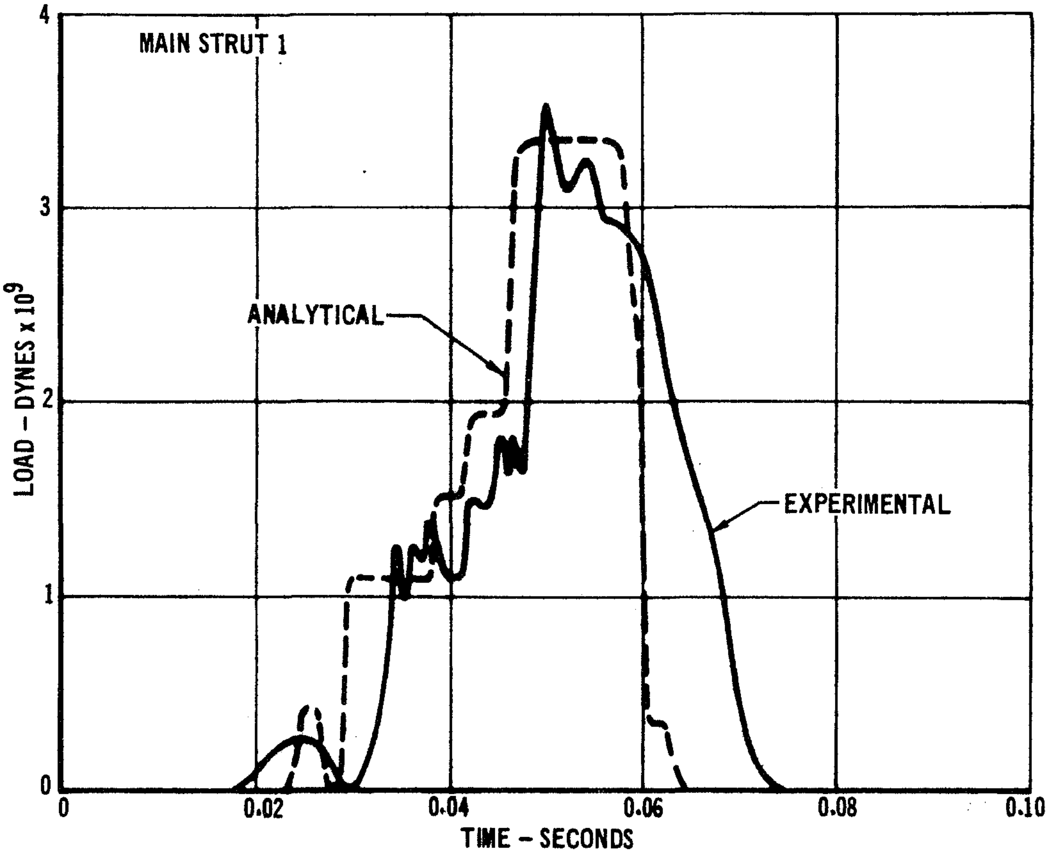


FIGURE 6–17 CENTER OF GRAVITY ACCELERATIONS WITH FLEXIBLE CENTER BODY



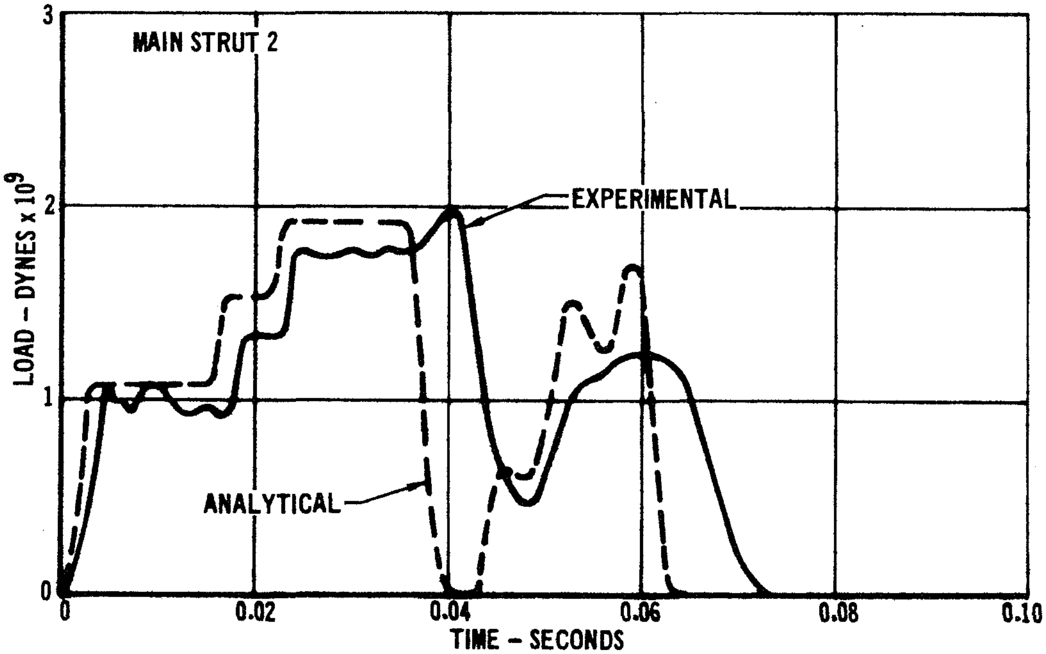


FIGURE 6-18 MAIN STRUT LOADS WITH FLEXIBLE CENTER BODY

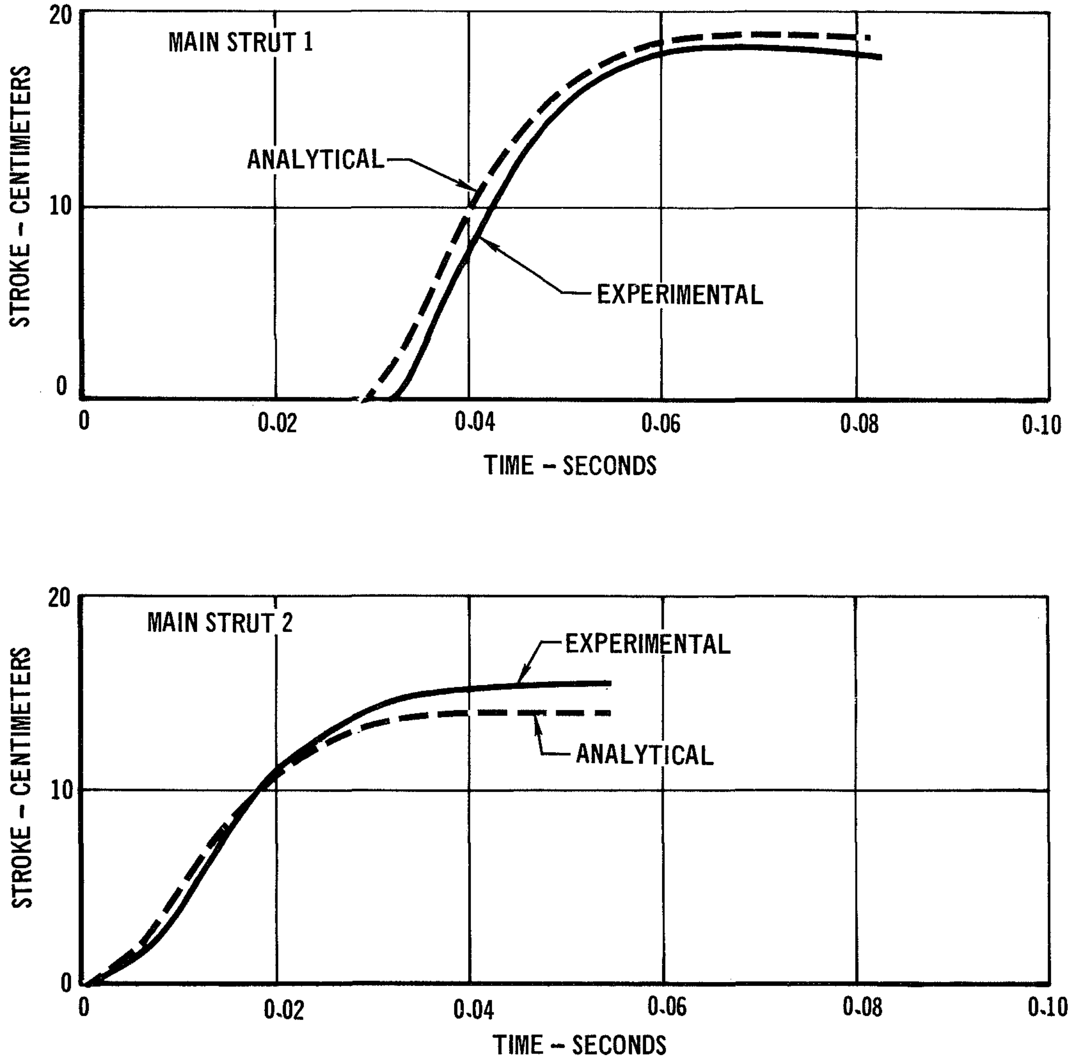


FIGURE 6–19 MAIN STRUT STROKES WITH FLEXIBLE CENTER BODY

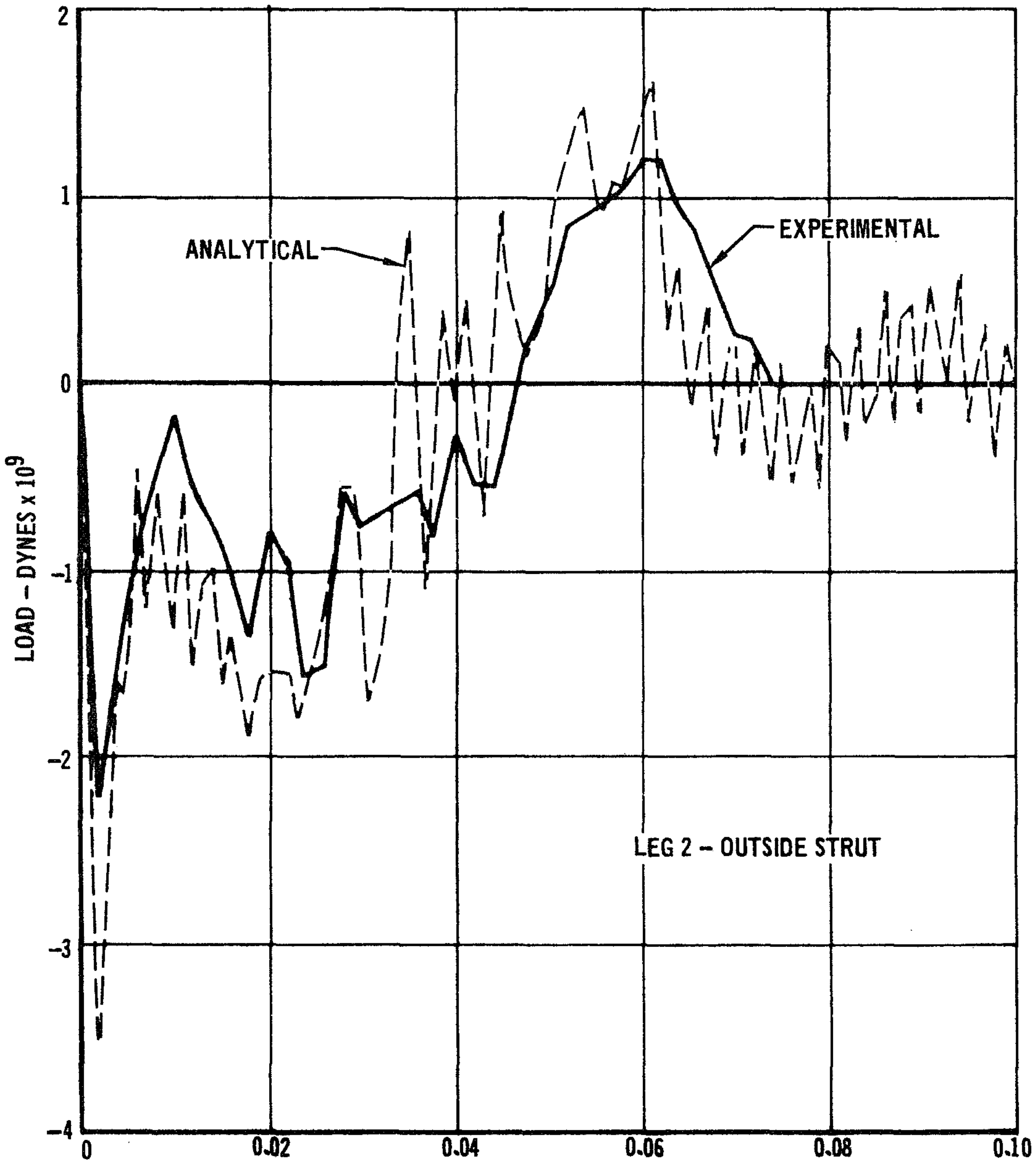


FIGURE 6-20 OUTSIDE DRAG STRUT LOAD WITH FLEXIBLE CENTER BODY

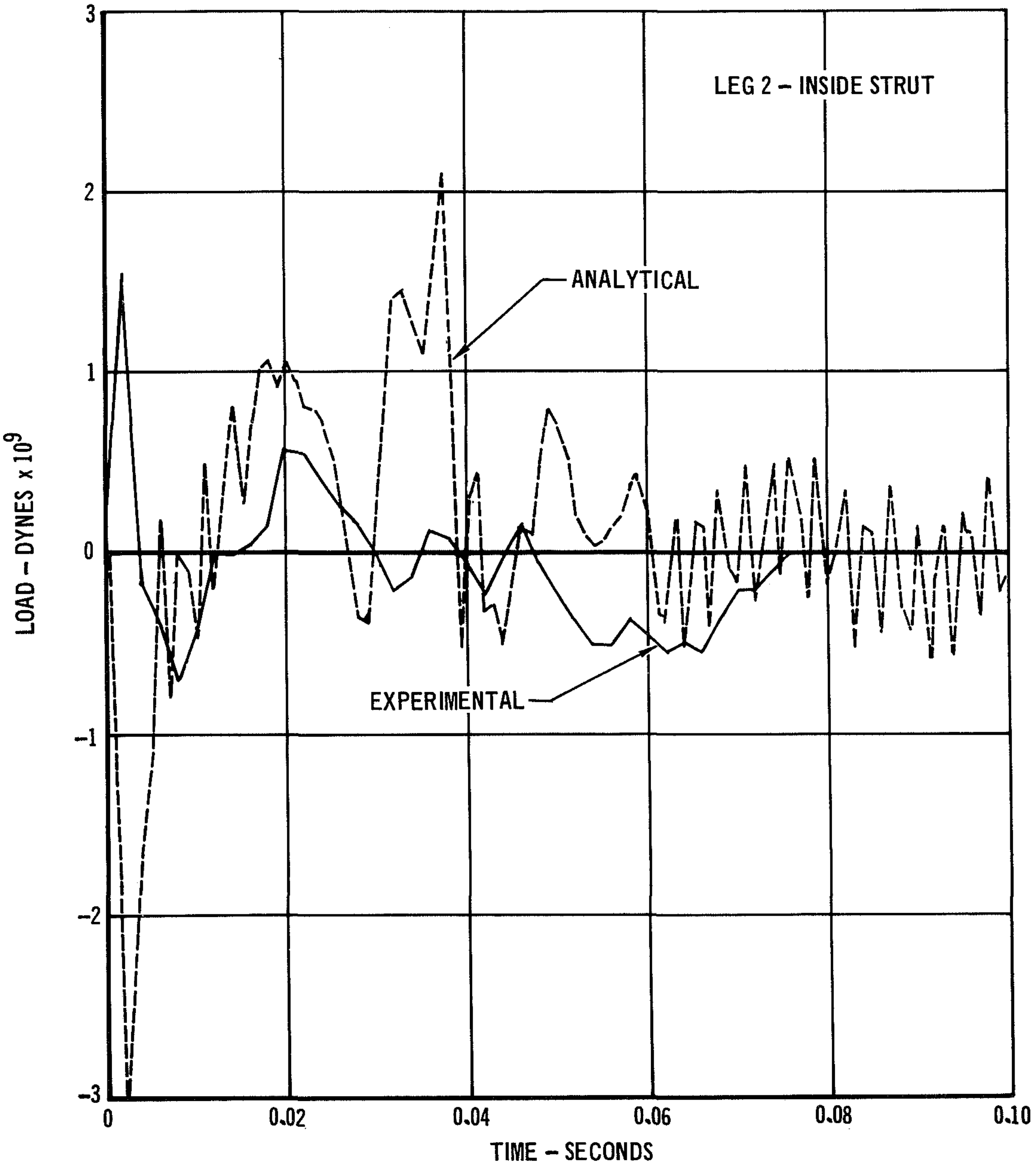


FIGURE 6–21 INSIDE DRAG STRUT LOAD WITH FLEXIBLE CENTER BODY

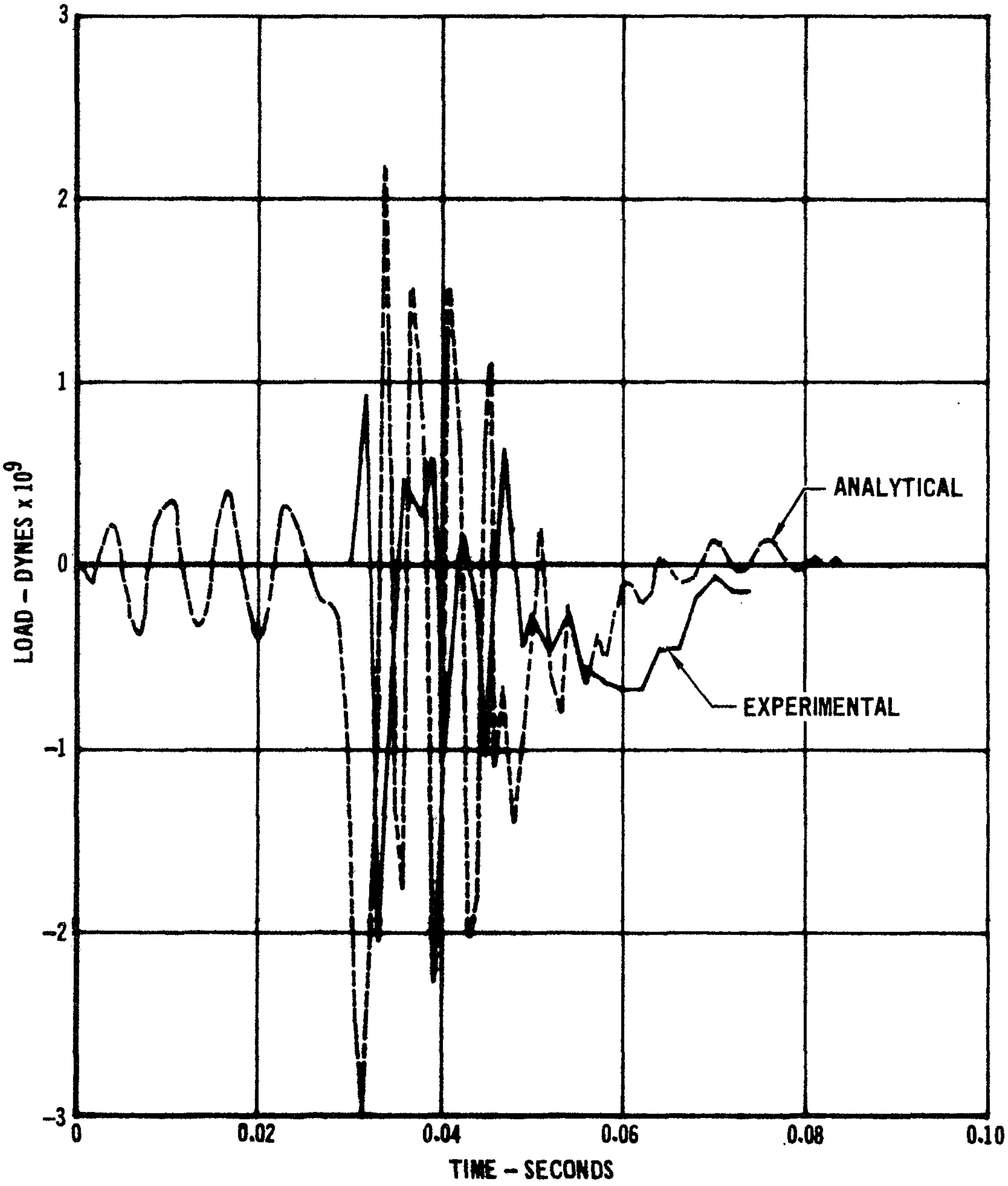


FIGURE 6-22 LEG 1 DRAG STRUT LOAD WITH FLEXIBLE CENTER BODY

6.2.3 Center Body Landing Loads - Observing the center of gravity acceleration time histories shown in Figure 6-17, it is seen that there are several points of apparent high acceleration of the center body structure. The time history data stored on magnetic tape during this run was retrieved with the Center Body Landing Loads Program and total load distributions on the center body were determined at several of these points during the landing. The load distribution at the landing time of 0.011 seconds was selected as critical because it yielded the largest loads in the -Z direction at the center body center of gravity. Appendix E presents the load distribution obtained with the Center Body Landing Loads Program for this time.

6.2.4 Center Body Internal Load Distribution - The Center Body Option of the Structural Analysis Program was employed to determine joint displacements and internal load distributions in the center body structure for landing loads defined in Section 6.2.3 at a time of 0.011 seconds. The center body idealization shown in Figure 6-8 was employed. A listing of input information and output data for this problem is presented in Appendix A. Header card (see Figure 4-32) for this data case does not appear in the output listing although it was the first card of the data set.

In the center body finite element idealization, attach points for legs 2 and 3 were assumed to be joints 12, 15, 17, 22, 25, and 27. This is an approximation since the physical attach points for the drag struts are located at the ends of clevis fittings which are adjacent to the above joints. The center body was supported at joints 12, 15, 17, 22, 25 and 27 which were assumed to be pinned with zero translational displacements and zero applied moments. The center body landing loads distribution at 0.011 seconds is given in Appendix E. Due to the assumed drag strut locations in the center body idealization, unbalances in these loads are indicated, especially moment about the lander Y axis. To insure a static balance, the negative of these unbalance loads were applied at the center of gravity (joint 33) in addition to the three force components at this joint. For computer input, nominal values were selected by default for all data option indicators except INDITR and TMAX. INDITR (maximum number of iterations allowed) was set at 2000. TMAX (maximum CP seconds allowed for solution) was specified as 100.

Included in computer output are displacements and external loads for all joints and internal loads in all bar elements. The displacements at the center of gravity and the internal bending moments about the local Z axis for the radial beam (elements 46 through 51) are of particular significance. Results of this analysis are compared with the results for a redesigned center body structure in Figure 6-28, Section 6.3.

In order to emphasize the importance of proper idealization of the structure when determining displacements and internal loads, the loading condition defined above was applied to a refined idealization of the Task Order Three Lander Center Body as discussed in Appendix G. Displacements and internal loads obtained when employing the refined idealization are different than those obtained when employing the idealization in Figure 6-8 and probably are more accurate. For the purpose of demonstrating program capabilities, the idealization shown in Figure 6-8 is entirely adequate and will be employed in the following sections.

It must be kept in mind that a more refined idealization requires more computer run time. For example, run time for the idealization in Figure 6-8 was 46 CP seconds while that for the refined idealization in Appendix G was 106 CP seconds. A compromise between idealization refinement and computer run time must be made.

6.3 <u>Analysis of Modified Task Order Three Lander</u> - The analysis of the Task Order Three lander, Section 6.2, indicated considerable flexibility of the center body structure in the lander Y-Z plane. This inplane flexibility is indicated in the measured Z center of gravity acceleration, Figure 6-15, obtained during the NASA drop test program.

A modification of the Task Order Three lander center body structure to increase the stiffness and improve the load carrying capability in the Y-Z plane was studied. Three configurations, incorporating the addition of a number of tension straps, were considered in arriving at the Modified Task Order Three lander. These consisted of 9-strap, 12-strap, and 18-strap configurations. The 12-strap configuration is shown in Figure 6-23. The 9-strap configuration was the same as the 12-strap configuration with the upper straps 63, 64, and 65 removed. The 18-strap configuration was like the 12-strap with six additional straps added to the upper portion of the center body. These upper straps were located in a manner similar to the lower straps 57 through 62.

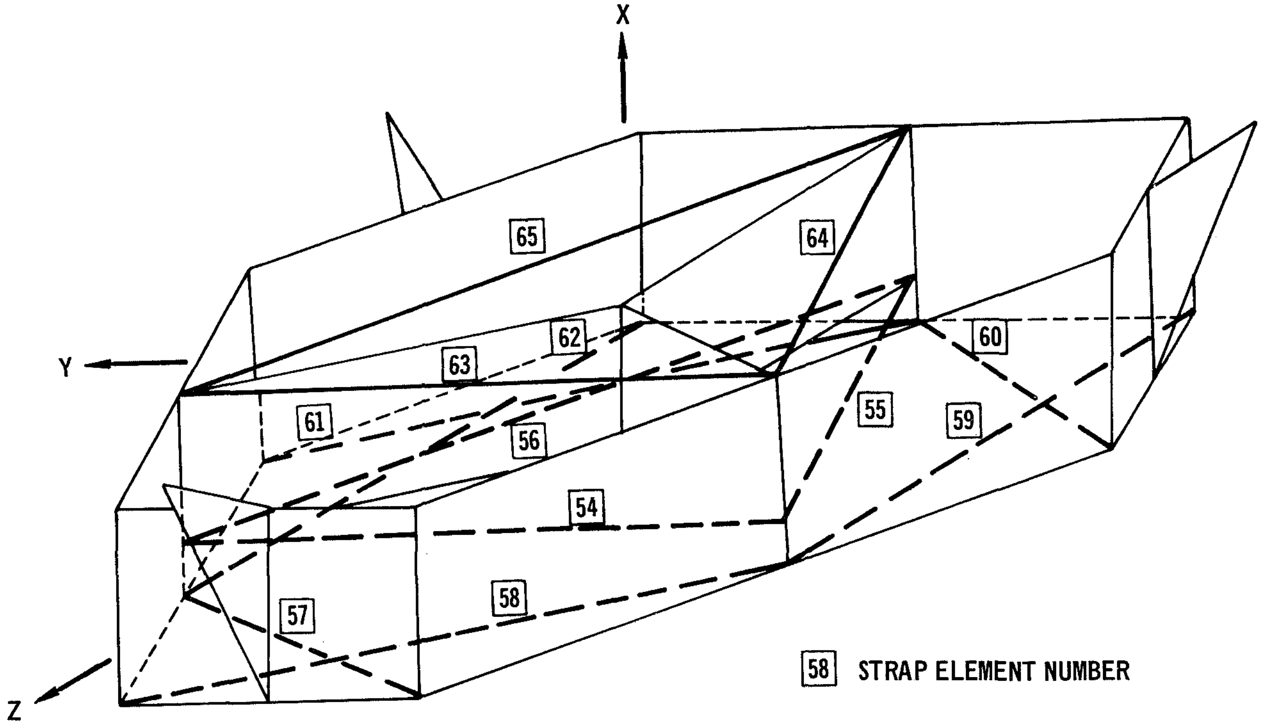
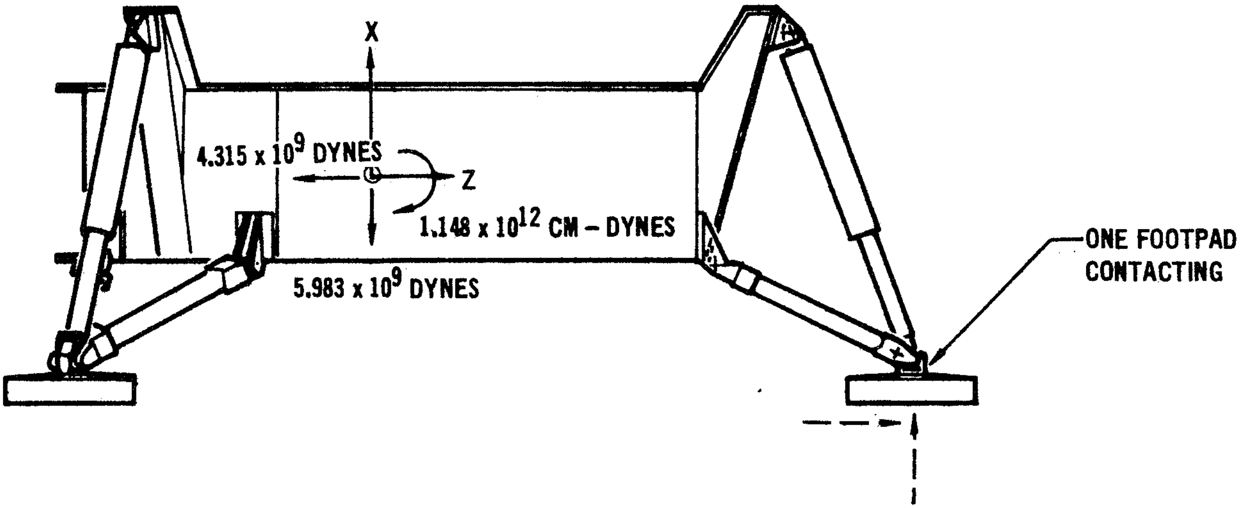


FIGURE 6-23 CENTER BODY IDEALIZATION INCLUDING TENSION STRAPS

A representative loading condition was applied to a finite element idealization of these three strap configurations employing the Center Body Option of the Structural Analysis Program. For this study, the basic center body structure was idealized as shown in Figure 6-8. Each set of tension straps was then added to this idealization, as indicated in Figure 6-23 for the 12-strap configuration. In each case, all straps were assumed to have a cross-sectional area of 1.29 square centimeters and a modulus of elasticity of 2.068×10^{12} dynes/cm^2.

Figure 6-24 shows the loading condition which was employed to evaluate each of the tension strap configurations. This condition was found to be the critical one when the most severe center of gravity accelerations noted during the analysis of the Task Order Three lander in Section 6.2 were combined for one and two leg landings. Employing the criterion of reducing both the radial beam bending moments about the local Z axes and inplane displacements, the comparison shown in Figure 6-25 was made for the three tension strap configurations. Based on the results shown in this figure, the 12-strap configuration was chosen as the Modified Task Order Three lander. This configuration resulted in a 3.1 reduction factor for radial beam bending moment and a 5.45 multiplication factor for center body inplane stiffness when compared to the Task Order Three lander structure. The 18-strap configuration was not selected because the slight improvement in the parameters of interest was not sufficient to justify six additional straps. The weight of the twelve tension straps was removed from the ballast weights in the idealization of the modified Task Order Three configuration to maintain the same total lander weight.



LOADS APPLIED AT CENTER OF GRAVITY WERE DETERMINED BY ASSUMING:

+ X ACCELERATION OF 50 g's
+ Z ACCELERATION OF 36 g's

WEIGHT OF HUB AND RADIAL BEAMS = 1.196×10^8 DYNES, CONCENTRATED AT CENTER OF GRAVITY

THEREFORE, LOADS SHOWN ARE:

$F_X = -m\ddot{X} = -5.983 \times 10^9$ DYNES

$F_Z = -m\ddot{Z} = -4.315 \times 10^9$ DYNES

M_Y = MOMENTS SUMMED ABOUT CONTACTING FOOTPAD = -1.148×10^{12} CM–DYNES

FIGURE 6-24 LOADING CONDITION FOR EVALUATION OF MODIFIED TASK ORDER THREE LANDER CENTER BODY STRUCTURE

STRAP CONFIGURATION	MAXIMUM RADIAL BEAM TRANSVERSE BENDING MOMENT* (CM–DYNES ULT.)	CENTER BODY INPLANE STIFFNESS** (DYNES/CM)
NONE	8.203×10^{10}	1.601×10^{9}
9	3.378×10^{10}	5.366×10^{9}
12	2.649×10^{10}	8.748×10^{9}
18	1.625×10^{10}	9.159×10^{9}

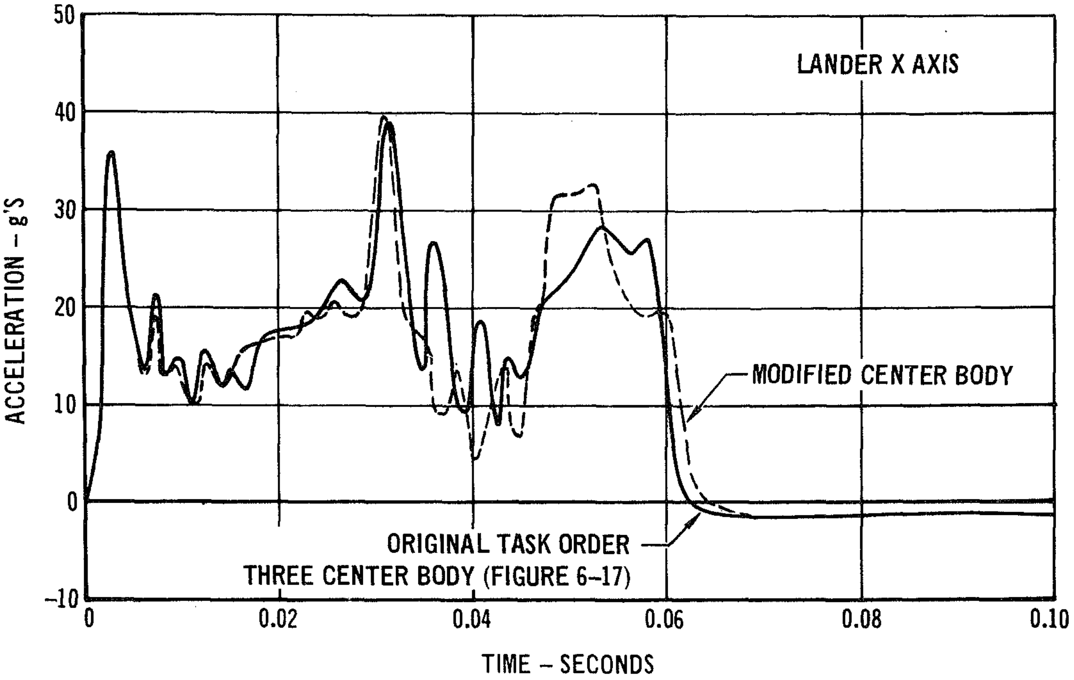
* BENDING MOMENTS ARE ABOUT LOCAL Z AXIS OF ELEMENT AT "Q" END

** CENTER BODY INPLANE STIFFNESS WAS OBTAINED BY APPLYING A UNIT LOAD AT THE CENTER OF GRAVITY IN THE –Z GLOBAL DIRECTION. THE STIFFNESS FACTOR IS THIS LOAD DIVIDED BY THE RESULTING CENTER OF GRAVITY GLOBAL Z DISPLACEMENT

FIGURE 6–25 COMPARISON OF VARIOUS TENSION STRAP CONFIGURATIONS

Employing the finite element idealization shown in Figures 6-8 and 6-23, a modal analysis was performed on the Modified Task Order Three lander center body. Three modes, with frequencies of 75.2 Hz, 149 Hz, and 163 Hz, were obtained whose modal deformations were predominantly in the lander Y-Z plane. These modal frequencies are much higher than the comparable frequencies obtained for the Task Order Three Lander, Section 6.2.1, thus illustrating the stiffening effect of the straps.

The drop test three landing conditions were used to evaluate the Modified Task Order Three lander center body structure. Using the above three modes to represent the center body flexibility effects, the accelerations and drag strut load shown in Figures 6-26 and 6-27 were predicted with the Landing Loads and Motions Program. These quantities are superimposed on the predicted results obtained for the original Task Order Three lander center body. The oscillatory nature of the Z acceleration is due to the large excitation of the 75.2 Hz mode for the initial conditions considered.



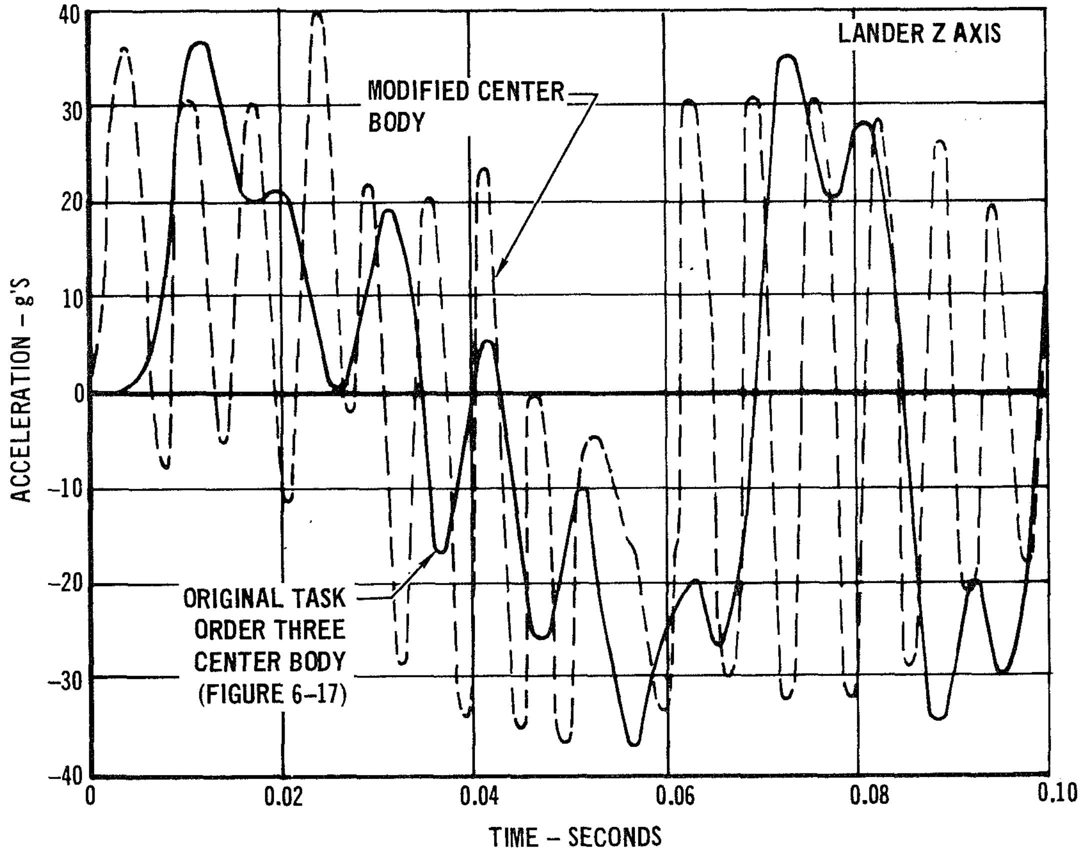


FIGURE 6-26 CENTER OF GRAVITY ACCELERATIONS WITH MODIFIED CENTER BODY STRUCTURE

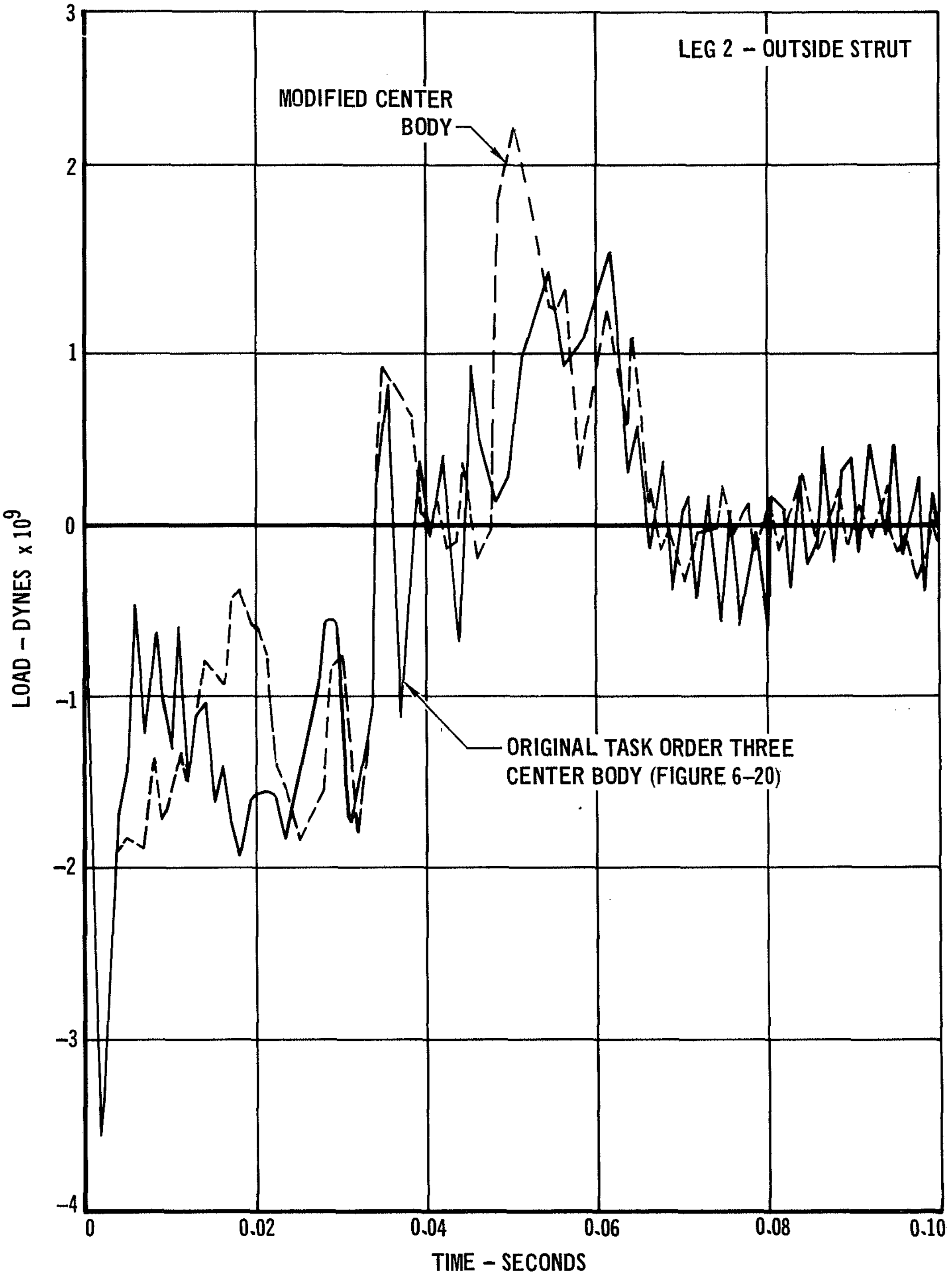


FIGURE 6–27 OUTSIDE DRAG STRUT LOAD WITH MODIFIED CENTER BODY STRUCTURE

The distribution of landing loads throughout the center body structure was obtained for a number of points during the landing time history. At a real time of 0.011 seconds, the center body experienced high loads and this point was chosen for analysis with the Center Body Option of the Structural Analysis Program. Employing the center body finite element idealization shown in Figures 6-8 and 6-23, internal member loads and joint displacements at this time were determined. A comparison of these loads and displacements with the loads and displacements obtained for the Task Order Three lander, Section 6.2.4, are presented in Figure 6-28. As can be seen, the radial beam bending moments in the Modified Task Order Three center body are lower than those obtained for the Task Order Three Center body. In addition, the global Z displacement of the center body center of gravity is less for the Modified Task Order Three center body. This trend of lower member loads and reduced displacements is a result of the additional center body load paths and increased stiffness provided by the tension straps.

6.4 Landing Analysis of Full-Mass Task Order Three Lander - Several additional studies for the Task Order Three lander were performed. For all of the following discussions, the structural configuration, landing gear properties, and footpad idealization were the same as discussed in Section 6.2. The mass properties were that of a full mass Task Order Three lander and a local acceleration of gravity of 375 cm/sec^2 was employed.

A lander, horizontal with respect to the local acceleration of gravity, with a vertical velocity of -610 cm/sec along the gravity vector, and landing on a 10° slope, was considered. The initial orientation of the lander was such that a single footpad impacted the landing surface. For comparison, three short runs were made with these initial conditions and the following soil properties.

SS Dense Soil (Figure 5-10):

Primary soil mechanics routine

Relative soil density = .69

Soil unit weight = 1681 dynes/cm^3

Soil internal friction angle = 39°

SS Loose Soil (Figure 5-10):

Primary soil mechanics routine

Relative soil density = 0

Soil unit weight = 1488 dynes/cm^3

Soil internal friction angle = 29°

Secondary Soil:

Secondary soil mechanics routine

Soil elasticity constant = 144 x 10^7 dynes/cm^3

Coefficient of friction = 0.9

The Secondary Soil is the same soil idealization as that employed during the correlation studies conducted for the third drop of the Task Order Three lander discussed in Section 6.2.2.

	JOINT 33 DISPLACEMENT (CM)		
	X	Y	Z
TASK ORDER THREE LANDER (SECTION 6.2.4)	-9.577×10^{-2}	-1.322×10^{-4}	-4.289×10^{-1}
MODIFIED TASK ORDER THREE LANDER	-4.794×10^{-2}	-1.555×10^{-4}	-7.834×10^{-2}

	RADIAL BEAM BENDING MOMENT ABOUT LOCAL Z AXIS AT "Q" END (CM – DYNE)					
	ELEMENT 46	ELEMENT 47	ELEMENT 48	ELEMENT 49	ELEMENT 50	ELEMENT 51
TASK ORDER THREE LANDER (SECTION 6.2.4)	8.652×10^{9}	8.022×10^{9}	-3.648×10^{9}	-8.028×10^{9}	5.625×10^{5}	-3.108×10^{6}
MODIFIED TASK ORDER THREE LANDER	1.863×10^{9}	1.573×10^{9}	-1.869×10^{9}	-1.611×10^{9}	1.818×10^{7}	3.716×10^{6}

FIGURE 6–28 COMPARISON OF COMPUTER RESULTS FOR ORIGINAL AND MODIFIED TASK ORDER THREE LANDER

Figure 6-29 presents a comparison between the resulting main strut time histories for the three soils considered. As can be seen, the largest main strut load and stroking motion resulted when the Secondary Soil was used. This is reasonable since these soil properties were used to idealize the very rigid landing surface of the Task Order Three lander drop test program. The main strut stroke was much less for the SS Loose Soil than for the other soils.

A second set of initial conditions for the full mass lander were as follows:

Ground slope = 10°

Initial rotation about X_g axis = 20°

Vertical velocity = -610 cm/sec

For this run, the SS Dense Soil employed in the previous case was used.

Figures 6-30 and 6-31 present the accelerations and main strut loads resulting from this spatial landing. The spatial nature of this landing is obvious from these time histories. The main struts in each leg load up sequentially as their respective footpads impact the landing surface. These loads then drop off as the lander's initial kinetic energy is absorbed by the crushable struts.

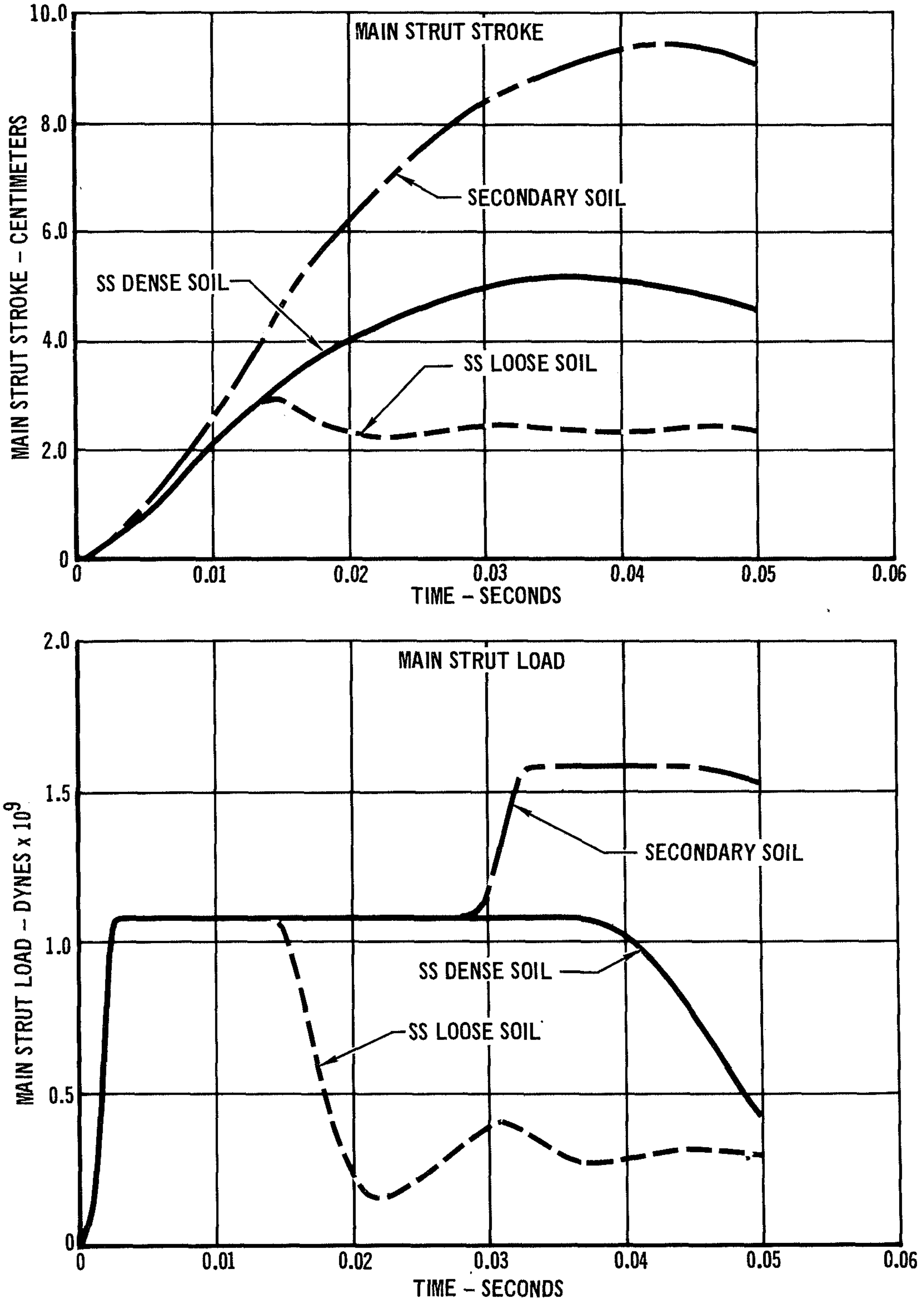


FIGURE 6-29 MAIN STRUT TIME HISTORIES FOR VARIOUS LANDING SURFACE SOIL IDEALIZATIONS

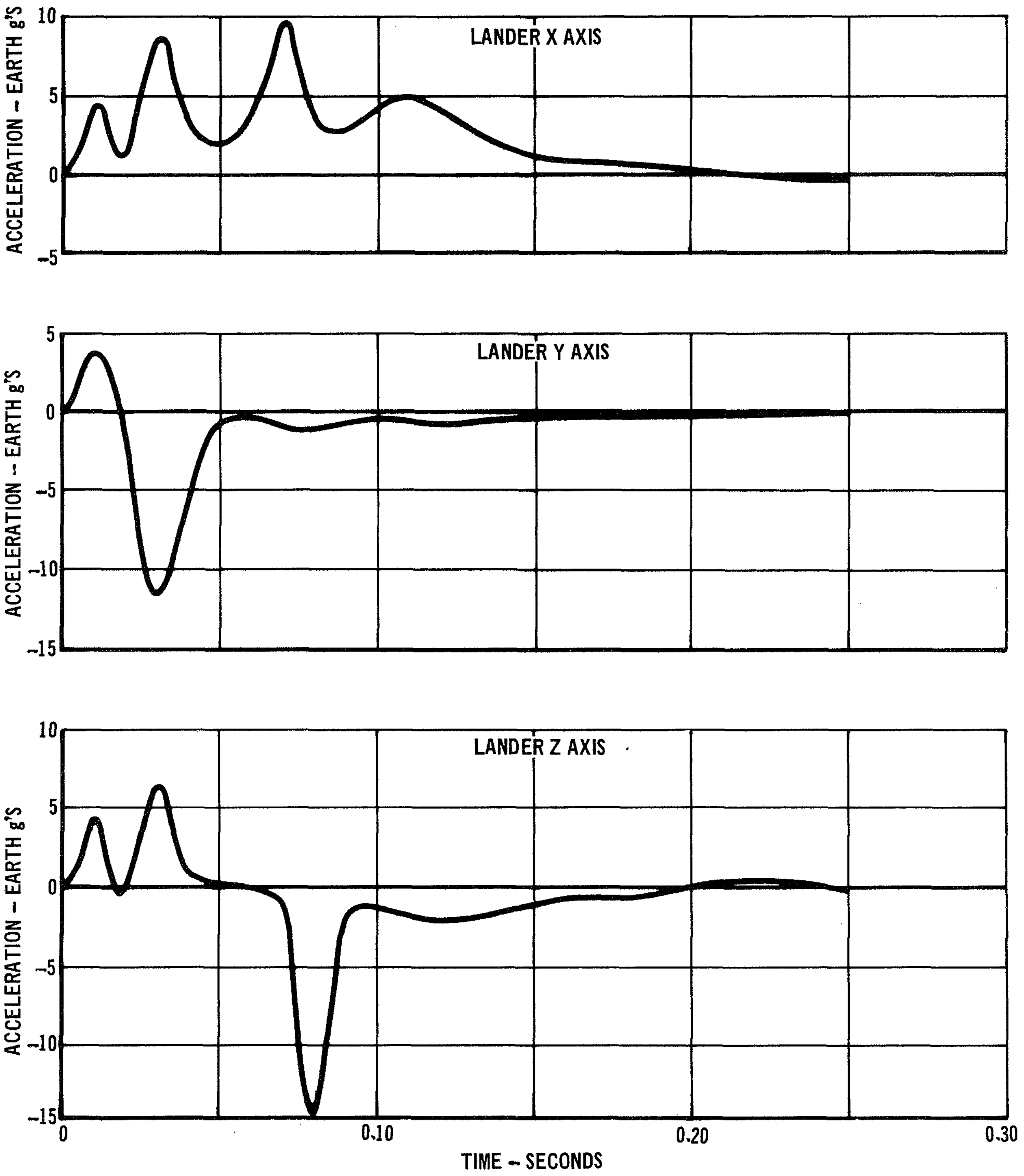


FIGURE 6–30 FULL-MASS TASK ORDER THREE LANDER CENTER OF GRAVITY ACCELERATION TIME HISTORIES

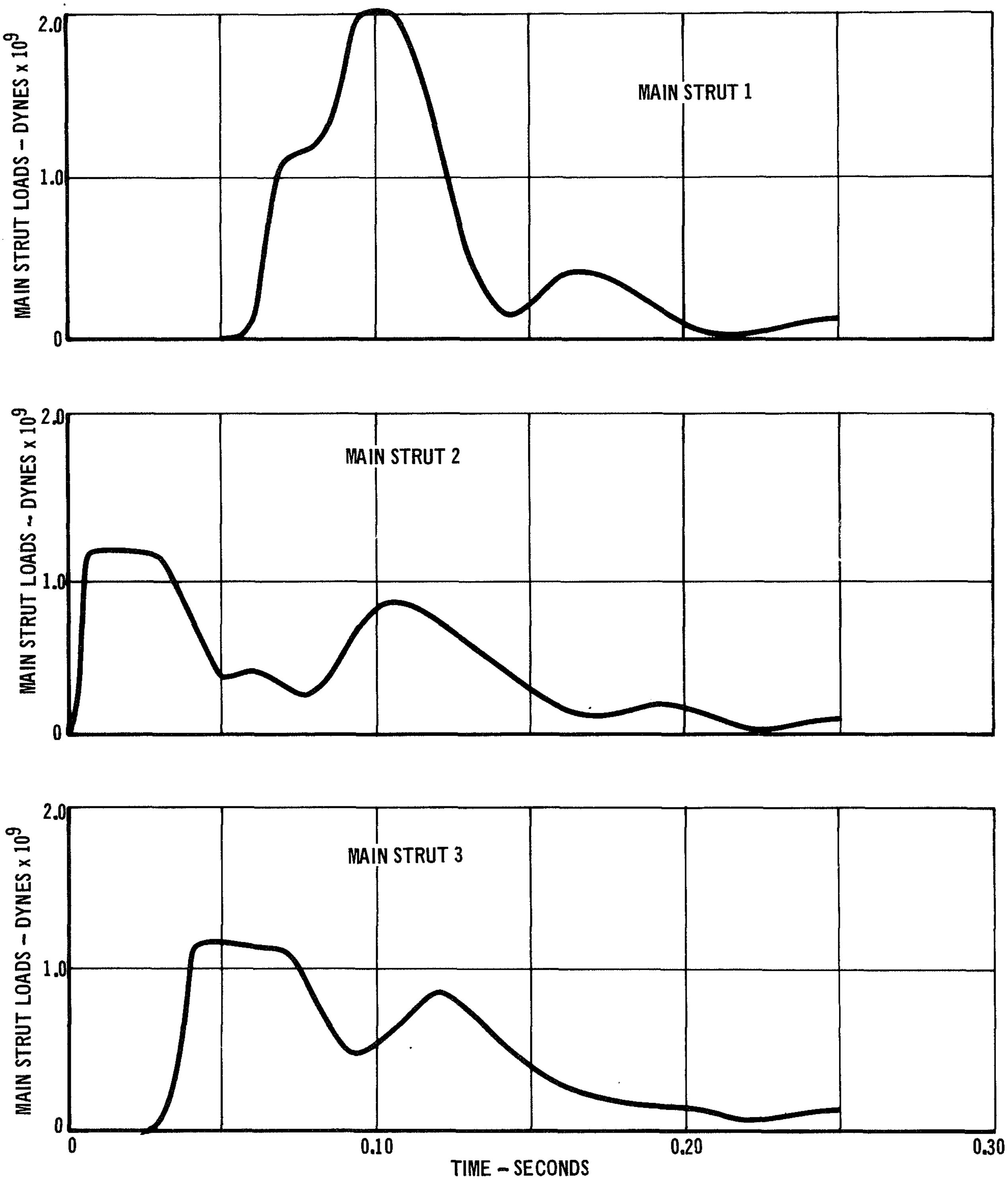


FIGURE 6–31 FULL-MASS TASK ORDER THREE LANDER MAIN STRUT LOAD TIME HISTORIES

7. CONCLUSIONS

Two methods of analysis and associated computer programs were developed for the investigation of legged landers. One program, the Landing Loads and Motions Program, is used to predict landing loads and spatial motions of a lander. The second, the Structural Analysis Program, determines the internal load distributions in the lander center body structure or frequencies and mode shapes for a free-free center body structure. Energy absorption characteristics of an individual landing gear are also determined with this program. Each program contains options and capabilities which were shown to be important in the analysis of legged lander configurations. For example, in the Landing Loads and Motions Program, options to include the effect of center body flexibility and a comprehensive soil mechanics routine were included. An option to investigate large displacement stroking behavior of a landing gear was incorporated in the Structural Analysis Program.

In addition to the limited analyses made within the constraints of Task Order Five, comparisons of experimental and analytical results for ten additional unpublished tests conducted at the NASA Langley Research Center show that the programs are working satisfactorily. However, the effects of friction between the footpads and the landing surface are very difficult to simulate analytically because of the uncertainties involved in defining these effects during dynamic model testing. The limited analyses presented in this report do serve to demonstrate primary program capabilities. As a result of these analyses, the following conclusions are possible:

(1) Center body flexibility significantly affects landing loads and motions. This was illustrated during the landing studies conducted on the Task Order Three lander, Section 6.2.2. For a rigid center body, the drag strut loads (Figure 6-16) were higher than with a flexible center body (Figure 6-20). In addition, the center of gravity accelerations in the rigid center body (Figure 6-15) built up more rapidly than was indicated in either the test data or in the analysis with flexibility included (Figure 6-17). Very good correlation with test data resulted when center body flexibility was included.

(2) Soil properties significantly affect landing gear strut behavior. In Figure 6-29, it was shown that much higher landing gear strut strokes and loads result when landing on hard soil. When landing in loose sand, strut stroke increased gradually with time with resulting lower strut loads. Overall lander motions are highly dependent upon soil properties. Therefore, it is important that programs for studying landing motions, include the capability for properly representing soil characteristics.

(3) Care must be exercised in the finite element idealization of the center body structure. As discussed in Section 6.2, the predicted internal load distributions were different for two different idealizations of the same structure. However, it was also shown that the first idealization presented in Section 6.2 resulted in very good center body flexibility data for use with the Landing Loads and Motions Program. In general, a more sophisticated idealization is required for determination of internal loads than for generation of center body flexibility information for the landing program.

(4) Landing gear orientation and landing surface friction greatly influence the large displacement stroking behavior of a landing gear configuration. Comparison between Figures 6-2 and 6-3 indicate that the drag strut loads in an inverted tripod gear are much different for two different gear orientations. As shown in Figure 6-6, the energy absorption capabilities of a landing gear are dependent on the coefficient of friction employed and the orientation of the gear with respect to the landing surface.

(5) Increasing center body stiffness does not necessarily reduce peak center of gravity accelerations. As indicated in Figure 6-26, the peak predicted accelerations for the original Task Order Three center body and the Modified Task Order Three center body were of the same magnitude. The increased stiffness of the modified structure resulted in a higher frequency oscillation of the Z acceleration. However, in this case, the individual member loads were reduced due to the additional load paths available in the structure.

8. REFERENCES

1. Otto, O. R., Melliere, R. A., and Lopatin, A.: "Design and Fabrication of a Full-Size Landing Impact Test Model of the Mars Legged Lander Configuration," NASA CR-111771, September 1970.
2. Otto, O. R., Dorr, D. J., Laurenson, R. M., Burton, D. J., and Moore, R. L.: "Analyses and Limited Evaluations of Payload and Landing System Structures for the Survivable Soft Landing of Instrument Payloads," NASA CR-66914, April 1970.
3. Serpanos, J. E.: "A User-Coded Matrix Generator for the Displacement Method - Format II Second Version of Fortran Matrix Abstraction Technique," Air Force Flight Dynamics Laboratory Technical Report 66-207, Volume IV, March 1967.
4. Varga, R. S.: Matrix Iterative Analysis, Prentice Hall, Inc., Englewood Cliffs, N. J., 1962.
5. Hestenes, M. and Stiefel, E.: "Method of Conjungate Gradients for Solving Linear Systems," Report 1659, Nation Bureau of Standards, 1952.
6. Hurty, Walter C. and Robinstein, Moshe F.: Dynamics of Structures, Prentice-Hall, INC., Englewood Cliffs, N. J., 1964, pp 43-45.
7. Young, Ronald and McCallum, Helen: "The Determination of Frequencies and Modes of Undamped Structures Using Finite-Element Stiffness and Consistent-Mass Matrices," Aeroelastic and Structures Research Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1965.
8. Martin, H. C.: "On the Derivation of Stiffness Matrices for the Analysis of Large Deflection and Stability Problems," Proceedings of the Conference on Matrix Methods in Structural Mechanics, TR-66-80, 1966, Wright-Patterson AFB, Ohio.
9. Thomson, William T.: Vibration Theory and Applications, Prentice-Hall Inc., Englewood Cliffs, N. J., 1965, pp 294-299.
10. Bisplinghoff, Raymond L., Ashley, Holt, and Lafman, Robert L.: Aeroelasticity, Addison-Wesley Publishing Company, Inc., Cambridge, Massachusetts, 1955, pp 632-635.

11. "Summary Report - Lunar Module (LM) Soil Mechanics Study," Report #AM-68-4, Bendix Energy Controls Division, South Bend, Indiana, November 1968.
12. Anderson, R. G.: "Application of LM Soil-Footpad Interaction Model to Surveyor Landing Dynamics Simulation," Paper 68K11, Bendix Energy Controls Division, South Bend, Indiana, November 1968.

APPENDIX A

EXAMPLE OF INPUT AND OUTPUT

CENTER BODY OPTION

STRUCTURAL ANALYSIS PROGRAM

```
                              STRUCTURAL ANALYSIS PROGRAM --- LEGGED LANDER
                   MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
                              MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EAST

STRUCTURAL ANALYSIS DATA - CARD CODE

                         BLANK - 0        COMMENTS
                                 1        NODAL POINT DEFINITIONS
                                 2        REFERENCE POINTS
                                 3        NODAL POINT RESTRAINT DEFINITIONS
                                 4        FORCE VECTORS
                                 5        MOMENT VECTORS
                                 6        BAR DEFINITIONS
                                 7        SHEAR PANEL DEFINITIONS
                                 8        FORMATED-DATA TERMINATOR

     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         JOINT INFORMATION CARDS
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1  1  13.5890   31.7246   85.7758                      1
1  2  -28.0670  31.7246   85.7758                      2
1  3  13.5890   0.        85.7758                      3
1  4  -28.0670  0.        85.7758                      4
1  5  29.4834   0.        113.538                      5
1  6  13.5890   -31.7246  85.7758                      6
1  7  -28.0670  -31.7246  85.7758                      7
1  8  13.5890   -60.9346  35.1790                      8
1  9  -17.9070  -60.9346  35.1790                      9
1 10  -28.0670  -60.9346  35.1790                      10
1 11  13.5890   -90.1446  -15.4179                     11
1 12  -28.0670  -90.1446  -15.4178 -1 -1 -1
1 13  13.5890   -74.295   -42.8752                     12
1 14  -28.0670  -74.295   -42.8752                     13
1 15  29.4834   -98.3234  -56.7690 -1 -1 -1
1 16  13.5890   -58.42    -70.358                      14
1 17  -28.0670  -58.42    -70.358  -1 -1 -1
1 18  13.5890   0.        -70.358                      15
1 19  -17.9070  0.        -70.358                      16
1 20  -28.067   0.        -70.358                      17
1 21  13.589    58.42     -70.358                      18
1 22  -28.067   58.42     -70.358  -1 -1 -1
1 23  13.589    74.295    -42.8752                     19
1 24  -28.067   74.295    -42.8752                     20
1 25  29.4834   98.3234   -56.769  -1 -1 -1
1 26  13.589    90.1446   -15.4179                     21
1 27  -28.067   90.1446   -15.4178 -1 -1 -1
1 28  13.589    60.9346   35.179                       22
1 29  -17.9070  60.9346   35.179                       23
1 30  -28.067   60.9346   35.179                       24
1 31  13.589    0.        0.                           25
1 32  -17.907   0.        0.                           26
1 33  0.        0.        0.                           27   1
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         FORCE AND MOMENT LIMIT CARDS
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
3  1                      0.
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPECIFIED FORCE VECTOR CARDS
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
4  1 1.801E+07-7.227E+03 4.612E+07
4  2-1.093E+08-1.273E+08 3.736E+08
4  3 1.737E+07 2.084E+05 5.559E+07
4  4 1.139E+07-1.549E+05 4.403E+07
```

```
4    5 4.910E+06 2.318E+05 5.892E+07
4    6 1.873E+07 4.510E+05 4.572E+07
4    7-1.086E+08 1.265E+09 3.742E+08
4    8-2.922E+06 7.793E+07-2.389E+06
4    9-1.875E+06 3.738E+07 7.265E+05
4   10-1.838E+06 3.220E+07 1.003E+07
4   11-2.505E+07 3.670E+07 2.372E+07
4   12-5.850E+07 2.191E+07 1.500E+07
4   13-3.531E+07 1.743E+05 3.324E+07
4   14-1.000E+08-1.351E+05 1.254E+06
4   15-1.017E+08-1.153E+05-1.479E+08
4   16-6.288E+07 3.229E+04-7.277E+07
4   17-4.657E+07 4.866E+04-4.652E+07
4   18-9.941E+07 1.123E+06 1.633E+05
4   19-5.825E+07-2.199E+07 1.528E+07
4   20-3.516E+07-1.716E+05 3.307E+07
4   21-2.514E+07-3.660E+07 2.393E+07
4   22-3.313E+06-7.698E+07-2.225E+06
4   23-2.086E+06-3.740E+07 7.221E+06
4   24-2.033E+06-3.224E+07 9.952E+06
4   25-7.454E+08 6.454E+05-1.370E+09
4   26-3.276E+08-3.646E+05-1.040E+09
4   27-3.560E+07 3.004E+04-1.290E+08
        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            SPECIFIED MOMENT VECTOR CARDS
        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
5    1-9.185E+05 2.957E+03-2.753E+06
        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            BAR INFORMATION CARDS
        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
6    1  1  2  3      11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
6    2  1  3 28      16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
6    3  2  4 30      24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
6    4  3  4  1      14.1123    110.95    4.87      6.4557 6.826E11 2.620E11
6    5  3  5  1      7.6613     55.53     2.22      1.0281 6.826E11 2.620E11
6    6  4  5  2      4.8387     6.98      3.68      .62431 6.826E11 2.620E11
6    7  3  5 23      16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
6    8  4  7 30      24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
6    9  6  7  3      11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
6   10  6  8  3      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   11  7 10  4      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   12  8  9 11      20.729     255.28    109.46    2.7804 6.826E11 2.620E11
6   13  9 10 12      16.3742    253.57    27.46     2.1977 6.826E11 2.620E11
6   14  8 11 13      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   15 10 12 14      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   16 11 12 13      11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
6   17 11 13  8      16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
6   18 12 14 10      24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
6   19 13 14 11      14.1123    110.95    4.87      6.4557 6.826E11 2.620E11
6   20 13 15 11      7.6613     55.53     2.22      1.0281 6.826E11 2.620E11
6   21 14 15 12      4.8387     6.98      3.68      .6243  6.826E11 2.620E11
6   22 13 16 18      16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
6   23 14 17 20      24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
6   24 16 17 14      11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
6   25 16 18 13      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   26 17 20 14      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   27 18 19 16      20.7290    255.28    109.46    2.7804 6.826E11 2.620E11
6   28 19 20 17      16.3742    253.57    27.46     2.1977 6.826E11 2.620E11
6   29 18 21 23      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   30 20 22 24      9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
6   31 21 22 24      11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
6   32 21 23 18      16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
6   33 22 24 20      24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
6   34 23 24 22      14.1123    110.95    4.87      6.4557 6.826E11 2.620E11
6   35 23 25 21      7.6613     55.53     2.22      1.0281 6.826E11 2.620E11
```

```
6  36 24 25 22    4.8387    6.98     3.68     .6243  6.826E11 2.620E11
6  37 23 26 28    16.1290   111.43   274.90   2.1644 6.826E11 2.620E11
6  38 24 27 30    24.8322   435.21   700.43   3.3299 6.826E11 2.620E11
6  39 26 27 24    11.7742   106.70   168.07   1.5775 6.826E11 2.620E11
6  40 26 28 23    9.2774    29.58    4010.2   1.2445 6.826E11 2.620E11
6  41 27 30 24    9.2774    29.58    4010.2   1.2445 6.826E11 2.620E11
6  42 28 29  1    20.7290   255.28   109.46   2.7804 6.826E11 2.620E11
6  43 29 30  2    16.3742   253.57   27.46    2.1977 6.826E11 2.620E11
6  44  1 28  3    9.2774    29.58    4010.2   1.2445 6.826E11 2.620E11
6  45  2 30  4    9.2774    29.58    4010.2   1.2445 6.826E11 2.620E11
6  46 28 31  8  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  47 29 32  9  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  48  8 31 18  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  49  9 32 19  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  50 18 31  8  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  51 19 32  9  1 8.0645    47.28    2004.2   1.0822 6.826E11 2.620E11
6  52 31 33  3    58.7741   2776.3   2776.3   5551.7 6.826E11 2.620E11
6  53 32 33  4    58.7741   2776.3   2776.3   5551.7 6.826E11 2.620E11
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   ANALYSIS OF TASK ORDER THREE LANDER CENTERBODY STRUCTURE
               --------- UNITS ---------
                     FORCE - DYNES
                     LENGTH - CM
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       DATA TERMINATOR CARD
   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
8
```

```
STRUCTURAL ANALYSIS CONTROL DATA

GENERAL DATA

INDRKT =     0       , 1 IMPLIES READ ALL MATRICES FROM TAPE
INDSFG =     1       , 0 IMPLIES WRITE GLOBAL BAR MATRICES
INDSFL =     1       , 0 IMPLIES WRITE LOCAL BAR MATRICES
INDWKT =     0       , 1 IMPLIES SAVE ALL MATRICES ON TAPE
INDWTS =     0       , 1 IMPLIES PRINT TOTAL STIFFNESS MATRIX
ISFDIM =  12904      , MAX. STORAGE FOR STIFFNESS MATRIX

DISPLACEMENT/ROTATION SOLUTION DATA

ERRTOL = 1.000E-04   , ITERATION SOLUTION TOLERANCE
IND1SL =     0       , 1 IMPLIES AN INITIAL SOLUTION IN SOLVEC
INDITR =   2000      , MAX. SOLUTION ITERATION CYCLES
INDPLS =     0       , 1 IMPLIES CONSIDER PLASTICITY
INDRLX =     2       , ITERATIVE SOLUTION METHOD
INDWKT =     0       , 1 IMPLIES SAVE SOLVEC ON TAPE
MINRST =     6       , MIN. ALLOWABLE RESTRAINTS
RELAXF =   1.00      , RELAXATION FACTOR
TMAX   =   100.000   , ITERATION CP TERMINATION TIME
```

```
                               NODAL POINT DISPLACEMENTS AND ROTATIONS
                                      GLOBAL COORDINATE SYSTEM

 NODAL POINT      DISPLACEMENT    DISPLACEMENT    DISPLACEMENT     ROTATION        ROTATION        ROTATION
  NUMBER,              X               Y               Z               X               Y               Z


      1          -2.594057E-01   -1.284398E-05    9.954863E-02  -2.29041 2E-03  -2.493686E-03  -3.99052[illegible]E-05
      2          -2.596587E-01   -1.103752E-03    6.875395E-04  -7.50859 5E-04  -2.517113E-03  -2.20754[illegible]E-05
      3          -2.594768E-01    2.344819E-04    1.379575E-01   1.36598 0E-05  -3.005784E-03   2.305041E-05
      4          -2.596246E-01   -7.071421E-04    1.293253E-02  -1.33039 3E-05  -2.949074E-03   2.295757E-05
      5          -3.447087E-01    2.594447E-04    1.874016E-01   6.55063 2E-06  -3.100999E-03   1.417392E-05
      6          -2.579515E-01    4.806842E-04    9.875488E-02   2.30981 1E-03  -2.453771E-03   8.500476E-05
      7          -2.582035E-01   -3.037241E-04    1.504430E-03   7.26947 4E-04  -2.476936E-03   6.763974E-05
      8          -1.300134E-01    2.133496E-01   -2.929892E-02   1.95377 5E-03  -2.134581E-03   7.201446E-04
      9          -1.300064E-01    1.997026E-01   -9.144344E-02   4.56293 5E-04  -1.929153E-03   6.76428[illegible]E-04
     10          -1.299991E-01    1.848225E-01   -1.073696E-01  -1.54699 9E-04  -1.446096E-03   1.831387E-03
     11           1.195158E-03    1.044167E-01    3.429583E-02  -3.89797 1E-03  -1.287170E-03   2.293985E-03
     12           0.              0.              0.            -5.67498 0E-06  -1.318790E-03   2.132501E-03
     13           4.343227E-03    2.070061E-02   -1.512224E-02  -1.52443 4E-03   6.796597E-05   6.018718E-05
     14           3.046273E-03    5.448616E-03    2.885772E-03   1.93654 0E-04  -1.106379E-04   1.856412E-04
     15           0.              0.              0.            -6.92768 7E-04   4.233830E-04  -1.661343E-04
     16          -2.075337E-03   -6.087500E-03   -3.239054E-02  -1.84866 1E-03   5.487169E-04  -2.619314E-05
     17           0.              0.              0.            -7.72363 2E-04   2.783482E-04   1.108644E-05
     18          -2.139731E-03    1.646172E-05   -4.034778E-01  -8.23254 7E-09  -1.378152E-03  -1.181616E-08
     19          -2.121140E-03    4.047092E-06   -4.402838E-01  -2.11749 7E-08  -1.246994E-03   3.966774E-07
     20          -1.739390E-03    1.752866E-06   -4.351520E-01  -1.87034 6E-08   1.379476E-03   1.526218E-08
     21          -2.072126E-03    6.124341E-03   -3.239479E-02   1.84854 6E-03   5.493697E-04   2.617136E-05
     22           0.              0.              0.             7.72427 6E-04   2.790772E-04  -1.135609E-05
     23           4.365225E-03   -2.072861E-02   -1.508775E-02   1.53188 2E-03   6.648816E-05  -6.310193E-05
     24           3.161616E-03   -5.449777E-03    2.886710E-03  -1.93665 6E-04  -1.126262E-04  -1.901363E-04
     25           0.              0.              0.             6.86955 0E-04   4.229063E-04   1.697941E-04
     26           1.188954E-03   -1.048444E-01    3.456437E-02   3.91476 4E-03  -1.295317E-03  -2.304838E-03
     27           0.              0.              0.             5.67409 6E-06  -1.326562E-03  -2.204177E-03
     28          -1.307529E-01   -2.196456E-01   -2.894178E-02  -1.97583 1E-03  -2.150572E-03  -7.172035E-04
     29          -1.307460E-01   -2.001335E-01   -9.156317E-02  -4.43762 1E-04  -1.944082E-03  -6.726332E-04
     30          -1.307380E-01   -1.853135E-01   -1.076610E-01   1.81348 9E-04  -1.464383E-03  -1.884615E-03
     31          -9.576417E-02   -4.788775E-05   -4.131009E-01   1.32338 4E-06  -1.174504E-03   6.175342E-06
     32          -9.575433E-02   -2.432919E-04   -4.499126E-01   1.25665 3E-06  -1.204229E-03   6.149842E-06
     33          -9.576677E-02   -1.322386E-04   -4.288944E-01   1.28962 6E-06  -1.154116E-03   6.222322E-06
1068 ITERATIONS WERE REQUIRED TO REACH A MAXIMUM RELATIVE DIFFERENCE OF  9.6539364E-05
```

```
                                NODAL POINT FORCES AND MOMENTS
                                   GLOBAL COORDINATE SYSTEM

 NODAL POINT         FORCE          FORCE          FORCE        MOMENT            MOMENT            MOMENT
   NUMBER              X              Y              Z             X                 Y                 Z


      1         1.301000E+07  -7.227000E+03   4.612000E+07   0.                0.                0.
      2        -1. 98000E+08  -1.273000E+08   3.735000E+08   0.                0.                0.
      3         1.737000E+07   2.084000E+05   5.559000E+07   0.                0.                0.
      4         1.139000E+07  -1.549000E+05   4.403000E+07   0.                0.                0.
      5         4.910000E+06   2.318000E+05   5.892000E+07   0.                0.                0.
      6         1.373000E+07   4.510000E+05   4.572000E+07   0.                0.                0.
      7        -1.086000E+08   1.265000E+08   3.742000E+08   0.                0.                0.
      8        -2.322000E+06   7.703000E+07  -2.389000E+06   0.                0.                0.
      9        -1.875000E+06   3.738000E+07   7.265000E+06   0.                0.                0.
     10        -1.538000E+06   3.220000E+07   1.003000E+07   0.                0.                0.
     11        -2.505000E+07   3.670000E+07   2.372000E+07   0.                0.                0.
     12        -5.734348E+08  -6.866081E+08   3.508397E+08   0.                0.                0.
     13        -5.350000E+07   2.191000E+07   1.500000E+07   0.                0.                0.
     14        -3.531000E+07   1.743000E+06   3.324000E+07   0.                0.                0.
     15         5.494562E+08  -8.074434E+08  -5.364948E+08   0.                0.                0.
     16        -1.000000E+08  -1.351000E+06   1.254000E+06   0.                0.                0.
     17         7.321028E+08   3.767355E+08   1.001838E+09   0.                0.                0.
     18        -1.117000E+08  -1.153000E+05  -1.479000E+08   0.                0.                0.
     19        -6.288000E+07   3.229000E+04  -7.277000E+07   0.                0.                0.
     20        -4.557000E+07   4.866000E+04  -4.652000E+07   0.                0.                0.
     21        -9.341000E+07   1.123000E+06   1.633000E+06   0.                0.                0.
     22         7.308529E+08  -3.776764E+08   1.001466E+09   0.                0.                0.
     23        -5.825000E+07  -2.199000E+07   1.528000E+07   0.                0.                0.
     24        -3.516000E+07  -1.716000E+06   3.307000E+07   0.                0.                0.
     25         5.508640E+08   8.099014E+08  -5.387859E+08   0.                0.                0.
     26        -2.514000E+07  -3.660000E+07   2.393000E+07   0.                0.                0.
     27        -5.712072E+08   6.851741E+08   3.521723E+08   0.                0.                0.
     28        -3.313000E+16  -7.698000E+07  -2.225000E+06   0.                0.                0.
     29        -2.086000E+06  -3.740000E+07   7.221000E+06   0.                0.                0.
     30        -2.133000E+06  -3.224000E+07   9.952000E+06   0.                0.                0.
     31        -3.454000E+08   6.454000E+05  -1.370000E+09   0.                0.                0.
     32        -3.276000E+08  -3.646000E+05  -1.040000E+09   0.                0.                0.
     33        -3.560000E+07   3.004000E+04  -1.290000E+08  -9.188000E+05   2.957000E+09  -2.758000E+06

   TOTAL        6.405044E+03   9.765102E+04   6.298171E+03

 RELATIVE SOLUTION ERRORS IN PERCENT
                     .00023         .00330         .00021
```

```
                                     BAR FORCES AND MOMENTS
                                    LOCAL COORDINATE SYSTEMS

 BAR     NODAL POINT NUMBERS             FORCE          FORCE          FORCE          MOMENT         MOMENT         MOMENT
NUMBER      P     Q     R ,                X              Y              Z              X              Y              Z

   1        1     2     3   POINT P  -4.382243E+07  -2.879997E+07  -1.048451E+08   1.527523E+07   2.119184E+09  -6.31[illegible]203E+08
                            POINT Q   4.882243E+07   2.879997E+07   1.048451E+08  -1.527523E+07   2.24824[illegible]E+09  -5.686714E+08

   2        1     3    28   POINT P   8.583158E+07   6.558738E+07  -1.384407E+07  -9.153672E+06   5.919729E+08  -4.4[illegible]3820E+09
                            POINT Q  -8.583158E+07  -6.558738E+07   1.384407E+07   9.153672E+06  -1.[illegible]2775[illegible]E+08   6.5[illegible]4553E+09

   3        2     4    30   POINT P   2.119086E+08   1.379957E+07  -3.625722E+06  -1.187904E+07   7.361937E+08  -6.687698E+09
                            POINT Q  -2.119086E+08  -1.379957E+07   3.625722E+06   1.187904E+07  -6.211691E+08   7.12[illegible]8[illegible]E+09

   4        3     4     1   POINT P  -3.418606E+07  -2.090930E+05  -5.503318E+05  -1.094831E+06   6.936732E+06  -4.523774E+06
                            POINT Q   3.418606E+07   2.090930E+05   5.503318E+05   1.094831E+06   1.598789E+07  -4.186206E+05

   5        3     5     1   POINT P  -8.918537E+07  -1.328559E+05  -4.698598E+05   9.459827E+04   1.202607E+07  -4.207454E+06
                            POINT Q   8.918537E+07   1.328559E+05   4.698598E+05  -9.459827E+04   3.[illegible]06229E+06  -4.3[illegible]2786E+04

   6        4     5     2   POINT P   4.326431E+07   4.288699E+04  -9.266836E+04  -3.600683E+04   8.932963E+06   2.6[illegible]3297E+06
                            POINT Q  -4.326431E+07  -4.288699E+04   9.266836E+04   3.600683E+04  -3.011260E+06   1.372715E+05

   7        3     6    28   POINT P   8.544166E+07  -6.709352E+07   1.331067E+07   9.867145E+06   1.[illegible]53138E+08  -6.569454E+09
                            POINT Q  -8.544166E+07   6.709352E+07  -1.331067E+07  -9.867145E+06  -5.775895E+08   4.44[illegible]939E+09

   8        4     7    30   POINT P   2.123405E+08  -1.206587E+07   2.881598E+06   1.298391E+07   6.276850E+08  -7.123227E+09
                            POINT Q  -2.123405E+08   1.206587E+07  -2.881598E+06  -1.298391E+07  -7.191026E+08   6.7[illegible]442E+09

   9        6     7     3   POINT P  -4.862856E+07  -2.888498E+07   1.037496E+08  -1.570495E+07  -2.097087E+09  -6.319781E+08
                            POINT Q   4.862856E+07   2.888498E+07  -1.037496E+08   1.570495E+07  -2.224705E+09  -5.712544E+08

  10        6     8     3   POINT P  -1.594680E+08  -1.567732E+08  -1.662653E+07   3.960516E+06   2.412396E+09  -4.456542E+09
                            POINT Q   1.594680E+08   1.567732E+08   1.662653E+07  -3.960516E+06  -1.441023E+09  -4.702638E+09

  11        7    10     4   POINT P  -1.873301E+08  -2.406391E+08  -5.706150E+07   1.169124E+07   2.560600E+09  -6.724742E+09
                            POINT Q   1.873301E+08   2.406391E+08   5.706150E+07  -1.169124E+07   7.731100E+08  -7.334145E+09

  12        8     9    11   POINT P   3.133247E+06   1.858049E+08   3.175125E+07  -3.463495E+07  -3.461788E+08   4.031278E+09
                            POINT Q  -3.133247E+06  -1.858049E+08  -3.175125E+07   3.463495E+07  -6.538586E+08   1.820832E+09

  13        9    10    12   POINT P   7.997026E+06   2.749702E+08  -4.717858E+08  -3.462681E+07   4.783456E+09  -1.824670E+09
                            POINT Q  -7.997026E+06  -2.749702E+08   4.717858E+08   3.462681E+07   9.880049E+06   4.618367E+09

  14        8    11    13   POINT P  -2.588178E+08   9.056839E+07  -1.025100E+08   9.971527E+06   5.477076E+09   4.666035E+09
                            POINT Q   2.588178E+08  -9.056839E+07   1.025100E+08  -9.971527E+06   5.118799E+08   6.232525E+09

  15       10    12    14   POINT P   6.288825E+07   2.540180E+08  -6.700072E+07   1.810623E+06   3.845303E+09   7.368758E+09
                            POINT Q  -6.288825E+07  -2.540180E+08   6.700072E+07  -1.810623E+06   6.908818E+07   7.471765E+09

  16       11    12    13   POINT P  -2.305926E+08   2.755679E+08   9.268548E+06   3.861881E+07  -3.916082E+08   5.602939E+09
                            POINT Q   2.305926E+08  -2.755679E+08  -9.268548E+06  -3.861881E+07   5.517508E+06   5.876119E+09

  17       11    13     8   POINT P  -3.288813E+08   3.223619E+08  -3.580061E+08  -4.672166E+07   5.338310E+09  -5.844829E+08
                            POINT Q   3.288813E+08  -3.223619E+08   3.580061E+08   4.672166E+07   6.011774E+09   1.080452E+10
```

```
                                            BAR FORCES AND MOMENTS
                                           LOCAL COORDINATE SYSTEMS

 BAR     NODAL POINT NUMBERS                FORCE          FORCE          FORCE          MOMENT         MOMENT         MOMENT
NUMBER      P      Q      R ,                 X              Y              Z              X              Y              Z


  18       12     14     10    POINT P   -1.201164E+08  -3.559466E+08  -4.098429E+08  -6.442613E+07   5.840005E+09  -7.510180E+09
                               POINT Q    1.201164E+08   3.559466E+08   4.098429E+08   6.442613E+07   7.153491E+09  -3.774610E+09

  19       13     14     11    POINT P   -2.999368E+08  -2.693242E+08  -3.809306E+05   6.976098E+07  -7.931317E+06  -5.443194E+09
                               POINT Q    2.999368E+08   2.693242E+08   3.809306E+05  -6.976098E+07   2.379936E+07  -5.775776E+09

  20       13     15     11    POINT P   -1.115395E+09   6.056762E+07  -1.950897E+06  -2.060756E+06   4.889821E+07   1.938506E+09
                               POINT Q    1.115395E+09  -6.056762E+07   1.950897E+06   2.060756E+06   1.350683E+07  -1.076132E+06

  21       14     15     12    POINT P    3.491764E+06  -3.041003E+05  -2.818253E+05   2.361462E+06   3.152152E+07  -1.920829E+07
                               POINT Q   -3.491764E+06   3.041003E+05   2.818253E+05  -2.361462E+06  -1.351304E+07  -2.235554E+05

  22       13     16     18    POINT P   -5.390495E+08  -6.193668E+08  -1.637794E+08  -5.632838E+06   3.932439E+08  -9.051805E+09
                               POINT Q    5.390495E+08   6.193668E+08   1.637794E+08   5.632838E+06   4.804836E+09  -1.060585E+10

  23       14     17     20    POINT P    1.209494E+08  -3.369852E+08  -1.485482E+08  -9.526940E+06  -1.393976E+09   3.694369E+09
                               POINT Q   -1.209494E+08   3.369852E+08   1.485482E+08   9.526940E+06   6.108642E+09  -1.438971E+10

  24       16     17     14    POINT P    4.004135E+08  -1.238612E+08  -2.397152E+08   1.067887E+07   4.531354E+09  -2.203040E+09
                               POINT Q   -4.004135E+08   1.238612E+08   2.397152E+08  -1.067887E+07   5.454224E+09  -2.956520E+09

  25       16     18     13    POINT P   -6.616721E+08   3.853367E+08   1.367173E+08   1.075441E+07  -5.220281E+09   1.061675E+10
                               POINT Q    6.616721E+08  -3.853367E+08  -1.367173E+08  -1.075441E+07  -2.766746E+09   1.189462E+10

  26       17     20     14    POINT P   -1.900115E+05   5.013956E+08   2.331411E+08  -6.145715E+06  -6.291294E+09   1.437882E+10
                               POINT Q    1.900115E+05  -5.013956E+08  -2.331411E+08   6.145715E+06  -7.328809E+09   1.491270E+10

  27       18     19     16    POINT P    8.352166E+06  -4.252396E+05  -1.301373E+08  -2.993428E+02   2.360546E+09  -8.956695E+06
                               POINT Q   -8.352166E+06   4.252396E+05   1.301373E+08   2.993428E+02   1.738260E+09  -4.436650E+06

  28       19     20     17    POINT P    4.199635E+08  -3.992058E+05  -9.562713E+08   1.400679E+02   9.703436E+09   4.469866E+06
                               POINT Q   -4.199635E+08   3.992058E+05   9.562713E+08  -1.400679E+02   1.228030E+07  -8.525796E+06

  29       18     21     23    POINT P   -6.620967E+08  -3.853362E+08  -1.370277E+08  -1.075806E+07   2.775726E+09  -1.189457E+10
                               POINT Q    6.620967E+08   3.853362E+08   1.370277E+08   1.075806E+07   5.229435E+09  -1.061677E+10

  30       20     22     24    POINT P    1.900115E+05  -5.013946E+08  -2.334340E+08   6.141646E+06   7.337373E+09  -1.491271E+10
                               POINT Q   -1.900115E+05   5.013946E+08   2.334340E+08  -6.141646E+06   6.299841E+09  -1.437876E+10

  31       21     22     24    POINT P    3.397941E+08  -1.233825E+08   2.400862E+08  -1.067710E+07  -4.539310E+09  -2.193142E+09
                               POINT Q   -3.997941E+08   1.233825E+08  -2.400862E+08   1.067710E+07  -5.461722E+09  -2.946481E+09

  32       21     23     18    POINT P   -5.394360E+08   6.205329E+08   1.632994E+08   5.696638E+06  -4.799573E+09   1.060623E+10
                               POINT Q    5.394360E+08  -6.205329E+08  -1.632994E+08  -5.696638E+06  -3.832745E+08   9.088431E+09

  33       22     24     20    POINT P    1.208259E+08   3.369142E+08   1.476248E+08   9.648216E+06  -6.102872E+09   1.438929E+10
                               POINT Q   -1.208259E+08  -3.369142E+08  -1.476248E+08  -9.648216E+06   1.417511E+09  -3.696206E+09

  34       23     24     22    POINT P   -3.014755E+08   2.690715E+08  -4.460332E+05  -7.006383E+07  -6.637976E+06   5.437760E+09
                               POINT Q    3.014755E+08  -2.690715E+08   4.460332E+05   7.006383E+07   2.521794E+07   5.770682E+09
```

```
                                          BAR FORCES AND MOMENTS
                                         LOCAL COORDINATE SYSTEMS

  BAR     NODAL POINT NUMBERS            FORCE          FORCE          FORCE          MOMENT         MOMENT         MOMENT
 NUMBER     P      Q      R ,              X              Y              Z              X              Y              Z

   35      23     25     21   POINT P  -1.119495E+09  -6.132402E+07  -1.976272E+06   2.133977E+06   4.960297E+07  -1.962748E+09
                              POINT Q   1.119495E+09   6.132402E+07   1.976272E+06  -2.133977E+06   1.361378E+07   1.122493E+06

   36      24     25     22   POINT P   4.173026E+06   3.079259E+05  -2.867356E+05  -2.345557E+06   3.195032E+07   1.930631E+07
                              POINT Q  -4.173026E+06  -3.079259E+05   2.867356E+05   2.345557E+06  -1.362806E+07   3.699830E+05

   37      23     26     28   POINT P  -3.298882E+08  -3.246212E+08   3.583704E+08   4.690425E+07  -6.028253E+09  -1.086276E+10
                              POINT Q   3.298882E+08   3.246212E+08  -3.583704E+08  -4.690425E+07  -5.333381E+09   5.710933E+10

   38      24     27     30   POINT P  -1.199927E+08   3.560894E+08   4.101261E+08   6.470065E+07  -7.171846E+09   3.776773E+09
                              POINT Q   1.199927E+08  -3.560894E+08  -4.101261E+08  -6.470065E+07  -5.830631E+09   7.512544E+09

   39      26     27     24   POINT P  -2.293956E+08   2.759769E+08  -1.055208E+07  -3.878545E+07   4.168584E+08   5.612745E+09
                              POINT Q   2.293956E+08  -2.759769E+08   1.055208E+07   3.878545E+07   2.269907E+07   5.883348E+09

   40      26     28     23   POINT P  -2.599978E+08  -9.056280E+07   1.036847E+08  -1.006008E+07  -5.413518E+08  -6.096623E+08
                              POINT Q   2.599978E+08   9.056280E+07  -1.036847E+08   1.006008E+07  -5.516229E+09  -4.681298E+09

   41      27     30     24   POINT P   6.362890E+07  -2.537659E+08   6.831450E+07  -1.929124E+06  -1.020316E+08  -7.473612E+09
                              POINT Q  -6.362890E+07   2.537659E+08  -6.831450E+07   1.929124E+06  -3.989114E+09  -7.352184E+09

   42      28     29      1   POINT P   3.096110E+06  -1.873529E+08   3.232333E+07   3.543492E+07  -3.556837E+08  -4.063108E+09
                              POINT Q  -3.096110E+06   1.873529E+08  -3.232333E+07  -3.543492E+07  -6.623720E+08  -1.837760E+09

   43      29     30      2   POINT P   8.825360E+06  -2.763893E+08  -4.702663E+08   3.542690E+07   4.767884E+09   1.841662E+09
                              POINT Q  -8.825360E+06   2.763893E+08   4.702663E+08  -3.542690E+07   1.002149E+07  -4.649778E+09

   44       1     28      3   POINT P  -1.590604E+08  -1.577479E+08   1.695516E+07  -4.231039E+06  -2.438808E+09  -4.499341E+09
                              POINT Q   1.590604E+08   1.577479E+08  -1.695516E+07   4.231039E+06   1.448235E+09  -4.716782E+09

   45       2     30      4   POINT P  -1.879883E+08  -2.394464E+08   5.733671E+07  -1.193982E+07  -2.589172E+09  -6.672427E+09
                              POINT Q   1.879883E+08   2.394464E+08  -5.733671E+07   1.193982E+07  -7.606170E+08  -7.316778E+09

   46      28     31      8   POINT P  -1.481512E+08   1.229681E+08  -8.646563E+07   4.863881E+06   3.498194E+08   0.
                              POINT Q   1.481512E+08  -1.229681E+08   8.646563E+07  -4.863881E+06   5.733938E+09   8.652084E+09

   47      29     32      9   POINT P  -4.738507E+08   1.140176E+08  -3.751545E+06   3.949656E+06  -4.105504E+09   0.
                              POINT Q   4.738507E+08  -1.140176E+08   3.751545E+06  -3.949656E+06   4.369464E+09   8.022325E+09

   48       8     31     18   POINT P  -1.477460E+08  -1.229060E+08  -8.610530E+07  -4.789116E+06   3.401349E+08   0.
                              POINT Q   1.477460E+08   1.229060E+08   8.610530E+07   4.789116E+06   5.718269E+09  -8.647720E+09

   49       9     32     19   POINT P  -4.747608E+08  -1.141033E+08  -3.121859E+06  -3.880420E+06  -4.129586E+09   0.
                              POINT Q   4.747608E+08   1.141033E+08   3.121859E+06   3.880420E+06   4.349241E+09  -8.028355E+09

   50      18     31      8   POINT P   7.529094E+08   7.995244E+03   1.802733E+08  -2.493369E+04  -2.382027E+09   0.
                              POINT Q  -7.529094E+08  -7.995244E+03  -1.802733E+08   2.493369E+04  -1.030164E+10   5.625294E+05

   51      19     32      9   POINT P   7.533622E+08  -4.417866E+04   3.488801E+08  -2.318474E+04  -1.144171E+10   0.
                              POINT Q  -7.533622E+08   4.417866E+04  -3.488801E+08   2.318474E+04  -1.310480E+10  -3.108322E+06
```

```
                                   BAR FORCES AND MOMENTS
                                  LOCAL COORDINATE SYSTEMS

  BAR     NODAL POINT NUMBERS              FORCE          FORCE          FORCE          MOMENT         MOMENT         MOMENT
 NUMBER    P      Q      R ,                 X              Y              Z              X              Y              Z


   52     31     33      3   POINT P  -7.695605E+06  -2.562079E+08   1.041596E+06  -3.613494E+06  -1.362884E+07  -4.584080E+03
                             POINT Q   7.695605E+06   2.562079E+08  -1.041596E+06   3.613494E+06  -5.254138E+05   1.102470E+03

   53     32     33      4   POINT P   2.786800E+07   3.852027E+08   1.105555E+06  -2.678310E+06  -1.756909E+07   8.752367E+09
                             POINT Q  -2.786800E+07  -3.852027E+08  -1.105555E+06   2.678310E+06  -2.228071E+06  -1.854543E+09
```

```
                                    CPU TIME USAGE TABLE (SEC)

                                                                      TIME IN      TIME OUT       TOTAL
 INPUT AND INITIALIZATION OVERLAY                                        .347         1.363        1.02

 STRUCTURAL STIFFNESS MATRIX GENERATION OVERLAY                         1.363         2.465        1.10

 DISPLACEMENT, ROTATION, FORCE, AND MOMENT SOLUTION OVERLAY             2.465        42.421       39.96
```

APPENDIX B

EXAMPLE OF INPUT AND OUTPUT
MODAL ANALYSIS ROUTINE
STRUCTURAL ANALYSIS PROGRAM

```
                    STRUCTURAL ANALYSIS PROGRAM --- LEGGED LANDER
          MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
                 MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EAST

STRUCTURAL ANALYSIS DATA - CARD CODE

                        BLANK - 0        COMMENTS
                                1        NODAL POINT DEFINITIONS
                                2        REFERENCE POINTS
                                3        NODAL POINT RESTRAINT DEFINITIONS
                                4        FORCE VECTORS
                                5        MOMENT VECTORS
                                6        BAR DEFINITIONS
                                7        SHEAR PANEL DEFINITIONS
                                8        FORMATED-DATA TERMINATOR

 1  1  13.5890   31.7246   85.7758
 1  2  -28.0670  31.7246   85.7758
 1  3  13.5890   0.        85.7758
 1  4  -28.0670  0.        85.7758
 1  5  29.4894   0.        113.538
 1  6  13.5890   -31.7245  85.7758
 1  7  -28.0670  -31.7246  85.7758
 1  8  13.5890   -60.9346  35.1790
 1  9  -17.9070  -60.9346  35.1790
 1 10  -28.0670  -60.9346  35.1790
 1 11  13.5890   -90.1446  -15.4178
 1 12  -28.0670  -90.1446  -15.4178
 1 13  13.5890   -74.2950  -42.8752
 1 14  -28.0670  -74.2950  -42.8752
 1 15  29.4894   -98.3234  -56.7690
 1 16  13.5890   -58.42    -70.358
 1 17  -28.0670  -58.42    -70.358
 1 18  13.5890   0.        -70.358
 1 19  -17.9070  0.        -70.358
 1 20  -28.067   0.        -70.358
 1 21  13.589    58.42     -70.358
 1 22  -28.067   58.42     -70.358
 1 23  13.589    74.295    -42.8752
 1 24  -28.067   74.295    -42.8752
 1 25  29.4894   98.3234   -56.769
 1 26  13.589    90.1446   -15.4178
 1 27  -28.067   90.1446   -15.4178
 1 28  13.589    50.9346   35.179
 1 29  -17.9070  50.9346   35.179
 1 30  -28.067   50.9346   35.179
 1 31  13.589    0.        0.
 1 32  -17.907   0.        0.
 1 33  0.        0.        0.
 6  1  1  2  3    11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
 6  2  1  3 28    16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
 6  3  2  4 30    24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
 6  4  3  4  1    14.1129    110.95    4.87      6.4557 6.826E11 2.620E11
 6  5  3  5  1    7.6613     55.53     2.22      1.0281 6.826E11 2.620E11
 6  6  4  5  2    4.8387     6.98      3.68      .62431 6.826E11 2.620E11
 6  7  3  6 28    16.1290    111.43    274.90    2.1644 6.826E11 2.620E11
 6  8  4  7 30    24.8322    435.21    700.43    3.3299 6.826E11 2.620E11
 6  9  6  7  3    11.7742    106.70    168.10    1.5775 6.826E11 2.620E11
 6 10  6  8  3    9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
 6 11  7 10  4    9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
 6 12  8  9 11    20.729     255.28    109.46    2.7804 6.826F11 2.620E11
 6 13  9 10 12    16.3742    253.57    27.46     2.1977 6.826E11 2.620E11
 6 14  8 11 13    9.2774     29.58     4010.2    1.2445 6.826E11 2.620E11
```

```
6  15 10 12 14     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  16 11 12 13     11.7742   106.70    168.10    1.5775 6.826E11 2.620E11
6  17 11 13  8     16.1290   111.43    274.90    2.1644 6.826E11 2.620E11
6  18 12 14 10     24.8322   435.21    700.43    3.3299 6.826E11 2.620E11
6  19 13 14 11     14.1129   110.95    4.87      6.4557 6.826E11 2.620E11
6  20 13 15 11     7.6613    55.53     2.22      1.0281 6.826E11 2.620E11
6  21 14 15 12     4.8387    6.98      3.68      .6243  6.826E11 2.620E11
6  22 13 16 18     16.1290   111.43    274.90    2.1644 6.826E11 2.620E11
6  23 14 17 20     24.8322   435.21    700.43    3.3299 6.826E11 2.620E11
6  24 16 17 14     11.7742   106.70    168.10    1.5775 6.826E11 2.620E11
6  25 16 18 13     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  26 17 20 14     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  27 18 19 16     20.7290   255.28    109.46    2.7804 6.826E11 2.620E11
6  28 19 20 17     16.3742   253.57    27.46     2.1977 6.826E11 2.620E11
6  29 18 21 23     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  30 20 22 24     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  31 21 22 24     11.7742   106.70    168.10    1.5775 6.826E11 2.620E11
6  32 21 23 18     16.1290   111.43    274.90    2.1644 6.826E11 2.620E11
6  33 22 24 20     24.8322   435.21    700.43    3.3299 6.826E11 2.620E11
6  34 23 24 22     14.1129   110.95    4.87      6.4557 6.826E11 2.620E11
6  35 23 25 21     7.6613    55.53     2.22      1.0281 6.826E11 2.620E11
6  36 24 25 22     4.8387    6.98      3.68      .6243  6.826E11 2.620E11
6  37 23 26 28     16.1290   111.43    274.90    2.1644 6.826E11 2.620E11
6  38 24 27 30     24.8322   435.21    700.43    3.3299 6.826E11 2.620E11
6  39 26 27 24     11.7742   106.70    168.07    1.5775 6.826E11 2.620E11
6  40 26 28 23     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  41 27 30 24     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  42 28 29  1     20.7290   255.28    109.46    2.7804 6.826E11 2.620E11
6  43 29 30  2     16.3742   253.57    27.46     2.1977 6.826E11 2.620E11
6  44  1 28  3     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  45  2 30  4     9.2774    29.58     4010.2    1.2445 6.826E11 2.620E11
6  46 28 31  8   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  47 29 32  9   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  48  8 31 18   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  49  9 32 19   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  50 18 31  8   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  51 19 32  9   1 8.0645    47.28     2004.2    1.0822 6.826E11 2.620E11
6  52 31 33  3     58.7741   2776.3    2776.3    5551.7 6.826E11 2.620E11
6  53 32 33  4     58.7741   2776.3    2776.3    5551.7 6.826E11 2.620E11
8
```

```
STRUCTURAL ANALYSIS CONTROL DATA

GENERAL DATA

INDPKT =     0      , 1 IMPLIES READ ALL MATRICES FROM TAPE
INDSFG =     1      , 0 IMPLIES WRITE GLOBAL BAR MATRICES
INDSFL =     1      , 0 IMPLIES WRITE LOCAL BAR MATRICES
INDWKT =     0      , 1 IMPLIES SAVE ALL MATRICES ON TAPE
INDWTS =     0      , 1 IMPLIES PRINT TOTAL STIFFNESS MATRIX
ISFDIM =12904       , MAX. STORAGE FOR STIFFNESS MATRIX


MODAL ANALYSIS DATA

INDNMA =     1      , RUN NORMAL MODE ANALYSIS
INDWNM =     1      , 1 IMPLIES WRITE MODE DATA ON TAPE
IPEDTO =    99      , ORDER OF REDUCE SYSTEM
NEIGVL =    20      , REQUIRED NON-RIGID BODY MODES

AMASS IS THE DIAGONAL MASS VECTOR OF THE REDUCE SYSTEM
                      3.799E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03  6.099E+03  3.549E+03  3.549E+03  3.549E+03  2.490E+03
                      2.490E+03  2.490E+03  3.483E+03  3.483E+03  3.483E+03  3.789E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03
                      6.099E+03  3.533E+03  3.533E+03  3.533E+03  1.879E+03  1.879E+03  1.879E+03  1.736E+03  1.736E+03  1.736E+03
                      3.789E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03  6.099E+03  3.549E+03  3.549E+03  3.549E+03  2.127E+03
                      2.127E+03  2.127E+03  3.483E+03  3.483E+03  3.483E+03  3.789E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03
                      6.099E+03  3.812E+03  3.812E+03  3.812E+03  2.346E+03  2.346E+03  2.346E+03  1.736E+03  1.736E+03  1.736E+03
                      3.789E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03  6.099E+03  3.549E+03  3.549E+03  3.549E+03  2.127E+03
                      2.127E+03  2.127E+03  3.483E+03  3.483E+03  3.483E+03  3.789E+03  3.789E+03  3.789E+03  6.099E+03  6.099E+03
                      6.099E+03  3.533E+03  3.533E+03  3.533E+03  1.879E+03  1.879E+03  1.879E+03  1.736E+03  1.736E+03  1.736E+03
                      3.487E+04  3.487E+04  3.487E+04  3.270E+04  3.270E+04  3.270E+04  3.578E+03  3.578E+03  3.578E+03
IPWKR CONTAINS THE ROWS TO KEEP IN THE REDUCED SYSTEM
                              1          2          3          7          8          9         13         14         15         19
                             20         21         25         26         27         31         32         33         37         38
                             39         43         44         45         49         50         51         55         56         57
                             61         62         63         67         68         69         73         74         75         79
                             80         81         85         86         87         91         92         93         97         98
                             99        103        104        105        109        110        111        115        116        117
                            121        122        123        127        128        129        133        134        135        139
                            140        141        145        146        147        151        152        153        157        158
                            159        163        164        165        169        170        171        175        176        177
                            181        182        183        187        188        189        193        194        195
```

```
                                   GENERALIZED INERTIA PROPERTIES

VARIABLES                   1                2                3                4                5

   WNX               8.40331315E+04   1.12127882E+05   2.96850987E+04   3.42179713E+04   5.06933976E+04
   WNY               5.43594868E+04   2.66459996E+04   5.37421616E+04   6.16972988E+04   7.68004646E+04
   WNZ               1.97048967E+04   8.55226510E+04   5.41736060E+04   6.14136396E+04   7.68867378E+04
    PX               2.43423465E+01  -3.53679722E-01   4.67697521E+03  -2.57896930E+02  -1.07756323E+02
    PY              -2.42335120E+06  -1.66506291E+02  -1.10836184E+05   1.80101818E+03   4.44512439E+01
    PZ               2.48351470E+06   1.70546160E+02   1.00304156E+05  -1.55043815E+03   2.11936905E+02
    GM  DIAGONAL     8.40487575E+04   1.12148266E+05   6.88004331E+04   7.86644549E+04   1.07190300E+05
```

```
                              NATURAL FREQUENCIES (RADIANS/SEC)

                           1               2               3               4               5

                     2.17351330E+02  2.17377127E+02  3.06055553E+02  3.06529126E+02  3.94729713E+02

                         CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE         1               2               3               4               5

     1        X-TRANS.   -7.64582489E-04 -2.44301281E-03 -3.98575639E-01 -9.87226277E-01 -8.79674797E-01
     1        Y-TRANS.   -3.09653265E-04 -8.04528241E-03  9.66601179E-03  6.06508516E-01  4.87913279E-01
     1        Z-TRANS.    8.73360464E-01 -5.86504763E-01  2.97151277E-01  5.04744526E-01  5.82113821E-01

     2        X-TRANS.   -7.58548871E-04 -2.42282525E-03 -3.99066798E-01 -9.87226277E-01 -8.79674797E-01
     2        Y-TRANS.   -4.30401914E-04 -1.26662378E-02 -9.60952849E-03 -6.37469586E-01 -6.59620596E-01
     2        Z-TRANS.    8.72368550E-01 -5.89046423E-01 -2.95186640E-01 -5.32420925E-01 -7.69105691E-01

     3        X-TRANS.   -2.59211498E-04  3.97825501E-07 -3.86296660E-01  6.39902676E-03  1.94579946E-04
     3        Y-TRANS.    4.31750705E-07 -7.93330387E-03  8.56827112E-03  6.08272506E-01  4.88943089E-01
     3        Z-TRANS.    9.67557094E-01  6.63031595E-05  3.15078585E-01 -5.10705596E-03 -6.56910569E-05

     4        X-TRANS.   -7.77723360E-04  3.62913769E-07 -3.87278978E-01  6.41727494E-03  1.95392954E-04
     4        Y-TRANS.    5.36824826E-07 -1.26567091E-02 -8.99312377E-03 -6.38686131E-01 -6.59620596E-01
     4        Z-TRANS.    8.97323094E-01  6.16871752E-05 -2.92485265E-01  4.75304136E-03  7.00813008E-05

     5        X-TRANS.   -5.36309826E-02 -3.12289330E-06 -8.11640472E-01  1.33029197E-02  2.96205962E-04
     5        Y-TRANS.   -2.44923047E-05  5.50956363E-01  9.79616896E-03  6.96472019E-01  4.83523035E-01
     5        Z-TRANS.    1.00000000E+00  6.84475132E-05  5.62131631E-01 -9.11800487E-03 -1.25474255E-04

     6        X-TRANS.   -7.64617519E-04  2.44374196E-03 -3.70579568E-01  1.00000000E+00  8.79674797E-01
     6        Y-TRANS.    3.10526176E-04 -8.04524047E-03  7.42141454E-03  6.06569343E-01  4.87913279E-01
     6        Z-TRANS.    8.73308045E-01  5.86624459E-01  2.82662083E-01 -5.14111922E-01 -5.82113821E-01

     7        X-TRANS.   -7.58582668E-04  2.42355799E-03 -3.71070727E-01  1.00000000E+00  8.80216802E-01
     7        Y-TRANS.    4.31476741E-04 -1.26661782E-02 -8.35216110E-03 -6.37469586E-01 -6.59620596E-01
     7        Z-TRANS.    8.72316169E-01  5.89166364E-01 -2.79960707E-01  5.41788321E-01  7.69647696E-01

     8        X-TRANS.   -1.59738033E-03  1.40958166E-03  3.22691552E-01  5.74513382E-01 -3.82972629E-03
     8        Y-TRANS.    6.42685210E-01 -2.82964038E-01  2.01424361E-01  6.77007299E-02 -1.31327913E-01
     8        Z-TRANS.    4.99994558E-01  7.46027047E-01  1.64783890E-01 -2.14598540E-01 -2.28888889E-01

     9        X-TRANS.   -1.56067247E-03  1.33559793E-03  3.22937132E-01  5.74939173E-01 -3.82981030E-03
     9        Y-TRANS.    6.42781317E-01 -2.82951430E-01 -1.20112967E-01 -6.30170316E-02  9.75609756E-02
     9        Z-TRANS.    4.99560899E-01  7.45567299E-01 -7.31090373E-02  1.32542579E-01  1.63794038E-01

    10        X-TRANS.   -1.57812693E-03  1.37163774E-03  3.22691552E-01  5.74391727E-01 -3.82710027E-03
    10        Y-TRANS.    6.39983011E-01 -2.76640912E-01 -2.35584479E-01 -1.25729927E-01  1.73008130E-01
    10        Z-TRANS.    5.01605094E-01  7.42150279E-01 -1.43664047E-01  2.56751825E-01  2.93333333E-01

    11        X-TRANS.   -2.95658276E-03  1.51243315E-03  9.99754420E-01  1.20437956E-01 -8.87262873E-01
    11        Y-TRANS.    7.60073416E-01  4.94111533E-01  7.29371316E-01  1.28832117E-01 -7.53929539E-01
    11        Z-TRANS.    4.32949065E-01  3.80239355E-01 -1.46782908E-01 -2.60888078E-01  1.35013550E-01

    12        X-TRANS.   -2.94442421E-03  1.49663707E-03  1.00000000E+00  1.20012165E-01 -8.87804878E-01
```

```
                         CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE         1                2                3                4                5

    12       Y-TRANS.    7.64954128E-01   4.90792256E-01  -7.57367387E-01  -1.42274939E-01   1.00000000E+00
    12       Z-TRANS.    4.29695377E-01   3.00832978E-01   1.63752456E-01   2.76094891E-01  -1.88617886E-01

    13       X-TRANS.    1.10464586E-04  -2.44133329E-05   1.93393910E-01  -3.61861314E-01  -3.51327913E-05
    13       Y-TRANS.    4.21694583E-01   8.28495119E-01   3.79666012E-01  -1.04987835E-01  -2.45203252E-01
    13       Z-TRANS.    2.37541799E-01   4.92840928E-01  -3.49705305E-01  -3.98114355E-01   4.29318639E-01

    14       X-TRANS.    3.70019353E-04   4.93961797E-04   1.93934135E-01  -3.62895377E-01  -2.62113821E-05
    14       Y-TRANS.    3.93843760E-01   7.65952495E-01  -3.83840864E-01   9.38807786E-02   3.32845528E-01
    14       Z-TRANS.    2.15227415E-01   4.59153826E-01   3.79666012E-01   4.07846715E-01  -5.74525745E-01

    15       X-TRANS.    2.69010444E-02   5.38364987E-02   4.00540275E-01  -7.58515815E-01   1.58590786E-03
    15       Y-TRANS.    2.26490270E-01   9.95880024E-01   5.18172838E-01  -2.84124088E-01  -2.41138211E-01
    15       Z-TRANS.    6.07667741E-01   2.68734323E-01  -3.48968566E-01  -5.48965937E-01   4.24769648E-01

    16       X-TRANS.    3.68816742E-03  -5.43841413E-04  -6.15667976E-01  -3.41240876E-01   9.88346883E-01
    16       Y-TRANS.    1.45328090E-03   1.00000000E+00   6.47838900E-03  -2.99635036E-01   2.64878049E-01
    16       Z-TRANS.   -5.48291723E-03   5.91606768E-01  -5.63113949E-01  -5.10766423E-01   7.23035230E-01

    17       X-TRANS.    3.66975199E-03  -5.38010650E-04  -6.15422397E-01  -9.41849148E-01   9.88346883E-01
    17       Y-TRANS.    9.66003308E-04   9.98522579E-01  -5.33398821E-03   3.02980535E-01  -3.35392954E-01
    17       Z-TRANS.   -1.19730161E-02   5.92989649E-01   5.97249509E-01   5.34549878E-01  -9.59891599E-01

    18       X-TRANS.    3.14894165E-03   1.22584663E-07  -6.19597250E-01   9.92700730E-03   1.81192412E-05
    18       Y-TRANS.   -4.45210559E-05   9.98224417E-01  -4.20432220E-03  -2.99939173E-01   2.68726287E-01
    18       Z-TRANS.   -6.13274776E-01  -4.21099122E-05  -1.79543222E-01   2.86678832E-03  -3.95663957E-06

    19       X-TRANS.    3.07528500E-03   1.17732579E-07  -6.20088409E-01   9.93309002E-03   1.81734417E-05
    19       Y-TRANS.   -4.44715860E-05   9.97295519E-01   2.18541257E-03   1.55778589E-01  -1.88563686E-01
    19       Z-TRANS.   -5.13914779E-01  -4.20564779E-05   1.31655206E-01  -2.09975669E-03   4.78102981E-06

    20       X-TRANS.    3.11015996E-03   1.20228086E-07  -6.19597250E-01   9.92700730E-03   1.80487805E-05
    20       Y-TRANS.   -4.44818815E-05   9.97606258E-01   4.25583391E-03   3.03406326E-01  -3.39241192E-01
    20       Z-TRANS.   -6.07007874E-01  -4.15482996E-05   2.58104126E-01  -4.11922141E-03   5.31544715E-06

    21       X-TRANS.    3.68817926E-03   5.44140590E-04  -5.91846758E-01   9.60705596E-01  -9.88346883E-01
    21       Y-TRANS.   -1.54247899E-03   9.99999779E-01  -1.48747544E-02  -2.99330900E-01   2.64878049E-01
    21       Z-TRANS.   -5.43011646E-03  -5.91607730E-01  -5.48379175E-01   5.28588808E-01  -7.23035230E-01

    22       X-TRANS.    3.66976456E-03   5.38316534E-04  -5.91601179E-01   9.61313869E-01  -9.88346883E-01
    22       Y-TRANS.   -1.05505610E-03   9.98522454E-01   1.38285855E-02   3.02676399E-01  -3.35447154E-01
    22       Z-TRANS.   -1.19202667E-02  -5.92990972E-01   5.82023576E-01  -5.53527981E-01   9.59891599E-01

    23       X-TRANS.    1.10512526E-04   2.43401510E-05   2.03487230E-01   3.55413625E-01  -3.69051491E-05
    23       Y-TRANS.   -4.21768507E-01   8.28437206E-01  -3.82367387E-01  -9.27007299E-02  -2.45094851E-01
    23       Z-TRANS.    2.37585808E-01  -4.92808538E-01  -3.38408644E-01   4.09124088E-01  -4.29864499E-01

    24       X-TRANS.    3.70146751E-04  -4.93978346E-04   2.04076621E-01   3.56447689E-01  -4.64281843E-05
    24       Y-TRANS.   -3.93912050E-01   7.65898484E-01   3.86051081E-01   7.14720195E-02   3.32791328E-01
    24       Z-TRANS.    2.15268216E-01  -4.59123979E-01   3.68123772E-01  -4.19829684E-01   5.74525745E-01

    25       X-TRANS.    2.69063978E-02  -5.38327712E-02   4.21660118E-01   7.45133820E-01  -1.75609756E-03
```

CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT NUMBER	ELEMENT REFERENCE	1	2	3	4	5
25	Y-TRANS.	-2.26579147E-01	9.95848202E-01	-5.25785855E-01	-2.67274939E-01	-2.40975610E-01
25	Z-TRANS.	6.07692423E-01	-2.68652123E-01	-3.33497053E-01	5.59854015E-01	-4.24769648E-01
26	X-TRANS.	-2.95637365E-03	-1.51277693E-03	9.96070727E-01	-1.52554745E-01	8.87262873E-01
26	Y-TRANS.	-7.60117493E-01	4.94007192E-01	-7.25687623E-01	1.52311436E-01	-7.53929539E-01
26	Z-TRANS.	4.32975876E-01	-3.00180144E-01	-1.39415521E-01	2.65450122E-01	-1.35013550E-01
27	X-TRANS.	-2.94421518E-03	-1.49698350E-03	9.96316306E-01	-1.52128954E-01	8.87262873E-01
27	Y-TRANS.	-7.54997375E-01	4.90687333E-01	7.52946955E-01	-1.66605839E-01	1.00000000E+00
27	Z-TRANS.	4.29722061E-01	-3.00773716E-01	1.55943026E-01	-2.81204380E-01	1.88672087E-01
28	X-TRANS.	-1.59723890E-03	-1.40936794E-03	3.06483301E-01	-5.84489051E-01	3.93550136E-03
28	Y-TRANS.	-6.42659981E-01	-2.83052187E-01	-1.99459725E-01	7.42092457E-02	-1.31327913E-01
28	Z-TRANS.	5.00061108E-01	-7.45958561F-01	1.70776031E-01	2.09184915E-01	2.28780488E-01
29	X-TRANS.	-1.56059758E-03	-1.33537968E-03	3.06728880E-01	-5.84854015E-01	3.93712737E-03
29	Y-TRANS.	-6.42756056E-01	-2.83039644E-01	1.18295678E-01	-6.69099757E-02	9.75067751E-02
29	Z-TRANS.	4.99627302E-01	-7.45498597E-01	-7.68172888E-02	-1.30048662E-01	-1.63739837E-01
30	X-TRANS.	-1.57804869E-03	-1.37142237E-03	3.06483301E-01	-5.84306569E-01	3.93387534E-03
30	Y-TRANS.	-6.39958309E-01	-2.76728754E-01	2.31974460E-01	-1.33272506E-01	1.72953930E-01
30	Z-TRANS.	5.01671159E-01	-7.42081213E-01	-1.50834971E-01	-2.51946472E-01	-2.93224932E-01
31	X-TRANS.	-2.03274563E-05	1.16369554E-07	3.48722986E-03	-3.94160584E-06	4.75176152E-05
31	Y-TRANS.	3.19899868E-05	-7.17412968E-01	2.52946955E-03	1.80352798E-01	9.40379404E-04
31	Z-TRANS.	-6.16432244E-01	-4.23244294E-05	-1.67387033E-01	2.67153285E-03	-4.68997290E-06
32	X-TRANS.	-2.02422691E-05	1.16366734E-07	3.48722986E-03	-3.93917275E-06	4.75067751E-05
32	Y-TRANS.	3.20053588E-05	-7.18027219E-01	-1.77725933E-03	-1.26703163E-01	3.00379404E-03
32	Z-TRANS.	-6.17769949E-01	-4.23227614E-05	1.18492141E-01	-1.88807786E-03	5.52845528E-06
33	X-TRANS.	-2.02913627E-05	1.16372161E-07	3.48722986E-03	-3.94038929E-06	4.75176152E-05
33	Y-TRANS.	3.19955994E-05	-7.17655508E-01	6.69941061E-04	4.77554745E-02	1.83035230E-03
33	Z-TRANS.	-6.16989651E-01	-4.23223351E-05	-4.39096267E-02	7.01946472E-04	-2.99674797E-07

```
                               GENERALIZED INERTIA PROPERTIES

VARIABLES                6                7                8                9                10

    WMX          4.95876587E+04   6.16583024E+04   5.83085038E+04   1.07689532E+04   5.77606280E+03
    WMY          2.61634448E+04   2.30648816E+04   3.69414879E+04   8.14502536E+03   1.80811712E+03
    WMZ          2.63276932E+04   3.91984612E+04   2.17051152E+04   3.38052438E+03   4.37789279E+03
    PX           1.91219536E+01  -1.60387525E+01   1.83574012E+02  -2.78692195E+02   1.22624238E+01
    PY          -2.56508463E+01   4.46301049E+01  -2.33262960E+06   3.80844061E+05  -9.50505703E+02
    PZ           1.03682271E+01  -2.07531475E+01   2.32055734E+06  -3.80791830E+05   8.95655019E+02
    GM  DIAGONAL 5.10393983E+04   6.20608226E+04   5.84775534E+04   1.11472515E+04   5.98103636E+03
```

```
                                   NATURAL FREQUENCIES (RADIANS/SEC)

                                6                 7                 8                 9                10

                          5.42850223E+02    6.33268051E+02    6.34550229E+02    9.57393853E+02    9.57550540E+02


                                CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE          6                 7                 8                 9                10

     1        X-TRANS.     1.57879377E-01   -7.51736915E-02    3.30368268E-03    8.98575513E-03    7.02301689E-02
     1        Y-TRANS.    -4.95038911E-01    9.96294581E-01    3.18841310E-03   -1.48051912E-03   -1.22433954E-01
     1        Z-TRANS.    -8.68871595E-01    6.68828161E-01    2.09690359E-01   -3.88582737E-02   -7.38414898E-02

     2        X-TRANS.     1.58268482E-01   -7.47105141E-02    3.19358913E-03    9.15589956E-03    7.24122997E-02
     2        Y-TRANS.    -2.95428016E-01    9.09680408E-01    1.92470170E-03   -5.25525315E-04   -4.64703335E-02
     2        Z-TRANS.    -6.28599222E-01    6.10930987E-01    1.88040649E-01   -8.18051403E-02    5.57384149E-03

     3        X-TRANS.    -3.05155642E-05    1.37795276E-05    3.96034898E-03    1.02342698E-02   -3.13469034E-05
     3        Y-TRANS.    -4.95330739E-01    1.00000000E+00    1.76920282E-05   -5.44795660E-04   -1.18406236E-01
     3        Z-TRANS.     1.08949416E-05   -9.12922649E-06    1.82934007E-01   -8.71387603E-02    2.21697705E-04

     4        X-TRANS.    -3.07782101E-05    1.39740621E-05    3.05265527E-03    1.12728980E-02   -3.43915115E-05
     4        Y-TRANS.    -2.95525292E-01    9.10606762E-01    1.45842489E-05   -2.13721030E-04   -4.65569511E-02
     4        Z-TRANS.    -1.10603113E-05    2.35942566E-06    1.68772608E-01   -8.50590502E-02    2.01082720E-04

     5        X-TRANS.    -4.86673152E-05    2.40435387E-05   -1.71021818E-02    2.81262047E-02   -9.17280208E-05
     5        Y-TRANS.     3.88618677E-01    6.37332098E-01    1.16000485E-05    4.60302705E-03    1.00000000E+00
     5        Z-TRANS.     2.17996109E-05   -1.54284391E-05    1.98224654E-01   -1.01472008E-01    2.67345171E-04

     6        X-TRANS.    -1.57879377E-01    7.51736915E-02    3.30847024E-03    8.33878340E-03   -7.03334777E-02
     6        Y-TRANS.    -4.95038911E-01    9.96294581E-01   -3.15315530E-03    3.53888208E-04   -1.22433954E-01
     6        Z-TRANS.     8.68871595E-01   -6.68828161E-01    2.09665747E-01   -3.81748906E-02    7.40147250E-02

     7        X-TRANS.    -1.58365759E-01    7.47568319E-02    3.19839282E-03    8.48953742E-03   -7.24556085E-02
     7        Y-TRANS.    -2.95428016E-01    9.09680408E-01   -1.89555632E-03    9.88672423E-05   -4.64703335E-02
     7        Z-TRANS.     6.28599222E-01   -6.10930987E-01    1.88020893E-01   -8.18536634E-02   -5.18406236E-03

     8        X-TRANS.    -1.49027237E-03    5.62760537E-02    2.95292720E-02    4.52475437E-02   -5.83802512E-02
     8        Y-TRANS.     2.54085603E-01   -2.83001390E-01   -4.35471824E-02   -6.93341614E-03   -1.31485492E-02
     8        Z-TRANS.     4.39883268E-01    4.37702640E-02    2.21073272E-01   -2.55686177E-02   -5.06712863E-03

     9        X-TRANS.    -1.49027237E-03    5.56739231E-02    2.91980828E-02    4.53238128E-02   -5.84663683E-02
     9        Y-TRANS.     2.13424125E-01   -2.98749421E-01   -5.99777329E-02   -4.24327294E-02    4.12906020E-03
     9        Z-TRANS.     3.68190661E-01    6.35479339E-02    2.16770827E-01   -4.39405989E-02   -3.13555652E-02

    10        X-TRANS.    -1.48832685E-03    5.58591941E-02    2.93019615E-02    4.52357118E-02   -5.83363424E-02
    10        Y-TRANS.     2.00000000E-01   -2.49143122E-01   -3.48471742E-02   -5.08391126E-02    7.66565613E-03
    10        Z-TRANS.     3.44455253E-01    3.86521538E-02    1.97230224E-01   -4.92159568E-02   -3.76483326E-02

    11        X-TRANS.     1.55058366E-01    3.70819824E-02    5.57177430E-02    7.78242699E-02   -4.11087051E-02
    11        Y-TRANS.     1.00000000E+00    3.76609541E-01    1.00000000E+00    1.30915397E-01   -8.51017757E-02
    11        Z-TRANS.     6.25680934E-03   -3.60676239E-01   -3.97229132E-01   -9.06719100E-02    2.30142919E-02

    12        X-TRANS.     1.55544747E-01    3.67531264E-02    5.53835539E-02    8.02498248E-02   -4.22367042E-02
```

```
                          CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

 NODAL POINT   ELEMENT
   NUMBER     REFERENCE          6               7               8               9               10

     12        Y-TRANS.    5.94455253E-01  3.55859134E-01  9.20564106E-01 -1.19203609E-02 -5.49155479E-02
     12        Z-TRANS.    5.72470817E-02 -3.29087541E-01 -3.66027726E-01 -7.17139134E-02 -4.08055435E-03

     13        X-TRANS.   -1.41245136E-04  4.12876332E-03 -1.68176283E-03 -4.72652042E-03 -6.84278909E-03
     13        Y-TRANS.    2.53599222E-01  1.11069940E-01  5.02326428E-01  3.06201260E-02 -7.78258987E-02
     13        Z-TRANS.   -4.24708171E-01 -5.19221862E-01 -6.86190634E-01 -1.43645745E-01  2.41792984E-02

     14        X-TRANS.   -1.13229572E-04  3.31032886E-03 -1.22229882E-03 -5.30397539E-03 -7.50541360E-03
     14        Y-TRANS.    1.53501946E-01  1.03238560E-01  4.61443449E-01 -8.44241498E-03 -5.82070160E-02
     14        Z-TRANS.   -2.55155642E-01 -4.77072719E-01 -6.30515758E-01 -6.99697485E-02 -6.14984842E-03

     15        X-TRANS.    1.39591440E-03 -1.58591941E-02  9.21717522E-03 -1.57745907E-02 -1.86487657E-02
     15        Y-TRANS.   -1.85019455E-01  8.99027327E-03  3.56353244E-01 -6.38023822E-01  1.96491988E-01
     15        Z-TRANS.    3.35700389E-01 -3.71375637E-01 -4.18024081E-01  9.95694987E-01 -4.69034214E-01

     16        X-TRANS.   -1.55447471E-01 -3.01157943E-02 -5.84868727E-02 -8.59405731E-02  2.94716327E-02
     16        Y-TRANS.   -4.95525292E-01 -1.98100973E-01  2.40549834E-02 -2.02368779E-02 -1.92204417E-02
     16        Z-TRANS.   -8.57003891E-01 -6.98934692E-01 -9.57101972E-01 -1.79074881E-01  5.97228237E-02

     17        X-TRANS.   -1.55933852E-01 -2.99814729E-02 -5.80408587E-02 -8.85280879E-02  3.04330879E-02
     17        Y-TRANS.   -3.88132296E-01 -1.80361278E-01  1.84914968E-02 -2.85071427E-03 -5.85101776E-02
     17        Z-TRANS.   -5.67898833E-01 -6.42427050E-01 -8.84723679E-01 -6.68077149E-02 -6.57860546E-03

     18        X-TRANS.    2.15661479E-06 -5.73876795E-07 -5.85860327E-02 -9.03905843E-02  2.24426159E-04
     18        Y-TRANS.   -4.98638132E-01 -2.00231599E-01 -3.82976947E-06 -9.37610551E-05 -2.15461239E-02
     18        Z-TRANS.    2.38326848E-06 -6.61880500E-06  2.96924994E-01 -1.37816007E-02  3.56388047E-05

     19        X-TRANS.    2.15272374E-06 -5.85456230E-07 -5.79281858E-02 -9.05586917E-02  2.24815938E-04
     19        Y-TRANS.   -4.17023346E-01 -1.86521538E-01 -3.32514242E-06 -2.25986877E-04 -5.00216544E-02
     19        Z-TRANS.   -2.02918288E-06 -4.61278370E-06  3.21403990E-01  3.00674612E-02 -7.55738415E-05

     20        X-TRANS.    2.18190661E-06 -6.00741084E-07 -5.81319913E-02 -9.03760179E-02  2.24426159E-04
     20        Y-TRANS.   -3.89883268E-01 -1.81472904E-01 -3.14431007E-06 -2.60621126E-04 -5.73841490E-02
     20        Z-TRANS.   -3.37937743E-06 -2.67021769E-06  2.58411405E-01  3.96860012E-02 -1.00822867E-04

     21        X-TRANS.    1.55447471E-01  3.01157943E-02 -5.84846136E-02 -8.62211075E-02 -2.90385448E-02
     21        Y-TRANS.   -4.95525292E-01 -1.98054655E-01 -2.40625495E-02  2.00695544E-02 -1.93243828E-02
     21        Z-TRANS.    8.57003891E-01  6.98934692E-01 -9.57078269E-01 -1.79626173E-01 -5.88133391E-02

     22        X-TRANS.    1.55933852E-01  2.99814729E-02 -5.80386650E-02 -8.88171507E-02 -2.99913382E-02
     22        Y-TRANS.   -3.88132296E-01 -1.80361278E-01 -1.84977631E-02  2.31959200E-03 -5.85101776E-02
     22        Z-TRANS.    5.67898833E-01  6.42427050E-01 -8.84702136E-01 -6.67465833E-02  6.90342139E-03

     23        X-TRANS.    1.51653696E-04 -4.13385827E-03 -1.68038201E-03 -4.67521381E-03  6.86877436E-03
     23        Y-TRANS.    2.53501946E-01  1.11116258E-01 -5.02323190E-01 -3.13447850E-02 -7.76959723E-02
     23        Z-TRANS.    4.24708171E-01  5.19221862E-01 -6.86173203E-01 -1.43860149E-01 -2.34560416E-02

     24        X-TRANS.    1.23540856E-04 -3.31542381E-03 -1.22135597E-03 -5.24381560E-03  7.53572975E-03
     24        Y-TRANS.    1.53501946E-01  1.03288560E-01 -4.61440358E-01  7.91528249E-03 -5.92503248E-02
     24        Z-TRANS.    2.55155642E-01  4.77072719E-01 -6.30499676E-01 -6.99132120E-02  6.48765699E-03

     25        X-TRANS.   -1.37159533E-03  1.58499305E-02  9.21360491E-03 -1.55675458E-02  1.87786921E-02
```

CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

```
NODAL POINT   ELEMENT
  NUMBER     REFERENCE         6                7                8                9                10

    25        Y-TRANS.   -1.35019455E-01   9.00880037E-03  -3.56349144E-01   6.39774711E-01   1.93287137E-01
    25        Z-TRANS.   -3.35700389E-01   3.71421955E-01  -4.18012486E-01   1.00000000E+00   4.64270247E-01

    26        X-TRANS.   -1.55058366E-01  -3.70912459E-02   5.57178102E-02   7.81922355E-02   4.07232568E-02
    26        Y-TRANS.    1.00000000E+00   3.76655859E-01  -9.99988441E-01  -1.31694716E-01  -8.44521438E-02
    26        Z-TRANS.   -6.25291829E-03   3.60676239E-01  -3.97216669E-01  -9.08826046E-02  -2.25638805E-02

    27        X-TRANS.   -1.55447471E-01  -3.67577592E-02   5.53835922E-02   8.06288004E-02   4.18382633E-02
    27        Y-TRANS.    6.34455253E-01   3.55859194E-01  -9.20552490E-01   1.14199909E-02  -5.49588566E-02
    27        Z-TRANS.   -5.72568093E-02   3.29133358E-01  -3.66016652E-01  -7.16743190E-02   4.42615851E-03

    28        X-TRANS.    1.46692607E-03  -5.62760537E-02   2.95267844E-02   4.57790606E-02   5.81203984E-02
    28        Y-TRANS.    2.54085603E-01  -2.83001390E-01   4.35373765E-02   6.81408877E-03  -1.31788653E-02
    28        Z-TRANS.   -4.39883268E-01  -4.37841593E-02   2.21072733E-01  -2.55229386E-02   5.19272412E-03

    29        X-TRANS.    1.46692607E-03  -5.56739231E-02   2.91956148E-02   4.58561990E-02   5.82070160E-02
    29        Y-TRANS.    2.13424125E-01  -2.98749421E-01   5.99674374E-02   4.24694784E-02   3.92550888E-03
    29        Z-TRANS.   -3.68190661E-01  -6.35479389E-02   2.16768914E-01  -4.36534810E-02   3.15634474E-02

    30        X-TRANS.    1.46498054E-03  -5.58591941E-02   2.92994887E-02   4.57671396E-02   5.81203984E-02
    30        Y-TRANS.    2.00000000E-01  -2.49143122E-01   3.48385496E-02   5.09075206E-02   7.42312689E-03
    30        Z-TRANS.   -3.44455253E-01  -3.86521538E-02   1.97228862E-01  -4.88707766E-02   3.78822001E-02

    31        X-TRANS.   -9.22568093E-06   4.64566929E-06   2.84048208E-04   7.57794193E-04  -1.00563014E-05
    31        Y-TRANS.    1.78501946E-04  -3.49884206E-01  -5.76881566E-06  -3.95901430E-05  -8.69207449E-03
    31        Z-TRANS.    2.29474708E-06  -7.22556739E-06   3.36908208E-01  -1.16737598E-02   2.99913382E-05

    32        X-TRANS.   -3.22276265E-06   4.64566929E-06   2.83846018E-04   7.56797938E-04  -1.00476397E-05
    32        Y-TRANS.    3.74708171E-04  -3.81333951E-01  -6.46203099E-06   1.11323679E-04   2.43871806E-02
    32        Z-TRANS.   -1.91342412E-06  -5.51644280E-06   3.65463079E-01   3.29086297E-02  -8.18536163E-05

    33        X-TRANS.   -3.22665370E-06   4.64566329E-06   2.84039778E-04   7.57842991E-04  -1.00563014E-05
    33        Y-TRANS.    5.21592607E-04  -3.63131079E-01  -6.06227729E-06   2.53600763E-05   5.54352534E-03
    33        Z-TRANS.    4.82003891E-07  -6.48448356E-06   3.48896117E-01   7.51246695E-03  -1.81333911E-05
```

GENERALIZED INERTIA PROPERTIES

VARIABLES	11	12	13	14	15
WNX	1.31532959E+04	1.36318685E+04	8.20837363E+03	1.38481395E+04	1.10906860E+04
WNY	6.61479808E+03	1.27906844E+04	1.00846576E+04	9.12324109E+03	1.24273351E+05
WNZ	6.69924065E+03	1.26940061E+04	5.38717422E+03	1.69286570E+04	1.24311118E+05
PX	-4.99032160E+00	-2.17158100E+05	-1.84386186E+03	-3.75078907E+01	3.65542814E+05
PY	1.32640724E+01	8.45696133E+05	-1.07233413E+05	-1.04437073E+03	-8.44577469E+05
PZ	3.74334725E+01	8.48944686E+05	9.53436229E+04	8.58899507E+02	-8.45047303E+05
GM DIAGONAL	1.32336673E+04	1.95582795E+04	1.18401027E+04	1.99500187E+04	1.29837578E+05

```
                                  NATURAL FREQUENCIES (RADIANS/SEC)

                           11              12              13              14              15

                      1.00749277E+03  1.05969809E+03  1.20336962E+03  1.20433575E+03  1.37674831E+03


                            CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

 NODAL POINT   ELEMENT
   NUMBER     REFERENCE        11              12              13              14              15

     1         X-TRANS.   3.61389446E-02 -7.00685250E-03  2.05755578E-01 -7.69559033E-02 -6.93589744E-01
     1         Y-TRANS.  -1.06791360E-01  9.16776060E-03  7.48225152E-03  6.22569938E-02 -1.04743590E-02
     1         Z-TRANS.   8.20084614E-02  6.18377885E-02 -1.39452333E-01  1.87292556E-01 -3.81709402E-02

     2         X-TRANS.   3.84101536E-02 -9.41951775E-03  2.05755578E-01 -7.75248933E-02 -7.05555556E-01
     2         Y-TRANS.  -9.49899800E-02 -1.35767099E-03 -2.14629817E-03 -1.32527264E-01 -7.69658120E-04
     2         Z-TRANS.   1.81340459E-01 -1.52475824E-02 -1.70765720E-01  3.36842105E-02 -3.87863248E-02

     3         X-TRANS.  -6.33711868E-06 -1.83981379E-03  2.40745436E-01  2.22901849E-03 -7.94444444E-01
     3         Y-TRANS.  -1.02516143E-01  2.68700578E-06 -2.91455375E-04  6.19250830E-02 -1.67735043E-04
     3         Z-TRANS.   3.36677355E-05  5.42301287E-01  4.59305274E-01  4.22664770E-03 -5.01709402E-01

     4         X-TRANS.  -7.55288354E-06 -2.16003592E-02  2.14376268E-01  1.98624941E-03 -7.78632479E-01
     4         Y-TRANS.  -9.58345135E-02  7.35130874E-06  6.95360041E-04 -1.32053106E-01  1.11581197E-04
     4         Z-TRANS.  -5.59786239E-06 -7.83963551E-02 -2.52535497E-01 -2.32954007E-03 -1.38162393E-02

     5         X-TRANS.  -5.02115342E-05 -6.99552894E-01 -5.78346856E-01 -5.31531532E-03 -4.14487179E-01
     5         Y-TRANS.   1.00000000E+00 -3.92434103E-05 -5.78600406E-04  9.77714557E-02  7.73931624E-05
     5         Z-TRANS.   6.23914496E-05  1.00000000E+00  1.00000000E+00  9.20815552E-03 -7.60683761E-01

     6         X-TRANS.  -3.61389446E-02 -7.00950232E-03  2.04868154E-01  8.07491702E-02 -6.93589744E-01
     6         Y-TRANS.  -1.06791360E-01 -9.16219826E-03 -8.06795132E-03  6.21147463E-02  1.01367521E-02
     6         Z-TRANS.  -8.20084614E-02  6.18606020E-02 -1.37423935E-01 -1.89853011E-01 -3.78846154E-02

     7         X-TRANS.  -3.84324204E-02 -9.42217635E-03  2.04994929E-01  8.13181603E-02 -7.05982906E-01
     7         Y-TRANS.  -9.49899800E-02  1.37237670E-03  3.54208925E-03 -1.32479848E-01  9.94017094E-04
     7         Z-TRANS.  -1.81340459E-01 -1.52176914E-02 -1.70385396E-01 -3.68326221E-02 -3.89529915E-02

     8         X-TRANS.  -8.15408595E-05  2.02148106E-02  3.29107505E-02 -7.42057847E-02 -5.01282051E-01
     8         Y-TRANS.  -6.64217324E-02  2.07517318E-03 -7.12221095E-02 -7.08866761E-02 -1.86282051E-01
     8         Z-TRANS.  -1.14740592E-01 -1.09059794E-03 -1.35268763E-01 -9.65860597E-02  1.03632479E-01

     9         X-TRANS.  -8.24092630E-05  2.03143770E-02  3.29741379E-02 -7.43480322E-02 -5.01282051E-01
     9         Y-TRANS.  -9.47227789E-02  1.13390395E-02 -8.32657201E-02 -3.29397819E-02  1.74017094E-01
     9         Z-TRANS.  -1.63816522E-01 -6.10411602E-03 -1.19523327E-01 -1.11948791E-01 -1.01196581E-01

    10         X-TRANS.  -8.18303273E-05  1.90105162E-02  3.28600406E-02 -7.40635372E-02 -5.20085470E-01
    10         Y-TRANS.  -1.01024271E-01  2.26428063E-02 -7.88793103E-02 -1.94642010E-02  4.39743590E-01
    10         Z-TRANS.  -1.74972167E-01 -1.25001826E-02 -1.03803245E-01 -1.05832148E-01 -2.53504274E-01

    11         X-TRANS.   3.59385438E-02 -8.55834230E-03 -1.54792089E-01 -1.93409199E-01 -7.04700855E-01
    11         Y-TRANS.  -1.73836562E-02 -5.83561002E-02 -1.85851927E-01 -3.79943101E-02  2.47094017E-02
    11         Z-TRANS.  -1.34001336E-01 -2.26245723E-02 -6.31465517E-02 -6.82313893E-02  1.02136752E-02

    12         X-TRANS.   3.82097528E-02 -1.09870949E-02 -1.55172414E-01 -1.93172119E-01 -7.16239316E-01
```

```
                         CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE         11              12              13              14              15

    12        Y-TRANS.    -1.07370296E-01  1.46042893E-02 -4.62347870E-02 -1.88335704E-01  3.96153846E-02
    12        Z-TRANS.    -1.74192830E-01  7.07418857E-03 -1.28803245E-01 -3.10621147E-02  1.60940171E-02

    13        X-TRANS.     1.14940993E-03 -3.23411870E-03 -1.17761156E-01 -2.71977240E-01 -7.97863248E-01
    13        Y-TRANS.     4.81407259E-02 -4.67521340E-01  1.76343813E-01  4.66714083E-01  4.35897436E-01
    13        Z-TRANS.    -9.13382320E-02 -2.69383109E-01  1.50228195E-01  2.33475581E-01  2.59444444E-01

    14        X-TRANS.     1.04141617E-03 -2.28994163E-02 -1.04627282E-01 -2.41963016E-01 -7.81196581E-01
    14        Y-TRANS.     4.76953908E-02  6.09398153E-02 -6.20816430E-02 -2.75296349E-01  1.32136752E-02
    14        Z-TRANS.    -8.46804721E-02  3.53851146E-02 -1.39072008E-01 -8.51114272E-02  1.98418803E-03

    15        X-TRANS.    -2.11957248E-03 -6.98073280E-01  2.87145030E-01  6.52441916E-01 -4.06025641E-01
    15        Y-TRANS.    -5.05010020E-01 -8.59692968E-01  3.97439148E-01  1.00000000E+00  6.71367521E-01
    15        Z-TRANS.     8.61055444E-01 -5.01926913E-01  3.02104462E-01  5.31531532E-01  3.86111111E-01

    16        X-TRANS.    -3.40013360E-02 -7.89860927E-03 -4.65770791E-02 -2.72024656E-01 -6.94017094E-01
    16        Y-TRANS.     1.23135159E-01 -4.86491779E-02  2.57606491E-02 -2.01327643E-01  2.52692308E-02
    16        Z-TRANS.    -5.29948786E-02 -3.79502994E-02  6.78625761E-02 -1.42437174E-01  3.40940171E-02

    17        X-TRANS.    -3.62725451E-02 -1.03025551E-02 -4.61713996E-02 -2.72403983E-01 -7.05982906E-01
    17        Y-TRANS.     2.02560677E-01  1.28702593E-02 -9.22160243E-03 -2.06495970E-01  3.29358974E-02
    17        Z-TRANS.     4.77176575E-03  9.17938117E-03 -1.10509635E-01 -4.86960645E-02  1.44273504E-02

    18        X-TRANS.     4.28412380E-06  2.01194031E-02 -6.44903651E-02 -6.07396871E-04 -5.01282051E-01
    18        Y-TRANS.     1.31462926E-01 -1.27625874E-05  1.24987323E-03 -2.28022760E-01  1.93974359E-05
    18        Z-TRANS.     1.34224004E-06  2.19526879E-03 -1.12145030E-02 -1.08345187E-04 -2.14444444E-01

    19        X-TRANS.     4.30416388E-06  2.02243393E-02 -6.46298174E-02 -6.08345187E-04 -5.01282051E-01
    19        Y-TRANS.     1.87775551E-01 -2.02600972E-05  1.18141481E-03 -2.15078236E-01  1.67222222E-05
    19        Z-TRANS.    -6.08550434E-07  1.28509142E-02  2.39224138E-02  2.25462304E-04  2.02094017E-01

    20        X-TRANS.     4.27521710E-06  1.89144732E-02 -6.43128803E-02 -6.05974395E-04 -5.19658120E-01
    20        Y-TRANS.     2.00289468E-01 -2.18969178E-05  1.04893509E-03 -1.90753912E-01  1.41965812E-05
    20        Z-TRANS.    -1.00556669E-06  2.58707462E-02  3.09964503E-02  2.95448080E-04  5.07692308E-01

    21        X-TRANS.     3.40236028E-02 -7.90704244E-03 -4.95436105E-02  2.71123755E-01 -6.94017094E-01
    21        Y-TRANS.     1.23135159E-01  4.86257905E-02 -2.35420892E-02 -2.01801802E-01 -2.52350427E-02
    21        Z-TRANS.     5.29948786E-02 -3.79634224E-02  6.62905680E-02  1.43669986E-01  3.40726496E-02

    22        X-TRANS.     3.62725451E-02 -1.03113189E-02 -4.91506085E-02  2.71503082E-01 -7.05982906E-01
    22        Y-TRANS.     2.02560677E-01 -1.29148134E-02  1.14908722E-02 -2.06306306E-01 -3.29059829E-02
    22        Z-TRANS.    -4.76953908E-03  9.17992402E-03 -1.11029412E-01  4.66287340E-02  1.44188034E-02

    23        X-TRANS.    -1.14117123E-03 -3.23172245E-03 -1.20727688E-01  2.69748696E-01 -7.98290598E-01
    23        Y-TRANS.     4.81352594E-02  4.67508487E-01 -1.81414807E-01  4.63442390E-01 -4.35897436E-01
    23        Z-TRANS.     9.12936985E-02 -2.69400992E-01  1.52763692E-01 -2.30678046E-01  2.59487179E-01

    24        X-TRANS.    -1.03718548E-03 -2.28970946E-02 -1.07264199E-01  2.39971550E-01 -7.81196581E-01
    24        Y-TRANS.     4.76953908E-02 -6.09575510E-02  6.50988844E-02 -2.74110953E-01 -1.31965812E-02
    24        Z-TRANS.     8.47027388E-02  3.53700628E-02 -1.39959432E-01  8.25509720E-02  1.98247863E-03

    25        X-TRANS.     2.02961479E-03 -6.98075282E-01  2.94244422E-01 -6.47226174E-01 -4.05940171E-01
```

```
                          CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE        11               12               13               14               15

    25        Y-TRANS.   -5.05010020E-01   8.59809222E-01  -4.08341785E-01   9.92413466E-01  -6.71794872E-01
    25        Z-TRANS.   -8.61055444E-01  -5.01726703E-01   3.07809331E-01  -5.25841631E-01   3.86196581E-01

    26        X-TRANS.   -3.59385438E-02  -8.54552673E-03  -1.56820487E-01   1.90516833E-01  -7.04700855E-01
    26        Y-TRANS.   -1.73792029E-02   5.83530940E-02   1.86232252E-01  -3.45566619E-02  -2.46495726E-02
    26        Z-TRANS.    1.34001336E-01  -2.26494866E-02  -6.38691684E-02   6.70934092E-02   1.01752137E-02

    27        X-TRANS.   -3.82097528E-02  -1.09738861E-02  -1.57200811E-01   1.90279753E-01  -7.16666667E-01
    27        Y-TRANS.   -1.07370296E-01  -1.45940848E-02   4.83265720E-02  -1.87482219E-01  -3.96410256E-02
    27        Z-TRANS.    1.74192830E-01   7.04298510E-03  -1.29056795E-01   2.86818397E-02   1.61153846E-02

    28        X-TRANS.    7.98485861E-05   2.02226978E-02   3.20486815E-02   7.48221906E-02  -5.01282051E-01
    28        Y-TRANS.   -6.64217324E-02  -2.86520154E-03   7.19827586E-02  -6.95590327E-02   1.86282051E-01
    28        Z-TRANS.    1.14740592E-01  -1.11090559E-03  -1.36282961E-01   9.40730204E-02   1.03547009E-01

    29        X-TRANS.    8.06947228E-05   2.03222706E-02   3.21120690E-02   7.49644381E-02  -5.01282051E-01
    29        Y-TRANS.   -9.47227789E-02  -1.13217027E-02   8.36080122E-02  -3.14035088E-02  -1.73974359E-01
    29        Z-TRANS.    1.63816522E-01  -6.12949240E-03  -1.20740365E-01   1.09720247E-01  -1.01153846E-01

    30        X-TRANS.    8.00712536E-05   1.90183871E-02   3.19979716E-02   7.46799431E-02  -5.20085470E-01
    30        Y-TRANS.   -1.01024271E-01  -2.26237657E-02   7.90821501E-02  -1.80132764E-02  -4.39743590E-01
    30        Z-TRANS.    1.74972167E-01  -1.25266307E-02  -1.04969574E-01   1.03888099E-01  -2.53461538E-01

    31        X-TRANS.    2.11578713E-06   1.06762597E-01  -1.25849391E-03   3.64248459E-06   9.99572650E-01
    31        Y-TRANS.   -2.64974393E-04  -8.23606063E-07   6.21450304E-05  -1.01944049E-02  -1.99102564E-05
    31        Z-TRANS.    7.78445780E-07  -1.06872538E-04  -7.53296146E-03  -7.20246562E-05  -6.19230769E-04

    32        X-TRANS.    2.11111111E-06   1.06599437E-01  -1.25722617E-03   3.63300142E-06   9.97863248E-01
    32        Y-TRANS.   -2.23335560E-04   1.46093352E-06  -2.02205882E-04   3.55286866E-02   2.09444444E-05
    32        Z-TRANS.   -9.13159653E-07  -5.89392104E-05   2.68509128E-02   2.50497866E-04  -2.32820513E-04

    33        X-TRANS.    2.11511913E-06   1.06774819E-01  -1.25912779E-03   3.64201043E-06   1.00000000E+00
    33        Y-TRANS.   -2.46492986E-04   1.59690708E-07  -5.15973631E-05   9.47368421E-03  -2.33376068E-06
    33        Z-TRANS.    5.06568693E-08  -8.62542252E-05   7.25912779E-03   6.67140825E-05  -4.62393162E-04
```

GENERALIZED INERTIA PROPERTIES

VARIABLES	16	17	18	19	20
WNX	4.49637847E+04	4.41563255E+04	1.87547312E+04	6.83417642E+04	1.44983817E+04
WNY	4.66476114E+04	1.70675464E+04	2.17145558E+04	4.25923677E+04	1.13153543E+04
WNZ	1.81074884E+04	4.61491338E+04	2.11874286E+04	4.46586751E+04	1.89149873E+04
PX	4.23751146E+03	6.86745461E+00	4.01127806E+05	2.68242563E+02	-4.51132124E+01
PY	1.34490765E+05	1.46051485E+02	1.47309389E+06	5.01191735E+02	3.30674779E+02
PZ	-1.35196947E+05	-4.29227309E+02	1.48464538E+06	4.73760979E+02	-6.62234658E+02
GM DIAGONAL	5.48594422E+04	5.36365058E+04	3.08284078E+04	7.77964035E+04	2.23643617E+04

```
                                   NATURAL FREQUENCIES (RADIANS/SEC)

                                 16               17               18               19               20

                          2.48526099E+03   2.49077254E+03   2.67732900E+03   2.69850389E+03   2.81151036E+03

                              CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

  NODAL POINT   ELEMENT
    NUMBER     REFERENCE          16               17               18               19               20

       1       X-TRANS.   -1.34243176E-01   3.64965145E-01  -1.37659033E-01   3.96046852E-01  -4.20881831E-03
       1       Y-TRANS.   -4.94044665E-03   1.00000000E+00   3.17981340E-02   3.49975598E-01   8.33442845E-02
       1       Z-TRANS.    3.82547550E-01   3.50798231E-01   1.65055131E-01  -3.36017570E-01  -1.87707338E-01

       2       X-TRANS.   -1.39867659E-01   3.69237688E-01  -1.31806616E-01   4.05807711E-01  -6.00741363E-03
       2       Y-TRANS.   -5.51943755E-03  -6.70339555E-01   3.10347752E-02  -7.28892143E-01  -1.26498103E-01
       2       Z-TRANS.   -2.12820513E-01  -3.78007646E-01   3.59796438E-01   2.58174719E-01   2.57593388E-01

       3       X-TRANS.   -1.64267990E-01  -3.95997302E-04  -9.05852417E-02  -2.11298194E-04  -1.29207063E-03
       3       Y-TRANS.   -1.82961125E-03   9.91679784E-01  -7.71755725E-05   9.41922889E-01   8.24469595E-02
       3       Z-TRANS.   -1.74028122E-01  -5.05059591E-04   8.74469890E-01   3.34797462E-06  -3.79430692E-03

       4       X-TRANS.   -1.45574855E-01  -3.73285361E-04  -3.43002545E-02  -2.10078087E-04  -1.59667883E-03
       4       Y-TRANS.    1.22001654E-03  -6.55273218E-01   6.61238738E-05  -7.16203026E-01  -1.26909561E-01
       4       Z-TRANS.   -2.84036394E-01  -6.12772656E-04   8.49872774E-01   2.77208394E-04   8.96870468E-04

       5       X-TRANS.   -5.69561621E-02  -3.62716438E-04   9.52502120E-01  -1.04294778E-04  -5.75094300E-03
       5       Y-TRANS.    1.26964433E-04  -6.79109512E-02   1.40033927E-05  -1.40287945E-01  -3.29607951E-02
       5       Z-TRANS.   -3.01157982E-01  -6.35484596E-04   2.08821035E-01  -6.11517814E-05  -5.75172124E-04

       6       X-TRANS.   -1.32919768E-01  -3.65414896E-01  -1.37574215E-01  -3.96046852E-01   4.75880806E-03
       6       Y-TRANS.    1.24979322E-03   1.00000000E+00  -3.19508058E-02   3.49975598E-01   8.36901363E-02
       6       Z-TRANS.    3.83370968E-01  -3.49449057E-01   1.64970314E-01   3.36017570E-01   1.82706920E-01

       7       X-TRANS.   -1.38544251E-01  -3.69912300E-01  -1.31721798E-01  -4.06051733E-01   6.27625042E-03
       7       Y-TRANS.    8.01406121E-03  -6.70339535E-01  -3.08990670E-02  -7.28892143E-01  -1.26666701E-01
       7       Z-TRANS.   -2.14143921E-01   3.77198163E-01   3.59881255E-01  -2.57686676E-01  -2.53123921E-01

       8       X-TRANS.    3.40529363E-01  -5.84663818E-01  -1.87955895E-01  -2.11639824E-03   1.13387953E-01
       8       Y-TRANS.    2.75765095E-01  -8.59680684E-02  -1.69720102E-02   3.63836018E-01   1.64550163E-01
       8       Z-TRANS.    2.23904053E-01   2.83112210E-01  -2.43256997E-02   6.32991703E-01   1.45054896E-01

       9       X-TRANS.    3.42349049E-01  -5.87812008E-01  -1.88543618E-01  -2.09614446E-03   1.13420480E-01
       9       Y-TRANS.   -1.38213400E-01   2.10029233E-01   6.21119593E-02  -1.45534407E-01  -1.53409617E-01
       9       Z-TRANS.    5.36476427E-02  -1.42509562E-01  -3.70398643E-02  -2.56222548E-01  -8.62286243E-02

      10       X-TRANS.    3.33995037E-01  -5.73420233E-01  -1.89058524E-01  -1.97218155E-03   1.07443263E-01
      10       Y-TRANS.   -2.16453894E-01   2.45109052E-01   1.51223856E-01  -2.51830161E-01  -2.17452401E-01
      10       Z-TRANS.   -5.33250620E-04  -2.23903755E-01  -8.29262087E-02  -4.44363104E-01  -1.21782372E-01

      11       X-TRANS.    3.80066170E-01  -5.92984034E-02  -1.32230704E-01   3.91898497E-01   5.61630399E-02
      11       Y-TRANS.   -5.40612076E-01   6.99033396E-01  -2.15012723E-01  -2.02757443E-01   3.17480244E-01
      11       Z-TRANS.    5.89826303E-01  -1.59433326E-01  -9.33842239E-02   1.00000000E+00   1.29362204E-01

      12       X-TRANS.    3.86931348E-01  -5.69822352E-02  -1.24427481E-01   4.02635432E-01   3.29874573E-02
```

```
                       CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE        16               17               18               19               20

    12        Y-TRANS.   5.00413565E-01  -5.04384979E-01  -3.08142494E-01   1.76891166E-01  -2.66639984E-01
    12        Z-TRANS.  -3.87675765E-01   1.00382280E-01  -1.45122986E-01  -7.73060029E-01  -8.44618989E-02

    13        X-TRANS.   9.52026468E-02   1.35956825E-01  -6.39525021E-02   1.75988287E-02  -2.30168236E-01
    13        Y-TRANS.  -5.14971050E-01   1.18641781E-01  -7.64631043E-01  -5.00244021E-01   5.75548488E-01
    13        Z-TRANS.   6.93879239E-01  -4.87069935E-01  -4.49109415E-01   8.17471938E-01   3.12674496E-01

    14        X-TRANS.   8.57733664E-02   1.18956600E-01  -5.32739610E-03   1.96046852E-02  -2.81114746E-01
    14        Y-TRANS.   1.65922250E-01  -3.75758939E-01  -6.85496183E-01   3.90678380E-01  -1.60159369E-01
    14        Z-TRANS.  -5.68155500E-01   1.58646278E-01  -3.96183206E-01  -6.31771596E-01  -3.95028898E-02

    15        X-TRANS.   4.76923077E-02   3.42478075E-02   1.00000000E+00   4.70473402E-02  -9.94635077E-01
    15        Y-TRANS.  -1.07692308E-01  -2.38587812E-01  -1.74809160E-01   7.18643241E-02   8.20680651E-02
    15        Z-TRANS.  -1.29693962E-01  -1.01506634E-01  -9.78795589E-02  -1.26256711E-01   6.34445246E-02

    16        X-TRANS.  -2.31348222E-01   2.90532944E-01  -1.32145886E-01  -3.90678380E-01   2.48743661E-02
    16        Y-TRANS.   4.50206782E-03   3.81380706E-01  -1.54792197E-01  -8.16251830E-01   4.39813903E-01
    16        Z-TRANS.   1.00000000E+00  -3.42478075E-01  -1.35368957E-01   6.42996584E-01   2.67413432E-01

    17        X-TRANS.  -2.31430935E-01   2.97054194E-01  -1.24427481E-01  -3.98487067E-01  -1.84897134E-03
    17        Y-TRANS.  -9.66087676E-02  -1.69732404E-01  -2.78625954E-01   6.30551489E-01  -4.64868158E-01
    17        Z-TRANS.  -7.39867659E-01   2.79739150E-01  -1.94147583E-01  -5.08540752E-01  -2.32528226E-01

    18        X-TRANS.  -6.80810587E-01  -1.17247583E-03  -1.92281595E-01  -1.33504148E-04  -5.82122825E-04
    18        Y-TRANS.  -6.69148056E-04   3.83854284E-01   1.37404580E-04  -7.80624695E-01   3.87141387E-01
    18        Z-TRANS.  -2.54673284E-01  -4.41421183E-04  -5.91772689E-03  -2.01854563E-05  -3.91578220E-04

    19        X-TRANS.  -6.85277089E-01  -1.18034630E-03  -1.93129771E-01  -1.34187408E-04  -5.83825987E-04
    19        Y-TRANS.   7.83788255E-05  -3.91499888E-02  -4.14503817E-05   3.24792582E-01  -2.93891489E-01
    19        Z-TRANS.   3.02564103E-01   5.17652350E-04   7.76675148E-02   5.49536359E-05   5.04232482E-04

    20        X-TRANS.  -6.67080232E-01  -1.14908927E-03  -1.92875318E-01  -1.32918497E-04  -5.48897888E-04
    20        Y-TRANS.   2.56079404E-04  -1.40274342E-01  -7.98218830E-05   5.52952660E-01  -4.09216214E-01
    20        Z-TRANS.   3.86435070E-01   6.59545761E-04   1.78456319E-01   1.10566130E-04   7.03881662E-04

    21        X-TRANS.  -2.30355666E-01  -2.91432426E-01  -1.32315522E-01   3.90678380E-01  -2.51191701E-02
    21        Y-TRANS.  -5.83540116E-03   3.81380706E-01   1.55046650E-01  -8.16251830E-01   4.39349898E-01
    21        Z-TRANS.   9.99172870E-01   3.46076006E-01  -1.35114504E-01  -6.42996584E-01  -2.68534213E-01

    22        X-TRANS.  -2.30355666E-01  -2.97728806E-01  -1.24597116E-01   3.98243045E-01   9.25852346E-04
    22        Y-TRANS.   9.71877585E-02  -1.69440072E-01   2.78456319E-01   6.30795510E-01  -4.63733462E-01
    22        Z-TRANS.  -7.38875103E-01  -2.82437598E-01  -1.94317218E-01   5.08296730E-01   2.33621215E-01

    23        X-TRANS.   9.56989247E-02  -1.35597032E-01  -6.39694656E-02  -1.76134700E-02   2.31516706E-01
    23        Y-TRANS.   5.14474773E-01   1.20283337E-01   7.64800679E-01  -4.99511957E-01   5.78288984E-01
    23        Z-TRANS.   6.92307692E-01   4.89543512E-01  -4.48854962E-01  -8.17715959E-01  -3.14947678E-01

    24        X-TRANS.   8.61869313E-02  -1.18664268E-01  -5.34690416E-03  -1.95510005E-02   2.82733806E-01
    24        Y-TRANS.  -1.64598842E-01  -3.76433551E-01   6.85326548E-01   3.91166423E-01  -1.60644901E-01
    24        Z-TRANS.  -5.67659222E-01  -1.60535192E-01  -3.96352841E-01   6.31527574E-01   4.09593996E-02

    25        X-TRANS.   4.76923077E-02  -3.43152687E-02   1.00000000E+00  -4.59980478E-02   1.00000000E+00
```

```
                          CORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM

NODAL POINT   ELEMENT
  NUMBER     REFERENCE        16               17               18               19               20

    25        Y-TRANS.    1.08519438E-01  -2.38138071E-01   1.74724343E-01   7.20351391E-02   8.26397200E-02
    25        Z-TRANS.   -1.30024814E-01   1.01056892E-01  -9.78795589E-02   1.26159102E-01  -6.34806169E-02

    26        X-TRANS.    3.79900744E-01   6.06701147E-02  -1.32145886E-01  -3.92142509E-01  -5.62708952E-02
    26        Y-TRANS.    5.38130687E-01   6.91702271E-01   2.15182358E-01  -2.02513421E-01   3.20687781E-01
    26        Z-TRANS.    6.99330025E-01   1.61929331E-01  -9.30449534E-02  -1.00000000E+00  -1.31776111E-01

    27        X-TRANS.    3.86765922E-01   5.83764336E-02  -1.24342663E-01  -4.02879453E-01  -3.29417725E-02
    27        Y-TRANS.   -4.98593879E-01  -5.06183944E-01   3.07972858E-01   1.76915569E-01  -2.70017200E-01
    27        Z-TRANS.   -3.87344913E-01  -1.01709017E-01  -1.45377439E-01   7.73060029E-01   8.74784442E-02

    28        X-TRANS.    3.38461538E-01   5.85788172E-01  -1.87871077E-01   1.94460713E-03  -1.12707562E-01
    28        Y-TRANS.   -2.75434243E-01  -8.69574934E-02   1.68532655E-02   3.63836018E-01   1.65939992E-01
    28        Z-TRANS.    2.24896609E-01  -2.82212728E-01  -2.42408821E-02  -6.32991703E-01  -1.48249785E-01

    29        X-TRANS.    3.40281224E-01   5.88936362E-01  -1.88464801E-01   1.92020498E-03  -1.12739901E-01
    29        Y-TRANS.    1.37468983E-01   2.10501462E-01  -6.20186599E-02  -1.45632016E-01  -1.54212732E-01
    29        Z-TRANS.    5.31265509E-02   1.42702946E-01  -3.70653096E-02   2.56466569E-01   8.86485577E-02

    30        X-TRANS.    3.32009926E-01   5.74544637E-01  -1.88888889E-01   1.79331381E-03  -1.06795661E-01
    30        Y-TRANS.    2.15632754E-01   2.46008545E-01  -1.51060221E-01  -2.51830161E-01  -2.18565039E-01
    30        Z-TRANS.   -1.44582299E-03   2.23881258E-01  -8.29686175E-02   4.44363104E-01   1.25215297E-01

    31        X-TRANS.   -4.21936228E-04   1.55947930E-06   3.49957591E-02   1.89507077E-05  -6.58652820E-06
    31        Y-TRANS.    6.17535153E-04  -3.54396222E-01  -4.89652248E-05  -2.23230844E-03   1.19738962E-01
    31        Z-TRANS.   -3.57899030E-01  -6.15695975E-04  -3.15097540E-03  -2.36603221E-05  -5.76414898E-04

    32        X-TRANS.   -4.15467328E-04   1.50326053E-06   3.54198473E-02   1.90800390E-05  -6.60557687E-06
    32        Y-TRANS.   -6.31430935E-04   3.62041826E-01   5.03138253E-05   1.89482674E-03  -1.19215442E-01
    32        Z-TRANS.    3.63523573E-01   6.23791320E-04   2.52926209E-03   2.06295754E-05   5.69651449E-04

    33        X-TRANS.   -4.20843672E-04   1.54171351E-06   3.53519932E-02   1.91020010E-05  -6.63086219E-06
    33        Y-TRANS.    8.42845327E-05  -4.84821228E-02  -6.63189143E-06  -4.63884822E-04   1.78409465E-02
    33        Z-TRANS.   -4.99338296E-02  -8.64178038E-05  -7.40033927E-04  -4.74621767E-06  -8.77356378E-05
```

```
                                        CPU TIME USAGE TABLE (SEC)

                                                                TIME IN      TIME OUT       TOTAL

INPUT AND INITIALIZATION OVERLAY                                   .300         1.229         .93

STRUCTURAL STIFFNESS MATRIX GENERATION OVERLAY                    1.229         2.128         .90

NORMAL MODE ANALYSIS OVERLAY                                      2.128       116.605      114.48
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

EXAMPLE OF INPUT AND OUTPUT

LANDING GEAR OPTION

STRUCTURAL ANALYSIS PROGRAM

```
                         STRUCTURAL ANALYSIS PROGRAM --- LEGGED LANDER
                 MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
                         MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EAST

GEAR ANALYSIS DATA        - CARD CODE

                           BLANK - 0          COMMENT CARDS
                                   1          GEAR AND LOAD CARD
                                   2          FRICTION CARD
                                   3          APPLIED DISPLACEMENT CARD
                                   4          STRUT MATERIAL CARDS
                                   5          SOLUTION PARAMETER CARD
                                   6          NODAL POINT CARDS
                                   7          FOOTPAD CARD
                                   8          MATERIAL PARAMETER CARDS
                                   9          MATERIAL CRUSH CARDS
                                  10          COMPRESSION CRUSH DISTANCE CARDS
                                  11          TENSION CRUSH DISTANCE CARDS
                                  12          COMPRESSION SPRING RATE CARDS
                                  13          TENSION SPRING RATE CARDS
                                  14          COMPRESSION PLASTIC FORCE CARDS
                                  15          TENSION PLASTIC FORCE CARDS
                                  16          DATA TERMINATOR CARD

    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     GEAR AND LOAD CARD
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
1    1   1                   0.0       0.0       0.0
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     FRICTION CARD
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
2    1            15.24      .8
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     STRUT MATERIAL CARDS
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
4    1   1
4    2   2
4    3   1
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     SOLUTION PARAMETER CARD
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
5    60  100   2   1       5.65E10
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     NODAL POINT CARDS
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
6    1                0.        23.5        0.
6    2             60.8          0.      19.58
6    3                0.       -23.5        0.
6    4           -25.15          0.       50.8
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     FOOTPAD CARD
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
7    1   2        1.102E09   4.10E09                 1.27      4.45
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     MATERIAL PARAMETER CARDS
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
8    1   1                   7.30E11                 4.57
8    2   2
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     MATERIAL CRUSH CARDS
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
9    2   1   -1   27.94     1.E-20
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      COMPRESSION CRUSH DISTANCE CARDS
10     2             8.36      11.35      14.15      27.94      1.E20
      TENSICN CRUSH DISTANCE CARDS
11     2            1.E20
      COMPRESSION SPRING RATE CARDS
12     2          2482.E06   848.E06  1023.E06  1198.E06   2.98E10
      TENSICN SPRING RATE CARDS
13     2           2.82E10
      COMPRESSION PLASTIC FORCE CARDS
14     2           1.07E09   1.54E09   1.92E09   3.34E09    1.E30
      TENSICN PLASTIC FORCE CARDS
15     2            1.E30
      DATA TERMINATOR CARD
16
```

```
                                               INITIAL CONDITIONS FOR GEAR ANALYSIS
______________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________

                   COORDINATE DEFINITIONS                             EXTERNAL FORCES                             EXTERNAL MOMENTS
 NODAL           SURFACE COORDINATE SYSTEM                     SURFACE COORDINATE SYSTEM                     SURFACE COORDINATE SYSTEM
 POINT            X           Y           Z                     X          Y          Z                     X          Y          Z
______________________________________________________________________________________________________________________

   1       0.          2.35000E+01  0.                    0.         0.         0.                    0.         0.         0.
   2       6.08000E+01  0.          1.95800E+01          0.         0.         0.                    0.         0.         0.
   3       0.          -2.35000E+01  0.                   0.         0.         0.                    0.         0.         0.
   4      -2.51500E+01  0.          5.08000E+01          0.         0.         0.                    0.         0.         0.

STRUCTURAL    AXIAL        AXIAL        AXIAL                           SHEAR      SHEAR                            SHEAR      SHEAR
 ELEMENT      FORCE        STROKE       STIFFNESS                       FORCE      MOMENT                           FORCE      MOMENT
______________________________________________________________________________________________________________________

   1       0.          0.          0.          NODE 1     0.         0.         NODE  4    0.         0.
   2       0.          0.          0.          NODE 2     0.         0.         NODE  4    0.         0.
   3       0.          0.          0.          NODE 3     0.         0.         NODE  4    0.         0.
______________________________________________________________________________________________________________________

                                                                           NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 2
                           TOTAL ENERGY ABSORBED                                  FOOTPAD ATTENUATION ENERGY THIS STEP
     SURFACE COORDINATE SYSTEM        LANDER COORDINATE SYSTEM           SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM
       X        Y        Z        X        Y        Z        X        Y        Z        X        Y        Z
______________________________________________________________________________________________________________________

   0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
______________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________
```

```
                                              INCREMENTAL STEP     1
                                 19 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                 COORDINATE DEFINITIONS                          EXTERNAL FORCES                          EXTERNAL MOMENTS
  NODAL       SURFACE COORDINATE SYSTEM                  SURFACE COORDINATE SYSTEM                  SURFACE COORDINATE SYSTEM
  POINT          X            Y            Z              X            Y            Z              X            Y            Z

    1       0.           2.35000E+01  0.          -8.02075E+07 -7.57100E+07  1.64051E+08  0.           0.           0.
    2       6.08000E+01  0.           1.95800E+01 -4.59990E+08  0.           1.68227E+08  0.           0.           0.
    3       0.          -2.35000E+01  0.          -8.02075E+07  7.57100E+07  1.64051E+08  0.           0.           0.
    4      -2.48960E+01  0.           5.09206E+01  6.20405E+08  0.          -4.96329E+06  0.           0.           0.

 STRUCTURAL    AXIAL        AXIAL        AXIAL                     SHEAR        SHEAR                      SHEAR        SHEAR
  ELEMENT      FORCE        STROKE       STIFFNESS                 FORCE        MOMENT                     FORCE        MOMENT

    1      -1.97682E+08 -3.59645E-03  5.49657E+10   NODE 1      0.           0.            NODE  4      0.           0.
    2      -4.89787E+08 -1.97336E-01  2.48200E+09   NODE 2      0.           0.            NODE  4      0.           0.
    3      -1.97682E+08 -3.59645E-03  5.49657E+10   NODE 3      0.           0.            NODE  4      0.           0.

                                                                          NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 2
                      TOTAL ENERGY ABSORBED                                   FOOTPAD ATTENUATION ENERGY THIS STEP
     SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM         SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM
      X           Y            Z           X           Y           Z           X           Y           Z           X           Y           Z

  7.879E+07  0.         -2.994E+07  7.879E+07  0.         -2.994E+07  0.          0.          0.          0.          0.          0.

                                 ***********************************************************

                                              INCREMENTAL STEP     2
                                 20 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                 COORDINATE DEFINITIONS                          EXTERNAL FORCES                          EXTERNAL MOMENTS
  NODAL       SURFACE COORDINATE SYSTEM                  SURFACE COORDINATE SYSTEM                  SURFACE COORDINATE SYSTEM
  POINT          X            Y            Z              X            Y            Z              X            Y            Z

    1       0.           2.35000E+01  0.          -1.56136E+08 -1.48900E+08  3.23397E+08  0.           0.           0.
    2       6.08000E+01  0.           1.95800E+01 -9.19345E+08  0.           3.38504E+08  0.           0.           0.
    3       0.          -2.35000E+01  0.          -1.56136E+08  1.48900E+08  3.23397E+08  0.           0.           0.
    4      -2.46420E+01  0.           5.10399E+01  1.23162E+09  0.          -9.85296E+06  0.           0.           0.

 STRUCTURAL    AXIAL        AXIAL        AXIAL                     SHEAR        SHEAR                      SHEAR        SHEAR
  ELEMENT      FORCE        STROKE       STIFFNESS                 FORCE        MOMENT                     FORCE        MOMENT

    1      -3.88761E+08 -7.07240E-03  5.49688E+10   NODE 1      0.           0.            NODE  4      0.           0.
    2      -9.79684E+08 -3.94715E-01  2.48200E+09   NODE 2      0.           0.            NODE  4      0.           0.
    3      -3.88761E+08 -7.07240E-03  5.49688E+10   NODE 3      0.           0.            NODE  4      0.           0.

                                                                          NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 1
                      TOTAL ENERGY ABSORBED                                   FOOTPAD ATTENUATION ENERGY THIS STEP
     SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM         SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM
      X           Y            Z           X           Y           Z           X           Y           Z           X           Y           Z

  1.714E+09  0.         -1.183E+08  1.714E+09  0.         -1.183E+08  1.400E+09  0.          0.          1.400E+09  0.          0.
```

```
                                        INCREMENTAL STEP    9
                               20 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                  COORDINATE DEFINITIONS                       EXTERNAL FORCES                         EXTERNAL MOMENTS
 NODAL           SURFACE COORDINATE SYSTEM                 SURFACE COORDINATE SYSTEM                 SURFACE COORDINATE SYSTEM
 POINT          X              Y              Z              X              Y              Z              X              Y              Z

   1       0.             2.35000E+01    0.            -1.40797E+08   -1.44713E+08    3.19360E+08    0.             0.             0.
   2       6.08000E+01    0.             1.95600E+01   -9.98270E+08    0.             3.85173E+08    0.             0.             0.
   3       0.            -2.35000E+01    0.            -1.40797E+08    1.44713E+08    3.19360E+08    0.             0.             0.
   4      -2.28640E+01    0.             5.18609E+01    1.27986E+09    0.            -1.02389E+09    0.             0.             0.

STRUCTURAL     AXIAL          AXIAL          AXIAL                       SHEAR          SHEAR                       SHEAR          SHEAR
 ELEMENT       FORCE          STROKE         STIFFNESS                   FORCE          MOMENT                      FORCE          MOMENT

   1       -3.77831E+08   -6.87358E-03    5.49686E+10    NODE 1      0.             0.             NODE  4      0.             0.
   2       -1.07000E+09   -1.76881E+00    0.             NODE 2      0.             0.             NODE  4      0.             0.
   3       -3.77831E+08   -6.87358E-03    5.49686E+10    NODE 3      0.             0.             NODE  4      0.             0.

                                                                     NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 1
                     TOTAL ENERGY ABSORBED                                 FOOTPAD ATTENUATION ENERGY THIS STEP
    SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM          SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM
    X           Y           Z           X           Y           Z           X           Y           Z           X           Y           Z

 4.031E+09   0.         -9.746E+08   4.031E+09   0.         -9.746E+08   0.          0.          0.          0.          0.          0.

                    ************************************************************
                                        INCREMENTAL STEP   10
                               20 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                  COORDINATE DEFINITIONS                       EXTERNAL FORCES                         EXTERNAL MOMENTS
 NODAL           SURFACE COORDINATE SYSTEM                 SURFACE COORDINATE SYSTEM                 SURFACE COORDINATE SYSTEM
 POINT          X              Y              Z              X              Y              Z              X              Y              Z

   1       0.             2.35000E+01    0.            -1.36988E+08   -1.42380E+08    3.14886E+08    0.             0.             0.
   2       6.08000E+01    0.             1.95800E+01   -9.97426E+08    0.             3.87351E+08    0.             0.             0.
   3       0.            -2.35000E+01    0.            -1.36988E+08    1.42380E+08    3.14886E+08    0.             0.             0.
   4      -2.26100E+01    0.             5.19723E+01    1.27140E+09    0.            -1.01712E+09    0.             0.             0.

STRUCTURAL     AXIAL          AXIAL          AXIAL                       SHEAR          SHEAR                       SHEAR          SHEAR
 ELEMENT       FORCE          STROKE         STIFFNESS                   FORCE          MOMENT                      FORCE          MOMENT

   1       -3.71741E+08   -6.76279E-03    5.49685E+10    NODE 1      0.             0.             NODE  4      0.             0.
   2       -1.07000E+09   -1.96548E+00    0.             NODE 2      0.             0.             NODE  4      0.             0.
   3       -3.71741E+08   -6.76279E-03    5.49685E+10    NODE 3      0.             0.             NODE  4      0.             0.

                                                                     NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 1
                     TOTAL ENERGY ABSORBED                                 FOOTPAD ATTENUATION ENERGY THIS STEP
    SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM          SURFACE COORDINATE SYSTEM          LANDER COORDINATE SYSTEM
    X           Y           Z           X           Y           Z           X           Y           Z           X           Y           Z

 4.355E+09   0.         -1.088E+09   4.355E+09   0.         -1.088E+09   0.          0.          0.          0.          0.          0.
```

```
                                                        INCREMENTAL STEP   59
                                         12 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                      COORDINATE DEFINITIONS                    EXTERNAL FORCES                        EXTERNAL MOMENTS
    NODAL           SURFACE COORDINATE SYSTEM              SURFACE COORDINATE SYSTEM              SURFACE COORDINATE SYSTEM
    POINT          X            Y            Z            X            Y            Z            X             Y            Z

      1      0.           2.35000E+01  0.          -5.29597E+07 -1.22447E+08  2.90537E+08  0.           0.           0.
      2      6.08000E+01  0.           1.95800E+01 -1.71052E+09  0.           8.72077E+08  0.           0.           0.
      3      0.          -2.35000E+01  0.          -5.29597E+07  1.22447E+08  2.90537E+08  0.           0.           0.
      4     -1.01648E+01  0.           5.57597E+01  1.81644E+09  0.          -1.45315E+09  0.           0.           0.

  STRUCTURAL    AXIAL        AXIAL        AXIAL                   SHEAR        SHEAR                     SHEAR        SHEAR
   ELEMENT      FORCE       STROKE      STIFFNESS                 FORCE        MOMENT                    FORCE        MOMENT

      1     -3.19702E+08 -5.81619E-03  5.49677E+10  NODE 1     0.           0.           NODE  4     0.           0.
      2     -1.92000E+09 -1.17898E+01  0.           NODE 2     0.           0.           NODE  4     0.           0.
      3     -3.19702E+08 -5.81619E-03  5.49677E+10  NODE 3     0.           0.           NODE  4     0.           0.

                                                                               NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 1
                        TOTAL ENERGY ABSORBED                                      FOOTPAD ATTENUATION ENERGY THIS STEP
      SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM        SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM
       X          Y          Z          X          Y          Z           X          Y          Z          X          Y          Z

    2.023E+10  0.       -4.873E+09  2.023E+10  0.       -4.873E+09  0.        0.        0.        0.        0.        0.

                                   ************************************************************

                                                        INCREMENTAL STEP   60
                                         12 ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION

                      COORDINATE DEFINITIONS                    EXTERNAL FORCES                        EXTERNAL MOMENTS
    NODAL           SURFACE COORDINATE SYSTEM              SURFACE COORDINATE SYSTEM              SURFACE COORDINATE SYSTEM
    POINT          X            Y            Z            X            Y            Z            X             Y            Z

      1      0.           2.35000E+01  0.          -5.08776E+07 -1.20648E+08  2.86503E+08  0.           0.           0.
      2      6.08000E+01  0.           1.95800E+01 -1.70880E+09  0.           8.75438E+08  0.           0.           0.
      3      0.          -2.35000E+01  0.          -5.08776E+07  1.20648E+08  2.86503E+08  0.           0.           0.
      4     -9.91008E+00  0.           5.58055E+01  1.81056E+09  0.          -1.44844E+09  0.           0.           0.

  STRUCTURAL    AXIAL        AXIAL        AXIAL                   SHEAR        SHEAR                     SHEAR        SHEAR
   ELEMENT      FORCE       STROKE      STIFFNESS                 FORCE        MOMENT                    FORCE        MOMENT

      1     -3.15006E+08 -5.73075E-03  5.49676E+10  NODE 1     0.           0.           NODE  4     0.           0.
      2     -1.92000E+09 -1.19952E+01  0.           NODE 2     0.           0.           NODE  4     0.           0.
      3     -3.15006E+08 -5.73075E-03  5.49676E+10  NODE 3     0.           0.           NODE  4     0.           0.

                                                                               NUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS = 1
                        TOTAL ENERGY ABSORBED                                      FOOTPAD ATTENUATION ENERGY THIS STEP
      SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM        SURFACE COORDINATE SYSTEM         LANDER COORDINATE SYSTEM
       X          Y          Z          X          Y          Z           X          Y          Z          X          Y          Z

    2.069E+10  0.       -4.940E+09  2.069E+10  0.       -4.940E+09  0.        0.        0.        0.        0.        0.
```

```
                         CPU TIME USAGE TABLE (SEC)

                                              TIME IN     TIME OUT     TOTAL
 INPUT AND INITIALIZATION OVERLAY              50.025      50.030        .01

 GEAR INPUT AND ANALYSIS OVERLAY               50.030      55.827       5.80
```

APPENDIX D

EXAMPLE OF INPUT AND OUTPUT

LANDING LOADS AND MOTIONS PROGRAM

```
                                          LANDING LOADS AND MOTIONS PROGRAM - LEGGED LANDER
                                       MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
                                           MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST

                                                             INPUT DATA

 NEW CARDS READ THIS RUN -     1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
                              21   22   23  200  200  200  300  300  300   24   25   26   27   28   29   30   31   32  400  400
                             400  400  400  400   33   34   35   36   37   38   39   40   41   42  500  500  500  600  600  600
                             700  700  700  600  800  800  900  900  900 1000 1000 1000 1100 1100 1100 1100 1100 1100 1200 1200
                            1200 1200 1200 1200 1300 1300 1300 1300 1300 1300 1700 1700 1700
 CARD
  NO

          * * PROGRAM CONTROL DATA * *

   1 CASE NO =    306
   2 TIMAX =  1.000E-01 IPTCKT = 10 INLEG =  1 IFPPRT =  0
   3 NMODES =  3 NOOUT =  0
   4 INDFXD =  0 INDFYD =  1 INDFZD =  0
   5 INDFXR =  1 INDFYR =  0 INDFZR =  1
   6 HMAX =  1.000E-04 HMIN =  0.         EMAX =  0.         EMIN =  0.         IF =  0
   7 IVARF =  1 IMTH =  0 CUTERR =  1.000E-04
   8 NFORC =  1 IQOOUT =  0 JCKSAB =  0 IOSETK =  0

          * * INITIAL CONDITIONS * *

   9 ZETA =  0.          GRAV =  9.807E+02 GRAVE =  9.807E+02
  10 ANGX =  0.          ANGY =  9.500E+00 ANGZ =  0.
  11 WX =  0.          WY =  0.          WZ =  0.
  12 VELX = -1.146E+03 VELY =  0.         VELZ =  0.

          * * CENTER BODY DATA * *

  13 CEMASS =  1.805E+05
  14 CBIXX =  8.031E+08 CBIYY =  4.992E+08 CBIZZ =  4.968E+08
  15 CBIXY =  0.          CBIXZ =  0.          CBIYZ =  0.

    NO SECONDARY OUTPUT POINTS

          * * FOOTPAD DATA * *

  16 FPMASS =  3.497E+03
  17 RAD(I) =     1.260E+01   1.364E+01   1.605E+01   1.605E+01
  18 SS(I) =      1.159E+01   1.032E+01   8.540E+00   5.870E+00
  19 ATTHCK(I) =     1.270E+00   3.970E+00   5.720E+00
  20 ATTFRS(I) =     2.210E+06   5.170E+06   5.170E+06

          * * SOIL DATA * *

  21 NTYPE =  1
     (SECONDARY SOIL MECHANICS)
  22 SOILF(1) =  9.000E-01 ,  5.420E+06 ,  1.470E+07
```

```
          * * LEG DATA * *

  23 NCLEG =  3 ILEG =  0 DRAGST =  0.
 200 I =  1 XFP = -5.648E+01 YFP =  0.         ZFP =  1.448E+02
 200 I =  2 XFP = -5.648E+01 YFP = -1.254E+02 ZFP = -7.239E+01
 200 I =  3 XFP = -5.648E+01 YFP =  1.254E+02 ZFP = -7.239E+01
 300 I =  1 XMSCB =  2.949E+01 YMSCB =  0.         ZMSCB =  1.135E+02
 300 I =  2 XMSCB =  2.949E+01 YMSCB = -9.832E+01 ZMSCB = -5.677E+01
 300 I =  3 XMSCB =  2.949E+01 YMSCB =  9.832E+01 ZMSCB = -5.677E+01
  24 PFCMS =     1.090E+09   1.540E+09   1.920E+09   3.340E+09   3.340E+09
  25 PFTMS =     5.000E+09   5.000E+09   5.000E+09   5.000E+09   5.000E+09
  26 CCCPS =     8.360E+00   1.135E+01   1.415E+01   2.800E+01   2.900E+01
  27 CLTMS =     2.000E+00   3.000E+00   4.000E+00   5.000E+00   6.000E+00
  28 SRCPS =     2.482E+09   8.480E+08   1.023E+09   1.198E+09   1.198E+09
  29 SRTPS =     2.500E+09   2.500E+09   2.500E+09   2.500E+09   2.500E+09
  30 SCMXMS =  2.794E+01 STMXMS =  2.000E+00 SRUCMS =  1.420E+10 SRCTPS =  2.500E+07
  31 IRETMS =  0
  32 FRICPS =  0.         COEFMS =  0.         GAMMS =  1.000E+00 AE11 =  0.         AE12 =  0.
 400 I =  1 XDSCB = -3.139E+01 YDSCB =  2.350E+01 ZDSCB =  9.398E+01
 400 I =  2 XDSCB = -3.139E+01 YDSCB = -2.350E+01 ZDSCB =  9.398E+01
 400 I =  3 XDSCB = -3.139E+01 YDSCB = -9.314E+01 ZDSCB = -2.665E+01
 400 I =  4 XDSCB = -3.139E+01 YDSCB = -6.965E+01 ZDSCB = -6.734E+01
 400 I =  5 XDSCB = -3.139E+01 YDSCB =  9.314E+01 ZDSCB = -2.665E+01
 400 I =  6 XDSCB = -3.139E+01 YDSCB =  6.965E+01 ZDSCB = -6.734E+01
  33 PFCDS =     5.500E+10   1.100E+11   1.650E+11   2.200E+11   2.750E+11
  34 PFTDS =     5.500E+10   1.100E+11   1.650E+11   2.200E+11   2.750E+11
  35 CDCCS =     1.000E+00   2.000E+00   3.000E+00   4.000E+00   5.000E+00
  36 CDTDS =     1.000E+00   2.000E+00   3.000E+00   4.000E+00   5.000E+00
  37 SRCDS =     5.500E+10   5.500E+10   5.500E+10   5.500E+10   5.500E+10
  38 SRTDS =     5.500E+10   5.500E+10   5.500E+10   5.500E+10   5.500E+10
  39 SCMXDS =  5.000E+00 STMXDS =  5.000E+00 SRUCDS =  5.500E+10 SRCTDS =  5.500E+10
  40 IRETDS =  0
  41 FRICDS =  0.         COLFDS =  0.         GAMDS =  1.000E+00


          * * MODAL DATA * *

  42 MODEIN =  3
 500 I =  1 GM =  8.405E+04 OMEGA =  3.459E+01
 500 I =  2 GM =  6.880E+04 OMEGA =  4.817E+01
 500 I =  3 GM =  5.846E+04 OMEGA =  1.010E+02
 600 I =  1 WNX =  8.404E+04 WNY =  6.436E+04 WNZ =  1.971E+04
 600 I =  2 WNX =  2.968E+04 WNY =  5.374E+04 WNZ =  5.417E+04
 600 I =  3 WNX =  5.831E+04 WNY =  3.694E+04 WNZ =  2.171E+04
 700 I =  1 FX =  2.456E+01 PY = -2.423E+06 PZ =  2.463E+06
 700 I =  2 FX =  4.677E+03 PY = -1.108E+05 PZ =  1.003E+05
 700 I =  3 FX =  1.852E+02 PY = -2.333E+06 PZ =  2.321E+06
 800 I =  1 FMSX =     -5.360E-02  -8.116E-01  -1.713E-02
 800 I =  2 FMSX =      2.690E-02   4.005E-01   0.
 800 I =  3 FMSX =      2.690E-02   4.005E-01   0.
 900 I =  1 FMSY =      0.          0.          0.
 900 I =  2 FMSY =      2.265E-01   5.182E-01   3.564E-01
 900 I =  3 FMSY =     -2.265E-01  -5.182E-01  -3.564E-01
1000 I =  1 FMSZ =      1.000E+00   5.621E-01   1.982E-01
1000 I =  2 FMSZ =      6.077E-01  -3.490E-01  -4.180E-01
1000 I =  3 FMSZ =      6.077E-01  -3.490E-01  -4.180E-01
1100 I =  1 FDSX =      0.         -3.800E-01   0.
1100 I =  2 FDSX =      0.         -3.800E-01   0.
1100 I =  3 FDSX =      0.          1.000E+00   5.540E-02
1100 I =  4 FDSX =      0.         -6.000E-01  -5.800E-02
1100 I =  5 FDSX =      0.          1.000E+00   5.540E-02
1100 I =  6 FDSX =      0.         -6.000E-01  -5.800E-02
1200 I =  1 FDSY =      0.          0.          0.
1200 I =  2 FDSY =      0.          0.          0.
1200 I =  3 FDSY =      7.650E-01  -7.574E-01   9.206E-01
```

```
1200 I =  4 FDSY =    0.           0.           1.850E-02
1200 I =  5 FDSY =   -7.650E-01   7.574E-01   -9.206E-01
1200 I =  6 FDSY =    0.           0.          -1.850E-02
1300 I =  1 FDSZ =    8.724E-01  -2.800E-01    1.880E-01
1300 I =  2 FDSZ =    8.724E-01  -2.800E-01    1.880E-01
1300 I =  3 FDSZ =    4.297E-01   1.638E-01   -3.660E-01
1300 I =  4 FDSZ =   -1.200E-02   5.900E-01   -8.847E-01
1300 I =  5 FDSZ =    4.297E-01   1.638E-01   -3.660E-01
1300 I =  6 FDSZ =   -1.200E-02   5.900E-01   -8.847E-01
1700 I =  1 FCGX =  0.           PCGY =  0.           PCGZ = -6.170E-01
1700 I =  2 FCGX =  3.500E-03 PCGY =  0.           PCGZ = -4.590E-02
1700 I =  3 FCGX =  0.           PCGY =  0.           PCGZ =  3.489E-01
```

```
            LANDER INITIAL CONDITIONS (SURFACE COORDINATE SYSTEM)

                                       X              Y           Z

    INITIAL C.G. POSITION             79.24          0.00        0.00

    INITIAL C.G. VELOCITY          -1146.00          0.00        0.00

    INITIAL ANGULAR VELOCITY           0.00          0.00        0.00


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

           TIME =  0.            STEP SIZE =  1.000000E-04

  CENTER BODY MOTIONS

    TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                              VELOCITY AND ACCELERATION
          DISP         VEL        ACCEL                                       LANDER COORDINATE SYSTEM
      X  7.924E+01 -1.146E+03 -9.807E+02          EULER ANGLES  PHI    0.           X   0              0.
      Y  0.         0.         0.                               THETA  9.500E+00    Y   0.             0.
      Z  0.         0.         0.                               PSI    0.           Z   0              0.

  FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                        X                                  Y                                 Z
                            DISP        VEL       ACCEL       DISP        VEL        ACCEL       DISP        VEL        ACCEL
    FOOTPAD  1           4.743E+01 -1.146E+03 -9.807E+02  0.            0.          0.         1.521E+02  0.          0.
    FOOTPAD  2           1.159E+01 -1.146E+03 -9.807E+02 -1.254E+02     0.          0.        -6.208E+01  0.          0.
    FOOTPAD  3           1.159E+01 -1.146E+03 -9.807E+02  1.254E+02     0.          0.        -6.208E+01  0.          0.

  ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                             X             Y          Z
    CENTER OF GRAVITY  -9.663E-01   0.          -1.650E-01

  LANDING GEAR STRUT INFORMATION
    LEG      CRUSH                 LOAD      STROKE                      LOAD      STROKE                      LOAD      STROKE
     1        0.        MAIN STRUT  0.          0.        DRAG STRUT 1  0.          0.        DRAG STRUT 2  0.          0.
     2        0.        MAIN STRUT  0.          0.        DRAG STRUT 3  0.          0.        DRAG STRUT 4  0.          0.
     3        0.        MAIN STRUT  0.          0.        DRAG STRUT 5  0.          0.        DRAG STRUT 6  0.          0.


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

           TIME =  1.000000E-03  STEP SIZE =  1.000000E-04

  CENTER BODY MOTIONS

    TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                              VELOCITY AND ACCELERATION
          DISP         VEL        ACCEL                                       LANDER COORDINATE SYSTEM
      X  7.810E+01 -1.142E+03  1.166E+04          EULER ANGLES  PHI    0.           X   0              0.
      Y  0.         0.         0.                               THETA  9.500E+00    Y  -1.0[illegible]E+00 -3.989E+03
      Z -1.730E-04 -7.048E-01 -1.779E+03                        PSI    0.           Z   0.             0.

  FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                        X                                  Y                                 Z
                            DISP        VEL       ACCEL       DISP        VEL        ACCEL       DISP        VEL        ACCEL
```

```
   FOOTPAD  1                 4.629E+01 -1.149E+03 -1.100E+04   0.          0.          0.          1.521E+02  6.064E+01  6.006E+04
   FOOTPAD  2                 1.052E+01 -1.023E+03 -1.047E+04  -1.254E+02 -6.722E+01 -1.994E+05  -6.208E+01 -3.645E+01 -1.063E+05
   FOOTPAD  3                 1.052E+01 -1.023E+03 -1.047E+04   1.254E+02  6.722E+01  1.994E+05  -6.208E+01 -3.645E+01 -1.063E+05

 ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                           X         Y          Z
   CENTER OF GRAVITY   1.203E+01  0.         1.735E-01

 LANDING GEAR STRUT INFORMATION
   LEG      CRUSH                 LOAD      STROKE                     LOAD      STROKE                    LOAD      STROKE
    1      0.         MAIN STRUT -2.072E+05 -6.347E-05  DRAG STRUT 1 -6.531E+07 -1.187E-03  DRAG STRUT 2 -6.531E+07 -1.187E-03
    2      6.632E-01  MAIN STRUT -1.631E+08 -6.571E-02  DRAG STRUT 3 -1.351E+09 -2.456E-02  DRAG STRUT 4 -1.029E+09 -1.672E-02
    3      6.632E-01  MAIN STRUT -1.631E+08 -6.571E-02  DRAG STRUT 5 -1.351E+09 -2.456E-02  DRAG STRUT 6 -1.029E+09 -1.672E-02


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

         TIME =  2.000000E-03  STEP SIZE =  1.000000E-04


 CENTER BODY MOTIONS

   TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                                 VELOCITY AND ACCELERATION
            DISP        VEL        ACCEL                                         LANDER COORDINATE SYSTEM
      X   7.696E+01 -1.126E+03  3.470E+04          EULER ANGLES  PHI    0.          X   0.           0.
      Y   0.         0.         0.                               THETA  9.496E+00   Y  -6.401E+00 -1.458E+04
      Z  -1.891E-03 -3.392E+03 -5.478E+03                        PSI    0.          Z   0.           0.

 FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                      X                                 Y                                Z
                           DISP        VEL        ACCEL       DISP        VEL        ACCEL       DISP        VEL        ACCEL
   FOOTPAD  1            4.513E+01 -1.156E+03 -1.564E+04   0.          0.          0.          1.521E+02  3.067E+01  5.510E+04
   FOOTPAD  2            9.606E+00 -6.020E+02 -5.779E+04  -1.254E+02 -1.191E+02 -5.660E+05  -6.212E+01 -6.122E+01 -2.852E+05
   FOOTPAD  3            9.606E+00 -6.020E+02 -5.779E+04   1.254E+02  1.191E+02  5.660E+05  -6.212E+01 -6.122E+01 -2.852E+05

 ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                           X         Y          Z
   CENTER OF GRAVITY   3.582E+01  0.         3.275E-01

 LANDING GEAR STRUT INFORMATION
   LEG      CRUSH                 LOAD      STROKE                     LOAD      STROKE                    LOAD      STROKE
    1      0.         MAIN STRUT  9.600E+06  3.840E-03  DRAG STRUT 1 -1.116E+08 -2.030E-03  DRAG STRUT 2 -1.116E+08 -2.030E-03
    2      1.030E+00  MAIN STRUT -6.178E+08 -2.489E-01  DRAG STRUT 3 -3.541E+09 -6.459E-02  DRAG STRUT 4 -2.965E+09 -5.391E-02
    3      1.030E+00  MAIN STRUT -6.178E+08 -2.489E-01  DRAG STRUT 5 -3.541E+09 -6.459E-02  DRAG STRUT 6 -2.965E+09 -5.391E-02


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

         TIME =  3.000000E-03  STEP SIZE =  1.000000E-04


 CENTER BODY MOTIONS

   TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                                 VELOCITY AND ACCELERATION
            DISP        VEL        ACCEL                                         LANDER COORDINATE SYSTEM
      X   7.585E+01 -1.100E+03  3.223E+04          EULER ANGLES  PHI    0.          X   0.           0.
      Y   0.         0.         0.                               THETA  9.480E+00   Y  -2.067E+01 -2.051E+04
      Z  -7.924E-03 -6.668E+00 -6.457E+03                        PSI    0.          Z   0.           0.
```

```
          TIME =  9.900000E-02  STEP SIZE =  1.000000E-04

CENTER BODY MOTIONS

   TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                               VELOCITY AND ACCELERATION
            DISP        VEL        ACCEL                                       LANDER COORDINATE SYSTEM
        X  4.730E+01  3.743E+01 -1.125E+03          EULER ANGLES  PHI     0.         X   0.          0.
        Y  0.         0.         0.                               THETA -1.847E+00   Y  -1.051E+01 -3.822E+02
        Z  1.015E+01 -2.069E+02  6.083E+03                        PSI     0.         Z   0.          0.

FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                             X                                Y                                 Z
                              DISP       VEL        ACCEL       DISP       VEL        ACCEL       DISP       VEL        ACCEL
   FOOTPAD  1              8.155E+00  2.912E-04 -5.823E+00  0.         0.         0.          1.598E+02  1.528E+02  9.933E+01
   FOOTPAD  2              8.398E+00  6.757E+00 -1.456E+03 -1.297E+02  1.051E+00 -1.417E+04 -6.575E+01  1.590E+02  3.090E+04
   FOOTPAD  3              8.398E+00  6.757E+00 -1.456E+03  1.297E+02 -1.051E+00  1.417E+04 -6.575E+01  1.590E+02  3.090E+04

ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                              X          Y          Z
   CENTER OF GRAVITY  -9.464E-01  0.         6.237E+00

LANDING GEAR STRUT INFORMATION
   LEG      CRUSH                  LOAD        STROKE                         LOAD       STROKE                           LOAD       STROKE
    1    2.634E+00  MAIN STRUT  0.          -1.770E+01  DRAG STRUT 1 -2.007E+06 -3.649E-05  DRAG STRUT 2 -2.007E+06 -3.649E-05
    2    2.339E+00  MAIN STRUT  0.          -1.193E+01  DRAG STRUT 3  1.604E+08  2.917E-03  DRAG STRUT 4 -1.486E+08 -2.702E-03
    3    2.339E+00  MAIN STRUT  0.          -1.193E+01  DRAG STRUT 5  1.604E+08  2.917E-03  DRAG STRUT 6 -1.486E+08 -2.702E-03


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

          TIME =  1.000000E-01  STEP SIZE =  1.000000E-04

CENTER BODY MOTIONS

   TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                               VELOCITY AND ACCELERATION
            DISP        VEL        ACCEL                                       LANDER COORDINATE SYSTEM
        X  4.734E+01  3.615E+01 -1.530E+03          EULER ANGLES  PHI     0.         X   0.          0.
        Y  0.         0.         0.                               THETA -1.859E+00   Y  -1.254E+01 -1.022E+03
        Z  9.948E+00 -1.942E+02  1.902E+04                        PSI     0.         Z   0.          0.

FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                             X                                Y                                 Z
                              DISP       VEL        ACCEL       DISP       VEL        ACCEL       DISP       VEL        ACCEL
   FOOTPAD  1              8.156E+00  9.707E-01 -2.779E+03  0.         0.         0.          1.599E+02  1.299E+02  2.140E+04
   FOOTPAD  2              8.403E+00  2.367E+00 -1.149E+03 -1.297E+02 -3.931E+01 -1.037E+04 -6.556E+01  2.103E+02  2.599E+04
   FOOTPAD  3              8.403E+00  2.367E+00 -1.149E+03  1.297E+02  3.931E+01  1.037E+04 -6.556E+01  2.103E+02  2.599E+04

ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                              X          Y          Z
   CENTER OF GRAVITY  -9.299E-01  0.         1.943E+01

LANDING GEAR STRUT INFORMATION
   LEG      CRUSH                  LOAD        STROKE                         LOAD       STROKE                           LOAD       STROKE
    1    2.634E+00  MAIN STRUT  0.          -1.775E+01  DRAG STRUT 1 -4.065E+07 -7.390E-04  DRAG STRUT 2 -4.065E+07 -7.390E-04
    2    2.339E+00  MAIN STRUT  0.          -1.188E+01  DRAG STRUT 3  1.339E+08  2.434E-03  DRAG STRUT 4 -1.191E+08 -2.165E-03
    3    2.339E+00  MAIN STRUT  0.          -1.188E+01  DRAG STRUT 5  1.339E+08  2.434E-03  DRAG STRUT 6 -1.191E+08 -2.165E-03
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

          TIME =  1.010000E-01  STEP SIZE =  1.000000E-04

CENTER BODY MOTIONS

   TRANSLATION - SURFACE COORDINATE SYSTEM          ROTATION
                                                                                        VELOCITY AND ACCELERATION
              DISP         VEL        ACCEL                                             LANDER COORDINATE SYSTEM
       X   4.737E+01  5.443E+01 -1.904E+03              EULER ANGLES  PHI    0.          X   0.          0.
       Y   0.         0.         0.                                   THETA -1.870E+00   Y  -1.026E+01  2.307E+03
       Z   9.765E+00 -1.698E+02  2.862E+04                            PSI    0.          Z   0.          0.

FOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM
                                               X                                  Y                                  Z
                               DISP         VEL        ACCEL        DISP         VEL        ACCEL        DISP         VEL        ACCEL
   FOOTPAD  1               8.157E+00  9.852E-01 -3.527E+03   0.          0.          0.          1.600E+02  1.180E+02  3.030E+04
   FOOTPAD  2               8.405E+00  2.509E+00  2.637E+03  -1.296E+02 -1.407E+01  5.068E+04  -6.537E+01  1.597E+02 -7.723E+04
   FOOTPAD  3               8.405E+00  2.509E+00  2.637E+03   1.296E+02  1.407E+01 -5.068E+04  -6.537E+01  1.597E+02 -7.723E+04

ACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEM
                                X          Y           Z
   CENTER OF GRAVITY  -9.882E-01  0.          2.924E+01

LANDING GEAR STRUT INFORMATION
   LEG      CRUSH                       LOAD       STROKE                       LOAD       STROKE                       LOAD       STROKE
    1     2.634E+00  MAIN STRUT  0.          -1.779E+01  DRAG STRUT 1 -5.756E+07 -1.046E-03  DRAG STRUT 2 -5.756E+07 -1.046E-03
    2     2.339E+00  MAIN STRUT  0.          -1.183E+01  DRAG STRUT 3 -4.091E+08 -7.439E-03  DRAG STRUT 4  4.327E+08  7.868E-03
    3     2.339E+00  MAIN STRUT  0.          -1.183E+01  DRAG STRUT 5 -4.091E+08 -7.439E-03  DRAG STRUT 6  4.327E+08  7.868E-03

SUMMARY INFORMATION

MAXIMUM MAIN STRUT STROKES
                   COMPRESSION                    TENSION
   STRUT      TIME        STROKE           TIME        STROKE
     1     5.720E-02  -1.695E+01       1.350E-02   7.136E-02
     2     3.520E-02  -1.368E+01       0.          0.
     3     3.520E-02  -1.368E+01       0.          0.
MAXIMUM DRAG STRUT STROKES
                   COMPRESSION                    TENSION
   STRUT      TIME        STROKE           TIME        STROKE
     1     3.070E-02  -6.536E-02       3.260E-02   4.593E-02
     2     3.070E-02  -6.536E-02       3.260E-02   4.593E-02
     3     2.000E-03  -6.439E-02       5.990E-02   3.005E-02
     4     4.400E-03  -6.078E-02       3.710E-02   3.855E-02
     5     2.000E-03  -6.439E-02       5.990E-02   3.005E-02
     6     4.400E-03  -6.078E-02       3.710E-02   3.855E-02

   END OF CASE - TIME LIMIT

      CASE NO          366 RAN FOR  58.524 CP SECONDS
```

APPENDIX E

OPERATING INSTRUCTIONS

CENTER BODY LANDING LOADS PROGRAM

E.1 Introduction - The Center Body Landing Loads Program retrieves and lists, for specific points in time, the information placed on magnetic tape by the Landing Loads and Motions Program. Information continuously stored on tape consists of landing gear strut forces acting on the center body; angular positions, velocities, and accelerations and the translational accelerations of the center body center of gravity; and the displacements, velocities, and accelerations of the normal modes employed in representing the flexible center body structure. The Center Body Landing Loads Program uses the information on this tape, in addition to other required input data, to obtain the distribution of landing gear strut loads and inertia loads throughout the center body structure for the selected points in time. This load information may then be used in conjunction with the Center Body Option of the Structural Analysis Program to determine the internal member load distribution for the time point of interest.

E.2 Input Data - A number of input data cards are required by the Center Body Landing Loads Program. These properly initialize the program, define the mass properties of the center body, and define the points in time during the landing time history at which the loads are to be determined. The required format for these cards is shown in Figure E-1 while the input variables are defined in Figure E-2. Also indicated in Figure E-2 is whether the various input quantities must be input as integer or floating point numbers. All integer numbers and floating point numbers input with an E format must be right justified. Floating point numbers input with an F format may appear anywhere inside their allotted field on the data card.

In addition to the required card input, two magnetic tapes are required. The mode shape data and joint coordinate information is required on TAPE4. This tape is generated when performing the modal analysis on the center body structure with the Structural Analysis Program. Secondly, the time history data obtained from the Landing Loads and Motions Program must be available on TAPE3. Since the time history data on this tape corresponds to points in time when normal printed output was obtained from the Landing Loads and Motions Program, only center body loads at these points in time may be obtained with the Center Body Landing Loads Program.

COLUMN: 1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 45	46 50
NOJTS	NOLEG	NOMODE	NOTIME	NOSETS					
MODE(1)	MODE(2)	MODE(3)	MODE(4)	MODE(5)					
JOTMS(1)	JOTMS(2)	JOTMS(3)	JOTMS(4)	JOTMS(5)					
JOTDS(1)	JOTDS(2)	JOTDS(3)	JOTDS(4)	JOTDS(5)	JOTDS(6)	JOTDS(7)	JOTDS(8)	JOTDS(9)	JOTDS(10)
ZETA		GRAV							
TYM(I)		TYM(I+1)		TYM(I+2)		TYM(I+3)		TYM(I+4)	
AMASS(I)		AMASS(I+1)		AMASS(I+2)		AMASS(I+3)		AMASS(I+4)	

FIGURE E-1 INPUT DATA FORMAT – CENTER BODY LANDING LOADS PROGRAM

INPUT VARIABLE	TYPE NUMBER*	VARIABLE DEFINITION
NOJTS	I	NUMBER OF JOINTS IN CENTER BODY STRUCTURAL IDEALIZATION (74 MAXIMUM)
NOLEG	I	NUMBER OF LEGS (5 MAXIMUM)
NOMODE	I	NUMBER OF MODES INCLUDED IN ANALYSIS (5 MAXIMUM)
NOTIME	I	NUMBER OF TIME POINTS TO BE READ FROM SECONDARY TIME HISTORY TAPE (TAPE3) OBTAINED FROM LANDING LOADS AND MOTIONS PROGRAM (20 MAXIMUM)
NOSETS	I	NUMBER OF NORMAL MODE SETS APPEARING ON TAPE (TAPE4) OBTAINED DURING NORMAL MODE ANALYSIS WITH THE STRUCTURAL ANALYSIS PROGRAM
MODE(I)	I	MODE NUMBER OF I TH NORMAL MODE OBTAINED FROM THE STRUCTURAL ANALYSIS PROGRAM (5 MAXIMUM)
JOTMS(I)	I	JOINT NUMBER LOCATION OF I TH MAIN STRUT (5 MAXIMUM)
JOTDS(I)	I	JOINT NUMBER LOCATION OF I TH DRAG STRUT (10 MAXIMUM)
ZETA	F	GROUND SLOPE
GRAV	F	LOCAL ACCELERATION OF GRAVITY
TYM(I)	F	TIMES, IN ASCENDING ORDER, AT WHICH CENTER BODY LOADS ARE TO BE DETERMINED (20 MAXIMUM, 5 TO A DATA CARD)
AMASS(I)	F	MASS ASSOCIATED WITH I TH STRUCTURAL JOINT. (74 MAXIMUM, 5 TO A DATA CARD)

*I – INTEGER NUMBER; F – FLOATING POINT NUMBER

FIGURE E–2 INPUT DATA – CENTER BODY LANDING LOADS PROGRAM

The use of the input indicators NOSETS and MODE(I) require some additional comments. The modal analysis output obtained from the Structural Analysis Program is placed on magnetic tape in data sets each containing information for five modes. The first set contains the lowest five elastic modes, the second set contains the next five modes, etc. The input indicator NOSETS defines the number of these modal data sets that appear on TAPE4. The modes in the first data set are numbered 1 through 5, those in the second data set 6 through 10, etc. MODE(I) defines the mode number of the Ith mode which is required by the Center Body Landing Loads Program.

E.3 Output Data - For a selected point in time the landing loads at each of the joints used in the center body structural idealization are printed. These loads are the combination of the inertia loads, gravity loads, and landing gear strut loads. The components of the landing loads are given in the Lander Coordinate System and thus may be directly input to the Center Body Option of the Structural Analysis Program. In addition, the sum of forces and moments about the center of gravity are given for information purposes.

E.4 Program Operation - An example of the output from the Center Body Landing Loads Program is given on the following pages. This information was required during the analysis of the Task Order Three lander, Section 6.2.3, at a time of 0.011 seconds. This time was selected for determining internal loads distribution in the lander center body using the Center Body Option of the Structural Analysis Program as discussed in Section 6.2.4.

A listing of the Center Body Landing Loads Program follows the example output.

```
                                   CENTER BODY LANDING LOADS PROGRAM
                          MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER FIVE
                             MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST

        NOJTS  =        33
        NCLEG  =         3
        NCMODE =         3
        NOTIME =         1
        NCSETS =         4
        MODE   =         1          3          8
        JCTMS  =         5         15         25
        JOTCS  =         2          7         12         17         27         22
        ZETA   =  0.
        GRAV   =  9.807E+02
        TYM    =  1.100E-02
        MASS   =  3.789E+03 6.099E+03 3.549E+03 2.490E+03 3.483E+03 3.789E+03 6.099E+03 3.533E+03 1.679E+03 1.736E+03
                  3.789E+03 6.099E+03 3.549E+03 2.127E+03 3.483E+03 3.789E+03 6.099E+03 3.612E+03 2.046E+03 1.736E+03
                  3.789E+03 6.099E+03 3.549E+03 2.127E+03 3.483E+03 3.789E+03 6.099E+03 3.533E+03 1.679E+03 1.736E+03
                  3.487E+04 3.270E+04 3.578E+03

            JOINT COORDINATES

    JOINT        X           Y           Z
       1     1.359E+01   3.172E+01   8.578E+01
       2    -2.807E+01   3.172E+01   8.578E+01
       3     1.359E+01   0.          8.578E+01
       4    -2.807E+01   0.          8.578E+01
       5     2.949E+01   0.          1.135E+02
       6     1.359E+01  -3.172E+01   8.578E+01
       7    -2.807E+01  -3.172E+01   8.578E+01
       8     1.359E+01  -6.093E+01   3.518E+01
       9    -1.791E+01  -6.093E+01   3.518E+01
      10    -2.807E+01  -6.093E+01   3.518E+01
      11     1.359E+01  -9.014E+01  -1.542E+01
      12    -2.807E+01  -9.014E+01  -1.542E+01
      13     1.359E+01  -7.430E+01  -4.288E+01
      14    -2.807E+01  -7.430E+01  -4.288E+01
      15     2.949E+01  -9.832E+01  -5.677E+01
      16     1.359E+01  -5.842E+01  -7.036E+01
      17    -2.807E+01  -5.842E+01  -7.036E+01
      18     1.359E+01   0.         -7.036E+01
      19    -1.791E+01   0.         -7.036E+01
      20    -2.807E+01   0.         -7.036E+01
      21     1.359E+01   5.842E+01  -7.036E+01
      22    -2.807E+01   5.842E+01  -7.036E+01
      23     1.359E+01   7.430E+01  -4.288E+01
      24    -2.807E+01   7.430E+01  -4.288E+01
      25     2.949E+01   9.832E+01  -5.677E+01
      26     1.359E+01   9.014E+01  -1.542E+01
      27    -2.807E+01   9.014E+01  -1.542E+01
      28     1.359E+01   6.093E+01   3.518E+01
      29    -1.791E+01   6.093E+01   3.518E+01
      30    -2.807E+01   6.093E+01   3.518E+01
      31     1.359E+01   0.          0.
      32    -1.791E+01   0.          0.
      33     0.          0.          0.

            MODE SHAPES

    MODE =  1
```

```
     JCINT      X            Y            Z
       1   -7.646E-04   -3.097E-04    8.734E-01
       2   -7.585E-04   -4.304E-04    8.724E-01
       3   -2.592E-04    4.316E-07    9.676E-01
       4   -7.777E-04    5.368E-07    8.973E-01
       5   -5.363E-02   -2.449E-05    1.000E+00
       6   -7.646E-04    3.105E-04    8.733E-01
       7   -7.586E-04    4.315E-04    8.723E-01
       8   -1.597E-03    6.427E-01    5.000E-01
       9   -1.561E-03    6.428E-01    4.996E-01
      10   -1.578E-03    6.400E-01    5.016E-01
      11   -2.957E-03    7.601E-01    4.329E-01
      12   -2.944E-03    7.650E-01    4.297E-01
      13    1.105E-04    4.217E-01    2.375E-01
      14    3.700E-04    3.938E-01    2.152E-01
      15    2.690E-02    2.265E-01    6.077E-01
      16    3.688E-03    1.453E-03   -5.483E-03
      17    3.670E-03    9.660E-04   -1.197E-02
      18    3.149E-03   -4.452E-05   -6.133E-01
      19    3.075E-03   -4.447E-05   -6.139E-01
      20    3.110E-03   -4.448E-05   -6.070E-01
      21    3.688E-03   -1.542E-03   -5.436E-03
      22    3.670E-03   -1.055E-03   -1.192E-02
      23    1.105E-04   -4.218E-01    2.376E-01
      24    3.701E-04   -3.939E-01    2.153E-01
      25    2.691E-02   -2.266E-01    6.077E-01
      26   -2.956E-03   -7.601E-01    4.330E-01
      27   -2.944E-03   -7.650E-01    4.297E-01
      28   -1.597E-03   -6.427E-01    5.001E-01
      29   -1.561E-03   -6.426E-01    4.996E-01
      30   -1.578E-03   -6.400E-01    5.017E-01
      31   -2.033E-05    3.199E-05   -6.164E-01
      32   -2.024E-05    3.201E-05   -6.178E-01
      33   -2.029E-05    3.200E-05   -6.170E-01

     MODE =  3

     JCINT      X            Y            Z
       1   -3.986E-01    9.606E-03    2.972E-01
       2   -3.991E-01   -9.616E-03   -2.952E-01
       3   -3.863E-01    8.568E-03    3.151E-01
       4   -3.873E-01   -8.993E-03   -2.925E-01
       5   -8.116E-01    9.796E-03    5.621E-01
       6   -3.706E-01    7.421E-03    2.827E-01
       7   -3.711E-01   -8.352E-03   -2.800E-01
       8    3.227E-01    2.014E-01    1.648E-01
       9    3.229E-01   -1.201E-01   -7.311E-02
      10    3.227E-01   -2.356E-01   -1.437E-01
      11    9.998E-01    7.294E-01   -1.466E-01
      12    1.000E+00   -7.574E-01    1.636E-01
      13    1.934E-01    3.797E-01   -3.497E-01
      14    1.939E-01   -3.838E-01    3.797E-01
      15    4.005E-01    5.182E-01   -3.490E-01
      16   -6.157E-01    6.478E-03   -5.631E-01
      17   -6.154E-01   -5.364E-03    5.972E-01
      18   -6.196E-01   -4.204E-03   -1.795E-01
      19   -6.201E-01    2.185E-03    1.317E-01
      20   -6.196E-01    4.256E-03    2.581E-01
      21   -5.918E-01   -1.487E-02   -5.484E-01
      22   -5.916E-01    1.363E-02    5.820E-01
      23    2.035E-01   -3.824E-01   -3.384E-01
      24    2.041E-01    3.861E-01    3.681E-01
      25    4.217E-01   -5.256E-01   -3.335E-01
      26    9.961E-01   -7.257E-01   -1.394E-01
```

```
     27    9.963E-01   7.529E-01   1.559E-01
     28    3.065E-01  -1.995E-01   1.708E-01
     29    3.067E-01   1.163E-01  -7.682E-02
     30    3.065E-01   2.320E-01  -1.508E-01
     31    3.487E-03   2.529E-03  -1.674E-01
     32    3.487E-03  -1.777E-03   1.185E-01
     33    3.487E-03   6.699E-04  -4.391E-02

     MODE =   8

   JOINT       X           Y           Z
      1    3.304E-03   3.188E-03   2.097E-01
      2    3.194E-03   1.925E-03   1.880E-01
      3    3.960E-03   1.769E-05   1.829E-01
      4    3.053E-03   1.456E-05   1.688E-01
      5   -1.710E-02   1.160E-05   1.982E-01
      6    3.308E-03  -3.153E-03   2.097E-01
      7    3.198E-03  -1.896E-03   1.880E-01
      8    2.953E-02  -4.355E-02   2.211E-01
      9    2.920E-02  -5.996E-02   2.168E-01
     10    2.930E-02  -3.485E-02   1.972E-01
     11    5.572E-02   1.000E+00  -3.972E-01
     12    5.538E-02   9.206E-01  -3.660E-01
     13   -1.682E-03   5.023E-01  -6.862E-01
     14   -1.222E-03   4.614E-01  -6.305E-01
     15    9.217E-03   3.564E-01  -4.180E-01
     16   -5.849E-02   2.405E-02  -9.571E-01
     17   -5.804E-02   1.849E-02  -8.847E-01
     18   -5.659E-02  -3.830E-06   2.969E-01
     19   -5.793E-02  -3.325E-06   3.214E-01
     20   -5.813E-02  -3.144E-06   2.584E-01
     21   -5.848E-02  -2.406E-02  -9.571E-01
     22   -5.804E-02  -1.850E-02  -8.847E-01
     23   -1.680E-03  -5.023E-01  -6.862E-01
     24   -1.221E-03  -4.614E-01  -6.305E-01
     25    9.214E-03  -3.563E-01  -4.180E-01
     26    5.572E-02  -1.000E+00  -3.972E-01
     27    5.538E-02  -9.206E-01  -3.660E-01
     28    2.953E-02   4.354E-02   2.211E-01
     29    2.920E-02   5.997E-02   2.168E-01
     30    2.930E-02   3.484E-02   1.972E-01
     31    2.840E-04  -5.769E-06   3.369E-01
     32    2.838E-04  -6.462E-06   3.655E-01
     33    2.840E-04  -6.062E-06   3.489E-01
```

```
        TIME HISTORY DATA

        TIME =  1.100E-02
    XSDD =  1.009E+04 YSDD =  0.         ZSDD =  6.134E+03
    PHI  =  0.        THIA =  1.539E-01 FS1  =  0.
    WX   =  0.        WY   = -2.179E+00 WZ   =  0.
    WXD  =  0.        WYD  = -1.941E+02 WZD  =  0.
    MODE =     1      GC   =  1.117E+00 GCD  =  7.398E+01 GCDD = -3.051E+04
    MODE =     2      GC   =  2.523E-01 GCD  =  1.757E+01 GCDD = [illegible]
    MODE =     3      GC   = -2.004E-02 GCD  =  1.636E+01 GCDD =  1.655E+04

        MAIN STRUT LOADS

    I    =     1      FMSX = -1.679E+07 FMSY =  0.         FMSZ =  5.982E+06
    I    =     2      FMSX =  9.986E+08 FMSY =  3.836E+08 FMSZ =  2.192E+08
    I    =     3      FMSX =  9.986E+08 FMSY = -3.836E+08 FMSZ =  2.192E+08

        DRAG STRUT LOADS

    I    =     1      FDSX = -1.371E+08 FDSY = -1.266E+08 FDSZ =  2.727E+08
    I    =     2      FDSX = -1.371E+08 FDSY =  1.266E+08 FDSZ =  2.727E+08
    I    =     3      FDSX =  1.694E+08 FDSY =  3.372E+08 FDSZ =  4.399E+05
    I    =     4      FDSX = -1.480E+08 FDSY = -4.944E+08 FDSZ = -4.621E+07
    I    =     5      FDSX =  1.694E+08 FDSY = -3.372E+08 FDSZ =  4.399E+08
    I    =     6      FDSX = -1.480E+08 FDSY =  4.944E+08 FDSZ = -4.621E+07

        CENTER BODY LANDING LOADS DISTRIBUTION

 JOINT       X           Y           Z
     1   1.801E+07  -7.227E+03   4.612E+07
     2  -1.098E+06  -1.273E+08   3.736E+08
     3   1.737E+07   2.084E+05   5.559E+07
     4   1.139E+07  -1.549E+05   4.403E+07
     5   4.910E+06   2.316E+05   5.892E+07
     6   1.873E+07   4.510E+05   4.572E+07
     7  -1.086E+08   1.265E+08   3.742E+08
     8  -2.922E+06   7.703E+07  -2.389E+06
     9  -1.675E+06   3.736E+07   7.265E+06
    10  -1.638E+06   3.220E+07   1.003E+07
    11  -2.505E+07   3.670E+07   2.372E+07
    12   1.282E+08   3.436E+08   5.363E+08
    13  -5.850E+07   2.191E+07   1.500E+07
    14  -3.531E+07   1.743E+06   3.324E+07
    15   9.399E+08   3.971E+08   2.349E+06
    16  -1.000E+08  -1.351E+06   1.254E+06
    17  -3.094E+06  -4.965E+08   4.452E+07
    18  -1.017E+06  -1.153E+05  -1.479E+08
    19  -6.288E+07   3.229E+04  -7.277E+07
    20  -4.657E+07   4.866E+04  -4.652E+07
    21  -9.941E+07   1.123E+06   1.633E+06
    22  -3.084E+08   4.966E+08   4.387E+07
    23  -5.825E+07  -2.199E+07   1.526E+07
    24  -3.516E+07  -1.716E+06   3.307E+07
    25   9.404E+08  -3.973E+08   2.352E+08
    26  -2.514E+07  -3.660E+07   2.393E+07
    27   1.280E+06  -3.436E+08   5.360E+08
    28  -3.313E+06  -7.698E+07  -2.225E+06
    29  -2.086E+06  -3.740E+07   7.221E+06
    30  -2.033E+06  -3.224E+07   9.952E+06
    31  -3.454E+08   6.454E+05  -1.370E+09
    32  -3.276E+08  -3.646E+05  -1.040E+09
    33  -3.562E+07   2.041E+04  -1.290E+08
```

```
          LANDING LOADS SUMMED AT C.G.

FX = -2.277E+04
FY = -9.633E+03
FZ = -2.316E+04
TX =  9.188E+05
TY = -2.957E+09
TZ =  2.758E+06
```

```
      PROGRAM CBLL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,TAPE4)
C
C     CENTER BODY LANDING LOADS PROGRAM
C     MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
C     MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST
C
C     THIS PROGRAM COMBINES THE EFFECTS OF THE LANDING GEAR STRUT LOADS
C     AND INERTIA LOADS AND OUTPUTS THESE FOR THE STRUCTURAL ANALYSIS
C     PROGRAM
C
C     THE FOLLOWING TAPE ARE REQUIRED -
C         1. TAPE3 - TIME HISTORY INFORMATION FROM THE LANDING LOADS
C                    AND MOTIONS PROGRAM
C         2. TAPE4 - MODAL DATA FROM THE STRUCTURAL ANALYSIS PROGRAM
C
      DIMENSION DUMMY(75),XJT(80),YJT(80),ZJT(80),PHIX(80,30),
     *          PHIY(80,30),PHIZ(80,30),AMASS(80),XPHI(80,5),
     *          YPHI(80,5),ZPHI(80,5),MODE(5),JOTMS(5),JOTDS(10),
     *          TYM(20),GC(5),GCD(5),GCDD(5),FMSX(5),FMSY(5),FMSZ(5),
     *          FDSX(10),FDSY(10),FDSZ(10),DC(3,3),FX(80),FY(80),FZ(80),
     *          FXJT(80),FYJT(80),FZJT(80),
     *          SAX(80),SAY(80),SAZ(80)
C
C     PROGRAM INITIALIZATION DATA
C
      READ(5,500)NOJTS,NOLEG,NOMODE,NOTIME,NOSETS
      READ(5,500)(MODE(I),I=1,NOMODE)
      READ(5,500)(JOTMS(I),I=1,NOLEG)
      NODS=2*NOLEG
      READ(5,500)(JOTDS(I),I=1,NODS)
      READ(5,501)ZETA,GRAV
      READ(5,501)(TYM(I),I=1,NOTIME)
      READ(5,501)(AMASS(I),I=1,NOJTS)
C
C     PRINT PROGRAM INITIALIZATION DATA
C
      WRITE(6,600)
      WRITE(6,601)NOJTS,NOLEG,NOMODE,NOTIME,NOSETS
      WRITE(6,602)(MODE(I),I=1,NOMODE)
      WRITE(6,603)(JOTMS(I),I=1,NOLEG)
      WRITE(6,604)(JOTDS(I),I=1,NODS)
      WRITE(6,605)ZETA,GRAV
      WRITE(6,607)(TYM(I),I=1,NOTIME)
      WRITE(6,606)(AMASS(I),I=1,NOJTS)
C
C     READ MODAL DATA FROM TAPE4
C
      REWIND 4
      I1=0
      I2=0
      I3=0
      I4=0
      I5=0
  150 CONTINUE
      READ(4,400)MMM,(DUMMY(I),I=1,5)
```

```
  155 CONTINUE                                                          CBLL 560
      IF(MMM.EQ.21)GO TO 150                                            CBLL 570
      IF(MMM.EQ.121)GO TO 150                                           CBLL 580
      IF(MMM.EQ.221)GO TO 160                                           CBLL 590
      IF(MMM.EQ.321)GO TO 165                                           CBLL 600
      IF(MMM.EQ.421)GO TO 156                                           CBLL 610
      IF(MMM.EQ.521)GO TO 150                                           CBLL 620
      IF(MMM.EQ.721)GO TO 150                                           CBLL 630
      WRITE(6,404)                                                      CBLL 640
      STOP                                                              CBLL 650
  156 CONTINUE                                                          CBLL 660
      I3=0                                                              CBLL 670
      I4=I4+5                                                           CBLL 680
      I5=I5+1                                                           CBLL 690
      IF(I5.EQ.NOSETS)GO TO 200                                         CBLL 700
      CALL SKFILE(4,1)                                                  CBLL 710
      GO TO 150                                                         CBLL 720
C                                                                       CBLL 730
C     JOINT COORDINATES                                                 CBLL 740
C                                                                       CBLL 750
  160 CONTINUE                                                          CBLL 760
      IF(I5.NE.0)GO TO 150                                              CBLL 770
      I1=I1+1                                                           CBLL 780
      XJT(I1)=DUMMY(3)                                                  CBLL 790
      YJT(I1)=DUMMY(1)                                                  CBLL 800
      ZJT(I1)=DUMMY(2)                                                  CBLL 810
      GO TO 150                                                         CBLL 820
C                                                                       CBLL 830
C     MODE SHAPES                                                       CBLL 840
C                                                                       CBLL 850
  165 CONTINUE                                                          CBLL 860
      I2=I2+1                                                           CBLL 870
      GO TO(166,168,170)I2                                              CBLL 880
  166 I3=I3+1                                                           CBLL 890
      DO 167 I=1,5                                                      CBLL 900
  167 PHIX(I3,I4+I)=DUMMY(I)                                            CBLL 910
      GO TO 150                                                         CBLL 920
  168 DO 169 I=1,5                                                      CBLL 930
  169 PHIY(I3,I4+I)=DUMMY(I)                                            CBLL 940
      GO TO 150                                                         CBLL 950
  170 DO 171 I=1,5                                                      CBLL 960
  171 PHIZ(I3,I4+I)=DUMMY(I)                                            CBLL 970
      I2=0                                                              CBLL 980
      GO TO 150                                                         CBLL 990
C                                                                       CBLL1000
C     SORT MODE SHAPE DATA                                              CBLL1010
C                                                                       CBLL1020
  200 CONTINUE                                                          CBLL1030
      NNN=5*NOSETS                                                      CBLL1040
      DO 215 I=1,NNN                                                    CBLL1050
      K=10                                                              CBLL1060
      IF(I.EQ.MODE(1))K=1                                               CBLL1070
      IF(I.EQ.MODE(2))K=2                                               CBLL1080
      IF(I.EQ.MODE(3))K=3                                               CBLL1090
      IF(I.EQ.MODE(4))K=4                                               CBLL1100
      IF(I.EQ.MODE(5))K=5                                               CBLL1110
```

```
      IF(K.EQ.10)GO TO 215                                              CBLL1120
      DO 215 J=1,NOJTS                                                  CBLL1130
      XPHI(J,K)=PHIX(J,I)                                               CBLL1140
      YPHI(J,K)=PHIY(J,I)                                               CBLL1150
      ZPHI(J,K)=PHIZ(J,I)                                               CBLL1160
  215 CONTINUE                                                          CBLL1170
C                                                                       CBLL1180
C     PRINT DATA READ FROM TAPE4                                        CBLL1190
C                                                                       CBLL1200
      WRITE(6,401)                                                      CBLL1210
      DO 220 I=1,NOJTS                                                  CBLL1220
  220 WRITE(6,402)I,XJT(I),YJT(I),ZJT(I)                                CBLL1230
      WRITE(6,403)                                                      CBLL1240
      DO 225 J=1,NOMODE                                                 CBLL1250
      WRITE(6,407)MODE(J)                                               CBLL1260
      DO 225 I=1,NOJTS                                                  CBLL1270
      WRITE(6,402)I,XPHI(I,J),YPHI(I,J),ZPHI(I,J)                       CBLL1280
  225 CONTINUE                                                          CBLL1290
C                                                                       CBLL1300
C     DETERMINE C. G. LOCATION                                          CBLL1310
C                                                                       CBLL1320
      TOTM=0.0                                                          CBLL1330
      XTOT=0.0                                                          CBLL1340
      YTOT=0.0                                                          CBLL1350
      ZTOT=0.0                                                          CBLL1360
      DO 230 J=1,NOJTS                                                  CBLL1370
      TOTM=TOTM+AMASS(J)                                                CBLL1380
      XTOT=XTOT+AMASS(J)*XJT(J)                                         CBLL1390
      YTOT=YTOT+AMASS(J)*YJT(J)                                         CBLL1400
  230 ZTOT=ZTOT+AMASS(J)*ZJT(J)                                         CBLL1410
      XCG=XTOT/TOTM                                                     CBLL1420
      YCG=YTOT/TOTM                                                     CBLL1430
      ZCG=ZTOT/TOTM                                                     CBLL1440
      DO 235 J=1,NOJTS                                                  CBLL1450
      XJT(J)=XJT(J)-XCG                                                 CBLL1460
      YJT(J)=YJT(J)-YCG                                                 CBLL1470
  235 ZJT(J)=ZJT(J)-ZCG                                                 CBLL1480
C                                                                       CBLL1490
C     READ TIME HISTORY DATA FROM TAPE3                                 CBLL1500
C                                                                       CBLL1510
      REWIND 3                                                          CBLL1520
      NOREAD=14+9*NOLEG+3*NOMODE                                        CBLL1530
      IJ=0                                                              CBLL1540
  700 CONTINUE                                                          CBLL1550
      IJ=IJ+1                                                           CBLL1560
      IF(IJ.GT.NOTIME)STOP                                              CBLL1570
  710 READ(3)(DUMMY(I),I=1,NOREAD)                                      CBLL1580
      IF(ABS(DUMMY(1)-TYM(IJ)).GT.(DUMMY(2)/10.)) GO TO 710             CBLL1590
C                                                                       CBLL1600
C     TIME POINT HAS BEEN FOUND                                         CBLL1610
C                                                                       CBLL1620
      TYM(IJ)=DUMMY(1)                                                  CBLL1630
      XSDD = DUMMY(3)                                                   CBLL1640
      YSDD = DUMMY(4)                                                   CBLL1650
      ZSDD =DUMMY(5)                                                    CBLL1660
      PHI  =DUMMY(6)                                                    CBLL1670
```

```
      THTA =DUMMY(7)                                                    CBLL1680
      PSI  =DUMMY(8)                                                    CBLL1690
      WX   =DUMMY(9)                                                    CBLL1700
      WY   =DUMMY(10)                                                   CBLL1710
      WZ   =DUMMY(11)                                                   CBLL1720
      WXD  =DUMMY(12)                                                   CBLL1730
      WYD  =DUMMY(13)                                                   CBLL1740
      WZD  =DUMMY(14)                                                   CBLL1750
      II=14                                                             CBLL1760
      DO 701 I=1,NOMODE                                                 CBLL1770
      I1=II+1+3*(I-1)                                                   CBLL1780
      GC(I)=DUMMY(I1)                                                   CBLL1790
      GCD(I)=DUMMY(I1+1)                                                CBLL1800
  701 GCDD(I)=DUMMY(I1+2)                                               CBLL1810
      II=14+3*NOMODE                                                    CBLL1820
      DO 702 I=1,NOLEG                                                  CBLL1830
      I1=II+1+3*(I-1)                                                   CBLL1840
      FMSX(I)=DUMMY(I1)                                                 CBLL1850
      FMSY(I)=DUMMY(I1+1)                                               CBLL1860
  702 FMSZ(I)=DUMMY(I1+2)                                               CBLL1870
      II=14+3*NOMODE+3*NOLEG                                            CBLL1880
      DO 703 I=1,NODS                                                   CBLL1890
      I1=II+1+3*(I-1)                                                   CBLL1900
      FDSX(I)=DUMMY(I1)                                                 CBLL1910
      FDSY(I)=DUMMY(I1+1)                                               CBLL1920
  703 FDSZ(I)=DUMMY(I1+2)                                               CBLL1930
      WRITE(6,619)                                                      CBLL1940
      WRITE(6,614)                                                      CBLL1950
      WRITE(6,610)TYM(IJ),XSDD,YSDD,ZSDD,PHI,THTA,PSI,WX,WY,WZ,         CBLL1960
     *               WXD,WYD,WZD                                        CBLL1970
      DO 704 I=1,NOMODE                                                 CBLL1980
  704 WRITE(6,611)I,GC(I),GCD(I),GCDD(I)                                CBLL1990
      WRITE(6,620)                                                      CBLL2000
      DO 705 I=1,NOLEG                                                  CBLL2010
  705 WRITE(6,612)I,FMSX(I),FMSY(I),FMSZ(I)                             CBLL2020
      WRITE(6,616)                                                      CBLL2030
      DO 706 I=1,NODS                                                   CBLL2040
  706 WRITE(6,613)I,FDSX(I),FDSY(I),FDSZ(I)                             CBLL2050
C                                                                       CBLL2060
C     CALCULATE DIRECTION COSINE MATRIX                                 CBLL2070
C                                                                       CBLL2080
      COSPHI=COS(PHI)                                                   CBLL2090
      SINPHI=SIN(PHI)                                                   CBLL2100
      COSTHA=COS(THTA)                                                  CBLL2110
      SINTHA=SIN(THTA)                                                  CBLL2120
      COSPSI=COS(PSI)                                                   CBLL2130
      SINPSI=SIN(PSI)                                                   CBLL2140
      DC(1,1)=COSTHA*COSPHI                                             CBLL2150
      A=SINTHA*COSPSI                                                   CBLL2160
      B=COSPHI*SINPSI                                                   CBLL2170
      DC(1,2)=SINPHI*A-B                                                CBLL2180
      C=SINPHI*SINPSI                                                   CBLL2190
      DC(1,3)=COSPHI*A+C                                                CBLL2200
      DC(2,1)=COSTHA*SINPSI                                             CBLL2210
      DC(2,2)=SINTHA*C+COSPHI*COSPSI                                    CBLL2220
      DC(2,3)=SINTHA*B-SINPHI*COSPSI                                    CBLL2230
```

```
      DC(3,1)=-SINTHA                                                   CBLL2240
      DC(3,2)=SINPHI*COSTHA                                             CBLL2250
      DC(3,3)=COSPHI*COSTHA                                             CBLL2260
      ZETA=-ZETA                                                        CBLL2270
C                                                                       CBLL2280
C     ACCELERATION COMPONENTS                                           CBLL2290
C                                                                       CBLL2300
      GX=-GRAV*COS(ZETA)                                                CBLL2310
      GZ=GRAV*SIN(ZETA)                                                 CBLL2320
      GRAVX=DC(1,1)*GX+DC(3,1)*GZ                                       CBLL2330
      GRAVY=DC(1,2)*GX+DC(3,2)*GZ                                       CBLL2340
      GRAVZ=DC(1,3)*GX+DC(3,3)*GZ                                       CBLL2350
      XCGA=DC(1,1)*XSDD+DC(2,1)*YSDD+DC(3,1)*ZSDD                       CBLL2360
      YCGA=DC(1,2)*XSDD+DC(2,2)*YSDD+DC(3,2)*ZSDD                       CBLL2370
      ZCGA=DC(1,3)*XSDD+DC(2,3)*YSDD+DC(3,3)*ZSDD                       CBLL2380
      WXX=-(WY**2+WZ**2)                                                CBLL2390
      WXY=WX*WY                                                         CBLL2400
      WXZ=WX*WZ                                                         CBLL2410
      WYY=-(WX**2+WZ**2)                                                CBLL2420
      WYZ=WY*WZ                                                         CBLL2430
      WZZ=-(WX**2+WY**2)                                                CBLL2440
      XSUM=0.                                                           CBLL2450
      YSUM=0.                                                           CBLL2460
      ZSUM=0.                                                           CBLL2470
      TXSUM=0.                                                          CBLL2480
      TYSUM=0.                                                          CBLL2490
      TZSUM=0.                                                          CBLL2500
C                                                                       CBLL2510
C     DETERMINE JOINT ACCELERATIONS                                     CBLL2520
C                                                                       CBLL2530
      DO 750 J=1,NOJTS                                                  CBLL2540
      ELX=XJT(J)                                                        CBLL2550
      ELY=YJT(J)                                                        CBLL2560
      ELZ=ZJT(J)                                                        CBLL2570
      ELDX=0.0                                                          CBLL2580
      ELDY=0.0                                                          CBLL2590
      ELDZ=0.0                                                          CBLL2600
      ELDDX=0.0                                                         CBLL2610
      ELDDY=0.0                                                         CBLL2620
      ELDDZ=0.0                                                         CBLL2630
      DO 720 I=1,NOMODE                                                 CBLL2640
      ELX=ELX+XPHI(J,I)*GC(I)                                           CBLL2650
      ELY=ELY+YPHI(J,I)*GC(I)                                           CBLL2660
      ELZ=ELZ+ZPHI(J,I)*GC(I)                                           CBLL2670
      ELDX=ELDX+XPHI(J,I)*GCD(I)                                        CBLL2680
      ELDY=ELDY+YPHI(J,I)*GCD(I)                                        CBLL2690
      ELDZ=ELDZ+ZPHI(J,I)*GCD(I)                                        CBLL2700
      ELDDX=ELDDX+XPHI(J,I)*GCDD(I)                                     CBLL2710
      ELDDY=ELDDY+YPHI(J,I)*GCDD(I)                                     CBLL2720
      ELDDZ=ELDDZ+ZPHI(J,I)*GCDD(I)                                     CBLL2730
  720 CONTINUE                                                          CBLL2740
      SAX(J)=ELX                                                        CBLL2750
      SAY(J)=ELY                                                        CBLL2760
      SAZ(J)=ELZ                                                        CBLL2770
      ACLX =  XCGA+ELDDX-WZD*ELY+WYD*ELZ+2.*(-WZ*ELDY+WY*ELDZ)          CBLL2780
     *        +WXX*ELX+WXY*ELY+WXZ*ELZ                                  CBLL2790
```

```
      ACLY =  YCGA+ELDDY+WZD*ELX-WXD*ELZ+2.*(WZ*ELDX-WX*ELDZ)
     *        +WXY*ELX+WYY*ELY+WYZ*ELZ
      ACLZ =  ZCGA+ELDDZ-WYD*ELX+WXD*ELY+2.*(-WY*ELDX+WX*ELDY)
     *        +WXZ*ELX+WYZ*ELY+WZZ*ELZ
      FX(J)=-ACLX*AMASS(J)
      FY(J)=-ACLY*AMASS(J)
      FZ(J)=-ACLZ*AMASS(J)
  750 CONTINUE
C
C     SUM FORCES AT EACH JOINT
C
      WRITE(6,615)
      DO 755 J=1,NOJTS
      FXJT(J)=0.
      FYJT(J)=0.
  755 FZJT(J)=0.
      DO 760 I=1,NOLEG
      J=JOTMS(I)
      FXJT(J)=FMSX(I)
      FYJT(J)=FMSY(I)
  760 FZJT(J)=FMSZ(I)
      DO 765 I=1,NODS
      J=JOTDS(I)
      FXJT(J)=FDSX(I)
      FYJT(J)=FDSY(I)
  765 FZJT(J)=FDSZ(I)
      DO 780 J=1,NOJTS
      FXJT(J)=FXJT(J)+FX(J)+AMASS(J)*GRAVX
      FYJT(J)=FYJT(J)+FY(J)+AMASS(J)*GRAVY
      FZJT(J)=FZJT(J)+FZ(J)+AMASS(J)*GRAVZ
C
C     SUM FORCES AND MOMENTS AT C. G.
C
      XSUM=XSUM+FXJT(J)
      YSUM=YSUM+FYJT(J)
      ZSUM=ZSUM+FZJT(J)
      TXSUM=TXSUM+SAY(J)*FZJT(J)-SAZ(J)*FYJT(J)
      TYSUM=TYSUM+SAZ(J)*FXJT(J)-SAX(J)*FZJT(J)
      TZSUM=TZSUM+SAX(J)*FYJT(J)-SAY(J)*FXJT(J)
      WRITE(6,402)J,FXJT(J),FYJT(J),FZJT(J)
  780 CONTINUE
C
C     SUMMARY INFORMATION
C
      WRITE(6,617)XSUM,YSUM,ZSUM,TXSUM,TYSUM,TZSUM
      GO TO 700
C
C     FORMAT STATEMENTS
C
  400 FORMAT(5X,I3,2X,7E10.3)
  401 FORMAT(/15X,17HJOINT COORDINATES//4X,5HJOINT,7X,1HX,11X,1HY,11X,
     *      1HZ)
  402 FORMAT(6X,I2,1X,3(2X,E10.3))
  403 FORMAT(//,15X,11HMODE SHAPES/)
  404 FORMAT(//10X,45HINCORRECT INFORMATION READ FROM TAPE4 - STOP )
  407 FORMAT(/6X,7HMODE = ,I2,//,4X,5HJOINT,7X,1HX,11X,1HY,11X,1HZ)
```

```
500 FORMAT(15I5)
501 FORMAT(5E10.3)
600 FORMAT(1H1,51X,33HCENTER BODY LANDING LOADS PROGRAM
   *                                         /42X,53HMASTER AGREEMENT, C
   *ONTRACT NAS1-8137, TASK ORDER FIVE/46X,45HMCDONNELL DOUGLAS ASTRON
   *AUTICS COMPANY - EAST,////)
601 FORMAT(10X, 9HNOJTS   = ,I10/10X,9HNOLEG   = ,I10/10X,9HNOMODE = ,
   *        I10/10X,9HNOTIME = ,I10/10X,9HNOSETS = ,I10)
602 FORMAT(10X,9HMODE    = ,5I10)
603 FORMAT(10X,9HJOTMS   = ,5I10)
604 FORMAT(10X,9HJOTDS   = ,10I10)
605 FORMAT(10X,9HZETA    = ,E10.3/10X,9HGRAV    = ,E10.3)
606 FORMAT(10X,9HMASS    = ,10E10.3,/(19X,10E10.3))
607 FORMAT(10X,9HTYM     = ,10E10.3/29X,10E10.3)
610 FORMAT(15X    ,7HTIME = ,E10.3/10X,7HXSDD = ,E10.3,8H YSDD = ,
   *        E10.3,8H ZSDD = ,E10.3/10X,7HPHI  = ,E10.3,8H THTA = ,
   *         E10.3,8H PSI  = ,E10.3/10X,7HWX   = ,E10.3,8H WY   = ,
   *        E10.3,8H WZ   = ,E10.3/10X,7HWXD  = ,E10.3,8H WYD  = ,
   *        E10.3,8H WZD  = ,E10.3)
611 FORMAT(10X,7HMODE = ,I5,5X,8H GC    = ,E10.3,8H GCD  = ,E10.3,
   *       8H GCDD = ,E10.3)
612 FORMAT(10X,7HI     = ,I5,5X,8H FMSX = ,E10.3,8H FMSY = ,E10.3,
   *       8H FMSZ = ,E10.3)
613 FORMAT(10X,7HI     = ,I5,5X,8H FDSX = ,E10.3,8H FDSY = ,E10.3,
   *       8H FDSZ = ,E10.3)
614 FORMAT(/,15X,17HTIME HISTORY DATA,/)
615 FORMAT(//15X,38HCENTER BODY LANDING LOADS DISTRIBUTION//4X,5HJOINT
   *       ,7X,1HX,11X,1HY,11X,1HZ)
616 FORMAT(/,15X,16HDRAG STRUT LOADS,/)
617 FORMAT(//,15X,28HLANDING LOADS SUMMED AT C.G.//,6X,5HFX = ,
   *       E10.3/6X,5HFY = ,E10.3/6X,5HFZ = ,E10.3/6X,5HTX = ,
   *       E10.3/6X,5HTY = ,E10.3/6X,5HTZ = ,E10.3)
619 FORMAT(1H1)
620 FORMAT(/,15X,16HMAIN STRUT LOADS,/)
    END
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F

NUMERICAL INTEGRATION ROUTINE
LANDING LOADS AND MOTIONS PROGRAM

F.1 Introduction - The methods and procedures employed during the numerical integration of the equations of motion in the Landing Loads and Motions Program are described in the following paragraphs. These operations are performed in the subroutines RKCUT and INITUP. The sequence of program calls for these subroutines are shown in Figure 5-17 and subroutine listings are given in Appendix I.

F.2 Method - The second-order equations of motion in the Landing Loads and Motions Program are reduced to an equivalent set of simultaneous first-order equations. Each of these first order equations takes the general form of

$$\dot{y} = f(y,t) \text{ where } y(t_o) = y_o \tag{F-1}$$

At any step n in the integration of these equations with respect to time t, the quantities t_n, y_n, and $\dot{y}_n = f(y_n, t_n)$ are available. To compute y_{n+1} and thus $\dot{y}_{n+1} = f(y_{n+1}, t_{n+1})$, where $t_{n+1} = t_n + \Delta t$, and Δt (program variable HZ) is the integration interval, the following Runge-Kutta formulas are used:

$$\begin{array}{lll} y_A = y_n + \frac{\Delta t}{2}\,\dot{y}_n, & \dot{q}_n = \dot{y}_n, & \dot{y}_A = f(y_A,\ t_n + \frac{\Delta t}{2}) \\ y_B = y_n + \frac{\Delta t}{2}\,\dot{y}_A, & \dot{q}_A = \dot{q}_n + 2\dot{y}_A, & \dot{y}_B = f(y_B,\ t_n + \frac{\Delta t}{2}) \\ y_C = y_n + \Delta t\,\dot{y}_B, & \dot{q}_B = \dot{q}_A + 2\dot{y}_B, & \dot{y}_C = f(y_C,\ t_n + \Delta t) \\ y_{n+1} = y_n + \Delta t\,(\dot{q}_B + \dot{y}_C), & \dot{q}_C = \dot{y}_C, & \dot{y}_{n+1} = f(y_{n+1},\ t_n + \Delta t) \end{array} \tag{F-2}$$

References (F-1) and (F-2) present an explanation of these formulas and a discussion of the Runge-Kutta method.

These formulas are applied to the system of equations using either a fixed or variable integration interval, Δt. When a variable Δt is used, the routine computes a truncation error indicator at each step of the integration, based on the quantity

$$E_{n+1} = \text{Max}\ \{\text{Min}\ [|\Delta t(\dot{y}^i_{n+1} - \dot{y}^i_C)\ |,\ |\ \frac{\Delta t(\dot{y}^i_{n+1} - \dot{y}^i_C)}{y^i_{n+1}}\ |]\} \tag{F-3}$$

where i = 1, 2, ..., N and N is the total number of integrated variables, and the superscript i denotes the ith integrated variable. The notation of Equation (F-3), indicates that for each ith integrated variable, the minimum of the two quantities

$$|\ \Delta t\ (\dot{y}^i_{n+1} - \dot{y}^i_C)\ |\ \text{and}\ |\ \frac{\Delta t\ (\dot{y}^i_{n+1} - \dot{y}^i_C)}{y^i_{n+1}}\ |$$

is saved. The truncation error indicator, E_{n+1}, is then set equal to the maximum of these minimums. The value of E_{n+1} is compared with the two constants E_{max} (EMAX) and E_{min} (EMIN) which are the input values for the maximum and minimum allowable truncation errors, respectively. This comparison results in a modification of Δt as follows:

1. If $E_{n+1} < E_{min}$ for two consecutive integration steps, Δt is doubled and the integration continues with the new Δt. However, if Δt equals Δt_{max} (HMAX), Δt is not doubled.
2. If $E_{min} \leq E_{n+1} \leq E_{max}$, Δt is left unchanged.
3. If $E_{n+1} \geq E_{max}$, Δt is halved and the values at step n, which are saved in COMMON, are used to continue the integration. However, if Δt equals Δt_{min} (HMIN), Δt is not halved.

This procedure is continuously applied to the center body equations of motion, Equation (5-23). The equations of motion for the footpads, Equation (5-4), are continuously integrated by the above procedure when the "contacting footpad" option of the program is employed. If the "noncontacting footpad" option is specified, a footpad's equations of motion are integrated only when the footpad is determined to be in contact with the landing surface. For a footpad approaching the landing surface, contact is defined as the condition when the bottom of the footpad is found to be within a tolerance band above and below the landing surface, as shown in Figure F-1(a). If this footpad is found to have penetrated below this tolerance band as shown in Figure F-1(b), a linear interpolation procedure is employed to estimate at what time the footpad would have entered the tolerance band. The integration is backed up and then continued from this point in time with the footpad considered contacting.

F.3 Subroutine Calls - There are six calls to the integration routine. A brief description of each of the entries into the subroutines is given below.

Subroutine	Entry Point	Operation
INITUP	LOC	Sets up a list of cutoff variables and their cutoff values.
INITUP	INUPD	Sets up a list of all integrated variables.
RKCUT	SETUP	Initializes numerical integration procedure for type of integration requested.
RKCUT	INTEG	Performs all integration and computes truncation error indicator.
RKCUT	UPDATE	Updates integration variables and modifies integration interval when required.
RKCUT	CUT	Checks cutoff limits for introduction of footpad equations into integration routine.

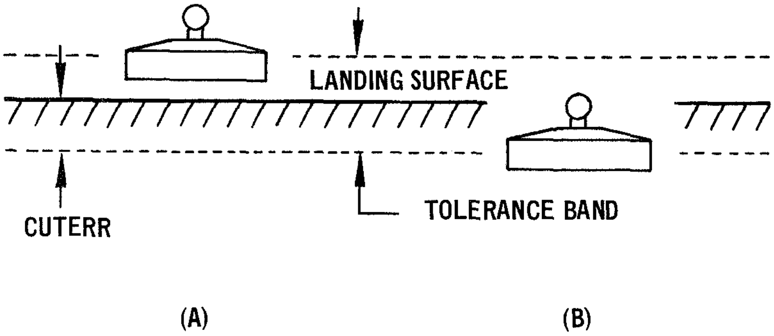


FIGURE F-1 INTEGRATION CONTROL TOLERANCE FOR FOOTPAD EQUATIONS OF MOTION

F-4 References

F-1 Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, New York, 1962, pp. 212-213.

F-2 Conte, S. D., Elementary Numerical Analysis, McGraw-Hill Book Company, New York, 1965, pp. 204-258.

THIS PAGE INTENTIONALLY LEFT BLANK

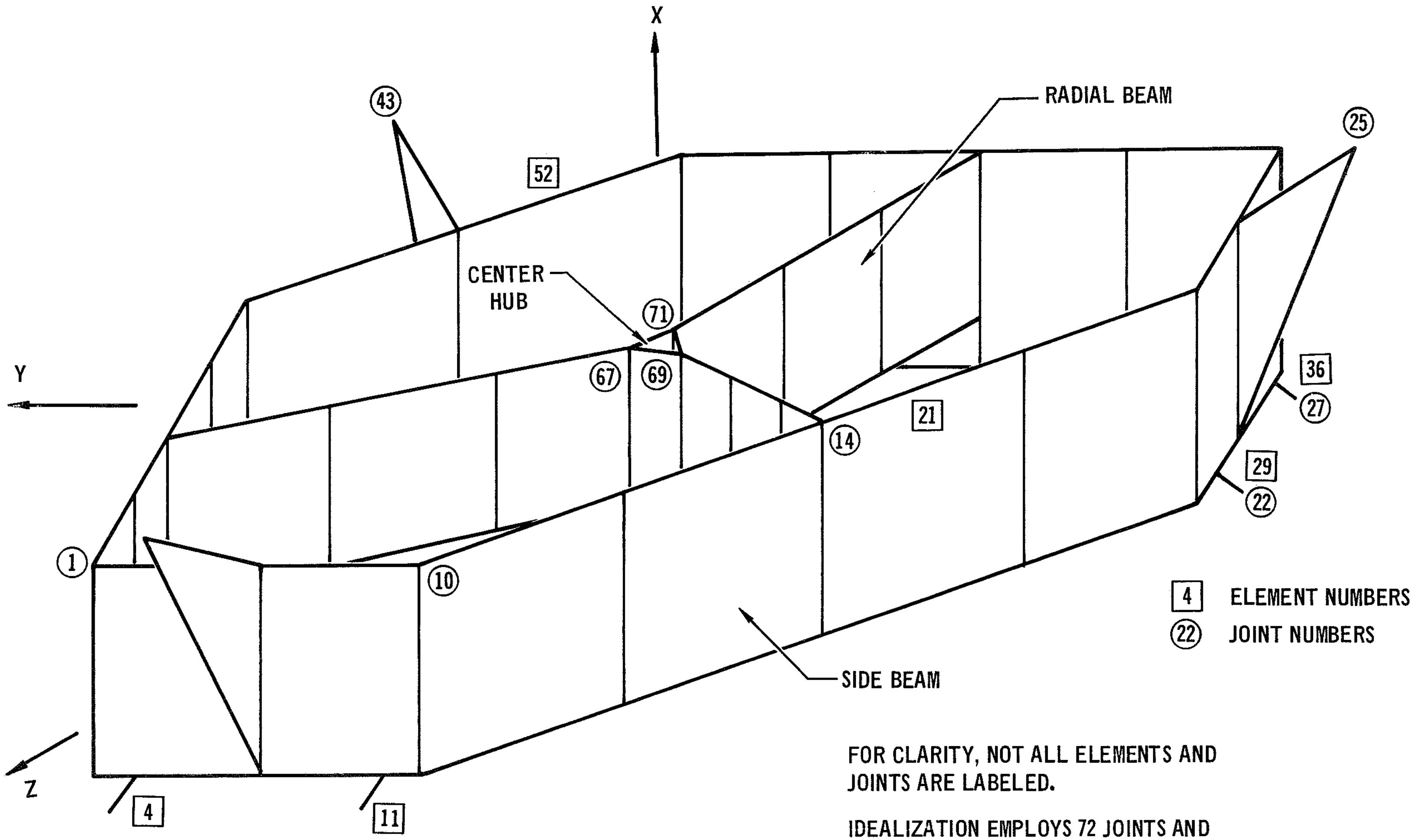
APPENDIX G

CENTER BODY INTERNAL LOAD DISTRIBUTION FOR TASK ORDER THREE LANDER USING REFINED IDEALIZATION

A refined idealization of the Task Order Three lander center body, shown in Figure G-1, was employed to determine displacements and internal loads for the loading condition defined in Section 6.2.4. This idealization, which employs 72 joints and 108 bar elements, includes a more refined model of the side beams and radial beams. The center hub was modeled with nine elements to more nearly represent a cylinder. In addition, clevis fittings, which connect the drag struts of a gear to the center body, were idealized with elements such as 4, 11, 29, and 36. Accordingly, joints 22 and 27 (and 40 and 45 which are hidden) are the actual center body attach points for gears 2 and 3 and were assumed to be pinned supports. Main strut attach points (joints 25 and 43) for gears 2 and 3 were also assumed to be pinned supports.

Loads that were applied to the joints in the idealization of Figure 6-8 were applied to coincident joints in Figure G-1. For example, loads that were applied to joint 6 in Figure 6-8 were applied to joint 10 in the refined idealization. Loads that were applied to joints 31, 32 and 33 (on the center hub) in Figure 6-8 were distributed to joints 67 through 72, some of which are hidden in Figure G-1. These loads were distributed to the refined idealization joint locations in a statically equivalent manner.

A comparison of selected displacements and internal loads obtained with the 33 joint idealization and the refined idealization is given in Figure G-2.



FOR CLARITY, NOT ALL ELEMENTS AND JOINTS ARE LABELED.

IDEALIZATION EMPLOYS 72 JOINTS AND 108 BAR ELEMENTS

FIGURE G–1 REFINED CENTER BODY STRUCTURAL IDEALIZATION

	JOINT DISPLACEMENTS (CM)		
	X	Y	Z
JOINT 1 (FIGURE 6-8)*	-2.594×10^{-1}	-1.284×10^{-5}	9.955×10^{-2}
JOINT 1 (FIGURE G-1)**	-5.325×10^{-1}	-1.877×10^{-4}	1.775×10^{-1}
JOINT 8 (FIGURE 6-8)	-1.342×10^{-1}	2.195×10^{-1}	-2.636×10^{-2}
JOINT 14 (FIGURE G-1)	-2.365×10^{-1}	4.810×10^{-1}	-1.007×10^{-1}

*FIGURE 6-8 ILLUSTRATES THE 33 JOINT IDEALIZATION.
**FIGURE G-1 ILLUSTRATES THE REFINED (72 JOINT) IDEALIZATION.

	ELEMENT INTERNAL LOADS AT "P" END					
	FORCE X (DYNES)	FORCE Y (DYNES)	FORCE Z (DYNES)	MOMENT X (CM-DYNES)	MOMENT Y (CM-DYNES)	MOMENT Z (CM-DYNES)
ELEMENT 14 (FIGURE 6-8)	-2.588×10^{8}	9.057×10^{7}	-1.025×10^{8}	9.972×10^{6}	5.477×10^{9}	4.688×10^{9}
ELEMENT 21 (FIGURE G-1)	-2.140×10^{8}	1.230×10^{8}	-1.778×10^{8}	1.088×10^{7}	3.830×10^{9}	8.550×10^{9}
ELEMENT 32 (FIGURE 6-8)	-5.394×10^{8}	6.205×10^{8}	1.633×10^{8}	5.697×10^{6}	-4.800×10^{9}	1.061×10^{10}
ELEMENT 52 (FIGURE G-1)	-1.115×10^{9}	1.163×10^{9}	1.822×10^{8}	5.800×10^{7}	-6.190×10^{9}	1.756×10^{10}

FIGURE G-2 COMPARISON OF COMPUTER RESULTS FOR 33 AND 72 JOINT IDEALIZATIONS OF TASK ORDER THREE LANDER CENTER BODY

APPENDIX H

PROGRAM LISTING

STRUCTURAL ANALYSIS PROGRAM

```
      OVERLAY ( SASLP, 0, 0 )

      PROGRAM  SAPT5 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1              TAPE1,TAPE2,TAPE9)
      COMMON COM(30)
      EQUIVALENCE ( COM( 17), INDRKT )
      EQUIVALENCE ( COM( 24), INDNMA )
      EQUIVALENCE ( COM(30), INDPGM )
C
C                THE DIMENSION OF COMAIN CAN BE INCREASED OR DECREASED
C                     TO CONTROL THE STORAGE AVAILABLE FOR THE STIFFNESS
C                     MATRIX (IF ISFDIM IS ALSO CHANGED).
C
C     COMMON / CMAIN / COMAIN(14455)
      COMMON / CMAIN / COMAIN(16955)
    5 CONTINUE
C              INITIALIZE WRSTRK ROUTINE
      CALL WRSTR1(I,J,DUM)
C
C          DATA AND VARIABLE SET-UP
C
      CALL SECOND (TIME0)
      CALL OVERLAY (5LSASLP,1,0,6HRECALL)
      CALL SECOND (TIME1)
      IF ( INDPGM .EQ. 0 ) GO TO 8
C
C          GEAR ANALYSIS INPUT AND PROCESSING ROUTINES
C
      CALL OVERLAY ( 6LGEARRT,5,0,6HRECALL)
      CALL SECOND(TIME2)
      GO TO 20
    8 CONTINUE
C
C          CENTER BODY OPTION ROUTINES
C          LOCAL STIFFNESS AND TRANSFORMATION MATRICES AND STRUCTURAL
C                STIFFNESS MATRIX GENERATION
C
      IF (INDRKT.EQ.0) CALL OVERLAY (5LSASLP,2,0,6HRECALL)
      CALL SECOND (TIME2)
C     ***************************************************
C     *  OVERLAYS THREE AND FOUR ARE MUTUALLY EXCLUSIVE  *
C     ***************************************************
      IF (INDNMA.NE.0) GO TO 10
C
C          DISPLACEMENT, ROTATION, FORCE, AND MOMENT SOLUTION
C
      CALL OVERLAY (5LSASLP,3,0,6HRECALL)
      CALL SECOND (TIME3)
      GO TO 20
C
C          NORMAL MODE ANALYSIS SECTION
C
10    CALL OVERLAY (5LSASLP,4,0,6HRECALL)
      CALL SECOND (TIME4)
20    CONTINUE
```

```
      T01=TIME1-TIME0                                                   MAIN 530
      T12=TIME2-TIME1                                                   MAIN 540
      WRITE(6,40)TIME0,TIME1,T01                                        MAIN 550
      IF ( INDPGM .EQ. 0 ) GO TO 25                                     MAIN 560
      WRITE(6,44)TIME1,TIME2,T12                                        MAIN 570
      GO TO 5                                                           MAIN 580
   25 CONTINUE                                                          MAIN 590
      WRITE(6,42)TIME1,TIME2,T12                                        MAIN 600
      IF (INDNMA.EQ.0) GO TO 30                                         MAIN 610
      T23OR4=TIME4-TIME2                                                MAIN 620
      WRITE (6,50) TIME2,TIME4,T23OR4                                   MAIN 630
      GO TO 5                                                           MAIN 640
30    CONTINUE                                                          MAIN 650
      T23OR4=TIME3-TIME2                                                MAIN 660
      WRITE (6,60) TIME2,TIME3,T23OR4                                   MAIN 670
      GO TO 5                                                           MAIN 680
C                                                                       MAIN 690
40    FORMAT (1H138X,26HCPU TIME USAGE TABLE (SEC)///63X32H TIME IN     MAIN 700
     1TIME OUT      TOTAL,/33H0INPUT AND INITIALIZATION OVERLAY,26X2(2XFMAIN 710
     2  10.3),1XF10.2  /)                                               MAIN 720
   42 FORMAT ( 47H0STRUCTURAL STIFFNESS MATRIX GENERATION OVERLAY,      MAIN 730
     1            14XF10.3, 2XF10.3, 1XF10.2 /)                         MAIN 740
   44 FORMAT ( 32H0GEAR INPUT AND ANALYSIS OVERLAY, 27X2(2XF10.3),      MAIN 750
     1            1XF10.2)                                              MAIN 760
50    FORMAT (29H0NORMAL MODE ANALYSIS OVERLAY,32XF10.3,2XF10.3,1XF10.2)MAIN 770
60    FORMAT (59H0DISPLACEMENT, ROTATION, FORCE, AND MOMENT SOLUTION OVEMAIN 780
     1RLAY,2XF10.3,2XF10.3,1XF10.2//)                                   MAIN 790
      END                                                               MAIN 800
```

```
      SUBROUTINE ERPNT1 (ALPHA,LNGTH,ICODE)                             ERPN  10
C         PRINT ERROR MESSAGES                                          ERPN  20
C              ICODE = 0,  NON-FATAL ERROR, CONTINUE                    ERPN  30
C              ICODE = 1,  FATAL ERROR, STOP                            ERPN  40
C              ICODE =-1,  FATAL ERROR, SET FATAL ERROR INDICATOR AND   ERPN  50
C                                                            CONTINUE   ERPN  60
      DATA IFATAL / 0 /                                                 ERPN  70
      DIMENSION ALPHA(13)                                               ERPN  80
      IF(ICODE.NE.0) GO TO 70                                           ERPN  82
      WRITE(6,50) (ALPHA(I),I=1,LNGTH)                                  ERPN  84
      GO TO 30                                                          ERPN  86
   70 CONTINUE                                                          ERPN  88
      WRITE (6,40) (ALPHA(I),I=1,LNGTH)                                 ERPN  90
      IF (ICODE) 10,30,20                                               ERPN 100
10    IFATAL=1                                                          ERPN 110
      GO TO 30                                                          ERPN 120
20    STOP                                                              ERPN 130
30    RETURN                                                            ERPN 140
C         CHECK ENTRY FOR FATAL INDICATOR                               ERPN 150
      ENTRY ERPNT2                                                      ERPN 160
      IF (IFATAL.EQ.1) STOP                                             ERPN 170
      RETURN                                                            ERPN 180
C                                                                       ERPN 190
   40 FORMAT(20H ***ERROR***        /13A10//)                           ERPN 200
   50 FORMAT(20H ***WARNING***      /13A10//)                           ERPN 205
      END                                                               ERPN 210
```

```
      SUBROUTINE WRSTRK (IROW,JCOL,STFNZ)
C
C         IROW = ROW NO.       JCOL = COLUMN NO.
C         STFNZ  = VARIABLE
C
      DIMENSION JCOLD(6),  STFNZD(6), COL(6)
      DATA COL / 6*4HCOL. /
C
C               THE NON  ZERO ELEMENTS OF THE STRUCTURAL STIFFNESS
C         MATRIX ARE PASSED ONE AT A TIME AND PRINTED BY ROWS SIX
C         VALUES PER LINE
      IF (IRSAVE.EQ.IROW) GO TO 10
C
C         IF NEW ROW, PRINT ANY VARIABLES OF THE LAST ROW BEING HELD IN
C                                   A LINE AND START A NEW LINE
      INDRNS=1
      GO TO 20
10    IF (ICOUNT.NE.6) GO TO 40
C
C         IF SIX VARIABLES HAVE BEEN STORED IN A LINE, PRINT IT AND
C                                   START A NEW LINE.
20    CONTINUE
      IF (INDLIN.NE.1) GO TO 50
      WRITE (6,80) IRSAVE,(COL(I),JCOLD(I),STFNZD(I),I=1,ICOUNT)
      LINET=LINET+2
      INDLIN=2
30    ICOUNT=0
      IF (INDRNS.EQ.0) GO TO 40
      INDRNS=0
      INDLIN=1
      IRSAVE=IROW
40    ICOUNT=ICOUNT+1
      JCOLD(ICOUNT)=JCOL
      STFNZD(ICOUNT)=STFNZ
      IF (LINET.GT.56) GO TO 60
      RETURN
50    WRITE (6,90) (COL(I),JCOLD(I),STFNZD(I),I=1,ICOUNT)
      LINET=LINET+1
      GO TO 30
60    IF (IROW.LT.0) RETURN
      WRITE (6,70)
      LINET=3
      RETURN
C
C         INITIALIZE THIS ROUTINE
C
      ENTRY WRSTR1
      LINET  = 59
      ICOUNT =  0
      INDLIN =  1
      INDRNS =  0
      IRSAVE =  1
      RETURN
C
C         FORMAT 100 IS USED TO PRINT THE FIRST LINE FOR EACH ROW
```

```
C         FORMAT 200 IS USED FOR ALL OTHER LINES ( INDLIN .NE. 1)       WRST 560
C                                                                       WRST 570
C                                                                       WRST 580
70    FORMAT (28H1STRUCTURAL STIFFNESS MATRIX//)                        WRST 590
80    FORMAT (5H0ROW I3,6(1XA4,I4,1H=E10.3))                            WRST 600
90    FORMAT (8X,6(1XA4,I4,1H=E10.3))                                   WRST 610
      END                                                               WRST 620
```

```
      OVERLAY ( SASLP, 1, 0 )                                           OV10  10

      PROGRAM INITAL                                                    INIT  10
      COMMON COM(30)                                                    INIT  20
      EQUIVALENCE ( COM( 30), INDPGM )                                  INIT  30
C                                                                       INIT  40
C         READ DATA CARDS AND SORT INTO PROPER ORDER                    INIT  50
C                                                                       INIT  60
      CALL DATSET                                                       INIT  70
      IF(INDPGM.EQ.1)RETURN                                             INIT  80
C                                                                       INIT  90
C         READ VARIABLE INPUT VALUES ARRANGED BY (DATSET)               INIT 100
C                                                                       INIT 110
      CALL RDDATA                                                       INIT 120
      RETURN                                                            INIT 130
      END                                                               INIT 140
```

```
      SUBROUTINE DATSET                                                 DATS  10
      DIMENSION CARD(8), IDATA(7)                                       DATS  20
      COMMON COM(30)                                                    DATS  30
      EQUIVALENCE ( COM( 1), NJOINT )                                   DATS  40
      EQUIVALENCE ( COM( 2), NFORCE )                                   DATS  50
      EQUIVALENCE ( COM( 3), NMOMNT )                                   DATS  60
      EQUIVALENCE ( COM( 4), NBAR   )                                   DATS  70
      EQUIVALENCE ( COM( 5), NJPNT  )                                   DATS  80
      EQUIVALENCE ( COM(20), NLIMIT )                                   DATS  90
      EQUIVALENCE ( COM ( 29 ), NSHPAN   )                              DATS 100
      EQUIVALENCE ( COM(30), INDPGM )                                   DATS 110
      DIMENSION ERL(5)                                                  DATS 120
      DIMENSION ERJ( 5), ERF( 5), ERM( 5), ERB( 5)                      DATS 130
      DIMENSION ERSH(5)                                                 DATS 140
      DATA ERSH /10H THE SHEAR, 10H PANELS AR, 10HE NOT NUMB,           DATS 150
     1          10HERED SEQUE, 10HNTIALLY.   /                          DATS 160
      DATA ERL / 10H THE LIMIT, 10HS  ARE NOT, 10H NUMBERED ,           DATS 170
     1          10HSEQUENTIAL, 10HLY        /                           DATS 180
      DATA ERJ / 10H THE NODAL, 10H POINTS AR, 10HE NOT NUMB,           DATS 190
     1          10HERED SEQUE,  10HNTIALLY       /                      DATS 200
      DATA ERF / 10H THE FORCE, 10H VECTORS A, 10HRE NOT NUM,           DATS 210
     1           10HBERED SEQU, 10HENTIALLY  /                          DATS 220
      DATA ERM / 10H THE MOMEN, 10HT VECTORS , 10HARE NOT NU,           DATS 230
     1          10HMBERED SEQ, 10HUENTIALLY /                           DATS 240
      DATA ERB / 10H THE BAR D, 10HEFINITIONS, 10H ARE NOT N,           DATS 250
     1          10HUMBERED SE, 10HQUENTIALLY/                           DATS 260
      DIMENSION EDHDC(6)                                                DATS 270
      DATA GEAR / 10HGEAR           /                                   DATS 280
      DATA STRUCT / 10HSTRUCTURE  /                                     DATS 290
      DATA EDHDC /    10H THE DATA , 10HCASE DOES , 10HNOT HAVE A,      DATS 300
     1                10HN ACCEPTAB, 10HLE HEADER , 10HCARD.       /    DATS 310
      NJOINT=0                                                          DATS 320
      NLIMIT=0                                                          DATS 330
      NFORCE=0                                                          DATS 340
      NMOMNT=0                                                          DATS 350
      NBAR=0                                                            DATS 360
      NJPNT=0                                                           DATS 370
      NSHPAN = 0                                                        DATS 380
      REWIND 1                                                          DATS 390
      MJOINT  = 0                                                       DATS 400
      MJPNT   = 0                                                       DATS 410
      MLIMIT  = 0                                                       DATS 420
      MFORCE  = 0                                                       DATS 430
      MMOMNT  = 0                                                       DATS 440
      MBAR    = 0                                                       DATS 450
      MSHPAN = 0                                                        DATS 460
      IFRIST = 0                                                        DATS 470
      IBADCD = 0                                                        DATS 480
      IFRST = 0                                                         DATS 490
10    READ (5,200) CARD                                                 DATS 500
      IF ( EOF, 5) 12, 14                                               DATS 510
   12 WRITE(6, 240)                                                     DATS 520
  240 FORMAT (39H1END OF JOB---END OF DATA SET ON UNIT 5 )              DATS 530
      STOP                                                              DATS 540
   14 CONTINUE                                                          DATS 550
```

```
      IF ( IFRST .NE. 0 ) GO TO 17
      WRITE (6,190)
      IFRST = 1
   17 CONTINUE
      IF ( IFRIST .NE. 0 ) GO TO 16
C
C         CHECK FOR DATA SET HEADER CARD  (GEAR) OR (STRUCTURE)
C              LEFT JUSTIFIED IN COL. 1-10
C
      INDPGM = 1
      IF ( CARD(1) .EQ. GEAR ) GO TO 13
      INDPGM = 0
      IF ( CARD(1) .EQ. STRUCT) GO TO 15
      IF ( IBADCD  .EQ. 0 ) CALL ERPNT1( EDHDC, 6, 0 )
      IBADCD = 1
      WRITE(6,230) CARD
      GO TO 10
   13 CONTINUE
      IF ( IBADCD .NE. 0 ) WRITE(6,190)
      RETURN
   15 CONTINUE
      IF ( IBADCD .NE. 0 ) WRITE(6,190)
      WRITE(6,192)
      IFRIST = 1
      GO TO 10
   16 CONTINUE
      WRITE (6,230) CARD
C
C              CONVERT CODE AND SUBSCRIPT TO INTEGER
C
      DECODE (5,210,CARD(1)) ICODE, INDEX
C
C         ANY CODE NO. .GT. 7 IS CONSIDERED AN END OF RECORD
      IF (ICODE.GT.7) GO TO 110
C
C         IF COLUMN ONE IS BLANK OR ZERO CONSIDER IT A COMMENT
      IF (ICODE.EQ.0) GO TO 10
C
      WRITE(1,300) ICODE
      WRITE(1,200) CARD
      GO TO (20,40,50,60,70,80,90), ICODE
20    NJOINT=NJOINT+1
      MJOINT =  MAX0( MJOINT, INDEX )
      GO TO 10
40    NJPNT=NJPNT+1
      MJPNT  =  MAX0( MJPNT , INDEX )
      GO TO 10
50    NLIMIT=NLIMIT+1
      MLIMIT =  MAX0( MLIMIT, INDEX )
      GO TO 10
60    NFORCE=NFORCE+1
      MFORCE =  MAX0( MFORCE, INDEX )
      GO TO 10
70    NMOMNT=NMOMNT+1
      MMOMNT =  MAX0( MMOMNT, INDEX )
      GO TO 10
```

```
80    NBAR=NBAR+1                                                       DATS1120
      MBAR   =  MAXO( MBAR   , INDEX )                                  DATS1130
      GO TO 10                                                          DATS1140
   90 CONTINUE                                                          DATS1150
      MSHPAN = MAXO( MSHPAN, INDEX )                                    DATS1160
      NSHPAN = NSHPAN + 1                                               DATS1170
      GO TO 10                                                          DATS1180
  110 CONTINUE                                                          DATS1190
C         CHECK TO SEE IF INPUT CARDS ARE SEQUENCED PROPERLY            DATS1200
C                  (THIS WILL NOT CATCH ALL KEY PUNCH ERRORS)           DATS1210
      NJPNT = NJPNT + NJOINT                                            DATS1220
      MJPNT = MAXO(MJPNT,MJOINT)                                        DATS1230
      IF (  MJPNT  .NE. NJPNT   ) CALL ERPNT1( ERJ, 5, -1 )             DATS1240
      IF (  MLIMIT .NE. NLIMIT  ) CALL ERPNT1( ERL, 5, -1 )             DATS1250
      IF (  MFORCE .NE. NFORCE  ) CALL ERPNT1( ERF, 5, -1 )             DATS1260
      IF (  MMOMNT .NE. NMOMNT  ) CALL ERPNT1( ERM, 5, -1 )             DATS1270
      IF (  MBAR   .NE. NBAR    ) CALL ERPNT1( ERB, 5, -1 )             DATS1280
      IF ( MSHPAN .NE. NSHPAN ) CALL ERPNT1( ERSH, 5, -1 )              DATS1290
C                                                                       DATS1300
C         IF A FATAL ERROR HAS OCCURRED, STOP                           DATS1310
C                                                                       DATS1320
      CALL ERPNT2                                                       DATS1330
C                                                                       DATS1340
      END FILE 1                                                        DATS1350
      RETURN                                                            DATS1360
C                                                                       DATS1370
  190 FORMAT ( 1H1, 43X 45HSTRUCTURAL ANALYSIS PROGRAM --- LEGGED LANDERDATS1380
     1/37X60HMASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVDATS1390
     2E/45X44HMCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EAST              DATS1400
     3 /// )                                                            DATS1410
  192 FORMAT    (                                                       DATS1420
     337H STRUCTURAL ANALYSIS DATA - CARD CODE, //28X    27HBLANK - 0   DATS1430
     4      COMMENTS ,/32X1H1,13X,23HNODAL POINT DEFINITIONS,/32X1H2,13XDATS1440
     5,16HREFERENCE POINTS,/32X1H3,13X,33HNODAL POINT RESTRAINT DEFINITIDATS1450
     6ONS,/32X1H4,13X,13HFORCE VECTORS,/32X1H5,13X,15HMOMENT VECTORS ,/3DATS1460
     7 2X1H6, 13X, 15HBAR DEFINITIONS, / 32X1H7,13X                     DATS1470
     8  23HSHEAR PANEL DEFINITIONS,/32X1H8,13X,24HFORMATED-DATA TERMINATDATS1480
     9OR /   )                                                          DATS1490
200   FORMAT (8A10)                                                     DATS1500
  210 FORMAT ( I1, I4 )                                                 DATS1510
220   FORMAT (1XI4)                                                     DATS1520
230   FORMAT (1X8A10)                                                   DATS1530
  300 FORMAT( I4 )                                                      DATS1540
      END                                                               DATS1550
```

```
      SUBROUTINE RDDATA                                                 RDDA  10
      COMMON    NJOINT, NFORCE, NMOMNT, NBAR   , NJPNT , ITINJO, IBAR   RDDA  20
     1        , NROW   , INDSFL, INDSFG, INDISL, INDITR, ERRTOL, RELAXF RDDA  30
     2        , INDRLX, INDWKT, INDRKT, INDPLS, MINRST, NLIMIT, NRWTOR  RDDA  40
     3        , IREDTO, NEIGVL, INDNMA, INDWNM, TMAX   ,ISFDIM, INDWTS  RDDA  50
     4        , NSHPAN, INDPGM                                          RDDA  60
C          TOTAL LENGTH WITH STRSTF(0) .EQ. 4049                        RDDA  70
      COMMON / CMAIN /                                                  RDDA  80
     1     RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),  RDDA  90
     2     RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130), RDDA 100
     3     KS     (   130), BAREA (   130), BARIN (   130), BARIT (   130), RDDA 110
     4     BARJ   (   130), BARYM (   130), BARSM (   130), RKN    (   130), RDDA 120
     5     RKT    (   130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88), RDDA 130
     6     DPRTIT(    88), IROWL (   445), STRSTF(1)                   RDDA 140
      COMMON/SP/IAPAN(30), IBPAN(30), ICPAN(30), IDPAN(30),             RDDA 150
     1          VSPAN(30), ESPAN(30), TSPAN(30)                         RDDA 160
      DIMENSION STIFML(12,12), TRANSM(3,3), AMASS(102), IRWKP(103)      RDDA 170
      EQUIVALENCE ( KS(1), STIFML(1,1)), ( BARIT(1), TRANSM(1,1))       RDDA 180
      EQUIVALENCE ( RLIMTU(1),AMASS(1)), ( RLIMTL(15),IRWKP(1  ))       RDDA 190
      DIMENSION ERORD(4), ERRWKP(5)                                     RDDA 200
      DIMENSION ERLT(5)                                                 RDDA 210
      DIMENSION ERRED(4)                                                RDDA 220
      DIMENSION ERFROW(5)                                               RDDA 230
      DIMENSION ERLN(5)                                                 RDDA 240
      DIMENSION ERPQ(4)                                                 RDDA 250
      DIMENSION ERBN(4), ERJNL(4)                                       RDDA 260
      DIMENSION ERJN( 5), ERJR( 7), ERFR( 6), ERMT(6)                   RDDA 270
      DIMENSION EREGVL(8)                                               RDDA 280
      DATA EREGVL/ 10H AN EXCESS, 10HIVE NUMBER, 10H OF EIGENV,         RDDA 290
     1             10HALUES HAVE, 10H BEEN REQU, 10HESTED BY U,         RDDA 300
     2             10HSE OF NEIG, 10HVL.                  /             RDDA 310
      DATA ERRED / 40H IREDTO CAN NOT BE GREATER THAN NROW         /    RDDA 320
      DATA ERORD /40H IREDTO CAN NOT BE LARGER THAN 102        /        RDDA 330
      DATA ERRWKP/50H THE IRWKP ARRAY IS NOT FILLED IN ASCENDING ORDER /RDDA 340
      DATA ERPQ / 35H A ZERO LENGTH BAR IS NOT PERMITTED    /           RDDA 350
      DATA ERBN/ 10H THERE MUS, 10HT BE AT LE, 10HAST ONE BA,           RDDA 360
     1          10HR          /                                         RDDA 370
      DATA ERJNL/10H THERE MUS, 10HT BE AT LE, 10HAST TWO NO,           RDDA 380
     1          10HDAL POINTS/                                          RDDA 390
      DATA ERLN /50H A RESTRAINT INDICATOR IS OUT OF BOUNDS            /RDDA 400
      DATA ERFROW /45H INPUT DATA DISAGREES WITH MATRICES ON TAPE9     /RDDA 410
      DATA ERLT /50H LOWER REACTION LIMIT CONFLICTS WITH UPPER LIMIT   /RDDA 420
      DATA ERJN /10H A BAR DEF,10HINITION US,10HES AN UNDE,10HFINED NODARDDA 430
     1          ,10HL  POINT  /                                         RDDA 440
      DATA ERJR /10H A BAR DEF,10HINITION US,10HES AN UNDE,10HFINED POINRDDA 450
     1          ,10HT TO DEFIN,10HE THE BEND,10HING PLANES /            RDDA 460
      DATA ERFR /10H A NODAL P,10HOINT IS AS,10HSOCIATED W,10HITH AN UNDRDDA 470
     1          ,10HEFINED FOR,10HCE VECTOR /                           RDDA 480
      DATA ERMT /10H A NODAL P,10HOINT IS AS,10HSOCIATED W,10HITH AN UNDRDDA 490
     1          ,10HEFINED MOM,10HENT VECTOR/                           RDDA 500
      DIMENSION  ERSP3(3), ERSP4(4), ERSP5(5), ERSP6(4), ERSP7(4),      RDDA 510
     *     ERSP8(4), ERSP9(4)                                           RDDA 520
      DATA ERSP3 / 10H ONLY 130 , 10HBARS ARE A, 10HLLOWED.    /        RDDA 530
      DATA ERSP4 / 10H ONLY  74 , 10HBAR NODE P, 10HOINTS ARE ,         RDDA 540
     1             10HALLOWED.   /                                      RDDA 550
```

```
      DATA ERSP9 / 10H ONLY  30 , 10HSHEAR PANE, 10HLS ARE ALL,         RDDA 560
     1             10HOWED.      /                                       RDDA 570
      DATA ERSP5 / 10H ONLY  26 , 10HREFERENCE , 10H NODE POIN,         RDDA 580
     1             10HTS ARE ALL, 10HOWED.      /                        RDDA 590
      DATA ERSP6 / 10H ONLY  88 , 10HLIMIT CARD, 10HS ARE ALLO,         RDDA 600
     1             10HWED.      /                                        RDDA 610
      DATA ERSP7 / 10H ONLY  74 , 10HFORCE VECT, 10HORS ARE AL,         RDDA 620
     1             10HLOWED.    /                                        RDDA 630
      DATA ERSP8 / 10H ONLY  74 , 10HMOMENT VEC, 10HTORS ARE A,         RDDA 640
     1             10HLLOWED.   /                                        RDDA 650
      NAMELIST / INDATA / INDSFL, INDSFG, IND1SL, INDITR, ERRTOL,SOLVEC RDDA 660
     1         , RELAXF, INDRLX                                          RDDA 670
     2    , INDRKT, INDWKT, INDPLS, MINRST                               RDDA 680
     3    , INDNMA, IRWKP, AMASS, IREDTO, INDWNM, TMAX                   RDDA 690
     4 , ISFDIM, INDWTS                                                  RDDA 700
     *     ,NEIGVL                                                       RDDA 710
C                                                                        RDDA 720
C          CHECK INPUT DATA AGAINST VARIABLE DIMENSION LIMITS            RDDA 730
      IF ( NJOINT .GT. 74 ) CALL ERPNT1( ERSP4, 4, -1 )                  RDDA 740
      IF ( NJPNT .GT. 100 ) CALL ERPNT1( ERSP5, 5, -1 )                  RDDA 750
      IF (NLIMIT .GT.  88 ) CALL ERPNT1( ERSP6, 4, -1 )                  RDDA 760
      IF (NFORCE .GT.  74 ) CALL ERPNT1( ERSP7, 4, -1 )                  RDDA 770
      IF (NMOMNT .GT.  74 ) CALL ERPNT1( ERSP8, 4, -1 )                  RDDA 780
      IF (NSHPAN .GT.  30 ) CALL ERPNT1( ERSP9, 4, -1 )                  RDDA 790
      IF ( NBAR .GT. 130) CALL ERPNT1( ERSP3, 3, -1 )                    RDDA 800
      CALL ERPNT2                                                        RDDA 810
      REWIND 1                                                           RDDA 820
    1 READ(1,135)  ICODE                                                 RDDA 830
  135 FORMAT ( I4 )                                                      RDDA 840
      IF ( EOF, 1 ) 58, 4                                                RDDA 850
    4 CONTINUE                                                           RDDA 860
      GO TO (  21, 22, 23, 24, 25, 26, 27), ICODE                        RDDA 870
   21 CONTINUE                                                           RDDA 880
C          READ  CODE 1  DATA                                            RDDA 890
      READ(1,140) INDX,(RJXYZ(INDX,J),J=1,3),(IJRFM(INDX,J),J=1,6),     RDDA 900
     1                                       (IJALFM(INDX,J),J=1,2)      RDDA 910
      GO TO 1                                                            RDDA 920
   22 CONTINUE                                                           RDDA 930
C          READ  CODE 2  DATA                                            RDDA 940
      READ(1,140) INDX,(RJXYZ(INDX,J),J=1,3)                             RDDA 950
      GO TO 1                                                            RDDA 960
   23 CONTINUE                                                           RDDA 970
C          READ  CODE 3  DATA                                            RDDA 980
      READ(1,140) INDX, RLIMTU(INDX), RLIMTL(INDX), DPRTIT(INDX)         RDDA 990
      GO TO 1                                                            RDDA1000
   24 CONTINUE                                                           RDDA1010
C          READ  CODE 4  DATA                                            RDDA1020
      READ(1,140) INDX, (FRCVCT(INDX,J),J=1,3)                           RDDA1030
      GO TO 1                                                            RDDA1040
   25 CONTINUE                                                           RDDA1050
C          READ CODE 5 DATA                                              RDDA1060
      READ(1,140) INDX, (RMTVCT(INDX,J),J=1,3)                           RDDA1070
      GO TO 1                                                            RDDA1080
   26 CONTINUE                                                           RDDA1090
C          READ CODE 6 DATA                                              RDDA1100
      READ(1,160) INDX,  IBARP(INDX), IBARQ(INDX), IBARR(INDX),         RDDA1110
```

```
     1                 KS    (INDX), BAREA(INDX), BARIN(INDX),          RDDA1120
     2                 BARIT(INDX),BARJ (INDX), BARYM(INDX),            RDDA1130
     3                 BARSM(INDX), RKN    (INDX), RKT   (INDX)         RDDA1140
      GO TO 1                                                           RDDA1150
C          READ CODE 7 DATA                                             RDDA1160
   27 CONTINUE                                                          RDDA1170
      READ(1,150) INDX, IAPAN(INDX), IBPAN(INDX), ICPAN(INDX),          RDDA1180
     1     IDPAN(INDX), VSPAN(INDX), ESPAN(INDX), TSPAN(INDX)           RDDA1190
      GO TO 1                                                           RDDA1200
   58 CONTINUE                                                          RDDA1210
C                                                                       RDDA1220
C          CHECK BAR DEFINITION                                         RDDA1230
C                                                                       RDDA1240
      DO 60 I=1,NBAR                                                    RDDA1250
      IF (IBARP(I).GT.NJOINT.OR.IBARP(I).LT.1) CALL ERPNT1 (ERJN,5,-1)  RDDA1260
      IF (IBARQ(I).GT.NJOINT.OR.IBARQ(I).LT.1) CALL ERPNT1 (ERJN,5,-1)  RDDA1270
      IF (IBARP(I).EQ.IBARQ(I)) CALL ERPNT1 (ERPQ,4,-1)                 RDDA1280
60    IF (IBARR(I).GT.NJPNT) CALL ERPNT1 (ERJR,7,-1)                    RDDA1290
C                                                                       RDDA1300
C          CHECK JOINT DEFINITIONS FOR MISSING FORCE, MOMENT VECTORS,   RDDA1310
C                               AND RESTRAINT INFORMATION               RDDA1320
      DO 80 I=1,NJOINT                                                  RDDA1330
      DO 70 J=1,6                                                       RDDA1340
      K=IJRFM(I,J)                                                      RDDA1350
      IF (IABS(K).GT.NLIMIT) CALL ERPNT1 (ERLN,5,-1)                    RDDA1360
      IF (K.LE.0) GO TO 70                                              RDDA1370
      IF (RLIMTU(K).LE.RLIMTL(K)) CALL ERPNT1 (ERLT,5,-1)               RDDA1380
70    CONTINUE                                                          RDDA1390
      IF (IJALFM(I,1).GT.NFORCE.OR.IJALFM(I,1).LT.0) CALL ERPNT1 (ERFR,6RDDA1400
     1,-1)                                                              RDDA1410
      IF (IJALFM(I,2).GT.NMOMNT.OR.IJALFM(I,2).LT.0) CALL ERPNT1 (ERMT,6RDDA1420
     1,-1)                                                              RDDA1430
80    CONTINUE                                                          RDDA1440
C                                                                       RDDA1450
C          CALL SHEAR PANEL SUBROUTINE                                  RDDA1460
C                                                                       RDDA1470
      CALL SHRPAN                                                       RDDA1480
      IF ( NBAR .GT. 130) CALL ERPNT1( ERSP3, 3, -1 )                   RDDA1490
      CALL ERPNT2                                                       RDDA1500
C                                                                       RDDA1510
C          INITIALIZE DATA                                              RDDA1520
C                                                                       RDDA1530
      NROW=6*NJOINT                                                     RDDA1540
      INDITR=NROW*3                                                     RDDA1550
      INDPLS=0                                                          RDDA1560
      INDWKT=0                                                          RDDA1570
      INDRKT=0                                                          RDDA1580
      MINRST=6                                                          RDDA1590
      IND1SL=0                                                          RDDA1600
      ERRTOL=.0001                                                      RDDA1610
      INDSFG=1                                                          RDDA1620
      INDSFL=1                                                          RDDA1630
      INDWTS = 0                                                        RDDA1640
      INDRLX = 2                                                        RDDA1650
      RELAXF = 1.                                                       RDDA1660
      ISFDIM = 12904                                                    RDDA1670
```

```
      DO 85 I = 1, NROW                                                 RDDA1680
   85 SOLVEC(I) = 0.0                                                   RDDA1690
      INDNMA=0                                                          RDDA1700
      IREDTO=NROW                                                       RDDA1710
      INDWNM=0                                                          RDDA1720
      NEIGVL = 5                                                        RDDA1730
      TMAX=9999.                                                        RDDA1740
C                                                                       RDDA1750
C         READ INDICATORS AND CONTROL DATA IN BY NAMELIST               RDDA1760
C                                                                       RDDA1770
      READ (5,INDATA)                                                   RDDA1780
      WRITE(6,210) INDRKT,INDSFG,INDSFL,INDWKT,INDWTS,ISFDIM            RDDA1790
      IF (INDNMA.EQ.0) GO TO 100                                        RDDA1800
C         CHECK NORMAL MODE ANALYSIS DATA                               RDDA1810
C         NEIGVL MUST BE A MULTIPLE OF 5                                RDDA1820
      I = NEIGVL/5                                                      RDDA1830
      IF (    I    *5 .NE. NEIGVL ) NEIGVL = 5*(I+1)                    RDDA1840
      WRITE(6,230) INDNMA,INDWNM,IREDTO,NEIGVL                          RDDA1850
      WRITE(6,240) (AMASS(I),I=1,IREDTO)                                RDDA1860
      NEIGVL = NEIGVL + 6                                               RDDA1870
      IF ( NEIGVL .GT.IREDTO) CALL ERPNT1(EREGVL, 8, -1 )               RDDA1880
      IF (IREDTO.GT.102) CALL ERPNT1 (ERORD,4,-1)                       RDDA1890
      IF (IREDTO.EQ.NROW) GO TO 100                                     RDDA1900
      WRITE(6,250) (IRWKP(I),I=1,IREDTO)                                RDDA1910
      IF (IREDTO.GT.NROW) CALL ERPNT1 (ERRED,4,-1)                      RDDA1920
      J=IREDTO-1                                                        RDDA1930
      DO 90 I=1,J                                                       RDDA1940
90    IF (IRWKP(I).GE.IRWKP(I+1)) CALL ERPNT1 (ERRWKP,5,-1)             RDDA1950
100   CONTINUE                                                          RDDA1960
      IF (INDRKT.EQ.0) GO TO 130                                        RDDA1970
C                                                                       RDDA1980
C         READ STRUCTURAL STIFFNESS MATRIX DATA AND LOCAL STIFFNESS     RDDA1990
C                            AND TRANSFORMATION MATRICES                RDDA2000
      REWIND 9                                                          RDDA2010
      READ (9) NROW1                                                    RDDA2020
      IF (NROW.NE.NROW1) CALL ERPNT1 (ERFROW,5,1)                       RDDA2030
      NROW1=NROW1+1                                                     RDDA2040
      READ (9) (IROWL(I),I=1,NROW1)                                     RDDA2050
      JJ=1                                                              RDDA2060
      DO 110 I=1,NROW                                                   RDDA2070
      II=IROWL(I+1)-1                                                   RDDA2080
      READ (9) (STRSTF(J),J=JJ,II)                                      RDDA2090
110   JJ=II+1                                                           RDDA2100
      READ (9) STRSTF(JJ)                                               RDDA2110
      IF (INDNMA.NE.0) GO TO 130                                        RDDA2120
C                                                                       RDDA2130
C         FILE 2 IS NOT NEEDED IF OVERLAY 3 IS NOT CALLED               RDDA2140
C                                                                       RDDA2150
      REWIND 2                                                          RDDA2160
      DO 120 I=1,NBAR                                                   RDDA2170
      READ (9) (TRANSM(J,1),J=1,9)                                      RDDA2180
      WRITE (2) (TRANSM(J,1),J=1,9)                                     RDDA2190
      READ (9) (STIFML(J,1),J=1,144)                                    RDDA2200
120   WRITE (2) (STIFML(J,1),J=1,144)                                   RDDA2210
      IF (IND1SL.NE.0) READ (9) (SOLVEC(I),I=1,NROW)                    RDDA2220
130   IF (NJOINT.LT.2) CALL ERPNT1 (ERJNL,4,-1)                         RDDA2230
```

```
      IF (NBAR.LT.1) CALL ERPNT1 (ERBN,4,-1)
      IF ( INDNMA .NE. 0 ) GO TO 138
      ERRTOL = ABS(ERRTOL)
      WRITE(6,220) ERRTOL,IND1SL,INDITR,INDPLS,INDRLX,INDWKT,
     1             MINRST,RELAXF,TMAX
      IF ( IND1SL .NE. 0 ) WRITE(6,260) (SOLVEC(I),I=1,NROW )
  138 CONTINUE
      CALL ERPNT2
      RETURN
C
  140 FORMAT ( 1XI4, 3E10.3,6I3,2I4 )
  150 FORMAT ( 1XI4, 4I3, 3E10.2 )
  160 FORMAT (1XI4, 3I3, A3, E8.3, 5E9.3, 2F5.2 )
  210 FORMAT (33H1STRUCTURAL ANALYSIS CONTROL DATA  //
     *        13H0GENERAL DATA    //
     *   9H INDRKT =,I5,7X,39H, 1 IMPLIES READ ALL MATRICES FROM TAPE /
     *   9H INDSFG =,I5,7X 37H, 0 IMPLIES WRITE GLOBAL BAR MATRICES /
     *   9H INDSFL =,I5,7X 36H, 0 IMPLIES WRITE LOCAL BAR MATRICES  /
     *   9H INDWKT =,I5,7X,37H, 1 IMPLIES SAVE ALL MATRICES ON TAPE  /
     *   9H INDWTS =,I5,7X,40H, 1 IMPLIES PRINT TOTAL STIFFNESS MATRIX /
     *   9H ISFDIM =,I7,5X,35H, MAX. STORAGE FOR STIFFNESS MATRIX  /
     *   )
  220 FORMAT (36H0DISPLACEMENT/ROTATION SOLUTION DATA //
     *   9H ERRTOL =,E10.3,2X30H, ITERATION SOLUTION TOLERANCE     /
     *   9H IND1SL =,I5,7X,41H, 1 IMPLIES AN INITIAL SOLUTION IN SOLVEC /
     *   9H INDITR =,I7,5X 32H, MAX. SOLUTION ITERATION CYCLES      /
     *   9H INDPLS =,I5,7X,31H, 1 IMPLIES CONSIDER PLASTICITY     /
     *   9H INDRLX =,I5,7X,28H, ITERATIVE SOLUTION METHOD /
     *   9H INDWKT =,I5,7X,31H, 1 IMPLIES SAVE SOLVEC ON TAPE /
     *   9H MINRST =,I5,7X,27H, MIN. ALLOWABLE RESTRAINTS         /
     *   9H RELAXF =,F7.2,5X20H, RELAXATION FACTOR  /
     *   9H TMAX   =,F10.3, 2X31H, ITERATION CP TERMINATION TIME  /
     *   )
  230 FORMAT (20H0MODAL ANALYSIS DATA          / /
     *   9H INDNMA =,I5,7X,26H, RUN NORMAL MODE ANALYSIS          /
     *   9H INDWNM =,I5,7X,35H, 1 IMPLIES WRITE MODE DATA ON TAPE /
     *   9H IREDTO =,I5,7X,25H, ORDER OF REDUCED SYSTEM  /
     *   9H NEIGVL =,I5,7X,31H, REQUIRED NON-RIGID BODY MODES  /
     *   )
  240 FORMAT  (56H AMASS IS THE DIAGONAL MASS VECTOR OF THE REDUCED SYST
     *EM / (21X10E11.3))
  250 FORMAT ( 54H IRWKP CONTAINS THE ROWS TO KEEP IN THE REDUCED SYSTEM
     *  / (21X10I11))
  260 FORMAT (30H SOLVEC IS THE SOLUTION VECTOR /  (21X10E11.3)    )
      END
```

```
      SUBROUTINE SHRPAN
      COMMON COM(30)
      EQUIVALENCE ( COM(  1), NJOINT )
      EQUIVALENCE ( COM(  4), NBAR   )
      EQUIVALENCE ( COM( 29), NSHPAN )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3    KS     (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4    BARJ  (  130), BARYM (  130), BARSM (  130), RKN    (  130),
     5    RKT    (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)
      COMMON/SP/IAPAN(30), IBPAN(30), ICPAN(30), IDPAN(30),
     1           VSPAN(30), ESPAN(30), TSPAN(30)
      DIMENSION       VABC(12), AB(3), BC(3), CD(3), DA(3)
      EQUIVALENCE(VABC(1),AB(1)),(VABC(4),BC(1)),(VABC(7),CD(1)),
     1           (VABC(10),DA(1))
      DIMENSION ERSP1(6), ERSP2(5)
      DATA ERSP2 / 10H A SHEAR P, 10HANEL USES , 10HAN UNDEFIN,
     1             10HED NODAL P, 10HOINT.      /
      DATA ERSP1 / 10H SHEAR PAN, 10HEL CORNER , 10HPOINTS DO ,
     1             10HNOT DEFINE, 10H A RECTANG, 10HLE.          /
      DATA IBLK3S /10H          /
      IF ( NSHPAN .EQ. 0 ) RETURN
C         CHECK SHEAR PANEL NODE POINT DEFINITIONS
      DO  10   I = 1, NSHPAN
      IF ( IAPAN(I) .GT. NJOINT ) CALL ERPNT1( ERSP2, 5, -1 )
      IF ( IBPAN(I) .GT. NJOINT ) CALL ERPNT1( ERSP2, 5, -1 )
      IF ( ICPAN(I) .GT. NJOINT ) CALL ERPNT1( ERSP2, 5, -1 )
      IF ( IDPAN(I) .GT. NJOINT ) CALL ERPNT1( ERSP2, 5, -1 )
   10 CONTINUE
      CALL ERPNT2
      DO  200  ISP = 1, NSHPAN
      IA = IAPAN(ISP )
      IB = IBPAN(ISP )
      IC = ICPAN(ISP )
      ID = IDPAN(ISP )
C         CALCULATE THE VECTORS AB, BC, CD, AND DA  TO CHECK
C                       CONDITIONS OF THE SHEAR PANEL
      AL = 0.0
      BL = 0.0
      CL = 0.0
      DO  20   I  =  1, 3
      AB(I) = RJXYZ(IB,I) - RJXYZ(IA,I)
      BC(I) = RJXYZ(IC,I) - RJXYZ(IB,I)
      CD(I) = RJXYZ(ID,I) - RJXYZ(IC,I)
      DA(I) = RJXYZ(IA,I) - RJXYZ(ID,I)
      AL  = AL + AB(I)**2
      BL  = BL + BC(I)**2
      CL  = CL + CD(I)**2
   20 CONTINUE
      AL  = SQRT( AL )
      BL  = SQRT( BL )
      CL  = SQRT( CL )
C
```

```
C         IS THIS SHEAR PANEL A RECTANGLE                               SHRP 560
C                                                                       SHRP 570
C              CHECK CORNER ANGLES (90)                                 SHRP 580
      DO  40   I = 1, 7, 3                                              SHRP 590
      CALL DOTX( ALL, VABC(I), VABC(I+3) )                              SHRP 600
      IF ( ABS( ALL) .LT. 1.E-6 ) GO TO 40                              SHRP 610
      CALL ERPNT1( ERSP1, 6, -1 )                                       SHRP 620
      GO TO 200                                                         SHRP 630
   40 CONTINUE                                                          SHRP 640
C              CHECK PLANE CONDITIONS                                   SHRP 650
      CALL CROSSX(DA, AB, BC )                                          SHRP 660
      CALL CROSSX(AB, BC, CD )                                          SHRP 670
      CALL DOTX  (BC(1), AB, AB )                                       SHRP 680
      CALL DOTX  (BC(2), DA, DA )                                       SHRP 690
      BC(1)=SQRT( BC(1))                                                SHRP 700
      BC(2) = SQRT( BC(2))                                              SHRP 710
      DO  60   I = 1, 3                                                 SHRP 720
      BC(3)=ABS(DA(I)/BC(2)-AB(I)/BC(1))                                SHRP 730
      CD(1) =  .5*BC(3) + 1.E-5                                         SHRP 740
      IF ( BC(3)/  CD(1)    .LE. .0001 )  GO TO 60                      SHRP 750
      CALL ERPNT1 ( ERSP1, 6, -1 )                                      SHRP 760
      GO TO 200                                                         SHRP 770
   60 CONTINUE                                                          SHRP 780
      IFRIST = 0                                                        SHRP 790
      NBAR = NBAR + 1                                                   SHRP 800
      IBARP(NBAR) = IA                                                  SHRP 810
      IBARQ(NBAR) = IC                                                  SHRP 820
      IBARR(NBAR) =-ID                                                  SHRP 830
   80 CONTINUE                                                          SHRP 840
      AREA =  (AL**2+BL**2)**1.5 * TSPAN(ISP)/(4.*AL*BL*(1.+VSPAN(ISP)))SHRP 850
      KS   (NBAR) = IBLK3S                                              SHRP 860
      BAREA(NBAR) = AREA                                                SHRP 870
      BARIN(NBAR) = 0.0                                                 SHRP 880
      BARIT(NBAR) = 0.0                                                 SHRP 890
      BARJ (NBAR) = 0.0                                                 SHRP 900
      BARSM(NBAR) = 0.0                                                 SHRP 910
      BARYM(NBAR) = ESPAN(ISP)                                          SHRP 920
      IF ( IFRIST .NE. 0 ) GO TO 200                                    SHRP 930
      IFRIST = 1                                                        SHRP 940
      NBAR = NBAR + 1                                                   SHRP 950
      IBARP(NBAR) = ID                                                  SHRP 960
      IBARQ(NBAR) = IB                                                  SHRP 970
      IBARR(NBAR) =-IA                                                  SHRP 980
      AL   = CL                                                         SHRP 990
      GO TO 80                                                          SHRP1000
  200 CONTINUE                                                          SHRP1010
      RETURN                                                            SHRP1020
      END                                                               SHRP1030
```

```
      SUBROUTINE CROSSX( S, A, B )                                      CROS  10
      DIMENSION A(1), B(1), S(1)                                        CROS  20
C          S .EQ. THE VECTOR OR CROSS PRODUCT OF VECTORS A AND B        CROS  30
      S(1) =  A(2)*B(3) - A(3)*B(2)                                     CROS  40
      S(2) = -A(1)*B(3) + A(3)*B(1)                                     CROS  50
      S(3) =  A(1)*B(2) - A(2)*B(1)                                     CROS  60
      RETURN                                                            CROS  70
      ENTRY DOTX                                                        CROS  80
C          S(1) .EQ. THE DOT OR INNER PRODUCT OF TWO VECTORS A AND B    CROS  90
      S(1) =  A(1)*B(1) + A(2)*B(2) + A(3)*B(3)                         CROS 100
      RETURN                                                            CROS 110
      END                                                               CROS 120
```

```
      OVERLAY ( SASLP, 2, 0 )

      PROGRAM STIFF
      COMMON COM(30)
      EQUIVALENCE ( COM (  24), INDNMA )
      EQUIVALENCE ( COM(  4 ), NBAR   )
      EQUIVALENCE ( COM ( 7 ), IBAR    )
      EQUIVALENCE ( COM(  9 ), INDSFL )
      EQUIVALENCE ( COM( 10 ), INDSFG )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130),
     3    KS     (   130), BAREA (   130), BARIN (   130), BARIT (   130),
     4    BARJ   (   130), BARYM (   130), BARSM (   130), RKN    (   130),
     5    RKT    (   130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88),
     6    DPRTIT(    88), IROWL (   445), STRSTF(1)
      DIMENSION ISTSTF(1)
      EQUIVALENCE ( STRSTF(1), ISTSTF(1))
C
C         SET UP STRUCTURAL STIFFNESS MATRIX
C
      CALL SETSTF
C
C         BUILD STRUCTURAL STIFFNESS MATRIX BY SUMMING BAR STIFFNESS
C                                                              MATRICES
      IF ( INDNMA .NE. 0 ) GO TO 6
      CALL WRSTD1
    6 CONTINUE
      DO 30 IBAR=1,NBAR
C         COMPUTE STIFFNESS MATRIX AND TRANSFORMATION MATRIX FOR BAR
C               NUMBER IBAR
      CALL STFTRN
C
C         SAVE LOCAL STIFFNESS M AND TRANSFORMATION M ON FILE TAPE2
C
      IF ( INDNMA .NE. 0 ) GO TO 8
      CALL WRSTDK
    8 CONTINUE
      IF (INDSFL.NE.0) GO TO 10
      CALL WRBDAT
10    CONTINUE
C         TRANSFORM BAR STIFFNESS MATRIX TO GLOBAL COORDINATE SYSTEM
      CALL TRASMK
      IF (INDSFG.NE.0) GO TO 20
      CALL BRSTRA
20    CONTINUE
C         PLACE TRANSFORMED BAR STIFFNESS MATRIX IN STRUCTURAL
C               STIFFNESS MATRIX
      CALL STORSM
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETSTF
      COMMON COM(30)
      EQUIVALENCE ( COM( 4), NBAR    )
      EQUIVALENCE ( COM ( 1 ), NJOINT )
      EQUIVALENCE ( COM( 8), NROW    )
      EQUIVALENCE ( COM ( 27),ISFDIM )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130),
     3    KS     (  130), BAREA (   130), BARIN (   130), BARIT (   130),
     4    BARJ   (  130), BARYM (   130), BARSM (   130), RKN    (   130),
     5    RKT    (  130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88),
     6    DPRTIT(    88), IROWL (   445), STRSTF(1)
      DIMENSION ILSET(74,74), ISTSTF(I)
      EQUIVALENCE ( STRSTF(1), ISTSTF(1), ILSET(1,1) )
      DIMENSION ILSETN(74)
      DIMFNSION  NZPRW(74)
      NULL=NJOINT/2
C
C         COMPILE AN ARRAY OF RELATFD NODAL POINTS TO SET-UP STIFFNESS M
C
      DO  5  I = 1, 74
    5 NZPRW(I) = 0
      DO 10 I=1,NULL
      ILSET(I,1)=I
   10 NZPRW(I) = 1
C
      DO 20 I=1,NBAR
      IP=IBARP(I)
      IQ=IBARQ(I)
      NRP=NZPRW(IP)+1
      NRQ=NZPRW(IQ)+1
      NZPRW(IP)=NRP
      NZPRW(IQ)=NRQ
      ILSET(IP,NRP)=IQ
20    ILSET(IQ,NRQ)=IP
C
      NULL=NULL+1
      DO 30 I=NULL,NJOINT
      NRP=NZPRW(I)+1
      NZPRW(I)=NRP
30    ILSET(I,NRP)=I
C
C         CALCULATE THE TOTAL NO. OF STORAGE WORDS NEEDED FOR THF
C              STIFFNESS MATRIX AND COMPARE WITH WHAT IS AVAILABLE
C
      NULL = 0
      DO  32   I = 1, NJOINT
   32 NULL = NULL +  NZPRW(I)
      NULL = NULL * 42 + NJOINT*6   + 2
      IF (NULL .LE. ISFDIM ) GO TO 34
      WRITE(6, 200) NULL, ISFDIM
      STOP
   34 CONTINUE
C
```

```
C
C          SORT ABOVE ARRAY BY NODAL POINT
C
      DO 90 I=1,NJOINT
      NRQ=NZPRW(I)
      IF (NRQ.EQ.1) GO TO 90
40    NRP=0
      DO 80 J=2,NRQ
      IF (ILSET(I,J-1)-ILSET(I,J)) 80,50,70
50    NRQ=NRQ-1
      NZPRW(I)=NRQ
      DO 60 L=J,NRQ
60    ILSET(I,L)=ILSET(I,L+1)
      GO TO 40
70    NULL=ILSET(I,J)
      ILSET(I,J)=ILSET(I,J-1)
      ILSET(I,J-1)=NULL
      NRP=1
80    CONTINUE
      IF (NRP.NE.0) GO TO 40
90    CONTINUE
C
C          WRITE ARRAY ON FILE 1
C
      REWIND 1
      DO 100 I=1,NJOINT
      NRQ=NZPRW(I)
100   WRITE (1) (ILSET(I,J),J=1,NRQ)
C
C          BUILD  STIFFNESS MATRIX STORAGE ARRAY
C
      REWIND 1
      NULL=1
      NROW=0
      DO 140 I=1,NJOINT
      NRP=NZPRW(I)
      NRQ=7*NRP+1
      READ (1) (ILSETN(J),J=1,NRP)
      NFIRST=NULL+1
      NROW=NROW+1
C
      L = NRQ + NULL - 1
      DO 110 J = NULL, L
110   STRSTF(J)=0.0
      ISTSTF(NULL)=-NROW
      IROWL(NROW)=NULL
      NULL=NFIRST
C
      DO 120 J=1,NRP
      ISTSTF(NULL)=(6*(ILSETN(J)-1)+1)
120   NULL=NULL+7
      NLAST=NULL-1
C
      DO 130 L=1,5
      NROW=NROW+1
      ISTSTF(NULL)=-NROW
```

```
      IROWL(NROW)=NULL                                                  SETS1120
      NULL=NULL+1                                                       SETS1130
      DO 130 J=NFIRST,NLAST                                             SETS1140
      STRSTF(NULL)=STRSTF(J)                                            SETS1150
130   NULL=NULL+1                                                       SETS1160
140   CONTINUE                                                          SETS1170
      ISTSTF(NULL)=-1                                                   SETS1180
      IROWL(NROW+1)=NULL                                                SETS1190
      RETURN                                                            SETS1200
  200 FORMAT (16H0**** ERROR ****, I6, 22H WORDS OF STORAGE ARE ,       SETS1210
     1         39HREQUIRED FOR THE STIFFNESS MATRIX (ONLY, I6    ,  /   SETS1220
     2  22X    45H WORDS OF STORAGE ARE AVAILABLE WITH PRESENT ,        SETS1230
     3         12HDIMENSIONING )                                        SETS1240
      END                                                               SETS1250
```

```
      SUBROUTINE STFTRN
      COMMON COM(30)
      EQUIVALENCE ( COM( 6), ITINJO )
      EQUIVALENCE ( COM( 7), IBAR   )
      DIMENSION S1DEM(144)
      EQUIVALENCE ( STIFML, S1DEM )
      COMMON / CMAIN /
     1     RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2     RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130),
     3     KS     (  130), BAREA (   130), BARIN (   130), BARIT (   130),
     4     BARJ   (  130), BARYM (   130), BARSM (   130), RKN    (   130),
     5     RKT    (  130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88),
     6     DPRTIT(    88), IROWL (   445), STRSTF(1)
      DIMENSION  RBARR(1)
      EQUIVALENCE ( IBARR(1), RBARR(1) )
      COMMON / OVER2 / STIFML(12,12), TRANSM(3,3)
      DATA I3BLK / 3H    /
      ITINJO=1
      IF (ABS(BARIT(IBAR))+ABS(BARIN(IBAR))+ABS(BARJ(IBAR)).FO.0) ITINJO
     1=0
      IP=IBARP(IBAR)
      IQ=IBARQ(IBAR)
      IR =IABS( IBARR(IBAR))
C
C         COMPUTE UNIT VECTORS ALONG  LOCAL AXIS FOR GLOBAL COORDINATE
C                                 TRANSFORMATION MATRIX
      X=RJXYZ(IQ,1)-RJXYZ(IP,1)
      Y=RJXYZ(IQ,2)-RJXYZ(IP,2)
      Z=RJXYZ(IQ,3)-RJXYZ(IP,3)
      BARLGT=SQRT(X*X+Y*Y+Z*Z)
      IF ( IBARR(IBAR) .LT. 0 ) RBARR(IBAR) = -BARLGT
      TRANSM(1,1)=X/BARLGT
      TRANSM(1,2)=Y/BARLGT
      TRANSM(1,3)=Z/BARLGT
      IF (ITINJO.NE.0) GO TO 20
      DO 10 I=2,3
      DO 10 J=1,3
   10 TRANSM(I,J)=0.0
      GO TO 30
   20 CONTINUE
      X=RJXYZ(IR,1)-RJXYZ(IP,1)
      Y=RJXYZ(IR,2)-RJXYZ(IP,2)
      Z=RJXYZ(IR,3)-RJXYZ(IP,3)
      DPD=TRANSM(1,1)*X+TRANSM(1,2)*Y+TRANSM(1,3)*Z
      DDR=SQRT(X*X+Y*Y+Z*Z-DPD*DPD)
      TRANSM(2,1)=(X-TRANSM(1,1)*DPD)/DDR
      TRANSM(2,2)=(Y-TRANSM(1,2)*DPD)/DDR
      TRANSM(2,3)=(Z-TRANSM(1,3)*DPD)/DDR
      TRANSM(3,1)=TRANSM(1,2)*TRANSM(2,3)-TRANSM(2,2)*TRANSM(1,3)
      TRANSM(3,2)=-TRANSM(1,1)*TRANSM(2,3)+TRANSM(2,1)*TRANSM(1,3)
      TRANSM(3,3)=TRANSM(1,1)*TRANSM(2,2)-TRANSM(2,1)*TRANSM(1,2)
C
C         COMPUTE STIFFNESS MATRIX
C
   30 CONTINUE
```

```
C                                                                       STFT 560
C          ZERO RIGHT HALF OF STIFFNESS MATRIX                          STFT 570
C                                                                       STFT 580
      DO 40 I=1,12                                                      STFT 590
      DO 40 J=I,12                                                      STFT 600
40    STIFML(I,J)=0.0                                                   STFT 610
C                                                                       STFT 620
C          FILL LEFT HALF OF STIFFNESS MATRIX AND THE DIAGONAL          STFT 630
C                                                                       STFT 640
      S1DEM(1)=BARYM(IBAR)*BAREA(IBAR)/BARLGT                           STFT 650
      S1DEM(37)=-S1DEM(1)                                               STFT 660
      S1DEM(40)=S1DEM(1)                                                STFT 670
      IF (ITINJO.EQ.0) GO TO 160                                        STFT 680
      S1DEM(14)=12.*BARYM(IBAR)*BARIN(IBAR)/BARLGT**3                   STFT 690
      S1DEM(27)=12.*BARYM(IBAR)*BARIT(IBAR)/BARLGT**3                   STFT 700
      S1DEM(50)=-S1DEM(14)                                              STFT 710
      S1DEM(53)=S1DEM(14)                                               STFT 720
      S1DEM(63)=-S1DEM(27)                                              STFT 730
      S1DEM(66)=S1DEM(27)                                               STFT 740
      S1DEM(79)=BARSM(IBAR)*BARJ(IBAR)/BARLGT                           STFT 750
      S1DEM(87)=-6.*BARYM(IBAR)*BARIT(IBAR)/BARLGT**2                   STFT 760
      S1DEM(90)=-S1DEM(87)                                              STFT 770
      S1DEM(92)=4.*BARYM(IBAR)*BARIT(IBAR)/BARLGT                       STFT 780
      S1DEM(98)=6.*BARYM(IBAR)*BARIN(IBAR)/BARLGT**2                    STFT 790
      S1DEM(101)=-S1DEM(98)                                             STFT 800
      S1DEM(105)=4.*BARYM(IBAR)*BARIN(IBAR)/BARLGT                      STFT 810
      S1DEM(115)=-S1DEM(79)                                             STFT 820
      S1DEM(118)=S1DEM(79)                                              STFT 830
      S1DEM(123)=S1DEM(87)                                              STFT 840
      S1DEM(126)=-S1DEM(87)                                             STFT 850
      S1DEM(128)=2.*BARYM(IBAR)*BARIT(IBAR)/BARLGT                      STFT 860
      S1DEM(131)=S1DEM(92)                                              STFT 870
      S1DEM(134)=S1DEM(98)                                              STFT 880
      S1DEM(137)=-S1DEM(98)                                             STFT 890
      S1DEM(141)=2.*BARYM(IBAR)*BARIN(IBAR)/BARLGT                      STFT 900
      S1DEM(144)=S1DEM(105)                                             STFT 910
      IF (KS(IBAR).EQ.I3BLK) GO TO 160                                  STFT 920
C          DECODE KS(IBAR)                                              STFT 930
C                IR = 1 IMPLIES SHEAR STRAIN IS INCLUDED                STFT 940
C                IP = 1 FREE 5P.      IP = 2 FREE 5Q                    STFT 950
C                IQ = 1 FREE 6P.      IQ = 2 FREE 6Q                    STFT 960
      DECODE (3,180,KS(IBAR) )IR,IP,IQ                                  STFT 970
      X=BARSM(IBAR)*BAREA(IBAR)/BARLGT                                  STFT 980
      IF (RKN(IBAR).EQ.0.0) GO TO 50                                    STFT 990
      AN=STIFML(2,2)*RKN(IBAR)/X                                        STFT1000
      GO TO 60                                                          STFT1010
50    AN=0.0                                                            STFT1020
60    CONTINUE                                                          STFT1030
      IF (RKT(IBAR).EQ.0.0) GO TO 70                                    STFT1040
      AT=STIFML(3,3)*RKT(IBAR)/X                                        STFT1050
      GO TO 80                                                          STFT1060
70    AT=0.0                                                            STFT1070
80    CONTINUE                                                          STFT1080
      IF (IR.EQ.1) GO TO 120                                            STFT1090
      IF (IP.EQ.0) GO TO 120                                            STFT1100
C                                                                       STFT1110
```

```
C          Y        MOMENT FREE                                         STFT1120
C                                                                       STFT1130
      X=1./(4.+AT)                                                      STFT1140
      STIFML(3,3)=X*STIFML(3,3)                                         STFT1150
      STIFML(3,6)=X*STIFML(3,6)                                         STFT1160
      STIFML(6,6)=X*STIFML(6,6)                                         STFT1170
      STIFML(8,11)=0.0                                                  STFT1180
      IF (IP.EQ.2) GO TO 90                                             STFT1190
      I=11                                                              STFT1200
      J=8                                                               STFT1210
      GO TO 100                                                         STFT1220
90    I=8                                                               STFT1230
      J=11                                                              STFT1240
100   CONTINUE                                                          STFT1250
      Y=X+X                                                             STFT1260
      STIFML(3,I)=Y*STIFML(3,I)                                         STFT1270
      STIFML(6,I)=Y*STIFML(6,I)                                         STFT1280
      Y=Y+X                                                             STFT1290
      STIFML(I,I)=Y*STIFML(6,I)                                         STFT1300
      STIFML(3,J)=0.0                                                   STFT1310
      STIFML(6,J)=0.0                                                   STFT1320
      STIFML(J,J)=0.0                                                   STFT1330
110   IF (IQ.NE.0) GO TO 130                                            STFT1340
C                                                                       STFT1350
C          Z MOMENT NOT FREE                                            STFT1360
C                                                                       STFT1370
      IF (RKN(IBAR).EQ.0.0) GO TO 160                                   STFT1380
      X=1./(1.+AN)                                                      STFT1390
      STIFML(2,2)=X*STIFML(2,2)                                         STFT1400
      STIFML(2,5)=X*STIFML(2,5)                                         STFT1410
      STIFML(2,9)=X*STIFML(2,9)                                         STFT1420
      STIFML(2,12)=X*STIFML(2,12)                                       STFT1430
      STIFML(5,5)=X*STIFML(5,5)                                         STFT1440
      STIFML(5,9)=X*STIFML(5,9)                                         STFT1450
      STIFML(5,12)=X*STIFML(5,12)                                       STFT1460
      X=1.-3./(4.*(1.+1./AN))                                           STFT1470
      STIFML(9,9)=X*STIFML(9,9)                                         STFT1480
      STIFML(9,12)=(1.-3./(2.*(1.+1./AN)))*STIFML(9,12)                 STFT1490
      STIFML(12,12)=X*STIFML(12,12)                                     STFT1500
      GO TO 160                                                         STFT1510
C                                                                       STFT1520
C          Y MOMENT NOT FREE                                            STFT1530
C                                                                       STFT1540
120   IF (RKT(IBAR).EQ.0.0) GO TO 110                                   STFT1550
      X=1./(1.+AT)                                                      STFT1560
      STIFML(3,3)=X*STIFML(3,3)                                         STFT1570
      STIFML(3,6)=X*STIFML(3,6)                                         STFT1580
      STIFML(3,8)=X*STIFML(3,8)                                         STFT1590
      STIFML(3,11)=X*STIFML(3,11)                                       STFT1600
      STIFML(6,6)=X*STIFML(6,6)                                         STFT1610
      STIFML(6,8)=X*STIFML(6,8)                                         STFT1620
      STIFML(6,11)=X*STIFML(6,11)                                       STFT1630
      X=1.-3./(4.*(1.+1./AT))                                           STFT1640
      STIFML(8,8)=X*STIFML(8,8)                                         STFT1650
      STIFML(8,11)=(1.-3./(2.*(1.+1./AT)))*STIFML(8,11)                 STFT1660
      STIFML(11,11)=X*STIFML(11,11)                                     STFT1670
```

```
      GO TO 110                                                         STFT1680
C                                                                       STFT1690
C          Z MOMENT FREE                                                STFT1700
C                                                                       STFT1710
130   X=1./(4.+AN)                                                      STFT1720
      STIFML(2,2)=X*STIFML(2,2)                                         STFT1730
      STIFML(2,5)=X*STIFML(2,5)                                         STFT1740
      STIFML(5,5)=X*STIFML(5,5)                                         STFT1750
      STIFML(9,12)=0.0                                                  STFT1760
      IF (IQ.EQ.2) GO TO 140                                            STFT1770
      I=12                                                              STFT1780
      J=9                                                               STFT1790
      GO TO 150                                                         STFT1800
140   I=9                                                               STFT1810
      J=12                                                              STFT1820
150   CONTINUE                                                          STFT1830
      Y=X+X                                                             STFT1840
      STIFML(2,I)=Y*STIFML(2,I)                                         STFT1850
      STIFML(5,I)=Y*STIFML(5,I)                                         STFT1860
      Y=Y+X                                                             STFT1870
      STIFML(I,I)=Y*STIFML(12,12)                                       STFT1880
      STIFML(2,J)=0.0                                                   STFT1890
      STIFML(5,J)=0.0                                                   STFT1900
      STIFML(J,J)=0.0                                                   STFT1910
C                                                                       STFT1920
C          FILL LEFT  HALF OF STIFFNESS MATRIX                          STFT1930
C                                                                       STFT1940
160   CONTINUE                                                          STFT1950
      DO 170 I=1,11                                                     STFT1960
      II=I+1                                                            STFT1970
      DO 170 J=II,12                                                    STFT1980
170   STIFML(J,I)=STIFML(I,J)                                           STFT1990
      RETURN                                                            STFT2000
C                                                                       STFT2010
180   FORMAT (3I1)                                                      STFT2020
      END                                                               STFT2030
```

```
      SUBROUTINE STORSM
      COMMON COM(30)
      EQUIVALENCE ( COM( 6), ITINJO )
      EQUIVALENCE ( COM( 7), IBAR   )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3    KS    (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4    BARJ  (  130), BARYM (  130), BARSM (  130), RKN   (  130),
     5    RKT   (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)
      DIMENSION ISTSTF(1)
      EQUIVALENCE ( STRSTF(1), ISTSTF(1))
      COMMON / OVER2 / STIFML(12,12), TRANSM(3,3)
      DIMENSION IRC13(6), IRC24(6)
      DATA IRC13 / 1, 2, 3, 7, 8, 9 /
      DATA IRC24 / 4, 5, 6, 10, 11, 12 /
      ISP=(IBARP(IBAR)-1)*6+1
      ISQ=(IBARQ(IBAR)-1)*6+1
C
C         FIND INCREMENT FROM START OF ROW P TO COLUMNS P AND Q
C                                CALL  IPP AND IPQ
      J1=0
      II=IROWL(ISP)
      I1=II+1
   10 IF (ISTSTF(I1).NE.ISP) GO TO 20
      IPP=I1-II
      J1=J1+1
      GO TO 30
   20 IF (ISTSTF(I1).NE.ISQ) GO TO 30
      IPQ=I1-II
      J1=J1+1
   30 I1=I1+7
      IF (J1.NE.2) GO TO 10
C
C         ADD  TO  THE 6 P ROWS
C
      IF (ITINJO.NE.0) GO TO 40
      I1=II+IPP+1
      STRSTF(I1)=STRSTF(I1)+STIFML(1,1)
      I1=II+IPQ+1
      STRSTF(I1)=STRSTF(I1)+STIFML(1,4)
      GO TO 70
   40 CONTINUE
      DO 60 I=1,6
      IP=II+IPP
      IQ=II+IPQ
      I1=IRC13(I)
      DO 50 J=1,6
      J1=IRC13(J)
      JJ=IP+J
      STRSTF(JJ)=STRSTF(JJ)+STIFML(I1,J1)
      J1=IRC24(J)
      JJ=IQ+J
   50 STRSTF(JJ)=STRSTF(JJ)+STIFML(I1,J1)
```

```
      J1=ISP+I                                                          STOR 560
60    II=IROWL(J1)                                                      STOR 570
C                                                                       STOR 580
C         FIND INCREMENT FROM START OF ROW Q TO COLUMNS P AND Q         STOR 590
C                             CALL IPP                                  STOR 600
70    CONTINUE                                                          STOR 610
      J1=0                                                              STOR 620
      II=IROWL(ISQ)                                                     STOR 630
      I1=II+1                                                           STOR 640
80    IF (ISTSTF(I1).NE.ISQ) GO TO 90                                   STOR 650
      IQQ=I1-II                                                         STOR 660
      J1=J1+1                                                           STOR 670
      GO TO 100                                                         STOR 680
90    IF (ISTSTF(I1).NE.ISP) GO TO 100                                  STOR 690
      IQP=I1-II                                                         STOR 700
      J1=J1+1                                                           STOR 710
100   I1=I1+7                                                           STOR 720
      IF (J1.NE.2) GO TO 80                                             STOR 730
C                                                                       STOR 740
C         ADD  TO THE 6 Q ROWS                                          STOR 750
C                                                                       STOR 760
      IF (ITINJO.NE.0) GO TO 110                                        STOR 770
      I1=II+IQQ+1                                                       STOR 780
      STRSTF(I1)=STRSTF(I1)+STIFML(4,1)                                 STOR 790
      I1=II+IQP+1                                                       STOR 800
      STRSTF(I1)=STRSTF(I1)+STIFML(4,4)                                 STOR 810
      RETURN                                                            STOR 820
110   CONTINUE                                                          STOR 830
      DO 130 I=1,6                                                      STOR 840
      IQ=II+IQQ                                                         STOR 850
      IP=II+IQP                                                         STOR 860
      I1=IRC24(I)                                                       STOR 870
      DO 120 J=1,6                                                      STOR 880
      J1=IRC13(J)                                                       STOR 890
      JJ=IP+J                                                           STOR 900
      STRSTF(JJ)=STRSTF(JJ)+STIFML(I1,J1)                               STOR 910
      J1=IRC24(J)                                                       STOR 920
      JJ=IQ+J                                                           STOR 930
120   STRSTF(JJ)=STRSTF(JJ)+STIFML(I1,J1)                               STOR 940
      J1=ISQ+I                                                          STOR 950
130   II=IROWL(J1)                                                      STOR 960
      RETURN                                                            STOR 970
      END                                                               STOR 980
```

```
      SUBROUTINE TRASMK
      COMMON COM(30)
      EQUIVALENCE ( COM( 6), ITINJO )
      COMMON / OVER2 / STIFML(12,12), TRANSM(3,3)
      DIMENSION A(12), IJ(4), IJC(12)
      DIMENSION IRC(12)
      EQUIVALENCE(AXF, STIFML(1,1))
      DATA IJ / 0,3,6,9 /
      DATA IJC / 3*0, 3*3, 3*6, 3*9 /
      DATA IRC / 1,2,3,1,2,3,1,2,3,1,2,3 /
      ITINJO=1
C
C         IF THERE IS BENDING GO TO 8
C
      IF (ITINJO.NE.0) GO TO 20
      DO 10 I=1,3
      DO 10 J=1,3
      AHOLD=AXF*TRANSM(1,J)*TRANSM(1,I)
      STIFML(I,J)=AHOLD
      STIFML(I,J+3)=-AHOLD
      STIFML(I+3,J)=-AHOLD
   10 STIFML(I+3,J+3)=AHOLD
      RETURN
   20 CONTINUE
C
C         MULTIPLY TRANSFORMATION MATRIX TIMES STIFFNESS MATRIX
C                   STORE RESULT IN STIFFNESS MATRIX
C
      DO 40 K=1,12
      ICR=0
      DO 30 IC=1,4
      DO 30 I=1,3
      ICR=ICR+1
      A(ICR)=0.0
      DO 30 J=1,3
      JJ=J+IJ(IC)
      IF (STIFML(K,JJ).EQ.0.) GO TO 30
      A(ICR)=A(ICR)+STIFML(K,JJ)*TRANSM(J,I)
   30 CONTINUE
      DO 40 J=1,12
   40 STIFML(K,J)=A(J)
C
C
C         MULTIPLY THE RESULT OF THE ABOVE TIMES THE TRANSPOSE OF THE
C                   TRANSFORMATION MATRIX
C
      DO 60 K=1,12
      DO 50 ICR=K,12
      A(ICR)=0.0
      I=IRC(ICR)
      DO 50 J=1,3
      JJ=J+IJC(ICR)
   50 A(ICR)=A(ICR)+STIFML(JJ,K)*TRANSM(J,I)
      DO 60 J=K,12
   60 STIFML(J,K)=A(J)
      DO 70 I=1,11
```

```
      IC=I+1                                                            TRAS 570
      DO 70 J=IC,12                                                     TRAS 580
70    STIFML(I,J)=STIFML(J,I)                                           TRAS 590
C                                                                       TRAS 600
C           AS A RESULT OF THE ABOVE, STIFML CONTAINS THE STIFFNESS     TRAS 610
C                 MATRIX TRANSFORMED TO THE GLOBAL  COORDINATE SYSTEM   TRAS 620
C                                                                       TRAS 630
      RETURN                                                            TRAS 640
      END                                                               TRAS 650
```

```
      SUBROUTINE WRBDAT
      COMMON COM(30)
      EQUIVALENCE ( COM( 7), IBAR    )
      COMMON / OVER2 / STIFML(12,12), TRANSM(3,3)
      COMMON / CMAIN /
     1     RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2     RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3     KS     (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4     BARJ  (  130), BARYM (  130), BARSM (  130), RKN    (  130),
     5     RKT   (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6     DPRTIT(   88), IROWL (  445), STRSTF(1)
C
C         IF INDSFL .NE. 0, PRINT LOCAL STIFFNESS MATRIX AND
C               TRANSFORMATION MATRIX FOR BAR(I)
C
      WRITE (6,10) IBAR,IBARP(IBAR),IBARQ(IBAR),((STIFML(I,J),J=1,12),I=
     11,12),((TRANSM(I,J),J=1,3),I=1,3)
      RETURN
C
C         IF INDSFG .NE. 0, PRINT THE TRANSFORMED LOCAL STIFFNESS MATRIX
C
      ENTRY BRSTRA
      WRITE (6,20) ((STIFML(I,J),J=1,12),I=1,12)
      RETURN
C
10    FORMAT (13H1 BAR NUMBER I4/,25H0 POINT P IS NODAL POINT I4/,25H  P
     1OINT Q IS NODAL POINT I4/,18H0 STIFFNESS MATRIX//,12(12E11.3,/),23
     2H0 TRANSFORMATION MATRIX//,3(3E11.3,/))
20    FORMAT (30H0 TRANSFORMED STIFFNESS MATRIX//,,12(12E11.3,/))
      END
```

```
      SUBROUTINE WRSTDK
      COMMON / OVER2 / STIFML(12,12), TRANSM(3,3)
C
C         WRITE LOCAL STIFFNESS AND ASSOCIATED TRANSFORMATION
C               MATRICES ON FILE 2 FOR LATER USE IN FMBARS (AND, IF
C               INDWKT .NE. 0, IN PANDTK)
C
      WRITE (2) ((TRANSM(I,J),J=1,3),I=1,3)
      WRITE (2) ((STIFML(I,J),J=1,12),I=1,12)
      RETURN
      ENTRY WRSTD1
      REWIND 2
      RETURN
      END
```

```
      OVERLAY ( SASLP, 3, 0 )                                           OV30  10

      PROGRAM FINIAL                                                    FINI  10
      COMMON COM(30)                                                    FINI  20
      EQUIVALENCE ( COM(16), INDWKT )                                   FINI  30
      EQUIVALENCE ( COM( 8), NROW   )                                   FINI  40
      COMMON / CMAIN /                                                  FINI  50
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),   FINI  60
     2    RMTVCT(74,3), IBARP ( 130), IBARQ (  130), IBARR (  130),     FINI  70
     3    KS    (  130), BAREA (  130), BARIN (  130), BARIT (  130),   FINI  80
     4    BARJ  (  130), BARYM (  130), BARSM (  130), RKN   (  130),   FINI  90
     5    RKT   (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),   FINI 100
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)                       FINI 110
C                                                                       FINI 120
C         PANDTK, PRINTS THE NON ZERO ELEMENTS OF THE STRUCTURAL        FINI 130
C              STIFFNESS MATRIX AND CONDENSES THE BLOCK STORAGE WHERE   FINI 140
C                                                             POSSIBLE  FINI 150
      CALL PANDTK                                                       FINI 160
C                                                                       FINI 170
C         SOLVE                                                         FINI 180
C                                                                       FINI 190
      CALL SOLVE                                                        FINI 200
      IF (INDWKT.EQ.0) GO TO 10                                         FINI 210
      WRITE (9) (SOLVEC(I),I=1,NROW)                                    FINI 220
      END FILE 9                                                        FINI 230
10    CONTINUE                                                          FINI 240
C                                                                       FINI 250
C         CALCULATE LOCAL FORCE-MOMENT VECTORS FOR EACH BAR             FINI 260
C                                                                       FINI 270
      CALL FMBARS                                                       FINI 280
      RETURN                                                            FINI 290
      END                                                               FINI 300
```

```
      SUBROUTINE PANDTK
      COMMON COM(30)
      EQUIVALENCE ( COM( 8), NROW    )
      EQUIVALENCE ( COM(16), INDWKT )
      EQUIVALENCE ( COM(  4 ), NBAR    )
      EQUIVALENCE ( COM ( 28), INDWTS )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130),
     3    KS     (  130), BAREA (   130), BARIN (   130), BARIT (   130),
     4    BARJ   (  130), BARYM (   130), BARSM (   130), RKN    (   130),
     5    RKT    (  130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88),
     6    DPRTIT(    88), IROWL (   445), STRSTF(1)
      DIMENSION STF(519), ISTF(519)
      DIMENSION  ISTSTF(1), STFDIA(444), STIFML(12,12), TRANSM(3,3)
      EQUIVALENCE ( STRSTF(1), ISTSTF(1))
      EQUIVALENCE ( BAREA (1), STFDIA(1))
      EQUIVALENCE ( RJXYZ(1,1),STIFML(1,1))
      EQUIVALENCE ( RJXYZ(50,2),TRANSM(1,1)   )
      COMMON / OVER3 / COM3(519)
      EQUIVALENCE ( COM3( 1 ), STF, ISTF )
      LDIA = 0
      LDIAS = 0
      IF (INDWKT.EQ.0) GO TO 10
C
C         WRITE THE NO. OF ROWS AND THEIR LOCATION IN STIFFNESS MATRIX
C                    ON FILE 9
C
      REWIND 9
      WRITE (9) NROW
      J=NROW+1
      WRITE (9) (IROWL(I),I=1,J)
10    CONTINUE
C
C         PRINT NON ZERO ROW ELEMENTS AND DECIDE WHETHER THE ROW
C               SHOULD BE STORED IN BLOCK OR ELEMENT FORMAT
C
      LOCSTF=1
      DO 110 I=1,NROW
      ISUMNZ=0
      ISUMGP=0
20    LOCSTF=LOCSTF+1
      IF (ISTSTF(LOCSTF).LT.0) GO TO 50
      IC=ISTSTF(LOCSTF)-1
      ISUMGP=ISUMGP+1
      DO 40 J=1,6
      LOCSTF=LOCSTF+1
      IF (ABS(STRSTF(LOCSTF)).LT.1.E-9) GO TO 40
      ICC=IC+J
      IF ( INDWTS .NE.  0     )  CALL WRSTRK ( I, ICC, STRSTF(LOCSTF))
      IF (I.NE.ICC) GO TO 30
      LDIA=LOCSTF
      GO TO 40
30    ISUMNZ=ISUMNZ+1
40    CONTINUE
```

```
      GO TO 20                                                          PAND 560
50    CONTINUE                                                          PAND 570
C                                                                       PAND 580
C         IF INDWKT .NE. 0, WRITE STIFFNESS MATRIX IN BLOCK FORMAT      PAND 590
C                   ON FILE 9 TO SAVE FOR FUTURE RUNS OF THE PROGRAM    PAND 600
C                                                                       PAND 610
C                                                                       PAND 620
      LCSTF1=IROWL(I)                                                   PAND 630
      IF (INDWKT.EQ.0) GO TO 60                                         PAND 640
      NULL=LOCSTF-1                                                     PAND 650
      WRITE (9) (STRSTF(J),J=LCSTF1,NULL)                               PAND 660
60    CONTINUE                                                          PAND 670
C                                                                       PAND 680
C         REMOVE THE DIAGONAL ELEMENTS OF THE STRUCTURAL STIFFNESS      PAND 690
C                   MATRIX, AND SAVE THEM IN THE STFDIA ARRAY           PAND 700
C                                                                       PAND 710
      IF ( LDIA .EQ. LDIAS ) GO TO 70                                   PAND 720
      LDIAS = LDIA                                                      PAND 730
      STFDIA(I)=STRSTF(LDIA)                                            PAND 740
      STRSTF(LDIA)=0.0                                                  PAND 750
      GO TO 80                                                          PAND 760
70    STFDIA(I)=0.0                                                     PAND 770
80    CONTINUE                                                          PAND 780
      SUMNZ = ISUMNZ                                                    PAND 790
      SUMGP = ISUMGP                                                    PAND 800
      IF ( SUMNZ / SUMGP  .GT. 3. ) GO TO 110                           PAND 810
C                                                                       PAND 820
C         IF HALF OR MORE OF THE ROW ELEMENTS ARE ZERO, REMOVE THEM     PAND 830
C                   AND PLACE THAT ROW IN ELEMENT FORMAT                PAND 840
C                                                                       PAND 850
      NULL=1                                                            PAND 860
      DO 90 J=1,ISUMGP                                                  PAND 870
      LCSTF1=LCSTF1+1                                                   PAND 880
      IC=ISTSTF(LCSTF1)-1                                               PAND 890
      DO 90 K=1,6                                                       PAND 900
      LCSTF1=LCSTF1+1                                                   PAND 910
      IF (ABS(STRSTF(LCSTF1)).LT.1.E-9) GO TO 90                        PAND 920
      NULL=NULL+1                                                       PAND 930
      ISTF(NULL)=IC+K                                                   PAND 940
      NULL=NULL+1                                                       PAND 950
      STF(NULL)=STRSTF(LCSTF1)                                          PAND 960
90    CONTINUE                                                          PAND 970
      ISTF(1)=-NULL/2                                                   PAND 980
      LCSTF1=IROWL(I)                                                   PAND 990
      DO 100 J=1,NULL                                                   PAND1000
      LCSTF1=LCSTF1+1                                                   PAND1010
100   STRSTF(LCSTF1)=STF(J)                                             PAND1020
110   CONTINUE                                                          PAND1030
C                                                                       PAND1040
C         PRINT LAST LINE IN WRSTRK                                     PAND1050
C                                                                       PAND1060
      IF ( INDWTS .NE. 0 ) CALL WRSTRK(-1,-1,1.)                        PAND1070
C                                                                       PAND1080
C         IF INDWKT .NE. 0, THEN THE LOCAL STIFFNESS AND TRANSFORMATION PAND1090
C                   MATRICES MUST BE SAVED ON FILE 9                    PAND1100
C                                                                       PAND1110
```

```
      IF (INDWKT.EQ.0) GO TO 130                                        PAND1120
      REWIND 2                                                          PAND1130
      DO 120 I=1,NBAR                                                   PAND1140
      READ (2) (TRANSM(J,1),J=1,9)                                      PAND1150
      WRITE (9) (TRANSM(J,1),J=1,9)                                     PAND1160
      READ (2) (STIFML(J,1),J=1,144)                                    PAND1170
  120 WRITE (9) (STIFML(J,1),J=1,144)                                   PAND1180
  130 RETURN                                                            PAND1190
      END                                                               PAND1200
```

```
      SUBROUTINE SOLVE
      COMMON COM(30)
      EQUIVALENCE ( COM( 1), NJOINT )
      EQUIVALENCE ( COM( 8), NROW   )
      EQUIVALENCE ( COM(11), INDISL )
      EQUIVALENCE ( COM(12), INDITR )
      EQUIVALENCE ( COM(13), ERRTOL )
      EQUIVALENCE ( COM(14), RELAXF )
      EQUIVALENCE ( COM(15), INDRLX )
      EQUIVALENCE ( COM(18), INDPLS )
      EQUIVALENCE ( COM(19), MINRST )
      EQUIVALENCE ( COM(26), TMAX )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3    KS     (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4    BARJ  (  130), BARYM (  130), BARSM (  130), RKN    (  130),
     5    RKT   (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)
      DIMENSION SOLVC2(444), ISTSTF(1), STFDIA(444)
      DIMENSION RCG(444), APCG(444)
      EQUIVALENCE (  RJXYZ(1,1), NONROW(1) )
      EQUIVALENCE (  FRCVCT(1,1), SOLVC2(1) )
      EQUIVALENCE (  STRSTF(1 ),  ISTSTF(1))
      EQUIVALENCE (  BAREA (1 ),  STFDIA(1))
      EQUIVALENCE ( BARJ(50), RCG(1) )
      COMMON / OVER3 / FORCMG(1)
      DIMENSION ALP(6), NONROW(300)
      DIMENSION ERRFMN(6)
      DIMENSION ERMRST(5)
      DIMENSION ERDIA(5)
      DIMENSION ERRL(6)
      DATA ERRL /50H THE VARIABLE ASSOCIATED WITH THIS ROW HAS A CONST,
     1           10HANT VALUE      /
      DATA ERDIA /10H THE DIAGO, 10HNAL ELEMEN, 10HT FOR THIS,
     1            10H ROW CAN N, 10HOT BE ZERO  /
      DATA ERMRST /10H THE NUMBE, 10HR OF RESTR, 10HAINTS IS L,
     1             10HESS THAN M, 10HINRST       /
      DATA ALP /10HX--FORCE   , 10HY--FORCE   , 10HZ--FORCE   ,
     1          10HX--MOMENT  , 10HY--MOMENT  , 10HZ--MOMENT  /
      DATA ERRFMN / 10H TOO MANY , 10HROWS ARE B, 10HEING REMOV,
     1               10HED FOR THE, 10H SIZE OF N, 10HONROW            /
      WRITE (6,420)
C
C          SET-UP MINIMUM ERROR FOR THE AITKEN DELTA SQUARED PROCESS
C
      IF (INDRLX.EQ.0) ERRTL2=AMIN1(ERRTOL*ERRTOL,1.E-10)
C
C         CALCULATE FORCE AND MOMENT VECTOR
C
      DO 10 I=1,300
10    NONROW(I)=0
      DO 20 I=1,NROW
20    FORCMG(I)=0.0
      NULL=0
```

```
      DO 80 I=1,NJOINT
      IC=IJALFM(I,1)
      IF (IC.NE.0) GO TO 30
      NULL=NULL+3
      GO TO 50
30    DO 40 J=1,3
      NULL=NULL+1
40    FORCMG(NULL)=FRCVCT(IC,J)
50    IC=IJALFM(I,2)
      IF (IC.NE.0) GO TO 60
      NULL=NULL+3
      GO TO 80
60    DO 70 J=1,3
      NULL=NULL+1
70    FORCMG(NULL)=RMTVCT(IC,J)
80    CONTINUE
C
C          STORE ROW NUMBERS NOT REQUIRED IN THE SOLUTION IN NONROW
C          INDISL = 0 IMPLIES NO INITIAL SOLUTION IS AVAILABLE
C
      INDISL = 0
      NULL=0
      J=0
      I=0
      DO 130 K=1,NJOINT
      DO 130 L=1,6
      I=I+1
      SOLVC2(I)=SOLVEC(I)
      IF (IJRFM(K,L).EQ.0) GO TO 100
      NULL=NULL+1
      IF (NULL.GT.300) CALL ERPNT1 (ERRFMN,6,1)
      NONROW(NULL)=I
      IC=IABS(IJRFM(K,L))
      IF (DPRTIT(IC).EQ..0) GO TO 90
      INDISL = 1
      SOLVEC(I)=DPRTIT(IC)
      SOLVC2(I)=DPRTIT(IC)
      GO TO 130
90    SOLVEC(I)=0.0
      SOLVC2(I)=0.0
      GO TO 130
100   IF (STFDIA(I).EQ.0.0) GO TO 110
      IF ( INDRLX .EQ. 2 ) GO TO 130
      IF (INDISL.NE.0) GO TO 130
      GO TO 120
110   CONTINUE
      J=1
      WRITE (6,430) I
      GO TO 130
120   CONTINUE
      SOLVEC(I)=FORCMG(I)/STFDIA(I)
      SOLVC2(I)=SOLVEC(I)
130   CONTINUE
      IF (J.NE.0) STOP
      NNONRW=NULL
C
```

```
C                                                                       SOLV1120
C         ITERATION LOOP                                                SOLV1130
C                                                                       SOLV1140
140   IF (NNONRW.LT.MINRST) CALL ERPNT1 (ERMRST,5,1)                    SOLV1150
      IF ( INDRLX .NE. 2 ) GO TO 1010                                   SOLV1160
      IF ( INDISL .EQ. 1 ) INDISL = 1                                   SOLV1170
      FVMAX = 1.E-4                                                     SOLV1180
      DO  1000 I = 1, NROW                                              SOLV1190
      IF ( FVMAX .LT. ABS(FORCMG(I))) FVMAX = ABS(FORCMG(I) )           SOLV1200
      APCG(I) = 0.0                                                     SOLV1210
      RCG(I) = FORCMG(I)                                                SOLV1220
      IF ( INDISL .NE. 0 ) GO TO 1000                                   SOLV1230
      SOLVC2(I) = 0.0                                                   SOLV1240
      SOLVEC(I) = FORCMG(I)                                             SOLV1250
 1000 CONTINUE                                                          SOLV1260
 1010 CONTINUE                                                          SOLV1270
      ITRNO=0                                                           SOLV1280
      IUPDAT=4                                                          SOLV1290
      INDUPD=0                                                          SOLV1300
      INDOPT=0                                                          SOLV1310
      GO TO 150                                                         SOLV1320
  145 CONTINUE                                                          SOLV1330
      DO  1015 I = 1, NROW                                              SOLV1340
      RCG(I) = FORCMG(I) - APCG(I)                                      SOLV1350
 1015 SOLVEC(I) = RCG(I)                                                SOLV1360
150   ERRMAX=0.0                                                        SOLV1370
      PAP =0.0                                                          SOLV1380
      RCG2=0.0                                                          SOLV1390
      NULL=1                                                            SOLV1400
      ITRNO=ITRNO+1                                                     SOLV1410
      DO 270 I=1,NROW                                                   SOLV1420
      IF (I.NE.NONROW(NULL)) GO TO 160                                  SOLV1430
      NULL=NULL+1                                                       SOLV1440
      GO TO 270                                                         SOLV1450
160   SAVE=0.0                                                          SOLV1460
      LOCSTF=IROWL(I)+1                                                 SOLV1470
      IF (ISTSTF(LOCSTF).LT.0) GO TO 190                                SOLV1480
C                                                                       SOLV1490
C         ROW IN BLOCK FORMAT, ONE COLUMN INDICATOR FOR 6 NON-ZERO      SOLV1500
C                                                               ELEMENTS SOLV1510
170   ICOL=ISTSTF(LOCSTF)-1                                             SOLV1520
      DO 180 J=1,6                                                      SOLV1530
      ICOL=ICOL+1                                                       SOLV1540
      LOCSTF=LOCSTF+1                                                   SOLV1550
180   SAVE=SAVE+STRSTF(LOCSTF)*SOLVEC(ICOL)                             SOLV1560
      LOCSTF=LOCSTF+1                                                   SOLV1570
      IF (ISTSTF(LOCSTF).LT.0) GO TO 210                                SOLV1580
      GO TO 170                                                         SOLV1590
C                                                                       SOLV1600
C         ROW IN ELEMENT FORMAT, ONE COLUMN INDICATOR FOR EACH NON-ZERO SOLV1610
C                                                               ELEMENT  SOLV1620
190   NONELE=-ISTSTF(LOCSTF)                                            SOLV1630
      DO 200 J=1,NONELE                                                 SOLV1640
      LOCSTF=LOCSTF+1                                                   SOLV1650
      ICOL=ISTSTF(LOCSTF)                                               SOLV1660
      LOCSTF=LOCSTF+1                                                   SOLV1670
```

```
200   SAVE=SAVE+STRSTF(LOCSTF)*SOLVEC(ICOL)                             SOLV1680
  210 CONTINUE                                                          SOLV1690
      IF ( INDRLX .NE. 2 ) GO TO 1100                                   SOLV1700
      APCG(I) = SAVE + STFDIA(I)*SOLVEC(I)                              SOLV1710
      PAP     = SOLVEC(I)*APCG(I) + PAP                                 SOLV1720
      RCG2    = RCG2 + RCG(I)*SOLVEC(I)                                 SOLV1730
      GO TO 270                                                         SOLV1740
 1100 CONTINUE                                                          SOLV1750
C                                                                       SOLV1760
C         SUBTRACT SAVE FROM FORCMG(I) AND DIVIDE BY THE DIAGONAL       SOLV1770
C         ELEMENT (WHICH IS NOT CONSIDERED IN THE ABOVE CALCULATION)    SOLV1780
C              TO GET X(N+1)                                            SOLV1790
      IF (INDRLX.EQ.0) GO TO 220                                        SOLV1800
C                                                                       SOLV1810
C         USE OVERRELAXATION METHOD                                     SOLV1820
C                                                                       SOLV1830
      SAVE=RELAXF*(FORCMG(I)-SAVE)/STFDIA(I)+(1.-RELAXF)*SOLVEC(I)      SOLV1840
      ERR1=ABS(SOLVEC(I)-SAVE)                                          SOLV1850
      GO TO 250                                                         SOLV1860
220   SAVE=(FORCMG(I)-SAVE)/STFDIA(I)                                   SOLV1870
      ERR1=ABS(SOLVEC(I)-SAVE)                                          SOLV1880
      IF (ITRNO.LT.IUPDAT) GO TO 250                                    SOLV1890
C                                                                       SOLV1900
C         USE THE AITKEN DELTA SQUARED PROCESS TO IMPROVE THE           SOLV1910
C         CONVERGENCE PROCESS                                           SOLV1920
      IF (ERR1/(ABS(SAVE)+1.E-14).LT.ERRTL2) GO TO 230                  SOLV1930
      ERR2=ABS(SOLVC2(I)-SOLVEC(I))                                     SOLV1940
      IF (ERR1.GE.ERR2) GO TO 230                                       SOLV1950
      SAVE1=SAVE-(SAVE-SOLVEC(I))**2/(SAVE-2.*SOLVEC(I)+SOLVC2(I))      SOLV1960
      INDUPD=1                                                          SOLV1970
      ERR1=ABS(SOLVEC(I)-SAVE1)                                         SOLV1980
      SOLVC2(I)=SAVE1                                                   SOLV1990
      GO TO 240                                                         SOLV2000
230   SOLVC2(I)=SAVE                                                    SOLV2010
240   SOLVEC(I)=SAVE                                                    SOLV2020
      ERR=ERR1/(ABS(SOLVC2(I))+1.E-12)                                  SOLV2030
      GO TO 260                                                         SOLV2040
250   CONTINUE                                                          SOLV2050
      SOLVC2(I)=SOLVEC(I)                                               SOLV2060
      SOLVEC(I)=SAVE                                                    SOLV2070
      ERR=ERR1/(ABS(SAVE)+1.E-12)                                       SOLV2080
260   CONTINUE                                                          SOLV2090
C                                                                       SOLV2100
      IF (ERR.GT.ERRMAX) ERRMAX=ERR                                     SOLV2110
270   CONTINUE                                                          SOLV2120
      IF ( INDRLX .NE. 2 ) GO TO 1270                                   SOLV2130
      IF ( ITRNO .NE. 1 ) GO TO 1220                                    SOLV2140
      IF ( IND1SL.NE. 0 ) GO TO 145                                     SOLV2150
 1220 CONTINUE                                                          SOLV2160
      IF ( PAP .EQ. 0. ) PAP = 1.E-9                                    SOLV2170
      ALPA = RCG2 / PAP                                                 SOLV2180
      RCG2  = 0.0                                                       SOLV2190
      DO  1240 J = 1, NROW                                              SOLV2200
      SOLVC2(J) = SOLVC2(J) + ALPA*SOLVEC(J)                            SOLV2210
      RCG(J) =  RCG(J) - ALPA* APCG(J)                                  SOLV2220
      ERRMAX = AMAX1( ABS(RCG(J)), ERRMAX )                             SOLV2230
```

```
 1240 RCG2      = RCG2 + RCG(J)*APCG(J)                                 SOLV2240
      ERRMAX = ERRMAX/ FVMAX                                            SOLV2250
      BETA   = -RCG2 /  PAP                                             SOLV2260
      DO 1260  J = 1, NROW                                              SOLV2270
 1260 SOLVEC(J)  = RCG(J) + BETA* SOLVEC(J)                             SOLV2280
 1270 CONTINUE                                                          SOLV2290
      IF (INDUPD.EQ.0) GO TO 290                                        SOLV2300
      DO 280 K=1,NROW                                                   SOLV2310
280   SOLVEC(K)=SOLVC2(K)                                               SOLV2320
      IUPDAT=IUPDAT+2                                                   SOLV2330
      INDUPD=0                                                          SOLV2340
290   CONTINUE                                                          SOLV2350
C                                                                       SOLV2360
C          CHECK LOOP TERMINATION CONDITIONS                            SOLV2370
C                                                                       SOLV2380
      IF (INDITR.EQ.ITRNO) GO TO 320                                    SOLV2390
      IF (INDITR.LT.ITRNO) GO TO 310                                    SOLV2400
      IF (INDOPT.GT.0) GO TO 320                                        SOLV2410
      CALL SECOND (TIME)                                                SOLV2420
      IF (TIME.GT.TMAX-5.) GO TO 300                                    SOLV2430
      IF (ERRMAX.GT.ERRTOL) GO TO 150                                   SOLV2440
      GO TO 320                                                         SOLV2450
300   WRITE (6,440)                                                     SOLV2460
      INDPLS=0                                                          SOLV2470
      GO TO 320                                                         SOLV2480
310   CONTINUE                                                          SOLV2490
      WRITE (6,450) INDITR,ERRMAX                                       SOLV2500
320   CONTINUE                                                          SOLV2510
      IF ( INDRLX .NE. 2 ) GO TO 1310                                   SOLV2520
      DO 1300  J = 1, NROW                                              SOLV2530
 1300 SOLVEC(J) = SOLVC2(J)                                             SOLV2540
      INDOPT = 2                                                        SOLV2550
 1310 CONTINUE                                                          SOLV2560
      CALL PNTFMV (SOLVEC,NJOINT,1)                                     SOLV2570
      WRITE (6,460) ITRNO,ERRMAX                                        SOLV2580
      IUPDAT=IUPDAT+IUPDAT                                              SOLV2590
      INDOPT=INDOPT+1                                                   SOLV2600
      IF (INDOPT.LT.2) GO TO 150                                        SOLV2610
C                                                                       SOLV2620
C          FIND THE COMPLETE FORCE-MOMENT VECTOR                        SOLV2630
C                                                                       SOLV2640
      DO 370 K=1,NNONRW                                                 SOLV2650
      SAVE=0.0                                                          SOLV2660
      I=NONROW(K)                                                       SOLV2670
      IF (SOLVEC(I).NE.0.0) SAVE=STFDIA(I)*SOLVEC(I)                    SOLV2680
      LOCSTF=IROWL(I)+1                                                 SOLV2690
      IF ( ISTSTF(LOCSTF) .EQ. 0 ) GO TO 370                            SOLV2700
      IF (ISTSTF(LOCSTF).LT.0) GO TO 350                                SOLV2710
330   ICOL=ISTSTF(LOCSTF)-1                                             SOLV2720
      DO 340 J=1,6                                                      SOLV2730
      ICOL=ICOL+1                                                       SOLV2740
      LOCSTF=LOCSTF+1                                                   SOLV2750
340   SAVE=SAVE+STRSTF(LOCSTF)*SOLVEC(ICOL)                             SOLV2760
      LOCSTF=LOCSTF+1                                                   SOLV2770
      IF (ISTSTF(LOCSTF).LT.0) GO TO 370                                SOLV2780
      GO TO 330                                                         SOLV2790
```

```
350   NONELE=-ISTSTF(LOCSTF)                                            SOLV2800
      DO 360 J=1,NONELE                                                 SOLV2810
      LOCSTF=LOCSTF+1                                                   SOLV2820
      ICOL=ISTSTF(LOCSTF)                                               SOLV2830
      LOCSTF=LOCSTF+1                                                   SOLV2840
360   SAVE=SAVE+STRSTF(LOCSTF)*SOLVEC(ICOL)                             SOLV2850
370   FORCMG(I)=SAVE                                                    SOLV2860
      CALL PNTFMV (FORCMG,NJOINT,-1)                                    SOLV2870
C                                                                       SOLV2880
C          INDPLS = 0 IMPLIES NO PLASTICITY                             SOLV2890
C                                                                       SOLV2900
      IF (INDPLS.EQ.0) RETURN                                           SOLV2910
      NULL=0                                                            SOLV2920
      INDBDL=NNONRW                                                     SOLV2930
      LOCSTF=0                                                          SOLV2940
      DO 410 I=1,NJOINT                                                 SOLV2950
      DO 410 J=1,6                                                      SOLV2960
      NULL=NULL+1                                                       SOLV2970
      IF (IJRFM(I,J).EQ.0) GO TO 410                                    SOLV2980
      LOCSTF=LOCSTF+1                                                   SOLV2990
      IF (IJRFM(I,J).LT.0) GO TO 410                                    SOLV3000
      ICOL=IJRFM(I,J)                                                   SOLV3010
      IF (RLIMTU(ICOL).GE.FORCMG(NULL)) GO TO 380                       SOLV3020
      FORCMG(NULL)=RLIMTU(ICOL)                                         SOLV3030
      GO TO 390                                                         SOLV3040
380   IF (RLIMTL(ICOL).LE.FORCMG(NULL)) GO TO 410                       SOLV3050
      FORCMG(NULL)=RLIMTL(ICOL)                                         SOLV3060
390   NNONRW=NNONRW-1                                                   SOLV3070
      IJRFM(I,J)=0                                                      SOLV3080
      DO 400 K=LOCSTF,NNONRW                                            SOLV3090
400   NONROW(K)=NONROW(K+1)                                             SOLV3100
      LOCSTF=LOCSTF-1                                                   SOLV3110
      IF (STFDIA(NULL).EQ.0.0) CALL ERPNT1 (ERDIA,5,-1)                 SOLV3120
      IF (SOLVEC(NULL).NE.0.0) CALL ERPNT1 (ERRL,6,-1)                  SOLV3130
      WRITE (6,470) ALP(J),I                                            SOLV3140
410   CONTINUE                                                          SOLV3150
      CALL ERPNT2                                                       SOLV3160
      IF (INDBDL.EQ.NNONRW) RETURN                                      SOLV3170
      INDISL = 1                                                        SOLV3180
      GO TO 140                                                         SOLV3190
C                                                                       SOLV3200
420   FORMAT (1H1)                                                      SOLV3210
430   FORMAT (29H THE DIAGONAL ELEMENT FOR ROW,I4,16H CAN NOT BE ZERO)  SOLV3220
440   FORMAT (66H SOLUTION ITERATIONS STOPPED BECAUSE JOB IS APPROACHINGSOLV3230
     1 TIME LIMIT)                                                      SOLV3240
450   FORMAT (10HOMORE THANI5,28H ITERATIONS, MAXIMUM ERROR =E16.8)     SOLV3250
460   FORMAT (1XI4,68H ITERATIONS WERE REQUIRED TO REACH A MAXIMUM RELATSOLV3260
     1IVE DIFFERENCE OF E14.7)                                          SOLV3270
470   FORMAT (5H THE A10,37HRESTRAINT WAS VIOLATED FOR JOINT NO. I4)    SOLV3280
      END                                                               SOLV3290
```

```
      SUBROUTINE PNTFMV (ARRAY,NJOINT,IND)
      COMMON COM(30)
      EQUIVALENCE ( COM( 8 ), NROW   )
      DIMENSION ARRAY(1)
      LCOUNT=70
      I1=1
      I2=6
      DO 30 I=1,NJOINT
      IF (LCOUNT.LT.56) GO TO 20
C
C          WRITE PAGE TITLE AT TOP OF NEXT PAGE
C
      LCOUNT=7
      IF (IND.LT.0) GO TO 10
      WRITE (6,50)
      GO TO 20
10    CONTINUE
      WRITE (6,60)
20    CONTINUE
      WRITE (6,70) I,(ARRAY(K),K=I1,I2)
      I1=I1+6
      I2=I2+6
      LCOUNT=LCOUNT+1
30    CONTINUE
      IF (IND.GT.0) RETURN
      K=0
      XF=0.0
      YF=0.0
      ZF=0.0
      DO 40 I=1,NJOINT
      K=K+1
      XF=XF+ARRAY(K)
      YF=YF+ARRAY(K+1)
      ZF=ZF+ARRAY(K+2)
40    K=K+5
      WRITE (6,80) XF,YF,ZF
      SAV = 0
      DO  45   I = 1, NROW
   45 SAV = AMAX1(SAV, ABS(ARRAY(I) ) )
      XF = XF/SAV *100.
      YF = YF/SAV *100.
      ZF = ZF/SAV *100.
      WRITE(6,90) XF,YF,ZF
      RETURN
C
50    FORMAT (1H1,36X39HNODAL POINT DISPLACEMENTS AND ROTATIONS/44X24HGL
     1OBAL COORDINATE SYSTEM/1H0,4X11HNODAL POINT,5X3(2X12HDISPLACEMENT1
     2X),3(4X8HROTATION3X)/7X7HNUMBER,2(14X1HX,14X1HY,14X1HZ)///)
60    FORMAT (1H1,40X30HNODAL POINT FORCES AND MOMENTS/44X24HGLOBAL COOR
     1DINATE SYSTEM/1H0,4X11HNODAL POINT,5X3(5X5HFORCE5X),3(5X6HMOMENT4X
     2)/7X7HNUMBER 2(14X1HX,14X1HY,14X1HZ)///)
70    FORMAT (8XI3,9X6(E15.6))
80    FORMAT (1H0,6X5HTOTAL,8X3F15.6)
   90 FORMAT  ( 1H0 5X35HRELATIVE SOLUTION ERRORS IN PERCENT  /
     1           20X   3F15.5       )
```

```
      END                                                               PNTF 560
```

```
      SUBROUTINE FMBARS
      COMMON COM(30)
      EQUIVALENCE ( COM( 4), NBAR    )
      EQUIVALENCE ( COM ( 29 ), NSHPAN  )
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3    KS     (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4    BARJ   (  130), BARYM (  130), BARSM (  130), RKN    (  130),
     5    RKT    (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)
      DIMENSION  RBARR(1)
      EQUIVALENCE ( IBARR(1), RBARR(1) )
      DIMENSION STIFML(12,12), TRANSM(3,3)
      EQUIVALENCE ( RJXYZ(1,1), STIFML(1,1)  )
      EQUIVALENCE ( RJXYZ(50,2),TRANSM(1,1) )
      DIMENSION IPR(6), DISPL(12), DISPL2(12), FMLSTF(12)
      DATA IPR / 1, 2, 3, 7, 8, 9 /
      LCOUNT = 70
      NPANEL = 0
      LCUNT2 = 70
      I1OR2  = 0
C         CALCULATE THE NO. OF NON SHEAR PANEL BARS
      NBARI  = NBAR - 2*NSHPAN
      REWIND 2
      DO 50 I=1,NBAR
      IF ( I .LE. NBARI )  GO TO 8
      IF ( LCUNT2 .LT. 56 ) GO TO 6
      WRITE(6,80)
      LCUNT2= 7
    6 CONTINUE
      BARLGT = ABS( RBARR(I) )
      I1OR2  = I1OR2 + 1
      GO TO 12
    8 CONTINUE
      IF (LCOUNT.LT.56) GO TO 10
C
C         WRITE PAGE TITLE AT TOP OF NEXT PAGE
C
      LCOUNT=7
      WRITE (6,60)
10    CONTINUE
      IR=IBARR(I)
   12 CONTINUE
      IP=IBARP(I)
      IQ=IBARQ(I)
      READ (2) ((TRANSM(K,J),J=1,3),K=1,3)
      READ (2) ((STIFML(K,J),J=1,12),K=1,12)
C
C         SET UP DISPLACEMENT VECTOR
C
      JJ=(IQ-1)*6
      II=(IP-1)*6
      DO 20 K=1,6
      II=II+1
```

```
      I1=IPR(K)
      DISPL(I1)=SOLVEC(II)
      JJ=JJ+1
      I1=I1+3
20    DISPL(I1)=SOLVEC(JJ)
C
C          TRANSFORM DISPLACEMENT VECTOR
C
      II=-3
      IROW=0
      DO 30 K=1,4
      II=II+3
      DO 30 J=1,3
      IROW=IROW+1
      DISPL2(IROW)=0.0
      DO 30 JJ=1,3
      I1=II+JJ
30    DISPL2(IROW)=DISPL2(IROW)+TRANSM(J,JJ)*DISPL(I1)
C
C          CALCULATE LOCAL FORCE-MOMENT VECTOR
C
      DO 40 K=1,12
      FMLSTF(K)=.0
      DO 40 J=1,12
40    FMLSTF(K)=FMLSTF(K)+STIFML(K,J)*DISPL2(J)
       IF ( I .LE. NBARI ) GO TO 48
      SHFLOW =  2.*FMLSTF(1) / BARLGT
      IF ( I1OR2 .NE. 1 ) GO TO 42
      IPS   = IP
      IQS   = IQ
      SHFSV = SHFLOW
      GO TO 50
   42 CONTINUE
      NPANEL = NPANEL + 1
      SHFSV  = (ABS(SHFLOW) + ABS(SHFSV ) ) * .5
      SHFSV = SIGN( SHFSV, SHFLOW )
      WRITE(6,90) NPANEL, IPS, IQ, IQS, IP, SHFSV
      I1OR2 = 0
      LCUNT2 = LCUNT2 + 2
      GO TO 50
   48 CONTINUE
      WRITE (6,70) I,IP,IQ,IR,(FMLSTF(K),K=1,3),(FMLSTF(K),K=7,9),(FMLST
     1F(K),K=4,6),(FMLSTF(K),K=10,12)
      LCOUNT=LCOUNT+3
50    CONTINUE
      RETURN
C
60    FORMAT (1H1,49X22HBAR FORCES AND MOMENTS/49X24HLOCAL COORDINATE SY
     1STEMS/1H0,29H  BAR      NODAL POINT NUMBERS,11X3(5X5HFORCE5X),3(5X6
     2HMOMENT4X)/30H  NUMBER      P      Q       R   ,4X2(14X1HX,14X1HY,14X1
     3HZ)///)
70    FORMAT (3XI3,6XI3,4XI3,4XI3,11H   POINT P ,6E15.6/29X,11H   POINT
     1Q ,6E15.6/)
   80 FORMAT(1H151X17HSHEAR PANEL FLOWS /
     1        49X24HLOCAL COORDINATE SYSTEMS /
     246H0 PANEL    NODAL POINT NUMBERS      EQUIVALENT   /
```

```
  346H  NUMBER    A    B    C    D        SHEAR FLOW,   /// )         FMBA1120
90 FORMAT ( 1H0, I5, I8, 3I5, 7X E10.3 ,  7XE10.3 )                   FMBA1130
   END                                                                FMBA1140
```

```
      OVERLAY ( SASLP, 4, 0 )                                          OV40  10

      PROGRAM NLMDAL                                                   NLMD  10
      DIMENSION SM(444)                                                NLMD  20
      DIMENSION AMASSV(102)                                            NLMD  30
      DIMENSION RJX(1), RJY(1), RJZ(1)                                 NLMD  40
      DIMENSION AMASS(102), IRWKP(103)                                 NLMD  50
      DIMENSION EGVALU(444), EGVECT(2220), WORK1(444),                 NLMD  60
     1             WORK2(444), ADIA(444), BOFDIA(444)                  NLMD  70
      DIMENSION ROWNXT(444), IRWTOR(444), T(444)                       NLMD  80
      COMMON COM(30)                                                   NLMD  90
      EQUIVALENCE ( COM( 8), NROW    )                                 NLMD 100
      EQUIVALENCE ( COM(21), NRWTOR )                                  NLMD 110
      EQUIVALENCE ( COM(22), IREDTO , MORDER )                         NLMD 120
      EQUIVALENCE ( COM(23), NEIGVL )                                  NLMD 130
      COMMON / CMAIN /                                                 NLMD 140
     1     RJXYZ (100,3), IJREM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3), NLMD 150
     2     RMTVCT( 74,3), IBARP (   130), IBARQ (   130), IBARR (   130), NLMD 160
     3     KS      (   130), BAREA (   130), BARIN (   130), BARIT (   130), NLMD 170
     4     BARJ   (   130), BARYM (   130), BARSM (   130), RKN     (   130), NLMD 180
     5     RKT    (   130), SOLVEC(   444), RLIMTU(    88), RLIMTL(    88), NLMD 190
     6     DPRTIT(    88), IROWL (   445), STRSTF(1)                  NLMD 200
      EQUIVALENCE ( RJXYZ(1,1),RJX (1)),(RJXYZ(1,2),RJY(1))            NLMD 210
      EQUIVALENCE ( RJXYZ(1,3),RJZ(1)),(FRCVCT(1,1),SM(1), ADIA(1) )   NLMD 220
      EQUIVALENCE ( BAREA(1   ),ROWNXT(1)), ( SOLVEC(1), IRWTOR(1)   ) NLMD 230
      EQUIVALENCE (RLIMTU(1   ),AMASS(1) ), (RLIMTL(15), IRWKP(1)    ) NLMD 240
      EQUIVALENCE ( STRSTF(1), ISTSTF )                                NLMD 250
      EQUIVALENCE ( STRSTF( 5400), EGVALU )                            NLMD 260
      EQUIVALENCE ( STRSTF( 5500), EGVECT )                            NLMD 270
      EQUIVALENCE ( STRSTF(7964), WORK1 )                              NLMD 280
      EQUIVALENCE ( STRSTF(8408), WORK2   )                            NLMD 290
      EQUIVALENCE ( STRSTF(9740), T        )                           NLMD 300
C           INITIALIZE THE PRINT ROUTINE                               NLMD 310
      CALL PRNTO                                                       NLMD 320
      DO 10 I=1,MORDER                                                 NLMD 330
   10 AMASSV(I)=AMASS(I)                                               NLMD 340
      ORDER=MORDER                                                     NLMD 342
      SUM=0.0                                                          NLMD 344
      DO 11 I=1,MORDER                                                 NLMD 346
   11 SUM=SUM+AMASS(I)                                                 NLMD 348
      SMMAX=SUM/ORDER                                                  NLMD 350
      DO 12 I=1,MORDER                                                 NLMD 352
   12 AMASS(I)=AMASS(I)/SMMAX                                          NLMD 354
C           REDUCE STIFFNESS MATRIX                                    NLMD 356
      CALL REDUCE                                                      NLMD 358
      KOUNT=MORDER*MORDER                                              NLMD 360
      SUM=0.0                                                          NLMD 362
      DO 13 I=1,MORDER                                                 NLMD 364
      IDIAG=(I-1)*MORDER+I                                             NLMD 366
   13 SUM=SUM+STRSTF(IDIAG)                                            NLMD 368
      SKMAX=SUM/ORDER                                                  NLMD 370
      DO 14 I=1,KOUNT                                                  NLMD 375
   14 STRSTF(I)=STRSTF(I)/SKMAX                                        NLMD 380
C                                   T                                  NLMD 390
```

```
C           MATRIX = ((M)**-.5) *(K)*((M)**-.5)
C
      CALL STFMAS (STRSTF,AMASS,MORDER)
C
C           HOUSEHOLDER METHOD TO OBTAIN MATRIX IN TRIPLE-DIAGONAL FORM
C
      CALL TRIDIA (MORDER,STRSTF,ADIA,BOFDIA)
C
C           FIND THE NEIGVL + 6 SMALLEST EIGENVALUES
C
      CALL EIGVAL (MORDER,EGVALU,ADIA,BOFDIA,WORK1,WORK2,NEIGVL)
C
C           FIND THE EIGENVECTORS FOR THE NEIGVL EIGENVALUES ABOVE
C                 FOR MATRIX
      RATIO=SKMAX/SMMAX
      IGV1 = 2
      IGV2 = 6
      REWIND 9
   15 CONTINUE
      IGV1 = IGV1 + 5
      IGV2 = IGV2 + 5
      J=1
      DO 20 I=IGV1, IGV2
      CALL EIGVEC (MORDER,EGVALU(I),STRSTF,ADIA,BOFDIA,EGVECT(J),WORK1,W
     1ORK2)
21    EGVALU(I)=RATIO*EGVALU(I)
20    J=J+NROW
C
C           CALCULATE EIGENVECTOR FOR REDUCED MATRIX FORM
C
      K=0
      DO 50 M=1,5
      SAVE= 0.0
      DO 30 J=1,MORDER
      I=K+J
      EGVECT(I)=AMASS(J)*EGVECT(I)
      IF (ABS(SAVE).LT.ABS(EGVECT(I))) SAVE=EGVECT(I)
30    CONTINUE
      DO 40 J=1,MORDER
      I=K+J
40    EGVECT(I)=EGVECT(I)/SAVE
50    K=K+NROW
      CALL PRNT1 (EGVALU,EGVECT,AMASSV,IRWKP,NROW,MORDER,IRWTOR,WORK1)
C
C           TAKE ABOVE EIGENVECTORS AND USE THE TRANSFORMATION MATRICES
C                 TO GET FINAL EIGENVECTORS
      IF (IREDTO.EQ.NROW) GO TO 60
      CALL FNALEV (EGVECT,AMASS,MORDER,NEIGVL,NROW,NRWTOR,T,WORK1,IRWKP,
     1IRWTOR)
60    CONTINUE
C
C           PRINT
C
      CALL PRNT2 (EGVALU,EGVECT,AMASSV,IRWKP,NROW,MORDER,IRWTOR,WORK1)
      IF (   IGV2 .LT. NEIGVL ) GO TO 15
      RETURN
      END
```

```
         IDENT PUNPACK
        ENTRY PACK,UNPACK
        VFD   42/4CPACK,18/3
PACK    DATA  0
        SA1   B2          ***
        SB2   X1          *     B2 = NL
        SB4   B0                B4 = 0          INITIALLY
        SB5   9                 B5 = 9          INITIALLY
        SB6   60                B6 = 60         INITIALLY
        SB7   1                 B7 = 1          INITIALLY
        SA5   BIT0ONE           X5 = -0
WORDITS IX7   X7-X7             X7 = 0
        SA2   BIT1ONE           X2 = 020000000000000000000
        SA4   B1+B4             X4 = INPUTA( 1 + B4 )
        ZR    X4,FSTBIT         IF ( X4 .EQ. 0.0 ) GO TO FSTBIT
        BX7   X5                X7 = X5
        BX6   X4                X6 = X4
        SA6   B3+B5             OUTPUTA( 1 + BS ) = X6
        SB5   B5+1              B5 = B5 + 1
        ZR    FSTBIT            GO TO FSTBIT
LOOP    SA4   B1+B4             X4 = INPUTA( B1 + B4 )
        ZR    X4,EQZERO         IF ( X4 .EQ. 0.0 ) GO TO EQZERO
        BX7   X7+X2             X7 = LOGICAL SUM X7 AND X2
        BX6   X4                X6 = X4
        SA6   B3+B5             OUTPUTA(  1 + B5 ) = X6
        SB5   B5+1              B5 = B5+1
EQZERO  AX2   1                 CIRCULAR SHIFT X2 1 BIT RIGHT
FSTBIT  SB4   B4+1              B4 = B4 + 1
        EQ    B4,B2,ENDCHK      IF ( B4 .EQ. B2 ) GO TO ENDCHK
        NE    B4,B6,LOOP        IF ( B4 .NE. B6 ) GO TO LOOP
        SA7   B3+B7             OUTPUTA(  1 + B7 ) = X7
        SB7   B7+1              B7 = B7 + 1
        SB6   B6+60             B6 = B6 + 60
        ZR    WORDITS           GO TO WORDI
ENDCHK  SA7   B3+B7             OUTPUTA(  1 + B7 ) = X7
        SX6   B5                X6 = B5
        SA6   B3                OUTPUTA(  1  ) = X6
        ZR    PACK              RETURN
        VFD   42/6CUNPACK,18/3
UNPACK  DATA  0
        SA2   B2          ***   B2 = NF
        SB2   X2          *
        SB4   9                 B4 = 9          INITIALLY
        SB5   0                 B5 = 0          INITIALLY
        SA1   B1          ***
        SB6   X1          *     B6 = LENGTH OF INPUTA ARRAY
        SB7   0                 B7 = 0          INITIALLY
        SX5   1                 X5 = 1          INITIALLY
        IX6   X1-X1             X6 = 0          INITIALLY
WORDJTS EQ    B4,B6,LOOPJ2      IF ( B4 .EQ. B6 ) GO TO LOOPJ2
        SA4   X5+B1             X4 = INPUTA( 1 + X5 )
        SX5   X5+1              X5 = X5 + 1
        SB7   B7+60             B7 = B7 + 60
LOOPJ1  PL    X4,X4PLUS         IF ( X4 .POSITIVE. ) GO TO X4PLUS
        SA3   B1+B4             X3 = INPUTA( 1 + B4 )
```

```
         BX7    X3                X7 = X3                                     PACK 560
         SA7    B3+B5             OUTPUTA( 1 + B5 ) = X7                      PACK 570
         SB4    B4+1              B4 = B4+1                                   PACK 580
         ZR     COMCOD1           GO TO COMCOD1                               PACK 590
X4PLUS   SA6    B3+B5             OUTPUTA( 1 + B5 ) = X6 = 0.0                PACK 600
COMCOD1  LX4    1                 CIRCULAR SHIFT X4 1 BIT LEFT                PACK 610
         SB5    B5+1              B5 = B5+1                                   PACK 620
         EQ     B5,B2,UNPACK      IF ( B5 .EQ. 32 ) RETURN                    PACK 630
         EQ     B5,B7,WORDJTS     IF ( B5 .EQ. B7 ) GO TO WORDJTS             PACK 640
         NE     B4,B6,LOOPJ1      IF ( B4 .NE. B6 ) GO TO LOOPJ1              PACK 650
LOOPJ2   SA6    B3+B5             OUTPUTA( 1 + B5 ) = X6 = 0.0                PACK 660
         SB5    B5+1              B5 = B5 + 1                                 PACK 670
         NE     B5,B2,LOOPJ2      IF ( B5 .NE. 32 ) GO TO LOOPJ2              PACK 680
         ZR     UNPACK            RETURN                                      PACK 690
BITOONE  DATA   04000000000000000000                                          PACK 700
BITIONE  DATA   02000000000000000000                                          PACK 710
         END                                                                  PACK 720
```

```
      SUBROUTINE REDUCE
      DIMENSION ISTSTF(1)
      DIMENSION IOBUF(444)
      COMMON / CMAIN /
     1    RJXYZ (100,3), IJRFM ( 74,6), IJALFM( 74,2), FRCVCT( 74,3),
     2    RMTVCT( 74,3), IBARP (  130), IBARQ (  130), IBARR (  130),
     3    KS    (  130), BAREA (  130), BARIN (  130), BARIT (  130),
     4    BARJ  (  130), BARYM (  130), BARSM (  130), RKN   (  130),
     5    RKT   (  130), SOLVEC(  444), RLIMTU(   88), RLIMTL(   88),
     6    DPRTIT(   88), IROWL (  445), STRSTF(1)
      DIMENSION IRWKP(103), SM(444), ROWNXT(444)
      DIMENSION T(444), IRCPST(444), IRWTOR(444)
      EQUIVALENCE ( STRSTF(1), ISTSTF(1))
      EQUIVALENCE ( FRCVCT(1,1), SM(1)), ( BAREA(1), ROWNXT(1))
      EQUIVALENCE ( SOLVEC(1)   , IRWTOR(1)), ( RLIMTL(15), IRWKP(1))
      EQUIVALENCE ( IROWL(1), IOBUF(1))
      EQUIVALENCE ( STRSTF(7964 ), IRCPST )
      EQUIVALENCE ( STRSTF(9740 ), T      )
      INTEGER TAPEA, TAPEB
      COMMON COM(30)
      EQUIVALENCE ( COM (  8), NROW   )
      EQUIVALENCE ( COM ( 22), IREDTO)
      EQUIVALENCE ( COM ( 21), NRWTOR)
      EQUIVALENCE ( COM ( 28), INDWTS )
C
      DIMENSION ERSD(13)
      DIMENSION ERZD(12)
      DATA ERSD   /50H TOO MANY ROWS HAVE BEEN REDUCED FROM THE STIFFNES
     1,60HS MATRIX LEADING TO A VERY SMALL DIAGONAL ELEMENT IN A ROW B,
     220HEING REMOVED         /
      DATA ERZD   /50H TOO MANY ROWS HAVE BEEN REDUCED FROM THE STIFFNES
     1,60HS MATRIX LEADING TO A ZERO DIAGONAL ELEMENT IN A ROW BEING R,
     210HEMOVED     /
      TAPEA = 1
      TAPEB = 2
C
C         SET UP IRWTOR WITH THE ROWS TO BE REDUCED IN DESCENDING ORDER
C
      IRWKP(IREDTO+1)=0
      J=1
      IRWTOR(1)=0
      IF (IREDTO.EQ.NROW) GO TO 30
      NRWTOR=NROW-IREDTO
      LAST=NRWTOR+1
      DO 20 I=1,NROW
      IF (I.EQ.IRWKP(J)) GO TO 10
      LAST=LAST-1
      IRWTOR(LAST)=I
      GO TO 20
   10 J=J+1
   20 CONTINUE
   30 CONTINUE
C
C         WRITE STIFFNESS MATRIX ON TAPEA BY ROWS, WITH THE EXCEPTION
C               THAT THE FIRST ROW TO BE REMOVED IS STORED IN ROWNXT
```

```
C                                                                       REDU 510
      REWIND TAPEA                                                      REDU 520
      L=1                                                               REDU 530
      DO 160 I=1,NROW                                                   REDU 540
      IF=1                                                              REDU 550
40    L=L+1                                                             REDU 560
      IF (ISTSTF(L).LT.0) GO TO 110                                     REDU 570
      IC=ISTSTF(L)                                                      REDU 580
      IF (IC.GT.IF) GO TO 70                                            REDU 590
50    IL=IC+5                                                           REDU 600
      DO 60 J=IF,IL                                                     REDU 610
      L=L+1                                                             REDU 620
      IF (ABS(STRSTF(L)).LT.1.E-8) STRSTF(L)=0.0                        REDU 630
      IF (STRSTF(L).EQ.0.0) GO TO 60                                    REDU 640
      IF ( INDWTS .NE.  0     )  CALL WRSTRK ( I, J, STRSTF(L) )        REDU 650
60    SM(J)=STRSTF(L)                                                   REDU 660
      IF=IL+1                                                           REDU 670
      GO TO 40                                                          REDU 680
70    IL=IC-1                                                           REDU 690
      IF (IF.EQ.0.OR.IL.EQ.0) GO TO 80                                  REDU 700
      IF (IABS(IF).LT.444.AND.IABS(IL).LT.444) GO TO 90                 REDU 710
80    CONTINUE                                                          REDU 720
      WRITE (6,340) IF,IL                                               REDU 730
      IF=L-200                                                          REDU 740
      IL=L+200                                                          REDU 750
      WRITE (6,350) (STRSTF(J),J=IF,IL)                                 REDU 760
      WRITE (6,360) (ISTSTF(J),J=IF,IL)                                 REDU 770
      STOP                                                              REDU 780
90    CONTINUE                                                          REDU 790
      DO 100 J=IF,IL                                                    REDU 800
100   SM(J)=0.0                                                         REDU 810
      IF=IC                                                             REDU 820
      GO TO 50                                                          REDU 830
110   CONTINUE                                                          REDU 840
      IF (IF.GT.NROW) GO TO 130                                         REDU 850
      DO 120 J=IF,NROW                                                  REDU 860
120   SM(J)=0.0                                                         REDU 870
130   CONTINUE                                                          REDU 880
      IF (I.NE.IRWTOR(1)) GO TO 150                                     REDU 890
      DO 140 J=1,NROW                                                   REDU 900
140   ROWNXT(J)=SM(J)                                                   REDU 910
      GO TO 160                                                         REDU 920
C                             PACK AND WRITE THE ROW IN BLOCK FORMAT    REDU 930
  150 CALL PACK ( SM(1), NROW, IOBUF(    1   ) )                        REDU 940
      KWRIT = IOBUF(1)                                                  REDU 950
      WRITE( TAPEA ) KWRIT, ( IOBUF(J),J=1,KWRIT )                      REDU 960
160   CONTINUE                                                          REDU 970
C             PRINT LAST LINE                                           REDU 980
      IF ( INDWTS .NE. 0 ) CALL WRSTRK(-1,-1,1.)                        REDU 990
      DO 170 J=1,NROW                                                   REDU1000
170   IRCPST(J)=J                                                       REDU1010
C                                                                       REDU1020
C             SET NUMBER OF ACTIVE ROWS                                 REDU1030
C                                                                       REDU1040
      NACTRW=NROW                                                       REDU1050
      KREAD = 0                                                         REDU1060
```

```
      KREAD1 = 1                                                        REDU1070
      IF (IREDTO.EQ.NROW) GO TO 320                                     REDU1080
C                                                                       REDU1090
C         START OF LOOP TO REDUCE STRUCTURAL STIFFNESS MATRIX           REDU1100
C                                                                       REDU1110
      ICOUNT=0                                                          REDU1112
      SAVTOT=0.                                                         REDU1114
      KWRIT1= 4402                                                      REDU1120
      KWRIT = 2502                                                      REDU1130
      REWIND 9                                                          REDU1140
      IRWTOR(NRWTOR+1)=0                                                REDU1150
      DO 310 IRWR=1,NRWTOR                                              REDU1160
      REWIND TAPEA                                                      REDU1170
      REWIND TAPEB                                                      REDU1180
      IROW=IRWTOR(IRWR)                                                 REDU1190
      IR1=IRCPST(IROW)                                                  REDU1200
      IF (IRWR.NE.NRWTOR) GO TO 180                                     REDU1210
C                    REDUCE KWRIT1 TO FIT LATER BUFFER LIMITS           REDU1220
      KWRIT1= 2830                                                      REDU1230
      IR2=0                                                             REDU1240
C              IR1  CURRENT POSITION (ROW + COL. NO.) OF ROW BEING      REDU1250
C                                                                REMOVEDREDU1260
C              IR2  CURRENT POSITION OF NEXT ROW TO BE REMOVED          REDU1270
C                                                                       REDU1280
      GO TO 190                                                         REDU1290
180   IR2=IRWTOR(IRWR+1)                                                REDU1300
      IR2=IRCPST(IR2)                                                   REDU1310
190   CONTINUE                                                          REDU1320
      IRR=IR1-1                                                         REDU1330
C                                                                       REDU1340
C         BUILD TRANSFORMATION MATRIX OF ORDER (NACTRW)*(NACTRW-1)      REDU1350
C              SAVE ONLY LAST ROW, SINCE FIRST NACTRW-1 ROWS ARE THE    REDU1360
C              IDENTITY MATRIX                                          REDU1370
      NACTR2=NACTRW-1                                                   REDU1380
      SAVE=ROWNXT(IR1)                                                  REDU1390
      ICOUNT=ICOUNT+1                                                   REDU1392
      SAVTOT = SAVTOT + ABS(SAVE)                                       REDU1394
      AVSAVE=SAVTOT/ICOUNT                                              REDU1396
      IF(SAVE.EQ.0.) CALL ERPNT1(ERZD,12,1)                             REDU1400
      IF(ABS(SAVE)/AVSAVE.LT.1.E-06) CALL ERPNT1(ERSD,13,0)             REDU1405
      DO 200 I=1,IRR                                                    REDU1410
200   T(I)=-ROWNXT(I)/SAVE                                              REDU1420
      IF ( IR1 .GT. NACTR2 ) GO TO 212                                  REDU1430
      DO 210 I=IR1,NACTR2                                               REDU1440
210   T(I)=-ROWNXT(I+1)/SAVE                                            REDU1450
  212 CONTINUE                                                          REDU1460
C         **********                                                    REDU1470
C                                                                       REDU1480
C         SAVE LAST ROW OF TRANSFORMATION MATRIX ON FILE UNIT 9         REDU1490
C                                                                       REDU1500
      WRITE(9) ( T(K), K=1,NACTR2)                                      REDU1510
C                                                                       REDU1520
C                                                                       REDU1530
C         START LOOP TO PROCESS ACTIVE ROW IN COORDINATE REDUCTION      REDU1540
C                                                                       REDU1550
      DO 290 ICRNTR=1,NACTR2                                            REDU1560
```

```
C         READ STIFFNESS MATRIX ROW, ICRNTR                             REDU1570
C              BREAK DOWN THE INPUT BLOCKS                              REDU1580
      IF ( KREAD .GT. KREAD1 ) GO TO 215                                REDU1590
      KREAD1 = 1                                                        REDU1600
      READ ( TAPEA ) KREAD,( STRSTF(I),I=1,KREAD )                      REDU1610
  215 CALL UNPACK ( STRSTF( KREAD1 ), NACTRW, SM(1) )                   REDU1620
      KREAD1 = KREAD1 + ISTSTF(KREAD1)                                  REDU1630
C                                                                       REDU1640
C                                                                       REDU1650
      SAVE=SM(IR1)                                                      REDU1660
      IF (SAVE.NE.0.) GO TO 230                                         REDU1670
      IF ( IR1 .GT. NACTR2 ) GO TO 260                                  REDU1680
      DO 220 I=IR1,NACTR2                                               REDU1690
220   SM(I)=SM(I+1)                                                     REDU1700
      GO TO 260                                                         REDU1710
230   DO 240 I=1,IRR                                                    REDU1720
240   SM(I)=SM(I)+SAVE*T(I)                                             REDU1730
      IF ( IR1 .GT. NACTR2 ) GO TO 260                                  REDU1740
      DO 250 I=IR1,NACTR2                                               REDU1750
250   SM(I)=SM(I+1)+SAVE*T(I)                                           REDU1760
260   CONTINUE                                                          REDU1770
C                                                                       REDU1780
C         IF THIS ROW IS THE NEXT TO BE REMOVED, SAVE THE ROW IN ROWNXT REDU1790
C                                                                       REDU1800
      IF (ICRNTR.NE.IR2) GO TO 280                                      REDU1810
      DO 270 I=1,NACTR2                                                 REDU1820
270   ROWNXT(I)=SM(I)                                                   REDU1830
      IF ( ICRNTR .NE. NACTR2 ) GO TO 290                               REDU1840
      IF ( KWRIT .NE. 2502 ) GO TO 285                                  REDU1850
      GO TO 290                                                         REDU1860
C                                                                       REDU1870
C         WRITE THE  REDUCED ROW ON TO TAPEB                            REDU1880
C                                                                       REDU1890
C              BLOCK OUTPUT                                             REDU1900
  280 CALL PACK ( SM(1), NACTR2, STRSTF(KWRIT) )                        REDU1910
      KWRIT = KWRIT + ISTSTF(KWRIT)                                     REDU1920
      IF ( ICRNTR .EQ. NACTR2 ) GO TO 285                               REDU1930
      IF ( KWRIT .LT.KWRIT1) GO TO 287                                  REDU1940
  285 ISTSTF(2501) = KWRIT - 2502                                       REDU1950
      KWRIT = KWRIT - 1                                                 REDU1960
      WRITE( TAPEB )( STRSTF(I), I= 2501, KWRIT )                       REDU1970
      KWRIT = 2502                                                      REDU1980
  287 CONTINUE                                                          REDU1990
C                                                                       REDU2000
290   CONTINUE                                                          REDU2010
C                                                                       REDU2020
C         REVERSE READ/WRITE TAPE VALUES                                REDU2030
C                                                                       REDU2040
      I=TAPEA                                                           REDU2050
      TAPEA=TAPEB                                                       REDU2060
      TAPEB=I                                                           REDU2070
C         **********                                                    REDU2080
C                                                                       REDU2090
C         UPDATE IRCPST ARRAY TO ADJUST FOR THE ROW + COLUMN REMOVED    REDU2100
C                                                                       REDU2110
      IRCPST(IROW)=-IRCPST(IROW)                                        REDU2120
```

```
      K=IROW+1                                                          REDU2130
      DO 300 I=K,NROW                                                   REDU2140
      IF (IRCPST(I).LT.0) GO TO 300                                     REDU2150
      IRCPST(I)=IRCPST(I)-1                                             REDU2160
300   CONTINUE                                                          REDU2170
C                                                                       REDU2180
C         UPDATE NO. OF ACTIVE ROWS                                     REDU2190
C                                                                       REDU2200
      NACTRW=NACTR2                                                     REDU2210
310   CONTINUE                                                          REDU2220
C                                                                       REDU2230
C         COPY STIFFNESS MATRIX TO CORE                                 REDU2240
C                                                                       REDU2250
320   CONTINUE                                                          REDU2260
      IRR=0                                                             REDU2270
      REWIND TAPEA                                                      REDU2280
      DO 330 I=1,NACTRW                                                 REDU2290
C                                                                       REDU2300
C                        BREAK DOWN THE INPUT BLOCKS                    REDU2310
C                                                                       REDU2320
      IF ( KREAD .GT. KREAD1 ) GO TO 325                                REDU2330
      KREAD1 = 1                                                        REDU2340
      READ ( TAPEA ) KREAD, ( IOBUF(I), I=1, KREAD )                    REDU2350
  325 CALL UNPACK ( IOBUF( KREAD1 ), NACTRW, SM(1) )                    REDU2360
      KREAD1 = KREAD1 + IOBUF( KREAD1 )                                 REDU2370
      DO 330 J=1,NACTRW                                                 REDU2380
      IRR=IRR+1                                                         REDU2390
330   STRSTF(IRR)=SM(J)                                                 REDU2400
C                                                                       REDU2410
C                REVERSE  THE RECORD POSITION IN THE TRANSFORMATION     REDU2420
C                         MATRIX FILE AND PLACE ON UNIT 1               REDU2430
C                                                                       REDU2440
      IF(IREDTO.EQ.NROW) RETURN                                         REDU2445
      REWIND 1                                                          REDU2450
      NACTRW = IREDTO                                                   REDU2460
      DO  335  I = 1, NRWTOR                                            REDU2470
      BACKSPACE 9                                                       REDU2480
      READ(9)(SM(L),L=1, NACTRW)                                        REDU2490
      WRITE(1)(SM(L),L=1, NACTRW)                                       REDU2500
      BACKSPACE 9                                                       REDU2510
      NACTRW = NACTRW + 1                                               REDU2520
  335 CONTINUE                                                          REDU2530
C                                                                       REDU2540
      RETURN                                                            REDU2550
C                                                                       REDU2560
340   FORMAT (7H IF IL ,2I10)                                           REDU2570
350   FORMAT (10E12.5)                                                  REDU2580
360   FORMAT (10I12)                                                    REDU2590
      END                                                               REDU2600
```

```
      SUBROUTINE STFMAS (STRSTF,AMASS,MORDER)                           STFM  10
      DIMENSION STRSTF(MORDER,MORDER), AMASS(1)                         STFM  20
C                                                                       STFM  30
C         COMPUTE  (M)**-1/2  = ((M)**-1/2)T  FOR DIAGONAL MATRIX       STFM  40
C                                                                       STFM  50
      DO 10 I=1,MORDER                                                  STFM  60
10    AMASS(I)=1/SQRT(AMASS(I))                                         STFM  70
C                                                                       STFM  80
C         COMPUTE ((M)**-1/2)T * (K) * (M)**-1/2                        STFM  90
C                STORE THE RESULT IN LOWER TRIANGLE INCLUDING DIAGONAL  STFM 100
C                       FORMAT IN K                                     STFM 110
      DO 20 I=1,MORDER                                                  STFM 120
      DO 20 J=I,MORDER                                                  STFM 130
20    STRSTF(I,J)=STRSTF(I,J)*AMASS(J)                                  STFM 140
      LOCSTF=0                                                          STFM 150
      DO 30 I=1,MORDER                                                  STFM 160
      DO 30 J=1,I                                                       STFM 170
      LOCSTF=LOCSTF+1                                                   STFM 180
30    STRSTF(LOCSTF,1)=STRSTF(J,I)*AMASS(J)                             STFM 190
      RETURN                                                            STFM 200
      END                                                               STFM 210
```

```
      SUBROUTINE FNALEV (EGVECT,AMASS,MORDER,NEIGVL,NROW,NRWTOR,T,T2,IRW
     1KP,IRWTOR)
      DIMENSION T2(1), IRWKP(1), IRWTOR(1)
      DIMENSION AMASS(1), EGVECT(1), T(1)
C
C         CALCULATE EIGENVECTORS FOR THE NON-REDUCED MATRIX FROM
C              TRANSFORMATION MATRICES STORED IN FILE 9 BY SUBROUTINE
C              REDUCE
C
      REWIND 1
      NACTRW=MORDER
      DO 90 I=1,NRWTOR
      READ (1) (T(L),L=1,NACTRW)
      IF (I.EQ.1) GO TO 40
      I4=1
      I1=1
      I2=NRWTOR
      I3=MORDER+1
      DO 30 J=1,NACTRW
      IF (I4.EQ.I) GO TO 10
      IF (IRWKP(I1).GT.IRWTOR(I2)) GO TO 20
   10 T2(I1)=T(J)
      I1=I1+1
      GO TO 30
   20 T2(I3)=T(J)
      I2=I2-1
      I3=I3+1
      I4=I4+1
   30 CONTINUE
      GO TO 60
   40 CONTINUE
      DO 50 J=1,NACTRW
   50 T2(J)=T(J)
   60 CONTINUE
      K=0
      DO 80 J=1,5
      SAVE=0.0
      DO 70 M=1,NACTRW
      L=K+M
   70 SAVE=SAVE+T2(M)*EGVECT(L)
      EGVECT(L+1)=SAVE
   80 K=K+NROW
      NACTRW=NACTRW+1
   90 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EEPNT (EGVALU,EGVECT,NEIGVL,NROW)                      EEPN  10
      DIMENSION EGVALU(1), EGVECT(1)                                    EEPN  20
C                                                                       EEPN  30
      K1=1                                                              EEPN  40
      K2=NROW                                                           EEPN  50
      DO 10 I=1,NEIGVL                                                  EEPN  60
      WRITE (6,20) I,EGVALU(I)                                          EEPN  70
C     IF ( I. .LT. 7 ) GO TO 40                                         EEPN  80
      WRITE (6,30) (EGVECT(K),K=K1,K2)                                  EEPN  90
      K1=K1+NROW                                                        EEPN 100
      K2=K2+NROW                                                        EEPN 110
10    CONTINUE                                                          EEPN 120
      RETURN                                                            EEPN 130
C                                                                       EEPN 140
20    FORMAT (13H0 EIGENVALUE(,I2,3H) =,E14.7)                          EEPN 150
30    FORMAT (10X30H THE CORRESPONDING EIGENVECTOR/(10X10E12.4))        EEPN 160
      END                                                               EEPN 170
```

```
      SUBROUTINE TRIDIA (NR,R,A,PQ)                                     TRID  10
C     TRI-DIAGONALIZES SYMMETRIC MATRIX BY HOUSEHOLDER  METHOD          TRID  20
C         NR = ORDER OF MATRIX                                          TRID  30
C         R  = 1-DIMENSIONAL ARRY OF NR/2*(NR+1) ELEMENTS               TRID  40
C           1. CONTAINS LOWER TRIANGULAR PART PLUS DIAGONAL OF MATRIX   TRID  50
C           2. AFTER TRIPLE-DIAGONALIZATION CONTAINS THE DIAGONAL PLUS  TRID  60
C                                                     W VECTORS         TRID  70
C         A  = 1-DIMENSIONAL ARRAY OF NR ELEMENTS, CONTAINS DIAGONAL    TRID  80
C           ELEMENTS OF TRIPLE-DIAGONAL MATRIX                          TRID  90
C         PQ = 1-DIMENSIONAL ARRAY OF NR ELEMENTS,                      TRID 100
C           1. CONTAINS ELEMENTS OF P VECTOR                            TRID 110
C           2. CONTAINS ELEMENTS OF Q VECTOR                            TRID 120
C           3. CONTAINS THE OFF-DIAGONAL TERM OF TRIPLE-DIAGONAL MATRIX TRID 130
C              (END RESULT =(3))                                        TRID 140
      DIMENSION R(1),PQ(1),A(1)                                         TRID 150
      IA=0                                                              TRID 160
      NR1=NR-1                                                          TRID 170
      DO 180 I=2,NR1                                                    TRID 180
      IA=IA+I                                                           TRID 190
C     CALCULATE ELEMENTS OF W VECTOR                                    TRID 200
      S=0.                                                              TRID 210
      JA=IA                                                             TRID 220
      DO 10 J=I,NR                                                      TRID 230
      S=S+R(JA)**2                                                      TRID 240
      JA=JA+J                                                           TRID 250
10    CONTINUE                                                          TRID 260
      SS=SQRT(S)                                                        TRID 270
      PQ(I-1)=-SIGN(SS,R(IA))                                           TRID 280
      IF (S) 20,190,20                                                  TRID 290
20    X=-PQ(I-1)/S                                                      TRID 300
30    R(IA)=SQRT(.500*(1.00+R(IA)*X))                                   TRID 310
      IF (S) 50,40,50                                                   TRID 320
40    R(IA)=0.                                                          TRID 330
50    IF (R(IA)) 60,200,60                                              TRID 340
60    X=.500*X/R(IA)                                                    TRID 350
70    JA=IA                                                             TRID 360
      DO 80 J=I,NR1                                                     TRID 370
      JA=JA+J                                                           TRID 380
      R(JA)=X*R(JA)                                                     TRID 390
80    CONTINUE                                                          TRID 400
C     CALCULATE ELEMENTS OF P VECTOR                                    TRID 410
      JAI=IA+1                                                          TRID 420
      DO 130 J=I,NR                                                     TRID 430
      JA=JAI                                                            TRID 440
      KA=IA                                                             TRID 450
      PQ(J)=0.                                                          TRID 460
      DO 120 K=I,NR                                                     TRID 470
      PQ(J)=PQ(J)+R(KA)*R(JA)                                           TRID 480
      IF (K-J) 90,100,100                                               TRID 490
90    JA=JA+1                                                           TRID 500
      GO TO 110                                                         TRID 510
100   JA=JA+K                                                           TRID 520
110   KA=KA+K                                                           TRID 530
120   CONTINUE                                                          TRID 540
      JAI=JAI+J                                                         TRID 550
```

```
130   CONTINUE                                                          TRID 560
C     CALCULATE ELEMENTS OF Q VECTOR                                    TRID 570
      AK=0.                                                             TRID 580
      JA=IA                                                             TRID 590
      DO 140 J=I,NR                                                     TRID 600
      AK=AK+R(JA)*PQ(J)                                                 TRID 610
      JA=JA+J                                                           TRID 620
140   CONTINUE                                                          TRID 630
      AK=2.00*AK                                                        TRID 640
      JA=IA                                                             TRID 650
      DO 150 J=I,NR                                                     TRID 660
      PQ(J)=2.00*PQ(J)-AK*R(JA)                                         TRID 670
      JA=JA+J                                                           TRID 680
150   CONTINUE                                                          TRID 690
C     CALCULATE ELEMENTS OF NEW R MATRIX                                TRID 700
      KK=IA                                                             TRID 710
      JA=IA                                                             TRID 720
      DO 170 J=I,NR                                                     TRID 730
      KA=IA                                                             TRID 740
      DO 160 K=I,J                                                      TRID 750
      KK=KK+1                                                           TRID 760
      R(KK)=R(KK)-PQ(K)*R(JA)-R(KA)*PQ(J)                               TRID 770
      KA=KA+K                                                           TRID 780
160   CONTINUE                                                          TRID 790
      KK=KK+I-1                                                         TRID 800
      JA=JA+J                                                           TRID 810
170   CONTINUE                                                          TRID 820
180   CONTINUE                                                          TRID 830
      GO TO 210                                                         TRID 840
190   X=0.                                                              TRID 850
      GO TO 30                                                          TRID 860
200   X=0.                                                              TRID 870
      GO TO 70                                                          TRID 880
C     SORT ALPHAS AND BETAS                                             TRID 890
210   IA=0                                                              TRID 900
      DO 220 I=1,NR                                                     TRID 910
      IA=IA+I                                                           TRID 920
      A(I)=R(IA)                                                        TRID 930
220   CONTINUE                                                          TRID 940
      PQ(NR)=R(IA-1)                                                    TRID 950
      N=NR                                                              TRID 960
230   N=N-1                                                             TRID 970
      PQ(N)=PQ(N-1)                                                     TRID 980
      IF (2-N) 230,240,240                                              TRID 990
240   PQ(1)=0.                                                          TRID1000
      RETURN                                                            TRID1010
      END                                                               TRID1020
```

```
      SUBROUTINE EIGVAL (LP,E,A,B,F,W,MVAL)
C     FINDS EIGENVALUES OF TRI-DIAGONAL MATRICES BY ORTEGA METHOD
C         LP = ORDER OF MATRIX
C         E  = 1-DIMENSIONAL ARRAY OF LP ELEMENTS. THE EIGENVALUFS ARE
C                STORED IN THIS ARRAY IN ASCENDING ORDER
C         A  = 1-DIMENSIONAL ARRAY OF LP ELEMENTS. CONTAINS DIAGONAL
C                ELEMENTS OF TRIPLE-DIAGONAL MATRIX
C         B  = 1-DIMENSIONAL ARRAY OF LP ELEMENTS. CONTAINS OFF-DIAGONAL
C                ELEMENTS OF TRIPLE-DIAGONAL MATRIX
C         F AND W = 1-DIMENSION ARRAY OF LP ELEMENTS EACH USFD AS
C                WORK AREAS
C         MVAL = NUMBER OR EIGENVALUES TO BE CALCULATED.
      DIMENSION E(1),A(1),B(1),F(1),W(1)
C     FIND UPPER AND LOWER BOUNDS AND NORMALIZE INPUT
      BD=ABS(A(1))
      DO 10 I=2,LP
10    BD=AMAX1(BD,ABS(A(I))+B(I)**2,0.00)
      BD=BD+1.
      DO 20 I=1,LP
      A(I)=A(I)/BD
      B(I)=B(I)/BD
      W(I)=1.
20    E(I)=-1.
      DO 230 K=1,MVAL
30    IF ((W(K)-E(K))/AMAX1(ABS(W(K)),ABS(E(K)),1.0E-9)-1.0E-12) 230,230
     1,40
40    X=(W(K)+E(K))*.5
C         FIND NUMBER OF EIGENVALUES, N, GREATER THAN OR EQUAL TO X
      S2=1.
      IS2=1
      F(1)=A(1)-X
      IF (F(1)) 50,60,60
50    IS1=-1
      S1=-1.
      N=0
      GO TO 70
60    IS1=1
      S1=1.
      N=1
70    DO 170 I=2,LP
      IF (B(I)) 80,120,80
80    IF (B(I-1)) 90,130,90
90    IF (ABS(F(I-1))+ABS(F(I-2))-1.0E-15) 100,110,110
100   F(I-1)=F(I-1)*1.0E15
      F(I-2)=F(I-2)*1.0E15
110   F(I)=(A(I)-X)*F(I-1)-B(I)**2*F(I-2)
      GO TO 140
120   F(I)=(A(I)-X)*SIGN(1.00,S1)
      GO TO 140
130   F(I)=(A(I)-X)*F(I-1)-SIGN(B(I)**2,S2)
140   S2=S1
      IS2=IS1
      IF (F(I)) 150,160,150
150   S1=SIGN(ABS(S1),F(I))
      IS1=S1
```

```
      IF (IS2+IS1) 160,170,160                                          EIVL 560
160   N=N+1                                                             EIVL 570
170   CONTINUE                                                          EIVL 580
C          TRAP EIGENVALUES IN SMALLER AND SMALLER BOUNDS               EIVL 590
      N=LP-N                                                            EIVL 600
      IF (N-K) 200,180,180                                              EIVL 610
180   DO 190 J=K,N                                                      EIVL 620
190   W(J)=X                                                            EIVL 630
200   N=N+1                                                             EIVL 640
      IF (LP-N) 30,210,210                                              EIVL 650
210   DO 220 J=N,LP                                                     EIVL 660
      IF (X-E(J)) 30,30,220                                             EIVL 670
220   E(J)=X                                                            EIVL 680
      GO TO 30                                                          EIVL 690
230   CONTINUE                                                          EIVL 700
C          RESTORE INPUT AND ORDER EIGENVALUES                          EIVL 710
      DO 240 I=1,LP                                                     EIVL 720
      A(I)=A(I)*BD                                                      EIVL 730
240   B(I)=B(I)*BD                                                      EIVL 740
      DO 250 I=1,MVAL                                                   EIVL 750
250   E(I)=(W(I)+E(I))*BD*.5                                            EIVL 760
      RETURN                                                            EIVL 770
      END                                                               EIVL 780
```

```
      SUBROUTINE EIGVEC (NR,E,R,A,B,V,P,Q)
C     GIVEN AN EIGENVALUE, FIND THE CORRESPONDING EIGENVECTOR USING
C                                                WILKINSON METHOD
C         NR = ORDER OF MATRIX
C         R  = 1-DIMENSION ARRAY OF NR/2*(NR+1) ELEMENTS CONTAINING
C              THE W VECTOR
C         E  = EIGENVALUE
C         A  = 1-DIMENSION ARRAY OF NR ELEMENTS CONTAINING DIAGONAL
C              TERMS OF TRIPLE-DIAGONAL MATRIX
C         B  = 1-DIMENSION ARRAY OF NR ELEMENTS CONTAINING OFF-DIAGONAL
C              TERMS OF TRIPLE-DIAGONAL MATRIX
C         V  = 1-DIMENSION ARRAY OF NR ELEMENTS CONTAINING THE
C              EIGENVECTOR OF THE ORIGINAL SYMMETRIC MATRIX
C         P AND Q = 1-DIMENSIONAL ARRAYS OF NR ELEMENTS EACH USED AS
C              WORK AREAS
      DIMENSION  R(1),A(1),B(1),V(1),P(1),Q(1)
      NR1=NR-1
C     SET UP SIMULTANEOUS EQUATIONS  I.E. COMPUTE P, Q, R
      X=A(1)-E
      Y=B(2)
      DO 70 I=1,NR1
      IF (ABS(B(I+1))-ABS(X)) 30,10,50
 10   IF (X) 30,20,30
 20   X=1.0E-10
 30   P(I)=X
      Q(I)=Y
      V(I)=0.
      X=A(I+1)-E-B(I+1)*Y/X
      IF (NR1-I) 40,70,40
 40   Y=B(I+2)
      GO TO 70
 50   P(I)=B(I+1)
      Q(I)=A(I+1)-E
      Z=X/P(I)
      X=Y-Z*Q(I)
      IF (NR1-I) 60,70,60
 60   V(I)=B(I+2)
      Y=-Z*V(I)
 70   CONTINUE
C     SOLVE FOR EIGENVECTOR OF TRI-DIAGONAL MATRIX
      IF (X) 90,80,90
 80   V(NR)=1.E10
      GO TO 100
 90   V(NR)=1.00/X
 100  I=NR1
      V(I)=(1.00-Q(I)*V(NR))/P(I)
      X=V(NR)**2+V(I)**2
 110  I=I-1
      IF (I) 120,130,120
 120  V(I)=(1.00-Q(I)*V(I+1)-V(I)*V(I+2))/P(I)
      X=X+V(I)**2
      GO TO 110
C     NORMALIZE VECTOR
 130  X=SQRT(X)
      DO 140 I=1,NR
```

```
      V(I)=V(I)/X                                                       EIVC 560
140   CONTINUE                                                          EIVC 570
C     COMPUTE EIGENVECTOR OF ORIGINAL MATRIX                            EIVC 580
      I=NR                                                              EIVC 590
150   I=I-1                                                             EIVC 600
      J=0                                                               EIVC 610
      K=1                                                               EIVC 620
160   K=K+1                                                             EIVC 630
      J=J+K                                                             EIVC 640
      IF (K-I) 160,170,160                                              EIVC 650
170   Y=0.                                                              EIVC 660
      JA=J                                                              EIVC 670
      DO 180 K=I,NR                                                     EIVC 680
      Y=Y+R(JA)*V(K)                                                    EIVC 690
      JA=JA+K                                                           EIVC 700
180   CONTINUE                                                          EIVC 710
      Y=2.00*Y                                                          EIVC 720
      JA=J                                                              EIVC 730
      DO 190 K=I,NR                                                     EIVC 740
      V(K)=V(K)-Y*R(JA)                                                 EIVC 750
      JA=JA+K                                                           EIVC 760
190   CONTINUE                                                          EIVC 770
      IF (I-2) 150,200,150                                              EIVC 780
200   RETURN                                                            EIVC 790
      END                                                               EIVC 800
```

```
      SUBROUTINE PRNT1 (EGVALU,EGVECT,AMASS,IRWKP,NROW,MORDER,IRWTOR,WOR
     1KA)
      DIMENSION WORKA(1)
      DIMENSION EGVALU(1), EGVECT(NROW,5), AMASS(1), IRWKP(1), IRWTOR(1)
      COMMON COM(30)
      EQUIVALENCE ( COM( 1 ), NJOINT )
      EQUIVALENCE ( COM(25), INDWNM )
      COMMON / CMAIN / COMAIN(1)
      EQUIVALENCE ( COMAIN(   1), RJX     )
      EQUIVALENCE ( COMAIN( 101), RJY     )
      EQUIVALENCE ( COMAIN( 201), RJZ     )
      DIMENSION WNX(5), WNY(5), WNZ(5), PX(5), PY(5), PZ(5), GM(5)
     1     , RJX(1), RJY(1), RJZ(1)
      DIMENSION ALPHA(6)
      DIMENSION IGVNO(5)
      DATA ALPHA / 8HX-TRANS., 8HY-TRANS., 8HZ-TRANS. ,
     1             8HX-ROTAT., 8HY-ROTAT., 8HZ-ROTAT. /
      DATA ITAPE / 9 /
      IF ( IFRIST .NE. 0 ) GO TO 4
      IFRIST = 1
      WRITE (6,280) (EGVALU(I),I=1,6)
C             SETUP EIGENVECTOR NO. SYMBOLS
      DO  2     I = 1, 5
    2 IGVNO(I) = I
      LR = 2
      MR = 6
      GO TO 6
    4 CONTINUE
C             UPDATE EIGENVECTOR NO. SYMBOLS
      DO  8     I = 1, 5
    8 IGVNO(I) = IGVNO(I) + 5
    6 CONTINUE
      LR = LR + 5
      MR = MR + 5
      DO  10  J = LR, MR
10    EGVALU(J)=SQRT(EGVALU(J))
      WRITE(6,290) (IGVNO(I),I=1,5), (EGVALU(I),I=LR,MR)
      IF (NROW.EQ.MORDER) GO TO 20
      WRITE (6,300)
      GO TO 40
20    DO 30 I=1,NROW
30    IRWKP(I)=I
      WRITE (6,400)
40    CONTINUE
      WRITE (6,310)
      LCOUNT=11
C
C             ZERO OUT SUMMING FIELDS
C
      DO 50 I=1,5
      WNX(I)=0.0
      WNY(I)=0.0
      WNZ(I)=0.0
      PX(I)=0.0
      PY(I)=0.0
```

```
      PZ(I)=0.0                                                         PRNT 560
50    GM(I  )=0.0                                                       PRNT 570
C                                                                       PRNT 580
C         CALCULATE ABOVE VARIABLES                                     PRNT 590
C                                                                       PRNT 600
      N =  (IRWKP(1)-1)/6+1                                             PRNT 610
      DO 140 I=1,MORDER                                                 PRNT 620
      M=IRWKP(I)                                                        PRNT 630
      K=M-(M-1)/6*6                                                     PRNT 640
      M=(M-1)/6+1                                                       PRNT 650
      IF (LCOUNT.LT.56) GO TO 60                                        PRNT 660
      WRITE (6,320)                                                     PRNT 670
      IF(NROW.EQ.MORDER)GO TO 500                                       PRNT 675
      WRITE (6,300)                                                     PRNT 680
      GO TO 501                                                         PRNT 682
  500 WRITE(6,400)                                                      PRNT 684
  501 CONTINUE                                                          PRNT 686
      WRITE(6,310) (IGVNO(I),I=1,5)                                     PRNT 690
      LCOUNT=5                                                          PRNT 700
      N=M                                                               PRNT 710
      GO TO 70                                                          PRNT 720
60    CONTINUE                                                          PRNT 730
      IF (N.EQ.M) GO TO 70                                              PRNT 740
      WRITE (6,330)                                                     PRNT 750
      LCOUNT=LCOUNT+1                                                   PRNT 760
      N=M                                                               PRNT 770
70    CONTINUE                                                          PRNT 780
C                                                                       PRNT 790
C         WRITE MODE SHAPES OF REDUCED SYSTEM                           PRNT 800
C                                                                       PRNT 810
      WRITE (6,340) M,ALPHA(K),(EGVECT(I,J),J=1,5)                      PRNT 820
      LCOUNT=LCOUNT+1                                                   PRNT 830
      IF (K.GT.3) GO TO 140                                             PRNT 840
      IF (K-2) 80,100,120                                               PRNT 850
C         X TRANSLATION MODE SHAPE ELEMENT                              PRNT 860
80    DO 90 J=1,5                                                       PRNT 870
      SAVE=EGVECT(I,J)*AMASS(I)                                         PRNT 880
      PX(J)=PX(J)+RJX(M)*SAVE                                           PRNT 890
      SAVE=SAVE*EGVECT(I,J)                                             PRNT 900
      WNY(J)=WNY(J)+SAVE                                                PRNT 910
90    WNZ(J)=WNZ(J)+SAVE                                                PRNT 920
      GO TO 140                                                         PRNT 930
C         Y TRANSLATION MODE SHAPE ELEMENT                              PRNT 940
100   DO 110 J=1,5                                                      PRNT 950
      SAVE=EGVECT(I,J)*AMASS(I)                                         PRNT 960
      PY(J)=PY(J)+RJY(M)*SAVE                                           PRNT 970
      SAVE=SAVE*EGVECT(I,J)                                             PRNT 980
      WNX(J)=WNX(J)+SAVE                                                PRNT 990
110   WNZ(J)=WNZ(J)+SAVE                                                PRNT1000
      GO TO 140                                                         PRNT1010
C         Z TRANSLATION MODE SHAPE ELEMENT                              PRNT1020
120   DO 130 J=1,5                                                      PRNT1030
      SAVE=EGVECT(I,J)*AMASS(I)                                         RRNT1040
      PZ(J)=PZ(J)+RJZ(M)*SAVE                                           PRNT1050
      SAVE=SAVE*EGVECT(I,J)                                             PRNT1060
      WNX(J)=WNX(J)+SAVE                                                PRNT1070
```

```
130   WNY(J)=WNY(J)+SAVE                                                PRNT1080
140   CONTINUE                                                          PRNT1090
      DO 150 I=1,5                                                      PRNT1100
      DO 150 K=1,MORDER                                                 PRNT1110
      GM(I) = GM(I) + AMASS(K)*EGVECT(K,I)**2                           PRNT1120
  150 CONTINUE                                                          PRNT1130
      WRITE (6,350)                                                     PRNT1140
      WRITE(6,360) (IGVNO(I),I=1,5)                                     PRNT1150
      WRITE (6,370) (WNX(I),I=1,5),(WNY(I),I=1,5),(WNZ(I),I=1,5),(PX(I),PRNT1160
     1I=1,5),(PY(I),I=1,5),(PZ(I),I=1,5), (GM( I ),I=1,5 )              PRNT1170
      RETURN                                                            PRNT1180
C                                                                       PRNT1190
C         PRINT MODE  SHAPES FOR COMPLETE SYSTEM                        PRNT1200
C                                                                       PRNT1210
      ENTRY PRNT2                                                       PRNT1220
      IF ((NROW.EQ.MORDER).AND.(INDWNM.EQ.0)) RETURN                    PRNT1230
      IF (INDWNM.EQ.0) GO TO 170                                        PRNT1240
      SAVE=NJOINT                                                       PRNT1250
      SAVE1=MORDER                                                      PRNT1260
      WRITE (ITAPE,380) SAVE,SAVE1                                      PRNT1270
      DO 160 I=1,MORDER                                                 PRNT1280
160   WORKA(I)=IRWKP(I)                                                 PRNT1290
      CALL P1A721 (121,WORKA,MORDER,ITAPE)                              PRNT1300
      WRITE (ITAPE,390) (RJY(I),RJZ(I),RJX(I),I ,I=1,NJOINT)            PRNT1310
      LC=0                                                              PRNT1320
170   CONTINUE                                                          PRNT1330
      IF (NROW.EQ.MORDER) GO TO 180                                     PRNT1340
      WRITE(6,290) (IGVNO(I),I=1,5), (EGVALU(I),I=LR,MR)                PRNT1350
      WRITE (6,400)                                                     PRNT1360
      WRITE(6,310) (IGVNO(I),I=1,5)                                     PRNT1370
      LCOUNT=11                                                         PRNT1380
180   CONTINUE                                                          PRNT1390
      M=NROW-MORDER                                                     PRNT1400
      K=1                                                               PRNT1410
      N=0                                                               PRNT1420
      DO 260 I=1,NJOINT                                                 PRNT1430
      DO 250 J=1,3                                                      PRNT1440
      N=N+1                                                             PRNT1450
190   IF (N.NE.IRWKP(K)) GO TO 200                                      PRNT1460
      N1=K                                                              PRNT1470
      K=K+1                                                             PRNT1480
      GO TO 220                                                         PRNT1490
200   IF (N.NE.IRWTOR(M)) GO TO 210                                     PRNT1500
      N1=NROW-M+1                                                       PRNT1510
      M=M-1                                                             PRNT1520
      GO TO 220                                                         PRNT1530
210   CONTINUE                                                          PRNT1540
      IF (N.GT.IRWKP(K)) K=K+1                                          PRNT1550
      IF (N.GT.IRWTOR(M)) M=M-1                                         PRNT1560
      GO TO 190                                                         PRNT1570
220   CONTINUE                                                          PRNT1580
      IF (INDWNM.EQ.0) GO TO 230                                        PRNT1590
      LC=LC+1                                                           PRNT1600
      WRITE (ITAPE,410) (EGVECT(N1,L),L=1,5),LC                         PRNT1610
230   CONTINUE                                                          PRNT1620
      IF (NROW.EQ.MORDER) GO TO 250                                     PRNT1630
```

```
      WRITE (6,340) I,ALPHA(J),(EGVECT(N1,L),L=1,5)                     PRNT1640
      LCOUNT=LCOUNT+1                                                   PRNT1650
      IF (LCOUNT.LT.56) GO TO 240                                       PRNT1660
      WRITE (6,320)                                                     PRNT1670
      WRITE (6,400)                                                     PRNT1680
      WRITE(6,310) (IGVNO(I),I=1,5)                                     PRNT1690
      LCOUNT=5                                                          PRNT1700
      GO TO 250                                                         PRNT1710
240   CONTINUE                                                          PRNT1720
      IF (J.NE.3) GO TO 250                                             PRNT1730
      WRITE (6,330)                                                     PRNT1740
      LCOUNT=LCOUNT+1                                                   PRNT1750
250   CONTINUE                                                          PRNT1760
      N=N+3                                                             PRNT1770
260   CONTINUE                                                          PRNT1780
      IF (INDWNM.EQ.0) GO TO 270                                        PRNT1790
      LR1 = LR - 1                                                      PRNT1800
      WRITE (ITAPE,420) (EGVALU(LR1+I),GM(I  ),I,I=1,5)                 PRNT1810
      WRITE (ITAPE,430) (WNX(I),WNY(I),WNZ(I),PX(I),PY(I),PZ(I),I,I=1,5)PRNT1820
      CALL P1A721 (721,AMASS,MORDER,ITAPE)                              PRNT1830
      END FILE ITAPE                                                    PRNT1840
270   CONTINUE                                                          PRNT1850
      RETURN                                                            PRNT1860
C                                                                       PRNT1870
C                                                                       PRNT1880
C         INITIALIZE FOR EACH EIGENVALUE GROUP                          PRNT1890
C                                                                       PRNT1900
      ENTRY PRNT0                                                       PRNT1910
      IFRIST = 0                                                        PRNT1920
      RETURN                                                            PRNT1930
280   FORMAT (45H1EIGENVALUES ASSOCIATED WITH RIGID BODY MODES//(5XE14.7PRNT1940
     1))                                                                PRNT1950
290   FORMAT (1H1,41X33HNATURAL FREQUENCIES (RADIANS/SEC)//             PRNT1960
     1     20X5I18//27X5E18.8//)                                        PRNT1970
300   FORMAT (36X44HCORRESPONDING MODE SHAPES FOR REDUCED SYSTEM)       PRNT1980
310   FORMAT (24H0  NODAL POINT   ELEMENT/25H     NUMBER     REFERENCE, PRNT1990
     1    11XI2, 4I18/ )                                                PRNT2000
320   FORMAT (1H1)                                                      PRNT2010
330   FORMAT (1H )                                                      PRNT2020
340   FORMAT (6XI3,8XA8,2X5E18.8)                                       PRNT2030
350   FORMAT (1H1,42X30HGENERALIZED INERTIA PROPERTIES/)                PRNT2040
  360 FORMAT ( 11H  VARIABLES,9X5I18/)                                  PRNT2050
370   FORMAT (1H04X,3HWNX,19X5E18.8/1H04X,3HWNY,19X5E18.8/1H04X,3HWNZ,19PRNT2060
     1X5E18.8/1H04X,3H PX,19X5E18.8/1H04X,3H PY,19X5E18.8/1H04X,3H PZ,19PRNT2070
     2X5E18.8/1H05X,15HGM  DIAGONAL   ,6X5E18.8/(27X5E18.8))            PRNT2080
380   FORMAT (6X2H21,2X10H   5.0    ,2F10.3,37X3H  1)                   PRNT2090
390   FORMAT (5X3H221,2X3E10.3,37XI3)                                   PRNT2100
400   FORMAT (36X45HCORRESPONDING MODE SHAPES FOR COMPLETE SYSTEM)      PRNT2110
410   FORMAT (5X3H3212X,5E10.3,17XI3)                                   PRNT2120
420   FORMAT (5X3H4212X,2E10.3,47XI3)                                   PRNT2130
430   FORMAT (5X3H5212X,6E10.3,7XI3)                                    PRNT2140
      END                                                               PRNT2150
```

```
      SUBROUTINE P1A721 (NO,ARRAY,LNGTH,I6)                             P1A7  10
      DIMENSION ARRAY(1)                                                P1A7  20
      K=1                                                               P1A7  30
      I=1                                                               P1A7  40
      DO 10 J=6,LNGTH,6                                                 P1A7  50
      WRITE (I6,20) NO,(ARRAY(L),L=I,J),K                               P1A7  60
      K=K+1                                                             P1A7  70
      I=I+6                                                             P1A7  80
   10 CONTINUE                                                          P1A7  90
      IF (I.GT.LNGTH) RETURN                                            P1A7 100
      WRITE (I6,20) NO,(ARRAY(L),L=I,LNGTH),K                           P1A7 110
      RETURN                                                            P1A7 120
C                                                                       P1A7 130
   20 FORMAT (5X,I3,2X,6E10.3,7XI3)                                     P1A7 140
      END                                                               P1A7 150
```

```
      OVERLAY ( GEARRT,5,0)                                             OV50  10

      PROGRAM GEAREX                                                    GEAR  10
      COMMON / CMAIN / COMSAP(2000)                                     GEAR  20
      EQUIVALENCE ( COMSAP(1), LGRTYP )                                 GEAR  30
C                 I/O UNIT  5    EQUALS   INPUT UNIT                    GEAR  40
C                 I/O UNIT  6    EQUALS   OUTPUT UNIT                   GEAR  50
C                 I/O UNIT  1    EQUALS   WORK AREA--V.R.               GEAR  60
C                 ZERO OUT COMMON                                       GEAR  70
      DO  150  I = 1, 2000                                              GEAR  80
  150 COMSAP(I) = 0.0                                                   GEAR  90
C                                                                       GEAR 100
C        CALL THE INPUT ROUTINE                                         GEAR 110
C                                                                       GEAR 120
      CALL OVERLAY ( 6LGEARRT,5,1,6HRECALL)                             GEAR 130
      IF( LGRTYP .EQ. 2) GO TO 200                                      GEAR 140
C                                                                       GEAR 150
C        CALL LANDING GEAR ANALYSIS ROUTINE FOR INVERTED TRIPOD GEAR    GEAR 160
C                                                                       GEAR 170
      CALL OVERLAY ( 6LGEARRT,5,2,6HRECALL)                             GEAR 180
      GO TO 300                                                         GEAR 190
  200 CONTINUE                                                          GEAR 200
C                                                                       GEAR 210
C        CALL LANDING GEAR ANALYSIS ROUTINE FOR CANTILEVER GEAR         GEAR 220
C                                                                       GEAR 230
      CALL OVERLAY ( 6LGEARRT,5,3,6HRECALL)                             GEAR 240
  300 CONTINUE                                                          GEAR 250
      RETURN                                                            GEAR 260
      END                                                               GEAR 270
```

```
      SUBROUTINE OUTPUT
      COMMON / CMAIN  /
     1     LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB   ,
     2     NSTEP , NITER , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),
     3     IGLDIR, NMATL , IST    , IMATL(4)       , MNO    , FPCRST(3),
     4     INDPL(4)       , ELASBN(4)      , ELASAX(4)      ,
     5     BINERT(4)      , AREA(4)        , IRET(4)        ,
     6     IFIRST(4)      , SCMAX(4)       , STMAX(4)       ,
     7     SRULC(4)   , SRULT(4),  CDC(5,4), SRC(5,4)  ,
     8    CDT(5,4)        , CRC(5,4)       , SRT(5,4)       ,
     9     PFC(5,4)       , PFT(5,4)       , NPN    , XYZPOS(3,5)
     9   , DC(3,3)        , PHI            , THTA          , PSI
     9   , CGYCDL(3),  EGYCDL(3)
      COMMON / CMAIN  /
     A     NITERS, NDISPS, STLN2 ,
     B     IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),
     C     STLN(4)        , STLNO(4)                , STRP(4)   ,
     D     SPNGC(4)       , SPNGT(4)       , DISTC(4)       ,
     E     DISTT(4)       , PFORC (4)      , PFORT(4)       ,
     F     FOREVC(4)      , FOREVT(4)      , IPOSC(4)       ,
     G     IPOST(4)       ,INDULC(4)      , INDULT(4)      ,
     H     STR(4)         , AXIALF(4)      , AXSTIF(4)      ,
     I     IPREV(4)       , XYZDS (3)      , SFORCE(30)     ,
     J     CFORCE(30)     , SUMFM4(6)      ,R24XYZ(3)     ,
     K     TM(36)         , FMVCTL(12,4)    , FMVCTG(12,4)    ,
     L     YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),
     M     YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)
      DATA IPCNT, IPAGCT / 99, 0 /
      IF ( IPCNT .LT. 2 ) GO TO 700
      IPCNT = 0.0
      IPAGCT  = IPAGCT + 1
      WRITE(6,799)IPAGCT
  799 FORMAT ( 1H1, 123X 4HPAGE, I4 )
      GO TO 570
  700 CONTINUE
      WRITE(6,760)
  760 FORMAT ( 1H0, 33X6(10H**********) )
  570 CONTINUE
      IPCNT = IPCNT + 1
      NDISPS = NDISPS + 1
      WRITE(6,800)NDISPS, NITERS
  800 FORMAT ( 1H0,53X17H INCREMENTAL STEP,I5 /
     1   38XI4,47H ITERATIONS REQUIRED FOR TOLERANCE SATISFACTION / )
      ENTRY OUTPT1
      N = 1
      WRITE(6, 999)
  999 FORMAT ( 8H          ,13X22HCOORDINATE DEFINITIONS,  20X
     1                           15HEXTERNAL FORCES        ,  24X
     2                           16HEXTERNAL MOMENTS   /   8H   NODAL
     2       4X 3( 7X25HSURFACE COORDINATE SYSTEM, 7X)   /
     3   3X5HPOINT, 4X3(6X1HX,12X1HY,12X1HZ,6X), / )
      DO 210   I = 1, NPOINT
      IF ( I .EQ. 4 ) GO TO 200
      WRITE(6,1000)I,(XYZPOS(J,I),J=1,3),(FMVCTG(J,N),J=1,3 ),
     1                 (FMVCTG(J,N),J=7,9)
```

```
      N = N + 1                                                         OUTP 560
      GO TO 210                                                         OUTP 570
  200 CONTINUE                                                          OUTP 580
      WRITE(6,1000)I,(XYZPOS(J,I),J=1,3),(SUMFM4(J),J=1,6)              OUTP 590
  210 CONTINUE                                                          OUTP 600
 1000 FORMAT (4XI2,4X9E13.5)                                            OUTP 610
      N = 1                                                             OUTP 620
      WRITE(6,1099)                                                     OUTP 630
 1099 FORMAT (20H0STRUCTURAL     AXIAL, 7X7H AXIAL , 7X5HAXIAL,         OUTP 640
     *      20X5HSHEAR,                                                 OUTP 650
     1     8X5HSHEAR, 21X5HSHEAR, 8X5HSHEAR  /                          OUTP 660
     2          48H  ELEMENT       FORCE        STROKE       STIFFNESS, 18X OUTP 670
     3        5HFORCE,  8X6HMOMENT, 20X5HFORCE, 8X6HMOMENT / )          OUTP 680
      DO  220 I = 1, NELEM                                              OUTP 690
C               ONE LOCAL SHEAR FORCE/MOMENT IS ZERO                    OUTP 700
      SHARFP = YSHRFP(I) + ZSHRFP(I)                                    OUTP 710
      SHARMP = YSHRMP(I) + ZSHRMP(I)                                    OUTP 720
      SHARFQ = YSHRFQ(I) + ZSHRFQ(I)                                    OUTP 730
      SHARMQ = YSHRMQ(I) + ZSHRMQ(I)                                    OUTP 740
      J      =  I                                                       OUTP 750
      IF ( I .EQ. 4 ) J = 5                                             OUTP 760
  220 WRITE(6,1100)I,AXIALF(I),STR(I) ,AXSTIF(I),J,SHARFP,SHARMP,       OUTP 770
     1                                                  SHARFQ,SHARMQ   OUTP 780
 1100 FORMAT ( 4XI2,4X3E13.5,3X4HNODE I2,3X2E13.5,3X7HNODE  4,3X2E13.5) OUTP 790
      WRITE(6,1199) NFPCRL                                              OUTP 800
 1199 FORMAT( 1H0,77X 42HNUMBER OF UNCRUSHED FOOTPAD CRUSH LEVELS =,I2 /OUTP 810
     123X 21HTOTAL ENERGY ABSORBED,38X 36HFOOTPAD ATTENUATION ENERGY THIOUTP 820
     1S STEP  /                                                         OUTP 830
     21X, 2(4X25HSURFACE COORDINATE SYSTEM9X24HLANDER COORDINATE SYSTEM OUTP 840
     3 ,4X) / 1X4(5X1HX,10X1HY,10X1HZ,5X) )                             OUTP 850
      WRITE(6,1200)(EGYXYZ(I),I=1,3),(EGYCDL(I),I=1,3),                 OUTP 860
     1              (CGYXYZ(I),I=1,3),(CGYCDL(I),I=1,3)                 OUTP 870
 1200 FORMAT ( 1H0, 12E11.3 )                                           OUTP 880
      RETURN                                                            OUTP 890
      END                                                               OUTP 900
```

```
      SUBROUTINE ENERGY( FORC2 )
      DIMENSION FORC2(3)
      DIMENSION FORCL1(3), FORCL2(3), XYZDSL(3)
      DIMENSION FORC1(3),           IXYZ(3)
      COMMON / CMAIN  /
     1    LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB   ,
     2    NSTEP , NITER , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),
     3    IGLDIR, NMATL , IST    , IMATL(4)      , MNO    , FPCRST(3),
     4    INDPL(4)       , ELASBN(4)      , ELASAX(4)      ,
     5    BINERT(4)      , AREA(4)        , IRET(4)        ,
     6    IFIRST(4)      , SCMAX(4)       , STMAX(4)       ,
     7    SRULC(4)   , SRULT(4),  CDC(5,4), SRC(5,4)   ,
     8   CDT(5,4)        , CRC(5,4)       , SRT(5,4)       ,
     9    PFC(5,4)       , PFT(5,4)       , NPN    , XYZPOS(3,5)
     9  , DC(3,3)        , PHI            , THTA         , PSI
     9  , CGYCDL(3),   EGYCDL(3)
      COMMON / CMAIN  /
     A    NITERS, NDISPS, STLN2 ,
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),
     C    STLN(4)        , STLNO(4)                , STRP(4)   ,
     D    SPNGC(4)       , SPNGT(4)       , DISTC(4)       ,
     E    DISTT(4)       , PFORC (4)      , PFORT(4)       ,
     F    FOREVC(4)      , FOREVT(4)      , IPOSC(4)       ,
     G    IPOST(4)       , INDULC(4)      , INDULT(4)      ,
     H    STR(4)         , AXIALF(4)      , AXSTIF(4)      ,
     I    IPREV(4)       , XYZDS (3)      , SFORCE(30)     ,
     J    CFORCE(30)     , SUMFM4(6)      ,R24XYZ(3)     ,
     K    TM(36)         , FMVCTL(12,4)    , FMVCTG(12,4)    ,
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)
      DATA IXYZ / 1HX, 1HY, 1HZ /
C         ZERO OUT FOOTPAD CRUSH ENERGY
      DO 10 I = 1, 3
   10 CGYXYZ(I) = 0.0
      IF ( IFPATN .EQ. 2 ) GO TO 50
      IF ( NFPCRL .EQ. 0 ) GO TO 50
C         CALCULATE THE FORCE COMPONENT AT THE FOOTPAD JOINT
C              NORMAL TO THE LANDING SURFACE
      RESULF = FORC2( IDSPDR )
C
C         CALCULATE THE FOOTPAD CRUSH ENERGY FOR THIS STEP IN SURFACE
C              COORDINATES AND ADD IT TO TOTAL ENERGY IN SURFACE
C              COORDINATES
C
      K  = NFPCRL
      J  = NFPCRL
      DO    30    I =1,J
      IF ( RESULF .LT. FPCRLD(K) ) GO TO 30
      NFPCRL = NFPCRL - 1
      CGYXYZ(IDSPDR) = CGYXYZ(IDSPDR) + FPCRLD(K)*FPCRST(K)
   30 K   = K - 1
      IF ( NFPCRL .EQ. J ) GO TO 50
      EGYXYZ(IDSPDR) = EGYXYZ(IDSPDR) + CGYXYZ(IDSPDR)
   50 CONTINUE
C
```

```
C         CALCULATE FOOTPAD CRUSH ENERGY FOR THIS STEP IN LANDER          ENGY 540
C              COORDINATES BY DIRECT TRANSFORMATION.                      ENGY 550
C         CALCULATE ENERGY FOR THIS STEP DUE TO DISPLACEMENT OF FOOTPAD   ENGY 560
C              JOINT IN BOTH SURFACE AND LANDER COORDINATES               ENGY 565
C                                                                         ENGY 570
      DO  410  I = 1, 3                                                   ENGY 580
      CGYCDL(I) = 0.0                                                     ENGY 590
      FORCL2(I) = 0.0                                                     ENGY 600
      XYZDSL(I) = 0.0                                                     ENGY 610
      DO  400  J = 1, 3                                                   ENGY 620
      CGYCDL(I) = CGYCDL(I) + DC(J,I)*CGYXYZ(J)                           ENGY 630
      FORCL2(I) = FORCL2(I) + DC(J,I)* FORC2(J)                           ENGY 640
      XYZDSL(I) = XYZDSL(I) +  DC(J,I)* XYZDS (J)                         ENGY 650
  400 CONTINUE                                                            ENGY 660
C                                                                         ENGY 661
C         CALCULATE TOTAL ENERGY IN LANDER COORDINATES AS THE SUM OF      ENGY 662
C              THE PREVIOUS TOTAL AND THE FOOTPAD CRUSH ENERGY FOR        ENGY 663
C              THIS STEP AND THE ENERGY DUE TO FOOTPAD JOINT              ENGY 664
C              DISPLACEMENT FOR THIS STEP                                 ENGY 665
C                                                                         ENGY 666
      EGYCDL(I) = EGYCDL(I) +  XYZDSL(I) *(FORCL1(I)+FORCL2(I)) *.5       ENGY 670
      EGYCDL(I) = EGYCDL(I) + CGYCDL(I)                                   ENGY 680
  410 FORCL1(I) = FORCL2(I)                                               ENGY 690
C                                                                         ENGY 700
C         CALCULATE TOTAL ENERGY IN SURFACE COORDINATES AS THE SUM OF     ENGY 710
C              THE PREVIOUS TOTAL AND THE FOOTPAD CRUSH ENERGY FOR        ENGY 720
C              THIS STEP AND THE ENERGY DUE TO FOOTPAD JOINT              ENGY 722
C              DISPLACEMENT FOR THIS STEP                                 ENGY 724
C                                                                         ENGY 730
      DO  60   I = 1, 3                                                   ENGY 740
      EGYXYZ(I) = EGYXYZ(I) + XYZDS(I)*(FORC1(I) + FORC2(I)) * .5         ENGY 750
   60 FORC1(I) = FORC2(I)                                                 ENGY 760
C         CHECK ENERGY LIMIT                                              ENGY 770
      IF ( IGLDIR .EQ. .0 ) GO TO 80                                      ENGY 780
      IF ( ENGABS .GE. 0.) GO TO 64                                       ENGY 790
      IF ( EGYXYZ(IGLDIR) .GT. ENGABS ) GO TO 80                          ENGY 800
      GO TO 66                                                            ENGY 810
   64 CONTINUE                                                            ENGY 820
      IF (  EGYXYZ( IGLDIR )  .LT. ENGABS ) GO TO 80                      ENGY 830
   66 CONTINUE                                                            ENGY 840
      WRITE(6,71)                                                         ENGY 842
   71 FORMAT(IH0, 33X 6(10H**********) )                                  ENGY 844
      WRITE(6,70)EGYXYZ(IGLDIR),IXYZ(IGLDIR),ENGABS                       ENGY 850
   70 FORMAT(24H0**** THE TOTAL ENERGY (E14.7,                            ENGY 860
     1        9H) IN THE A1,19H SURFACE COORDINATE,                       ENGY 870
     2        38H DIRECTION EXCEEDS THE ENERGY CUTOFF (,                  ENGY 880
     3        E14.7,6H) ****)                                             ENGY 885
      IGLDIR = 0                                                          ENGY 890
   80 CONTINUE                                                            ENGY 900
      RETURN                                                              ENGY 910
      ENTRY ENERG1                                                        ENGY 920
      DO 100 I = 1, 3                                                     ENGY 930
      FORCL1(I) = 0.0                                                     ENGY 940
      FORC1(I) = 0.0                                                      ENGY 950
  100 EGYXYZ(I)= 0.0                                                      ENGY 960
      NFPCR1 = NFPCRL                                                     ENGY 970
```

```
      RETURN                                                            ENGY 980
      END                                                               ENGY 990
```

```
      SUBROUTINE TRNFSM( STIFML, TRANSM,NORDER )                        TRSM  10
      DIMENSION STIFML(NORDER,NORDER),TRANSM(3,3)                       TRSM  20
      DIMENSION A(12), IJ(4), IJC(12)                                   TRSM  30
      DIMENSION IRC(12)                                                 TRSM  40
      DATA IJ / 0,3,6,9 /                                               TRSM  50
      DATA IJC / 3*0, 3*3, 3*6, 3*9 /                                   TRSM  60
      DATA IRC / 1,2,3,1,2,3,1,2,3,1,2,3 /                              TRSM  70
      ITINJO=1                                                          TRSM  80
C                                                                       TRSM  90
C         MULTIPLY TRANSFORMATION MATRIX TIMES STIFFNESS MATRIX         TRSM 100
C                   STORE RESULT IN STIFFNESS MATRIX                    TRSM 110
C                                                                       TRSM 120
      NORDR3 = NORDER / 3                                               TRSM 130
      DO  40 K = 1, NORDER                                              TRSM 140
      ICR=0                                                             TRSM 150
      DO  30 IC =1, NORDR3                                              TRSM 160
      DO 30 I=1,3                                                       TRSM 170
      ICR=ICR+1                                                         TRSM 180
      A(ICR)=0.0                                                        TRSM 190
      DO 30 J=1,3                                                       TRSM 200
      JJ=J+IJ(IC)                                                       TRSM 210
      IF (STIFML(K,JJ).EQ.0.) GO TO 30                                  TRSM 220
      A(ICR)=A(ICR)+STIFML(K,JJ)*TRANSM(J,I)                            TRSM 230
30    CONTINUE                                                          TRSM 240
      DO  40 J = 1, NORDER                                              TRSM 250
40    STIFML(K,J)=A(J)                                                  TRSM 260
C                                                                       TRSM 270
C                                                                       TRSM 280
C         MULTIPLY THE RESULT OF THE ABOVE TIMES THE TRANSPOSE OF THE   TRSM 290
C                   TRANSFORMATION MATRIX                               TRSM 300
C                                                                       TRSM 310
      DO  60 K = 1, NORDER                                              TRSM 320
      DO  50 ICR = K, NORDER                                            TRSM 330
      A(ICR)=0.0                                                        TRSM 340
      I=IRC(ICR)                                                        TRSM 350
      DO 50 J=1,3                                                       TRSM 360
      JJ=J+IJC(ICR)                                                     TRSM 370
50    A(ICR)=A(ICR)+STIFML(JJ,K)*TRANSM(J,I)                            TRSM 380
      DO  60 J = K, NORDER                                              TRSM 390
60    STIFML(J,K)=A(J)                                                  TRSM 400
      ICR = NORDER - 1                                                  TRSM 410
      DO  70 I = 1, ICR                                                 TRSM 420
      IC=I+1                                                            TRSM 430
      DO  70 J = IC,NORDER                                              TRSM 440
70    STIFML(I,J)=STIFML(J,I)                                           TRSM 450
C                                                                       TRSM 460
C         AS A RESULT OF THE ABOVE, STIFML CONTAINS THE STIFFNESS       TRSM 470
C              MATRIX TRANSFORMED TO THE GLOBAL  COORDINATE SYSTEM      TRSM 480
C                                                                       TRSM 490
      RETURN                                                            TRSM 500
      END                                                               TRSM 510
```

```
      SUBROUTINE TRALMG( FMG, FML, TM, NG )
C         F.M. GLOBAL = (TM)T . (F.M. LOCAL)
C             FM. VECTOR =FXP,FYP,FZP,FXQ,FYQ,FZQ,MXP...,MXQ,MYQ,MZQ
      DIMENSION FMG(12), FML(12), TM(3,3)
      N = 0
      M = 0
      DO  30   I = 1, NG
C
C         TRANSFORM LOCAL FORCE-MOMENTS TO GLOBAL FORCE-MOMENTS
C
      DO  20   J = 1, 3
      M      = M + 1
      FMG(M) = 0.0
      DO  20   K = 1, 3
      NK = N + K
      FMG(M) = FMG(M) + TM(K,J) * FML(NK)
   20 CONTINUE
   30 N  = N + 3
      RETURN
      END
```

```
      SUBROUTINE STFMIT( STIFML, PO, DPQ, AE )                          STAX  10
      DIMENSION STIFML(6,6)                                             STAX  20
C                                                                       STAX  30
C         COMPUTE THE LOCAL STIFFNESS MATRIX OF AN AXIAL STRUT          STAX  40
C                                                                       STAX  50
C                    PO = CURRENT AXIAL FORCE                           STAX  60
C                    AE = CURRENT AXIAL STIFFNESS                       STAX  70
C                    DPQ= LENGTH OF THE PQ ELEMENT                      STAX  80
      DO  100  I = 2, 6                                                 STAX  90
      K  = I - 1                                                        STAX 100
      DO  100  J = 1, K                                                 STAX 110
  100 STIFML(I,J) =  0.00                                               STAX 120
      STIFML(1,1) = AE                                                  STAX 130
      STIFML(2,2) = PO / DPQ                                            STAX 140
      STIFML(3,3) = STIFML(2,2)                                         STAX 150
      STIFML(4,4) = AE                                                  STAX 160
      STIFML(5,5) = STIFML(2,2)                                         STAX 170
      STIFML(6,6) = STIFML(2,2)                                         STAX 180
      STIFML(4,1) = -AE                                                 STAX 190
      STIFML(5,2) = -STIFML(2,2)                                        STAX 200
      STIFML(6,3) = -STIFML(2,2)                                        STAX 210
      DO  200  I = 2,6                                                  STAX 220
      K    =  I - 1                                                     STAX 230
      DO  200  J = 1, K                                                 STAX 240
  200 STIFML(J,I) = STIFML(I,J)                                         STAX 250
      RETURN                                                            STAX 260
      END                                                               STAX 270
```

```
      SUBROUTINE TRANSM( P, Q, R, TM, TOTLPQ )
      DIMENSION P(3), Q(3), TM(3,3), R(3)
      DOUBLE PRECISION X, Y, Z, TOTL   , TMD(3,3), DPE, DER
C               L1 M1 N1      THE TRANSFORMATION MATRIX REQUIRED TO
C         TM =  L2 M2 N2        CONVERT GLOBAL COORDINATES TO LOCAL
C               L3 M3 N3        COORDINATES
      X = Q(1) - P(1)
      Y = Q(2) - P(2)
      Z = Q(3) - P(3)
C         COMPUTE THE TOTAL LENGTH OF THE BAR PQ
      TOTL   = DSQRT( X*X + Y*Y + Z*Z )
      TOTLPQ=TOTL
C         COMPUTE THE DIRECTION COSINES OF THE LOCAL X AXIS    PQ
      TMD(1,1)= X / TOTL
      TMD(1,2) = Y / TOTL
      TMD(1,3) = Z / TOTL
      X = R(1) - P(1)
      Y = R(2) - P(2)
      Z = R(3) - P(3)
      DPE = TMD(1,1)*X + TMD(1,2)*Y + TMD(1,3)*Z
      DER = DSQRT( X*X + Y*Y + Z*Z - DPE*DPE )
C         COMPUTE THE DIRECTION COSINES OF THE LOCAL Y AXIS    ER
C              WHERE E IS THE POINT OF INTERSECTION ON PQ OF THE
C              PERPENDICULAR DRAWN FROM R TO PQ
      TMD(2,1) = ( X - TMD(1,1)*DPE )/DER
      TMD(2,2) = ( Y - TMD(1,2)*DPE )/DER
      TMD(2,3) = ( Z - TMD(1,3) *DPE )/DER
C         COMPUTE THE DIRECTION COSINES OF THE LOCAL Z AXIS    ERXPQ
      TMD(3,1) = TMD(1,2)*TMD(2,3) - TMD(2,2)*TMD(1,3)
      TMD(3,2) = TMD(2,1)*TMD(1,3) - TMD(1,1)*TMD(2,3)
      TMD(3,3)= TMD(1,1)*TMD(2,2) - TMD(2,1)*TMD(1,2)
      DO 10 I=1,3
      DO 10 J=1,3
   10 TM(I,J) = TMD(I,J)
      RETURN
      END
```

```
      SUBROUTINE DMFSS(A,N,EPS,IRANK,TRAC)
C
C        SUBROUTINE DMFSS DETERMINES THE RANK (IRANK) AND LINEARLY
C        INDEPENDENT ROWS AND COLUMNS OF A SYMMETRIC POSITIVE
C        SEMI-DEFINITE MATRIX (A) OF ORDER N AND PREPARES THE MATRIX
C        FOR CALCULATION OF THE LEAST SQUARES SOLUTION OF MINIMAL LENGTH
C
C
C        DIMENSIONED DUMMY VARIABLES
      DIMENSION A(1),TRAC(1)
      DOUBLE PRECISION SUM,A,TRAC,PIV,HOLD
C
C        TEST OF SPECIFIED DIMENSION
      IF(N)36,36,1
C
C        INITIALIZE TRIANGULAR FACTORIZATION
    1 IRANK=0
      ISUB=0
      KPIV=0
      J=0
      PIV=0.D0
C
C        SEARCH FIRST PIVOT ELEMENT
      DO 3 K=1,N
      J=J+K
      TRAC(K)=A(J)
      IF(A(J)-PIV)3,3,2
    2 PIV=A(J)
      KSUB=J
      KPIV=K
    3 CONTINUE
C
C        START LOOP OVER ALL ROWS OF A
      DO 32 I=1,N
      ISUB=ISUB+I
      IM1=I-1
    4 KMI=KPIV-I
      IF(KMI)35,9,5
C
C        PERFORM PARTIAL COLUMN INTERCHANGE
    5 JI=KSUB-KMI
      IDC=JI-ISUB
      JJ=ISUB-IM1
      DO 6 K=JJ,ISUB
      KK=K+IDC
      HOLD=A(K)
      A(K)=A(KK)
    6 A(KK)=HOLD
C
C        PERFORM PARTIAL ROW INTERCHANGE
      KK=KSUB
      DO 7 K=KPIV,N
      II=KK-KMI
      HOLD=A(KK)
      A(KK)=A(II)
```

```
      A(II)=HOLD                                            DMFS 560
    7 KK=KK+K                                               DMFS 570
C                                                           DMFS 580
C        PERFORM REMAINING INTERCHANGE                      DMFS 590
      JJ=KPIV-1                                             DMFS 600
      II=ISUB                                               DMFS 610
      DO 8 K=I,JJ                                           DMFS 620
      HOLD=A(II)                                            DMFS 630
      A(II)=A(JI)                                           DMFS 640
      A(JI)=HOLD                                            DMFS 650
      II=II+K                                               DMFS 660
    8 JI=JI+1                                               DMFS 670
    9 IF(IRANK)22,10,10                                     DMFS 680
C                                                           DMFS 690
C        RECORD INTERCHANGE IN TRANSPOSITION VECTOR         DMFS 700
   10 TRAC(KPIV)=TRAC(I)                                    DMFS 710
      TRAC(I)=KPIV                                          DMFS 720
C                                                           DMFS 730
C        MODIFY CURRENT PIVOT ROW                           DMFS 740
      KK=IM1-IRANK                                          DMFS 750
      KMI=ISUB-KK                                           DMFS 760
      PIV=0.D0                                              DMFS 770
      IDC=IRANK+1                                           DMFS 780
      JI=ISUB-1                                             DMFS 790
      JK=KMI                                                DMFS 800
      JJ=ISUB-I                                             DMFS 810
      DO 19 K=I,N                                           DMFS 820
      SUM=0.D0                                              DMFS 830
C                                                           DMFS 840
C        BUILD UP SCALAR PRODUCT IF NECESSARY               DMFS 850
      IF(KK)13,13,11                                        DMFS 860
   11 DO 12 J=KMI,JI                                        DMFS 870
      SUM=SUM-A(J)*A(JK)                                    DMFS 880
   12 JK=JK+1                                               DMFS 890
   13 JJ=JJ+K                                               DMFS 900
      IF(K-I)14,14,16                                       DMFS 910
   14 SUM=A(ISUB)+SUM                                       DMFS 920
C                                                           DMFS 930
C        TEST RADICAND FOR LOSS OF SIGNIFICANCE             DMFS 940
      IF(SUM-DABS(A(ISUB)*DBLE(EPS)))20,20,15               DMFS 950
   15 A(ISUB)=DSQRT(SUM)                                    DMFS 960
      KPIV=I+1                                              DMFS 970
      GOTO 19                                               DMFS 980
   16 SUM=(A(JK)+SUM)/A(ISUB)                               DMFS 990
      A(JK)=SUM                                             DMFS1000
C                                                           DMFS1010
C        SEARCH FOR NEXT PIVOT ROW                          DMFS1020
      IF(A(JJ))19,19,17                                     DMFS1030
   17 TRAC(K)=TRAC(K)-SUM*SUM                               DMFS1040
      HOLD=TRAC(K)/A(JJ)                                    DMFS1050
      IF(PIV-HOLD)18,19,19                                  DMFS1060
   18 PIV=HOLD                                              DMFS1070
      KPIV=K                                                DMFS1080
      KSUB=JJ                                               DMFS1090
   19 JK=JJ+IDC                                             DMFS1100
      GOTO 32                                               DMFS1110
```

```
C                                                                       DMFS1120
C        CALCULATE MATRIX OF DEPENDENCIES U                             DMFS1130
   20 IF(IRANK)21,21,37                                                 DMFS1140
   21 IRANK=-1                                                          DMFS1150
      GOTO 4                                                            DMFS1160
   22 IRANK=IM1                                                         DMFS1170
      II=ISUB-IRANK                                                     DMFS1180
      JI=II                                                             DMFS1190
      DO 26 K=1,IRANK                                                   DMFS1200
      JI=JI-1                                                           DMFS1210
      JK=ISUB-1                                                         DMFS1220
      JJ=K-1                                                            DMFS1230
      DO 26 J=I,N                                                       DMFS1240
      IDC=IRANK                                                         DMFS1250
      SUM=0.D0                                                          DMFS1260
      KMI=JI                                                            DMFS1270
      KK=JK                                                             DMFS1280
      IF(JJ)25,25,23                                                    DMFS1290
   23 DO 24 L=1,JJ                                                      DMFS1300
      IDC=IDC-1                                                         DMFS1310
      SUM=SUM-A(KMI)*A(KK)                                              DMFS1320
      KMI=KMI-IDC                                                       DMFS1330
   24 KK=KK-1                                                           DMFS1340
   25 A(KK)=(SUM+A(KK))/A(KMI)                                          DMFS1350
   26 JK=JK+J                                                           DMFS1360
C                                                                       DMFS1370
C        CALCULATE I+TRANSPOSE(U)*U                                     DMFS1380
      JJ=ISUB-I                                                         DMFS1390
      PIV=0.D0                                                          DMFS1400
      KK=ISUB-1                                                         DMFS1410
      DO 31 K=I,N                                                       DMFS1420
      JJ=JJ+K                                                           DMFS1430
      IDC=0                                                             DMFS1440
      DO 28 J=K,N                                                       DMFS1450
      SUM=0.D0                                                          DMFS1460
      KMI=JJ+IDC                                                        DMFS1470
      DO 27 L=II,KK                                                     DMFS1480
      JK=L+IDC                                                          DMFS1490
   27 SUM=SUM+A(L)*A(JK)                                                DMFS1500
      A(KMI)=SUM                                                        DMFS1510
   28 IDC=IDC+J                                                         DMFS1520
      A(JJ)=A(JJ)+1.D0                                                  DMFS1530
      TRAC(K)=A(JJ)                                                     DMFS1540
C                                                                       DMFS1550
C        SEARCH NEXT DIAGONAL ELEMENT                                   DMFS1560
      IF(PIV-A(JJ))29,30,30                                             DMFS1570
   29 KPIV=K                                                            DMFS1580
      KSUB=JJ                                                           DMFS1590
      PIV=A(JJ)                                                         DMFS1600
   30 II=II+K                                                           DMFS1610
      KK=KK+K                                                           DMFS1620
   31 CONTINUE                                                          DMFS1630
      GOTO 4                                                            DMFS1640
   32 CONTINUE                                                          DMFS1650
   33 IF(IRANK)35,34,35                                                 DMFS1660
   34 IRANK=N                                                           DMFS1670
```

```
   35 RETURN                                                            DMFS1680
C                                                                       DMFS1690
C        ERROR RETURNS                                                  DMFS1700
C                                                                       DMFS1710
C        RETURN IN CASE OF ILLEGAL DIMENSION                            DMFS1720
   36 IRANK=-1                                                          DMFS1730
      RETURN                                                            DMFS1740
C                                                                       DMFS1750
C        INSTABLE FACTORIZATION OF I+TRANSPOSE(U)*U                     DMFS1760
   37 IRANK=-2                                                          DMFS1770
      RETURN                                                            DMFS1780
      END                                                               DMFS1790
```

```
      SUBROUTINE DMLSS(A,N,IRANK,TRAC,INC,RHS,IER)                      DMLS  10
C                                                                       DMLS  20
C        SUBROUTINE DMLSS CALCULATES THE LEAST SQUARES SOLUTION OF      DMLS  30
C        MINIMAL LENGTH OF A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS    DMLS  40
C        WITH SYMMETRIC POSITIVE SEMI-DEFINITE COEFFICIENT MATRIX (A)   DMLS  50
C        WHOSE RANK (IRANK) IS KNOWN AND MAY BE LESS THAN THE ORDER (N) DMLS  60
C                                                                       DMLS  70
C                                                                       DMLS  80
C        DIMENSIONED DUMMY VARIABLES                                    DMLS  90
      DIMENSION A(1),TRAC(1),RHS(1)                                     DMLS 100
      DOUBLE PRECISION SUM,A,RHS,TRAC,HOLD                              DMLS 110
C                                                                       DMLS 120
C        TEST OF SPECIFIED DIMENSIONS                                   DMLS 130
      IDEF=N-IRANK                                                      DMLS 140
      IF(N)33,33,1                                                      DMLS 150
    1 IF(IRANK)33,33,2                                                  DMLS 160
    2 IF(IDEF)33,3,3                                                    DMLS 170
C                                                                       DMLS 180
C        CALCULATE AUXILIARY VALUES                                     DMLS 190
    3 ITE=IRANK*(IRANK+1)/2                                             DMLS 200
      IX2=IRANK+1                                                       DMLS 210
      NP1=N+1                                                           DMLS 220
      IER=0                                                             DMLS 230
C                                                                       DMLS 240
C        INTERCHANGE RIGHT HAND SIDE                                    DMLS 250
      JJ=1                                                              DMLS 260
      II=1                                                              DMLS 270
    4 DO 6 I=1,N                                                        DMLS 280
      J=TRAC(II)                                                        DMLS 290
      IF(J)31,31,5                                                      DMLS 300
    5 HOLD=RHS(II)                                                      DMLS 310
      RHS(II)=RHS(J)                                                    DMLS 320
      RHS(J)=HOLD                                                       DMLS 330
    6 II=II+JJ                                                          DMLS 340
      IF(JJ)32,7,7                                                      DMLS 350
C                                                                       DMLS 360
C        PERFORM STEP 2 IF NECESSARY                                    DMLS 370
    7 ISW=1                                                             DMLS 380
      IF(INC*IDEF)8,28,8                                                DMLS 390
C                                                                       DMLS 400
C        CALCULATE X1 = X1 + U * X2                                     DMLS 410
    8 ISTA=ITE                                                          DMLS 420
      DO 10 I=1,IRANK                                                   DMLS 430
      ISTA=ISTA+1                                                       DMLS 440
      JJ=ISTA                                                           DMLS 450
      SUM=0.D0                                                          DMLS 460
      DO 9 J=IX2,N                                                      DMLS 470
      SUM=SUM+A(JJ)*RHS(J)                                              DMLS 480
    9 JJ=JJ+J                                                           DMLS 490
   10 RHS(I)=RHS(I)+SUM                                                 DMLS 500
      GOTO(11,28,11),ISW                                                DMLS 510
C                                                                       DMLS 520
C        CALCULATE X2 = TRANSPOSE(U) * X1                               DMLS 530
   11 ISTA=ITE                                                          DMLS 540
      DO 15 I=IX2,N                                                     DMLS 550
```

```
      JJ=ISTA
      SUM=0.D0
      DO 12 J=1,IRANK
      JJ=JJ+1
   12 SUM=SUM+A(JJ)*RHS(J)
      GOTO(13,13,14),ISW
   13 SUM=-SUM
   14 RHS(I)=SUM
   15 ISTA=ISTA+I
      GOTO(16,29,30),ISW
C
C        INITIALIZE STEP (4) OR STEP (8)
   16 ISTA=IX2
      IEND=N
      JJ=ITE+ISTA
C
C        DIVISION OF X1 BY TRANSPOSE OF TRIANGULAR MATRIX
   17 SUM=0.D0
      DO 20 I=ISTA,IEND
      IF(A(JJ))18,31,18
   18 RHS(I)=(RHS(I)-SUM)/A(JJ)
      IF(I-IEND)19,21,21
   19 JJ=JJ+ISTA
      SUM=0.D0
      DO 20 J=ISTA,I
      SUM=SUM+A(JJ)*RHS(J)
   20 JJ=JJ+1
C
C        DIVISION OF X1 BY TRIANGULAR MATRIX
   21 SUM=0.D0
      II=IEND
      DO 24 I=ISTA,IEND
      RHS(II)=(RHS(II)-SUM)/A(JJ)
      IF(II-ISTA)25,25,22
   22 KK=JJ-1
      SUM=0.D0
      DO 23 J=II,IEND
      SUM=SUM+A(KK)*RHS(J)
   23 KK=KK+J
      JJ=JJ-II
   24 II=II-1
   25 IF(IDEF)26,30,26
   26 GOTO(27,11,8),ISW
C
C        PERFORM STEP (5)
   27 ISW=2
      GOTO 8
C
C        PERFORM STEP (6)
   28 ISTA=1
      IEND=IRANK
      JJ=1
      ISW=2
      GOTO 17
C
C        PERFORM STEP (8)
```

```
   29 ISW=3                                                              DMLS1120
      GOTO 16                                                            DMLS1130
C                                                                        DMLS1140
C        REINTERCHANGE CALCULATED SOLUTION                               DMLS1150
   30 II=N                                                               DMLS1160
      JJ=-1                                                              DMLS1170
      GOTO 4                                                             DMLS1180
C                                                                        DMLS1190
C        ERROR RETURN IN CASE OF ZERO DIVISOR                            DMLS1200
   31 IER=1                                                              DMLS1210
   32 RETURN                                                             DMLS1220
C                                                                        DMLS1230
C        ERROR RETURN IN CASE OF ILLEGAL DIMENSION                       DMLS1240
   33 IER=-1                                                             DMLS1250
      RETURN                                                             DMLS1260
      END                                                                DMLS1270
```

```
      SUBROUTINE STRUT ( IST    , SAVE1  , SAVE2  , SAVE3  , SAVE4  ,       STRT  10
     1                   SAVE5  , SAVE6  , SAVE7  , SAVE8  , ISAVE1,        STRT  20
     2                   ISAVE2, ISAVE3, STRP   , IPOSC  , IPOST  ,         STRT  30
     3                   IRET   , SCMAX  , STMAX  , CDC    , SRULT  ,       STRT  40
     4                   SRULC  , SRT    , SRC    , PFT    , PFC    ,       STRT  50
     5                   STR    , FORCE  , AXSTIF , IFIRST, ITER   ,        STRT  60
     6                   CDT    , ISTOP2 )                                  STRT  70
      DIMENSION CDC(1), CDT(1), SRT(1), SRC(1), PFT(1), PFC(1)              STRT  80
C                                                                           STRT  90
C     THIS SUBROUTINE DETERMINES THE FORCE AND AXIAL STIFFNESS OF A         STRT 100
C     TYPICAL LANDING GEAR STRUT                                            STRT 110
C                                                                           STRT 120
C                                                                           STRT 130
C     INITIALIZE SUBROUTINE                                                 STRT 140
C                                                                           STRT 150
      ISTOP2 = 0                                                            STRT 155
      ABSTR=ABS(STR)                                                        STRT 160
      SPNGC=SAVE1                                                           STRT 170
      SPNGT=SAVE2                                                           STRT 180
      DISTC=SAVE3                                                           STRT 190
      DISTT=SAVE4                                                           STRT 200
      PFORC=SAVE5                                                           STRT 210
      PFORT=SAVE6                                                           STRT 220
      FOREVC=SAVE7                                                          STRT 230
      FOREVT=SAVE8                                                          STRT 240
      INDULC=ISAVE1                                                         STRT 250
      INDULT=ISAVE2                                                         STRT 260
      IJKC=1                                                                STRT 270
      IJKT=1                                                                STRT 280
      IPREV=ISAVE3                                                          STRT 290
C                                                                           STRT 300
C     DETERMINE STROKING DIRECTION                                          STRT 310
C                                                                           STRT 320
      IF(STR.LT.0.0) GO TO 500                                              STRT 330
      IF(STR.GT.0.0)GO TO 400                                               STRT 340
C                                                                           STRT 350
C     *    *    *    *    *                                                 STRT 360
C     ZERO STRUT STROKE                                                     STRT 370
C     *    *    *    *    *                                                 STRT 380
C                                                                           STRT 390
      FORCE=0.0                                                             STRT 400
      GO TO 2002                                                            STRT 410
C                                                                           STRT 420
C     *    *    *    *    *                                                 STRT 430
C     STRUT IN TENSION                                                      STRT 440
C     *    *    *    *    *                                                 STRT 450
C                                                                           STRT 460
  400 CONTINUE                                                              STRT 470
      IF(ABSTR.GT.STMAX.AND.ITER.EQ.1) GO TO 5001                           STRT 480
C                                                                           STRT 490
C     WAS PREVIOUS STEP ON COMPRESSION SIDE                                 STRT 500
C                                                                           STRT 510
      IF(IPREV.EQ.0)GO TO 402                                               STRT 520
      IF(IPREV.NE.-1)GO TO 402                                              STRT 530
      IF(PFORC.EQ.0.0)GO TO 402                                             STRT 540
```

```
      IPREV=+1                                                          STRT 550
      IF(IRET.EQ.1) GO TO 651                                           STRT 560
      GO TO 626                                                         STRT 570
  402 CONTINUE                                                          STRT 580
      IPREV=+1                                                          STRT 590
      IF(ABSTR.GT.DISTT)GO TO 401                                       STRT 600
      FORCE=0.0                                                         STRT 610
      GO TO 2001                                                        STRT 620
  401 CONTINUE                                                          STRT 630
      IF(STR.LT.STRP)GO TO 425                                          STRT 640
C                                                                       STRT 650
C     STRUT LOADING                                                     STRT 660
C                                                                       STRT 670
      IF(SPNGT.EQ.0.0)GO TO 405                                         STRT 680
      FORCE=SPNGT*(ABSTR-DISTT)                                         STRT 690
      IF(FORCE.LT.FOREVT)GO TO 2001                                     STRT 700
      FOREVT=0.0                                                        STRT 710
      INDULT=0                                                          STRT 720
      SPNGT=0.0                                                         STRT 730
      IF(IJKT.EQ.1)GO TO 407                                            STRT 740
      DISTT=CDT(IJKT-1)-PFT(IJKT-1)/SRT(IJKT)                           STRT 750
      GO TO 408                                                         STRT 760
  407 DISTT=0.0                                                         STRT 770
  408 CONTINUE                                                          STRT 780
      IF (FORCE.LT.PFT(IJKT))GO TO 405                                  STRT 790
      FORCE=PFT(IJKT)                                                   STRT 800
      IF(ABSTR.LT.CDT(IJKT))GO TO 2001                                  STRT 810
  405 CONTINUE                                                          STRT 820
  406 IF(ABSTR.LT.CDT(IJKT))GO TO 410                                   STRT 830
      DISTT=CDT(IJKT)-PFT(IJKT)/SRT(IJKT+1)                             STRT 840
      IJKT=IJKT+1                                                       STRT 850
      GO TO 406                                                         STRT 860
  410 CONTINUE                                                          STRT 870
      FORCE=SRT(IJKT)*(ABSTR-DISTT)                                     STRT 880
      IF(FORCE.GT.PFT(IJKT))FORCE=PFT(IJKT)                             STRT 890
      GO TO 2001                                                        STRT 900
C                                                                       STRT 910
C     STRUT UNLOADING                                                   STRT 920
C                                                                       STRT 930
  425 CONTINUE                                                          STRT 940
      IF(IRET.EQ.1)GO TO 450                                            STRT 950
      IF(INDULT.NE.0)GO TO 475                                          STRT 960
  426 SPNGT=SRULT                                                       STRT 970
      IF(PFORT.NE.PFT(IPOST))SPNGT=SRT(IPOST)                           STRT 975
      FOREVT=PFORT                                                      STRT 980
      GO TO 470                                                         STRT 990
  450 CONTINUE                                                          STRT1000
      IF(INDULT.NE.0)GO TO 475                                          STRT1010
  451 CONTINUE                                                          STRT1020
      FOREVT=PFORT                                                      STRT1030
      IF(PFORT.EQ.PFT(IPOST            )) GO TO 465                     STRT1040
      SPNGT=SRT(IPOST            )                                      STRT1050
      GO TO 470                                                         STRT1060
  465 CONTINUE                                                          STRT1070
      SPNGT=SRULT                                                       STRT1080
      IF(IPOST.EQ.5) GO TO 470                                          STRT1090
```

```
      SPNGT=SRT(IPOST+1       )                                         STRT1100
  470 CONTINUE                                                          STRT1110
      INDULT=1                                                          STRT1120
      DISTT=ABS(STRP)-PFORT/SPNGT                                       STRT1130
      IF(IPREV.EQ.-1)GO TO 602                                          STRT1140
  475 CONTINUE                                                          STRT1150
      FORCE=SPNGT*(ABSTR-DISTT)                                         STRT1160
      IF(ABSTR.LT.DISTT)FORCE=0.0                                       STRT1170
      GO TO 2001                                                        STRT1180
C                                                                       STRT1190
C     *   *   *   *   *                                                 STRT1200
C     STRUT IN COMPRESSION                                              STRT1210
C     *   *   *   *   *                                                 STRT1220
C                                                                       STRT1230
  500 CONTINUE                                                          STRT1240
      IF(ABSTR.GT.SCMAX.AND.ITER.EQ.1) GO TO 5000                       STRT1250
C                                                                       STRT1260
C     WAS PREVIOUS STEP ON TENSION SIDE                                 STRT1270
C                                                                       STRT1280
      IF(IPREV.EQ.0)GO TO 602                                           STRT1290
      IF(IPREV.NE.+1)GO TO 602                                          STRT1300
      IF(PFORT.EQ.0.0)GO TO 602                                         STRT1310
      IPREV=-1                                                          STRT1320
      IF(IRET.EQ.+1) GO TO 451                                          STRT1330
      GO TO 426                                                         STRT1340
  602 CONTINUE                                                          STRT1350
      IPREV=-1                                                          STRT1360
      IF(ABSTR.GT.DISTC)GO TO 600                                       STRT1370
      FORCE=0.0                                                         STRT1380
      GO TO 2000                                                        STRT1390
  600 CONTINUE                                                          STRT1400
      IF(ITER.NE.1.AND.PFORC.NE.PFC(IPOSC          )) GO TO 699         STRT1410
      IF(STR.GT.STRP)GO TO 625                                          STRT1420
  699 CONTINUE                                                          STRT1430
C                                                                       STRT1440
C     STRUT LOADING                                                     STRT1450
C                                                                       STRT1460
      IF(SPNGC.EQ.0.0)GO TO 605                                         STRT1470
      FORCE=SPNGC*(ABSTR-DISTC)                                         STRT1480
      IF(FORCE.LT.FOREVC)GO TO 2000                                     STRT1490
      FOREVC=0.0                                                        STRT1500
      INDULC=0                                                          STRT1510
      SPNGC=0.0                                                         STRT1520
      IF(IJKC.EQ.1)GO TO 607                                            STRT1530
      DISTC=CDC(IJKC-1)-PFC(IJKC-1)/SRC(IJKC)                           STRT1540
      GO TO 608                                                         STRT1550
  607 DISTC=0.0                                                         STRT1560
  608 CONTINUE                                                          STRT1570
      IF(FORCE.LT.PFC(IJKC))GO TO 605                                   STRT1580
      FORCE=PFC(IJKC)                                                   STRT1590
      IF(ABSTR.LT.CDC(IJKC))GO TO 2000                                  STRT1600
  605 CONTINUE                                                          STRT1610
  606 IF(ABSTR.LT.CDC(IJKC))GO TO 610                                   STRT1620
      DISTC=CDC(IJKC)-PFC(IJKC)/SRC(IJKC+1)                             STRT1630
      IJKC=IJKC+1                                                       STRT1640
      GO TO 606                                                         STRT1650
```

```
  610 CONTINUE                                                          STRT1660
      FORCE=SRC(IJKC)*(ABSTR-DISTC)                                     STRT1670
      IF(FORCE.GT.PFC(IJKC))FORCE=PFC(IJKC)                             STRT1680
      GO TO 2000                                                        STRT1690
C                                                                       STRT1700
C     STRUT UNLOADING                                                   STRT1710
C                                                                       STRT1720
  625 CONTINUE                                                          STRT1730
      IF(IRET.EQ.1)GO TO 650                                            STRT1740
      IF(INDULC.NE.0)GO TO 675                                          STRT1750
  626 SPNGC=SRULC                                                       STRT1760
      IF(PFORC.NE.PFC(IPOSC))SPNGC=SRC(IPOSC)                           STRT1765
      FOREVC=PFORC                                                      STRT1770
      GO TO 670                                                         STRT1780
  650 CONTINUE                                                          STRT1790
      IF(INDULC.NE.0)GO TO 675                                          STRT1800
  651 CONTINUE                                                          STRT1810
      FOREVC=PFORC                                                      STRT1820
      IF(PFORC.EQ.PFC(IPOSC            )) GO TO 665                     STRT1830
      SPNGC=SRC(IPOSC            )                                      STRT1840
      GO TO 670                                                         STRT1850
  665 CONTINUE                                                          STRT1860
      SPNGC=SRULC                                                       STRT1870
      IF(IPOSC.EQ.5) GO TO 670                                          STRT1880
      SPNGC=SRC(IPOSC+1       )                                         STRT1890
  670 CONTINUE                                                          STRT1900
      INDULC=1                                                          STRT1910
      DISTC=ABS(STRP)-PFORC/SPNGC                                       STRT1920
      IF(IPREV.EQ.+1)GO TO 402                                          STRT1930
  675 CONTINUE                                                          STRT1940
      FORCE=SPNGC*(ABSTR-DISTC)                                         STRT1950
      IF(ABSTR.LT.DISTC)FORCE=0.0                                       STRT1960
C                                                                       STRT1970
C     UPDATE INDICATORS                                                 STRT1980
C                                                                       STRT1990
 2000 CONTINUE                                                          STRT2000
      PFORC=FORCE                                                       STRT2010
      GO TO 2005                                                        STRT2020
 2001 CONTINUE                                                          STRT2030
      PFORT=FORCE                                                       STRT2040
      GO TO 2005                                                        STRT2050
 2002 CONTINUE                                                          STRT2060
      PFORC=0.0                                                         STRT2070
      PFORT=0.0                                                         STRT2080
 2005 CONTINUE                                                          STRT2090
      IF(ITER.NE.1) GO TO 60                                            STRT2100
      IPOSC=IJKC                                                        STRT2110
      IPOST=IJKT                                                        STRT2120
      SAVE1=SPNGC                                                       STRT2130
      SAVE2=SPNGT                                                       STRT2140
      SAVE3=DISTC                                                       STRT2150
      SAVE4=DISTT                                                       STRT2160
      SAVE5=PFORC                                                       STRT2170
      SAVE6=PFORT                                                       STRT2180
      SAVE7=FOREVC                                                      STRT2190
      SAVE8=FOREVT                                                      STRT2200
```

```
      ISAVE1=INDULC                                                     STRT2210
      ISAVE2=INDULT                                                     STRT2220
      STRP=STR                                                          STRT2230
      ISAVE3=IPREV                                                      STRT2240
   60 CONTINUE                                                          STRT2250
      IF(STR.EQ.0.)GO TO 100                                            STRT2260
      FORCE = (STR/ABSTR)*FORCE                                         STRT2270
      GO TO 101                                                         STRT2280
  100 FORCE=0.                                                          STRT2290
  101 CONTINUE                                                          STRT2300
C                                                                       STRT2302
C     CALCULATE AXIAL STIFFNESS                                         STRT2304
C                                                                       STRT2306
      IF(STR.EQ.0.) GO TO 10                                            STRT2310
      IF(STR.GT.0.) GO TO 15                                            STRT2320
      IF(FORCE.EQ.0.)GO TO 12                                           STRT2330
      IF(ABS(FORCE).EQ.PFC(IJKC)) GO TO 12                              STRT2340
      IF(SPNGC.EQ.0.)GO TO 14                                           STRT2350
      AXSTIF=SPNGC                                                      STRT2360
      GO TO 50                                                          STRT2370
   14 AXSTIF=SRC(IJKC)                                                  STRT2380
      GO TO 50                                                          STRT2390
   15 IF(FORCE.EQ.0.)GO TO 12                                           STRT2400
   11 IF(FORCE.EQ.PFT(IJKT)) GO TO 12                                   STRT2410
      IF(SPNGT.EQ.0.)GO TO 13                                           STRT2420
      AXSTIF=SPNGT                                                      STRT2430
      GO TO 50                                                          STRT2440
   13 AXSTIF=SRT(IJKT)                                                  STRT2450
      GO TO 50                                                          STRT2460
   12 AXSTIF=0.                                                         STRT2470
      GO TO 50                                                          STRT2480
   10 IF(IFIRST.LT.0) GO TO 20                                          STRT2490
      AXSTIF=SRT(1       )                                              STRT2500
      GO TO 50                                                          STRT2510
   20 AXSTIF=SRC(1       )                                              STRT2520
   50 CONTINUE                                                          STRT2530
      RETURN                                                            STRT2540
C                                                                       STRT2550
C     STRUT BOTTOMED OUT ON COMPRESSION SIDE                            STRT2560
C                                                                       STRT2570
 5000 CONTINUE                                                          STRT2580
      WRITE(6,9000)IST                                                  STRT2590
      GO TO 5002                                                        STRT2600
C                                                                       STRT2610
C     STRUT BOTTOMED OUT ON TENSION SIDE                                STRT2620
C                                                                       STRT2630
 5001 CONTINUE                                                          STRT2640
      WRITE(6,9001)IST                                                  STRT2650
 5002 CONTINUE                                                          STRT2660
      ISTOP2 = -1                                                       STRT2670
      RETURN                                                            STRT2680
C                                                                       STRT2690
C     FORMAT STATEMENTS                                                 STRT2700
C                                                                       STRT2710
 9000 FORMAT(///,10X,*STRUT*,I3,* BOTTOMED OUT ON COMPRESSION SIDE*)    STRT2720
 9001 FORMAT(///,10X,*STRUT*,I3,* BOTTOMED OUT ON TENSION SIDE*)        STRT2730
```

```
      END                                                                 STRT2740
```

```
      OVERLAY ( GEARRT,5,1)                                            OV51   10

      PROGRAM INPUT                                                    INPT   10
      DIMENSION SAV(3)                                                 INPT   20
      DIMENSION       ALPHA(8)                                         INPT   30
      COMMON / CMAIN  /                                                INPT   40
     1    LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB  ,      INPT   50
     2    NSTEP , NITER , TOL   , ENGABS, IFPATN, NFPCRL, FPCRLD(3),   INPT   60
     3    IGLDIR, NMATL , IST   , IMATL(4)      , MNO   , FPCRST(3),   INPT   70
     4    INDPL(4)      , ELASBN(4)     , ELASAX(4)     ,              INPT   80
     5    BINERT(4)     , AREA(4)       , IRET(4)       ,              INPT   90
     6    IFIRST(4)     , SCMAX(4)      , STMAX(4)      ,              INPT  100
     7    SRULC(4)   , SRULT(4),  CDC(5,4), SRC(5,4)   ,              INPT  110
     8   CDT(5,4)       , CRC(5,4)      , SRT(5,4)      ,              INPT  120
     9    PFC(5,4)      , PFT(5,4)      , NPN    , XYZPOS(3,5)         INPT  130
     9  , DC(3,3)       , PHI          , THTA          , PSI           INPT  140
     9  , CGYCDL(3),  EGYCDL(3)                                        INPT  150
      COMMON / CMAIN  /                                                INPT  160
     A    NITERS, NDISPS, STLN2 ,                                      INPT  170
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),           INPT  180
     C    STLN(4)       , STLNO(4)              , STRP(4)   ,          INPT  190
     D    SPNGC(4)      , SPNGT(4)      , DISTC(4)      ,              INPT  200
     E    DISTT(4)      , PFORC (4)     , PFORT(4)      ,              INPT  210
     F    FOREVC(4)     , FOREVT(4)     , IPOSC(4)      ,              INPT  220
     G    IPOST(4)      , INDULC(4)     , INDULT(4)     ,              INPT  230
     H    STR(4)        , AXIALF(4)     , AXSTIF(4)     ,              INPT  240
     I    IPREV(4)      , XYZDS (3)     , SFORCE(30)    ,              INPT  250
     J    CFORCE(30)    , SUMFM4(6)     ,R24XYZ(3)     ,               INPT  260
     K    TM(36)        , FMVCTL(12,4)    , FMVCTG(12,4)   ,           INPT  270
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),       INPT  280
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)        INPT  290
      DATA DEGRAD / 57.295779513 /                                     INPT  300
      DATA BLANK / 1H  /                                               INPT  310
      WRITE(6,40)                                                      INPT  320
   40 FORMAT  ( 37H GEAR ANALYSIS DATA        - CARD CODE   //         INPT  330
     1        28X31HBLANK - 0         COMMENT CARDS    /               INPT  340
     2        28X44H    1             GEAR AND LOAD CARD         /     INPT  350
     3        23X47H    2             FRICTION CARD                /   INPT  360
     4        28X51H    3             APPLIED DISPLACEMENT CARD      / INPT  370
     5        28X45H    4             STRUT MATERIAL CARDS      /      INPT  380
     6        28X49H    5             SOLUTION PARAMETER CARD       /  INPT  390
     7        28X41H    6             NODAL POINT CARDS        /       INPT  400
     8        28X38H    7             FOOTPAD CARD        /            INPT  410
     9        28X49H    8             MATERIAL PARAMETER CARDS      /  INPT  420
     A        28X44H    9             MATERIAL CRUSH CARDS      /      INPT  430
     B        28X51H   10             COMPRESSION CRUSH DISTANCE CARDS /INPT  440
     C        28X51H   11             TENSION CRUSH DISTANCE CARDS     /INPT  450
     D        28X51H   12             COMPRESSION SPRING RATE CARDS    /INPT  460
     E        28X51H   13             TENSION SPRING RATE CARDS        /INPT  470
     F        28X51H   14             COMPRESSION PLASTIC FORCE CARDS  /INPT  480
     G        28X51H   15             TENSION PLASTIC FORCE CARDS      /INPT  490
     H        28X42H   16             DATA TERMINATOR CARD      /  )   INPT  500
C                                                                      INPT  510
C         READ THE INPUT FILE AND PLACE ON UNIT 1 FOR CARD CODE        INPT  520
```

```
C                                        PROCESSING
      REWIND 1
   90 READ(5,100)(ALPHA(I),I=1,8)
  100 FORMAT ( 8A10)
      IF ( EOF, 5 ) 103, 105
  103 CONTINUE
      WRITE(6,290)
  290 FORMAT(39H1END OF JOB---END OF DATA SET ON UNIT 5   )
      STOP
  105 CONTINUE
      WRITE(6,1111)(ALPHA(I),I=1,8)
      DECODE (2, 451, ALPHA(1) ) I
  451 FORMAT (I2)
      IF ( I .EQ. 0 ) GO TO 90
 1111 FORMAT ( 1X8A10)
C          CHECK TO INSURE THAT CARD CODE IS RIGHT JUSTIFIED
      DECODE ( 2, 106, ALPHA(1) ) A, B
      IF ( B .NE. BLANK ) GO TO 107
      WRITE(1,106) B, A
  106 FORMAT ( 2A1 )
      GO TO 108
  107 CONTINUE
      WRITE(1,110) ALPHA(1)
      DECODE ( 2,280,ALPHA(1)) I
      IF ( I .GT. 15 ) GO TO 120
  280 FORMAT  ( I2 )
  110 FORMAT ( A2 )
  108 WRITE(1,100)(ALPHA(I),I=1,8)
      GO TO 90
  120 CONTINUE
      REWIND 1
C                                    END OF INITIAL READ
  150 READ(1,200) ICCODE
  200 FORMAT ( I2 )
      IF ( EOF, 1 ) 1000, 205
  205 CONTINUE
C
C          ONLY CARDS WITH CARD CODES OF 1-15 ARE ACCEPTABLE
C
      IF ( ICCODE .GT. 15 ) GO TO 1000
      IF ( ICCODE .GT. 0  ) GO TO 220
      READ(1,100)(ALPHA(I),I=1,8)
      WRITE(6,210)(ALPHA(I),I=1,8)
  210 FORMAT (28H ****ERROR**** BAD CARD CODE,12X8A10)
      GO TO 150
  220 CONTINUE
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 ),ICCODE
C          GEAR AND LOAD CARD
 1    READ ( 1,270) LGRTYP, LDIND, STLN2, PSI, THTA, PHI
      GO TO 150
C          FRICTION CARD
 2    READ(1, 240) IDSPDR, SPDISP, COEFF
      GO TO 150
C          APPLIED DISPLACEMENT CARD
 3    READ(1, 250) (XYZDSP(I), I=1,3)
      GO TO 150
```

```
C         STRUT MATERIAL CARDS                                          INPT1090
    4 READ(1, 260) IST, IMATL(IST)                                      INPT1100
      GO TO 150                                                         INPT1110
C         SOLUTION PARAMETER CARD                                       INPT1120
    5 READ(1,260) NSTEP, NITER, NMATL, IGLDIR, ENGABS, TOL              INPT1130
      GO TO 150                                                         INPT1140
C         NODAL POINT CARDS                                             INPT1150
    6 READ(1,240) NPN   , (XYZPOS(I,NPN),I=1,3)                         INPT1160
      GO TO 150                                                         INPT1170
C         FOOTPAD CARD                                                  INPT1180
    7 READ(1,270)IFPATN, NFPCRL,(FPCRLD(I),I=1,3),(FPCRST(I),I=1,3)     INPT1190
      GO TO 150                                                         INPT1200
C         MATERIAL PARAMETER CARDS                                      INPT1210
    8 READ(1,270) MNO   , INDPL(MNO),ELASBN(MNO),ELASAX(MNO),           INPT1220
     1                           BINERT(MNO),AREA  (MNO)                INPT1230
      GO TO 150                                                         INPT1240
C         MATERIAL CRUSH CARDS                                          INPT1250
    9 READ(1,230) MNO   , IRET (MNO),IFIRST(MNO), SCMAX(MNO),           INPT1260
     1                   STMAX(MNO),SRULC (MNO), SRULT(MNO)             INPT1270
      GO TO 150                                                         INPT1280
C         COMPRESSION CRUSH DISTANCE CARDS                              INPT1290
   10 READ(1,240) MNO   , (CDC(I,MNO),I=1,5)                            INPT1300
      GO TO 150                                                         INPT1310
C         TENSION CRUSH DISTANCE CARDS                                  INPT1320
   11 READ(1,240) MNO   , (CDT(I,MNO),I=1,5)                            INPT1330
      GO TO 150                                                         INPT1340
C         COMPRESSION SPRING RATE CARDS                                 INPT1350
   12 READ(1,240) MNO   , (SRC(I,MNO),I=1,5)                            INPT1360
      GO TO 150                                                         INPT1370
C         TENSION SPRING RATE CARDS                                     INPT1380
   13 READ(1,240) MNO   , (SRT(I,MNO),I=1,5)                            INPT1390
      GO TO 150                                                         INPT1400
C         COMPRESSION PLASTIC FORCE CARDS                               INPT1410
   14 READ(1,240) MNO   , (PFC(I,MNO),I=1,5)                            INPT1420
      GO TO 150                                                         INPT1430
C         TENSION PLASTIC FORCE CARDS                                   INPT1440
   15 READ(1,240) MNO   , (PFT(I,MNO),I=1,5)                            INPT1450
      GO TO 150                                                         INPT1460
 1000 CONTINUE                                                          INPT1470
C                CONVERT EULER ANGLES TO RADIANS                        INPT1480
      PHI = PHI / DEGRAD                                                INPT1490
      THTA= THTA/ DEGRAD                                                INPT1500
      PSI = PSI / DEGRAD                                                INPT1510
C         SET DISPLACEMENT STEP NUMBER                                  INPT1520
      NDISPS = 0                                                        INPT1530
      IF ( LDIND .EQ. 2 ) GO TO 410                                     INPT1540
C                                                                       INPT1550
C         SET FRICTION PLANE INDICATOR IDSPD1 AND IDSPD2                INPT1560
C                                                                       INPT1570
      IDSPD1 = 0                                                        INPT1580
      DO  400  I = 1, 3                                                 INPT1590
      IF ( I .EQ. IDSPDR ) GO TO 400                                    INPT1600
      IDSPD2 = I                                                        INPT1610
      IF ( IDSPD1 .NE. 0 ) GO TO 400                                    INPT1620
      IDSPD1 = I                                                        INPT1630
  400 CONTINUE                                                          INPT1640
```

```
C          SET NORMAL DISPLACEMENT                                      INPT1650
      XYZDSP(IDSPDR) = SPDISP                                           INPT1660
  410 CONTINUE                                                          INPT1670
C                                                                       INPT1680
C          SET APPLIED DISPLACEMENT INCREMENTS                          INPT1690
C                                                                       INPT1700
      DO  420  I = 1, 3                                                 INPT1710
  420 XYZDS(I) = XYZDSP(I) / NSTEP                                      INPT1720
C          INITIALIZE THE ENERGY SUBROUTINE                             INPT1730
      CALL ENERG1                                                       INPT1740
      WRITE(6,320)                                                      INPT1750
  320 FORMAT ( 1H1, 45X 36HINITIAL CONDITIONS FOR GEAR ANALYSIS   ///)  INPT1760
C                                                                       INPT1770
C          GET THE LANDER TO SURFACE COORDINATE TRANSFORMATION MATRIX,DC INPT1780
C                                                                       INPT1790
      CALL GEOM                                                         INPT1800
C                                                                       INPT1810
C          CONVERT NODE COORDINATES TO SURFACE COORDINATES              INPT1820
C                                                                       INPT1830
      NPOINT= 4                                                         INPT1840
      IF ( LGRTYP .EQ. 2 ) NPOINT= 5                                    INPT1850
      DO  700  I = 1, NPOINT                                            INPT1860
      DO  650  J = 1,3                                                  INPT1870
      SAV(J) = 0.0                                                      INPT1880
      DO  650  K = 1,3                                                  INPT1890
  650 SAV(J) = SAV(J) + DC(J,K)*XYZPOS(K,I)                             INPT1900
      DO  700  J = 1,3                                                  INPT1910
      IF ( ABS(SAV(J) ) .GT. 1.E-10 ) GO TO 690                         INPT1920
      XYZPOS(J,I) = .0                                                  INPT1930
      GO TO 700                                                         INPT1940
  690 XYZPOS(J,I) = SAV(J)                                              INPT1950
  700 CONTINUE                                                          INPT1960
      RETURN                                                            INPT1970
  230 FORMAT (  5X 3I5,      4E10.5 )                                   INPT1980
  240 FORMAT (  5X  I5, 10X 5E10.5 )                                    INPT1990
  250 FORMAT (  5X       15X 4E10.5 )                                   INPT2000
  260 FORMAT (  5X 4I5,  5X 2E10.5 )                                    INPT2010
  270 FORMAT (  5X 2I5,  5X 6E10.5 )                                    INPT2020
      END                                                               INPT2030
```

```
      SUBROUTINE GEOM                                                   GEOM  10
      COMMON / CMAIN  /                                                 GEOM  20
     1    LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB  ,       GEOM  30
     2    NSTEP , NITER , TOL   , ENGABS, IFPATN, NFPCRL, FPCRLD(3),    GEOM  40
     3    IGLDIR, NMATL , IST   , IMATL(4)       , MNO   , FPCRST(3),   GEOM  50
     4    INDPL(4)      , ELASBN(4)      , ELASAX(4)      ,             GEOM  60
     5    BINERT(4)     , AREA(4)        , IRET(4)        ,             GEOM  70
     6    IFIRST(4)     , SCMAX(4)       , STMAX(4)       ,             GEOM  80
     7    SRULC(4)  , SRULT(4),  CDC(5,4), SRC(5,4)  ,                  GEOM  90
     8   CDT(5,4)       , CRC(5,4)       , SRT(5,4)       ,             GEOM 100
     9    PFC(5,4)      , PFT(5,4)       , NPN    , XYZPOS(3,5)         GEOM 110
     9  , DC(3,3)       , PHI           , THTA          , PSI           GEOM 120
     9  , CGYCDL(3),  EGYCDL(3)                                         GEOM 130
      COMMON / CMAIN  /                                                 GEOM 140
     A    NITERS, NDISPS, STLN2 ,                                       GEOM 150
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),            GEOM 160
     C    STLN(4)       , STLNO(4)                , STRP(4)  ,          GEOM 170
     D    SPNGC(4)      , SPNGT(4)       , DISTC(4)       ,             GEOM 180
     E    DISTT(4)      , PFORC (4)      , PFORT(4)       ,             GEOM 190
     F    FOREVC(4)     , FOREVT(4)      , IPOSC(4)       ,             GEOM 200
     G    IPOST(4)      , INDULC(4)      , INDULT(4)      ,             GEOM 210
     H    STR(4)        , AXIALF(4)      , AXSTIF(4)      ,             GEOM 220
     I    IPREV(4)      , XYZDS (3)      , SFORCE(30)     ,             GEOM 230
     J    CFORCE(30)    , SUMFM4(6)      ,R24XYZ(3)     ,               GEOM 240
     K    TM(9,4)       , FMVCTL(12,4)    , FMVCTG(12,4)    ,           GEOM 250
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),        GEOM 260
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)         GEOM 270
C                   **********                                          GEOM 280
C         CALCULATE THE DC TRANSFORMATION MATRIX FOR CONVERTING         GEOM 290
C               LANDER COORDINATES TO SURFACE COORDINATES               GEOM 300
C                   **********                                          GEOM 310
C         CALCULATE THE TRIG. FUNCTIONS ONLY ONCE IN GEOM               GEOM 320
C                   **********                                          GEOM 330
      COSPHI = COS( PHI  )                                              GEOM 340
      SINPHI = SIN( PHI  )                                              GEOM 350
      COSTHA = COS( THTA )                                              GEOM 360
      SINTHA = SIN( THTA )                                              GEOM 370
      COSPSI = COS( PSI  )                                              GEOM 380
      SINPSI = SIN( PSI  )                                              GEOM 390
      DC(1,1) = COSTHA * COSPSI                                         GEOM 400
      A       = SINTHA * COSPSI                                         GEOM 410
      B       = COSPHI * SINPSI                                         GEOM 420
      DC(2,1) = SINPHI * A - B                                          GEOM 430
      C       = SINPHI * SINPSI                                         GEOM 440
      DC(3,1) = COSPHI * A + C                                          GEOM 450
      DC(1,2) = COSTHA * SINPSI                                         GEOM 460
      DC(2,2) = SINTHA * C + COSPHI*COSPSI                              GEOM 470
      DC(3,2) = SINTHA * B - SINPHI*COSPSI                              GEOM 480
      DC(1,3) = - SINTHA                                                GEOM 490
      DC(2,3) = SINPHI*COSTHA                                           GEOM 500
      DC(3,3) = COSPHI*COSTHA                                           GEOM 510
C                   **********                                          GEOM 520
      RETURN                                                            GEOM 530
      END                                                               GEOM 540
```

```
      OVERLAY ( GEARRT,5,2)                                             OV52  10
      PROGRAM INVTRP                                                    INVT  10
      DIMENSION TSTIFM(3,3)                                             INVT  20
      DIMENSION STIFM(6,6), CDLXYZ (6)                                  INVT  30
      COMMON / CMAIN  /                                                 INVT  40
     1    LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB   ,      INVT  50
     2    NSTEP , NITER , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),   INVT  60
     3    IGLDIR, NMATL , IST    , IMATL(4)       , MNO    , FPCRST(3),  INVT  70
     4    INDPL(4)      , ELASBN(4)      , ELASAX(4)      ,             INVT  80
     5    BINERT(4)     , AREA(4)        , IRET(4)        ,             INVT  90
     6    IFIRST(4)     , SCMAX(4)       , STMAX(4)       ,             INVT 100
     7    SRULC(4)  , SRULT(4),  CDC(5,4), SRC(5,4)  ,                  INVT 110
     8   CDT(5,4)       , CRC(5,4)       , SRT(5,4)       ,             INVT 120
     9    PFC(5,4)      , PFT(5,4)       , NPN    , XYZPOS(3,5)         INVT 130
     9  , DC(3,3)       , PHI            , THTA           , PSI         INVT 140
     9  , CGYCDL(3),  EGYCDL(3)                                         INVT 150
      COMMON / CMAIN  /                                                 INVT 160
     A    NITERS, NDISPS, STLN2 ,                                       INVT 170
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),            INVT 180
     C    STLN(4)       , STLNO(4)                      , STRP(4)  ,    INVT 190
     D    SPNGC(4)      , SPNGT(4)       , DISTC(4)       ,             INVT 200
     E    DISTT(4)      , PFORC (4)      , PFORT(4)       ,             INVT 210
     F    FOREVC(4)     , FOREVT(4)      , IPOSC(4)       ,             INVT 220
     G    IPOST(4)      , INDULC(4)      , INDULT(4)      ,             INVT 230
     H    STR(4)        , AXIALF(4)      , AXSTIF(4)      ,             INVT 240
     I    IPREV(4)      , XYZDS (3)      , SFORCE(30)     ,             INVT 250
     J    CFORCE(30)    , SUMFM4(6)      ,R24XYZ(3)      ,              INVT 260
     K    TM(36)        , FMVCTL(12,4)    , FMVCTG(12,4)    ,           INVT 270
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),        INVT 280
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)         INVT 290
      NITERS = 0                                                        INVT 300
      NELEM=3                                                           INVT 310
      NITERC = 1                                                        INVT 320
      ICOS = 10 + (IDSPDR-1) * 3                                        INVT 330
      IFIN = 0                                                          INVT 340
      NITERS    =0                                                      INVT 350
C                                                                       INVT 360
C         CALCULATE INITIAL STRUT LENGTHS AND LOCAL COORDINATE TRANS  M.INVT 370
      CALL TRANSM(XYZPOS(1,1),XYZPOS(1,4),XYZPOS(1,2),TM( 1),STLNO(1) ) INVT 380
      CALL TRANSM(XYZPOS(1,2),XYZPOS(1,4),XYZPOS(1,3),TM(10),STLNO(2) ) INVT 390
      CALL TRANSM(XYZPOS(1,3),XYZPOS(1,4),XYZPOS(1,2),TM(19),STLNO(3) ) INVT 400
C               SAVE MAIN STRUT LENGTH                                  INVT 410
      STLNS  = STLNO(2)                                                 INVT 420
      DO  110  I = 1, 3                                                 INVT 430
  110 CDLXYZ(I)  = XYZDS(I)                                             INVT 440
      DO 120 I=1,3                                                      INVT 450
  120 STLN(I)=STLNO(I)                                                  INVT 460
      CALL OUTPT1                                                       INVT 470
  130 CONTINUE                                                          INVT 480
C                                                                       INVT 490
C         UPDATE FOOTPAD JOINT COORDINATES                              INVT 500
C                                                                       INVT 510
      DO  140  I = 1, 3                                                 INVT 520
```

```
  140 XYZPOS(I,NPOINT) = XYZPOS(I,NPOINT) + CDLXYZ(I)                   INVT 530
      IF ( NITERS .LE. 1 ) GO TO 145                                    INVT 540
      RMOVE = SQRT((BASEPX-XYZPOS(1,4))**2 + (BASEPY-XYZPOS(2,4))**2    INVT 550
     1            + (BASEPZ-XYZPOS(3,4))**2 )                           INVT 560
      IF ( RMOVE .LT. RADRAG ) GO TO 145                                INVT 570
      WRITE(6,144)                                                      INVT 580
  144 FORMAT (43H1ERROR *** NO SOLUTION FOUND AT THIS LEVEL. )          INVT 590
      CALL OUTPT1                                                       INVT 600
      RETURN                                                            INVT 610
  145 CONTINUE                                                          INVT 620
C          CALCULATE CURRENT STRUT LENGTH AND GET TRANS. M.             INVT 630
      CALL TRANSM(XYZPOS(1,1),XYZPOS(1,4),XYZPOS(1,2),TM( 1),STLN (1) ) INVT 640
      CALL TRANSM(XYZPOS(1,2),XYZPOS(1,4),XYZPOS(1,3),TM(10),STLN (2) ) INVT 650
      CALL TRANSM(XYZPOS(1,3),XYZPOS(1,4),XYZPOS(1,2),TM(19),STLN (3) ) INVT 660
C                                                                       INVT 670
C          CALCULATE STRUT FORCES AND AXIAL STIFFNESSES                 INVT 680
C                                                                       INVT 690
      NITERS = NITERS + 1                                               INVT 700
      ISTOP1 = 0                                                        INVT 705
      DO  170  IST=1, NELEM                                             INVT 710
C          COMPUTE STROKE                                               INVT 720
      STR(IST) = STLN(IST) - STLNO(IST)                                 INVT 730
C          GET THE MATERIAL NO.                                         INVT 740
      MNO =  IMATL(IST)                                                 INVT 750
      IF ( INDPL(MNO) .EQ. 2 ) GO TO 160                                INVT 760
C          COMPUTE AXIAL FORCE AND STIFFNESS - ELASTIC MATERIAL IN STRUT INVT 770
      AXSTIF(IST) = AREA(MNO)*ELASAX(MNO)/STLN(IST)                     INVT 780
      AXIALF(IST) = STR(IST)*AXSTIF(IST  )                              INVT 790
      GO TO 165                                                         INVT 800
C          COMPUTE AXIAL FORCE AND STIFFNESS - PLASTIC MATERIAL IN STRUT INVT 810
  160 CONTINUE                                                          INVT 820
      CALL STRUT (   IST, SPNGC(IST)  , SPNGT(IST)  , DISTC(IST)  ,     INVT 830
     1                    DISTT(IST)  , PFORC(IST)  , PFORT(IST)  ,     INVT 840
     2                    FOREVC(IST),  FOREVT(IST),  INDULC(IST),      INVT 850
     3                    INDULT(IST),  IPREV(IST)  , STRP(IST)   ,     INVT 860
     4                    IPOSC(IST)  , IPOST(IST)  , IRET(MNO)   ,     INVT 870
     5                    SCMAX(MNO)  , STMAX(MNO)  , CDC(1,MNO)  ,     INVT 880
     6                    SRULT(MNO)  , SRULC(MNO)  , SRT(1,MNO)  ,     INVT 890
     7                    SRC(1,MNO)  , PFT(1,MNO)  , PFC(1,MNO)  ,     INVT 900
     8                    STR(IST)    , AXIALF(IST),  AXSTIF(IST),      INVT 910
     9                    IFIRST(MNO),  NITERS      , CDT(1,MNO)  ,     INVT 920
     A                    ISTOP2 )                                      INVT 925
      IF ( ISTOP2 .LT. 0 ) ISTOP1 = 1                                   INVT 930
  165 CONTINUE                                                          INVT 940
C          SET LOCAL FORCE VECTOR WITH AXIAL FORCES                     INVT 950
      FMVCTL(1,IST) = -AXIALF(IST)                                      INVT 960
      FMVCTL(4,IST) =  AXIALF(IST)                                      INVT 970
C          CONVERT LOCAL FORCE VECTOR TO GLOBAL FORCE VECTOR            INVT 980
      CALL TRALMG( FMVCTG(1,IST), FMVCTL(1,IST), TM( 9*IST-8), 2 )      INVT 990
  170 CONTINUE                                                          INVT1000
      IF ( ISTOP1 .EQ. 1 )  RETURN                                      INVT1005
C                                                                       INVT1010
C          SUM FORCES AT NODE POINT FOUR                                INVT1020
C                                                                       INVT1030
      DO  180  I = 1, 3                                                 INVT1040
      SUMFM4(I)  = 0.0                                                  INVT1050
```

```
      K = I + 3                                                         INVT1060
      DO  180  J = 1, NELEM                                             INVT1070
  180 SUMFM4(I)  = SUMFM4(I) + FMVCTG(K,J)                              INVT1080
C                                                                       INVT1090
C         IF THREE FOOTPAD JOINT DISPLACEMENTS ARE APPLIED, GO TO       INVT1100
C         OUTPUT AND ENERGY SECTION                                     INVT1105
C                                                                       INVT1110
      IF ( LDIND .EQ. 2 ) GO TO 400                                     INVT1120
C         SKIP CONVERGENCE CHECK ON FIRST PASS                          INVT1130
C         CHECK FOR CONVERGENCE                                         INVT1140
C                END OF LOOP                                            INVT1150
      IF ( IFIN .EQ. 1 ) GO TO 400                                      INVT1160
      IF (NITERS.GT. NITER ) GO TO 400                                  INVT1170
C         COMPUTE FRICTION FORCE                                        INVT1180
      FRICTF =  COEFF *  SUMFM4( IDSPDR )                               INVT1190
      ABSFIR = SQRT( SUMFM4(IDSPD1)**2 + SUMFM4(IDSPD2)**2 )            INVT1200
      IF ( NITERS .NE. NITERC ) GO TO 190                               INVT1210
      IF ( ABSFIR .LE. ABS( FRICTF ) ) GO TO 400                        INVT1220
C         SET NORMAL FOOTPAD JOINT DISPLACEMENT INCREMENT TO ZERO       INVT1230
      CDLXYZ(IDSPDR ) = 0.0                                             INVT1240
C         PICK UP THE COS OF THE ANGLE (ALF) BETWEEN THE MAIN STRUT AND INVT1250
C                                           THE NORMAL                  INVT1260
      COSALF = ABS(TM( ICOS)      )                                     INVT1270
      IF ( COSALF .LT. 1. ) GO TO 500                                   INVT1280
      WRITE(6,501)                                                      INVT1290
  501 FORMAT (  100H1 ERROR *** DATA CASE TERMINATED, ANGLE BETWEEN MAININVT1300
     1 STRUT AND THE NORMAL TO THE LANDING SURFACE IS   )               INVT1310
      WRITE(6,502)                                                      INVT1320
  502 FORMAT ( 60X 14H.EQ. 0 DEGREES   )                                INVT1330
      CALL OUTPT1                                                       INVT1340
      RETURN                                                            INVT1350
  500 CONTINUE                                                          INVT1360
      IF ( COSALF .GT. 0.) GO TO 505                                    INVT1370
      WRITE(6,501)                                                      INVT1380
      WRITE(6,503      )                                                INVT1390
  503 FORMAT ( 60X 15H.EQ. 90 DEGREES   )                               INVT1400
      CALL OUTPT1                                                       INVT1410
      RETURN                                                            INVT1420
  505 CONTINUE                                                          INVT1430
      TANALF =  SQRT( 1. - COSALF**2 ) / COSALF                         INVT1440
      RADLIM = AMAX1( XYZDS(IDSPDR) / TANALF, XYZDS(IDSPDR) * TANALF )  INVT1450
      RADLIM   =   AMIN1(RADLIM,SQRT(STLNS**2-(STLNS-XYZDS(IDSPDR))**2)) INVT1460
C         SET OUTER RADIUS LIMIT                                        INVT1470
      RADRAG = RADLIM * 1.05                                            INVT1480
      RDSTEP = RADRAG / 100.                                            INVT1490
C               DETERMINE THE CENTER OF SOLUTION CIRCLE                 INVT1500
      BASEPX = XYZPOS(1,4)                                              INVT1510
      BASEPY = XYZPOS(2,4)                                              INVT1520
      BASEPZ = XYZPOS(3,4)                                              INVT1530
  190 CONTINUE                                                          INVT1540
C         RDSTEP IS POSITIVE AND DELTF2 NEGATIVE UNTIL SOLUTION         INVT1550
C              POINT IS PASSED                                          INVT1555
      DELTF2 = ABS( FRICTF ) - ABSFIR                                   INVT1560
C           TEST FOR CONVERGENCE                                        INVT1570
      IF ( ABS(DELTF2) .LE. .0005*ABS(FRICTF) ) GO TO 400               INVT1580
      IF ( RDSTEP .LT. 0.0 ) GO TO 235                                  INVT1590
```

```
      IF ( DELTF2 .LT. 0.0 ) GO TO 230                                  INVT1600
C              CALCULATE BACK STEP TO SOLUTION                          INVT1610
      RDSTPS = RDSTEP                                                   INVT1615
      RDSTEP = - DELTF2*RDSTEP / ( ABS(DELTF1 ) + DELTF2 )              INVT1620
C         SAVE DIFFERENCE IN RDSTEP                                     INVT1622
      RDSTPS = - (RDSTEP + RDSTPS)*.5                                   INVT1624
      GO TO 230                                                         INVT1630
  235 CONTINUE                                                          INVT1640
C         ONCE SOLUTION POINT IS PASSED, CONTINUE TAKING BACK STEPS     INVT1650
C         UNTIL DELTF2 AGAIN BECOMES NEGATIVE AFTER WHICH ONE           INVT1660
C         MORE FORWARD STEP IS TAKEN                                    INVT1670
      RDSTEP = RDSTPS                                                   INVT1680
      IF ( DELTF2 .GT. 0.0 ) GO TO 230                                  INVT1690
      RDSTEP = DELTF2 * RDSTEP / ( ABS(DELTF2 ) + DELTF1 )              INVT1700
      IFIN=1                                                            INVT1715
  230 CONTINUE                                                          INVT1720
      COS1 = SUMFM4(IDSPD1) / ABSFIR                                    INVT1730
      COS2 = SUMFM4(IDSPD2) / ABSFIR                                    INVT1740
  240 CONTINUE                                                          INVT1750
C         COMPUTE NEXT SLIDING DISPLACEMENT INCREMENTS                  INVT1760
      CDLXYZ(IDSPD1) = - COS1 * RDSTEP                                  INVT1770
      CDLXYZ(IDSPD2) = - COS2 * RDSTEP                                  INVT1780
      XYZDS( IDSPD1 ) = XYZDS(IDSPD1) + CDLXYZ(IDSPD1)                  INVT1790
      XYZDS( IDSPD2 ) = XYZDS(IDSPD2) + CDLXYZ(IDSPD2)                  INVT1800
C         SAVE FRICTION FORCE UNBALANCE                                 INVT1810
      DELTF1 = DELTF2                                                   INVT1820
      GO TO 130                                                         INVT1830
  400 CONTINUE                                                          INVT1840
C                                                                       INVT1850
C         OUTPUT  AND  ENERGY SECTION                                   INVT1860
C                                                                       INVT1870
      CALL ENERGY(  SUMFM4 )                                            INVT1880
      CALL OUTPUT                                                       INVT1890
      IF ( NITERS .LE. NITER ) GO TO 405                                INVT1900
      WRITE(6,403)                                                      INVT1910
  403 FORMAT  (34H1MAXIMUM ITERATION NUMBER EXCEEDED )                  INVT1920
      RETURN                                                            INVT1930
  405 CONTINUE                                                          INVT1940
      NITERS    =0                                                      INVT1950
      IF ( NDISPS .GE. NSTEP ) RETURN                                   INVT1960
      IF ( IFPATN .EQ. 1 ) GO TO 425                                    INVT1970
      IF ( LDIND .EQ. 2 ) GO TO 130                                     INVT1980
  425 CONTINUE                                                          INVT1990
      IF ( SUMFM4(IDSPDR) .GE.  0.0 )  GO TO 428                        INVT2000
      WRITE(6,426)                                                      INVT2010
  426 FORMAT (21H1NEGATIVE NORMAL LOAD   )                              INVT2020
      RETURN                                                            INVT2030
  428 CONTINUE                                                          INVT2040
      IF ( LDIND .EQ. 2 ) GO TO 130                                     INVT2050
      NITERS = 1                                                        INVT2060
      XYZDS( IDSPD1 ) = 0.0                                             INVT2070
      XYZDS( IDSPD2 ) = 0.0                                             INVT2080
      DO 430 I = 1, 3                                                   INVT2090
  430 CDLXYZ(I) = XYZDS(I)                                              INVT2100
C              SAVE MAIN STRUT LENGTH                                   INVT2110
      STLNS  = STLN(2)                                                  INVT2120
```

```
      NITERC = 2                                                        INVT2130
      IFIN = 0                                                          INVT2140
C                                                                       INVT2150
C         SET STRUT SAVE VARIABLES FOR THIS POSITION                    INVT2160
C                                                                       INVT2170
      ISTOP1 = 0                                                        INVT2175
      DO 440 IST = 1, 3                                                 INVT2180
      MNO = IMATL(IST )                                                 INVT2190
      IF ( INDPL(MNO) .NE. 2 ) GO TO 440                                INVT2200
      CALL STRUT (  IST, SPNGC(IST) , SPNGT(IST) , DISTC(IST) ,         INVT2210
     1                   DISTT(IST) , PFORC(IST) , PFORT(IST) ,         INVT2220
     2                   FOREVC(IST), FOREVT(IST), INDULC(IST),         INVT2230
     3                   INDULT(IST), IPREV(IST) , STRP(IST)  ,         INVT2240
     4                   IPOSC(IST) , IPOST(IST) , IRET(MNO)  ,         INVT2250
     5                   SCMAX(MNO) , STMAX(MNO) , CDC(1,MNO) ,         INVT2260
     6                   SRULT(MNO) , SRULC(MNO) , SRT(1,MNO) ,         INVT2270
     7                   SRC(1,MNO) , PFT(1,MNO) , PFC(1,MNO) ,         INVT2280
     8                   STR(IST)   , AXIALF(IST), AXSTIF(IST),         INVT2290
     9                   IFIRST(MNO), NITERS     , CDT(1,MNO) ,         INVT2300
     A                   ISTOP2 )                                       INVT2305
      IF ( ISTOP2 .LT. 0 ) ISTOP1 = 1                                   INVT2310
  440 CONTINUE                                                          INVT2320
      IF ( ISTOP1 .EQ. 1 )  RETURN                                      INVT2325
      GO TO 130                                                         INVT2330
      END                                                               INVT2340
```

```
      OVERLAY ( GEARRT,5,3)

      PROGRAM CANTIL
      COMMON / CMAIN  /
     1     LGRTYP, LDIND  , INITER, NPOINT, NREFP  , NELEM  , NUMB   ,
     2     NSTEP  , NITER  , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),
     3     IGLDIR, NMATL  , IST    , IMATL(4)        , MNO    , FPCRST(3),
     4     INDPL(4)        , ELASBN(4)       , ELASAX(4)       ,
     5     BINERT(4)       , AREA(4)         , IRET(4)         ,
     6     IFIRST(4)       , SCMAX(4)        , STMAX(4)        ,
     7     SRULC(4)   , SRULT(4),  CDC(5,4), SRC(5,4)   ,
     8    CDT(5,4)         , CRC(5,4)        , SRT(5,4)        ,
     9     PFC(5,4)        , PFT(5,4)        , NPN    , XYZPOS(3,5)
     9   , DC(3,3)         , PHI             , THTA            , PSI
     9   , CGYCDL(3),  EGYCDL(3)
      COMMON / CMAIN  /
     A     NITERS, NDISPS, STLN2 ,
     B     IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF  , XYZDSP(3),
     C     STLN(4)         , STLNO(4)                , STRP(4)   ,
     D     SPNGC(4)        , SPNGT(4)        , DISTC(4)        ,
     E     DISTT(4)        , PFORC  (4)      , PFORT(4)        ,
     F     FOREVC(4)       , FOREVT(4)       , IPOSC(4)        ,
     G     IPOST(4)        , INDULC(4)       , INDULT(4)       ,
     H     STR(4)          , AXIALF(4)       , AXSTIF(4)       ,
     I     IPREV(4)        , XYZPS  (3)      , SFORCE(30)      ,
     J     CFORCE(30)      , SUMFM4(6)       ,R24XYZ(3)      ,
     K     TM(9,4)         , FMVCTL(12,4)    , FMVCTG(12,4)    ,
     L     YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),
     M     YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)
      DIMENSION TSTIFM(15,15)
      DIMENSION CDLXYZ(15)
      DIMENSION STIFM6(6,6), STIFM(12,12), SARRY(15)
      DOUBLE PRECISION CDL(12),SAVED(78),TRAC(12)
      NITERS = 0
      I1ST  = 0
      I2ND  = 0
      IFIN  = 0
      NELEM  = 4
      ICOS = 1   + (IDSPDR-1)  *  3
      IFRIST  = 0
C          CALCULATE NODE POINT FOUR COORDINATES
      TL25 = 0.0
      DO  100  I = 1, 3
      XYZPOS(I,4) =    XYZPOS(I,5) - XYZPOS(I,2)
      TL25   = TL25 + XYZPOS(I,4)**2
  100 CONTINUE
      TL25   = SQRT( TL25  )
      DO  105  I = 1, 3
      R24XYZ(I) = XYZPOS(I,1)
  105 XYZPOS(I,4) = XYZPOS(I,2) + STLN2* XYZPOS(I,4)/ TL25
C          CALCULATE INITIAL STRUT LENGTHS AND LOCAL COORDINATE
C          TRANSFORMATION MATRICES
      CALL TRANSM(XYZPOS(1,1),XYZPOS(1,4),XYZPOS(1,2),TM(1,1),STLNO(1) )
      CALL TRANSM(XYZPOS(1,2),XYZPOS(1,4),R24XYZ( 1 ),TM(1,2),STLNO(2) )
```

```
      CALL TRANSM(XYZPOS(1,3),XYZPOS(1,4),XYZPOS(1,2),TM(1,3),STLNO(3) )CANT 530
      CALL TRANSM(XYZPOS(1,5),XYZPOS(1,4),R24XYZ( 1 ),TM(1,4),STLNO(4) )CANT 540
      DO 120   I = 1, 3                                                 CANT 550
  120 CDLXYZ(I) = XYZDS(I)                                              CANT 560
      DO 125 I = 4, 15                                                  CANT 570
  125 CDLXYZ(I) = 0.0                                                   CANT 580
      DO 130   I = 1,4                                                  CANT 590
  130 STLN(I) = STLNO(I)                                                CANT 600
C           SAVE TOTAL MAIN STRUT LENGTH = STRUT 2 + STRUT 4            CANT 610
      STLNS = STLN(2) + STLN(4)                                         CANT 620
C                                                                       CANT 630
C                  PRINT INITIAL POSITIONS                              CANT 640
C                                                                       CANT 650
      CALL OUTPT1                                                       CANT 660
      GO TO 150                                                         CANT 670
 1000 CONTINUE                                                          CANT 680
C                                                                       CANT 690
C          UPDATE COORDINATES OF NODE POINTS FOUR AND FIVE              CANT 700
C                                                                       CANT 710
      DO 140   I = 1, 3                                                 CANT 720
      K = I + 3                                                         CANT 730
      XYZPOS(I,NPOINT) = XYZPOS(I,NPOINT) + CDLXYZ(I)                   CANT 740
      XYZPOS(I,  4   ) = XYZPOS(I,  4   ) + CDLXYZ(K)                   CANT 750
  140 CONTINUE                                                          CANT 760
      IF ( NITERS .LE. 1 ) GO TO 145                                    CANT 770
      IF ( I2ND .NE. 1 ) GO TO 147                                      CANT 780
      I2ND = 0                                                          CANT 790
      CDLXYZ (1) = 0.0                                                  CANT 800
      CDLXYZ (2) = 0.0                                                  CANT 810
      CDLXYZ (3) = 0.0                                                  CANT 820
      RMOVE = SQRT((BASEPX-XYZPOS(1,5))**2 + (BASEPY-XYZPOS(2,5))**2    CANT 830
     1           + (BASEPZ-XYZPOS(3,5))**2 )                            CANT 840
      IF ( RMOVE .LT. RADRAG ) GO TO 147                                CANT 850
      WRITE(6,144)                                                      CANT 860
  144 FORMAT (43H1ERROR *** NO SOLUTION FOUND AT THIS LEVEL. )          CANT 870
      CALL OUTPT1                                                       CANT 880
      RETURN                                                            CANT 890
  145 CONTINUE                                                          CANT 900
      IF ( LDIND .EQ. 2 ) GO TO 146                                     CANT 910
C              DETERMINE THE CENTER OF SOLUTION CIRCLE                  CANT 920
      BASEPX = XYZPOS(1,5)                                              CANT 930
      BASEPY = XYZPOS(2,5)                                              CANT 940
      BASEPZ = XYZPOS(3,5)                                              CANT 950
  146 CDLXYZ (1) = 0.0                                                  CANT 960
      CDLXYZ (2) = 0.0                                                  CANT 970
      CDLXYZ (3) = 0.0                                                  CANT 980
  147 CONTINUE                                                          CANT 990
C                                                                       CANT1000
C          CALCULATE BENDING MOMENTS AND SHEAR FORCES                   CANT1010
C                                                                       CANT1020
      CALL BNDLDS                                                       CANT1030
C          SET LOCAL FORCE/MOMENT VECTOR WITH BENDING LOADS             CANT1040
      FMVCTL( 2,2)=  YSHRFP(2)                                          CANT1050
      FMVCTL( 5,2)=  YSHRFQ(2)                                          CANT1060
      FMVCTL(12,2)=  ZSHRMQ(2)                                          CANT1070
      FMVCTL( 2,4) =  YSHRFP(4)                                         CANT1080
```

```
      FMVCTL( 5,4) =  YSHRFQ(4)                                         CANT1090
      FMVCTL(12,4) =  ZSHRMQ(4)                                         CANT1100
C          CALCULATE CURRENT STRUT LENGTHS AND TRANSFORMATION MATRICES  CANT1110
      CALL TRANSM(XYZPOS(1,1),XYZPOS(1,4),XYZPOS(1,2),TM(1,1),STLN(1) ) CANT1120
      CALL TRANSM(XYZPOS(1,2),XYZPOS(1,4),R24XYZ( 1 ),TM(1,2),STLN(2) ) CANT1130
      CALL TRANSM(XYZPOS(1,3),XYZPOS(1,4),XYZPOS(1,2),TM(1,3),STLN(3) ) CANT1140
      CALL TRANSM(XYZPOS(1,5),XYZPOS(1,4),R24XYZ( 1 ),TM(1,4),STLN(4) ) CANT1150
  150 CONTINUE                                                          CANT1160
C                                                                       CANT1170
C          CALCULATE STRUT FORCES AND AXIAL STIFFNESSES                 CANT1180
C                                                                       CANT1190
      NITERS = NITERS + 1                                               CANT1200
      ISTOP1 = 0                                                        CANT1205
      DO  190  IST = 1, NELEM                                           CANT1210
C               COMPUTE STROKE                                          CANT1220
      STR(IST)  = STLN(IST) - STLNO(IST)                                CANT1230
C               GET MATERIAL NO.                                        CANT1240
      MNO  = IMATL(IST)                                                 CANT1250
      IF ( INDPL(MNO) .EQ. 2 ) GO TO 160                                CANT1260
C          COMPUTE AXIAL FORCE AND STIFFNESS----ELASTIC MATERIAL IN STRUTCANT1270
      AXSTIF(IST) = AREA(MNO)*ELASAX(MNO) / STLN(IST)                   CANT1280
      AXIALF(IST) = STR(IST) * AXSTIF(IST)                              CANT1290
      GO TO 170                                                         CANT1300
C          COMPUTE AXIAL FORCE AND STIFFNESS----PLASTIC MATERIAL IN STRUTCANT1310
  160 CONTINUE                                                          CANT1320
      CALL STRUT (  IST, SPNGC(IST) , SPNGT(IST) , DISTC(IST) ,         CANT1330
     1                   DISTT(IST) , PFORC(IST) , PFORT(IST) ,         CANT1340
     2                   FOREVC(IST), FOREVT(IST), INDULC(IST),         CANT1350
     3                   INDULT(IST), IPREV(IST) , STRP(IST)  ,         CANT1360
     4                   IPOSC(IST) , IPOST(IST) , IRET(MNO)  ,         CANT1370
     5                   SCMAX(MNO) , STMAX(MNO) , CDC(1,MNO) ,         CANT1380
     6                   SRULT(MNO) , SRULC(MNO) , SRT(1,MNO) ,         CANT1390
     7                   SRC(1,MNO) , PFT(1,MNO) , PFC(1,MNO) ,         CANT1400
     8                   STR(IST)   , AXIALF(IST), AXSTIF(IST),         CANT1410
     9                   IFIRST(MNO), NITERS     , CDT(1,MNO) ,         CANT1420
     A                   ISTOP2 )                                       CANT1422
      IF ( ISTOP2 .LT. 0 ) ISTOP1 = 1                                   CANT1425
  170 CONTINUE                                                          CANT1430
C          SET LOCAL FORCE/MOMENT VECTOR WITH AXIAL FORCES              CANT1440
      FMVCTL(1,IST) = -AXIALF(IST)                                      CANT1450
      FMVCTL(4,IST) =  AXIALF(IST)                                      CANT1460
  190 CONTINUE                                                          CANT1470
      IF ( ISTOP1 .EQ. 1 ) RETURN                                       CANT1475
C                                                                       CANT1480
C          CONVERT LOCAL FORCE/MOMENT VECTORS TO GLOBAL FORCE/MOMENT    CANT1490
C          VECTORS                                                      CANT1500
C                                                                       CANT1510
      DO  210  IST = 1, NELEM                                           CANT1520
      CALL TRALMG( FMVCTG(1,IST), FMVCTL(1,IST), TM(1,IST), 4 )         CANT1530
  210 CONTINUE                                                          CANT1540
C                                                                       CANT1550
C          SUM FORCE/MOMENTS AT NODE POINT FOUR                         CANT1560
C                         AND-SET UP SOLUTION ARRAY                     CANT1570
      DO  230  I = 1, 3                                                 CANT1580
      L    = I + 9                                                      CANT1590
      K    = I + 3                                                      CANT1600
```

```
      SUMOK1=FMVCTG(K,1)+FMVCTG(K,3)                                    CANT1610
      SUMOK2=FMVCTG(K,2)+FMVCTG(K,4)                                    CANT1620
      SUMFM4(I)=SUMOK1+SUMOK2                                           CANT1630
      SUMOK1=FMVCTG(L,1)+FMVCTG(L,3)                                    CANT1640
      SUMOK2=FMVCTG(L,2)+FMVCTG(L,4)                                    CANT1650
      SUMFM4(K)=SUMOK1+SUMOK2                                           CANT1660
      SARRY (K)= SUMFM4(I)                                              CANT1670
      SARRY (L)= SUMFM4(K)                                              CANT1680
      SARRY (I)= FMVCTG(I,4)                                            CANT1690
  230 CONTINUE                                                          CANT1700
      IF ( IFRIST .EQ. 0 ) GO TO 280                                    CANT1710
C         GET MOMENTS AT NODES TWO AND FIVE FOR SOLUTION ARRAY          CANT1720
      DO  240  I = 7,9                                                  CANT1730
      SARRY  (I+6)    = FMVCTG(I,2)                                     CANT1740
      SARRY  (I)      = FMVCTG(I,4)                                     CANT1750
  240 CONTINUE                                                          CANT1760
C                    END OF LOOP                                        CANT1770
      IF ( NITERS .GT. NITER ) GO TO 400                                CANT1780
      DO  260  I = 4, 15                                                CANT1790
      CDLXYZ(I) = -SARRY(I)                                             CANT1800
  260 CONTINUE                                                          CANT1810
C         CHECK FOR CONVERGENCE                                         CANT1820
      DO 265 I = 4, 15                                                  CANT1830
      IF ( ABS( CDLXYZ(I)) .GT. TOL ) GO TO 280                         CANT1840
  265 CONTINUE                                                          CANT1850
C         IF THREE FOOTPAD JOINT DISPLACEMENTS ARE APPLIED,             CANT1860
C         GO TO OUTPUT AND ENERGY SECTION                               CANT1870
      IF ( LDIND .EQ. 2 ) GO TO 400                                     CANT1880
      IF ( IFIN .EQ. 1 ) GO TO 400                                      CANT1890
C                COMPUTE FRICTION FORCE                                 CANT1900
      FRICTF  = COEFF * FMVCTG(IDSPDR,4)                                CANT1910
      ABSFIR = SQRT( FMVCTG(IDSPD1,4)**2 + FMVCTG(IDSPD2,4)**2)         CANT1920
      IF ( I1ST .NE. 0 ) GO TO 233                                      CANT1930
      I1ST = 1                                                          CANT1940
      IF ( ABSFIR .LE. ABS( FRICTF ) ) GO TO 400                        CANT1950
C         PICK UP THE COS OF THE ANGLE (ALF) BETWEEN THE MAIN STRUT AND CANT1960
C                                              THE NORMAL               CANT1970
      COSALF = ABS(TM( ICOS, 2 ) )                                      CANT1980
      IF ( COSALF .LT. 1. ) GO TO 500                                   CANT1990
      WRITE(6,501)                                                      CANT2000
  501 FORMAT (  100H1 ERROR *** DATA CASE TERMINATED, ANGLE BETWEEN MAINCANT2010
     1 STRUT AND THE NORMAL TO THE LANDING SURFACE IS   )               CANT2020
      WRITE(6,502)                                                      CANT2030
  502 FORMAT ( 60X 14H.EQ. 0 DEGREES   )                                CANT2040
      CALL OUTPT1                                                       CANT2050
      RETURN                                                            CANT2060
  500 CONTINUE                                                          CANT2070
      IF ( COSALF .GT. 0.) GO TO 505                                    CANT2080
      WRITE(6,501)                                                      CANT2090
      WRITE(6,503      )                                                CANT2100
  503 FORMAT ( 60X 15H.EQ. 90 DEGREES   )                               CANT2110
      CALL OUTPT1                                                       CANT2120
      RETURN                                                            CANT2130
  505 CONTINUE                                                          CANT2140
      TANALF =  SQRT( 1. - COSALF**2 ) / COSALF                         CANT2150
      RADLIM = AMAX1( XYZDS(IDSPDR) / TANALF, XYZDS(IDSPDR) * TANALF )  CANT2160
```

```
      RADLIM=AMIN1(RADLIM,SQRT(STLNS**2-(STLNS-XYZDS(IDSPDR))**2))
C         SET OUTER RADIUS LIMIT
      RADRAG = RADLIM * 1.05
      RDSTEP = RADRAG / 50.
  233 CONTINUE
C         RDSTEP IS POSITIVE AND DELTF2 NEGATIVE UNTIL SOLUTION POINT
C         IS PASSED
      DELTF2 = ABS( FRICTF ) - ABSFIR
C            TEST FOR CONVERGENCE
      IF ( ABS(DELTF2) .LE. .0005*ABS(FRICTF) ) GO TO 400
      IF ( RDSTEP .LT. 0.0 ) GO TO 235
      IF ( DELTF2 .LT. 0.0 ) GO TO 237
C              CALCULATE BACK STEP TO SOLUTION
      RDSTPS = RDSTEP
      RDSTEP = - DELTF2*RDSTEP / ( ABS(DELTF1 ) + DELTF2 )
C            SAVE DIFFERENCE IN RDSTEP
      RDSTPS =-(RDSTEP + RDSTPS  ) * .5
      GO TO 237
  235 CONTINUE
C         ONCE SOLUTION POINT IS PASSED, CONTINUE TAKING BACK STEPS
C         UNTIL DELTF2 AGAIN BECOMES NEGATIVE AFTER WHICH ONE
C         MORE FORWARD STEP IS TAKEN
      RDSTEP = RDSTPS
      IF ( DELTF2 .GT. 0.0 ) GO TO 237
      RDSTEP =   DELTF2*RDSTEP / ( ABS(DELTF2 ) + DELTF1 )
      IFIN=1
  237 CONTINUE
      COS1 = FMVCTG(IDSPD1,4) / ABSFIR
      COS2 = FMVCTG(IDSPD2,4) / ABSFIR
C         COMPUTE NEXT SLIDING DISPLACEMENT INCREMENTS
      CDLXYZ(IDSPD1) = - COS1 * RDSTEP
      CDLXYZ(IDSPD2) = - COS2 * RDSTEP
C         SAVE FRICTION FORCE UNBALANCE
      DELTF1 = DELTF2
      I2ND = 1
      DO 239 I = 4, 15
  239 CDLXYZ(I) = 0.0
  280 CONTINUE
      IFRIST = 1
C        INITIALIZE     THE  SOLUTION STIFFNESS MATRIX
      DO  290  I = 13, 15
      DO 290 J = 1, 9
      TSTIFM(J,I) = 0.0
  290 TSTIFM(I,J) = 0.0
C             COMPUTE THE SM  ELEMENT  FOUR
      IST  = 4
      CALL STFMCF( STIFM  )
      CALL TRNFSM( STIFM, TM(1,4), 12 )
      DO 300 I = 4,12
      DO 300 J = 1, 12
      TSTIFM(I,J) = STIFM(I,J)
  300 CONTINUE
C             COMPUTE THE SM. FOR ELEMENT TWO
      IST  = 2
      CALL STFMCF( STIFM  )
      CALL TRNFSM( STIFM, TM(1,2), 12 )
```

```
      DO  310  I = 4, 6
      DO  310  J = 4, 6
      K = J + 6
      L = J + 3
      M = J + 9
      TSTIFM(I,J) =  STIFM(I,J) + TSTIFM(I,J)
      TSTIFM(I,K) =  STIFM(I,K) + TSTIFM(I,K)
      TSTIFM(I,M) =  STIFM(I,L)
      N = I + 6
      TSTIFM(N,J) =  STIFM(N,J) + TSTIFM(N,J)
      TSTIFM(N,K) =  STIFM(N,K) + TSTIFM(N,K)
      TSTIFM(N,M) =  STIFM(N,L)
      N = I + 9
      TSTIFM(N,J) =  STIFM(I+3,J)
      TSTIFM(N,K) =  STIFM(I+3,K)
      TSTIFM(N,M) =  STIFM(I+3,L)
  310 CONTINUE
C              COMPUTE THE SM. FOR ELEMENT ONE AND THREE
      DO  320  IST = 1, 3, 2
      CALL STFMIT ( STIFM6, AXIALF(IST), STLN(IST), AXSTIF(IST) )
      CALL TRNFSM ( STIFM6, TM(1,IST),  6 )
      DO  330  I =  4, 6
      DO  330  J =  4, 6
      TSTIFM(I,J)=  TSTIFM(I,J)+STIFM6(I,J)
  330 CONTINUE
  320 CONTINUE
C         ALTER THE SOLUTION VECTOR FOR KNOWN DISPLACEMENTS AT NODE FIVE
      DO  340  I = 4, 15
      DO  340  J = 1, 3
  340 CDLXYZ(I) = CDLXYZ(I) - CDLXYZ(J)*TSTIFM(I,J)
C         THE 12 BY 12 MATRIX STORED IN TSTIFM IN COLUMNS 4-15 AND
C         ROWS 4-15 IS THE STIFFNESS MATRIX OF INTEREST. STORE THE
C         UPPER TRIANGULAR PORTION OF THIS MATRIX IN SAVED
      JJ=0
      DO 50 I=4,15
      DO 50 J=4,I
      JJ=JJ+1
   50 SAVED(JJ)=TSTIFM(J,I)
      DO 1 I=1,12
    1 CDL(I)=CDLXYZ(I+3)
      CALL DMFSS(SAVED,12,1.E-12,IRANK,TRAC)
      IF(IRANK.LE.0) GO TO 53
      INC=100
      GO TO 55
   53 WRITE(6,54)IRANK
   54 FORMAT(*1 BAD STIFFNESS MATRIX ----RANK = *,I5)
      CALL OUTPT1
      RETURN
   55 CONTINUE
      CALL DMLSS(SAVED,12,IRANK,TRAC,INC,CDL,IER)
      DO 6 I=1,12
    6 CDLXYZ(I+3)=CDL(I)
C
C                   CORRECT FOR ROUND OFF ERROR
C
      DO  360  I = 4,6
```

```
      IF ( ABS( CDLXYZ(I) ) .LT. 1.E-11 ) CDLXYZ(I) = 0.0               CANT3290
  360 CONTINUE                                                          CANT3300
      IF ( LDIND .EQ. 2 ) GO TO 1000                                    CANT3310
      XYZDS(IDSPD1) = XYZDS(IDSPD1) + CDLXYZ(IDSPD1)                    CANT3320
      XYZDS(IDSPD2) = XYZDS(IDSPD2) + CDLXYZ(IDSPD2)                    CANT3330
      GO TO 1000                                                        CANT3340
  400 CONTINUE                                                          CANT3350
      CALL ENERGY( FMVCTG(1,4) )                                        CANT3360
      CALL OUTPUT                                                       CANT3370
      IF ( NITERS .LE. NITER ) GO TO 405                                CANT3380
      WRITE(6,403)                                                      CANT3390
  403 FORMAT  (34H1MAXIMUM ITERATION NUMBER EXCEEDED )                  CANT3400
      RETURN                                                            CANT3410
  405 CONTINUE                                                          CANT3420
      IF ( NDISPS .GE. NSTEP ) RETURN                                   CANT3430
      IF( IFPATN .NE. 1 ) GO TO 430                                     CANT3440
  427 CONTINUE                                                          CANT3450
      IF ( FMVCTG(IDSPDR,4) .GE. 0.0 ) GO TO 430                        CANT3460
      WRITE(6,428)                                                      CANT3470
  428 FORMAT (21H1NEGATIVE NORMAL LOAD   )                              CANT3480
      RETURN                                                            CANT3490
  430 CONTINUE                                                          CANT3500
      IF ( LDIND .EQ. 2 ) GO TO 432                                     CANT3510
      XYZDS(IDSPD1)  = 0.0                                              CANT3520
      XYZDS(IDSPD2)  = 0.0                                              CANT3530
  432 CONTINUE                                                          CANT3540
      DO  434  I = 1, 3                                                 CANT3550
  434 CDLXYZ(I) = XYZDS(I)                                              CANT3560
      DO 436 I = 4, 15                                                  CANT3570
  436 CDLXYZ(I) = 0.0                                                   CANT3580
      I1ST = 0                                                          CANT3590
      IFIN = 0                                                          CANT3600
      I2ND = 0                                                          CANT3610
C           SAVE TOTAL MAIN STRUT LENGTH = STRUT 2 + STRUT 4            CANT3620
      STLNS = STLN(2) + STLN(4)                                         CANT3630
      NITERS  = 1                                                       CANT3640
C                                                                       CANT3650
C          SET STRUT SAVE VARIABLES FOR THIS POSITION                   CANT3660
C                                                                       CANT3670
      ISTOP1 = 0                                                        CANT3680
      DO  440  IST = 1, 4                                               CANT3690
      MNO = IMATL(IST)                                                  CANT3700
      IF ( INDPL(MNO) .NE. 2 ) GO TO 440                                CANT3710
      CALL STRUT (  IST, SPNGC(IST)  , SPNGT(IST)  , DISTC(IST)  ,      CANT3720
     1                   DISTT(IST)  , PFORC(IST)  , PFORT(IST)  ,      CANT3730
     2                   FOREVC(IST), FOREVT(IST), INDULC(IST),         CANT3740
     3                   INDULT(IST), IPREV(IST)  , STRP(IST)   ,       CANT3750
     4                   IPOSC(IST)  , IPOST(IST)  , IRET(MNO)   ,      CANT3760
     5                   SCMAX(MNO)  , STMAX(MNO)  , CDC(1,MNO)  ,      CANT3770
     6                   SRULT(MNO)  , SRULC(MNO)  , SRT(1,MNO)  ,      CANT3780
     7                   SRC(1,MNO)  , PFT(1,MNO)  , PFC(1,MNO)  ,      CANT3790
     8                   STR(IST)    , AXIALF(IST), AXSTIF(IST),        CANT3800
     9                   IFIRST(MNO), NITERS      , CDT(1,MNO)  ,       CANT3810
     A                   ISTOP2 )                                       CANT3815
      IF ( ISTOP2 .LT. 0 ) ISTOP1 = 1                                   CANT3820
  440 CONTINUE                                                          CANT3830
```

```
      IF ( ISTOP1 .EQ. 1 )  RETURN                                      CANT3840
      GO TO 280                                                         CANT3850
      END                                                               CANT3860
```

```
      SUBROUTINE BNDLDS                                                 BNDL  10
      COMMON / CMAIN  /                                                 BNDL  20
     1    LGRTYP, LDIND  , INITER, NPOINT, NREFP , NELEM , NUMB   ,     BNDL  30
     2    NSTEP , NITER  , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),  BNDL  40
     3    IGLDIR, NMATL  , IST    , IMATL(4)      , MNO   , FPCRST(3),  BNDL  50
     4    INDPL(4)       , ELASBN(4)      , ELASAX(4)      ,             BNDL  60
     5    BINERT(4)      , AREA(4)        , IRET(4)        ,             BNDL  70
     6    IFIRST(4)      , SCMAX(4)       , STMAX(4)       ,             BNDL  80
     7    SRULC(4)   , SRULT(4),   CDC(5,4), SRC(5,4)  ,                  BNDL  90
     8   CDT(5,4)        , CRC(5,4)       , SRT(5,4)       ,             BNDL 100
     9    PFC(5,4)       , PFT(5,4)       , NPN    , XYZPOS(3,5)         BNDL 110
     9  , DC(3,3)        , PHI           , THTA          , PSI           BNDL 120
     9  , CGYCDL(3),   EGYCDL(3)                                         BNDL 130
      COMMON / CMAIN  /                                                 BNDL 140
     A    NITERS, NDISPS, STLN2 ,                                       BNDL 150
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),            BNDL 160
     C    STLN(4)        , STLNO(4)                , STRP(4)   ,        BNDL 170
     D    SPNGC(4)       , SPNGT(4)       , DISTC(4)       ,             BNDL 180
     E    DISTT(4)       , PFORC (4)      , PFORT(4)       ,             BNDL 190
     F    FOREVC(4)      , FOREVT(4)      , IPOSC(4)       ,             BNDL 200
     G    IPOST(4)       , INDULC(4)      , INDULT(4)      ,             BNDL 210
     H    STR(4)         , AXIALF(4)      , AXSTIF(4)      ,             BNDL 220
     I    IPREV(4)       , XYZDS (3)      , SFORCE(30)     ,             BNDL 230
     J    CFORCE(30)     , SUMFM4(6)      ,R24XYZ(3)     ,               BNDL 240
     K    TM(36)         , FMVCTL(12,4)    , FMVCTG(12,4)    ,           BNDL 250
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),        BNDL 260
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)         BNDL 270
      DOUBLE PRECISION A, B, C, D, E, F, AL24, AL45, AL25, AL, DELTA,   BNDL 280
     1                 SINALF, COSALF, SINBET, COSBET, EB2, EB4, AI2,   BNDL 290
     2                 AI4, TERM, TERM1, V2, V4, BM, RM1, RM2, RM3      BNDL 300
      A=XYZPOS(1,5)-XYZPOS(1,2)                                         BNDL 310
      B=XYZPOS(2,5)-XYZPOS(2,2)                                         BNDL 320
      C=XYZPOS(3,5)-XYZPOS(3,2)                                         BNDL 330
      D=XYZPOS(1,5)-XYZPOS(1,4)                                         BNDL 340
      E=XYZPOS(2,5)-XYZPOS(2,4)                                         BNDL 350
      F=XYZPOS(3,5)-XYZPOS(3,4)                                         BNDL 360
      AL24 = DSQRT((A-D)**2 + (B-E)**2 + (C-F)**2)                      BNDL 370
      AL45 = DSQRT( D*D + E*E + F*F )                                   BNDL 380
      AL25 = DSQRT( A*A + B*B + C*C )                                   BNDL 390
      AL  = AL25                                                        BNDL 400
      DELTA = DSQRT((B*F-C*E)**2 + (C*D-A*F)**2 + (A*E-B*D)**2)/AL25    BNDL 410
      SINALF = DELTA / AL24                                             BNDL 420
      COSALF = DSQRT( 1. - SINALF*SINALF )                              BNDL 430
      SINBET = DELTA / AL45                                             BNDL 440
      COSBET = DSQRT( 1. - SINBET * SINBET )                            BNDL 450
      AL24 = AL24 * COSALF                                              BNDL 460
      AL45 = AL45 * COSBET                                              BNDL 470
      IM2=IMATL(2)                                                      BNDL 480
      IM4=IMATL(4)                                                      BNDL 490
      EB2=ELASBN(IM2)                                                   BNDL 500
      EB4=ELASBN(IM4)                                                   BNDL 510
      AI2=BINERT(IM2)                                                   BNDL 520
      AI4=BINERT(IM4)                                                   BNDL 530
      TERM=(AL24**3*AL45/(2.*AL)-AL24**4*AL45/(3.*AL*AL))/(EB2*AI2)+    BNDL 540
     1     (AL24**2*AL/3. - AL24**3 + AL24**4/AL - AL24**5/(3.*AL*AL))/ BNDL 550
```

```
     2      (EB4*AI4)
      TERM1=DELTA/(TERM*AL)
      V2=AL45*TERM1
      V4=AL24*TERM1
      BM=AL24*AL45*TERM1
      YSHRFP(2) = V2
      YSHRFQ(2) =-V2
      YSHRFP(4) = V4
      YSHRFQ(4) =-V4
      ZSHRMQ(2) = BM
      ZSHRMQ(4) = BM
      RM1 = XYZPOS(1,4) + (C*C*D - A*C*F - A*B*E + B*B*D)/(AL25**2)
      RM2 = XYZPOS(2,4) + (A*A*E - A*B*D - B*C*F + C*C*E)/(AL25**2)
      RM3 = XYZPOS(3,4) + (B*B*F - B*C*E - A*C*D + A*A*F)/(AL25**2)
      R24XYZ(1) = RM1
      R24XYZ(2) = RM2
      R24XYZ(3) = RM3
      RETURN
      END
```

```
      SUBROUTINE STFMCF( STIFML )
      COMMON / CMAIN  /
     1    LGRTYP, LDIND , INITER, NPOINT, NREFP , NELEM , NUMB   ,
     2    NSTEP , NITER , TOL    , ENGABS, IFPATN, NFPCRL, FPCRLD(3),
     3    IGLDIR, NMATL , IST    , IMATL(4)       , MNO    , FPCRST(3),
     4    INDPL(4)       , ELASBN(4)      , ELASAX(4)      ,
     5    BINERT(4)      , AREA(4)        , IRET(4)        ,
     6    IFIRST(4)      , SCMAX(4)       , STMAX(4)       ,
     7    SRULC(4)   , SRULT(4),  CDC(5,4), SRC(5,4)   ,
     8   CDT(5,4)        , CRC(5,4)       , SRT(5,4)       ,
     9    PFC(5,4)       , PFT(5,4)       , NPN    , XYZPOS(3,5)
     9  , DC(3,3)        , PHI              , THTA          , PSI
     9  , CGYCDL(3),  EGYCDL(3)
      COMMON / CMAIN  /
     A    NITERS, NDISPS, STLN2 ,
     B    IDSPDR, IDSPD1, IDSPD2, SPDISP, COEFF , XYZDSP(3),
     C    STLN(4)        , STLNO(4)                , STRP(4)   ,
     D    SPNGC(4)       , SPNGT(4)       , DISTC(4)       ,
     E    DISTT(4)       , PFORC (4)      , PFORT(4)       ,
     F    FOREVC(4)      , FOREVT(4)      , IPCSC(4)       ,
     G    IPOST(4)       , INDULC(4)      , INDULT(4)      ,
     H    STR(4)         , AXIALF(4)      , AXSTIF(4)      ,
     I    IPRLV(4)       , XYZDS (3)      , SFORCE(30)     ,
     J    CFORCE(30)     , SUMFM4(6)      ,R24XYZ(3)     ,
     K    TM(9,4)        , FMVCTL(12,4)    , FMVCTG(12,4)    ,
     L    YSHRFP(4), ZSHRFP(4), YSHRMP(4), ZSHRMP(4), CGYXYZ(3),
     M    YSHRFQ(4), ZSHRFQ(4), YSHRMQ(4), ZSHRMQ(4), EGYXYZ(3)
C
C         CALCULATE THE LOCAL STIFFNESS MATRIX FOR A BENDING STRUT
C
      DIMENSION STIFML(12,12)
      MNO = IMATL(IST)
      DO  100  I = 2, 12
      K    = I - 1
      DO  100  J = 1, K
  100 STIFML(I,J) = 0.0
      STIFML(1,1) = AXSTIF(IST)
      STIFML(2,2) = 6.*AXIALF(IST)/(5.*STLN(IST)) +
     1                   12.*ELASBN(MNO)*BINERT(MNO)/ STLN(IST)**3
      STIFML(3,3) = STIFML(2,2)
      STIFML(4,4) = STIFML(1,1)
      STIFML(5,5) = STIFML(2,2)
      STIFML(6,6) = STIFML(2,2)
      STIFML(7,7) = 0.0
      STIFML(8,8)    = 2.*AXIALF(IST)*STLN(IST)/15.+4.*ELASBN(MNO)
     1                                   *BINERT(MNO) / STLN(IST)
      STIFML(9,9)    = STIFML(8,8)
      STIFML(10,10) = 0.0
      STIFML(11,11) = STIFML(8,8)
      STIFML(12,12) = STIFML(8,8)
      STIFML(4,1) =-STIFML(1,1)
      STIFML(5,2) =-STIFML(2,2)
      STIFML(6,3) =-STIFML(2,2)
      STIFML(8,3) = -AXIALF(IST)*.1 - 6.*ELASBN(MNO)*BINERT(MNO)
     1                                    / STLN(IST)**2
```

```
      STIFML(8,6) = -STIFML(8,3)                                        STBN 560
      STIFML(9,2) = -STIFML(8,3)                                        STBN 570
      STIFML(9,5) =  STIFML(8,3)                                        STBN 580
      STIFML(11,3)=  STIFML(8,3)                                        STBN 590
      STIFML(11,6)= -STIFML(8,3)                                        STBN 600
      STIFML(12,2)= -STIFML(8,3)                                        STBN 610
      STIFML(12,5)=  STIFML(8,3)                                        STBN 620
      STIFML(11,8) = -AXIALF(IST)*STLN(IST)/30.                         STBN 630
     1             + 2.*ELASBN(MNO)*BINERT(MNO)/STLN(IST)               STBN 640
      STIFML(12,9) = STIFML(11,8)                                       STBN 650
      DO  200  I = 2, 12                                                STBN 660
      K  = I - 1                                                        STBN 670
      DO  200  J = 1, K                                                 STBN 680
  200 STIFML(J,I) = STIFML(I,J)                                         STBN 690
      RETURN                                                            STBN 700
      END                                                               STBN 710
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I

PROGRAM LISTING

LANDING LOADS AND MOTIONS PROGRAM

```
      OVERLAY ( LLMPT5, 0, 0 )

      PROGRAM LLMP    (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,TAPE4)
C
C     LANDING LOADS AND MOTIONS PROGRAM
C     MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE
C     MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST
C
C         FORTRAN I/O UNIT 3 - SECONDARY TIME HISTORY DATA PLACED ON
C                              THIS FILE (NFORC=0, NO SECONDARY TIME
C                              HISTORY OUTPUT - NFORC=1, SECONDARY
C                              HISTORY OUTPUT ON TAPE3).
C         FORTRAN I/O UNIT 4 - WORK FILE ASSOCIATED WITH MULTIPLE DATA
C                              CASES.
C         FORTRAN I/O UNIT 5 - CARD INPUT DATA.
C         FORTRAN I/O UNIT 6 - PRINTED OUTPUT.
C
      COMMON COMINT(1400)
      EQUIVALENCE ( COMINT ( 362 ), IDSETN )
      EQUIVALENCE ( COMINT (1399), NOCASE )
      EQUIVALENCE ( COMINT (1400), TYMIN   )
      COMMON / THISV / TIMHSA(275)
      DO 1000  I = 1, 1400
 1000 COMINT(I) = 0.0
      DO 2000 I = 1, 275
 2000 TIMHSA(I) = 0.0
C
C         CALL  THE INPUT AND INITIALIZATION ROUTINES
C
    1 CONTINUE
C
      CALL SECOND(TYMIN)
C
      CALL OVERLAY(6LLLMPT5,1,0,6HRECALL )
C
C         CALL THE LANDER LANDING, LOADS, AND MOTION SIMULATION
C              ROUTINES
      CALL OVERLAY(6LLLMPT5,2,0,6HRECALL )
C
      CALL SECOND(TYMOUT)
C
      RUNTYM=TYMOUT-TYMIN
      WRITE(6,3000)NOCASE,RUNTYM
      GO TO 1
C
C
C     FORMAT STATEMENT
 3000 FORMAT(//10X,8HCASE NO ,I10,9H RAN FOR ,F7.3,11H CP SECONDS)
      END
```

```
      OVERLAY ( LLMPT5, 1, 0 )                                          OVER  20

      PROGRAM  READIT                                                   READ  10
C                                                                       READ  20
C     THIS PORTION OF THE PROGRAM READS THE INPUT DATA FROM             READ  30
C     SEQUENCED DATA CARDS.                                             READ  40
C                                                                       READ  50
C     THE CARDS MUST BE PREPARED IN THE FOLLOWING FORMAT                READ  60
C                                                                       READ  70
C         COLUMN  1-4       6-9    11-20  21-30  31-40  41-50  51-60    READ  80
C                ICNTRL  CARD      DATA   DATA   DATA   DATA   DATA     READ  90
C                        SEQ.                                           READ 100
C                        RT. JUST.                                      READ 110
C                                                                       READ 120
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)                            READ 130
C                                                                       READ 140
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),                       READ 150
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),                     READ 160
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)                    READ 170
C                                                                       READ 180
      DIMENSION DUMMY(5),KONT(50),JKONT(20),NOCARD(500)                 READ 190
C                                                                       READ 200
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),     READ 210
     1    INDFPC(6),                                                    READ 220
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),   READ 230
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),   READ 240
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),   READ 250
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5), READ 260
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),   READ 270
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)                            READ 280
      EQUIVALENCE ( NFTPDS, NOLEG )                                     READ 290
      COMMON COMINT(400)                                                READ 300
      EQUIVALENCE ( COMINT(  1 ), ISAVCM )                              READ 310
      EQUIVALENCE ( COMINT(  2 ), JTEST  )                              READ 320
      EQUIVALENCE ( COMINT(  3 ), IBOTM      )                          READ 330
      EQUIVALENCE ( COMINT(  4 ), XSD    )                              READ 340
      EQUIVALENCE ( COMINT(  8 ), XSDD   )                              READ 350
      EQUIVALENCE ( COMINT( 12 ), YS     )                              READ 360
      EQUIVALENCE ( COMINT( 16 ), YSD    )                              READ 370
      EQUIVALENCE ( COMINT( 20 ), YSDD   )                              READ 380
      EQUIVALENCE ( COMINT( 24 ), ZS     )                              READ 390
      EQUIVALENCE ( COMINT( 28 ), ZSD    )                              READ 400
      EQUIVALENCE ( COMINT( 32 ), ZSDD   )                              READ 410
      EQUIVALENCE ( COMINT( 36 ), PHI    )                              READ 420
      EQUIVALENCE ( COMINT( 40 ), PHID   )                              READ 430
      EQUIVALENCE ( COMINT( 44 ), WX     )                              READ 440
      EQUIVALENCE ( COMINT( 48 ), WXD    )                              READ 450
      EQUIVALENCE ( COMINT( 52 ), THTA   )                              READ 460
      EQUIVALENCE ( COMINT( 56 ), THTAD  )                              READ 470
      EQUIVALENCE ( COMINT( 60 ), WY     )                              READ 480
      EQUIVALENCE ( COMINT( 64 ), WYD    )                              READ 490
      EQUIVALENCE ( COMINT( 68 ), PSI    )                              READ 500
      EQUIVALENCE ( COMINT( 72 ), PSID   )                              READ 510
      EQUIVALENCE ( COMINT( 76 ), WZ     )                              READ 520
```

```
      EQUIVALENCE ( COMINT( 80 ), WZD     )                              READ 530
      EQUIVALENCE ( COMINT( 84 ), GC      )                              READ 540
      EQUIVALENCE ( COMINT(104 ), GCD     )                              READ 550
      EQUIVALENCE ( COMINT(124 ), GCDD    )                              READ 560
      EQUIVALENCE ( COMINT(144 ), XFPS    )                              READ 570
      EQUIVALENCE ( COMINT(164 ), XFPSD   )                              READ 580
      EQUIVALENCE ( COMINT(184 ), XFPSDD  )                              READ 590
      EQUIVALENCE ( COMINT(204 ), YFPS    )                              READ 600
      EQUIVALENCE ( COMINT(264 ), ZFPS    )                              READ 610
      EQUIVALENCE ( COMINT(224 ), YFPSD   )                              READ 620
      EQUIVALENCE ( COMINT(244 ), YFPSDD  )                              READ 630
      EQUIVALENCE ( COMINT(284 ), ZFPSD   )                              READ 640
      EQUIVALENCE ( COMINT(304 ), ZFPSDD  )                              READ 650
      EQUIVALENCE ( COMINT( 324 ), TIME   )                              READ 660
      EQUIVALENCE ( COMINT( 325 ), HMAX   )                              READ 670
      EQUIVALENCE ( COMINT( 326 ), HMIN   )                              READ 680
      EQUIVALENCE ( COMINT( 327 ), EMIN   )                              READ 690
      EQUIVALENCE ( COMINT( 328 ), EMAX   )                              READ 700
      EQUIVALENCE ( COMINT( 329 ), XSI    )                              READ 710
      EQUIVALENCE ( COMINT ( 337 ), HZ     )                             READ 720
      EQUIVALENCE ( COMINT( 338 ), CUTERR)                               READ 730
      EQUIVALENCE ( COMINT( 339 ), IP     )                              READ 740
      EQUIVALENCE ( COMINT( 340 ), IVARH  )                              READ 750
      EQUIVALENCE ( COMINT( 341 ), IMTH   )                              READ 760
      EQUIVALENCE ( COMINT( 342 ), IPRNT  )                              READ 770
      EQUIVALENCE ( COMINT( 343 ), IFIN   )                              READ 780
      EQUIVALENCE ( COMINT( 344 ), IAD    )                              READ 790
      EQUIVALENCE ( COMINT( 352 ), IND    )                              READ 800
      EQUIVALENCE ( COMINT( 360 ), JCUT   )                              READ 810
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )                              READ 820
      EQUIVALENCE ( COMINT ( 362 ), IDSETN )                             READ 830
      EQUIVALENCE ( COMINT( 363 ), IVAL    )                             READ 840
      EQUIVALENCE ( COMINT( 364 ), XS      )                             READ 850
C                   COMINT(364-367) USED BY XS                           READ 860
      EQUIVALENCE ( COMINT (1399), NOCASE )                              READ 870
      COMMON                                                             READ 880
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY,  READ 890
     2    DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),        READ 900
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)      ,        READ 910
     4    GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)      ,        READ 920
     5    FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,                READ 930
     6    SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,                READ 940
     7    PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,                READ 950
     8    PDSY(10,5)    , PDSZ(10,5)    , FLXS    , FLYS , FLZS  ,       READ 960
     9    TLXL   ,  TLYL   , TLZL   , SLO    , XMSCB(5)      ,           READ 970
     C    YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,                READ 980
     D    YDSCB(10)     , ZDSCB(10)     , ILEG    , IMS   ,              READ 990
     E    FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ  ,             READ1000
     F    PVFPX , PVFPY , PVFPZ , NOLEG  , SLOMS(5) ,                    READ1010
     G    SLODS(10)                                                      READ1020
      COMMON                                                             READ1030
     1    PFCMS(5)      , PFCDS(5)      , PFTDS(5)      ,                READ1040
     2    PFTMS(5)      , SRCDS(5)      , SRCMS(5)      ,                READ1050
     3    SRTDS(5)      , SRTMS(5)      , COEFDS, COEFMS, GAMDS ,        READ1060
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,        READ1070
     5    STMXDS, STMXMS, CDCDS(5)      , CDCMS(5)      ,                READ1080
```

```
     6    CDTDS(5)      , CDTMS(5)      , FRICDS, FRICMS, IRETDS,           READ1090
     7    IRETMS, STROKE(10)    , STRKDS(10)    , STRKMS( 5)    ,           READ1100
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),         READ1110
     9    ATTPRS(3)     , ADIST , SOILP(3)      , NMODES,                   READ1120
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),           READ1130
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),        READ1140
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,           READ1150
     D    INDFZR, TIMAX , DRAGST, IFP                                       READ1160
C                                                                           READ1170
      COMMON                                                                READ1180
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),       READ1190
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),           READ1200
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),           READ1210
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,             READ1220
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)              READ1230
      COMMON                                                                READ1240
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),                    READ1250
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)                    READ1260
     3   ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,                    READ1270
     4    IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ              READ1280
      INTEGER STOP                                                          READ1290
      DATA NEXT/4HNEXT/,STOP/4HSTOP/,NOGO/0/                                READ1300
      IF ( ISAVCM .EQ. 0 ) GO TO 4000                                       READ1310
C                                                                           READ1320
C         INITIALIZE COMMONS BEFORE READING NEXT DATA CASE                  READ1330
C                                                                           READ1340
      REWIND 4                                                              READ1350
      READ (4) (COMINT(I),I=1,1399),(TIMHSA(I),I=1,275)                     READ1360
 4000 CONTINUE                                                              READ1370
      DO 5000 I=1,50                                                        READ1380
 5000 KONT(I)=0                                                             READ1390
      DO 5001 I=1,20                                                        READ1400
 5001 JKONT(I)=0                                                            READ1410
      MMM=0                                                                 READ1420
      WRITE(6,9000)                                                         READ1430
 9000 FORMAT(1H1,43X,                                                       READ1440
     *49HLANDING LOADS AND MOTIONS PROGRAM - LEGGED LANDER/38X,             READ1450
     *61HMASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NUMBER FIVE       READ1460
     */46X,                                                                 READ1470
     *45HMCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST  ////63X,            READ1480
     *10HINPUT DATA ////)                                                   READ1490
 9999 READ(5,9001)ICNTRL,NCARD,(DUMMY(I),I=1,5)                             READ1500
 9001 FORMAT(A4,1X,I4,1X,5E10.3)                                            READ1510
      IF(ICNTRL.EQ.NEXT) GO TO 7999                                         READ1520
      IF(ICNTRL.EQ.STOP) STOP                                               READ1530
      MMM=MMM+1                                                             READ1540
      NOCARD(MMM)=NCARD                                                     READ1550
      IF(NCARD.LE.0)GO TO 9004                                              READ1560
      IF(NCARD.GT.42)GO TO 9003                                             READ1570
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,         READ1580
     *       22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,         READ1590
     *       40,41,42                                                       READ1600
     *      ),NCARD                                                         READ1610
 9003 CONTINUE                                                              READ1620
      JJJ=NCARD/100                                                         READ1630
      IF(JJJ.LE.0.OR.JJJ.GT.17)GO TO 9004                                   READ1640
```

```
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,        READ1650
     *       1300,1400,1500,1600,1700                                   READ1660
     *      ),JJJ                                                       READ1670
 9004 CONTINUE                                                          READ1680
      WRITE(6,9002) NCARD                                               READ1690
 9002 FORMAT(24H INVALID CARD NUMBER -- ,I4)                            READ1700
      NOGO=1                                                            READ1710
      GO TO 9999                                                        READ1720
C                                                                       READ1730
C     ASSIGNMENT OF INPUT DATA                                          READ1740
C                                                                       READ1750
    1 CONTINUE                                                          READ1760
      KONT(NCARD)=KONT(NCARD)+1                                         READ1770
      NOCASE         =DUMMY(1)                                          READ1780
      GO TO 9999                                                        READ1790
C                                                                       READ1800
    2 CONTINUE                                                          READ1810
      KONT(NCARD)=KONT(NCARD)+1                                         READ1820
      TIMAX          =DUMMY(1)                                          READ1830
      IPTCNT         =DUMMY(2)                                          READ1840
      INLEG          =DUMMY(3)                                          READ1850
      IFPPRT         =DUMMY(4)                                          READ1860
      GO TO 9999                                                        READ1870
C                                                                       READ1880
    3 CONTINUE                                                          READ1890
      KONT(NCARD)=KONT(NCARD)+1                                         READ1900
      NMODES        =DUMMY(1)                                           READ1910
      NOOUT          =DUMMY(2)                                          READ1920
      GO TO 9999                                                        READ1930
C                                                                       READ1940
    4 CONTINUE                                                          READ1950
      KONT(NCARD)=KONT(NCARD)+1                                         READ1960
      INDFXD         =DUMMY(1)                                          READ1970
      INDFYD         =DUMMY(2)                                          READ1980
      INDFZD         =DUMMY(3)                                          READ1990
      GO TO 9999                                                        READ2000
C                                                                       READ2010
    5 CONTINUE                                                          READ2020
      KONT(NCARD)=KONT(NCARD)+1                                         READ2030
      INDFXR         =DUMMY(1)                                          READ2040
      INDFYR         =DUMMY(2)                                          READ2050
      INDFZR         =DUMMY(3)                                          READ2060
      GO TO 9999                                                        READ2070
C                                                                       READ2080
    6 CONTINUE                                                          READ2090
      KONT(NCARD)=KONT(NCARD)+1                                         READ2100
      HMAX           =DUMMY(1)                                          READ2110
      HMIN           =DUMMY(2)                                          READ2120
      EMAX           =DUMMY(3)                                          READ2130
      EMIN           =DUMMY(4)                                          READ2140
      IP             =DUMMY(5)                                          READ2150
      GO TO 9999                                                        READ2160
C                                                                       READ2170
    7 CONTINUE                                                          READ2180
      KONT(NCARD)=KONT(NCARD)+1                                         READ2190
      IVARH          =DUMMY(1)                                          READ2200
```

```
      IMTH            =DUMMY(2)                                         READ2210
      CUTERR          =DUMMY(3)                                         READ2220
      GO TO 9999                                                        READ2230
C                                                                       READ2240
    8 CONTINUE                                                          READ2250
      KONT(NCARD)=KONT(NCARD)+1                                         READ2260
      NFORC           =DUMMY(1)                                         READ2270
      IQUOUT          =DUMMY(2)                                         READ2280
      JCKSAB          =DUMMY(3)                                         READ2290
      IDSETN          =DUMMY(4)                                         READ2300
      GO TO 9999                                                        READ2310
C                                                                       READ2320
    9 CONTINUE                                                          READ2330
      KONT(NCARD)=KONT(NCARD)+1                                         READ2340
      ZETA            =DUMMY(1)                                         READ2350
      GRAV            =DUMMY(2)                                         READ2360
      GRAVE           =DUMMY(3)                                         READ2370
      GO TO 9999                                                        READ2380
C                                                                       READ2390
   10 CONTINUE                                                          READ2400
      KONT(NCARD)=KONT(NCARD)+1                                         READ2410
      ANGX            =DUMMY(1)                                         READ2420
      ANGY            =DUMMY(2)                                         READ2430
      ANGZ            =DUMMY(3)                                         READ2440
      GO TO 9999                                                        READ2450
C                                                                       READ2460
   11 CONTINUE                                                          READ2470
      KONT(NCARD)=KONT(NCARD)+1                                         READ2480
      WX              =DUMMY(1)                                         READ2490
      WY              =DUMMY(2)                                         READ2500
      WZ              =DUMMY(3)                                         READ2510
      GO TO 9999                                                        READ2520
C                                                                       READ2530
   12 CONTINUE                                                          READ2540
      KONT(NCARD)=KONT(NCARD)+1                                         READ2550
      VELX            =DUMMY(1)                                         READ2560
      VELY            =DUMMY(2)                                         READ2570
      VELZ            =DUMMY(3)                                         READ2580
      GO TO 9999                                                        READ2590
C                                                                       READ2600
   13 CONTINUE                                                          READ2610
      KONT(NCARD)=KONT(NCARD)+1                                         READ2620
      CBMASS          =DUMMY(1)                                         READ2630
      GO TO 9999                                                        READ2640
C                                                                       READ2650
   14 CONTINUE                                                          READ2660
      KONT(NCARD)=KONT(NCARD)+1                                         READ2670
      CBIXX           =DUMMY(1)                                         READ2680
      CBIYY           =DUMMY(2)                                         READ2690
      CBIZZ           =DUMMY(3)                                         READ2700
      GO TO 9999                                                        READ2710
C                                                                       READ2720
   15 CONTINUE                                                          READ2730
      KONT(NCARD)=KONT(NCARD)+1                                         READ2740
      CBIXY           =DUMMY(1)                                         READ2750
      CBIXZ           =DUMMY(2)                                         READ2760
```

```
      CBIYZ          =DUMMY(3)                                          READ2770
      GO TO 9999                                                        READ2780
C                                                                       READ2790
  100 CONTINUE                                                          READ2800
      JKONT(JJJ)=JKONT(JJJ)+1                                           READ2810
      II=JKONT(JJJ)                                                     READ2820
      XOUT(II)       =DUMMY(1)                                          READ2830
      YOUT(II)       =DUMMY(2)                                          READ2840
      ZOUT(II)       =DUMMY(3)                                          READ2850
      GO TO 9999                                                        READ2860
C                                                                       READ2870
   16 CONTINUE                                                          READ2880
      KONT(NCARD)=KONT(NCARD)+1                                         READ2890
      FPMASS         =DUMMY(1)                                          READ2900
      GO TO 9999                                                        READ2910
C                                                                       READ2920
   17 CONTINUE                                                          READ2930
      KONT(NCARD)=KONT(NCARD)+1                                         READ2940
      DO 7000 I=1,4                                                     READ2950
 7000 RAD(I)         =DUMMY(I)                                          READ2960
      GO TO 9999                                                        READ2970
C                                                                       READ2980
   18 CONTINUE                                                          READ2990
      KONT(NCARD)=KONT(NCARD)+1                                         READ3000
      DO 7001 I=1,4                                                     READ3010
 7001 SS(I)          =DUMMY(I)                                          READ3020
      GO TO 9999                                                        READ3030
C                                                                       READ3040
   19 CONTINUE                                                          READ3050
      KONT(NCARD)=KONT(NCARD)+1                                         READ3060
      DO 7002 I=1,3                                                     READ3070
 7002 ATTHCK(I)      =DUMMY(I)                                          READ3080
      GO TO 9999                                                        READ3090
C                                                                       READ3100
   20 CONTINUE                                                          READ3110
      KONT(NCARD)=KONT(NCARD)+1                                         READ3120
      DO 7003 I=1,3                                                     READ3130
 7003 ATTPRS(I)      =DUMMY(I)                                          READ3140
      GO TO 9999                                                        READ3150
C                                                                       READ3160
   21 CONTINUE                                                          READ3170
      KONT(NCARD)=KONT(NCARD)+1                                         READ3180
      NTYPE          =DUMMY(1)                                          READ3190
      GO TO 9999                                                        READ3200
C                                                                       READ3210
   22 CONTINUE                                                          READ3220
      KONT(NCARD)=KONT(NCARD)+1                                         READ3230
      DO 7004 I=1,3                                                     READ3240
 7004 SOILP(I)       =DUMMY(I)                                          READ3250
      GO TO 9999                                                        READ3260
C                                                                       READ3270
   23 CONTINUE                                                          READ3280
      KONT(NCARD)=KONT(NCARD)+1                                         READ3290
      NOLEG          =DUMMY(1)                                          READ3300
      ILEG           =DUMMY(2)                                          READ3310
      DRAGST         =DUMMY(3)                                          READ3320
```

```
      GO TO 9999                                                        READ3330
C                                                                       READ3340
  200 CONTINUE                                                          READ3350
      JKONT(JJJ)=JKONT(JJJ)+1                                           READ3360
      II=JKONT(JJJ)                                                     READ3370
      XFP(II)        =DUMMY(1)                                          READ3380
      YFP(II)        =DUMMY(2)                                          READ3390
      ZFP(II)        =DUMMY(3)                                          READ3400
      GO TO 9999                                                        READ3410
C                                                                       READ3420
  300 CONTINUE                                                          READ3430
      JKONT(JJJ)=JKONT(JJJ)+1                                           READ3440
      II=JKONT(JJJ)                                                     READ3450
      XMSCB(II)      =DUMMY(1)                                          READ3460
      YMSCB(II)      =DUMMY(2)                                          READ3470
      ZMSCB(II)      =DUMMY(3)                                          READ3480
      GO TO 9999                                                        READ3490
C                                                                       READ3500
   24 CONTINUE                                                          READ3510
      KONT(NCARD)=KONT(NCARD)+1                                         READ3520
      DO 7005 I=1,5                                                     READ3530
 7005 PFCMS(I)       =DUMMY(I)                                          READ3540
      GO TO 9999                                                        READ3550
C                                                                       READ3560
   25 CONTINUE                                                          READ3570
      KONT(NCARD)=KONT(NCARD)+1                                         READ3580
      DO 7006 I=1,5                                                     READ3590
 7006 PFTMS(I)       =DUMMY(I)                                          READ3600
      GO TO 9999                                                        READ3610
C                                                                       READ3620
   26 CONTINUE                                                          READ3630
      KONT(NCARD)=KONT(NCARD)+1                                         READ3640
      DO 7007 I=1,5                                                     READ3650
 7007 CDCMS(I)       =DUMMY(I)                                          READ3660
      GO TO 9999                                                        READ3670
C                                                                       READ3680
   27 CONTINUE                                                          READ3690
      KONT(NCARD)=KONT(NCARD)+1                                         READ3700
      DO 7008 I=1,5                                                     READ3710
 7008 CDTMS(I)       =DUMMY(I)                                          READ3720
      GO TO 9999                                                        READ3730
C                                                                       READ3740
   28 CONTINUE                                                          READ3750
      KONT(NCARD)=KONT(NCARD)+1                                         READ3760
      DO 7009 I=1,5                                                     READ3770
 7009 SRCMS(I)       =DUMMY(I)                                          READ3780
      GO TO 9999                                                        READ3790
C                                                                       READ3800
   29 CONTINUE                                                          READ3810
      KONT(NCARD)=KONT(NCARD)+1                                         READ3820
      DO 7010 I=1,5                                                     READ3830
 7010 SRTMS(I)       =DUMMY(I)                                          READ3840
      GO TO 9999                                                        READ3850
C                                                                       READ3860
   30 CONTINUE                                                          READ3870
      KONT(NCARD)=KONT(NCARD)+1                                         READ3880
```

```
      SCMXMS          =DUMMY(1)
      STMXMS          =DUMMY(2)
      SRUCMS          =DUMMY(3)
      SRUTMS          =DUMMY(4)
      GO TO 9999
C
   31 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      IRETMS          =DUMMY(1)
      GO TO 9999
C
   32 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      FRICMS          =DUMMY(1)
      COEFMS          =DUMMY(2)
      GAMMS           =DUMMY(3)
      AEI1            =DUMMY(4)
      AEI2            =DUMMY(5)
      GO TO 9999
C
  400 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      XDSCB(II)       =DUMMY(1)
      YDSCB(II)       =DUMMY(2)
      ZDSCB(II)       =DUMMY(3)
      GO TO 9999
C
   33 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7011 I=1,5
 7011 PFCDS(I)        =DUMMY(I)
      GO TO 9999
C
   34 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7012 I=1,5
 7012 PFTDS(I)        =DUMMY(I)
      GO TO 9999
C
   35 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7013 I=1,5
 7013 CDCDS(I)        =DUMMY(I)
      GO TO 9999
C
   36 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7014 I=1,5
 7014 CDTDS(I)        =DUMMY(I)
      GO TO 9999
C
   37 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7015 I=1,5
 7015 SRCDS(I)        =DUMMY(I)
```

```
      GO TO 9999
C
   38 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      DO 7016 I=1,5
 7016 SRTDS(I)      =DUMMY(I)
      GO TO 9999
C
   39 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      SCMXDS        =DUMMY(1)
      STMXDS        =DUMMY(2)
      SRUCDS        =DUMMY(3)
      SRUTDS        =DUMMY(4)
      GO TO 9999
C
   40 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      IRETDS        =DUMMY(1)
      GO TO 9999
C
   41 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      FRICDS        =DUMMY(1)
      COEFDS        =DUMMY(2)
      GAMDS         =DUMMY(3)
      GO TO 9999
C
   42 CONTINUE
      KONT(NCARD)=KONT(NCARD)+1
      MODEIN        =DUMMY(1)
      GO TO 9999
C
  500 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      GM(II)        =DUMMY(1)
      OMEGA(II)      =DUMMY(2)
      GO TO 9999
C
  600 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      WNX(II)       =DUMMY(1)
      WNY(II)       =DUMMY(2)
      WNZ(II)       =DUMMY(3)
      GO TO 9999
C
  700 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      PX(II)        =DUMMY(1)
      PY(II)        =DUMMY(2)
      PZ(II)        =DUMMY(3)
      GO TO 9999
C
```

```
  800 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7017 I=1,5
 7017 PMSX(II,I)     =DUMMY(I)
      GO TO 9999
C
  900 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7018 I=1,5
 7018 PMSY(II,I)     =DUMMY(I)
      GO TO 9999
C
 1000 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7019 I=1,5
 7019 PMSZ(II,I)     =DUMMY(I)
      GO TO 9999
C
 1100 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7020 I=1,5
 7020 PDSX(II,I)     =DUMMY(I)
      GO TO 9999
C
 1200 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7021 I=1,5
 7021 PDSY(II,I)     =DUMMY(I)
      GO TO 9999
C
 1300 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7022 I=1,5
 7022 PDSZ(II,I)     =DUMMY(I)
      GO TO 9999
C
 1400 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7023 I=1,5
 7023 POUTX(II,I)    =DUMMY(I)
      GO TO 9999
C
 1500 CONTINUE
      JKONT(JJJ)=JKONT(JJJ)+1
      II=JKONT(JJJ)
      DO 7024 I=1,5
 7024 POUTY(II,I)    =DUMMY(I)
      GO TO 9999
C
```

```
 1600 CONTINUE                                                          READ5570
      JKONT(JJJ)=JKONT(JJJ)+1                                           READ5580
      II=JKONT(JJJ)                                                     READ5590
      DO 7025 I=1,5                                                     READ5600
 7025 POUTZ(II,I)    =DUMMY(I)                                          READ5610
      GO TO 9999                                                        READ5620
C                                                                       READ5630
 1700 CONTINUE                                                          READ5640
      JKONT(JJJ)=JKONT(JJJ)+1                                           READ5650
      II=JKONT(JJJ)                                                     READ5660
      PCGX(II)       =DUMMY(1)                                          READ5670
      PCGY(II)       =DUMMY(2)                                          READ5680
      PCGZ(II)       =DUMMY(3)                                          READ5690
      GO TO 9999                                                        READ5700
C                                                                       READ5710
C     CHECK INPUT DATA                                                  READ5720
C                                                                       READ5730
 7999 CONTINUE                                                          READ5740
      DO 8000 I=1,42                                                    READ5750
      IF(KONT(I).EQ.1) GO TO 8000                                       READ5760
      IF(KONT(I).EQ.0.AND.JTEST.EQ.1) GO TO 8000                        READ5770
      WRITE(6,8001)I                                                    READ5780
      NOGO=1                                                            READ5790
 8000 CONTINUE                                                          READ5800
 8001 FORMAT(/45H INCORRECT NUMBER OF DATA CARDS - CARD NUMBER,I5)      READ5810
      DO 8007 I=1,17                                                    READ5820
      II=I*100                                                          READ5830
      IF(JKONT(I).EQ.0.AND.JTEST.EQ.1) GO TO 8007                       READ5840
      GO TO (8002,8003,8003,8004,8005,8005,8005,8006,8006, 8006,8012,   READ5850
     *     8012,8012,8013,8013,8013,8005),I                             READ5860
 8002 IF(JKONT(I).EQ.NOOUT) GO TO 8007                                  READ5870
      NOGO=1                                                            READ5880
      WRITE(6,8008)II                                                   READ5890
      GO TO 8007                                                        READ5900
 8003 IF(JKONT(I).EQ.NOLEG) GO TO 8007                                  READ5910
      NOGO=1                                                            READ5920
      WRITE(6,8008)II                                                   READ5930
      GO TO 8007                                                        READ5940
 8004 IF(JKONT(I).EQ.2*NOLEG) GO TO 8007                                READ5950
      NOGO=1                                                            READ5960
      WRITE(6,8008)II                                                   READ5970
      GO TO 8007                                                        READ5980
 8005 IF(JKONT(I).EQ.MODEIN) GO TO 8007                                 READ5990
      NOGO=1                                                            READ6000
      WRITE(6,8008)II                                                   READ6010
      GO TO 8007                                                        READ6020
 8006 IF(MODEIN.EQ.0.AND.JKONT(I).EQ.0) GO TO 8007                      READ6030
      IF(JKONT(I).EQ.NOLEG) GO TO 8007                                  READ6040
      NOGO=1                                                            READ6050
      WRITE(6,8008)II                                                   READ6060
      GO TO 8007                                                        READ6070
 8012 IF(MODEIN.EQ.0 .AND.JKONT(I).EQ.0) GO TO 8007                     READ6080
      IF(JKONT(I).EQ.2*NOLEG) GO TO 8007                                READ6090
      NOGO=1                                                            READ6100
      WRITE(6,8008)II                                                   READ6110
      GO TO 8007                                                        READ6120
```

```
 8013 IF(MODEIN.EQ.0.AND.JKONT(I).EQ.0) GO TO 8007                      READ6130
      IF(JKONT(I).EQ.NOOUT) GO TO 8007                                  READ6140
      NOGO=1                                                            READ6150
      WRITE(6,8008)II                                                   READ6160
 8007 CONTINUE                                                          READ6170
 8008 FORMAT(/64H DATA DECK DOES NOT AGREE WITH CONTROL INFORMATION - CAREAD6180
     *RD NUMBER,I5)                                                     READ6190
      IF(NMODES.LE.MODEIN)GO TO 8011                                    READ6200
      NOGO=1                                                            READ6210
      WRITE(6,8010)                                                     READ6220
 8010 FORMAT(/24H INCONSISTENT MODAL DATA )                             READ6230
 8011 CONTINUE                                                          READ6240
      WRITE(6,8009)(NOCARD(I),I=1,MMM)                                  READ6250
 8009 FORMAT(/27H NEW CARDS READ THIS RUN - 20(1X,I4)/(27X,20(1X,I4)))  READ6260
C                                                                       READ6270
      CALL DATAOT                                                       READ6280
C                                                                       READ6290
      IF(NOGO.EQ.1) STOP                                                READ6300
      JTEST=1                                                           READ6310
C                                                                       READ6320
      IF ( IDSETN .EQ. 0 ) GO TO 4010                                   READ6330
C                                                                       READ6340
C         IF ANOTHER DATA CASE IS INDICATED (IDSETN .NE. 0) THEN SAVE   READ6350
C            BLANK COMMON AND THISV COMMON FOR REINITIALIZATION BEFORE  READ6360
C            THE NEXT DATA CASE IS READ                                 READ6370
C                                                                       READ6380
      REWIND 4                                                          READ6390
      WRITE (4)(COMINT(I),I=1,1400),(TIMHSA(I),I=1,275)                 READ6400
      ISAVCM = 1                                                        READ6410
 4010 CONTINUE                                                          READ6420
C                                                                       READ6430
C     THIS PORTION OF THE PROGRAM INITIALIZES THE PROGRAM VARIABLES     READ6440
C                                                                       READ6450
C         DO NOT LET TIMAX .EQ. 0                                       READ6460
C                                                                       READ6470
      IF ( TIMAX .EQ. 0.0 ) TIMAX = HMAX+HMAX                           READ6480
      TIME=0.0                                                          READ6490
      ISTAB=0                                                           READ6500
      IBOTM=0                                                           READ6510
      IF(INLEG.EQ.1)CUTERR=100.*GRAVE                                   READ6520
      IF(NFORC.NE.0) REWIND 3                                           READ6530
C                                                                       READ6540
C     INITIALIZATION FOR SUBROUTINE SOIL                                READ6550
C                                                                       READ6560
      DO 62 I=1,3                                                       READ6570
   62 ATTHCK(I)=SS(1)-ATTHCK(I)                                         READ6580
      DO 63 I=1,NOLEG                                                   READ6590
      IMPACT(I)=0                                                       READ6600
      IPRTFP(I)=IPTCNT                                                  READ6610
      KOUNT(I)=0                                                        READ6620
   63 ATTH(I)=SS(1)                                                     READ6630
      ADIST=ATTHCK(3)                                                   READ6640
      RADIAN=57.295779513                                               READ6650
      IF(NTYPE.NE.0) GO TO 68                                           READ6660
      SOILP(1)=SOILP(1)/RADIAN                                          READ6670
      TANPHI=TAN(SOILP(1))                                              READ6680
```

```
      CMS=29.*EXP(1.4*SOILP(3))*TANPHI                                  READ6690
      SLRHO=SOILP(2)/GRAVE                                              READ6700
      TANPH2=TAN(SOILP(1)/2.)                                           READ6710
      SOILNU=(3.14159265/3.)*(1.+TANPH2)/(1.-TANPH2)                    READ6720
      IF(SOILP(3).GE.0.5) GO TO 67                                      READ6730
      CDCONT=(GRAV/GRAVE)*(4.+80.*SOILP(3))*TANPHI/(RAD(4)**2)          READ6740
      GO TO 68                                                          READ6750
   67 CDCONT=4.*(GRAV/GRAVE)*EXP(4.83*SOILP(3))*TANPHI/(RAD(4)**2)      READ6760
   68 CONTINUE                                                          READ6770
C                                                                       READ6780
C          INITIALIZE THE CENTER BODY MASS MATRIX                       READ6790
C                                                                       READ6800
      DO  70  I = 1, 6                                                  READ6810
      DO  70  J = 1, 6                                                  READ6820
   70 AM(I,J) = 0.0                                                     READ6830
      AM(1,1) =  CBMASS                                                 READ6840
      AM(2,2) =  CBMASS                                                 READ6850
      AM(3,3) =  CBMASS                                                 READ6860
      AM(4,4) =  CBIXX                                                  READ6870
      AM(5,5) =  CBIYY                                                  READ6880
      AM(6,6) =  CBIZZ                                                  READ6890
      AM(4,5) = -CBIXY                                                  READ6900
      AM(4,6) = -CBIXZ                                                  READ6910
      AM(5,6) = -CBIYZ                                                  READ6920
      AM(5,4) = -CBIXY                                                  READ6930
      AM(6,4) = -CBIXZ                                                  READ6940
      AM(6,5) = -CBIYZ                                                  READ6950
C                                                                       READ6960
C     INITIALIZATION FOR SUBROUTINE STRUT                               READ6970
C                                                                       READ6980
C     STRUT LENGTHS                                                     READ6990
C                                                                       READ7000
      DO 90 I=1,NOLEG                                                   READ7010
      DX=XMSCB(I)-XFP(I)                                                READ7020
      DY=YMSCB(I)-YFP(I)                                                READ7030
      DZ=ZMSCB(I)-ZFP(I)                                                READ7040
      SLOMS(I)=SQRT(DX*DX+DY*DY+DZ*DZ)                                  READ7050
      SLNGMS(I)=SLOMS(I)                                                READ7060
      DO 90 J=1,2                                                       READ7070
      NNN=2*(I-1)+J                                                     READ7080
      IF(ILEG.EQ.0) GO TO 84                                            READ7090
      CDX=DX/SLOMS(I)                                                   READ7100
      CDY=DY/SLOMS(I)                                                   READ7110
      CDZ=DZ/SLOMS(I)                                                   READ7120
      DEL=SLOMS(I)-DRAGST                                               READ7130
      DDX=XDSCB(NNN)-(XFP(I)+CDX*DEL)                                   READ7140
      DDY=YDSCB(NNN)-(YFP(I)+CDY*DEL)                                   READ7150
      DDZ=ZDSCB(NNN)-(ZFP(I)+CDZ*DEL)                                   READ7160
      GO TO 85                                                          READ7170
   84 CONTINUE                                                          READ7180
      DDX=XDSCB(NNN)-XFP(I)                                             READ7190
      DDY=YDSCB(NNN)-YFP(I)                                             READ7200
      DDZ=ZDSCB(NNN)-ZFP(I)                                             READ7210
   85 SLODS(NNN)=SQRT(DDX*DDX+DDY*DDY+DDZ*DDZ)                          READ7220
      SLNGDS(NNN)=SLODS(NNN)                                            READ7230
   90 CONTINUE                                                          READ7240
```

```
C                                                                       READ7250
C     MAIN STRUTS                                                       READ7260
C                                                                       READ7270
      DO 87 I=1,NOLEG                                                   READ7280
      STRPMS(I)=0.0                                                     READ7290
      URCMS(I)=0.0                                                      READ7300
      URTMS(I)=0.0                                                      READ7310
      SETCMS(I)=0.0                                                     READ7320
      SETTMS(I)=0.0                                                     READ7330
      PRFCMS(I)=0.0                                                     READ7340
      PRFTMS(I)=0.0                                                     READ7350
      FRVMSC(I)=0.0                                                     READ7360
      FRVMST(I)=0.0                                                     READ7370
      IPOCMS(I)=1                                                       READ7380
      IPOTMS(I)=1                                                       READ7390
      INDCMS(I)=0                                                       READ7400
      INDTMS(I)=0                                                       READ7410
      IPRMS(I)=0                                                        READ7420
   87 CONTINUE                                                          READ7430
C                                                                       READ7440
C     DRAG STRUTS                                                       READ7450
C                                                                       READ7460
      II=2*NOLEG                                                        READ7470
      DO 86 I=1,II                                                      READ7480
      STRPDS(I)=0.0                                                     READ7490
      URCDS(I)=0.0                                                      READ7500
      URTDS(I)=0.0                                                      READ7510
      SETCDS(I)=0.0                                                     READ7520
      SETTDS(I)=0.0                                                     READ7530
      PRFCDS(I)=0.0                                                     READ7540
      PRFTDS(I)=0.0                                                     READ7550
      FRVDSC(I)=0.0                                                     READ7560
      FRVDST(I)=0.0                                                     READ7570
      IPOCDS(I)=1                                                       READ7580
      IPOTDS(I)=1                                                       READ7590
      INDCDS(I)=0                                                       READ7600
      INDTDS(I)=0                                                       READ7610
      IPRDS(I)=0                                                        READ7620
   86 CONTINUE                                                          READ7630
C                                                                       READ7640
C     CONVERT ANGULAR QUANTITIES TO RADIANS                             READ7650
C                                                                       READ7660
      WX=WX/RADIAN                                                      READ7670
      WY=WY/RADIAN                                                      READ7680
      WZ=WZ/RADIAN                                                      READ7690
      ZETA=ZETA/RADIAN                                                  READ7700
      DO 88 I=1,NMODES                                                  READ7710
   88 OMEGA(I)=OMEGA(I)*2.*3.14159265                                   READ7720
C                                                                       READ7730
C     INITIALIZE ANGULAR QUANTITIES                                     READ7740
C                                                                       READ7750
      COX=COS(ANGX/RADIAN)                                              READ7760
      SIX=SIN(ANGX/RADIAN)                                              READ7770
      COY=COS(ANGY/RADIAN)                                              READ7780
      SIY=SIN(ANGY/RADIAN)                                              READ7790
      COZ=COS(ANGZ/RADIAN)                                              READ7800
```

```
      SIZ=SIN(ANGZ/RADIAN)                                              READ7810
      SIZT=SIN(ZETA)                                                    READ7820
      COZT=COS(ZETA)                                                    READ7830
      THTA=ASIN(-SIZT*COZ*COY-COZT*(SIZ*SIX-COZ*SIY*COX))               READ7840
      AAN=-SIZT*SIZ*COY+COZT*(COZ*SIX+SIZ*SIY*COX)                      READ7850
      BBD=SIZT*SIY+COZT*COY*COX                                         READ7860
      IF(ABS(BBD).LT.1.E-09)GO TO 104                                   READ7870
      PHI=ATAN2(AAN,BBD)                                                READ7880
      GO TO 101                                                         READ7890
  104 IF(AAN.GT.0.0)PHI=+3.14159265/2.                                  READ7900
      IF(AAN.LT.0.0)PHI=-3.14159265/2.                                  READ7910
  101 CONTINUE                                                          READ7920
      AAN=SIZ*COX+COZ*SIY*SIX                                           READ7930
      BBD=COZT*COZ*COY-SIZT*(SIZ*SIX-COZ*SIY*COX)                       READ7940
      IF(ABS(BBD).LT.1.E-09)GO TO 102                                   READ7950
      PSI=ATAN2(AAN,BBD)                                                READ7960
      GO TO 103                                                         READ7970
  102 IF(AAN.GT.0.0)PSI=+3.14159265/2.                                  READ7980
      IF(AAN.LT.0.0)PSI=-3.14159265/2.                                  READ7990
  103 CONTINUE                                                          READ8000
C                                                                       READ8010
C     INITIALIZE VELOCITIES                                             READ8020
C                                                                       READ8030
      ZETA=-ZETA                                                        READ8035
      GSINZT=SIN(ZETA)*GRAV                                             READ8040
      GCOSZT=COS(ZETA)*GRAV                                             READ8050
      XSD=VELX*COS(ZETA)+VELZ*SIN(ZETA)                                 READ8060
      YSD=VELY                                                          READ8070
      ZSD=-VELX*SIN(ZETA)+VELZ*COS(ZETA)                                READ8080
      RETURN                                                            READ8090
      END                                                               READ8100
```

```
      SUBROUTINE DATAOT
C
C     THIS SUBROUTINE LISTS THE INPUT DATA
C
      DIMENSION  NO(100),NUMB(25)
C
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),
     1    INDFPC(6),
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5),
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)
      EQUIVALENCE ( NFTPDS, NOLEG )
      COMMON COMINT(400)
      EQUIVALENCE ( COMINT(  4 ), XSD     )
      EQUIVALENCE ( COMINT( 16 ), YSD     )
      EQUIVALENCE ( COMINT( 28 ), ZSD     )
      EQUIVALENCE ( COMINT( 36 ), PHI     )
      EQUIVALENCE ( COMINT( 44 ), WX      )
      EQUIVALENCE ( COMINT( 52 ), THTA    )
      EQUIVALENCE ( COMINT( 60 ), WY      )
      EQUIVALENCE ( COMINT( 68 ), PSI     )
      EQUIVALENCE ( COMINT( 76 ), WZ      )
      EQUIVALENCE ( COMINT( 324 ), TIME   )
      EQUIVALENCE ( COMINT( 325 ), HMAX   )
      EQUIVALENCE ( COMINT( 326 ), HMIN   )
      EQUIVALENCE ( COMINT( 327 ), EMIN   )
      EQUIVALENCE ( COMINT( 328 ), EMAX   )
      EQUIVALENCE ( COMINT ( 337 ), HZ     )
      EQUIVALENCE ( COMINT( 338 ), CUTERR)
      EQUIVALENCE ( COMINT( 339 ), IP     )
      EQUIVALENCE ( COMINT( 340 ), IVARH )
      EQUIVALENCE ( COMINT( 341 ), IMTH   )
      EQUIVALENCE ( COMINT( 342 ), IPRNT )
      EQUIVALENCE ( COMINT( 343 ), IFIN   )
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )
      EQUIVALENCE ( COMINT ( 362 ), IDSETN )
      EQUIVALENCE ( COMINT (1399), NOCASE )
      COMMON
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY,
     2    DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)       ,
     4    GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)       ,
     5    FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,
     6    SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,
     7    PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,
     8    PDSY(10,5)    , PDSZ(10,5)    , FLXS    , FLYS , FLZS   ,
     9    TLXL   ,  TLYL  , TLZL  , SLO    , XMSCB(5)      ,
     C    YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,
     D    YDSCB(10)     , ZDSCB(10)     , ILEG    , IMS  ,
     E    FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ  ,
     F    PVFPX , PVFPY , PVFPZ , NOLEG  , SLOMS(5) ,
     G    SLODS(10)
```

```
      COMMON
     1    PFCMS(5)       , PFCDS(5)       , PFTDS(5)       ,
     2    PFTMS(5)       , SRCDS(5)       , SRCMS(5)       ,
     3    SRTDS(5)       , SRTMS(5)       , COEFDS, COEFMS, GAMDS ,
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,
     5    STMXDS, STMXMS, CDCDS(5)       , CDCMS(5)       ,
     6    CDTDS(5)       , CDTMS(5)       , FRICDS, FRICMS, IRETDS,
     7    IRETMS, STROKE(10)     , STRKDS(10)     , STRKMS( 5)     ,
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),
     9    ATTPRS(3)      , ADIST , SOILP(3)       , NMODES,
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,
     D    INDFZR, TIMAX , DRAGST, IFP
      COMMON
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)
      COMMON
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)
     3   ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,
     4    IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ
C
      DO 5000 I=1,100
 5000 NO(I)=I
      DO 5001 I=1,25
 5001 NUMB(I)=I*100
C
C     PROGRAM CONTROL DATA
C
      WRITE(6,101)NO(1),NOCASE
      WRITE(6,102)NO(2),TIMAX,IPTCNT,INLEG,IFPPRT
      WRITE(6,103)NO(3),NMODES,NOOUT
      WRITE(6,104)NO(4),INDFXD,INDFYD,INDFZD
      WRITE(6,105)NO(5),INDFXR,INDFYR,INDFZR
      WRITE(6,106)NO(6),HMAX,HMIN,EMAX,EMIN,IP
      WRITE(6,107)NO(7),IVARH,IMTH,CUTERR
      WRITE(6,108)NO(8),NFORC,IQUOUT,JCKSAB,IDSETN
      IF(NFORC.NE.0.AND.IDSETN.NE.0)WRITE(6,3000)
C
C     INITIAL CONDITIONS
C
      WRITE(6,109)NO(9),ZETA,GRAV,GRAVE
      WRITE(6,110)NO(10),ANGX,ANGY,ANGZ
      WRITE(6,111)NO(11),WX,WY,WZ
      WRITE(6,112)NO(12),VELX,VELY,VELZ
C
C     CENTER BODY DATA
C
      WRITE(6,113)NO(13),CBMASS
      WRITE(6,114)NO(14),CBIXX,CBIYY,CBIZZ
      WRITE(6,115)NO(15),CBIXY,CBIXZ,CBIYZ
      IF(NOOUT.EQ.0)GO TO 2
```

```
      DO 1 I=1,NOOUT
    1 WRITE(6,1100)NUMB(1),I,XOUT(I),YOUT(I),ZOUT(I)
      GO TO 3
    2 WRITE(6,1101)
    3 CONTINUE
C
C     FOOTPAD DATA
C
      WRITE(6,116)NO(16),FPMASS
      WRITE(6,117)NO(17),(RAD(I),I=1,4)
      WRITE(6,118)NO(18),(SS(I),I=1,4)
      WRITE(6,119)NO(19),(ATTHCK(I),I=1,3)
      WRITE(6,120)NO(20),(ATTPRS(I),I=1,3)
C
C     SOIL DATA
C
      IF(NTYPE.EQ.0)WRITE(6,121)NO(21),NTYPE
      IF(NTYPE.NE.0)WRITE(6,221)NO(21),NTYPE
      WRITE(6,122)NO(22),(SOILP(I),I=1,3)
C
C     LEG DATA
C
      WRITE(6,123)NO(23),NOLEG,ILEG,DRAGST
      DO 4 I=1,NOLEG
    4 WRITE(6,1200)NUMB(2),I,XFP(I),YFP(I),ZFP(I)
      DO 5 I=1,NOLEG
    5 WRITE(6,1300)NUMB(3),I,XMSCB(I),YMSCB(I),ZMSCB(I)
      WRITE(6,124)NO(24),(PFCMS(I),I=1,5)
      WRITE(6,125)NO(25),(PFTMS(I),I=1,5)
      WRITE(6,126)NO(26),(CDCMS(I),I=1,5)
      WRITE(6,127)NO(27),(CDTMS(I),I=1,5)
      WRITE(6,128)NO(28),(SRCMS(I),I=1,5)
      WRITE(6,129)NO(29),(SRTMS(I),I=1,5)
      WRITE(6,130)NO(30),SCMXMS,STMXMS,SRUCMS,SRUTMS
      WRITE(6,131)NO(31),IRETMS
      WRITE(6,132)NO(32),FRICMS,COEFMS,GAMMS,AEI1,AEI2
      II=2*NOLEG
      DO 6 I=1,II
    6 WRITE(6,1400)NUMB(4),I,XDSCB(I),YDSCB(I),ZDSCB(I)
      WRITE(6,133)NO(33),(PFCDS(I),I=1,5)
      WRITE(6,134)NO(34),(PFTDS(I),I=1,5)
      WRITE(6,135)NO(35),(CDCDS(I),I=1,5)
      WRITE(6,136)NO(36),(CDTDS(I),I=1,5)
      WRITE(6,137)NO(37),(SRCDS(I),I=1,5)
      WRITE(6,138)NO(38),(SRTDS(I),I=1,5)
      WRITE(6,139)NO(39),SCMXDS,STMXDS,SRUCDS,SRUTDS
      WRITE(6,140)NO(40),IRETDS
      WRITE(6,141)NO(41),FRICDS,COEFDS,GAMDS
C
C     MODAL DATA
C
      WRITE(6,142)NO(42),MODEIN
      IF(NMODES.EQ.0)WRITE(6,242)
      IF(MODEIN.EQ.0)RETURN
      DO 8 I=1,MODEIN
    8 WRITE(6,1500)NUMB(5),I,GM(I),OMEGA(I)
```

```
      DO 9 I=1,MODEIN                                                   DATA1680
    9 WRITE(6,1600)NUMB(6),I,WNX(I),WNY(I),WNZ(I)                       DATA1690
      DO 10 I=1,MODEIN                                                  DATA1700
   10 WRITE(6,1700)NUMB(7),I,PX(I),PY(I),PZ(I)                          DATA1710
      DO 11 I=1,NOLEG                                                   DATA1720
   11 WRITE(6,1800)NUMB(8),I,(PMSX(I,J),J=1,MODEIN)                     DATA1730
      DO 12 I=1,NOLEG                                                   DATA1740
   12 WRITE(6,1900)NUMB(9),I,(PMSY(I,J),J=1,MODEIN)                     DATA1750
      DO 13 I=1,NOLEG                                                   DATA1760
   13 WRITE(6,2000)NUMB(10),I,(PMSZ(I,J),J=1,MODEIN)                    DATA1770
      DO 14 I=1,II                                                      DATA1780
   14 WRITE(6,2100)NUMB(11),I,(PDSX(I,J),J=1,MODEIN)                    DATA1790
      DO 15 I=1,II                                                      DATA1800
   15 WRITE(6,2200)NUMB(12),I,(PDSY(I,J),J=1,MODEIN)                    DATA1810
      DO 16 I=1,II                                                      DATA1820
   16 WRITE(6,2300)NUMB(13),I,(PDSZ(I,J),J=1,MODEIN)                    DATA1830
      IF(NOOUT.EQ.0)GO TO 21                                            DATA1840
      DO 17 I=1,NOOUT                                                   DATA1850
   17 WRITE(6,2400)NUMB(14),I,(POUTX(I,J),J=1,MODEIN)                   DATA1860
      DO 18 I=1,NOOUT                                                   DATA1870
   18 WRITE(6,2500)NUMB(15),I,(POUTY(I,J),J=1,MODEIN)                   DATA1880
      DO 19 I=1,NOOUT                                                   DATA1890
   19 WRITE(6,2600)NUMB(16),I,(POUTZ(I,J),J=1,MODEIN)                   DATA1900
   21 CONTINUE                                                          DATA1910
      DO 20 I=1,MODEIN                                                  DATA1920
   20 WRITE(6,2700)NUMB(17),I,PCGX(I),PCGY(I),PCGZ(I)                   DATA1930
      RETURN                                                            DATA1940
C                                                                       DATA1950
C     FORMAT STATEMENTS                                                 DATA1960
C                                                                       DATA1970
  101 FORMAT(5H CARD/2X,2HNO//10X,28H* * PROGRAM CONTROL DATA * *//,    DATA1980
     *       I4,10H CASE NO =,I6)                                       DATA1990
  102 FORMAT(I4,9H TIMAX = ,E10.3,10H IPTCNT = ,I2,9H INLEG = ,I2,      DATA2000
     *       10H IFPPRT = ,I2)                                          DATA2010
  103 FORMAT(I4,10H NMODES = ,I2, 9H NOOUT = ,I2)                       DATA2020
  104 FORMAT(I4,10H INDFXD = ,I2,10H INDFYD = ,I2,10H INDFZD = ,I2)     DATA2030
  105 FORMAT(I4,10H INDFXR = ,I2,10H INDFYR = ,I2,10H INDFZR = ,I2)     DATA2040
  106 FORMAT(I4, 8H HMAX = ,E10.3, 8H HMIN = ,E10.3, 8H EMAX = ,E10.3,  DATA2050
     *       8H EMIN = ,E10.3,6H IP = ,I2)                              DATA2060
  107 FORMAT(I4,9H IVARH = ,I2,8H IMTH = ,I2,10H CUTERR = ,E10.3)       DATA2070
  108 FORMAT(I4,9H NFORC = ,I2,10H IQUOUT = ,I2,10H JCKSAB = ,I2,       DATA2080
     *10H IDSETN = ,I2)                                                 DATA2090
  109 FORMAT(//10X,26H* * INITIAL CONDITIONS * *//,I4, 8H ZETA = ,E10.3 DATA2100
     *       , 8H GRAV = ,E10.3, 9H GRAVE = ,E10.3)                     DATA2110
 110  FORMAT(I4,8H ANGX = ,E10.3,8H ANGY = ,E10.3,8H ANGZ = ,E10.3)     DATA2120
  111 FORMAT(I4,6H WX = ,E10.3,6H WY = ,E10.3,6H WZ = ,E10.3)           DATA2130
  112 FORMAT(I4,8H VELX = ,E10.3,8H VELY = ,E10.3,8H VELZ = ,E10.3)     DATA2140
  113 FORMAT(//10X,24H* * CENTER BODY DATA * *,//,I4,10H CBMASS = ,     DATA2150
     *       E10.3)                                                     DATA2160
  114 FORMAT(I4,9H CBIXX = ,E10.3,9H CBIYY = ,E10.3,9H CBIZZ = ,E10.3)  DATA2170
  115 FORMAT(I4,9H CBIXY = ,E10.3,9H CBIXZ = ,E10.3,9H CBIYZ = ,E10.3)  DATA2180
 1100 FORMAT(I4,5H I = ,I2,8H XOUT = ,E10.3,8H YOUT = ,E10.3,8H ZOUT = ,DATA2190
     *       E10.3)                                                     DATA2200
 1101 FORMAT(/5X,26HNO SECONDARY OUTPUT POINTS)                         DATA2210
  116 FORMAT(//10X,20H* * FOOTPAD DATA * *//I4,10H FPMASS = ,E10.3)     DATA2220
  117 FORMAT(I4,10H RAD(I) = ,4(2X,E10.3))                              DATA2230
```

```
  118 FORMAT(I4,9H SS(I) = ,4(2X,E10.3))                                DATA2240
  119 FORMAT(I4,13H ATTHCK(I) = ,3(2X,E10.3))                           DATA2250
  120 FORMAT(I4,13H ATTPRS(I) = ,3(2X,E10.3))                           DATA2260
  121 FORMAT(//10X,17H* * SOIL DATA * *,//,I4, 9H NTYPE = ,I2,/,4X,     DATA2270
     *       25H (PRIMARY SOIL MECHANICS))                              DATA2280
  221 FORMAT(//10X,17H* * SOIL DATA * *,//,I4,9H NTYPE = ,I2,/,4X,      DATA2290
     *       27H (SECONDARY SOIL MECHANICS))                            DATA2300
  122 FORMAT(I4,12H SOILP(I) = ,E10.3,3H , ,E10.3,3H , ,E10.3)          DATA2310
  123 FORMAT(//10X,16H* * LEG DATA * *,//,I4, 9H NOLEG = ,I2,           DATA2320
     *       8H ILEG = ,I2,10H DRAGST = ,E10.3)                         DATA2330
 1200 FORMAT(I4,5H I = ,I2,7H XFP = ,E10.3,7H YFP = ,E10.3,7H ZFP = ,   DATA2340
     *       E10.3)                                                     DATA2350
 1300 FORMAT(I4,5H I = ,I2,9H XMSCB = ,E10.3,9H YMSCB = ,E10.3,         DATA2360
     *       9H ZMSCB = ,E10.3)                                         DATA2370
  124 FORMAT(I4,9H PFCMS = ,5(2X,E10.3))                                DATA2380
  125 FORMAT(I4,9H PFTMS = ,5(2X,E10.3))                                DATA2390
  126 FORMAT(I4,9H CDCMS = ,5(2X,E10.3))                                DATA2400
  127 FORMAT(I4,9H CDTMS = ,5(2X,E10.3))                                DATA2410
  128 FORMAT(I4,9H SRCMS = ,5(2X,E10.3))                                DATA2420
  129 FORMAT(I4,9H SRTMS = ,5(2X,E10.3))                                DATA2430
  130 FORMAT(I4,10H SCMXMS = ,E10.3,10H STMXMS = ,E10.3,10H SRUCMS = ,  DATA2440
     *       E10.3,10H SRUTMS = ,E10.3)                                 DATA2450
  131 FORMAT(I4,10H IRETMS = ,I2)                                       DATA2460
  132 FORMAT(I4,10H FRICMS = ,E10.3,10H COEFMS = ,E10.3,9H GAMMS = ,    DATA2470
     *       E10.3,8H AEI1 = ,E10.3,8H AEI2 = ,E10.3)                   DATA2480
 1400 FORMAT(I4,5H I = ,I2,9H XDSCB = ,E10.3,9H YDSCB = ,E10.3,         DATA2490
     *       9H ZDSCB = ,E10.3)                                         DATA2500
  133 FORMAT(I4,9H PFCDS = ,5(2X,E10.3))                                DATA2510
  134 FORMAT(I4,9H PFTDS = ,5(2X,E10.3))                                DATA2520
  135 FORMAT(I4,9H CDCDS = ,5(2X,E10.3))                                DATA2530
  136 FORMAT(I4,9H CDTDS = ,5(2X,E10.3))                                DATA2540
  137 FORMAT(I4,9H SRCDS = ,5(2X,E10.3))                                DATA2550
  138 FORMAT(I4,9H SRTDS = ,5(2X,E10.3))                                DATA2560
  139 FORMAT(I4,10H SCMXDS = ,E10.3,10H STMXDS = ,E10.3,10H SRUCDS = ,  DATA2570
     *       E10.3,10H SRUTDS = ,E10.3)                                 DATA2580
  140 FORMAT(I4,10H IRETDS = ,I2)                                       DATA2590
  141 FORMAT(I4,10H FRICDS = ,E10.3,10H COEFDS = ,E10.3,9H GAMDS = ,    DATA2600
     *       E10.3)                                                     DATA2610
  142 FORMAT(//10X,18H* * MODAL DATA * *,//,I5,10H MODEIN = ,I2)        DATA2620
  242 FORMAT(/5X,25HRIGID CENTER BODY ASSUMED)                          DATA2630
 1500 FORMAT(I5,5H I = ,I2,6H GM = ,E10.3,9H OMEGA = ,E10.3)            DATA2640
 1600 FORMAT(I5,5H I = ,I2,7H WNX = ,E10.3,7H WNY = ,E10.3,7H WNZ = ,   DATA2650
     *       E10.3)                                                     DATA2660
 1700 FORMAT(I5,5H I = ,I2,6H PX = ,E10.3,6H PY = ,E10.3,6H PZ = ,E10.3)DATA2670
 1800 FORMAT(I5,5H I = ,I2,8H PMSX = ,5(2X,E10.3))                      DATA2680
 1900 FORMAT(I5,5H I = ,I2,8H PMSY = ,5(2X,E10.3))                      DATA2690
 2000 FORMAT(I5,5H I = ,I2,8H PMSZ = ,5(2X,E10.3))                      DATA2700
 2100 FORMAT(I5,5H I = ,I2,8H PDSX = ,5(2X,E10.3))                      DATA2710
 2200 FORMAT(I5,5H I = ,I2,8H PDSY = ,5(2X,E10.3))                      DATA2720
 2300 FORMAT(I5,5H I = ,I2,8H PDSZ = ,5(2X,E10.3))                      DATA2730
 2400 FORMAT(I5,5H I = ,I2,9H POUTX = ,5(2X,E10.3))                     DATA2740
 2500 FORMAT(I5,5H I = ,I2,9H POUTY = ,5(2X,E10.3))                     DATA2750
 2600 FORMAT(I5,5H I = ,I2,9H POUTZ = ,5(2X,E10.3))                     DATA2760
 2700 FORMAT(I5,5H I = ,I2,8H PCGX = ,E10.3,8H PCGY = ,E10.3,8H PCGZ = ,DATA2770
     *       E10.3)                                                     DATA2780
 3000 FORMAT(/5X,50HWARNING - PROGRAM CONTROL DATA INDICATES THIS CASE,/DATA2790
```

```
*       55HIS PART OF A MULTIPLE DATA SET RUN AND A SECONDARY TIME,/DATA2800
*       52HHISTORY TAPE IS TO BE GENERATED. THIS COMBINATION OF,/   DATA2810
*       42HPROGRAM OPTIONS MUST BE HANDLED WITH CARE.,)             DATA2820
 END                                                                DATA2830
```

```
      OVERLAY ( LLMPT5, 2, 0 )                                          OVER  30

      PROGRAM  LLMPEX                                                   LLEX  10
      COMMON / IZZZRM / IPTATL,IPTOTL, LOCNAM(90)                       LLEX  20
      DIMENSION XSI(8), IAD(8), IND(8)                                  LLEX  30
      EQUIVALENCE ( TIMHSA(1), LTIMHS )                                 LLEX  40
      DIMENSION TIMHSC(275)                                             LLEX  50
      DIMENSION TIMHSB(275)                                             LLEX  60
      DIMENSION IMHSA(1)                                                LLEX  70
      EQUIVALENCE ( IMHSA(1), TIMHSA(1) )                               LLEX  80
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)                            LLEX  90
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),                       LLEX 100
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),                     LLEX 110
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)                    LLEX 120
      DIMENSION XMN1(8)                                                 LLEX 130
      DIMENSION INOUTV(5)                                               LLEX 140
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),     LLEX 150
     1    INDFPC(6),                                                    LLEX 160
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),   LLEX 170
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),   LLEX 180
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),   LLEX 190
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5), LLEX 200
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),   LLEX 210
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)                            LLEX 220
      EQUIVALENCE ( NFTPDS, NOLEG )                                     LLEX 230
      COMMON COMINT(400)                                                LLEX 240
      EQUIVALENCE ( COMINT(  3 ), IBOTM      )                          LLEX 250
      EQUIVALENCE ( COMINT(  4 ), XSD    )                              LLEX 260
      EQUIVALENCE ( COMINT(  8 ), XSDD   )                              LLEX 270
      EQUIVALENCE ( COMINT( 12 ), YS     )                              LLEX 280
      EQUIVALENCE ( COMINT( 16 ), YSD    )                              LLEX 290
      EQUIVALENCE ( COMINT( 20 ), YSDD   )                              LLEX 300
      EQUIVALENCE ( COMINT( 24 ), ZS     )                              LLEX 310
      EQUIVALENCE ( COMINT( 28 ), ZSD    )                              LLEX 320
      EQUIVALENCE ( COMINT( 32 ), ZSDD   )                              LLEX 330
      EQUIVALENCE ( COMINT( 36 ), PHI    )                              LLEX 340
      EQUIVALENCE ( COMINT( 40 ), PHID   )                              LLEX 350
      EQUIVALENCE ( COMINT( 44 ), WX     )                              LLEX 360
      EQUIVALENCE ( COMINT( 48 ), WXD    )                              LLEX 370
      EQUIVALENCE ( COMINT( 52 ), THTA   )                              LLEX 380
      EQUIVALENCE ( COMINT( 56 ), THTAD  )                              LLEX 390
      EQUIVALENCE ( COMINT( 60 ), WY     )                              LLEX 400
      EQUIVALENCE ( COMINT( 64 ), WYD    )                              LLEX 410
      EQUIVALENCE ( COMINT( 68 ), PSI    )                              LLEX 420
      EQUIVALENCE ( COMINT( 72 ), PSID   )                              LLEX 430
      EQUIVALENCE ( COMINT( 76 ), WZ     )                              LLEX 440
      EQUIVALENCE ( COMINT( 80 ), WZD    )                              LLEX 450
      EQUIVALENCE ( COMINT( 84 ), GC     )                              LLEX 460
      EQUIVALENCE ( COMINT(104 ), GCD    )                              LLEX 470
      EQUIVALENCE ( COMINT(124 ), GCDD   )                              LLEX 480
      EQUIVALENCE ( COMINT(144 ), XFPS   )                              LLEX 490
      EQUIVALENCE ( COMINT(164 ), XFPSD  )                              LLEX 500
      EQUIVALENCE ( COMINT(184 ), XFPSDD )                              LLEX 510
      EQUIVALENCE ( COMINT(204 ), YFPS   )                              LLEX 520
```

```
      EQUIVALENCE ( COMINT(224 ), YFPSD   )                             LLEX 530
      EQUIVALENCE ( COMINT(244 ), YFPSDD )                              LLEX 540
      EQUIVALENCE ( COMINT(264 ), ZFPS    )                             LLEX 550
      EQUIVALENCE ( COMINT(284 ), ZFPSD   )                             LLEX 560
      EQUIVALENCE ( COMINT(304 ), ZFPSDD )                              LLEX 570
      EQUIVALENCE ( COMINT( 324 ), TIME   )                             LLEX 580
      EQUIVALENCE ( COMINT( 325 ), HMAX   )                             LLEX 590
      EQUIVALENCE ( COMINT( 326 ), HMIN   )                             LLEX 600
      EQUIVALENCE ( COMINT( 327 ), EMIN   )                             LLEX 610
      EQUIVALENCE ( COMINT( 328 ), EMAX   )                             LLEX 620
      EQUIVALENCE ( COMINT( 329 ), XSI    )                             LLEX 630
      EQUIVALENCE ( COMINT ( 337 ), HZ      )                           LLEX 640
      EQUIVALENCE ( COMINT( 338 ), CUTERR)                              LLEX 650
      EQUIVALENCE ( COMINT( 339 ), IP     )                             LLEX 660
      EQUIVALENCE ( COMINT( 340 ), IVARH )                              LLEX 670
      EQUIVALENCE ( COMINT( 341 ), IMTH   )                             LLEX 680
      EQUIVALENCE ( COMINT( 342 ), IPRNT )                              LLEX 690
      EQUIVALENCE ( COMINT( 343 ), IFIN   )                             LLEX 700
      EQUIVALENCE ( COMINT( 344 ), IAD    )                             LLEX 710
      EQUIVALENCE ( COMINT( 352 ), IND    )                             LLEX 720
      EQUIVALENCE ( COMINT( 360 ), JCUT   )                             LLEX 730
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )                             LLEX 740
      EQUIVALENCE ( COMINT( 363 ), IVAL     )                           LLEX 750
      EQUIVALENCE ( COMINT( 364 ), XS       )                           LLEX 760
C                   COMINT(364-367) USED BY XS                          LLEX 770
      EQUIVALENCE ( COMINT( 368 ), CURDT   )                            LLEX 780
      EQUIVALENCE ( COMINT( 369 ), XMN1    )                            LLEX 790
C       XMN1 USES COMINT(369-376)                                       LLEX 800
      COMMON                                                            LLEX 810
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY, LLEX 820
     2    DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),       LLEX 830
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)      ,       LLEX 840
     4    GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)      ,       LLEX 850
     5    FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,               LLEX 860
     6    SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,               LLEX 870
     7    PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,               LLEX 880
     8    PDSY(10,5)    , PDSZ(10,5)    , FLXS    , FLYS  , FLZS  ,     LLEX 890
     9    TLXL   ,   TLYL   , TLZL   , SLO    , XMSCB(5)      ,         LLEX 900
     C    YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,               LLEX 910
     D    YDSCB(10)     , ZDSCB(10)     , ILEG    , IMS   ,             LLEX 920
     E    FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ  ,            LLEX 930
     F    PVFPX , PVFPY , PVFPZ , NOLEG  , SLOMS(5) ,                   LLEX 940
     G    SLODS(10)                                                     LLEX 950
      COMMON                                                            LLEX 960
     1    PFCMS(5)      , PFCDS(5)      , PFTDS(5)      ,               LLEX 970
     2    PFTMS(5)      , SRCDS(5)      , SRCMS(5)      ,               LLEX 980
     3    SRTDS(5)   ,    SRTMS(5)      , COEFDS, COEFMS, GAMDS ,       LLEX 990
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,       LLEX1000
     5    STMXDS, STMXMS, CDCDS(5)      , CDCMS(5)      ,               LLEX1010
     6    CDTDS(5)      , CDTMS(5)      , FRICDS, FRICMS, IRETDS,       LLEX1020
     7    IRETMS, STROKE(10)    , STRKDS(10)    , STRKMS( 5)    ,       LLEX1030
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),    LLEX1040
     9    ATTPRS(3)     , ADIST , SOILP(3)      , NMODES,               LLEX1050
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),       LLEX1060
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),    LLEX1070
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,       LLEX1080
```

```
     D    INDFZR, TIMAX , DRAGST, IFP                                   LLEX1090
      COMMON                                                            LLEX1100
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),   LLEX1110
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),       LLEX1120
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),       LLEX1130
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,         LLEX1140
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)          LLEX1150
      COMMON                                                            LLEX1160
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),                LLEX1170
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)                LLEX1180
     3   ,SLNGMS(5)  ,SLNGDS(10), CURDSL, INLEG , IFPPRT,               LLEX1190
     4    IMPACT(5)  ,IPRTFP(5),  KOUNT(5)   ,ANGX,  ANGY,  ANGZ        LLEX1200
      DIMENSION AHOLD(6),   AMINV(6,6), FLTL(6)                         LLEX1210
      DIMENSION GF(5), GFO(5)                                           LLEX1220
      DIMENSION AMFP(5), FO(3), FE(6)                                   LLEX1230
      DIMENSION AME2(3)                                                 LLEX1240
C              PRECALCULATE SOME CONSTANTS USED LATER IN MODE WORK      LLEX1250
      DIMENSION  PYZ(5),PYZ2(5),PXZ(5),PXZ2(5),PXY(5),PXY2(5),          LLEX1260
     1   WNZNY(5),WNXNZ(5),WNYNX(5),PYNZ2(5),PZNX2(5),PXNY2(5),W2GM(5)  LLEX1270
C              SET INTEGRATION LIST UPDATE INDICATOR                    LLEX1280
      INDIVR = 0                                                        LLEX1290
      LTIMHS = 273                                                      LLEX1300
C         PRINT COUNT                                                   LLEX1310
      KPTCNT = 99999                                                    LLEX1320
C         PITCH ANGLE LIMIT                                             LLEX1330
      ATIPMX = 1.55                                                     LLEX1340
C         TIME HISTORY UPDATE INDICATOR                                 LLEX1350
      INDBPU = 1                                                        LLEX1360
C         INTEGRATION LOOP FIRST PASS INDICATOR                         LLEX1370
      INDIFP = 0                                                        LLEX1380
      IF ( NMODES .EQ. 0 ) GO TO 110                                    LLEX1390
      DO  100  I = 1, NMODES                                            LLEX1400
      PYZ(I)  =   PY(I)+PZ(I)                                           LLEX1410
      PYZ2(I) =   PYZ(I)+PYZ(I)                                         LLEX1420
      PXZ(I)  =   PX(I)+PZ(I)                                           LLEX1430
      PXZ2(I) =   PXZ(I)+PXZ(I)                                         LLEX1440
      PXY(I)  =   PX(I)+PY(I)                                           LLEX1450
      PXY2(I) =   PXY(I)+PXY(I)                                         LLEX1460
      WNZNY(I)=   WNZ(I)-WNY(I)                                         LLEX1470
      WNXNZ(I)=   WNX(I)-WNZ(I)                                         LLEX1480
      WNYNX(I)=   WNY(I)-WNX(I)                                         LLEX1490
      W2GM(I) =  OMEGA(I)*OMEGA(I)*GM(I)                                LLEX1500
      PYNZ2(I)=  2.*( PY(I)- PZ(I) )                                    LLEX1510
      PZNX2(I)=  2.*( PZ(I)- PX(I) )                                    LLEX1520
  100 PXNY2(I)=  2.*( PX(I)- PY(I) )                                    LLEX1530
C         SET MODAL ANALYSIS MASS TO ZERO                               LLEX1540
      DO  105  I = 1, 3                                                 LLEX1550
  105 AME1(I) = 0.0                                                     LLEX1560
  110 CONTINUE                                                          LLEX1570
C                                                                       LLEX1580
C         AT PRESENT ALL FOOT PADS ARE OF THE SAME MASS                 LLEX1590
C                                                                       LLEX1600
      DO  70  I = 1, NFTPDS                                             LLEX1610
   70 AMFP(I)  = FPMASS                                                 LLEX1620
C              INDFAR=3 IF ROTATIONS ARE NOT INTEGRATED                 LLEX1630
      INDFAR =  INDFXR+INDFYR+INDFZR                                    LLEX1640
```

```
C             PRECALCULATE MOMENT OF INERTIA CONSTANTS
      CIZZYY =  CBIZZ-CBIYY
      CIXXZZ =  CBIXX-CBIZZ
      CIYYXX =  CBIYY-CBIXX
C
C         SET INTEGRATION VARIABLE LIST AND CUTOFF LIST COUNTERS
C
      IPTATL = 0
      IPTOTL = 0
C                             INITIALIZE THE INTEGRATION ROUTINES
      CALL SETUP
C
C                             XS, XSD, AND XSDD       ARE INTEGRATION V.
      IF ( INDFXD .NE. 0 )  GO TO 210
      CALL INUPD ( 364 )
      CALL INUPD (   4    )
      CALL INUPD (   8    )
  210 IF ( INDFYD .NE. 0 )  GO TO 220
C                             YS, YSD, AND YSDD       ARE INTEGRATION V.
      CALL INUPD (  12    )
      CALL INUPD (  16    )
      CALL INUPD (  20    )
  220 IF ( INDFZD .NE. 0 )  GO TO 230
C                             ZS, ZSD, AND YSDD       ARE INTEGRATION V.
      CALL INUPD  ( 24    )
      CALL INUPD  ( 28    )
      CALL INUPD  ( 32    )
  230 IF ( INDFXR .NE. 0 ) GO TO 240
C                             WX, WXD, PHID, AND PHI  ARE INTEGRATION V.
      CALL INUPD (   44    )
      CALL INUPD (   48    )
      CALL INUPD (   40    )
      CALL INUPD (   36    )
  240 IF ( INDFYR .NE. 0 ) GO TO 250
C                             WY, WYD, THTAD, AND THTA ARE INTEGRATION V
      CALL INUPD (   60    )
      CALL INUPD (   64    )
      CALL INUPD (   56    )
      CALL INUPD (   52    )
  250 IF ( INDFZR .NE. 0 ) GO TO 260
C                             WZ, WZD, PSID, AND PSI  ARE INTEGRATION V.
      CALL INUPD (   76    )
      CALL INUPD (   80    )
      CALL INUPD (   72    )
      CALL INUPD (   68    )
  260 CONTINUE
C                             GC(I), GCD(I), AND GCDD(I) ARE INTFGRAT. V
      IF ( NMODES .EQ. 0 ) GO TO 290
      DO  280  I = 1, NMODES
      J = 4*I
      CALL INUPD ( 80+ J )
      CALL INUPD (100+ J )
  280 CALL INUPD (120+ J )
  290 CONTINUE
    1 CONTINUE
C
```

```
      CALL GEOM
C
      IF ( INDIFP .EQ. 1) GO TO 64
C
C              INITIALIZE FOOT PAD POSITIONS IN SURFACE COORDINATES
C                   ALL SO SET UP FOOT PAD CUTOFF VARIABLES AND
C                   INTEGRATION QUANTIES
C
      XS = 100.*GRAVE
      YS = 0.0
      ZS = 0.0
      DO  60   I = 1, NFTPDS
      A        =      DC(1,1)*XFP(I) + DC(1,2)*YFP(I) + DC(1,3)*ZFP(I)
     1              -  ATTH(I)
   60 XS = AMIN1( XS, A )
      XS = -XS
C
C     PRINT LANDER INITIAL CONDITONS
C
      WRITE(6,5010)XS,YS,ZS,XSD,YSD,ZSD,WXIN,WYIN,WZIN
 5010 FORMAT(1H1,/20X,46HLANDER INITIAL CONDITIONS (SURFACE COORDINATE ,
     *      7HSYSTEM),//37X,1H(,10X,1HY,
     *      10X,1HZ//6X,21HINITIAL C.G. POSITION,6X,3(F9.2,2X),//
     *      6X,21HINITIAL C.G. VELOCITY,6X,3(F9.2,2X)//
     *      6X,24HINITIAL ANGULAR VELOCITY,3X,3(F9.2,2X))
      DO  62   I = 1, NFTPDS
      XFPS(1,I) =  XS+DC(1,1)*XFP(I)+DC(1,2)*YFP(I)+DC(1,3)*ZFP(I)
      J  = 4*I
      IF ( XFPS(1,I)-ATTH(I) .LT. CUTERR ) GO TO 61
C         XFPS( I ) IS A CUTOFF VARIABLE
      CALL LOC(140+J,ATTH(I)+(.5+GRAVE*2.5E-06)*CUTERR,1)
      INDFPI( I ) = IPTATL
      INDFPC(I) = 0
      GO  TO 62
C              XFPS(I),YFPS(I),ZFPS(I),XFPSD(I),YFPSD(I), ZFPSD(I)
C                  XFPSDD(I), YFPSDD(I), ZFPSDD(I) ARE INTEGRATION VAR.
   61 CALL INUPD ( 140 + J )
      INDFPI( I) = IPTOTL
      CALL INUPD ( 160 + J )
      CALL INUPD ( 180 + J )
      CALL INUPD ( 200 + J )
      CALL INUPD ( 220  + J )
      CALL INUPD ( 240 + J )
      CALL INUPD ( 260 + J )
      CALL INUPD ( 280 + J )
      CALL INUPD ( 300 + J )
      YFPS(1,I) =  YS+DC(2,1)*XFP(I)+DC(2,2)*YFP(I)+DC(2,3)*ZFP(I)
      ZFPS(1,I) =  ZS+DC(3,1)*XFP(I)+DC(3,2)*YFP(I)+DC(3,3)*ZFP(I)
      A    =  WY * ZFP(I)  -  WZ * YFP(I)
      B    =  WZ * XFP(I)  -  WX * ZFP(I)
      C    =  WX * YFP(I)  -  WY * XFP(I)
      XFPSD(1,I) = XSD + DC(1,1)*A + DC(1,2)*B + DC(1,3)*C
      YFPSD(1,I) = YSD + DC(2,1)*A + DC(2,2)*B + DC(2,3)*C
      ZFPSD(1,I) = ZSD + DC(3,1)*A + DC(3,2)*B + DC(3,3)*C
C         SET THIS FOOT PAD*S CONTACT INDICATOR
      INDFPC ( I ) = 2
```

```
   62 CONTINUE
C
C                 SAVE NON INTEGRATION TIMEHISTORY VARIABLES IN CASE
C                               OF INITIAL BACKUP
      DO 63 I = 1, LTIMHS
   63 TIMHSC(I) = TIMHSA(I)
   64 CONTINUE
C
C          PROCESS THE FT. PAD. CALLS
C
      DO  380  I  = 1, 6
  380 FLTL(I) = 0.0
      IF ( NMODES .EQ. 0 ) GO TO 395
      DO  390 I = 1, 3
      AME2(I) = 0.0
  390 GF(I) = 0.0
  395 CONTINUE
      DO  430  IFP = 1, NFTPDS
      IF ( INDFPC(IFP) .LE. 0 ) GO TO 405
C
C     CONTACTING FOOTPAD
C
      IF(INDIFP.EQ.1) CALL FTPAD
C
      IF(IBOTM.NE.0)GO TO 3110
C          SUM UP THE RESULTAN LEG FORCES
      FLTL(1) = FLTL(1) + FLXS
      FLTL(2) = FLTL(2) + FLYS
      FLTL(3) = FLTL(3) + FLZS
      FLTL(4) = FLTL(4) + TLXL
      FLTL(5) = FLTL(5) + TLYL
      FLTL(6) = FLTL(6) + TLZL
      IF ( NMODES .EQ. 0 ) GO TO 406
C          SUM THE GEN. FORCES TO ELASTIC EFFECTS OF THE MAIN AND
C                 DRAG STRUTS
      DO   404  I = 1, NMODES
  404 GF(I) = GF(I) + GFLEGS(I)
  406 CONTINUE
C       F.P. ACCEL. =(GRAVITY LOADS   -STRUT LOADS+STOIL LOADS)/F.P. MASS
      XFPSDD(1,IFP) =(-GCOSZT*AMFP(IFP)+FSXSI(IFP)+SOILX(IFP))/AMFP(IFP)
      YFPSDD(1,IFP) =(                 +FSYSI(IFP)+SOILY(IFP))/AMFP(IFP)
      ZFPSDD(1,IFP) =( GSINZT*AMFP(IFP)+FSZSI(IFP)+SOILZ(IFP))/AMFP(IFP)
C                INTEGRATE TO GET XFPS, YFPS, AND ZFPS
      CALL INTEG ( XFPSDD(1,IFP), XFPSD(1,IFP) )
      CALL INTEG ( XFPSD (1,IFP), XFPS (1,IFP) )
      CALL INTEG ( YFPSDD(1,IFP), YFPSD(1,IFP) )
      CALL INTEG ( YFPSD (1,IFP), YFPS (1,IFP) )
      CALL INTEG ( ZFPSDD(1,IFP), ZFPSD(1,IFP) )
      CALL INTEG ( ZFPSD (1,IFP), ZFPS (1,IFP) )
      GO TO 430
C
C     NONCONTACTING  FOOTPAD
C
  405 CONTINUE
      XXX=XFP(IFP)
      YYY=YFP(IFP)
```

```
      ZZZ=ZFP(IFP)
      IF(NMODES.EQ.0)GO TO 320
      IF(ILEG.EQ.0)GO TO 304
      RRRT=DRAGST/SLNGMS(IFP)
      XXX=XMSCB(IFP)+RRRT*(XFP(IFP)-XMSCB(IFP))
      YYY=YMSCB(IFP)+RRRT*(YFP(IFP)-YMSCB(IFP))
      ZZZ=ZMSCB(IFP)+RRRT*(ZFP(IFP)-ZMSCB(IFP))
  304 CONTINUE
      XMST=XMSCB(IFP)
      YMST=YMSCB(IFP)
      ZMST=ZMSCB(IFP)
      NN1=2*(IFP-1)+1
      NN2=NN1+1
      XDS1T=XDSCB(NN1)
      YDS1T=YDSCB(NN1)
      ZDS1T=ZDSCB(NN1)
      XDS2T=XDSCB(NN2)
      YDS2T=YDSCB(NN2)
      ZDS2T=ZDSCB(NN2)
      DO 305 I=1,NMODES
      XMST=XMST+PMSX(IFP,I)*GC(1,I)
      YMST=YMST+PMSY(IFP,I)*GC(1,I)
      ZMST=ZMST+PMSZ(IFP,I)*GC(1,I)
      XDS1T=XDS1T+PDSX(NN1,I)*GC(1,I)
      YDS1T=YDS1T+PDSY(NN1,I)*GC(1,I)
      ZDS1T=ZDS1T+PDSZ(NN1,I)*GC(1,I)
      XDS2T=XDS2T+PDSX(NN2,I)*GC(1,I)
      YDS2T=YDS2T+PDSY(NN2,I)*GC(1,I)
      ZDS2T=ZDS2T+PDSZ(NN2,I)*GC(1,I)
  305 CONTINUE
  310 CONTINUE
      DXMS=XXX-XMST
      DYMS=YYY-YMST
      DZMS=ZZZ-ZMST
      DXDS1=XXX-XDS1T
      DYDS1=YYY-YDS1T
      DZDS1=ZZZ-ZDS1T
      DXDS2=XXX-XDS2T
      DYDS2=YYY-YDS2T
      DZDS2=ZZZ-ZDS2T
      FUNMS = DXMS**2+DYMS**2+DZMS**2-SLNGMS(IFP)**2
      IF(ILEG.NE.0)FUNMS=DXMS**2+DYMS**2+DZMS**2-DRAGST**2
      FUNDS1=DXDS1**2+DYDS1**2+DZDS1**2-SLNGDS(NN1)**2
      FUNDS2=DXDS2**2+DYDS2**2+DZDS2**2-SLNGDS(NN2)**2
      ERR=ABS(FUNMS)+ABS(FUNDS1)+ABS(FUNDS2)
      IF(ERR.LT.(.00005*GRAVE)) GO TO 320
      AAA=4.*(DYDS1*DZDS2-DZDS1*DYDS2)
      BBB=4.*(DZDS1*DXDS2-DXDS1*DZDS2)
      CCC=4.*(DYDS2*DXDS1-DXDS2*DYDS1)
      DELL=2.*(DXMS*AAA+DYMS*BBB+DZMS*CCC)
      DELX=-FUNMS*AAA-FUNDS1*BBB-FUNDS2*CCC
      DELY=4.*DXMS*(DZDS1*FUNDS2-DZDS2*FUNDS1)-FUNMS*BBB+4.*DZMS*
     *    (DXDS2*FUNDS1-DXDS1*FUNDS2)
      DELZ=4.*DXMS*(DYDS2*FUNDS1-DYDS1*FUNDS2)+4.*DYMS*(DXDS1*FUNDS2-
     *    DXDS2*FUNDS1)-FUNMS*CCC
      XXX=XXX+DELX/DELL
```

```
      YYY=YYY+DELY/DELL                                                 LLEX3890
      ZZZ=ZZZ+DELZ/DELL                                                 LLEX3900
      GO TO 310                                                         LLEX3910
  320 CONTINUE                                                          LLEX3920
      IF(ILEG.EQ.0)GO TO 321                                            LLEX3930
      IF(NMODES.EQ.0)GO TO 321                                          LLEX3935
      CX=(XXX-XMSCB(IPF))/DRAGST                                        LLEX3940
      CY=(YYY-YMSCB(IFP))/DRAGST                                        LLEX3950
      CZ=(ZZZ-ZMSCB(IFP))/DRAGST                                        LLEX3960
      DDD=SLNGMS(IFP)-DRAGST                                            LLEX3970
      XXX=XXX+CX*DDD                                                    LLEX3980
      YYY=YYY+CY*DDD                                                    LLEX3990
      ZZZ=ZZZ+CZ*DDD                                                    LLEX4000
  321 CONTINUE                                                          LLEX4010
      XFPS(1,IFP)=XS+DC(1,1)*XXX+DC(1,2)*YYY+DC(1,3)*ZZZ                LLEX4020
      YFPS(1,IFP)=YS+DC(2,1)*XXX+DC(2,2)*YYY+DC(2,3)*ZZZ                LLEX4030
      ZFPS(1,IFP)=ZS+DC(3,1)*XXX+DC(3,2)*YYY+DC(3,3)*ZZZ                LLEX4040
      NNN=2*(IFP-1)+1                                                   LLEX4050
      FORMS(IFP)=0.0                                                    LLEX4060
      FORDS(NNN)=0.0                                                    LLEX4070
      FORDS(NNN+1)=0.0                                                  LLEX4080
      IF(NFORC.EQ.0) GO TO 430                                          LLEX4090
      SAVMSX(IFP)=0.0                                                   LLEX4100
      SAVMSY(IFP)=0.0                                                   LLEX4110
      SAVMSZ(IFP)=0.0                                                   LLEX4120
      SAVDSX(NNN)=0.0                                                   LLEX4130
      SAVDSX(NNN+1)=0.0                                                 LLEX4140
      SAVDSY(NNN)=0.0                                                   LLEX4150
      SAVDSY(NNN+1)=0.0                                                 LLEX4160
      SAVDSZ(NNN)=0.0                                                   LLEX4170
      SAVDSZ(NNN+1)=0.0                                                 LLEX4180
  430 CONTINUE                                                          LLEX4190
      INDIFP = 1                                                        LLEX4200
C                                                                       LLEX4210
C          SET INDFPD TO INDICATE THE HIGHEST FOOTPAD STATUS            LLEX4220
C                                                                       LLEX4230
      INDFPD = INDFPC(1)                                                LLEX4240
      DO  435  IFP = 2,6                                                LLEX4250
  435 INDFPD  = MIN0( INDFPD, INDFPC(IFP) )                             LLEX4260
C                                                                       LLEX4270
C                UPDATE THE MASS MATRIX WITH THE ELASTIC MODE           LLEX4280
C                     CONTRIBUTIONS                                     LLEX4290
      IF ( NMODES .EQ. 0 ) GO TO 480                                    LLEX4300
      DO  450 I = 1,NMODES                                              LLEX4310
      AME2(1)=AME2(1) + ( PYZ2(I)+WNX(I)*GC(1,I))*GC(1,I)               LLEX4320
      AME2(2)=AME2(2) + ( PXZ2(I)+WNY(I)*GC(1,I))*GC(1,I)               LLEX4330
  450 AME2(3)=AME2(3) + ( PXY2(I)+WNZ(I)*GC(1,I))*GC(1,I)               LLEX4340
      DO  470 I = 1, 3                                                  LLEX4350
      J   = I + 3                                                       LLEX4360
  470 AM(J,J) = AM(J,J) - ( AME1(I) - AME2(I))                          LLEX4370
  480 CONTINUE                                                          LLEX4380
      IF ( INDFPD .GT. 1 ) GO TO 540                                    LLEX4390
C                                                                       LLEX4400
C                PROCESS THE NONCONTACTING FOOTPAD MASS CONTRIBUTIONS   LLEX4410
C                                                                       LLEX4420
      DO  490 I = 1, 3                                                  LLEX4430
```

```
      DO  490 J = 1, 3                                                  LLEX4440
      K   = J+3                                                         LLEX4450
      AM(I,K) = 0.0                                                     LLEX4460
  490 AM(K,I) = 0.0                                                     LLEX4470
      DO  530 IFP=1,NFTPDS                                              LLEX4480
      IF ( INDFPC(IFP) .GT. 0  ) GO TO 520                              LLEX4490
      DO  500  I = 1, 3                                                 LLEX4500
C               SET UPPER-RIGHT HAND CORNER OF THE MASS MATRIX          LLEX4510
      AM(I,4)=AM(I,4)+ AMFP(IFP)*(-DC(I,2)*ZFP(IFP)+DC(I,3)*YFP(IFP) )  LLEX4520
      AM(I,5)=AM(I,5)+ AMFP(IFP)*( DC(I,1)*ZFP(IFP)-DC(I,3)*XFP(IFP) )  LLEX4530
  500 AM(I,6)=AM(I,6)+ AMFP(IFP)*(-DC(I,1)*YFP(IFP)+DC(I,2)*XFP(IFP) )  LLEX4540
C               SET LOWER-LEFT HAND CORNER OF THE MASS MATRIX           LLEX4550
      AM(5,1) = AM(5,1) + AMFP(IFP) * ZFPS(1,IFP)                       LLEX4560
      AM(6,1) = AM(6,1) - AMFP(IFP) * YFPS(1,IFP)                       LLEX4570
      AM(6,2) = AM(6,2) + AMFP(IFP) * XFPS(1,IFP)                       LLEX4580
      IF (INDFPC(IFP).LT. 0 ) GO TO 530                                 LLEX4590
      INDFPC(IFP) = -1                                                  LLEX4600
      AK = AMFP(IFP)                                                    LLEX4610
  510 A   = YFP(IFP)*YFP(IFP)                                           LLEX4620
      B   = ZFP(IFP)*ZFP(IFP)                                           LLEX4630
      C   = XFP(IFP)*XFP(IFP)                                           LLEX4640
      AM(1,1) = AM(1,1) + AK                                            LLEX4650
      AM(4,4) = AM(4,4) + AK*( A + B )                                  LLEX4660
      AM(5,5) = AM(5,5) + AK*( C + B )                                  LLEX4670
      AM(6,6) = AM(6,6) + AK*( C + A )                                  LLEX4680
      AM(5,4) = AM(5,4) - AK* XFP(IFP) * YFP(IFP)                       LLEX4690
      AM(6,4) = AM(6,4) - AK* XFP(IFP) * ZFP(IFP)                       LLEX4700
      AM(6,5) = AM(6,5) - AK* YFP(IFP) * ZFP(IFP)                       LLEX4710
      GO TO 530                                                         LLEX4720
  520 IF ( INDFPC(IFP) .GT. 1 ) GO TO 530                               LLEX4730
      AK =  -AMFP(IFP)                                                  LLEX4740
      INDFPC(IFP) = 2                                                   LLEX4750
      GO TO 510                                                         LLEX4760
  530 CONTINUE                                                          LLEX4770
      AM(2,2) = AM(1,1)                                                 LLEX4780
      AM(3,3) = AM(1,1)                                                 LLEX4790
      AM(4,5) = AM(5,4)                                                 LLEX4800
      AM(4,6) = AM(6,4)                                                 LLEX4810
      AM(5,6) = AM(6,5)                                                 LLEX4820
      AM(4,2) = -AM(5,1)                                                LLEX4830
      AM(4,3) = -AM(6,1)                                                LLEX4840
      AM(5,3) = -AM(6,2)                                                LLEX4850
C                                                                       LLEX4860
C          IF NECESSARY, DEFIND THE MASS MATRIX INVERSE                 LLEX4870
C                                                                       LLEX4880
      CALL MATINV( AM(1,1), AMINV(1,1), 6 )                             LLEX4890
  540 CONTINUE                                                          LLEX4900
C                                                                       LLEX4910
C          SET UP THE FORCE/TORQUE ARRAY                                LLEX4920
C                                                                       LLEX4930
C               LEG FORCE/TORQUE EFFECTS                                LLEX4940
      DO   600  I = 1, 6                                                LLEX4950
  600 FTS(I) =FLTL(I)                                                   LLEX4960
C               PLANET GRAVITY EFFECTS                                  LLEX4970
      FTS(1) = FTS(1) - GCOSZT* AM(1,1)                                 LLEX4980
      FTS(3) = FTS(3) + GSINZT* AM(1,1)                                 LLEX4990
```

```
      A     = WX * WX
      B     = WY * WY
      C     = WZ * WZ
      IF ( INDFAR .EQ. 3 ) GO TO 610
      FO(1)= -WX*(WZ*CBIXY-WY*CBIXZ)-(C-B)*CBIYZ-WY*WZ*CIZZYY
      FO(2)= -WX*(WX*CBIYZ-WZ*CBIXY)-(A-C)*CBIXZ-WX*WZ*CIXXZZ
      FO(3)= -WZ*(WY*CBIXZ-WX*CBIYZ)-(B-A)*CBIXY-WX*WY*CIYYXX
C               RIGID BODY INERTIA LOAD EFFECTS
      FTS(4) = FTS(4)+FO(1)
      FTS(5) = FTS(5)+FO(2)
      FTS(6) = FTS(6)+FO(3)
  610 CONTINUE
      IF ( NMODES .EQ. 0 ) GO TO 640
C               INERTIA LOADS DUE TO ELASTIC MODES
      DO  625  I = 1,6
  625 FE(I)  = 0.0
      DO  630  I = 1, NMODES
      FE(1) = FE(1)+ ( PYZ(I)+WNX(I)*GC(1,I) )*GCD(1,I)
      FE(2) = FE(2)+ ( PXZ(I)+WNY(I)*GC(1,I) )*GCD(1,I)
      FE(3) = FE(3)+ ( PXY(I)+WNZ(I)*GC(1,I) )*GCD(1,I)
      FE(4) = FE(4)+ (PYNZ2(I)+ WNZNY(I)*GC(1,I) )*GC (1,I)
      FE(5) = FE(5)+ (PZNX2(I)+ WNXNZ(I)*GC(1,I) )*GC (1,I)
  630 FE(6) = FE(6)+ (PXNY2(I)+ WNYNX(I)*GC(1,I) )*GC (1,I)
      FE(4) = FE(4) + WX*( FE(1) + FE(1) )
      FE(5) = FE(5) + WY*( FE(2) + FE(2) )
      FE(6) = FE(6) + WZ*( FE(3) + FE(3) )
      FTS(4) = FTS(4) - FE(4)
      FTS(5) = FTS(5) - FE(5)
      FTS(6) = FTS(6) - FE(6)
  640 CONTINUE
C
C          CALCULATE THE CENTER BODY ACCELERATIONS AND ANGULAR RATES OF
C               CHANGE
      DO  670  I= 1, 6
      AHOLD(I) = 0.0
      DO  670  J= 1, 6
  670 AHOLD(I) = AHOLD(I)+ AMINV(I,J)*FTS(J)
      XSDD   = AHOLD(1)
      YSDD   = AHOLD(2)
      ZSDD   = AHOLD(3)
      WXD    = AHOLD(4)
      WYD    = AHOLD(5)
      WZD    = AHOLD(6)
      IF(INDFXD.NE.0)XSDD=0.0
      IF(INDFYD.NE.0)YSDD=0.0
      IF(INDFZD.NE.0)ZSDD=0.0
      IF(INDFXR.NE.0)WXD=0.0
      IF(INDFYR.NE.0)WYD=0.0
      IF(INDFZR.NE.0)WZD=0.0
      IF ( NMODES .EQ. 0 ) GO TO 750
C               DRAG STRUT
C               TOTAL GENERAL FORCES
C               ELASTIC-ROTATIONAL COUPLINGS (GFO)
C               MAIN STRUT
      DO  730  I = 1, NMODES
      GFO(I) = A*( PYZ(I)+WNX(I)*GC(1,I))+ B*(PXZ(I)+WNY(I)*GC(1,I))
```

```
     1                                    + C*(PXY(I)+WNZ(I)*GC(1,I))   LLEX5560
      GF(I) = GFO(I)+GF(I)                                              LLEX5570
C                                                                       LLEX5580
  730 GCDD(1,I)=(GF(I)-W2GM(I)*GC(1,I) ) / GM(I)                        LLEX5590
  750 CONTINUE                                                          LLEX5600
      IF ( INDFXD .NE. 0 ) GO TO 1000                                   LLEX5610
      CALL INTEG ( XSDD, XSD )                                          LLEX5620
      CALL INTEG ( XSD , XS  )                                          LLEX5630
 1000 IF ( INDFYD .NE. 0 ) GO TO 1010                                   LLEX5640
      CALL INTEG ( YSDD, YSD )                                          LLEX5650
      CALL INTEG ( YSD , YS  )                                          LLEX5660
 1010 IF ( INDFZD .NE. 0 ) GO TO 1020                                   LLEX5670
      CALL INTEG ( ZSDD, ZSD )                                          LLEX5680
      CALL INTEG ( ZSD , ZS  )                                          LLEX5690
 1020 IF ( INDFXR .NE. 0 ) GO TO 1030                                   LLEX5700
      CALL INTEG ( WXD , WX  )                                          LLEX5710
      CALL INTEG ( PHID, PHI )                                          LLEX5720
 1030 IF ( INDFYR .NE. 0 ) GO TO 1040                                   LLEX5730
      CALL INTEG ( WYD , WY    )                                        LLEX5740
      CALL INTEG ( THTAD, THTA )                                        LLEX5750
 1040 IF ( INDFZR .NE. 0 ) GO TO 1050                                   LLEX5760
      CALL INTEG ( WZD , WZ    )                                        LLEX5770
      CALL INTEG ( PSID,  PSI  )                                        LLEX5780
 1050 CONTINUE                                                          LLEX5790
      IF ( NMODES .EQ. 0 ) GO TO 1070                                   LLEX5800
      DO  1060 I= 1, NMODES                                             LLEX5810
      CALL INTEG ( GCDD(1,I), GCD(1,I) )                                LLEX5820
 1060 CALL INTEG ( GCD (1,I), GC (1,I) )                                LLEX5830
 1070 CONTINUE                                                          LLEX5840
C                    ***************************************************LLEX5850
      IF ( IPTATL .EQ. 0 ) GO TO 2010                                   LLEX5860
C                                                                       LLEX5870
C          CHECK FOR A VARIABLE PAST OR AT IT*S CUTOFF VALUE.           LLEX5880
C                                                                       LLEX5890
      CALL CUT                                                          LLEX5900
C                         JCUT VARIABLE AT CUTOFF                       LLEX5910
      IF ( JCUT .GT. 0 ) GO TO 3000                                     LLEX5920
      IF ( JCUT .EQ. 0 ) GO TO 2010                                     LLEX5930
C                         JCUT VARIABLE PAST IT*S CUTOFF VALUE          LLEX5940
      DO 2005 I = 1,45                                                  LLEX5950
 2005 TIMHSA(I) = TIMHSC(I)                                             LLEX5960
      DO 2006 I = 52, LTIMHS                                            LLEX5970
 2006 TIMHSA(I) = TIMHSC(I)                                             LLEX5980
      INDBPU = 1                                                        LLEX5990
C                                                                       LLEX6000
C          RESET FOOTPAD CONTACT INDICATOR IF THEY WERE JUST CHANGED    LLEX6010
C               AFTER THE CALL TO UPDATE.                               LLEX6020
C                                                                       LLEX6030
      IF ( INDIVR .EQ. 0 ) GO TO 1                                      LLEX6040
      DO  2007  I = 1, INDIVR                                           LLEX6050
      IFP = INOUTV(I)                                                   LLEX6060
 2007 INDFPC(IFP) = 2                                                   LLEX6070
      INDIVR = 0                                                        LLEX6080
      GO TO 1                                                           LLEX6090
 2010 CONTINUE                                                          LLEX6100
C                    ***************************************************LLEX6110
```

```
      IF ( IFIN .NE. 0 ) GO TO 2000
      IF ( IVARH .EQ. 1 ) GO TO 1910
      IF ( IVAL  .LT. 0 ) GO TO 2000
 1910 IF ( INDIVR .EQ. 0 ) GO TO 1920
      DO  424    IK= 1, INDIVR
      IFP  = INOUTV(IK)
C         THIS FOOTPAD HAS JUST LEFT THE LANDING SURFACE.  REMOVE
C               ITS  DISPLACEMENT, VELOCITY, AND ACCELATION VARIABLES
C               FROM THE INTEGRATION LIST
      J   = INDFPI(IFP)
      K   = J+9
      IF ( K .GT. IPTOTL )  GO TO 421
      DO  420  I = K, IPTOTL
      LOCNAM( J ) =  LOCNAM(I)
  420 J   = J + 1
  421 CONTINUE
      IPTOTL = IPTOTL - 9
C         SET XFPS AS A CUTOFF VARIABLE
      CALL LOC(140+4*IFP,ATTH(IFP)+(.5+GRAVE*2.5E-06)*CUTERR,1)
C
C         UPDATE THE INTEGRATION VARIABLE POINTERS
C
      DO 423 I = 1, NFTPDS
      IF ( INDFPC(I) .LE. 0 ) GO TO 423
      IF ( INDFPI(I).GT. INDFPI(IFP)) INDFPI(I) = INDFPI(I) - 9
  423 CONTINUE
      IF ( IK .EQ. INDIVR ) GO TO 425
      K = IK +1
      DO 422 I = K, INDIVR
      J = INOUTV(I)
      IF ( INDFPI(J) .GT. INDFPI(IFP)) INDFPI(J) = INDFPI(J) - 9
  422 CONTINUE
  425 CONTINUE
C         UPDATE THE CUTOFF VARIABLE POINTER
      INDFPI(IFP) = IPTATL
  424 CONTINUE
C                   ****************************************************
 1920 CONTINUE
C         IPRNT .EQ. 0 EACH TIME THE VARIABLE (TIME) IS A MULTIPLE
C               OF HMAX.
      IF (IPRNT .NE. 0 ) GO TO 2000
      KPTCNT = KPTCNT + 1
      IF(IFPPRT.EQ.0)GO TO 1999
      IPRINT=IPRTFP(1)
      DO 1998 I=1,NFTPDS
 1998 IF(IPRTFP(I).LT.IPRINT)IPRINT=IPRTFP(I)
 1999 CONTINUE
      IF(IFPPRT.EQ.0)IPRINT=IPTCNT
      IF(KPTCNT.LT.IPRINT)GO TO 2000
      KPTCNT = 0
      TIMTST = TIME
C
      CALL OUTPUT
      IF ( HZ .LT. 1.E-32 ) STOP
C
C         CHECK FOR TIME MAX.
```

```
      IF ( TIME.GE. TIMAX ) GO TO 3110                                  LLEX6680
C         CHECK FOR STABILITY                                           LLEX6690
      IF(ISTAB.NE.0)GO TO 3110                                          LLEX6700
C                   ****************************************************LLEX6710
 2000 CONTINUE                                                          LLEX6720
      IF ( IFIN  .NE. 0 ) GO TO 2060                                    LLEX6730
      IF ( IVARH .EQ. 1 ) GO TO 2060                                    LLEX6740
      IF ( IVAL  .GE. 0 ) GO TO 2060                                    LLEX6750
C         IF THE INTEGRATION IS GOING TO BE BACKED UP TO REDUCE THE     LLEX6760
C              STEP SIZE, THEN THE TIME HISTORY COMMON MUST BE BACKED   LLEX6770
C              UP AT THIS TIME                                          LLEX6780
 2020 CONTINUE                                                          LLEX6790
      DO 2030 I = 1,45                                                  LLEX6800
 2030 TIMHSA(I) = TIMHSC(I)                                             LLEX6810
      DO 2031 I = 52, LTIMHS                                            LLEX6820
 2031 TIMHSA(I) = TIMHSC(I)                                             LLEX6830
      INDBPU = 1                                                        LLEX6840
C         RESET FOOTPAD CONTACT INDICATOR IF THEY WERE JUST CHANGED     LLEX6850
C              AFTER THE CALL TO UPDATE.                                LLEX6860
      IF ( INDIVR .EQ. 0 ) GO TO 2060                                   LLEX6870
      DO  2037 I = 1, INDIVR                                            LLEX6880
      IFP = INOUTV(I)                                                   LLEX6890
 2037 INDFPC(IFP) = 2                                                   LLEX6900
      INDIVR = 0                                                        LLEX6910
C                                                                       LLEX6920
C              CALL INTEGRATION UPDATE                                  LLEX6930
C                                                                       LLEX6940
C                   ****************************************************LLEX6950
 2060 CALL UPDAT                                                        LLEX6960
         IF ( IFIN .NE. 0 ) GO TO 1                                     LLEX6970
      INDIVR=0                                                          LLEX6980
      DO 2070 IFP = 1, NFTPDS                                           LLEX6990
      IF ( INDFPC(IFP) .LE. 0 ) GO TO 2070                              LLEX7000
C         CHECK TO SEE IF FOOT PAD HAS LEFT THE GROUND                  LLEX7010
      IF(XFPS(1,IFP).LT.ATTH(IFP)+2.*CUTERR) GO TO 2070                 LLEX7020
C                                                                       LLEX7030
C         SAVE AND UPDATE INDICATORS FOR INTEGRATION VARIABLES TO BE    LLEX7040
C              REMOVE FROM THE INTEGRATION LIST IF THIS STEP            LLEX7050
C              IS ACCEPT BY THE CUTOFF ROUTINE AND THE VARIABLE STEP    LLEX7060
C              SIZE OPTION                                              LLEX7070
C                                                                       LLEX7080
      INDIVR = INDIVR + 1                                               LLEX7090
      INOUTV(INDIVR) = IFP                                              LLEX7100
C         SET  THE CONTACT INDICATOR                                    LLEX7110
      INDFPC(IFP)= 0                                                    LLEX7120
C         RECALCULATE THE F.P. LANDER COORDINATES                       LLEX7130
C              LANDER COORDINATES = DC**(-1)*(SURFACE COORDINATES F.P.  LLEX7140
C                                            - SURFACE COORDINATES C.G.) LLEX7150
      A   = XFPS(1,IFP) - XS                                            LLEX7160
      B   = YFPS(1,IFP) - YS                                            LLEX7170
      C   = ZFPS(1,IFP) - ZS                                            LLEX7180
      XFP(IFP)=DC(1,1)*A + DC(2,1)*B + DC(3,1)*C                        LLEX7190
      YFP(IFP)=DC(1,2)*A + DC(2,2)*B + DC(3,2)*C                        LLEX7200
      ZFP(IFP)=DC(1,3)*A + DC(2,3)*B + DC(3,3)*C                        LLEX7210
 2070 CONTINUE                                                          LLEX7220
      IF ( NMODES .EQ. 0 ) GO TO 2090                                   LLEX7230
```

```
      DO 2080 I = 1, 3                                                  LLEX7240
 2080 AME1(I) = AME2(I)                                                 LLEX7250
 2090 CONTINUE                                                          LLEX7260
      DO 2095 I = 1, LTIMHS                                             LLEX7270
      IF ( INDBPU .EQ. 1 ) GO TO 2095                                   LLEX7280
      TIMHSC(I)  = TIMHSB(I)                                            LLEX7290
 2095 TIMHSB(I) = TIMHSA(I)                                             LLEX7300
      INDBPU = 0                                                        LLEX7310
      GO TO 1                                                           LLEX7320
C                   ****************************************************LLEX7330
C                                                                       LLEX7340
C         MAKE SURE THIS TIME POINT IS PRINTED                          LLEX7350
C                                                                       LLEX7360
 3000 IF ( TIMTST .NE. TIME ) CALL OUTPUT                               LLEX7370
      TIMTST = TIME                                                     LLEX7380
      I =(IAD(JCUT) -140)                                               LLEX7390
C         FIND THE SUBSCRIPT OF THE FOOT PAD                            LLEX7400
C         ADD X-Y-ZFPS(I/4), X-Y-ZFPSD(I/4), AND X-Y-ZFPSDD(I/4)        LLEX7410
C              TO INTEGRATION LIST                                      LLEX7420
      CALL INUPD( 140 + I )                                             LLEX7430
      J = I/4                                                           LLEX7440
C                                                                       LLEX7450
C         UPDATE THE CUTOFF VARIABLE POINTERS                           LLEX7460
C                                                                       LLEX7470
      DO 3005 K = 1, NFTPDS                                             LLEX7480
      IF ( INDFPC(K) .GT. 0 ) GO TO 3005                                LLEX7490
      IF ( INDFPI(K) .GT. INDFPI(J)) INDFPI(K) = INDFPI(K) - 1          LLEX7500
 3005 CONTINUE                                                          LLEX7510
C         UPDATE THE INTEGRATION VARIABLE POINTER                       LLEX7520
      INDFPI(  J ) = IPTOTL                                             LLEX7530
      CALL INUPD( 160 + I )                                             LLEX7540
      CALL INUPD( 180 + I )                                             LLEX7550
      CALL INUPD( 200 + I )                                             LLEX7560
      CALL INUPD( 220 + I )                                             LLEX7570
      CALL INUPD( 240 + I )                                             LLEX7580
      CALL INUPD( 260 + I )                                             LLEX7590
      CALL INUPD( 280 + I )                                             LLEX7600
      CALL INUPD( 300 + I )                                             LLEX7610
      A    =  WY * ZFP(J)  -  WZ * YFP(J)                               LLEX7620
      B    =  WZ * XFP(J)  -  WX * ZFP(J)                               LLEX7630
      C    =  WX * YFP(J)  -  WY * XFP(J)                               LLEX7640
      XFPSD(1,J) = XSD + DC(1,1)*A + DC(1,2)*B + DC(1,3)*C              LLEX7650
      YFPSD(1,J) = YSD + DC(2,1)*A + DC(2,2)*B + DC(2,3)*C              LLEX7660
      ZFPSD(1,J) = ZSD + DC(3,1)*A + DC(3,2)*B + DC(3,3)*C              LLEX7670
      INDFPC( J )    = 1                                                LLEX7680
      I = JCUT + 1                                                      LLEX7690
      IF ( I .GT. IPTATL ) GO TO 3015                                   LLEX7700
      DO 3010  K=  I ,IPTATL                                            LLEX7710
      IAD(JCUT) = IAD(K)                                                LLEX7720
      XSI(JCUT) = XSI(K)                                                LLEX7730
      IND(JCUT) = IND(K)                                                LLEX7740
      XMN1(JCUT) = XMN1(K)                                              LLEX7750
 3010 JCUT = JCUT + 1                                                   LLEX7760
 3015 CONTINUE                                                          LLEX7770
      IPTATL = IPTATL-1                                                 LLEX7780
C                                                                       LLEX7790
```

```
C         CHECK FOR SIMULTANEOUS CUTOFF VARIABLES                       LLEX7800
C                                                                       LLEX7810
      IF ( IPTATL .EQ. 0 ) GO TO 3030                                   LLEX7820
      CALL CUT                                                          LLEX7830
      IF ( JCUT .GT. 0 ) GO TO 3000                                     LLEX7840
      IF ( JCUT .LT. 0 ) GO TO 3200                                     LLEX7850
C                                                                       LLEX7860
C         RESET THE INTEGRATION PRINT AND STEP SIZE CONDITIONS          LLEX7870
C                                                                       LLEX7880
 3030 CONTINUE                                                          LLEX7890
      CALL SETUP                                                        LLEX7900
      DO 3040 I = 1, 6                                                  LLEX7910
 3040 TIMHSA(I) = TIMHSB(I)                                             LLEX7920
      DO 3045 I = 43, 45                                                LLEX7930
 3045 TIMHSA(I) = TIMHSB(I)                                             LLEX7940
      DO 3050 I = 58, LTIMHS                                            LLEX7950
 3050 TIMHSA(I) = TIMHSB(I)                                             LLEX7960
      DO 3060 I = 1, LTIMHS                                             LLEX7970
 3060 TIMHSC(I) = TIMHSA(I)                                             LLEX7980
      INDBPU = 1                                                        LLEX7990
      GO TO 1                                                           LLEX8000
C                   ****************************************************LLEX8010
C                                                                       LLEX8020
C     END OF CASE                                                       LLEX8030
C                                                                       LLEX8040
 3110 CONTINUE                                                          LLEX8050
C                                                                       LLEX8060
      IF(IBOTM.NE.0)CALL OUTPUT                                         LLEX8070
C                                                                       LLEX8080
      CALL SUMMRY                                                       LLEX8090
C                                                                       LLEX8100
      RETURN                                                            LLEX8110
 3200 CONTINUE                                                          LLEX8120
      WRITE (6,3210)                                                    LLEX8130
 3210 FORMAT(52H0TWO VARIABLES PASSED CUTOFF AT THE SAME TIME POINT,    LLEX8140
     1       44H AND BOTH COULD NOT BE SUCCESSFULLY CUT BACK )          LLEX8150
      CALL OUTPUT                                                       LLEX8160
 9000 RETURN                                                            LLEX8170
      END                                                               LLEX8180
```

```
      SUBROUTINE FTPAD                                                  FTPD  10
C                                                                       FTPD  20
C     THIS SUBROUTINE CONTAINS THE LOGIC FOR DETERMINING THE SOIL       FTPD  30
C     FORCES AND LANDING GEAR STRUT FORCES                              FTPD  40
C                                                                       FTPD  50
C     FOLLOWING NOTATION USED -                                         FTPD  60
C         S.C.S. - SURFACE COORDINATE SYSTEM                            FTPD  70
C         L.C.S. - LANDER COORDINATE SYSTEM                             FTPD  80
C                                                                       FTPD  90
C     THE FOLLOWING INFORMATION IS RETURNED FOR THE LANDING GEAR UNDER  FTPD 100
C     CONSIDERATION.                                                    FTPD 110
C         STRUT FORCES ON CENTER BODY IN S.C.S.                         FTPD 120
C         STRUT FORCES ON FOOTPAD IN S.C.S.                             FTPD 130
C         STRUT TORQUES ON CENTER BODY IN L.C.S.                        FTPD 140
C         GENERALIZED FORCES                                            FTPD 150
C                                                                       FTPD 160
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),                       FTPD 170
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),                     FTPD 180
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)                    FTPD 190
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)                            FTPD 200
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),     FTPD 210
     1    INDFPC(6),                                                    FTPD 220
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),   FTPD 230
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),   FTPD 240
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),   FTPD 250
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5), FTPD 260
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),   FTPD 270
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)                            FTPD 280
      COMMON COMINT(400)                                                FTPD 290
      EQUIVALENCE ( COMINT(   3 ), IBOTM      )                         FTPD 300
      EQUIVALENCE ( COMINT(   4 ), XSD    )                             FTPD 310
      EQUIVALENCE ( COMINT(  12 ), YS     )                             FTPD 320
      EQUIVALENCE ( COMINT(  16 ), YSD    )                             FTPD 330
      EQUIVALENCE ( COMINT(  24 ), ZS     )                             FTPD 340
      EQUIVALENCE ( COMINT(  28 ), ZSD    )                             FTPD 350
      EQUIVALENCE ( COMINT(  44 ), WX     )                             FTPD 360
      EQUIVALENCE ( COMINT(  60 ), WY     )                             FTPD 370
      EQUIVALENCE ( COMINT(  76 ), WZ     )                             FTPD 380
      EQUIVALENCE ( COMINT(  84 ), GC     )                             FTPD 390
      EQUIVALENCE ( COMINT(104 ), GCD    )                              FTPD 400
      EQUIVALENCE ( COMINT(124 ), GCDD   )                              FTPD 410
      EQUIVALENCE ( COMINT(144 ), XFPS   )                              FTPD 420
      EQUIVALENCE ( COMINT(164 ), XFPSD  )                              FTPD 430
      EQUIVALENCE ( COMINT(184 ), XFPSDD )                              FTPD 440
      EQUIVALENCE ( COMINT(204 ), YFPS   )                              FTPD 450
      EQUIVALENCE ( COMINT(224 ), YFPSD  )                              FTPD 460
      EQUIVALENCE ( COMINT(244 ), YFPSDD )                              FTPD 470
      EQUIVALENCE ( COMINT(264 ), ZFPS   )                              FTPD 480
      EQUIVALENCE ( COMINT(284 ), ZFPSD  )                              FTPD 490
      EQUIVALENCE ( COMINT(304 ), ZFPSDD )                              FTPD 500
      EQUIVALENCE ( COMINT( 324), TIME   )                              FTPD 510
      EQUIVALENCE ( COMINT( 343 ), IFIN   )                             FTPD 520
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )                             FTPD 530
      EQUIVALENCE ( COMINT( 364 ), XS      )                            FTPD 540
C                   COMINT(364-367) USED BY XS                          FTPD 550
```

```
      EQUIVALENCE ( COMINT( 368 ), CURDT   )                            FTPD 560
      COMMON                                                            FTPD 570
     1     CBMASS, CBIXX  , CBIXZ  , CBIYY  , CBIYZ  , CBIZZ  , FPMASS,CBIXY, FTPD 580
     2     DC(3,3)        , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),     FTPD 590
     3     WNZ(5), PX(5)  , PY(5)  , PZ(5)  , GM(5)  , OMEGA(5)      ,  FTPD 600
     4     GRAV   , GRAVE , ZETA   , FTS (6)        , FSXSI(5)       ,  FTPD 610
     5     FSYSI(5)       , FSZSI(5)       , SOILX(5)       ,           FTPD 620
     6     SOILY(5)       , SOILZ(5)       , PMSX(5,5)      ,           FTPD 630
     7     PMSY(5,5)      , PMSZ(5,5)      , PDSX(10,5)     ,           FTPD 640
     8     PDSY(10,5)     , PDSZ(10,5)     , FLXS    , FLYS , FLZS   ,  FTPD 650
     9     TLXL   ,  TLYL   , TLZL   , SLO    , XMSCB(5)      ,         FTPD 660
     C     YMSCB(5)       , ZMSCB(5)       , XDSCB(10)      ,           FTPD 670
     D     YDSCB(10)      , ZDSCB(10)      , ILEG    , IMS   ,          FTPD 680
     E     FSTX   , FSTY   , FSTZ   , PVCBX   , PVCBY   , PVCBZ   ,     FTPD 690
     F     PVFPX , PVFPY  , PVFPZ  , NOLEG   , SLOMS(5) ,               FTPD 700
     G     SLODS(10)                                                    FTPD 710
      COMMON                                                            FTPD 720
     1     PFCMS(5)       , PFCDS(5)       , PFTDS(5)       ,           FTPD 730
     2     PFTMS(5)       , SRCDS(5)       , SRCMS(5)       ,           FTPD 740
     3     SRTDS(5)   ,     SRTMS(5)       , COEFDS, COEFMS, GAMDS ,    FTPD 750
     4     GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,      FTPD 760
     5     STMXDS, STMXMS, CDCDS(5)        , CDCMS(5)       ,           FTPD 770
     6     CDTDS(5)       , CDTMS(5)       , FRICDS, FRICMS, IRETDS,    FTPD 780
     7     IRETMS, STROKE(10)     , STRKDS(10)     , STRKMS( 5)     ,   FTPD 790
     8     LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),   FTPD 800
     9     ATTPRS(3)      , ADIST , SOILP(3)       , NMODES,            FTPD 810
     A     COEF   , GAMMA , FRIC   , SCMAX , STMAX , CDC(5), CDT(5),    FTPD 820
     B     SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),   FTPD 830
     C     CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,      FTPD 840
     D     INDFZR, TIMAX , DRAGST, IFP                                  FTPD 850
      COMMON                                                            FTPD 860
     1CMS,CDCONT,SOILNU,SLRHO,       NOOUT,XOUT(10),YOUT(10),ZOUT(10),  FTPD 870
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),       FTPD 880
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),       FTPD 890
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,         FTPD 900
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)          FTPD 910
      COMMON                                                            FTPD 920
     1     SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),               FTPD 930
     2     SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)               FTPD 940
     3    ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,               FTPD 950
     4     IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ         FTPD 960
C                                                                       FTPD 970
C     SUBROUTINE INITIALIZATION                                         FTPD 980
C                                                                       FTPD 990
      IF(NMODES.EQ.0)GO TO 11                                           FTPD1000
      DO 10 I=1,NMODES                                                  FTPD1010
      GFLEGS(I)=0.0                                                     FTPD1020
   10 CONTINUE                                                          FTPD1030
   11 CONTINUE                                                          FTPD1040
C                                                                       FTPD1050
C     *  *  *  *  *                                                     FTPD1060
C     DETERMINE LANDING GEAR STRUT LOADS                                FTPD1070
C     *  *  *  *  *                                                     FTPD1080
C                                                                       FTPD1090
C     *  *  *  *  *                                                     FTPD1100
C     MAIN STRUT                                                        FTPD1110
```

```
C     *   *   *   *   *
C
      IMS=1
C
C     CENTER BODY STRUT ATTACH POINT IN L.C.S.
C
      XT=XMSCB(IFP)
      YT=YMSCB(IFP)
      ZT=ZMSCB(IFP)
      IF(NMODES.EQ.0)GO TO 101
      DO 100 I=1,NMODES
      XT=XT+PMSX(IFP,I)*GC(1,I)
      YT=YT+PMSY(IFP,I)*GC(1,I)
  100 ZT=ZT+PMSZ(IFP,I)*GC(1,I)
  101 CONTINUE
      XTMS=XT
      YTMS=YT
      ZTMS=ZT
C
C     CENTER BODY STRUT ATTACH POINT IN S.C.S.
C
      PVCBX=XS+DC(1,1)*XT+DC(1,2)*YT+DC(1,3)*ZT
      PVCBY=YS+DC(2,1)*XT+DC(2,2)*YT+DC(2,3)*ZT
      PVCBZ=ZS+DC(3,1)*XT+DC(3,2)*YT+DC(3,3)*ZT
C
C     FOOTPAD END STRUT ATTACH POINT IN S.C.S.
C
      PVFPX = XFPS(1,IFP)
      PVFPY = YFPS(1,IFP)
      PVFPZ = ZFPS(1,IFP)
C
C     SET UP LOAD-STROKE CURVE
C
      DO 115 I=1,5
      PFC(I)=PFCMS(I)
      PFT(I)=PFTMS(I)
      SRC(I)=SRCMS(I)
      SRT(I)=SRTMS(I)
      CDC(I)=CDCMS(I)
      CDT(I)=CDTMS(I)
  115 CONTINUE
      COEF=COEFMS
      GAMMA=GAMMS
      SRULC=SRUCMS
      SRULT=SRUTMS
      SCMAX=SCMXMS
      STMAX=STMXMS
      FRIC=FRICMS
      SLO=SLOMS(IFP)
C
C     SET UP INDICATORS
C
      STRP=   STRPMS(IFP)
      SPNGC=  URCMS(IFP)
      SPNGT=  URTMS(IFP)
      DISTC=  SETCMS(IFP)
```

```
      DISTT=  SETTMS(IFP)
      PFORC=  PRFCMS(IFP)
      PFORT=  PRFTMS(IFP)
      FOREVC= FRVMSC(IFP)
      FOREVT= FRVMST(IFP)
      IPOSC=  IPOCMS(IFP)
      IPOST=  IPOTMS(IFP)
      INDULC= INDCMS(IFP)
      INDULT= INDTMS(IFP)
      IPREV=  IPRMS(IFP)
      IRET=IRETMS
C
      CALL STRUT(STRP,SPNGC,SPNGT,DISTC,DISTT,PFORC,PFORT,FOREVC,
     *           FOREVT,IPOSC,IPOST,INDULC,INDULT,IPREV,IRET,IFP)
C
C     SAVE INDICATORS
C
      IF(IBOTM.NE.0) RETURN
      STRKMS(IFP)=STROKE(IFP)
      FORMS(IFP)=FORCE
      IF(TIME.EQ.0.0) GO TO 117
      IF(IFIN.NE.0) GO TO 117
      STRPMS(IFP)=STROKE(IFP)
      SLNGMS(IFP)=CURMSL
      URCMS(IFP)    =SPNGC
      URTMS(IFP)   =SPNGT
      SETCMS(IFP)   =DISTC
      SETTMS(IFP)   =DISTT
      PRFCMS(IFP)   =PFORC
      PRFTMS(IFP)   =PFORT
      FRVMSC(IFP)   =FOREVC
      FRVMST(IFP)   =FOREVT
      IPOCMS(IFP)   =IPOSC
      IPOTMS(IFP)   =IPOST
      INDCMS(IFP)   =INDULC
      INDTMS(IFP)   =INDULT
      IPRMS(IFP)    =IPREV
C     SAVE MAXIMUM STROKE
      IF(STROKE(IFP).GT.SMXMSC(IFP))GO TO 118
      SMXMSC(IFP)=STROKE(IFP)
      TMXMSC(IFP)=TIME
      GO TO 117
  118 IF(STROKE(IFP).LT.SMXMST(IFP))GO TO 117
      SMXMST(IFP)=STROKE(IFP)
      TMXMST(IFP)=TIME
  117 CONTINUE
C
C     SAVE FORCES AND TORQUES
C
      FX=-DC(1,1)*FSTX-DC(2,1)*FSTY-DC(3,1)*FSTZ
      FY=-DC(1,2)*FSTX-DC(2,2)*FSTY-DC(3,2)*FSTZ
      FZ=-DC(1,3)*FSTX-DC(2,3)*FSTY-DC(3,3)*FSTZ
      IF(NFORC.EQ.0) GO TO 116
      SAVMSX(IFP)=FX
      SAVMSY(IFP)=FY
      SAVMSZ(IFP)=FZ
```

```
  116 CONTINUE                                                          FTPD2240
      FSXSI(IFP)=FSTX                                                   FTPD2250
      FSYSI(IFP)=FSTY                                                   FTPD2260
      FSZSI(IFP)=FSTZ                                                   FTPD2270
      FLXS=-FSTX                                                        FTPD2280
      FLYS=-FSTY                                                        FTPD2290
      FLZS=-FSTZ                                                        FTPD2300
      TLXL=YT*FZ-ZT*FY                                                  FTPD2310
      TLYL=ZT*FX-XT*FZ                                                  FTPD2320
      TLZL=XT*FY-YT*FX                                                  FTPD2330
      IF(NMODES.EQ.0)GO TO 121                                          FTPD2340
      DO 120 I=1,NMODES                                                 FTPD2350
  120 GFLEGS(I)=GFLEGS(I)+FX*PMSX(IFP,I)+FY*PMSY(IFP,I)+FZ*PMSZ(IFP,I)  FTPD2360
  121 CONTINUE                                                          FTPD2370
C                                                                       FTPD2380
C     *   *   *   *   *                                                 FTPD2390
C     DRAG STRUT                                                        FTPD2400
C     *   *   *   *   *                                                 FTPD2410
C                                                                       FTPD2420
      IMS=0                                                             FTPD2430
      DO 2500 III=1,2                                                   FTPD2440
      NNN=2*(IFP-1)+III                                                 FTPD2450
      SRBEND=0.                                                         FTPD2460
C                                                                       FTPD2470
C     CENTER BODY STRUT ATTACH POINT IN L.C.S.                          FTPD2480
C                                                                       FTPD2490
      XT=XDSCB(NNN)                                                     FTPD2500
      YT=YDSCB(NNN)                                                     FTPD2510
      ZT=ZDSCB(NNN)                                                     FTPD2520
      IF(NMODES.EQ.0)GO TO 2101                                         FTPD2530
      DO 2100 I=1,NMODES                                                FTPD2540
      XT=XT+PDSX(NNN,I)*GC(1,I)                                         FTPD2550
      YT=YT+PDSY(NNN,I)*GC(1,I)                                         FTPD2560
 2100 ZT=ZT+PDSZ(NNN,I)*GC(1,I)                                         FTPD2570
 2101 CONTINUE                                                          FTPD2580
C                                                                       FTPD2590
C     CENTER BODY ATTACH POINT IN S.C.S.                                FTPD2600
C                                                                       FTPD2610
      PVCBX=XS+DC(1,1)*XT+DC(1,2)*YT+DC(1,3)*ZT                         FTPD2620
      PVCBY=YS+DC(2,1)*XT+DC(2,2)*YT+DC(2,3)*ZT                         FTPD2630
      PVCBZ=ZS+DC(3,1)*XT+DC(3,2)*YT+DC(3,3)*ZT                         FTPD2640
      IF(ILEG.EQ.0)GO TO 2000                                           FTPD2650
C                                                                       FTPD2660
C     CANTILEVER GEAR                                                   FTPD2670
C                                                                       FTPD2680
C     FOOTPAD END STRUT ATTACH POINT IN S.C.S.                          FTPD2690
C                                                                       FTPD2700
      DEL=CURMSL-DRAGST                                                 FTPD2710
      IF(DEL.LT.0.0) DEL=0.0                                            FTPD2720
      PVFPX = XFPS(1,IFP)+CDXS*DEL                                      FTPD2730
      PVFPY = YFPS(1,IFP)+CDYS*DEL                                      FTPD2740
      PVFPZ = ZFPS(1,IFP)+CDZS*DEL                                      FTPD2750
C                                                                       FTPD2760
C     MAIN STRUT BENDING                                                FTPD2770
C                                                                       FTPD2780
      IF(AEI1.LE.0.0.OR.AEI2.LE.0.0) GO TO 2000                         FTPD2790
```

```
      DX=PVFPX-PVCBX                                                    FTPD2800
      DY=PVFPY-PVCBY                                                    FTPD2810
      DZ=PVFPZ-PVCBZ                                                    FTPD2820
      SSL=SQRT(DX*DX+DY*DY+DZ*DZ)                                       FTPD2830
      SX=DX/SSL                                                         FTPD2840
      SY=DY/SSL                                                         FTPD2850
      SZ=DZ/SSL                                                         FTPD2860
      SDOTM=CDXS*SX+CDYS*SY+CDZS*SZ                                     FTPD2870
      SDOTN=1.0-SDOTM**2                                                FTPD2950
      SN2=SDOTN**2                                                      FTPD2960
      D2=DRAGST**2                                                      FTPD2970
      D3=DRAGST**3                                                      FTPD2980
      AAA=(D3*DEL)/(3*CURMSL)*(1.-DRAGST/CURMSL)                        FTPD2990
      BBB=D2*(D2/CURMSL-D3/(3.*CURMSL**2)+CURMSL/3.-DRAGST)             FTPD3000
      SRBEND=1./(SN2*((AAA/AEI1)+(BBB/AEI2)))                           FTPD3010
 2000 CONTINUE                                                          FTPD3020
C                                                                       FTPD3030
C     SET UP LOAD-STROKE CURVE                                          FTPD3040
C                                                                       FTPD3050
      DO 2115 I=1,5                                                     FTPD3060
      PFC(I)=PFCDS(I)                                                   FTPD3070
      PFT(I)=PFTDS(I)                                                   FTPD3080
      IF(ILEG.EQ.0) GO TO 2114                                          FTPD3090
      IF(AEI1.LE.0.0.OR.AEI2.LE.0.0) GO TO 2114                         FTPD3100
      SRC(I)=(SRBEND*SRCDS(I))/(SRBEND+SRCDS(I))                        FTPD3110
      SRT(I)=(SRBEND*SRTDS(I))/(SRBEND+SRTDS(I))                        FTPD3120
      CDC(I)=CDCDS(I)+PFCDS(I)/SRBEND                                   FTPD3130
      CDT(I)=CDTDS(I)+PFTDS(I)/SRBEND                                   FTPD3140
      GO TO 2115                                                        FTPD3150
 2114 SRC(I)=SRCDS(I)                                                   FTPD3160
      SRT(I)=SRTDS(I)                                                   FTPD3170
      CDC(I)=CDCDS(I)                                                   FTPD3180
      CDT(I)=CDTDS(I)                                                   FTPD3190
 2115 CONTINUE                                                          FTPD3200
      COEF=COEFDS                                                       FTPD3210
      GAMMA=GAMDS                                                       FTPD3220
      SRULC=SRUCDS                                                      FTPD3230
      SRULT=SRUTDS                                                      FTPD3240
      SCMAX=SCMXDS                                                      FTPD3250
      STMAX=STMXDS                                                      FTPD3260
      FRIC=FRICDS                                                       FTPD3270
      SLO=SLODS(NNN)                                                    FTPD3280
C                                                                       FTPD3290
C     SET UP INDICATORS                                                 FTPD3300
C                                                                       FTPD3310
      STRP=   STRPDS(NNN)                                               FTPD3320
      SPNGC=  URCDS(NNN)                                                FTPD3330
      SPNGT=  URTDS(NNN)                                                FTPD3340
      DISTC=  SETCDS(NNN)                                               FTPD3350
      DISTT=  SETTDS(NNN)                                               FTPD3360
      PFORC=  PRFCDS(NNN)                                               FTPD3370
      PFORT=  PRFTDS(NNN)                                               FTPD3380
      FOREVC= FRVDSC(NNN)                                               FTPD3390
      FOREVT= FRVDST(NNN)                                               FTPD3400
      IPOSC=  IPOCDS(NNN)                                               FTPD3410
      IPOST= IPOTDS(NNN)                                                FTPD3420
```

```
      INDULC= INDCDS(NNN)                                               FTPD3430
      INDULT= INDTDS(NNN)                                               FTPD3440
      IPREV=  IPRDS(NNN)                                                FTPD3450
      IRET=IRETDS                                                       FTPD3460
C                                                                       FTPD3470
      CALL STRUT(STRP,SPNGC,SPNGT,DISTC,DISTT,PFORC,PFORT,FOREVC,       FTPD3480
     *           FOREVT,IPOSC,IPOST,INDULC,INDULT,IPREV,IRET,NNN)       FTPD3490
C                                                                       FTPD3500
C     SAVE INDICATORS                                                   FTPD3510
C                                                                       FTPD3520
      IF(IBOTM.NE.0)IFP=NNN*100                                         FTPD3530
      IF(IBOTM.NE.0) RETURN                                             FTPD3540
      STRKDS(NNN)=STROKE(NNN)                                           FTPD3550
      FORDS(NNN)=FORCE                                                  FTPD3560
      IF(TIME.EQ.0.0) GO TO 2127                                        FTPD3570
      IF(IFIN.NE.0) GO TO 2127                                          FTPD3580
      STRPDS(NNN)=STROKE(NNN)                                           FTPD3590
      SLNGDS(NNN)=CURDSL                                                FTPD3600
      URCDS(NNN)    =SPNGC                                              FTPD3610
      URTDS(NNN)   =SPNGT                                               FTPD3620
      SETCDS(NNN)   =DISTC                                              FTPD3630
      SETTDS(NNN)   =DISTT                                              FTPD3640
      PRFCDS(NNN)   =PFORC                                              FTPD3650
      PRFTDS(NNN)   =PFORT                                              FTPD3660
      FRVDSC(NNN)   =FOREVC                                             FTPD3670
      FRVDST(NNN)   =FOREVT                                             FTPD3680
      IPOCDS(NNN)   =IPOSC                                              FTPD3690
      IPOTDS(NNN)   =IPOST                                              FTPD3700
      INDCDS(NNN)   =INDULC                                             FTPD3710
      INDTDS(NNN)   =INDULT                                             FTPD3720
      IPRDS(NNN)    =IPREV                                              FTPD3730
C     SAVE MAXUMUM STROKE                                               FTPD3740
      IF(STROKE(NNN).GT.SMXDSC(NNN))GO TO 2128                          FTPD3750
      SMXDSC(NNN)=STROKE(NNN)                                           FTPD3760
      TMXDSC(NNN)=TIME                                                  FTPD3770
      GO TO 2127                                                        FTPD3780
 2128 IF(STROKE(NNN).LT.SMXDST(NNN))GO TO 2127                          FTPD3790
      SMXDST(NNN)=STROKE(NNN)                                           FTPD3800
      TMXDST(NNN)=TIME                                                  FTPD3810
 2127 CONTINUE                                                          FTPD3820
      IF(ILEG.EQ.0)GO TO 2119                                           FTPD3830
C                                                                       FTPD3840
C     SAVE FORCES AND TORQUES - CANTILEVER                              FTPD3850
C                                                                       FTPD3860
      FX=-DC(1,1)*FSTX-DC(2,1)*FSTY-DC(3,1)*FSTZ                        FTPD3870
      FY=-DC(1,2)*FSTX-DC(2,2)*FSTY-DC(3,2)*FSTZ                        FTPD3880
      FZ=-DC(1,3)*FSTX-DC(2,3)*FSTY-DC(3,3)*FSTZ                        FTPD3890
      IF(NFORC.EQ.0) GO TO 2118                                         FTPD3900
      SAVDSX(NNN)=FX                                                    FTPD3910
      SAVDSY(NNN)=FY                                                    FTPD3920
      SAVDSZ(NNN)=FZ                                                    FTPD3930
 2118 CONTINUE                                                          FTPD3940
      FLXS=FLXS-FSTX                                                    FTPD3950
      FLYS=FLYS-FSTY                                                    FTPD3960
      FLZS=FLZS-FSTZ                                                    FTPD3970
      TLXL=TLXL+YT*FZ-ZT*FY                                             FTPD3980
```

```
      TLYL=TLYL+ZT*FX-XT*FZ                                             FTPD3990
      TLZL=TLZL+XT*FY-YT*FX                                             FTPD4000
      IF(NMODES.EQ.0)GO TO 2117                                         FTPD4010
      DO 2116 I=1,NMODES                                                FTPD4020
 2116 GFLEGS(I)=GFLEGS(I)+FX*PDSX(NNN,I)+FY*PDSY(NNN,I)+FZ*PDSZ(NNN,I)  FTPD4030
 2117 CONTINUE                                                          FTPD4040
      FA=FSTX*CDXS+FSTY*CDYS+FSTZ*CDZS                                  FTPD4050
      FAX=FA*CDXS                                                       FTPD4060
      FAY=FA*CDYS                                                       FTPD4070
      FAZ=FA*CDZS                                                       FTPD4080
      FLX=FSTX-FAX                                                      FTPD4090
      FLY=FSTY-FAY                                                      FTPD4100
      FLZ=FSTZ-FAZ                                                      FTPD4110
      RAT2=(CURMSL-DRAGST)/CURMSL                                       FTPD4120
      IF(RAT2.LT.0.0)RAT2=0.0                                           FTPD4130
      RAT1=1.-RAT2                                                      FTPD4140
      FSXSI(IFP)=FSXSI(IFP)+RAT1*FLX                                    FTPD4150
      FSYSI(IFP)=FSYSI(IFP)+RAT1*FLY                                    FTPD4160
      FSZSI(IFP)=FSZSI(IFP)+RAT1*FLZ                                    FTPD4170
      FXX=FAX+RAT2*FLX                                                  FTPD4180
      FYY=FAY+RAT2*FLY                                                  FTPD4190
      FZZ=FAZ+RAT2*FLZ                                                  FTPD4200
      FX=DC(1,1)*FXX+DC(1,2)*FYY+DC(1,3)*FZZ                            FTPD4210
      FZ=DC(3,1)*FXX+DC(3,2)*FYY+DC(3,3)*FZZ                            FTPD4220
      FY=DC(2,1)*FXX+DC(2,2)*FYY+DC(2,3)*FZZ                            FTPD4230
      IF(NFORC.EQ.0) GO TO 2122                                         FTPD4240
      SAVMSX(IFP)=SAVMSX(IFP)+FX                                        FTPD4250
      SAVMSY(IFP)=SAVMSY(IFP)+FY                                        FTPD4260
      SAVMSZ(IFP)=SAVMSZ(IFP)+FZ                                        FTPD4270
 2122 CONTINUE                                                          FTPD4280
      FLXS=FLXS+FXX                                                     FTPD4290
      FLYS=FLYS+FYY                                                     FTPD4300
      FLZS=FLZS+FZZ                                                     FTPD4310
      TLXL=TLXL+YTMS*FZ-ZTMS*FY                                         FTPD4320
      TLYL=TLYL+ZTMS*FX-XTMS*FZ                                         FTPD4330
      TLZL=TLZL+XTMS*FY-YTMS*FX                                         FTPD4340
      IF(NMODES.EQ.0)GO TO 2201                                         FTPD4350
      DO 2200 I=1,NMODES                                                FTPD4360
 2200 GFLEGS(I)=GFLEGS(I)+FX*PMSX(IFP,I)+FY*PMSY(IFP,I)+FZ*PMSZ(IFP,I)  FTPD4370
 2201 CONTINUE                                                          FTPD4380
      GO TO 2500                                                        FTPD4390
C                                                                       FTPD4400
C     SAVE FORCES AND TORQUES - INVERTED TRIPOD                         FTPD4410
C                                                                       FTPD4420
 2119 CONTINUE                                                          FTPD4430
      FX=-DC(1,1)*FSTX-DC(2,1)*FSTY-DC(3,1)*FSTZ                        FTPD4440
      FY=-DC(1,2)*FSTX-DC(2,2)*FSTY-DC(3,2)*FSTZ                        FTPD4450
      FZ=-DC(1,3)*FSTX-DC(2,3)*FSTY-DC(3,3)*FSTZ                        FTPD4460
      IF(NFORC.EQ.0) GO TO 2123                                         FTPD4470
      SAVDSY(NNN)=FY                                                    FTPD4480
      SAVDSZ(NNN)=FZ                                                    FTPD4490
      SAVDSX(NNN)=FX                                                    FTPD4500
 2123 CONTINUE                                                          FTPD4510
      FSXSI(IFP)=FSXSI(IFP)+FSTX                                        FTPD4520
      FSYSI(IFP)=FSYSI(IFP)+FSTY                                        FTPD4530
      FSZSI(IFP)=FSZSI(IFP)+FSTZ                                        FTPD4540
```

```
      FLXS=FLXS-FSTX                                                    FTPD4550
      FLYS=FLYS-FSTY                                                    FTPD4560
      FLZS=FLZS-FSTZ                                                    FTPD4570
      TLXL=TLXL+YT*FZ-ZT*FY                                             FTPD4580
      TLYL=TLYL+ZT*FX-XT*FZ                                             FTPD4590
      TLZL=TLZL+XT*FY-YT*FX                                             FTPD4600
      IF(NMODES.EQ.0)GO TO 2121                                         FTPD4610
      DO 2120 I=1,NMODES                                                FTPD4620
 2120 GFLEGS(I)=GFLEGS(I)+FX*PDSX(NNN,I)+FY*PDSY(NNN,I)+FZ*PDSZ(NNN,I)  FTPD4630
 2121 CONTINUE                                                          FTPD4640
 2500 CONTINUE                                                          FTPD4650
C                                                                       FTPD4660
C     *  *  *  *  *                                                     FTPD4670
C     DETERMINE SOIL FORCES                                             FTPD4680
C     *  *  *  *  *                                                     FTPD4690
 3000 CONTINUE                                                          FTPD4700
      IF(XFPS(1,IFP)-ATTH(IFP).GT.0.0)GO TO 5000                        FTPD4710
C                                                                       FTPD4720
C     FOOTPAD IMPACT PRINT CONTROL                                      FTPD4730
C                                                                       FTPD4740
      IF(IFPPRT.EQ.0)GO TO 4010                                         FTPD4750
      IF(IFIN.NE.0)GO TO 4010                                           FTPD4760
      IF(IPTCNT.EQ.1)GO TO 4010                                         FTPD4770
      IF(IMPACT(IFP).NE.0)GO TO 4000                                    FTPD4780
      IMPACT(IFP)=1                                                     FTPD4790
      IPRTFP(IFP)=1                                                     FTPD4800
 4000 CONTINUE                                                          FTPD4810
      IF(IMPACT(IFP).EQ.2)GO TO 4010                                    FTPD4820
      KOUNT(IFP)=KOUNT(IFP)+1                                           FTPD4830
      IF(KOUNT(IFP).LE.IFPPRT)GO TO 4010                                FTPD4840
      IPRTFP(IFP)=IPTCNT                                                FTPD4850
      KOUNT(IFP)=0                                                      FTPD4860
      IMPACT(IFP)=2                                                     FTPD4870
 4010 CONTINUE                                                          FTPD4880
C                                                                       FTPD4890
      CALL SOIL                                                         FTPD4900
C                                                                       FTPD4910
C     CHECK MAGNITUDE OF IN PLANE SOIL FORCE                            FTPD4920
C                                                                       FTPD4930
      FF=SQRT(SOILY(IFP)**2+SOILZ(IFP)**2)                              FTPD4940
      IF(FF.LT.0.00001) GO TO 3200                                      FTPD4950
      CFFY=SOILY(IFP)/FF                                                FTPD4960
      CFFZ=SOILZ(IFP)/FF                                                FTPD4970
      FH=SQRT(FSYSI(IFP)**2+FSZSI(IFP)**2)                              FTPD4980
      IF(FH.GT.0.00001) GO TO 3102                                      FTPD4990
      HDOTF=0.0                                                         FTPD5000
      GO TO 3103                                                        FTPD5010
 3102 CFHY=FSYSI(IFP)/FH                                                FTPD5020
      CFHZ=FSZSI(IFP)/FH                                                FTPD5030
      HDOTF=CFHY*CFFY+CFHZ*CFFZ                                         FTPD5040
 3103 VS=SQRT(YFPSD(1,IFP)**2+ZFPSD(1,IFP)**2)                          FTPD5050
      AMV=FPMASS*VS/CURDT                                               FTPD5060
      IF(FF.LE.(-FH*HDOTF+AMV)) GO TO 3200                              FTPD5070
      IF(HDOTF.LE.0.)FF=-FH*HDOTF+AMV                                   FTPD5080
      IF(HDOTF.GT.0.)FF=AMV                                             FTPD5090
      IF(FF.LT.0.)FF=0.0                                                FTPD5100
```

```
      SOILY(IFP)=CFFY*FF                                                FTPD5110
      SOILZ(IFP) =CFFZ*FF                                               FTPD5120
 3200 CONTINUE                                                          FTPD5130
      RETURN                                                            FTPD5140
C                                                                       FTPD5150
C     *    *    *    *    *                                             FTPD5160
C     FOOTPAD OFF GROUND                                                FTPD5170
C     *    *    *    *    *                                             FTPD5180
C                                                                       FTPD5190
 5000 CONTINUE                                                          FTPD5200
      SOILX(IFP)=0.0                                                    FTPD5210
      SOILY(IFP)=0.0                                                    FTPD5220
      SOILZ(IFP)=0.0                                                    FTPD5230
C                                                                       FTPD5240
C     FOOTPAD IMPACT PRINT CONTROL                                      FTPD5250
C                                                                       FTPD5260
      IF(IFPPRT.EQ.0)GO TO 5020                                         FTPD5270
      IF(IFIN.NE.0)GO TO 5020                                           FTPD5280
      IF(IMPACT(IFP).NE.1)GO TO 5010                                    FTPD5290
      KOUNT(IFP)=KOUNT(IFP)+1                                           FTPD5300
      IF(KOUNT(IFP).LE.IFPPRT)GO TO 5020                                FTPD5310
      IPRTFP(IFP)=IPTCNT                                                FTPD5320
      KOUNT(IFP)=0                                                      FTPD5330
 5010 IMPACT(IFP)=0                                                     FTPD5340
 5020 CONTINUE                                                          FTPD5350
      RETURN                                                            FTPD5360
      END                                                               FTPD5370
```

```
      SUBROUTINE GEOM
C     THIS SUBROUITNE DETERMINES THE DIRECTION COSINE MATRIX AND
C     DETERMINES THE TIME DERIVATES OF THE EULER ANGLES
      COMMON COMINT(400)
      EQUIVALENCE ( COMINT( 36 ), PHI      )
      EQUIVALENCE ( COMINT( 40 ), PHID     )
      EQUIVALENCE ( COMINT( 44 ), WX       )
      EQUIVALENCE ( COMINT( 52 ), THTA     )
      EQUIVALENCE ( COMINT( 56 ), THTAD    )
      EQUIVALENCE ( COMINT( 60 ), WY       )
      EQUIVALENCE ( COMINT( 68 ), PSI      )
      EQUIVALENCE ( COMINT( 72 ), PSID     )
      EQUIVALENCE ( COMINT( 76 ), WZ       )
      COMMON
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY,
     2    DC(3,3)
      COSPHI = COS( PHI  )
      SINPHI = SIN( PHI  )
      COSTHA = COS( THTA )
      SINTHA = SIN( THTA )
      COSPSI = COS( PSI  )
      SINPSI = SIN( PSI  )
      DC(1,1) = COSTHA * COSPSI
      A       = SINTHA * COSPSI
      B       = COSPHI * SINPSI
      DC(1,2) = SINPHI * A - B
      C       = SINPHI * SINPSI
      DC(1,3) = COSPHI * A + C
      DC(2,1) = COSTHA * SINPSI
      DC(2,2) = SINTHA * C + COSPHI*COSPSI
      DC(2,3) = SINTHA * B - SINPHI*COSPSI
      DC(3,1) = - SINTHA
      DC(3,2) = SINPHI*COSTHA
      DC(3,3) = COSPHI*COSTHA
C                    **********
      IF(ABS(COSTHA)-0.1E-10) 100,100,200
  200 CONTINUE
      PSID  =(WZ*COSPHI + WY*SINPHI) / COSTHA
      PHID  = WX + SINTHA* PSID
      THTAD = WY*COSPHI - WZ*SINPHI
      RETURN
  100 CONTINUE
      PSID = 0.0
      PHID = WX
      THTAD = SQRT(WY*WY+WZ*WZ)
      RETURN
      END
```

```
      SUBROUTINE STRUT(STRP,SPNGC,SPNGT,DISTC,DISTT,PFORC,PFORT,FOREVC, STRU  10
     1                 FOREVT,IPOSC,IPOST,INDULC,INDULT,IPREV,          STRU  20
     2      IRET,NNN )                                                  STRU  30
C                                                                       STRU  40
C     THIS SUBROUTINE DETERMINES THE FORCES IN A TYPICAL                STRU  50
C     LANDING GEAR STRUT                                                STRU  60
C                                                                       STRU  70
C     THE STRUT FORCES ACTING ON THE FOOTPAD IN THE SURFACE             STRU  80
C     COORDINATE SYSTEM ARE RETURNED                                    STRU  90
C                                                                       STRU 100
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)                            STRU 110
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),                       STRU 120
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),                     STRU 130
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)                    STRU 140
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),     STRU 150
     1    INDFPC(6),                                                    STRU 160
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),   STRU 170
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),   STRU 180
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),   STRU 190
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5), STRU 200
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),   STRU 210
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)                            STRU 220
      COMMON COMINT(400)                                                STRU 230
      EQUIVALENCE ( COMINT(  3 ), IBOTM      )                          STRU 240
      EQUIVALENCE ( COMINT( 84 ), GC     )                              STRU 250
      EQUIVALENCE ( COMINT(104 ), GCD    )                              STRU 260
      EQUIVALENCE ( COMINT(124 ), GCDD   )                              STRU 270
      EQUIVALENCE ( COMINT(144 ), XFPS   )                              STRU 280
      EQUIVALENCF ( COMINT(164 ), XFPSD  )                              STRU 290
      EQUIVALENCF ( COMINT(184 ), XFPSDD )                              STRU 300
      EQUIVALENCE ( COMINT(204 ), YFPS   )                              STRU 310
      EQUIVALENCE ( COMINT(224 ), YFPSD  )                              STRU 320
      EQUIVALENCE ( COMINT(244 ), YFPSDD )                              STRU 330
      EQUIVALENCE ( COMINT(264 ), ZFPS   )                              STRU 340
      EQUIVALENCE ( COMINT(284 ), ZFPSD  )                              STRU 350
      EQUIVALENCE ( COMINT(304 ), ZFPSDD )                              STRU 360
      EQUIVALENCE ( COMINT(368 ), CURDT  )                              STRU 370
      COMMON                                                            STRU 380
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY, STRU 390
     2    DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),       STRU 400
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)      ,       STRU 410
     4    GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)      ,       STRU 420
     5    FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,               STRU 430
     6    SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,               STRU 440
     7    PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,               STRU 450
     8    PDSY(10,5)    , PDSZ(10,5)    , FLXS   , FLYS , FLZS  ,       STRU 460
     9    TLXL  ,  TLYL  , TLZL  , SLO   , XMSCB(5)     ,               STRU 470
     C    YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,               STRU 480
     D    YDSCB(10)     , ZDSCB(10)     , ILEG   , IMS  ,               STRU 490
     E    FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ  ,            STRU 500
     F    PVFPX , PVFPY , PVFPZ , NOLEG  , SLOMS(5) ,                   STRU 510
     G    SLODS(10)                                                     STRU 520
      COMMON                                                            STRU 530
     1    PFCMS(5)      , PFCDS(5)      , PFTDS(5)      ,               STRU 540
     2    PFTMS(5)      , SRCDS(5)      , SRCMS(5)      ,               STRU 550
```

```
     3    SRTDS(5)    ,    SRTMS(5)       , COEFDS, COEFMS, GAMDS ,        STRU 560
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,          STRU 570
     5    STMXDS, STMXMS, CDCDS(5)        , CDCMS(5)        ,              STRU 580
     6    CDTDS(5)       , CDTMS(5)       , FRICDS, FRICMS, IRETDS,        STRU 590
     7    IRETMS, STROKE(10)     , STRKDS(10)     , STRKMS( 5)     ,       STRU 600
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),       STRU 610
     9    ATTPRS(3)      , ADIST , SOILP(3)       , NMODES,                STRU 620
     A    COEF   , GAMMA , FRIC   , SCMAX , STMAX , CDC(5), CDT(5),        STRU 630
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),       STRU 640
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,          STRU 650
     D    INDFZR, TIMAX , DRAGST, IFP                                      STRU 660
      COMMON                                                               STRU 670
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),      STRU 680
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),          STRU 690
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),          STRU 700
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,            STRU 710
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)             STRU 720
      COMMON                                                               STRU 730
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),                   STRU 740
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)                   STRU 750
     3   ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,                   STRU 760
     4    IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ             STRU 770
C                                                                          STRU 780
C     INITIALIZE SUBROUTINE                                                STRU 790
C                                                                          STRU 800
      DX=PVCBX-PVFPX                                                       STRU 810
      DY=PVCBY-PVFPY                                                       STRU 820
      DZ=PVCBZ-PVFPZ                                                       STRU 830
      SLNGTH=SQRT(DX*DX+DY*DY+DZ*DZ)                                       STRU 840
      CDX=DX/SLNGTH                                                        STRU 850
      CDY=DY/SLNGTH                                                        STRU 860
      CDZ=DZ/SLNGTH                                                        STRU 870
      STR=SLNGTH-SLO                                                       STRU 880
      ABSTR=ABS(STR)                                                       STRU 890
      VELST=(STR-STRP)/CURDT                                               STRU 900
      ABVEL=ABS(VELST)                                                     STRU 910
      IF(IMS.NE.1)GO TO 10                                                 STRU 920
      CDXS=CDX                                                             STRU 930
      CDYS=CDY                                                             STRU 940
      CDZS=CDZ                                                             STRU 950
      CURMSV=VELST                                                         STRU 960
      CURMSL=SLNGTH                                                        STRU 970
   10 CONTINUE                                                             STRU 980
      IF(IMS.EQ.0)CURDSL=SLNGTH                                            STRU 990
      IJKC=IPOSC                                                           STRU1000
      IJKT=IPOST                                                           STRU1010
C                                                                          STRU1020
C     DETERMINE STROKING DIRECTION                                         STRU1030
C                                                                          STRU1040
      IF(STR.LT.0.0)GO TO 500                                              STRU1050
      IF(STR.GT.0.0)GO TO 400                                              STRU1060
C                                                                          STRU1070
C     *  *  *  *  *                                                        STRU1080
C     ZERO STRUT STROKE                                                    STRU1090
C     *  *  *  *  *                                                        STRU1100
C                                                                          STRU1110
```

```
      FORCE=0.0                                                         STRU1120
      GO TO 2002                                                        STRU1130
C                                                                       STRU1140
C     *  *  *  *  *                                                     STRU1150
C     STRUT IN TENSION                                                  STRU1160
C     *  *  *  *  *                                                     STRU1170
C                                                                       STRU1180
  400 CONTINUE                                                          STRU1190
      IF(ABSTR.GT.STMAX)GO TO 5001                                      STRU1200
C                                                                       STRU1210
C     WAS PREVIOUS STEP ON COMPRESSION SIDE                             STRU1220
C                                                                       STRU1230
      IF(IPREV.EQ.0)GO TO 402                                           STRU1240
      IF(IPREV.NE.-1)GO TO 402                                          STRU1250
      IF(PFORC.EQ.0.0)GO TO 402                                         STRU1260
      IPREV=+1                                                          STRU1270
      IF(IRET.EQ.1) GO TO 651                                           STRU1280
      GO TO 626                                                         STRU1290
  402 CONTINUE                                                          STRU1300
      IPREV=+1                                                          STRU1310
      IF(ABSTR.GT.DISTT)GO TO 401                                       STRU1320
      FORCE=0.0                                                         STRU1330
      GO TO 2001                                                        STRU1340
  401 CONTINUE                                                          STRU1350
      IF(STR.LT.STRP)GO TO 425                                          STRU1360
C                                                                       STRU1370
C     STRUT LOADING                                                     STRU1380
C                                                                       STRU1390
      IF(SPNGT.EQ.0.0)GO TO 405                                         STRU1400
      FORCE=SPNGT*(ABSTR-DISTT)                                         STRU1410
      IF(FORCE.LT.FOREVT)GO TO 2001                                     STRU1420
      FOREVT=0.0                                                        STRU1430
      INDULT=0                                                          STRU1440
      SPNGT=0.0                                                         STRU1450
      IF(IJKT.EQ.1)GO TO 407                                            STRU1460
      DISTT=CDT(IJKT-1)-PFT(IJKT-1)/SRT(IJKT)                           STRU1470
      GO TO 408                                                         STRU1480
  407 DISTT=0.0                                                         STRU1490
  408 CONTINUE                                                          STRU1500
      IF (FORCE.LT.PFT(IJKT))GO TO 405                                  STRU1510
      FORCE=PFT(IJKT)                                                   STRU1520
      IF(ABSTR.LT.CDT(IJKT))GO TO 2001                                  STRU1530
  405 CONTINUE                                                          STRU1540
  406 IF(ABSTR.LT.CDT(IJKT))GO TO 410                                   STRU1550
      DISTT=CDT(IJKT)-PFT(IJKT)/SRT(IJKT+1)                             STRU1560
      IPOST=IPOST+1                                                     STRU1570
      IJKT=IPOST                                                        STRU1580
      GO TO 406                                                         STRU1590
  410 CONTINUE                                                          STRU1600
      FORCE=SRT(IJKT)*(ABSTR-DISTT)                                     STRU1610
      IF(FORCE.GT.PFT(IJKT))FORCE=PFT(IJKT)                             STRU1620
      GO TO 2001                                                        STRU1630
C                                                                       STRU1640
C     STRUT UNLOADING                                                   STRU1650
C                                                                       STRU1660
  425 CONTINUE                                                          STRU1670
```

```
      IF(IRET.EQ.1)GO TO 450                                            STRU1680
      IF(INDULT.NE.0)GO TO 475                                          STRU1690
  426 SPNGT=SRULT                                                       STRU1700
      IF(PFORT.NE.PFT(IPOST))SPNGT=SRT(IPOST)                           STRU1705
      FOREVT=PFORT                                                      STRU1710
      GO TO 470                                                         STRU1720
  450 CONTINUE                                                          STRU1730
      IF(INDULT.NE.0)GO TO 475                                          STRU1740
  451 CONTINUE                                                          STRU1750
      FOREVT=PFORT                                                      STRU1760
      IF(PFORT.EQ.PFT(IJKT))GO TO 465                                   STRU1770
      SPNGT=SRT(IJKT)                                                   STRU1780
      GO TO 470                                                         STRU1790
  465 CONTINUE                                                          STRU1800
      SPNGT=SRULT                                                       STRU1810
      IF((IJKT+1).GT.5)GO TO 470                                        STRU1820
      SPNGT=SRT(IJKT+1)                                                 STRU1830
  470 CONTINUE                                                          STRU1840
      INDULT=1                                                          STRU1850
      DISTT=ABS(STRP)-PFORT/SPNGT                                       STRU1860
      IF(IPREV.EQ.-1)GO TO 602                                          STRU1870
  475 CONTINUE                                                          STRU1880
      FORCE=SPNGT*(ABSTR-DISTT)                                         STRU1890
      IF(ABSTR.LT.DISTT)FORCE=0.0                                       STRU1900
      GO TO 2001                                                        STRU1910
C                                                                       STRU1920
C     *  *  *  *  *                                                     STRU1930
C     STRUT IN COMPRESSTION                                             STRU1940
C     *  *  *  *  *                                                     STRU1950
C                                                                       STRU1960
  500 CONTINUE                                                          STRU1970
      IF(ABSTR.GT.SCMAX)GO TO 5000                                      STRU1980
C                                                                       STRU1990
C     WAS PREVIOUS STEP ON TENSION SIDE                                 STRU2000
C                                                                       STRU2010
      IF(IPREV.EQ.0)GO TO 602                                           STRU2020
      IF(IPREV.NE.+1)GO TO 602                                          STRU2030
      IF(PFORT.EQ.0.0)GO TO 602                                         STRU2040
      IPREV=-1                                                          STRU2050
      IF(IRET.EQ.+1)GO TO 451                                           STRU2060
      GO TO 426                                                         STRU2070
  602 CONTINUE                                                          STRU2080
      IPREV=-1                                                          STRU2090
      IF(ABSTR.GT.DISTC)GO TO 600                                       STRU2100
      FORCE=0.0                                                         STRU2110
      GO TO 2000                                                        STRU2120
  600 CONTINUE                                                          STRU2130
      IF(STR.GT.STRP)GO TO 625                                          STRU2140
C                                                                       STRU2150
C     STRUT LOADING                                                     STRU2160
C                                                                       STRU2170
      IF(SPNGC.EQ.0.0)GO TO 605                                         STRU2180
      FORCE=SPNGC*(ABSTR-DISTC)                                         STRU2190
      IF(FORCE.LT.FOREVC)GO TO 2000                                     STRU2200
      FOREVC=0.0                                                        STRU2210
      INDULC=0                                                          STRU2220
```

```
      SPNGC=0.0                                                         STRU2230
      IF(IJKC.EQ.1)GO TO 607                                            STRU2240
      DISTC=CDC(IJKC-1)-PFC(IJKC-1)/SRC(IJKC)                           STRU2250
      GO TO 608                                                         STRU2260
  607 DISTC=0.0                                                         STRU2270
  608 CONTINUE                                                          STRU2280
      IF(FORCE.LT.PFC(IJKC))GO TO 605                                   STRU2290
      FORCE=PFC(IJKC)                                                   STRU2300
      IF(ABSTR.LT.CDC(IJKC))GO TO 2000                                  STRU2310
  605 CONTINUE                                                          STRU2320
  606 IF(ABSTR.LT.CDC(IJKC))GO TO 610                                   STRU2330
      DISTC=CDC(IJKC)-PFC(IJKC)/SRC(IJKC+1)                             STRU2340
      IPOSC=IPOSC+1                                                     STRU2350
      IJKC=IPOSC                                                        STRU2360
      GO TO 606                                                         STRU2370
  610 CONTINUE                                                          STRU2380
      FORCE=SRC(IJKC)*(ABSTR-DISTC)                                     STRU2390
      IF(FORCE.GT.PFC(IJKC))FORCE=PFC(IJKC)                             STRU2400
      GO TO 2000                                                        STRU2410
C                                                                       STRU2420
C     STRUT UNLOADING                                                   STRU2430
C                                                                       STRU2440
  625 CONTINUE                                                          STRU2450
      IF(IRET.EQ.1)GO TO 650                                            STRU2460
      IF(INDULC.NE.0)GO TO 675                                          STRU2470
  626 SPNGC=SRULC                                                       STRU2480
      IF(PFORC.NE.PFC(IPOSC))SPNGC=SRC(IPOSC)                           STRU2485
      FOREVC=PFORC                                                      STRU2490
      GO TO 670                                                         STRU2500
  650 CONTINUE                                                          STRU2510
      IF(INDULC.NE.0)GO TO 675                                          STRU2520
  651 CONTINUE                                                          STRU2530
      FOREVC=PFORC                                                      STRU2540
      IF(PFORC.EQ.PFC(IJKC))GO TO 665                                   STRU2550
      SPNGC=SRC(IJKC)                                                   STRU2560
      GO TO 670                                                         STRU2570
  665 CONTINUE                                                          STRU2580
      SPNGC=SRULC                                                       STRU2590
      IF((IJKC+1).GT.5) GO TO 670                                       STRU2600
      SPNGC=SRC(IJKC+1)                                                 STRU2610
  670 CONTINUE                                                          STRU2620
      INDULC=1                                                          STRU2630
      DISTC=ABS(STRP)-PFORC/SPNGC                                       STRU2640
      IF(IPREV.EQ.+1)GO TO 402                                          STRU2650
  675 CONTINUE                                                          STRU2660
      FORCE=SPNGC*(ABSTR-DISTC)                                         STRU2670
      IF(ABSTR.LT.DISTC)FORCE=0.0                                       STRU2680
C                                                                       STRU2690
C     UPDATE INDICATORS                                                 STRU2700
C                                                                       STRU2710
 2000 CONTINUE                                                          STRU2720
      PFORC=FORCE                                                       STRU2730
      GO TO 2005                                                        STRU2740
 2001 CONTINUE                                                          STRU2750
      PFORT=FORCE                                                       STRU2760
      GO TO 2005                                                        STRU2770
```

```
 2002 CONTINUE                                                          STRU2780
      PFORC=0.0                                                         STRU2790
      PFORT=0.0                                                         STRU2800
 2005 CONTINUE                                                          STRU2810
      STROKE(NNN)=STR                                                   STRU2820
C                                                                       STRU2830
C     DETERMINE FRICTION AND DAMPING FORCES                             STRU2840
C                                                                       STRU2850
      FRIFOR= SIGN(1.,VELST)*(FRIC+COEF*(ABVEL**GAMMA))                 STRU2860
C                                                                       STRU2870
C     TOTAL STRUT FORCE                                                 STRU2880
      FORCE=SIGN(1.,STR)*FORCE+FRIFOR                                   STRU2890
      FSTX=CDX*FORCE                                                    STRU2900
      FSTY=CDY*FORCE                                                    STRU2910
      FSTZ=CDZ*FORCE                                                    STRU2920
      RETURN                                                            STRU2930
C                                                                       STRU2940
C     STRUT BOTTOMED OUT ON COMPRESSION SIDE                            STRU2950
C                                                                       STRU2960
 5000 CONTINUE                                                          STRU2970
      IBOTM=-1                                                          STRU2980
      RETURN                                                            STRU2990
C                                                                       STRU3000
C     STRUT BOTTOMED OUT ON TENSION SIDE                                STRU3010
C                                                                       STRU3020
 5001 CONTINUE                                                          STRU3030
      IBOTM=+1                                                          STRU3040
      RETURN                                                            STRU3050
      END                                                               STRU3060
```

```
      SUBROUTINE SOIL
C
C     THIS SUBROUTINE DETERMINES THE SOIL FORCES ACTING ON
C     A FOOTPAD
C
C     THE EVALUATION OF THE FOOTPAD ATTENUATION SYSTEM FORCES
C     IS ALSO INCLUDED HERE
C
C     THE SOIL/ATTENUATION FORCES ACTING ON THE FOOTPAD IN THE
C     SURFACE COORDINATE SYSTEM ARE RETURNED
C
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),
     1            XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),
     2           XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)
C
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),
     1     INDFPC(6),
     2     STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),
     3     URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),
     4     SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),
     5   INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5),
     6     IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),
     L     FRVMST( 5), IPOTDS(10), IPOTMS( 5)
      COMMON COMINT(400)
      EQUIVALENCE ( COMINT( 84 ), GC      )
      EQUIVALENCE ( COMINT(104 ), GCD     )
      EQUIVALENCE ( COMINT(124 ), GCDD    )
      EQUIVALENCE ( COMINT(144 ), XFPS    )
      EQUIVALENCE ( COMINT(164 ), XFPSD   )
      EQUIVALENCE ( COMINT(184 ), XFPSDD  )
      EQUIVALENCE ( COMINT(204 ), YFPS    )
      EQUIVALENCE ( COMINT(224 ), YFPSD   )
      EQUIVALENCE ( COMINT(244 ), YFPSDD  )
      EQUIVALENCE ( COMINT(264 ), ZFPS    )
      EQUIVALENCE ( COMINT(284 ), ZFPSD   )
      EQUIVALENCE ( COMINT(304 ), ZFPSDD  )
      EQUIVALENCE ( COMINT(324 ), TIME    )
      EQUIVALENCE ( COMINT( 343), IFIN   )
      EQUIVALENCE ( COMINT(368 ), CURDT   )
      COMMON
     1     CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY,
     2     DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),
     3     WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)      ,
     4     GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)      ,
     5     FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,
     6     SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,
     7     PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,
     8     PDSY(10,5)    , PDSZ(10,5)    , FLXS   , FLYS , FLZS  ,
     9     TLXL  ,  TLYL  , TLZL  , SLO    , XMSCB(5)      ,
     C     YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,
     D     YDSCB(10)     , ZDSCB(10)     , ILFG   , IMS  ,
     E     FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ  ,
     F     PVFPX , PVFPY , PVFPZ , NOLFG  , SLOMS(5) ,
     G     SLODS(10)
```

```
      COMMON                                                            SOIL 560
     1    PFCMS(5)        , PFCDS(5)        , PFTDS(5)        ,         SOIL 570
     2    PFTMS(5)        , SRCDS(5)        , SRCMS(5)        ,         SOIL 580
     3    SRTDS(5)    ,     SRTMS(5)        , COEFDS, COEFMS, GAMDS ,   SOIL 590
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,       SOIL 600
     5    STMXDS, STMXMS, CDCDS(5)          , CDCMS(5)        ,         SOIL 610
     6    CDTDS(5)        , CDTMS(5)        , FRICDS, FRICMS, IRETDS,   SOIL 620
     7    IRETMS, STROKE(10)    , STRKDS(10)      , STRKMS( 5)    ,     SOIL 630
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),    SOIL 640
     9    ATTPRS(3)       , ADIST , SOILP(3)        , NMODES,           SOIL 650
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),       SOIL 660
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),    SOIL 670
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,       SOIL 680
     D    INDFZR, TIMAX , DRAGST, IFP                                   SOIL 690
      COMMON                                                            SOIL 700
     1CMS,CDCONT,SOILNU,SLRHO,       NOOUT,XOUT(10),YOUT(10),ZOUT(10),  SOIL 710
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),       SOIL 720
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),       SOIL 730
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,         SOIL 740
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)          SOIL 750
      COMMON                                                            SOIL 760
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),                SOIL 770
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)                SOIL 780
     3   ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,                SOIL 790
     4    IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ          SOIL 800
C                                                                       SOIL 810
C     SUBROUTINE INITIALIZATION                                         SOIL 820
C                                                                       SOIL 830
      KJI=1                                                             SOIL 840
      IJK=1                                                             SOIL 850
      PIE=3.14159265                                                    SOIL 860
      PIE2=PIE/2.                                                       SOIL 870
      DEPTH=XFPS(1,IFP)-ATTH(IFP)                                       SOIL 880
      ADEPTH=ABS(DEPTH)                                                 SOIL 890
      FPSTR=ATTH(IFP)-ADEPTH                                            SOIL 900
      IF(DEPTH.GE.0.0)GO TO 2000                                        SOIL 910
C                                                                       SOIL 920
C     FOOTPRINT AREA                                                    SOIL 930
C                                                                       SOIL 940
      IF(FPSTR.LE.SS(4)) GO TO 101                                      SOIL 950
  100 IJK=IJK+1                                                         SOIL 960
      IF(IJK.GT.4) GO TO 101                                            SOIL 970
      IF(FPSTR.LT.SS(IJK)) GO TO 100                                    SOIL 980
      IF(FPSTR.EQ.SS(IJK))GO TO 102                                     SOIL 990
      DRDD=(RAD(IJK)-RAD(IJK-1))/(SS(IJK-1)-SS(IJK))                    SOIL1000
      RR=RAD(IJK)-DRDD*(FPSTR-SS(IJK))                                  SOIL1010
      GO TO 104                                                         SOIL1020
  101 RR=RAD(4)                                                         SOIL1030
      GO TO 103                                                         SOIL1040
  102 RR=RAD(IJK)                                                       SOIL1050
  103 DRDD=0.0                                                          SOIL1060
  104 AREA=PIE*RR*RR                                                    SOIL1070
      IF(NTYPE.EQ.0) GO TO 500                                          SOIL1080
C                                                                       SOIL1090
C     *   *   *   *   *                                                 SOIL1100
C     SECONDARY SOIL MECHANICS                                          SOIL1110
```

```
C     *   *   *   *   *                                                     SOIL1120
C                                                                           SOIL1130
      PRESS=ADEPTH*SOILP(2)                                                 SOIL1140
      IF(PRESS.GT.SOILP(3)) PRESS=SOILP(3)                                  SOIL1150
      SOILX(IFP)=PRESS*AREA                                                 SOIL1160
C                                                                           SOIL1170
C     CHECK MAGNITUDE OF SOIL FORCE                                         SOIL1180
C                                                                           SOIL1190
      SOILCR=FPMASS*(GCOSZT-XFPSD(1,IFP)/CURDT)-FSXSI(IFP)                  SOIL1200
      IF(SOILCR.LT.0.0)SOILCR=0.0                                           SOIL1210
      IF(SOILX(IFP).GT.SOILCR)SOILX(IFP)=SOILCR                             SOIL1220
      PRESS=SOILX(IFP)/AREA                                                 SOIL1230
  150 CONTINUE                                                              SOIL1240
      VBAR=SQRT(YFPSD(1,IFP)**2+ZFPSD(1,IFP)**2)                            SOIL1250
      IF(VBAR.LT.0.000001)GO TO 200                                         SOIL1260
      COEF=SOILP(1)                                                         SOIL1270
      CY=COEF*YFPSD(1,IFP)/VBAR                                             SOIL1280
      CZ=COEF*ZFPSD(1,IFP)/VBAR                                             SOIL1290
      GO TO 210                                                             SOIL1300
  200 CY=0.0                                                                SOIL1310
      CZ=0.0                                                                SOIL1320
  210 CONTINUE                                                              SOIL1330
      SOILY(IFP)=-CY*SOILX(IFP)                                             SOIL1340
      SOILZ(IFP)=-CZ*SOILX(IFP)                                             SOIL1350
      GO TO 1000                                                            SOIL1360
C                                                                           SOIL1370
C     *   *   *   *   *                                                     SOIL1380
C     PRIMARY SOIL MECHANICS                                                SOIL1390
C     *   *   *   *   *                                                     SOIL1400
C                                                                           SOIL1410
  500 CONTINUE                                                              SOIL1420
      VBAR=SQRT(XFPSD(1,IFP)**2+YFPSD(1,IFP)**2+ZFPSD(1,IFP)**2)            SOIL1430
      VHOR=SQRT(YFPSD(1,IFP)**2+ZFPSD(1,IFP)**2)                            SOIL1440
      THETAL=0.                                                             SOIL1450
      IF(VBAR.GT.0.000001)THETAL=ASIN(VHOR/VBAR)                            SOIL1460
      IF(XFPSD(1,IFP).GT.0.)THETAL=PIE-THETAL                               SOIL1470
      COST=ABS(COS(THETAL))                                                 SOIL1480
      SINT=SIN(THETAL)                                                      SOIL1490
      ATHICK=RR*ADEPTH                                                      SOIL1500
      ATHTA=AREA*COST+ATHICK*SINT                                           SOIL1510
      APERP=AREA*SINT+ATHICK*COST                                           SOIL1520
      FPHI=1.-2.*THETAL/PIE                                                 SOIL1530
      ALAM  =.25*(APERP/ATHTA)*(1.-EXP(-50.*THETAL))*(1.+SINT)              SOIL1540
      IF(THETAL.LT.PIE2) GO TO 505                                          SOIL1550
      FPHI=0.                                                               SOIL1560
      ALAM  =.5*(APERP/ATHTA)                                               SOIL1570
  505 CONTINUE                                                              SOIL1580
      IF(THETAL.GT.(3.*PIE/4.))ALAM  =0.0                                   SOIL1590
      CD=.8+CDCONT*RR*RR*FPHI                                               SOIL1600
      AAA=CD*SLRHO*ATHTA*VBAR*VBAR                                          SOIL1610
      BBB=3.*SOILNU*RR*RR*DRDD*((ATHTA/AREA)**1.5)*VBAR*VBAR*COST           SOIL1620
      CCC=CMS*SLRHO*GRAV*ATHTA                                              SOIL1630
      FAP=CCC*ADEPTH+AAA+BBB                                                SOIL1640
      IF(VBAR.LT.0.000001) GO TO 510                                        SOIL1650
      IF(ABS(XFPSD(1,IFP)).LT.0.000001) GO TO 506                           SOIL1660
      SGN=XFPSD(1,IFP)/ABS(XFPSD(1,IFP))                                    SOIL1670
```

```
      GO TO 507                                                         SOIL1680
  506 SGN=+1.0                                                          SOIL1690
  507 CONTINUE                                                          SOIL1700
      CX=XFPSD(1,IFP)/VBAR                                              SOIL1710
      IF(VHOR.LT.1.E-10) GO TO 508                                      SOIL1720
      CY=YFPSD(1,IFP)/VHOR                                              SOIL1730
      CZ=ZFPSD(1,IFP)/VHOR                                              SOIL1740
      GO TO 509                                                         SOIL1750
  508 CY=0.0                                                            SOIL1760
      CZ=0.0                                                            SOIL1770
  509 CONTINUE                                                          SOIL1780
      CH=VHOR/VBAR                                                      SOIL1790
      FH=(CH+CX*ALAM   )*FAP                                            SOIL1800
      SOILX(IFP)=(-CX+SGN*CH*ALAM   )*FAP                               SOIL1810
      IF(SOILX(IFP).LT.0.)SOILX(IFP)=0.0                                SOIL1820
      SOILY(IFP)=-CY*FH                                                 SOIL1830
      SOILZ(IFP)=-CZ*FH                                                 SOIL1840
      GO TO 520                                                         SOIL1850
  510 CONTINUE                                                          SOIL1860
      CY=0.0                                                            SOIL1870
      CZ=0.0                                                            SOIL1880
      SOILX(IFP)=FAP                                                    SOIL1890
      SOILY(IFP)=0.0                                                    SOIL1900
      SOILZ(IFP)=0.0                                                    SOIL1910
  520 CONTINUE                                                          SOIL1920
      PRESS=SOILX(IFP)/AREA                                             SOIL1930
C                                                                       SOIL1940
C     *  *  *  *  *                                                     SOIL1950
C     FOOTPAD ATTENUATION                                               SOIL1960
C     *  *  *  *  *                                                     SOIL1970
C                                                                       SOIL1980
 1000 CONTINUE                                                          SOIL1990
      IF(FPSTR.GE.ADIST)GO TO 1001                                      SOIL2000
      IF(PRESS.LE.ATTPRS(3))RETURN                                      SOIL2010
      ATTH(IFP)=ATTHCK(3)                                               SOIL2020
      RETURN                                                            SOIL2030
 1001 IF(FPSTR.LT.ATTHCK(KJI))GO TO 1002                                SOIL2040
      CRPRES=ATTPRS(KJI)                                                SOIL2050
      GO TO 1004                                                        SOIL2060
 1002 IF(KJI.EQ.3)GO TO 1003                                            SOIL2070
      KJI=KJI+1                                                         SOIL2080
      GO TO 1001                                                        SOIL2090
 1003 CRPRES=ATTPRS(3)                                                  SOIL2100
 1004 CONTINUE                                                          SOIL2110
      IF(CRPRES.GT.PRESS) RETURN                                        SOIL2120
      SOILX(IFP)=CRPRES*AREA                                            SOIL2130
      IF(NTYPE.EQ.0)GO TO 1015                                          SOIL2140
      ADD=CRPRES/SOILP(2)                                               SOIL2150
      GO TO 1020                                                        SOIL2160
 1015 CONTINUE                                                          SOIL2170
      IF(VBAR.LT.0.000001) GO TO 1016                                   SOIL2180
      FAP=SOILX(IFP)/(-CX+SGN*CH*ALAM)                                  SOIL2190
      ADD=(FAP-AAA-BBB)/CCC                                             SOIL2200
      GO TO 1020                                                        SOIL2210
 1016 ADD=SOILX(IFP)/CCC                                                SOIL2220
 1020 CONTINUE                                                          SOIL2230
```

```
      IF(ABS(ADD).GT.ADEPTH)ADD=ADEPTH                                  SOIL2240
      IF(IFIN.EQ.0)ATTH(IFP)=FPSTR+ADD                                  SOIL2250
      SOILY(IFP)=-CY*SOILX(IFP)                                         SOIL2260
      SOILZ(IFP)=-CZ*SOILX(IFP)                                         SOIL2270
      RETURN                                                            SOIL2280
C                                                                       SOIL2290
C     FOOTPAD OFF SURFACE                                               SOIL2300
C                                                                       SOIL2310
 2000 CONTINUE                                                          SOIL2320
      SOILX(IFP)=0.0                                                    SOIL2330
      SOILY(IFP)=0.0                                                    SOIL2340
      SOILZ(IFP)=0.0                                                    SOIL2350
      RETURN                                                            SOIL2360
      END                                                               SOIL2370
```

```
      SUBROUTINE INITUP ( IYD, CUTVAL, IDIRT )                          INIT  10
      COMMON / IZZZRM / IPTATL,IPTOTL, LOCNAM(90)                       INIT  20
      DIMENSION XS(8), IAD(8), IND(8)                                   INIT  30
      COMMON COMINT (400)                                               INIT  40
      EQUIVALENCE ( COMINT( 329 ), XS     )                             INIT  50
      EQUIVALENCE ( COMINT( 344 ), IAD    )                             INIT  60
      EQUIVALENCE ( COMINT( 352 ), IND    )                             INIT  70
      ENTRY LOC                                                         INIT  80
      IPTATL = IPTATL + 1                                               INIT  90
      IF ( IPTATL .LE. 8 ) GO TO 3                                      INIT 100
      WRITE (6,1)                                                       INIT 110
    1 FORMAT ( 65H ----------JOB TERMINATED, MORE THAN EIGHT CALLS TO LOINIT 120
     *C----------)                                                      INIT 130
      STOP                                                              INIT 140
    3 CONTINUE                                                          INIT 150
      IAD(IPTATL)     = IYD                                             INIT 160
      XS( IPTATL ) = CUTVAL                                             INIT 170
      IND(IPTATL ) = IDIRT                                              INIT 180
      RETURN                                                            INIT 190
      ENTRY INUPD                                                       INIT 200
      IPTOTL = IPTOTL + 1                                               INIT 210
      IF ( IPTOTL .LE. 90 ) GO TO 4                                     INIT 220
      WRITE (6,2)                                                       INIT 230
    2 FORMAT(67H ---------JOB TERMINATED, MORE THAN NINETY CALLS TO INUPINIT 240
     *D----------)                                                      INIT 250
      STOP                                                              INIT 260
    4 LOCNAM(IPTOTL) = IYD                                              INIT 270
      RETURN                                                            INIT 280
      END                                                               INIT 290
```

```
      SUBROUTINE RKCUT        (YD, Y )
      COMMON / IZZZRM / IPTATL,IPTOTL, LOCNAM(90)
      DIMENSION X(8),XMN1(8),XS(8),IAD(8),IND(8),Y(4),YD(4)
      COMMON COMINT (400)
      EQUIVALENCE ( COMINT( 324 ), T       )
      EQUIVALENCE ( COMINT( 325 ), HMAX    )
      EQUIVALENCE ( COMINT( 326 ), HMIN    )
      EQUIVALENCE ( COMINT( 327 ), EMIN    )
      EQUIVALENCE ( COMINT( 328 ), EMAX    )
      EQUIVALENCE ( COMINT( 329 ), XS      )
      EQUIVALENCE ( COMINT ( 337 ), HZ      )
      EQUIVALENCE ( COMINT( 338 ), CUTERR)
      EQUIVALENCE ( COMINT( 339 ), IP      )
      EQUIVALENCE ( COMINT( 340 ), IVARH )
      EQUIVALENCE ( COMINT( 341 ), IMTH    )
      EQUIVALENCE ( COMINT( 342 ), IPRNT )
      EQUIVALENCE ( COMINT( 343 ), IFIN    )
      EQUIVALENCE ( COMINT( 344 ), IAD     )
      EQUIVALENCE ( COMINT( 352 ), IND     )
      EQUIVALENCE ( COMINT( 360 ), J       )
      EQUIVALENCE ( COMINT( 363 ), IVAL      )
      EQUIVALENCE ( COMINT( 368 ), H         )
      EQUIVALENCE ( COMINT( 369 ), XMN1      )
C         XMN1 USES COMINT(369-376)
      ENTRY SETUP
      IERROR = 0
    1 IPT2=2**IP
      IPT1=0
      LIST=0
      INDUPD=0
      HZ=HMAX*2.00**(-IP)
      HD2=HZ/2.00
      H=HD2
      IALP=4
      IPRNT=0
      IFIN=0
      IVAL=0
      ISCNT = 0
      IBI1=4
      IBI2=2
      IBU1=1
      IF(IVARH)3,2,3
    2 IBU2=2
      EMAX= ABS(EMAX)
      EMIN= ABS(EMIN)
      RETURN
    3 IBU2=1
      RETURN
      ENTRY INTEG
      GO TO(60,55,55,39),IBI1
   39 Y(4)=Y(1)+H*YD(1)
      IF(IBI2-1)50,41,50
   41 ER= ABS(HZ*(YD(1)-YD(3)))
      IF(Y(1))43,44,43
   43 IF(ER-ER/ ABS(Y(1))) 44,44,4
```

```
    4 ER=ER/ ABS(Y(1))                                                  RKCT 560
   44 IF(ER-EMAX)45,46,46                                               RKCT 570
   45 IF(ER-EMIN)50,48,48                                               RKCT 580
   48 IVAL=1                                                            RKCT 590
      GO TO 50                                                          RKCT 600
   46 IVAL=-8300000                                                     RKCT 610
      IF ( HZ .EQ. HMIN ) IVAL = 1                                      RKCT 620
   50 YD(3)=YD(1)                                                       RKCT 630
      RETURN                                                            RKCT 640
   55 Y(4)=Y(2)+H*YD(1)                                                 RKCT 650
      YD(3)=YD(3)+2.00*YD(1)                                            RKCT 660
      RETURN                                                            RKCT 670
   60 Y(4)=Y(2)+H/6.00*(YD(3)+YD(1))                                    RKCT 680
      YD(3)=YD(1)                                                       RKCT 690
      RETURN                                                            RKCT 700
      ENTRY UPDAT                                                       RKCT 710
  100 IFIN=1                                                            RKCT 720
      IBU3=1                                                            RKCT 730
      IF(IPRNT)110,115,110                                              RKCT 740
  115 IPT1=IPT2                                                         RKCT 750
      IBU3=2                                                            RKCT 760
  110 IF(IALP-1)118,120,118                                             RKCT 770
  120 IPT1=IPT1-1                                                       RKCT 780
      IALP=4                                                            RKCT 790
      H=HD2                                                             RKCT 800
      IFIN=0                                                            RKCT 810
  121 DO 122 IMVER = 1, IPTOTL                                          RKCT 820
      KMVER = LOCNAM( IMVER )                                           RKCT 830
  122 COMINT( KMVER ) = COMINT( KMVER + 3 )                             RKCT 840
  150 IPRNT=IPT1                                                        RKCT 850
      IBI1=IALP                                                         RKCT 860
      IVAL=0                                                            RKCT 870
      RETURN                                                            RKCT 880
  118 IALP=IALP-1                                                       RKCT 890
      GO TO(125,126,127),IALP                                           RKCT 900
  126 IF(IBU2-1)302,301,302                                             RKCT 910
  302 IBI2=1                                                            RKCT 920
      IBU1=2                                                            RKCT 930
  301 H=HZ                                                              RKCT 940
      GO TO 121                                                         RKCT 950
  125 T=T+HD2                                                           RKCT 960
      GO TO 121                                                         RKCT 970
  127 GO TO(130,131),IBU1                                               RKCT 980
  130 DO 132 IMVER = 1, IPTOTL                                          RKCT 990
      KMVER = LOCNAM( IMVER )                                           RKCT1000
      COMINT( KMVER + 1 ) = COMINT( KMVER )                             RKCT1010
  132 COMINT( KMVER     ) = COMINT( KMVER + 3 )                         RKCT1020
      T=T+HD2                                                           RKCT1030
      GO TO 150                                                         RKCT1040
  131 IF(IVAL)135,136,135                                               RKCT1050
  136 IF(ISCNT-1)137,137,138                                            RKCT1060
  137 ISCNT=ISCNT+1                                                     RKCT1070
      GO TO 130                                                         RKCT1080
  138 HIPT1=IPT1/2                                                      RKCT1090
      XIPT1=IPT1                                                        RKCT1100
      XIPT1=XIPT1/2.00                                                  RKCT1110
```

```
      IF(XIPT1-HIPT1)130,140,130                                        RKCT1120
  140 IPT2=IPT2/2                                                       RKCT1130
      IPT1=IPT1/2                                                       RKCT1140
      ISCNT=0                                                           RKCT1150
      IVAL=0                                                            RKCT1160
      H=HZ                                                              RKCT1170
      HD2=HZ                                                            RKCT1180
      HZ=2.00*HZ                                                        RKCT1190
  139 IALP=4                                                            RKCT1200
      IBI2=2                                                            RKCT1210
      IBU1=1                                                            RKCT1220
      GO TO 150                                                         RKCT1230
  135 ISCNT=0                                                           RKCT1240
      IF(IVAL)160,160,130                                               RKCT1250
  160 IF(IPT1)130,161,161                                               RKCT1260
  161 IF(IBU3-1)163,165,163                                             RKCT1270
  163 IPT1=0                                                            RKCT1280
  165 IPT1=2*(IPT1+1)                                                   RKCT1290
      IPT2=2*IPT2                                                       RKCT1300
      T=T-HZ                                                            RKCT1310
      HZ=HD2                                                            RKCT1320
      IF ( HZ .LT. HMIN ) HZ = HMIN                                     RKCT1330
      HD2=HZ/2.00                                                       RKCT1340
      H=HD2                                                             RKCT1350
      DO 170 IMVER = 1, IPTOTL                                          RKCT1360
      KMVER = LOCNAM( IMVER )                                           RKCT1370
  170 COMINT( KMVER ) = COMINT( KMVER+1 )                               RKCT1380
      GO TO 139                                                         RKCT1390
      ENTRY CUT                                                         RKCT1400
      IF(IFIN)200,250,200                                               RKCT1410
  200 J=0                                                               RKCT1420
      IERROR = 1                                                        RKCT1430
      RETURN                                                            RKCT1440
  250 K=1                                                               RKCT1450
  260 IF (  K .LE. IPTATL ) GO TO 300                                   RKCT1460
  270 IF(K-1)280,200,280                                                RKCT1470
  280 LL=K-1                                                            RKCT1480
      DO 290 I = 1, LL                                                  RKCT1490
      KMVER = IAD(I)                                                    RKCT1500
  290 XMN1(I) = COMINT( KMVER )                                         RKCT1510
      GO TO 200                                                         RKCT1520
  300 XU=XS(K)+.5*CUTERR                                                RKCT1530
      XL=XS(K)-.5*CUTERR                                                RKCT1540
      IF(IND(K))500,310,500                                             RKCT1550
  310 KMVER = IAD(K)                                                    RKCT1560
      X(K) = COMINT( KMVER )                                            RKCT1570
      IF(X(K)-XL) 320,320,400                                           RKCT1580
  320 IF(IVAL)200,330,330                                               RKCT1590
  330 K=K+1                                                             RKCT1600
      IF(K-9)260,280,280                                                RKCT1610
  400 IF(X(K)-XU)410,600,600                                            RKCT1620
  410 J=K                                                               RKCT1630
      IERROR = 1                                                        RKCT1640
      RETURN                                                            RKCT1650
  500 KMVER = IAD(K)                                                    RKCT1660
      X(K)  = COMINT( KMVER )                                           RKCT1670
```

```
      IF(X(K)-XU)510,320,320                                            RKCT1680
  510 IF(X(K)-XL)600,600,410                                            RKCT1690
  600 IF ( IVAL .LT. 0 ) GO TO 200                                      RKCT1700
      IF ( IERROR .NE. 0 ) GO TO 1054                                   RKCT1710
      WRITE(6,1051)K                                                    RKCT1720
 1051 FORMAT(1H0,28H********** CUTOFF PASSED BY ,I1,56HTH CUTOFF VARIABLRKCT1730
     1E ON THE INITIAL CALL TO CUT **********     )                     RKCT1740
      STOP                                                              RKCT1750
 1054 CONTINUE                                                          RKCT1760
      HN=HZ/2.00*((XS(K)-XMN1(K))/(X(K)-XMN1(K)))                       RKCT1770
      T=T-HZ                                                            RKCT1780
      HZ=HN                                                             RKCT1790
      HD2=HZ/2.00                                                       RKCT1800
      H=HD2                                                             RKCT1810
      IALP=4                                                            RKCT1820
      IVAL=0                                                            RKCT1830
      IBI2=2                                                            RKCT1840
      IBU1=1                                                            RKCT1850
      IBI1=IALP                                                         RKCT1860
      DO  640  IMVER = 1, IPTOTL                                        RKCT1870
      KMVER = LOCNAM(IMVER )                                            RKCT1880
  640 COMINT(KMVER ) = COMINT( KMVER+1 )                                RKCT1890
      IFIN=1                                                            RKCT1900
      IPT1=IPT2                                                         RKCT1910
      ISCNT=0                                                           RKCT1920
      J=-1                                                              RKCT1930
      IERROR = 1                                                        RKCT1940
      RETURN                                                            RKCT1950
      END                                                               RKCT1960
```

```
      SUBROUTINE OUTPUT                                                 OUTP  10
      DIMENSION DISP(3),VEL(3),ACCEL(3),EL(3),ELD(3),ELDD(3),           OUTP  20
     *          AVEL(9),AACCEL(9),AA(3),BB(3),CC(3),DD(3),EE(3),        OUTP  30
     *          AOUT(3,10),NAC(10)                                      OUTP  40
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)                            OUTP  50
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),                       OUTP  60
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),                     OUTP  70
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)                    OUTP  80
      DIMENSION BOUT(75)                                                OUTP  90
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),     OUTP 100
     1    INDFPC(6),                                                    OUTP 110
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),   OUTP 120
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),   OUTP 130
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),   OUTP 140
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5), OUTP 150
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),   OUTP 160
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)                            OUTP 170
      EQUIVALENCE ( NFTPDS, NOLEG )                                     OUTP 180
      COMMON COMINT(400)                                                OUTP 190
      EQUIVALENCE ( COMINT(  3 ), IBOTM       )                         OUTP 200
      EQUIVALENCE ( COMINT(  4 ), XSD    )                              OUTP 210
      EQUIVALENCE ( COMINT(  8 ), XSDD   )                              OUTP 220
      EQUIVALENCE ( COMINT( 12 ), YS     )                              OUTP 230
      EQUIVALENCE ( COMINT( 16 ), YSD    )                              OUTP 240
      EQUIVALENCE ( COMINT( 20 ), YSDD   )                              OUTP 250
      EQUIVALENCE ( COMINT( 24 ), ZS     )                              OUTP 260
      EQUIVALENCE ( COMINT( 28 ), ZSD    )                              OUTP 270
      EQUIVALENCE ( COMINT( 32 ), ZSDD   )                              OUTP 280
      EQUIVALENCE ( COMINT( 36 ), PHI    )                              OUTP 290
      EQUIVALENCE ( COMINT( 40 ), PHID   )                              OUTP 300
      EQUIVALENCE ( COMINT( 44 ), WX     )                              OUTP 310
      EQUIVALENCE ( COMINT( 48 ), WXD    )                              OUTP 320
      EQUIVALENCE ( COMINT( 52 ), THTA   )                              OUTP 330
      EQUIVALENCE ( COMINT( 56 ), THTAD  )                              OUTP 340
      EQUIVALENCE ( COMINT( 60 ), WY     )                              OUTP 350
      EQUIVALENCE ( COMINT( 64 ), WYD    )                              OUTP 360
      EQUIVALENCE ( COMINT( 68 ), PSI    )                              OUTP 370
      EQUIVALENCE ( COMINT( 72 ), PSID   )                              OUTP 380
      EQUIVALENCE ( COMINT( 76 ), WZ     )                              OUTP 390
      EQUIVALENCE ( COMINT( 80 ), WZD    )                              OUTP 400
      EQUIVALENCE ( COMINT( 84 ), GC     )                              OUTP 410
      EQUIVALENCE ( COMINT(104 ), GCD    )                              OUTP 420
      EQUIVALENCE ( COMINT(124 ), GCDD   )                              OUTP 430
      EQUIVALENCE ( COMINT(144 ), XFPS   )                              OUTP 440
      EQUIVALENCE ( COMINT(164 ), XFPSD  )                              OUTP 450
      EQUIVALENCE ( COMINT(184 ), XFPSDD )                              OUTP 460
      EQUIVALENCE ( COMINT(204 ), YFPS   )                              OUTP 470
      EQUIVALENCE ( COMINT(224 ), YFPSD  )                              OUTP 480
      EQUIVALENCE ( COMINT(244 ), YFPSDD )                              OUTP 490
      EQUIVALENCE ( COMINT(264 ), ZFPS   )                              OUTP 500
      EQUIVALENCE ( COMINT(284 ), ZFPSD  )                              OUTP 510
      EQUIVALENCE ( COMINT(304 ), ZFPSDD )                              OUTP 520
      EQUIVALENCE ( COMINT( 324 ), TIME  )                              OUTP 530
      EQUIVALENCE ( COMINT( 325 ), HMAX  )                              OUTP 540
      EQUIVALENCE ( COMINT( 326 ), HMIN  )                              OUTP 550
```

```
      EQUIVALENCE ( COMINT( 327 ), EMIN   )                             OUTP 560
      EQUIVALENCE ( COMINT( 328 ), EMAX   )                             OUTP 570
      EQUIVALENCE ( COMINT( 329 ), XSI    )                             OUTP 580
      EQUIVALENCE ( COMINT ( 337 ), HZ     )                            OUTP 590
      EQUIVALENCE ( COMINT( 338 ), CUTERR)                              OUTP 600
      EQUIVALENCE ( COMINT( 339 ), IP     )                             OUTP 610
      EQUIVALENCE ( COMINT( 340 ), IVARH )                              OUTP 620
      EQUIVALENCE ( COMINT( 341 ), IMTH  )                              OUTP 630
      EQUIVALENCE ( COMINT( 342 ), IPRNT )                              OUTP 640
      EQUIVALENCE ( COMINT( 343 ), IFIN  )                              OUTP 650
      EQUIVALENCE ( COMINT( 344 ), IAD   )                              OUTP 660
      EQUIVALENCE ( COMINT( 352 ), IND   )                              OUTP 670
      EQUIVALENCE ( COMINT( 360 ), JCUT  )                              OUTP 680
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )                             OUTP 690
      EQUIVALENCE ( COMINT( 364 ), XS     )                             OUTP 700
C                   COMINT(364-367) USED BY XS                          OUTP 710
      COMMON                                                            OUTP 720
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY, OUTP 730
     2    DC(3,3)         , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),     OUTP 740
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)       ,      OUTP 750
     4    GRAV  , GRAVE , ZETA  , FTS (6)        , FSXSI(5)       ,     OUTP 760
     5    FSYSI(5)        , FSZSI(5)        , SOILX(5)        ,         OUTP 770
     6    SOILY(5)        , SOILZ(5)        , PMSX(5,5)       ,         OUTP 780
     7    PMSY(5,5)       , PMSZ(5,5)       , PDSX(10,5)      ,         OUTP 790
     8    PDSY(10,5)      , PDSZ(10,5)      , FLXS   , FLYS , FLZS   ,  OUTP 800
     9    TLXL  ,   TLYL  , TLZL  , SLO    , XMSCB(5)        ,          OUTP 810
     C    YMSCB(5)        , ZMSCB(5)        , XDSCB(10)       ,         OUTP 820
     D    YDSCB(10)       , ZDSCB(10)       , ILEG   , IMS  ,           OUTP 830
     E    FSTX  , FSTY  , FSTZ  , PVCBX   , PVCBY  , PVCBZ   ,          OUTP 840
     F    PVFPX , PVFPY , PVFPZ , NOLEG   , SLOMS(5) ,                  OUTP 850
     G    SLODS(10)                                                     OUTP 860
      COMMON                                                            OUTP 870
     1    PFCMS(5)        , PFCDS(5)        , PFTDS(5)        ,         OUTP 880
     2    PFTMS(5)        , SRCDS(5)        , SRCMS(5)        ,         OUTP 890
     3    SRTDS(5)   ,      SRTMS(5)        , COEFDS, COEFMS, GAMDS ,   OUTP 900
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,       OUTP 910
     5    STMXDS, STMXMS, CDCDS(5)          , CDCMS(5)        ,         OUTP 920
     6    CDTDS(5)        , CDTMS(5)        , FRICDS, FRICMS, IRETDS,   OUTP 930
     7    IRETMS, STROKE(10)      , STRKDS(10)      , STRKMS( 5)      , OUTP 940
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),    OUTP 950
     9    ATTPRS(3)       , ADIST , SOILP(3)        , NMODES,           OUTP 960
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),       OUTP 970
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),    OUTP 980
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,       OUTP 990
     D    INDFZR, TIMAX , DRAGST, IFP                                   OUTP1000
      COMMON                                                            OUTP1010
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),   OUTP1020
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),       OUTP1030
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),       OUTP1040
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,         OUTP1050
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)          OUTP1060
      COMMON                                                            OUTP1070
     1    SMXMSC(5),  TMXMSC(5),  SMXMST(5),  TMXMST(5),                OUTP1080
     2    SMXDSC(10), TMXDSC(10), SMXDST(10), TMXDST(10)                OUTP1090
     3   ,SLNGMS(5) ,SLNGDS(10), CURDSL, INLEG , IFPPRT,                OUTP1100
     4    IMPACT(5) ,IPRTFP(5),  KOUNT(5)  ,ANGX,  ANGY,  ANGZ          OUTP1110
```

```
      RADIAN=57.295779513                       OUTP1120
      DUM1=PHI*RADIAN                           OUTP1130
      DUM2=WX*RADIAN                            OUTP1140
      DUM3=WXD*RADIAN                           OUTP1150
      DUM4=THTA*RADIAN                          OUTP1160
      DUM5=WY*RADIAN                            OUTP1170
      DUM6=WYD*RADIAN                           OUTP1180
      DUM7=PSI*RADIAN                           OUTP1190
      DUM8=WZ*RADIAN                            OUTP1200
      DUM9=WZD*RADIAN                           OUTP1210
      AVEL(1)=0.                                OUTP1220
      AVEL(2)=WZ                                OUTP1230
      AVEL(3)=-WY                               OUTP1240
      AVEL(4)=-WZ                               OUTP1250
      AVEL(5)=0.                                OUTP1260
      AVEL(6)=WX                                OUTP1270
      AVEL(7)=WY                                OUTP1280
      AVEL(8)=-WX                               OUTP1290
      AVEL(9)=0.                                OUTP1300
      AACCEL(1)=0.                              OUTP1310
      AACCEL(2)=WZD                             OUTP1320
      AACCEL(3)=-WYD                            OUTP1330
      AACCEL(4)=-WZD                            OUTP1340
      AACCEL(5)=0.                              OUTP1350
      AACCEL(6)=WXD                             OUTP1360
      AACCEL(7)=WYD                             OUTP1370
      AACCEL(8)=-WXD                            OUTP1380
      AACCEL(9)=0.                              OUTP1390
      DISP(1)=XS                                OUTP1400
      DISP(2)=YS                                OUTP1410
      DISP(3)=ZS                                OUTP1420
      VEL(1)=XSD                                OUTP1430
      VEL(2)=YSD                                OUTP1440
      VEL(3)=ZSD                                OUTP1450
      ACCEL(1)=XSDD                             OUTP1460
      ACCEL(2)=YSDD                             OUTP1470
      ACCEL(3)=ZSDD                             OUTP1480
C                                               OUTP1490
C     CENTER OF GRAVITY MOTIONS                 OUTP1500
C                                               OUTP1510
      WRITE(6,350)                              OUTP1520
      WRITE(6,1000)TIME,HZ                      OUTP1530
      IF(NMODES.EQ.0)GO TO 90                   OUTP1540
      DO 50 I=1,3                               OUTP1550
      EL(I)=0.                                  OUTP1560
      ELD(I)=0.                                 OUTP1570
   50 ELDD(I)=0.                                OUTP1580
      DO 51 I=1,NMODES                          OUTP1590
      EL(1)=EL(1)+PCGX(I)*GC(1,I)               OUTP1600
      EL(2)=EL(2)+PCGY(I)*GC(1,I)               OUTP1610
      EL(3)=EL(3)+PCGZ(I)*GC(1,I)               OUTP1620
      ELD(1)=ELD(1)+PCGX(I)*GCD(1,I)            OUTP1630
      ELD(2)=ELD(2)+PCGY(I)*GCD(1,I)            OUTP1640
      ELD(3)=ELD(3)+PCGZ(I)*GCD(1,I)            OUTP1650
      ELDD(1)=ELDD(1)+PCGX(I)*GCDD(1,I)         OUTP1660
      ELDD(2)=ELDD(2)+PCGY(I)*GCDD(1,I)         OUTP1670
```

```
      ELDD(3)=ELDD(3)+PCGZ(I)*GCDD(1,I)                                 OUTP1680
   51 CONTINUE                                                          OUTP1690
      CALL GMPRD(DC,EL,AA,3,3,1)                                        OUTP1700
      DO 52 I=1,3                                                       OUTP1710
   52 DISP(I)=DISP(I)+AA(I)                                             OUTP1720
      CALL GMPRD(AVEL ,EL,AA,3,3,1)                                     OUTP1730
      DO 53 I=1,3                                                       OUTP1740
   53 BB(I)=ELD(I)+AA(I)                                                OUTP1750
      CALL GMPRD(DC,BB,CC,3,3,1)                                        OUTP1760
      DO 54 I=1,3                                                       OUTP1770
   54 VEL(I)=VEL(I)+CC(I)                                               OUTP1780
      CALL GMPRD(AVEL,AA,BB,3,3,1)                                      OUTP1790
      CALL GMPRD(AVEL,ELD,AA,3,3,1)                                     OUTP1800
      CALL GMPRD(AACCEL,EL,CC,3,3,1)                                    OUTP1810
      DO 55 I=1,3                                                       OUTP1820
   55 DD(I)=ELDD(I)+CC(I)+2.*AA(I)+BB(I)                                OUTP1830
      CALL GMPRD(DC,DD,AA,3,3,1)                                        OUTP1840
      DO 56 I=1,3                                                       OUTP1850
   56 ACCEL(I)=ACCEL(I)+AA(I)                                           OUTP1860
   90 CONTINUE                                                          OUTP1870
      WRITE(6,1001)DISP(1),VEL(1),ACCEL(1),DUM1,DUM2,DUM3,              OUTP1880
     *             DISP(2),VEL(2),ACCEL(2),DUM4,DUM5,DUM6,              OUTP1890
     *             DISP(3),VEL(3),ACCEL(3),DUM7,DUM8,DUM9               OUTP1900
      WRITE(6,1022)                                                     OUTP1910
      DO 100 I=1,NFTPDS                                                 OUTP1920
      IF(INDFPC(I).LE.0)GO TO 100                                       OUTP1930
      WRITE(6,1003)I,                                                   OUTP1940
     *             XFPS(1,I),XFPSD(1,I),XFPSDD(1,I),                    OUTP1950
     *             YFPS(1,I),YFPSD(1,I),YFPSDD(1,I),                    OUTP1960
     *             ZFPS(1,I),ZFPSD(1,I),ZFPSDD(1,I)                     OUTP1970
  100 CONTINUE                                                          OUTP1980
C                                                                       OUTP1990
C     ACCELERATION AND INTEGRATED QUANTITIES                            OUTP2000
C                                                                       OUTP2010
      DO 60 I=1,3                                                       OUTP2020
   60 AA(I)=(DC(1,I)*ACCEL(1)+DC(2,I)*ACCEL(2)+DC(3,I)*ACCEL(3))/GRAVE  OUTP2030
      IF(NOOUT.EQ.0)GO TO 202                                           OUTP2040
      DO 200 J=1,NOOUT                                                  OUTP2050
      DO 102 I=1,3                                                      OUTP2060
      ELD(I)=0.                                                         OUTP2070
  102 ELDD(I)=0.                                                        OUTP2080
      EL(1)=XOUT(J)                                                     OUTP2090
      EL(2)=YOUT(J)                                                     OUTP2100
      EL(3)=ZOUT(J)                                                     OUTP2110
      IF(NMODES.EQ.0)GO TO 111                                          OUTP2120
      DO 110 I=1,NMODES                                                 OUTP2130
      EL(1)=EL(1)+POUTX(J,I)*GC(1,I)                                    OUTP2140
      EL(2)=EL(2)+POUTY(J,I)*GC(1,I)                                    OUTP2150
      EL(3)=EL(3)+POUTZ(J,I)*GC(1,I)                                    OUTP2160
      ELD(1)=ELD(1)+POUTX(J,I)*GCD(1,I)                                 OUTP2170
      ELD(2)=ELD(2)+POUTY(J,I)*GCD(1,I)                                 OUTP2180
      ELD(3)=ELD(3)+POUTZ(J,I)*GCD(1,I)                                 OUTP2190
      ELDD(1)=ELDD(1)+POUTX(J,I)*GCDD(1,I)                              OUTP2200
      ELDD(2)=ELDD(2)+POUTY(J,I)*GCDD(1,I)                              OUTP2210
      ELDD(3)=ELDD(3)+POUTZ(J,I)*GCDD(1,I)                              OUTP2220
  110 CONTINUE                                                          OUTP2230
```

```
  111 CONTINUE                                                          OUTP2240
      CALL GMPRD(AVEL,EL,BB,3,3,1)                                      OUTP2250
      CALL GMPRD(AVEL,BB,CC,3,3,1)                                      OUTP2260
      CALL GMPRD(AACCEL,EL,BB,3,3,1)                                    OUTP2270
      CALL GMPRD(AVEL,ELD,DD,3,3,1)                                     OUTP2280
      DO 120 I=1,3                                                      OUTP2290
  120 EE(I)=ELDD(I)+BB(I)+2.*DD(I)+CC(I)                                OUTP2300
      AOUT(1,J)=(DC(1,1)*XSDD+DC(2,1)*YSDD+DC(3,1)*ZSDD+EE(1))/GRAVE    OUTP2310
      AOUT(2,J)=(DC(1,2)*XSDD+DC(2,2)*YSDD+DC(3,2)*ZSDD+EE(2))/GRAVE    OUTP2320
      AOUT(3,J)=(DC(1,3)*XSDD+DC(2,3)*YSDD+DC(3,3)*ZSDD+EE(3))/GRAVE    OUTP2330
  200 CONTINUE                                                          OUTP2340
  202 CONTINUE                                                          OUTP2350
      IF(NMODES.EQ.0)GO TO 220                                          OUTP2360
      IF(IQUOUT.EQ.0)GO TO 220                                          OUTP2370
      IF(NOOUT.NE.0)GO TO 210                                           OUTP2380
C     PRINT C.G. ACCELERATION AND INTEGRATED QUANTITIES                 OUTP2390
      WRITE(6,1005)                                                     OUTP2400
      WRITE(6,1007)AA(1),AA(2),AA(3),XS,XSD,XSDD                        OUTP2410
      WRITE(6,1008)YS,YSD,YSDD                                          OUTP2420
      WRITE(6,1009)ZS,ZSD,ZSDD                                          OUTP2430
      DO 201 I=1,NMODES                                                 OUTP2440
  201 WRITE(6,1010)I,GC(1,I),GCD(1,I),GCDD(1,I)                         OUTP2450
      GO TO 300                                                         OUTP2460
C     PRINT C.G. ACCELERATION, SECONDARY POINTS, AND INTEGRATED QUANT.  OUTP2470
  210 CONTINUE                                                          OUTP2480
      DO 95 I=1,10                                                      OUTP2490
   95 NAC(I)=I                                                          OUTP2500
      WRITE(6,1005)                                                     OUTP2505
      WRITE(6,1007)AA(1),AA(2),AA(3),XS,XSD,XSDD                        OUTP2510
      WRITE(6,1011)NAC(1),AOUT(1,1),AOUT(2,1),AOUT(3,1),YS,YSD,YSDD     OUTP2520
      WRITE(6,1012)NAC(2),AOUT(1,2),AOUT(2,2),AOUT(3,2),ZS,ZSD,ZSDD     OUTP2530
      DO 211 I=1,NMODES                                                 OUTP2540
      II=I+2                                                            OUTP2550
  211 WRITE(6,1013)NAC(II),AOUT(1,II),AOUT(2,II),AOUT(3,II),I,GC(1,I),  OUTP2560
     *                GCD(1,I),GCDD(1,I)                                OUTP2570
      IF(II.GE.NOOUT)GO TO 300                                          OUTP2580
      II=II+1                                                           OUTP2590
      DO 212 I=II,NOOUT                                                 OUTP2600
  212 WRITE(6,1014)NAC(I),AOUT(1,I),AOUT(2,I),AOUT(3,I)                 OUTP2610
      GO TO 300                                                         OUTP2620
C     PRINT C.G. ACCELERATIONS AND SECONDARY POINTS                     OUTP2630
  220 CONTINUE                                                          OUTP2640
      WRITE(6,1004)                                                     OUTP2650
      WRITE(6,1006)AA(1),AA(2),AA(3)                                    OUTP2660
      IF(NOOUT.EQ.0)GO TO 300                                           OUTP2670
      DO 96 I=1,10                                                      OUTP2675
   96 NAC(I)=I                                                          OUTP2676
      DO 221 I=1,NOOUT                                                  OUTP2680
  221 WRITE(6,1014)NAC(I),AOUT(1,I),AOUT(2,I),AOUT(3,I)                 OUTP2690
C                                                                       OUTP2700
C     STRUT INFORMATION                                                 OUTP2710
C                                                                       OUTP2720
  300 CONTINUE                                                          OUTP2730
      WRITE(6,1020)                                                     OUTP2740
      DO 205 I=1,NFTPDS                                                 OUTP2750
      II=2*(I-1)+1                                                      OUTP2760
```

```
      II2=II+1                                                          OUTP2770
      CRUSH=SS(1)-ATTH(I)                                               OUTP2780
  205 WRITE(6,1021)I,CRUSH  ,FORMS(I),STRKMS(I),II,FORDS(II),STRKDS(II),OUTP2790
     *II2,FORDS(II2),STRKDS(II2)                                        OUTP2800
C                                                                       OUTP2810
C     CHECK LANDER STABILITY                                            OUTP2820
C                                                                       OUTP2830
      IF(JCKSAB.EQ.0)GO TO 400                                          OUTP2840
C                                                                       OUTP2850
      CALL STABLE                                                       OUTP2860
C                                                                       OUTP2870
      IF(ISTAB.NE.0)GO TO 400                                           OUTP2880
      WRITE(6,1015)STAB,STABVL                                          OUTP2890
  400 CONTINUE                                                          OUTP2900
      IF(NFORC.EQ.0) RETURN                                             OUTP2910
C                                                                       OUTP2920
C     WRITE TIME HISTORY QUANTITIES ON TAPE FOR USE WITH LANDING        OUTP2930
C     LOADS PROGRAM                                                     OUTP2940
C         1. NFORC=1 RESULTS IN TIME HISTORY TAPE BEING GENERATED       OUTP2950
C         2. TIME HISTORY INFORMATION WRITTEN ON TAPE3                  OUTP2960
C                                                                       OUTP2970
      BOUT(1)=TIME                                                      OUTP2980
      BOUT(2)=HMAX                                                      OUTP2990
      BOUT(3)=XSDD                                                      OUTP3000
      BOUT(4)=YSDD                                                      OUTP3010
      BOUT(5)=ZSDD                                                      OUTP3020
      BOUT(6)=PHI                                                       OUTP3030
      BOUT(7)=THTA                                                      OUTP3040
      BOUT(8)=PSI                                                       OUTP3050
      BOUT(9)=WX                                                        OUTP3060
      BOUT(10)=WY                                                       OUTP3070
      BOUT(11)=WZ                                                       OUTP3080
      BOUT(12)=WXD                                                      OUTP3090
      BOUT(13)=WYD                                                      OUTP3100
      BOUT(14)=WZD                                                      OUTP3110
      NSP=14                                                            OUTP3120
      DO 450 I=1,NMODES                                                 OUTP3130
      NNL=NSP+3*I                                                       OUTP3140
      BOUT(NNL-2)=GC(1,I)                                               OUTP3150
      BOUT(NNL-1)=GCD(1,I)                                              OUTP3160
  450 BOUT(NNL)=GCDD(1,I)                                               OUTP3170
      NSP=14+3*NMODES                                                   OUTP3180
      DO 451 I=1,NOLEG                                                  OUTP3190
      NNL=NSP+3*I                                                       OUTP3200
      BOUT(NNL-2)=SAVMSX(I)                                             OUTP3210
      BOUT(NNL-1)=SAVMSY(I)                                             OUTP3220
  451 BOUT(NNL)=SAVMSZ(I)                                               OUTP3230
      II=2*NOLEG                                                        OUTP3240
      NSP=NSP+3*NOLEG                                                   OUTP3250
      DO 452 I=1,II                                                     OUTP3260
      NNL=NSP+3*I                                                       OUTP3270
      BOUT(NNL-2)=SAVDSX(I)                                             OUTP3280
      BOUT(NNL-1)=SAVDSY(I)                                             OUTP3290
  452 BOUT(NNL)=SAVDSZ(I)                                               OUTP3300
      NSP=NSP+3*II                                                      OUTP3310
      WRITE(3)(BOUT(I),I=1,NSP)                                         OUTP3320
```

```
      RETURN                                                            OUTP3330
C                                                                       OUTP3340
      ENTRY SUMMRY                                                      OUTP3350
C                                                                       OUTP3360
C     THIS PORTION OF THE SUBROUTINE PRINTS THE SUMMARY INFORMATION     OUTP3370
C                                                                       OUTP3380
      IF(NFORC.NE.0)END    FILE 3                                       OUTP3390
C                                                                       OUTP3400
      WRITE(6,2000)                                                     OUTP3410
      WRITE(6,2006)                                                     OUTP3420
      WRITE(6,2001)(I,TMXMSC(I),SMXMSC(I),TMXMST(I),SMXMST(I),I=1,NOLEG)OUTP3430
      WRITE(6,2007)                                                     OUTP3440
      NN=2*NOLEG                                                        OUTP3450
      WRITE(6,2001)(I,TMXDSC(I),SMXDSC(I),TMXDST(I),SMXDST(I),I=1,NN)   OUTP3460
      IF(IBOTM.NE.0)GO TO 1900                                          OUTP3470
      IF(ISTAB.EQ.0)WRITE(6,2003)                                       OUTP3480
      IF(ISTAB.EQ.1)WRITE(6,2004)                                       OUTP3490
      IF(ISTAB.EQ.2)WRITE(6,2005)                                       OUTP3500
      RETURN                                                            OUTP3510
 1900 CONTINUE                                                          OUTP3520
      IF(IFP.GT.10)GO TO 1950                                           OUTP3530
      IF(IBOTM.EQ.+1)GO TO 6000                                         OUTP3540
      WRITE(6,6201)IFP                                                  OUTP3550
      RETURN                                                            OUTP3560
 6000 WRITE(6,6200)IFP                                                  OUTP3570
      RETURN                                                            OUTP3580
 1950 CONTINUE                                                          OUTP3590
      IFP=IFP/100                                                       OUTP3600
      IF(IBOTM.EQ.+1)GO TO 6100                                         OUTP3610
      WRITE(6,6203)IFP                                                  OUTP3620
      RETURN                                                            OUTP3630
 6100 WRITE(6,6202)IFP                                                  OUTP3640
      RETURN                                                            OUTP3650
C                                                                       OUTP3660
C     FORMAT STATEMENTS                                                 OUTP3670
C                                                                       OUTP3680
  350 FORMAT(/)                                                         OUTP3690
 1000 FORMAT(/, 90H0* * * * * * * * * * * * * * * * * * * * * * * * * * OUTP3700
     *** * * * * * * * * * * * * * * * * *    //15X,7HTIME = ,E13.6,    OUTP3710
     *14H  STEP SIZE = ,E13.6)                                          OUTP3720
 1001 FORMAT(//3X,19HCENTER BODY MOTIONS//6X,39HTRANSLATION - SURFACE COOUTP3730
     *ORDINATE SYSTEM,10X,8HROTATION/91X,25HVELOCITY AND ACCELERATION/  OUTP3740
     *16X,4HDISP,8X,3HVEL,7X,5HACCEL,48X,24HLANDER COORDINATE SYSTEM/   OUTP3750
     *10X,2HX ,3(1X,E10.3),14X,20HEULER ANGLES  PHI   ,E10.3,3X,2HX ,   OUTP3760
     *2(1X,E10.3)/10X,2HY ,3(1X,E10.3),28X,6HTHETA ,E10.3,3X,2HY ,      OUTP3770
     *2(1X,E10.3)/10X,2HZ ,3(1X,E10.3),28X,6HPSI   ,E10.3,3X,2HZ ,      OUTP3780
     *2(1X,E10.3))                                                      OUTP3790
 1003 FORMAT(6X,7HFOOTPAD,I3,13X,1X,3(1X,E10.3),1X,                     OUTP3800
     *3(1X,E10.3),1X,3(1X,E10.3))                                       OUTP3810
 1004 FORMAT(/3X,53HACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEOUTP3820
     *M/                                                                OUTP3830
     *30X,1HX,10X,1HY,10X,1HZ)                                          OUTP3840
 1005 FORMAT(/3X,53HACCELERATIONS IN EARTH G*S - LANDER COORDINATE SYSTEOUTP3850
     *M,17X,33HCENTER BODY INTEGRATED QUANTITIES/                       OUTP3860
     *30X,1HX,10X,1HY,10X,1HZ,49X,4HDISP,8X,3HVEL,7X,5HACCEL)           OUTP3870
 1006 FORMAT(6X,18HCENTER OF GRAVITY ,3(1X,E10.3))                      OUTP3880
```

```
1007 FORMAT(6X,18HCENTER OF GRAVITY ,3(1X,E10.3) ,26X,13HCENTER BODY X,OUTP3890
    *2X,3(1X,E10.3))                                                   OUTP3900
1008 FORMAT(83X,13HCENTER BODY Y,2X,3(1X,E10.3))                       OUTP3910
1009 FORMAT(83X,13HCENTER BODY Z,2X,3(1X,E10.3))                       OUTP3920
1010 FORMAT(83X,4HMODE,I4,7X,3(1X,E10.3))                              OUTP3930
1011 FORMAT(6X,6HPOINT ,I2,10X,3(1X,E10.3),26X,13HCENTER BODY Y,2X,    OUTP3940
    *3(1X,E10.3))                                                      OUTP3950
1012 FORMAT(6X,6HPOINT ,I2,10X,3(1X,E10.3),26X,13HCENTER BODY Z,2X,    OUTP3960
    *3(1X,E10.3))                                                      OUTP3970
1013 FORMAT(6X,6HPOINT ,I2,10X,3(1X,E10.3),26X,4HMODE,I4,7X,3(1X,      OUTP3980
    *E10.3))                                                           OUTP3990
1014 FORMAT(6X,6HPOINT ,I2,10X,3(1X,E10.3))                            OUTP4000
1015 FORMAT(/3X,16HLANDER STABILITY/6X,20HSTABILITY ANGLE   = ,        OUTP4010
    *E10.3/6X,20HPITCHING VELOCITY = ,E10.3)                           OUTP4020
1020 FORMAT(/3X,30HLANDING GEAR STRUT INFORMATION/6X,3HLEG,6X,         OUTP4030
    *5HCRUSH,18X,4HLOAD,                                               OUTP4040
    *6X,6HSTROKE,20X,4HLOAD,6X,6HSTROKE,20X,4HLOAD,6X,6HSTROKE)        OUTP4050
1021 FORMAT(7X,I1,4X,E10.3,2X,10HMAIN STRUT,2(1X,E10.3),12H  DRAG STRUTOUTP4060
    *,I2,                                                              OUTP4070
    *2(1X,E10.3),12H  DRAG STRUT,I2,2(1X,E10.3))                       OUTP4080
1022 FORMAT(/3X,43HFOOTPAD MOTIONS - SURFACE COORDINATE SYSTEM         OUTP4090
    */47X,1HX,33X,1HY,                                                 OUTP4100
    *33X,1HZ/34X,4HDISP,8X,3HVEL,7X,5HACCEL,7X,4HDISP,8X,3HVEL,7X,     OUTP4110
    *5HACCEL,7X,4HDISP,8X,3HVEL,7X,5HACCEL)                            OUTP4120
2000 FORMAT(//3X,19HSUMMARY INFORMATION/)                              OUTP4130
2001 FORMAT(7X,I2,2X,2(2X,E10.3),2X,2(2X,E10.3))                       OUTP4140
2003 FORMAT(/,6X,24HEND OF CASE - TIME LIMIT)                          OUTP4150
2004 FORMAT(/,6X,31HEND OF CASE - PITCH INSTABILITY)                   OUTP4160
2005 FORMAT(/,6X,29HEND OF CASE - YAW INSTABILITY)                     OUTP4170
2006 FORMAT(3X,26HMAXIMUM MAIN STRUT STROKES,/                         OUTP4180
    *19X,11HCOMPRESSION,17X,7HTENSION/6X,5HSTRUT,5X,4HTIME,7X,         OUTP4190
    *6HSTROKE,9X,4HTIME,7X,6HSTROKE)                                   OUTP4200
2007 FORMAT(3X,26HMAXIMUM DRAG STRUT STROKES,/                         OUTP4210
    *19X,11HCOMPRESSION,17X,7HTENSION/6X,5HSTRUT,5X,4HTIME,7X,         OUTP4220
    *6HSTROKE,9X,4HTIME,7X,6HSTROKE)                                   OUTP4230
6200 FORMAT(//10X,35HMAIN STRUT BOTTOMED ON TENSION SIDE,/             OUTP4240
    *      30X,8HSTRUT = ,I2)                                          OUTP4250
6201 FORMAT(//10X,39HMAIN STRUT BOTTOMED ON COMPRESSION SIDE,/         OUTP4260
    *      30X,8HSTRUT = ,I2)                                          OUTP4270
6202 FORMAT(//10X,35HDRAG STRUT BOTTOMED ON TENSION SIDE,/             OUTP4280
    *      30X,8HSTRUT = ,I2)                                          OUTP4290
6203 FORMAT(//10X,39HDRAG STRUT BOTTOMED ON COMPRESSION SIDE,/         OUTP4300
    *      30X,8HSTRUT = ,I2)                                          OUTP4310
     END                                                               OUTP4320
```

```
      SUBROUTINE STABLE
C
C     THIS ROUTINE CHECKS THE STABILITY OF THE LANDER
C     THE FOLLOWING FLAG IS RETURNED
C         1. ISTAB=0 - LANDER STABLE
C         2. ISTAB=1 - PITCH INSTABILITY
C         3. ISTAB=2 - YAW INSTABILITY
C
      DIMENSION DX(5),DY(5),DZ(5),SGN(5),ANG(5)
C
      DIMENSION GC(4,5), GCD(4,5), GCDD(4,5)
C
      DIMENSION  XFPS(4,5), YFPS(4,5), ZFPS(4,5),
     1          XFPSD(4,5), YFPSD(4,5), ZFPSD(4,5),
     2         XFPSDD(4,5), YFPSDD(4,5), ZFPSDD(4,5)
C
      COMMON / THISV/ TIMHSA(1), ATTH(5),AM(6,6),AME1(3),INDFPI(6),
     1    INDFPC(6),
     2    STRPDS(10), STRPMS( 5), IPOCDS(10), IPOCMS( 5), URCDS (10),
     3    URTDS (10), URCMS ( 5), URTMS ( 5), SETCDS(10), SETTDS(10),
     4    SETCMS( 5), SETTMS( 5), INDCDS(10), INDTDS(10), INDCMS(10),
     5  INDTMS( 5), PRFCDS (10), PRFTDS (10), PRFCMS ( 5), PRFTMS ( 5),
     6    IPRDS (10), IPRMS ( 5), FRVDSC(10), FRVDST(10), FRVMSC( 5),
     L    FRVMST( 5), IPOTDS(10), IPOTMS( 5)
      EQUIVALENCE ( NFTPDS, NOLEG )
      COMMON COMINT(400)
      EQUIVALENCE ( COMINT(  4 ), XSD     )
      EQUIVALENCE ( COMINT(  8 ), XSDD    )
      EQUIVALENCE ( COMINT( 12 ), YS      )
      EQUIVALENCE ( COMINT( 16 ), YSD     )
      EQUIVALENCE ( COMINT( 20 ), YSDD    )
      EQUIVALENCE ( COMINT( 24 ), ZS      )
      EQUIVALENCE ( COMINT( 28 ), ZSD     )
      EQUIVALENCE ( COMINT( 32 ), ZSDD    )
      EQUIVALENCE ( COMINT( 36 ), PHI     )
      EQUIVALENCE ( COMINT( 40 ), PHID    )
      EQUIVALENCE ( COMINT( 44 ), WX      )
      EQUIVALENCE ( COMINT( 48 ), WXD     )
      EQUIVALENCE ( COMINT( 52 ), THTA    )
      EQUIVALENCE ( COMINT( 56 ), THTAD   )
      EQUIVALENCE ( COMINT( 60 ), WY      )
      EQUIVALENCE ( COMINT( 64 ), WYD     )
      EQUIVALENCE ( COMINT( 68 ), PSI     )
      EQUIVALENCE ( COMINT( 72 ), PSID    )
      EQUIVALENCE ( COMINT( 76 ), WZ      )
      EQUIVALENCE ( COMINT( 80 ), WZD     )
      EQUIVALENCE ( COMINT( 84 ), GC      )
      EQUIVALENCE ( COMINT(104 ), GCD     )
      EQUIVALENCE ( COMINT(124 ), GCDD    )
      EQUIVALENCE ( COMINT(144 ), XFPS    )
      EQUIVALENCE ( COMINT(164 ), XFPSD   )
      EQUIVALENCE ( COMINT(184 ), XFPSDD  )
      EQUIVALENCE ( COMINT(204 ), YFPS    )
      EQUIVALENCE ( COMINT(264 ), ZFPS    )
      EQUIVALENCE ( COMINT(224 ), YFPSD   )
```

```
      EQUIVALENCE ( COMINT(244 ), YFPSDD )                              STAB 560
      EQUIVALENCE ( COMINT(284 ), ZFPSD  )                              STAB 570
      EQUIVALENCE ( COMINT(304 ), ZFPSDD )                              STAB 580
      EQUIVALENCE ( COMINT( 324 ), TIME   )                             STAB 590
      EQUIVALENCE ( COMINT( 325 ), HMAX   )                             STAB 600
      EQUIVALENCE ( COMINT( 326 ), HMIN   )                             STAB 610
      EQUIVALENCE ( COMINT( 327 ), EMIN   )                             STAB 620
      EQUIVALENCE ( COMINT( 328 ), EMAX   )                             STAB 630
      EQUIVALENCE ( COMINT( 329 ), XSI    )                             STAB 640
      EQUIVALENCE ( COMINT ( 337 ), HZ      )                           STAB 650
      EQUIVALENCE ( COMINT( 338 ), CUTERR)                              STAB 660
      EQUIVALENCE ( COMINT( 339 ), IP     )                             STAB 670
      EQUIVALENCE ( COMINT( 340 ), IVARH  )                             STAB 680
      EQUIVALENCE ( COMINT( 341 ), IMTH   )                             STAB 690
      EQUIVALENCE ( COMINT( 342 ), IPRNT  )                             STAB 700
      EQUIVALENCE ( COMINT( 343 ), IFIN   )                             STAB 710
      EQUIVALENCE ( COMINT( 344 ), IAD    )                             STAB 720
      EQUIVALENCE ( COMINT( 352 ), IND    )                             STAB 730
      EQUIVALENCE ( COMINT( 360 ), JCUT   )                             STAB 740
      EQUIVALENCE ( COMINT( 361 ), IPTCNT )                             STAB 750
      EQUIVALENCE ( COMINT ( 362 ), IDSETN )                            STAB 760
      EQUIVALENCE ( COMINT( 363 ), IVAL    )                            STAB 770
      EQUIVALENCE ( COMINT( 364 ), XS      )                            STAB 780
C                   COMINT(364-367) USED BY XS                          STAB 790
      COMMON                                                            STAB 800
     1    CBMASS, CBIXX , CBIXZ , CBIYY , CBIYZ , CBIZZ , FPMASS,CBIXY, STAB 810
     2    DC(3,3)       , XFP(5), YFP(5), ZFP(5), WNX(5), WNY(5),       STAB 820
     3    WNZ(5), PX(5) , PY(5) , PZ(5) , GM(5) , OMEGA(5)      ,       STAB 830
     4    GRAV  , GRAVE , ZETA  , FTS (6)       , FSXSI(5)      ,       STAB 840
     5    FSYSI(5)      , FSZSI(5)      , SOILX(5)      ,               STAB 850
     6    SOILY(5)      , SOILZ(5)      , PMSX(5,5)     ,               STAB 860
     7    PMSY(5,5)     , PMSZ(5,5)     , PDSX(10,5)    ,               STAB 870
     8    PDSY(10,5)    , PDSZ(10,5)    , FLXS    , FLYS , FLZS   ,     STAB 880
     9    TLXL  ,  TLYL   , TLZL  , SLO    , XMSCB(5)    ,              STAB 890
     C    YMSCB(5)      , ZMSCB(5)      , XDSCB(10)     ,               STAB 900
     D    YDSCB(10)     , ZDSCB(10)     , ILEG    , IMS  ,              STAB 910
     E    FSTX  , FSTY  , FSTZ  , PVCBX  , PVCBY  , PVCBZ   ,           STAB 920
     F    PVFPX , PVFPY , PVFPZ , NOLEG  , SLOMS(5) ,                   STAB 930
     G    SLODS(10)                                                     STAB 940
      COMMON                                                            STAB 950
     1    PFCMS(5)      , PFCDS(5)      , PFTDS(5)      ,               STAB 960
     2    PFTMS(5)      , SRCDS(5)      , SRCMS(5)      ,               STAB 970
     3    SRTDS(5)      , SRTMS(5)      , COEFDS, COEFMS, GAMDS ,       STAB 980
     4    GAMMS , SRUCDS, SRUCMS, SRUTDS, SRUTMS, SCMXDS, SCMXMS,       STAB 990
     5    STMXDS, STMXMS, CDCDS(5)      , CDCMS(5)      ,               STAB1000
     6    CDTDS(5)      , CDTMS(5)      , FRICDS, FRICMS, IRETDS,       STAB1010
     7    IRETMS, STROKE(10)     , STRKDS(10)   , STRKMS( 5)    ,       STAB1020
     8    LENGTH,CDXS,CDYS,CDZS , NTYPE , RAD(4), SS(4) , ATTHCK(3),    STAB1030
     9    ATTPRS(3)     , ADIST , SOILP(3)      , NMODES,               STAB1040
     A    COEF  , GAMMA , FRIC  , SCMAX , STMAX , CDC(5), CDT(5),       STAB1050
     B    SRULT , SRULC , SRT(5), SRC(5), PFT(5), PFC(5), GFLEGS(5),    STAB1060
     C    CURMSV, CURMSL, INDFXD, INDFYD, INDFZD, INDFXR, INDFYR,       STAB1070
     D    INDFZR, TIMAX , DRAGST, IFP                                   STAB1080
C                                                                       STAB1090
      COMMON                                                            STAB1100
     1CMS,CDCONT,SOILNU,SLRHO,      NOOUT,XOUT(10),YOUT(10),ZOUT(10),   STAB1110
```

```
     2MODEIN,POUTX(10,5),POUTY(10,5),POUTZ(10,5),PCGX(5),PCGY(5),       STAB1120
     3PCGZ(5),AEI1,AEI2,FORMS(5),FORDS(10),FORCE,NFORC,SAVMSX(5),       STAB1130
     4SAVMSZ(5),SAVDSX(10),SAVDSY(10),SAVDSZ(10),IQUOUT,GSINZT,         STAB1140
     5GCOSZT,STAB,STABVL,ISTAB,JCKSAB,VELX,VELY,VELZ,SAVMSY(5)          STAB1150
      COMMON                                                            STAB1160
     1     SMXMSC(5),   TMXMSC(5),   SMXMST(5),   TMXMST(5),            STAB1170
     2     SMXDSC(10),  TMXDSC(10),  SMXDST(10),  TMXDST(10)            STAB1180
     3    ,SLNGMS(5)  ,SLNGDS(10),  CURDSL,  INLEG  ,  IFPPRT,          STAB1190
     4     IMPACT(5)  ,IPRTFP(5),   KOUNT(5)   ,ANGX,   ANGY,   ANGZ    STAB1200
C                                                                       STAB1210
C     SUBROUTINE INITIALIZATION                                         STAB1220
C                                                                       STAB1230
      DO 50  I=1,NOLEG                                                  STAB1240
   50 ANG(I)=2000.                                                      STAB1250
      ANX=GSINZT*YSD                                                    STAB1260
      ANY=-(GCOSZT*ZSD+GSINZT*XSD)                                      STAB1270
      ANZ=GCOSZT*YSD                                                    STAB1280
      AMAG=SQRT(ANX**2+ANY**2+ANZ**2)                                   STAB1290
      ANX=ANX/AMAG                                                      STAB1300
      ANY=ANY/AMAG                                                      STAB1310
      ANZ=ANZ/AMAG                                                      STAB1320
C                                                                       STAB1330
C     DETERMINE FOOTPAD POSITIONS RELEATIVE TO PLANE OF MOTION          STAB1340
C                                                                       STAB1350
      DO 100 I=1,NOLEG                                                  STAB1360
      DX(I)=XFPS(1,I)-XS                                                STAB1370
      DY(I)=YFPS(1,I)-YS                                                STAB1380
      DZ(I)=ZFPS(1,I)-ZS                                                STAB1390
      DDOTN=ANX*DX(I)+ANY*DY(I)+ANZ*DZ(I)                               STAB1400
      SGN(I)=SIGN(1.,DDOTN)                                             STAB1410
      IF(ABS(DDOTN).LE.1.E-05)SGN(I)=0.                                 STAB1420
  100 CONTINUE                                                          STAB1430
C                                                                       STAB1440
C     DETERMINE STABILITY ANGLE                                         STAB1450
C                                                                       STAB1460
      DO 200 I=1,NOLEG                                                  STAB1470
      IF(SGN(I).NE.0.)GO TO 110                                         STAB1480
      XL=DX(I)                                                          STAB1490
      YL=DY(I)                                                          STAB1500
      ZL=DZ(I)                                                          STAB1510
      GO TO 120                                                         STAB1520
  110 I1=I+1                                                            STAB1530
      IF(I.EQ.NOLEG)I1=1                                                STAB1540
      ASGN=SGN(I)*SGN(I1)                                               STAB1550
      IF(ASGN.NE.-1.0)GO TO 200                                         STAB1560
      ANUM=ANX*DX(I)+ANY*DY(I)+ANZ*DZ(I)                                STAB1570
      ADEN=ANZ*(DX(I1)-DX(I))+ANY*(DY(I1)-DY(I))+ANZ*(DZ(I1)-DZ(I))     STAB1580
      ALAMB=-ANUM/ADEN                                                  STAB1590
      XL=DX(I)+ALAMB*(DX(I1)-DX(I))                                     STAB1600
      YL=DY(I)+ALAMB*(DY(I1)-DY(I))                                     STAB1610
      ZL=DZ(I)+ALAMB*(DZ(I1)-DZ(I))                                     STAB1620
  120 CONTINUE                                                          STAB1630
      GDOTL=-GCOSZT*XL+GSINZT*ZL                                        STAB1640
      ANG(I)=ACOS(  GDOTL   /(GRAV*SQRT(XL**2+YL**2+ZL**2)))             STAB1650
      GCLDN=GSINZT*YL*ANX-(GCOSZT*ZL+GSINZT*XL)*ANY+GCOSZT*YL*ANZ       STAB1660
      ANG(I)=ANG(I)*SIGN(1.,GCLDN)                                      STAB1670
```

```
  200 CONTINUE                                                          STAB1680
C                                                                       STAB1690
C     DETERMINE LANDER STABILITY                                        STAB1700
C                                                                       STAB1710
      KOUNT=0                                                           STAB1720
      DO 300 I=1,NOLEG                                                  STAB1730
      IF(ANG(I).EQ.2000.) GO TO 301                                     STAB1740
      IF(ANG(I).GT.0.)GO TO 320                                         STAB1750
      GO TO 300                                                         STAB1760
  301 KOUNT=KOUNT+1                                                     STAB1770
  300 CONTINUE                                                          STAB1780
      IF(KOUNT .EQ.NOLEG)GO TO 310                                      STAB1790
C     PITCH INSTABILITY                                                 STAB1800
      ISTAB=1                                                           STAB1810
      RETURN                                                            STAB1820
C     YAW INSTABILITY                                                   STAB1830
  310 CONTINUE                                                          STAB1840
      ISTAB=2                                                           STAB1850
      RETURN                                                            STAB1860
C     LANDER STABLE                                                     STAB1870
  320 CONTINUE                                                          STAB1880
      STAB=ANG(I)*57.295779513                                          STAB1890
      ROTX=DC(1,1)*WX+DC(1,2)*WY+DC(1,3)*WZ                             STAB1900
      ROTY=DC(2,1)*WX+DC(2,2)*WY+DC(2,3)*WZ                             STAB1910
      ROTZ=DC(3,1)*WX+DC(3,2)*WY+DC(3,3)*WZ                             STAB1920
      STABVL=(ROTX*ANX+ROTY*ANY+ROTZ*ANZ)*57.295779513                  STAB1930
      STABVL=SIGN(1.,(STAB-PRSTAB))*ABS(STABVL)                         STAB1940
      IF(TIME.EQ.0.0)STABVL=0.0                                         STAB1950
      PRSTAB=STAB                                                       STAB1960
      RETURN                                                            STAB1970
      END                                                               STAB1980
```

```
      SUBROUTINE MATINV ( A, B, N )
C
C         COMPUTE THE SIMPLE MATRIX INVERSE OF A AND STORE IT IN B
C              A IS NOT ALTERED
C              A MUST BE NON-SINGULAR
C              A SHOULD HAVE DOMINANT DIAGONAL ELEMENTS AND BE OF
C                   SMALL ORDER.  IF THESE CONDITIONS ARE NOT MET,
C                   THEN A DIFFERENT INVERSE ROUTINE SHOULD BE USED.
C         A   = MATRIX  OF ORDER (N,N)
C         B   = INVERSE OF ORDER (N,N)
      DIMENSION A(N,1), B(N,1)
C         COPY A TO B
      DO  100  I = 1, N
      DO  100  J = 1, N
  100 B(I,J) = A(I,J)
C
C         CONVERT B TO INVERSE BY G.E. ON   BI WHERE I IS OVER B
C
C         IP = THE COL. BEING REDUCED TO ZEROS
      DO  400  IP = 1, N
      H      =       B(IP,IP)
C         FROM HERE ON CONSIDER THE IP COL. OF B TO BE THE IP COL. OF I
      B(IP,IP) = 1.
C         REDUCE DIAGONAL ELEMENT OF B TO 1, UPDATE I
      DO  200  J  = 1, N
  200 B(IP,J) = B(IP,J) / H
C         REDUCE THE IP COL. OF B TO ZERO
      DO  400  J  = 1, N
      IF ( J .EQ. IP)GO TO 400
C         UPDATE IP COL. OF I
      H = B(J,IP)
      B(J,IP)  = 0.0
      DO  300  K  = 1, N
      B(J,K) = B(J,K) - B(IP,K)*H
  300 CONTINUE
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GMPRD(A,B,R,N,M,L)
C
C     THIS SUBROUTINE MULTIPLIES TWO GENERAL MATRICES TO FORM A GENERAL
C     RESULTANT MATRIX
C
      DIMENSION A(1),B(1),R(1)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

THIS PAGE INTENTIONALLY LEFT BLANK

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C. 20546