NASA CR-66791

N69-30864

MASTER AGREEMENT

TASK ORDER ONE

Analyses and Limited Evaluations of Payload and Landing System Structures For Intermediate Type Planetary Landers

By O.R. Otto, D.J. Dorr,
C.D. Goodman, and R.M. Laurenson

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared Under Contract No. NAS 1-8137 by

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
EASTERN DIVISION
Saint Louis, Missouri

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

ANALYSES AND LIMITED EVALUATIONS OF
PAYLOAD AND LANDING SYSTEM STRUCTURES
FOR INTERMEDIATE TYPE PLANETARY LANDERS

This report presents a description of structural design and six degrees of freedom loads and motions computer programs developed for investigation of planetary landers in the intermediate landing load factor range (50 to 300 earth g units). Limited evaluations of the landing characteristics of crushable torus and inflatable torus lander concepts were conducted employing the computer programs. Both the crushable torus and the inflatable torus structural design programs have been shown to be capable of satisfactorily establishing configurations for specified criteria. Utilizing baseline configurations established with the structural design programs, the capability of the landing loads and motions programs to predict six degrees of freedom motions was demonstrated.

TABLE OF CONTENTS

1. SUMMARY

This report presents a description of computer programs developed by McDonnell Douglas Astronautics Company - Eastern Division under NASA contract NAS-1-8137-1(U) for investigation of planetary landers in the intermediate landing load factor range (50 to 300 earth g's). Crushable torus and inflatable torus lander concepts were judged most capable of meeting design constraints for this landing load factor range. Landing loads and motions and the efficient structural design of these concepts may be determined by these programs:

- The Crushable Torus Structural Design Program can be used to determine attenuator dimensions required to satisfy a given set of landing conditions, accommodating either spherical or toroidal shaped landers. This program, presented in Appendix A, may also be used to determine velocity capability and load factor for any unidirectional landing attitude between flat and end landing.
- The Crushable Torus Landing Loads and Motions Program, presented in Appendix B, allows evaluation of landing loads and six degrees of freedom motions of the crushable torus or crushable sphere.
- The Inflatable Torus Structural Design Program, presented in Appendix C, can be used to determine inflatable torus dimensions, internal pressures, and thicknesses based on input payload, velocity, and load factor requirements. Velocity capability and load factor for any desired unidirectional landing attitude may also be determined.

- The Inflatable Torus Landing Loads and Motions Program, presented in Appendix D, allows evaluation of landing loads, and six degrees of freedom motions of the inflatable torus.

An inflatable torus configuration and a crushable torus configuration meeting the specified planetary landing system constraints for the intermediate range of landing load factors were established by exercising the computer programs. Analyses and evaluations of these baseline configurations are presented in this report.

Inflatable systems allow considerably more flexibility than crushable systems in achieving desired load factor in the intermediate landing category. The lowest landing system weights for the specific intermediate lander load factor range of 50 to 300 g's were achieved with an inflatable torus employing a relatively flat payload. For the crushable systems investigated the lowest load factors were achieved with a spherical lander, while the lowest landing system weights were achieved with a crushable torus employing a relatively flat payload.

2. INTRODUCTION

Landing systems considered for unmanned planetary landers have been generally categorized by landing load factor range. Landing load factors greater than 300 earth g's are considered hard landers, and usually employ various forms of parachute decelerators to achieve desired terminal velocity. Soft landing of payloads requires the use of retro rockets to reduce approach velocity to yield landing load factors less than 50 earth g's.

A planetary mission with objectives based on a landing in the intermediate lander load factor range (50 to 300 earth g's) will require the delivery vehicle to be brought to some small velocity at a specified altitude above the planet surface at which time the payload and landing system will be ejected to fall freely to the planet surface. Landed payload for such a landing would contain instrumentation less fragile than for a soft landing, but the payload instrumentation would not have to be hardened to the point where weight is sacrificed. The prime requirement is that the instrumentation be capable of surviving the load levels associated with an intermediate type landing.

Landing systems appropriate to the intermediate landing category include various configurations of pneumatic systems (gas bags) and crushable systems. Since the terminal descent mode may result in both horizontal and vertical velocities at touchdown, it is important that each landing system have omnidirectional impact capabilities. Moreover, minimum landing system weights and stowage volumes are also important factors to be considered.

This report presents the results of planetary lander studies performed by McDonnell Douglas Astronautics Company - Eastern Division under Task Order One of Master Agreement Contract NAS-1-8137-1(U) issued by NASA (Langley Research Center). The lander configurations were determined by exercising computer programs developed under this contract for the intermediate range of landing load factors. These programs, which predict landing loads and motions and the efficient structural design of payload and landing impact system structures, are presented in appendices to this report along with instructions for their use.

Goals of the study were to evaluate candidate intermediate category planetary lander concepts, to study in depth the most promising crushable concept and inflatable concept, and to develop computer programs essential for design of these landers. Subsequent to selection of a crushable torus and an inflatable torus as the most promising concepts, the development of structural design and six degrees of freedom motions computer programs for each concept became the major objective of the study.

3. STRUCTURAL DESIGN CRITERIA

The following factors were considered in landing system and payload structure design: simplicity, reliability, stowability, structural compatibility, environmental compatibility, weight, and sterilizability. Methods were provided for accomplishing post-landing payload exposure to permit operation of science experiments such as 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.

3.1 INTERMEDIATE LANDER DESIGN CONSTRAINTS - The following specified design constraints were used in analysis of lander concepts but do not necessarily represent computer program constraints:

(1) The landing system has the capability to successfully land on surfaces with slopes of ± 34 degrees maximum relative to local horizontal, containing particles varying in size from sand to 5 inch diameter rocks.

(2) Postlanding orientation is provided for positive vertical axis alignment of the payload within ± 40 degrees of the local gravity vector.

(3) Materials considered for use in the structure are compatible with space environment and a maximum temperature range from $-100^{\circ}F$ to $+275^{\circ}F$.

(4) The atmospheric pressure at the surface was assumed to be 5 mb.

(5) The landing surface was assumed not to deform or absorb energy during impact.

(6) The surface coefficient of friction was considered within the range of zero to 0.3.

(7) Surface gravitational acceleration was assumed to be 12.3 ft/sec^2.

(8) Mass of the landed vehicle was 26 slugs, or less.

(9) The landed vehicle (payload, payload structure, and landing impact system) is compatible with a nine (9) foot base diameter, 120 degree blunted cone.

(10) The landing payload can be enclosed in a surface of revolution and has a size compatible with a maximum payload density of 60 lb/ft^3 and Item (9) cited above.

(11) Touchdown was assumed to occur within a flight path angle from 0 degrees to 30 degrees relative to the gravity vector and with a flight path velocity of 85 fps. It was assumed that this range of horizontal velocities covers the effects of side drift due to winds.

(12) The landing vehicle is unrestricted in directional orientation about the pitch, roll, or yaw axes.

(13) Payload decelerations equivalent to 50, 150, and 300 g's (Earth units) were investigated.

3.2 FACTORS OF SAFETY - The following factors were applied to the maximum loads (limit loads) encountered in planetary landing within the constraints specified in Section 3.1:

Crushable Material	1.00
Inflatable Burst Pressure Factor	2.50
Inflatable Bag Material Seam Factor	1.18

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

Additional stroke was provided to absorb a 25 percent increase in the design kinetic energy of the crushable and inflatable torus designs. Rocks were allowed to violate payload insulation as long as they did not bottom out on the payload itself, since the insulation can be deformed with very little force and thermal protection should not be seriously impaired.

3.3 COORDINATE SYSTEM - The general coordinate system used in both the crushable and inflatable loads and motions programs is defined in Figure 3.3-1. A detailed description of the system may be found in Section B.2.1 (Appendix B).

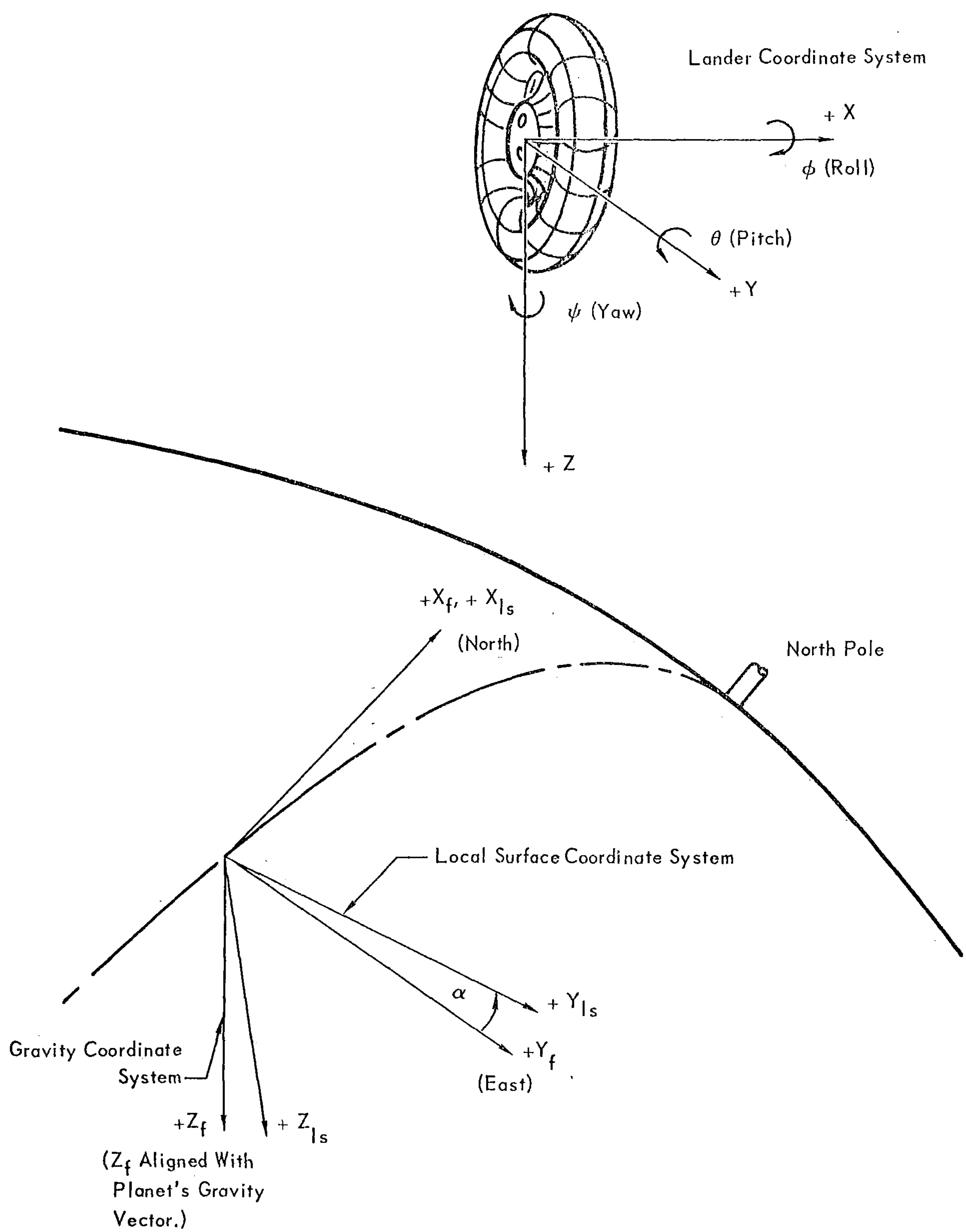

Lander Coordinate System
+ X
ϕ (Roll)
θ (Pitch)
+ Y
ψ (Yaw)
+ Z
$+X_f$, $+X_{ls}$
(North)
North Pole
Local Surface Coordinate System
α
$+Y_{ls}$
$+Y_f$
(East)
Gravity Coordinate System
$+Z_f$
$+Z_{ls}$
(Z_f Aligned With Planet's Gravity Vector.)

Figure 3.3-1

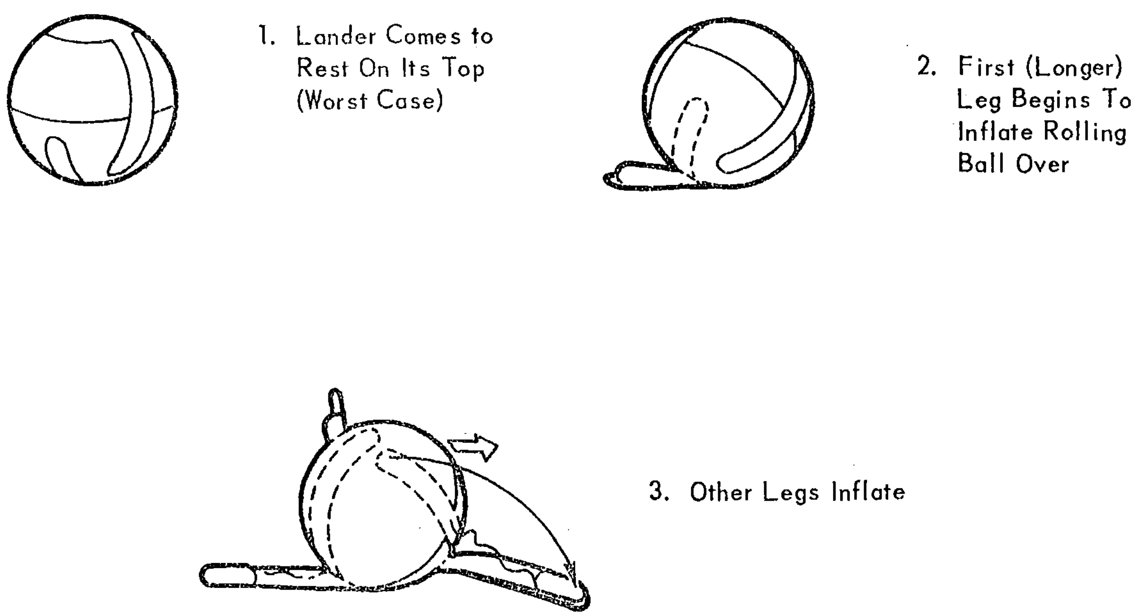
4. CONCEPT SELECTION

The crushable and inflatable concepts judged capable of meeting the intermediate category planetary lander design constraints are discussed in this section.

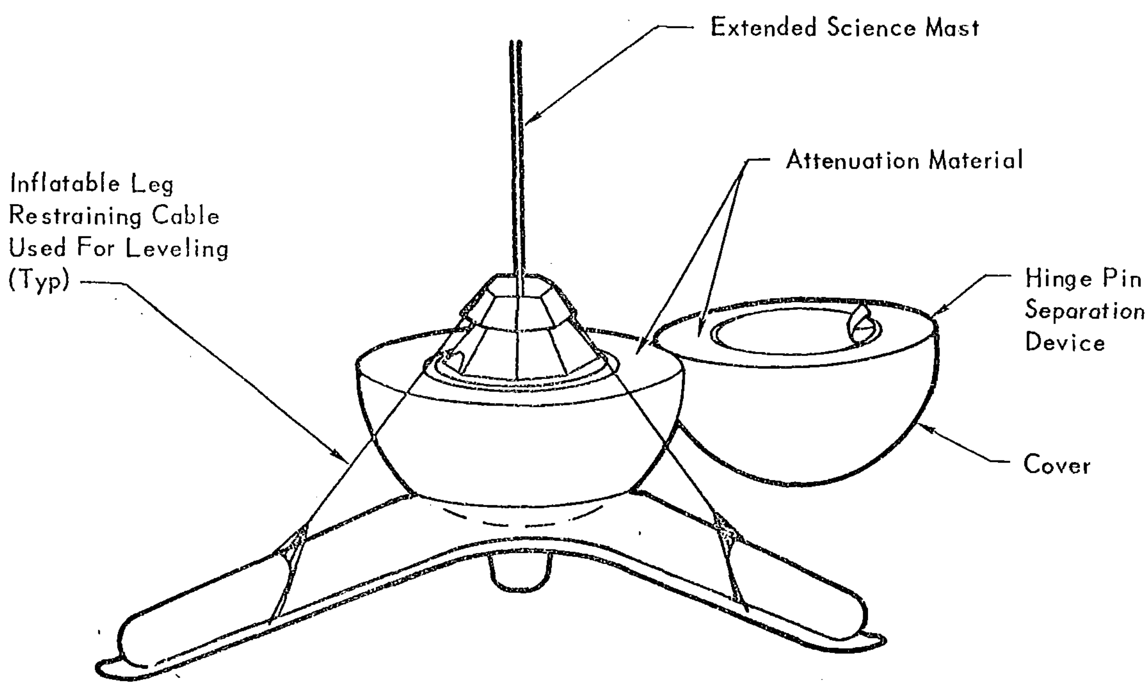
4.1 CRUSHABLE CONCEPTS STUDIED - Screening of concepts resulted in the elimination of many because of complexity or weight. Examples are spherical and toroidal landers employing mechanical erection techniques. The three concepts discussed in the following paragraphs were judged capable of meeting the constraints defined in Reference 1. They are the sphere, torus, and duo-hemisphere. The torus was selected for detailed analysis.

4.1.1 Sphere. - This concept, shown in Figure 4.1-1, consists of a spherical payload package surrounded by a crushable energy absorption material such as balsa wood or honeycomb. Erection and leveling is accomplished by three fabric legs wrapped around the exterior surface of the lander during impact. After landing, the legs are inflated causing the lander to be erected as illustrated in Figure 4.1-1. Leveling is accomplished by individually adjusting each leg with motor driven cable assemblies. A continuous, external cover is provided allowing this landing system to be designed for use over a wide range in landing velocities. Separation of this cover is required to permit removal of top hemisphere for exposing experiments. A simple "hinge pin" separation device is provided for this purpose.

CRUSHABLE SPHERE



LANDER ERECTING AND LEVELING SEQUENCE



LANDER POST-IMPACT CONFIGURATION

Figure 4.1-1

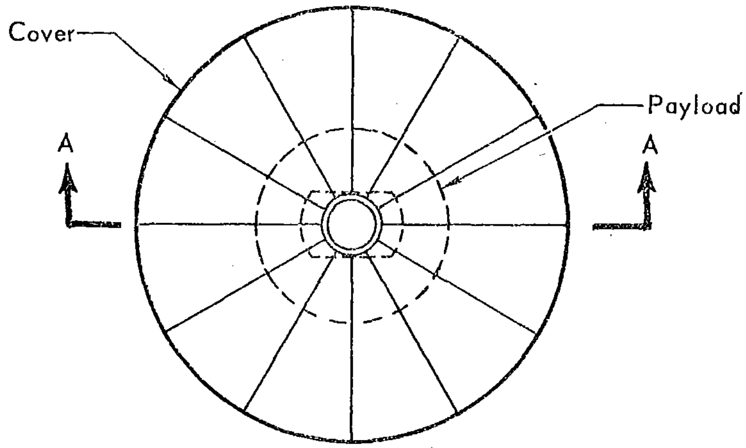
4.1.2 Torus. - This concept, shown in Figure 4.1-2, consists of a flat, cylindrical payload package surrounded at the edges by a toroidal shaped energy absorption material such as balsa wood or honeycomb. Because of its shape, the lander will come to rest on either of the flat sides. This necessitates either redundant experiments, flip-over of the entire lander, or gimballing of the deployable experiments. The latter approach is illustrated in Figure 4.1-2. Deployment of experiments is simplified because the center portion of payload is not covered with attenuant. A continuous external cover provides protection for the attenuant.

4.1.3 Duo-hemisphere Lander. - This concept, shown in Figure 4.1-3, consists of a cylindrical payload package protected by hemispherically shaped sections of energy absorption material located at each end. Because of its shape and offset center of gravity, the lander should come to rest on its side. Erection is accomplished by two fabric legs wrapped around the exterior surface of payload. After landing, the legs are inflated causing the entire lander to roll to an upright position for deployment of experiments. Spring actuated legs are deployed to stabilize the lander.

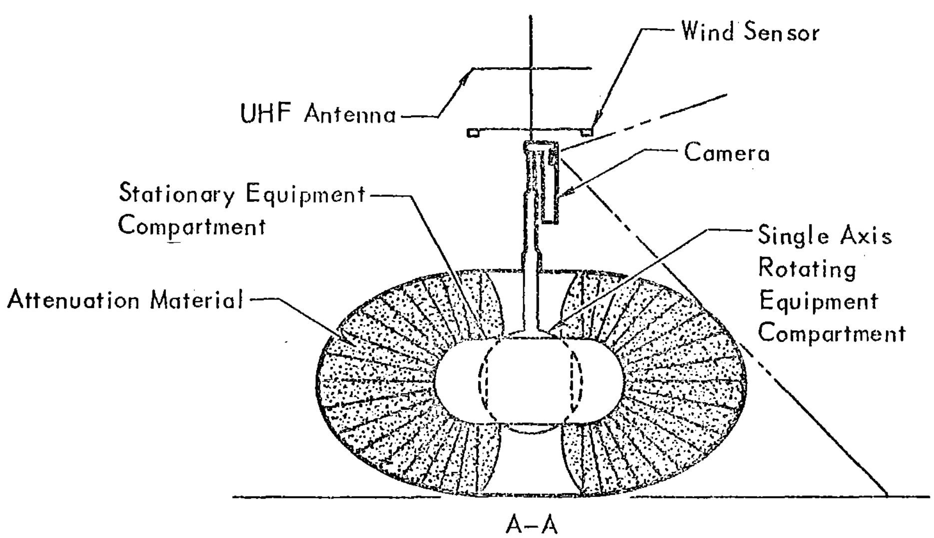
4.1.4 Evaluation. - Each concept was qualitatively evaluated based on previous study results of similar configurations. Comments regarding advantages and disadvantages of each configuration are presented in Figure 4.1-4.

4.1.5 Concept Selection. - Comparison of the three crushable concepts shows that each has individual characteristics that may be desirable for a specific payload and mission. The primary attributes of the sphere are;

CRUSHABLE TORUS



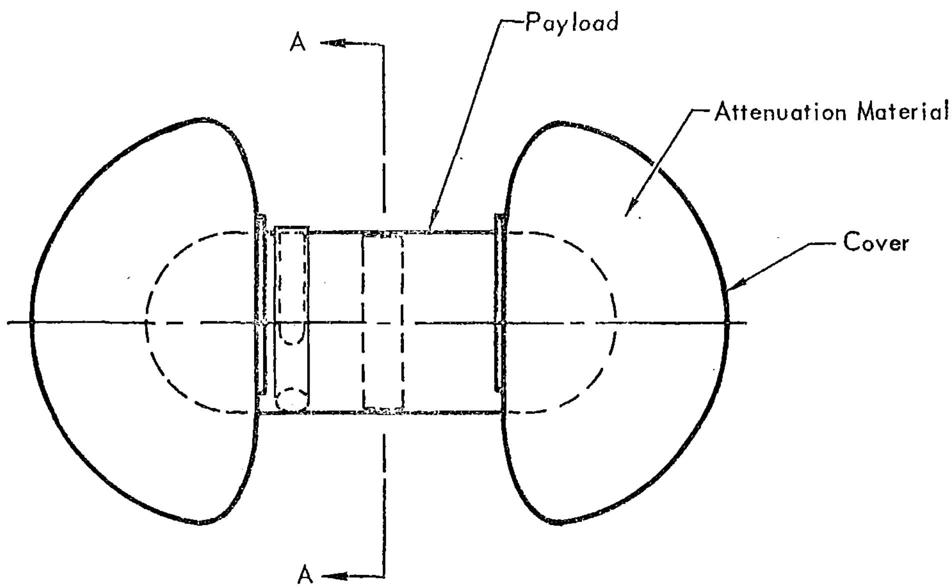
LANDER PRE-IMPACT CONFIGURATION



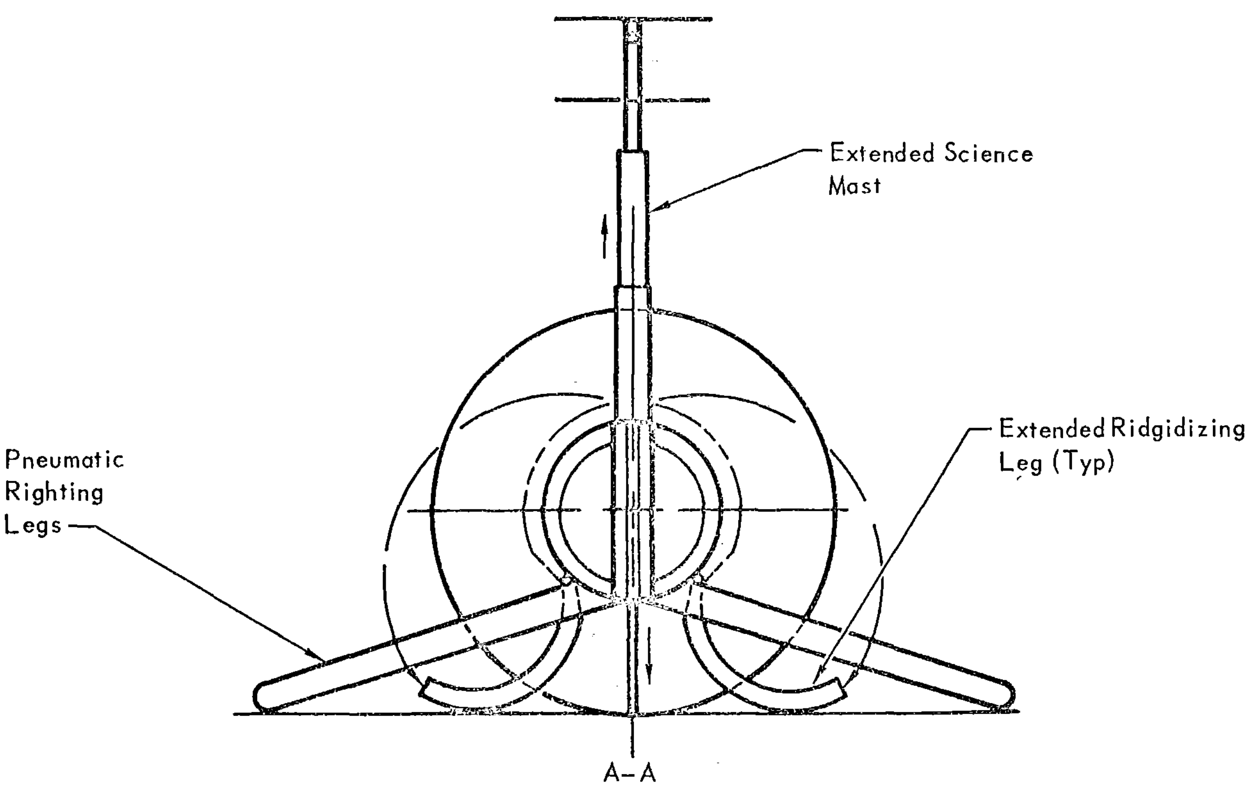
LANDER POST-IMPACT CONFIGURATION

Figure 4.1-2

DUO-HEMISPHERE LANDER



LANDER PRE-IMPACT CONFIGURATION



LANDER POST-IMPACT CONFIGURATION

Figure 4.1-3

ADVANTAGES AND DISADVANTAGES OF CRUSHABLE LANDING SYSTEMS

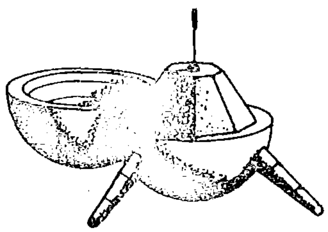
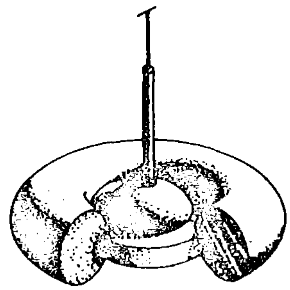
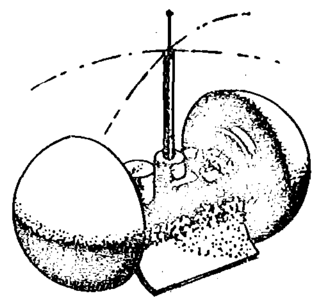
Configuration	 CRUSHABLE SPHERE	 CRUSHABLE TORUS	 DUO-HEMISPHERE
Advantages	1. Confidence in attenuation system because it is free of mechanisms. 2. Completely passive landing system. 3. Proven landing system feasibility. 4. Simple erection and leveling system. 5. Good visibility for surface experiments 6. Stows well in aeroshell. 7. Easy separation from aeroshell.	1. Confidence in attenuation system because it is free of mechanisms. 2. Completely passive landing system. 3. Payload shape offers efficient packaging of equipment. 4. Attenuant provides some protection from erosion by wind blown particles. 5. Easy to deploy soil sampling experiments. 6. Payload shape can be changed to accommodate particular experiments. 7. Stows well in aeroshell. 8. Easy separation from aeroshell.	1. Completely passive landing system. 2. Moderate landing load factors. 3. Payload dimensions can be changed to accommodate particular experiments. 4. Easy to deploy experiments. 5. Attenuant provides some protection from erosion by wind blown particles.
Disadvantages	1 Minimized landing system weight results in high landing load factors. 2. Erection and leveling system susceptible to puncture. 3. Mission success dependent on removing top hemisphere of attenuant. 4. Difficult equipment installation in spherical payload. 5. Difficult to deploy soil sampling experiments.	1. Bistable concept requires: a) Dual instrumentation, or, b) Gimballed payload or, c) Mechanism for turning lander over. 2. Minimized landing system weight results in high load factors. 3. Attenuant susceptible to crushing from inside by payload.	1. Concept may be limited to moderate or low velocities. 2. Attenuant susceptible to crushing from inside by payload. 3. Difficult to stow in aeroshell. 4. Erection system susceptible to puncture

Figure 4.1-4

high confidence level with this proven system, good stowability in the aeroshell, and good visibility for surface experiments. A torus is preferred if the paramount characteristics are efficient packaging of equipment, flexibility in packaging various experiments, and easy experiment deployment. The duo-hemisphere appears to offer somewhat lower landing load factors for equivalent landing system weight. Selection of the crushable torus concept as the landing system to be studied will allow investigation of both spheres and torii by appropriate dimensional variation. This will allow the broadest possible range of study of candidate crushable configurations for vehicles in the intermediate landing category. Therefore, we have chosen the crushable torus as the analytical model.

4.2 INFLATABLE CONCEPTS STUDIED - The four inflatable concepts discussed in the following paragraphs were judged capable of meeting the constraints defined in Reference (1). They are the sphere, single torus, triple torus, and multiple torus. The single torus was selected for detailed analysis.

4.2.1 Sphere. - This concept, shown in Figure 4.2-1, consists of a cylindrical payload suspended within an inflated spherical impact bag by numerous radial cords. An elastomer coats the fabric impact bag and provides both gas containment and scuff resistance. The payload is located off center, causing the lander to come to rest with the payload in an upright position. After the lander comes to rest, the impact bag is deflated and separated into two halves allowing the exposed payload to rest upright on the landing surface.

INFLATABLE SPHERE

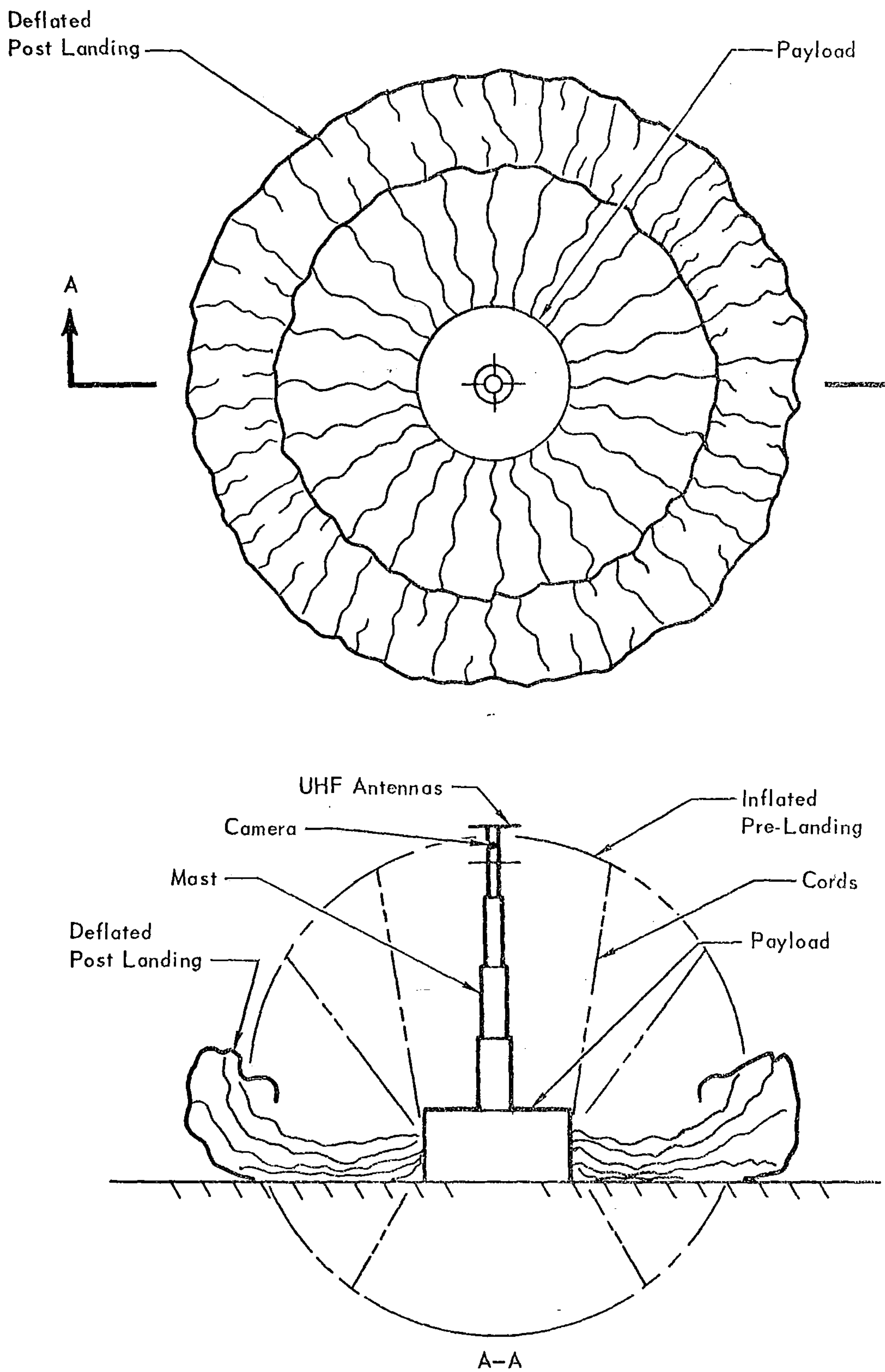


Figure 4.2-1

4.2.2 <u>Single Torus.</u> - This concept, shown in Figure 4.2-2, consists of a cylindrical payload mounted in the center of an inflated torus impact bag. An elastomer coats the fabric impact bag to provide gas containment and scuff resistance. The payload is trunnion mounted in a gimbal ring that is firmly attached to the torus. During landing, the payload is locked to the gimbal to prevent payload rotation and to provide continuous support to the payload. When the lander comes to rest, the gimbal is unlocked allowing the payload to rotate to an upright position. Rotation is accomplished by offsetting the payload center of gravity.

4.2.3 <u>Triple Torus.</u> - This concept, shown in Figure 4.2-3, consists of a cylindrical payload mounted in the center of three inflated impact bags. Two large torus bags are designed to attenuate landing loads and the third small torus is intended primarily to insure that the lander does not come to rest on its end. The fabric torus bags are coated with elastomer to provide gas containment and scuff resistance. The payload is trunnion mounted in a gimbal ring similar to the method used for the single torus.

Payload center of gravity is again offset allowing the payload to rotate to an upright position when unlocked. The upper torus bag is then deflated to provide greater exposure of the payload.

4.2.4 <u>Multiple Torus.</u> - This concept, shown in Figure 4.2-4, consists of a cylindrical payload supported by four peripheral beams. Four torus impact bags are attached to the ends of these beams. The fabric impact bags attenuate landing loads and are coated with elastomer to provide both gas

INFLATABLE SINGLE TORUS

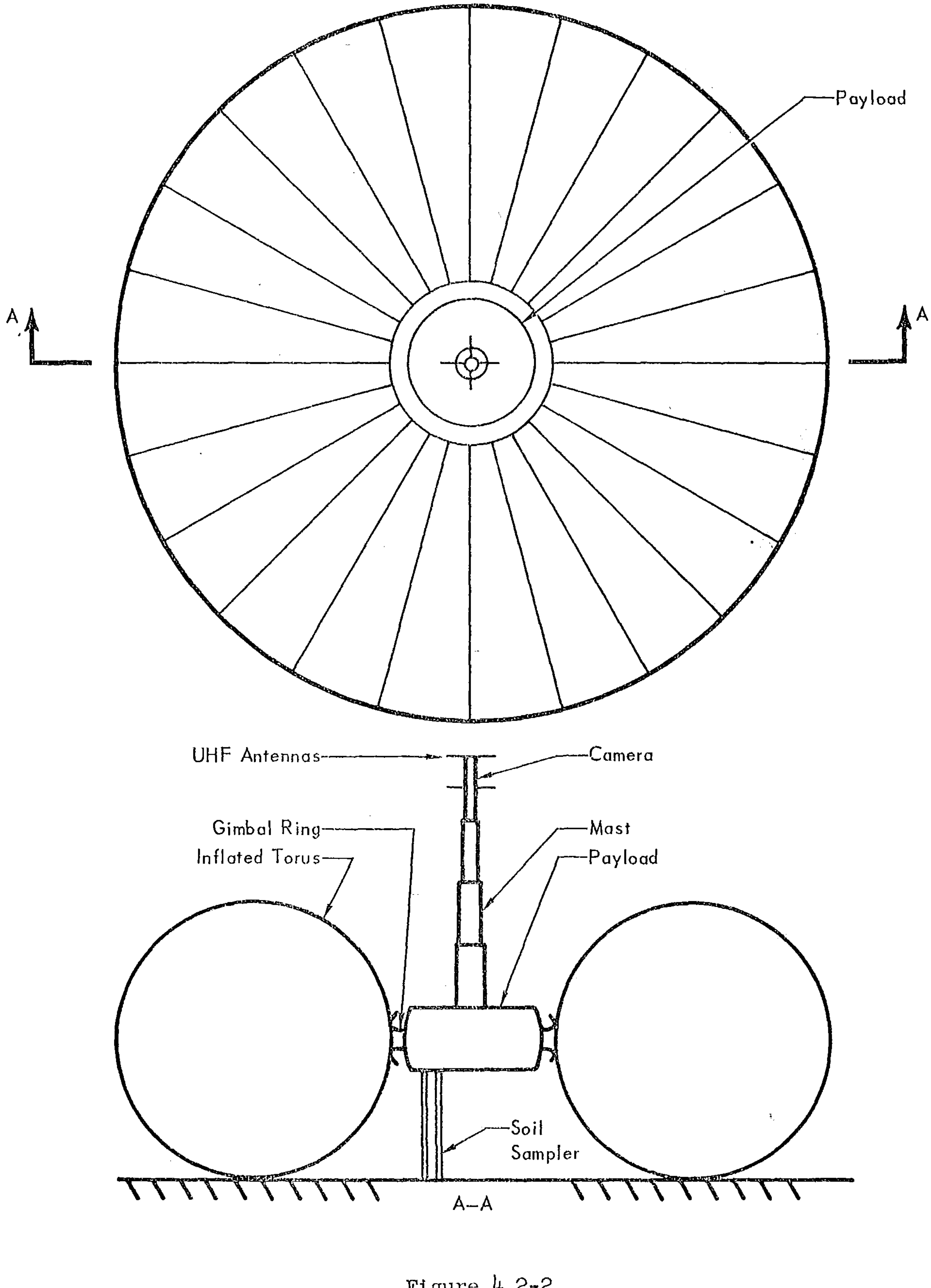
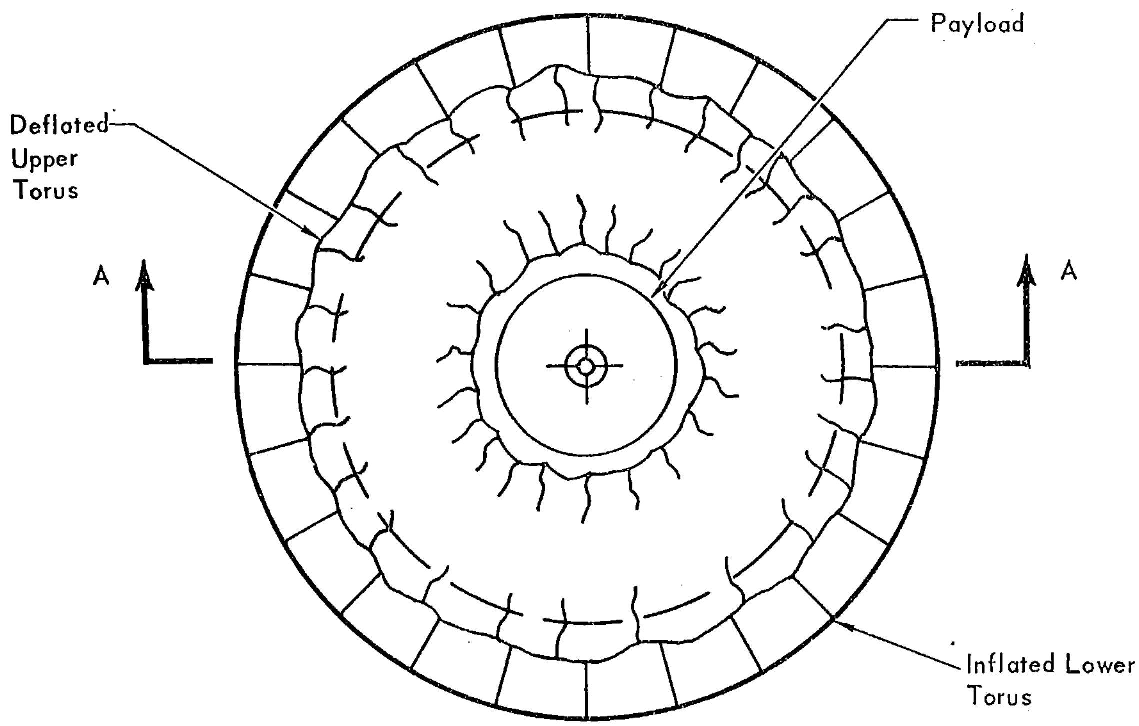
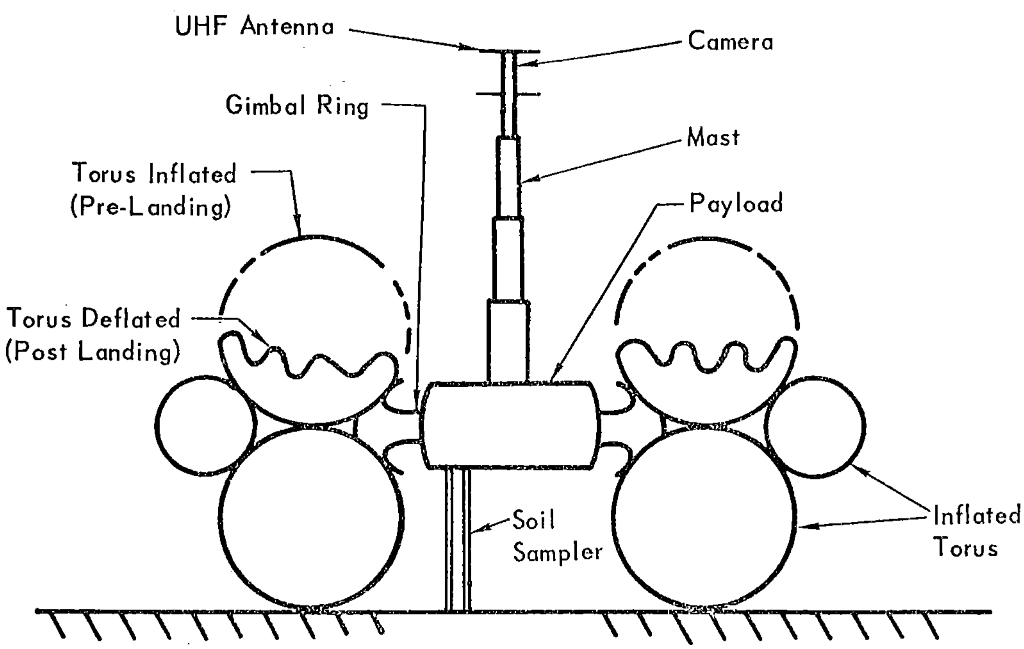


Figure 4.2-2

INFLATABLE TRIPLE TORUS





A–A

Figure 4.2-3

INFLATABLE MULTIPLE TORUS

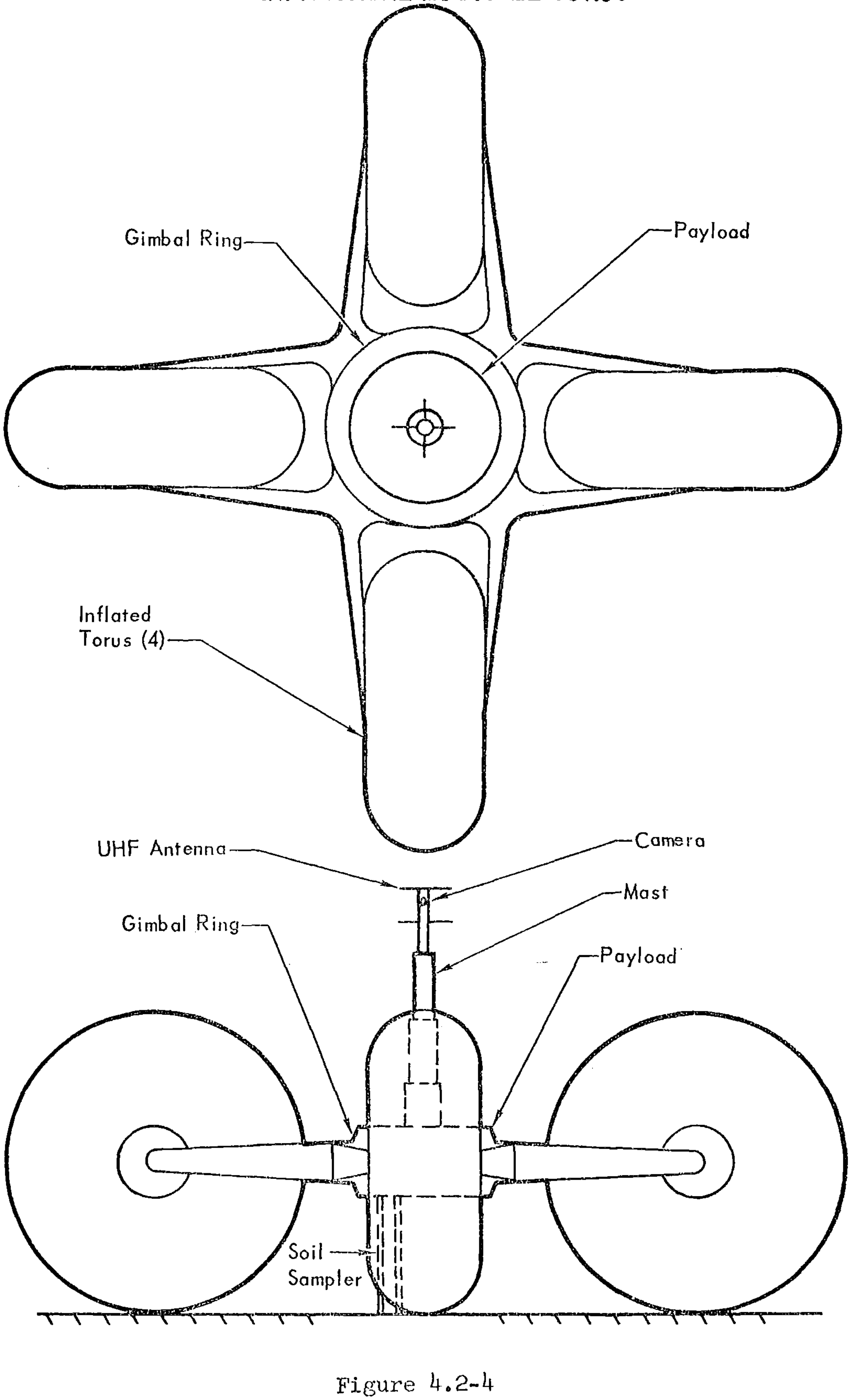


Figure 4.2-4

containment and scuff resistance. A trunnion mounted payload is used similar to the single torus configuration. Payload center of gravity is offset so that it swivels to an upright position when unlocked.

4.2.5 Evaluation. - The four inflatable concepts were qualitatively evaluated considering previous study results of similar configurations. Advantages and disadvantages of each configuration are presented in Figure 4.2-5.

4.2.6 Concept Selection. - The inflatable single torus concept was selected for detailed analysis. The primary reason for selecting this concept is that the designs for the impact bag and payload support structure are simple and straightforward. The spherical lander was not selected because of complexity associated with the methods for supporting and exposing the payload. Triple torus and multiple torus configurations are desirable if greater view angles and payload exposure are required. The added complexity of the triple torus concept and the high weight and poor stowability features of the multiple torus make both concepts somewhat undesirable. Study of the single torus will provide a basis for better understanding of the other pneumatic concepts.

ADVANTAGES AND DISADVANTAGES OF INFLATABLE LANDING SYSTEMS

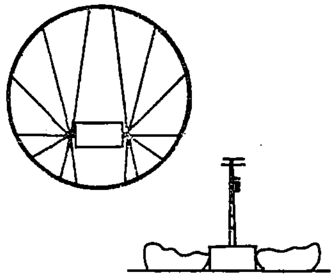
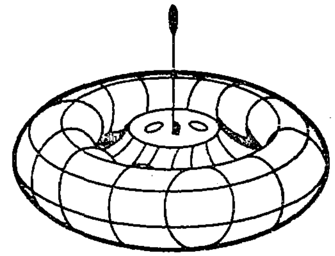
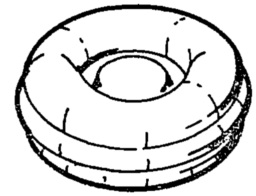
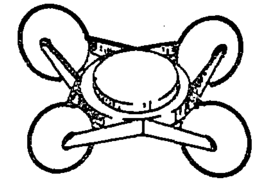
	SPHERE	SINGLE TORUS	TRIPLE TORUS	MULTIPLE TORUS
Configuration				
Advantages	1. Equipment is easily installed in cylindrical payload package. 2. Payload comes to rest in upright position without the use of gimbals. 3. Large footprint area allows a high degree of tolerance in surface conditions. 4. Experiments are easily deployed because of large exposed area of payload. 5. Equipment is easily insulated for thermal control.	1. Equipment is easily installed in cylindrical payload package. 2. Large footprint area allows a high degree of tolerance in surface conditions. 3. Lander is easily stowed within aeroshell. 4. Equipment is easily insulated for thermal control. 5. Simple and reliable landing system.	1. Equipment is easily installed in cylindrical payload package. 2. Large footprint area allows a high degree of tolerance in surface conditions. 3. Lander is easily stowed within aeroshell. 4. Equipment can be easily insulated for thermal control. 5. Good view angles for experiments.	1. Equipment is easily installed in cylindrical payload package. 2. Equipment is easily deployed because of large exposed area of payload 3. Equipment can be easily insulated for thermal control. 4. Good view angles for experiments.
Disadvantages	1. Scuff resistance is required over the entire bag surface. 2. Concept is susceptible to excessive rolling after impact. 3. Method of separating bag to expose payload is complicated and unreliable. 4. Surface conditions may restrict ability of lander to become upright. 5. Lander is difficult to stow within aeroshell.	1. Experiment deployment complicated because of limited view angles.	1. Requires complicated manifolding and valving system to inflate tori.	1. Landing system is heavy due to complex beam arrangement. 2. Requires complicated manifolding and valving system to inflate tori. 3. Concept is difficult to stow in aeroshell.

Figure 4.2-5

5. CRUSHABLE TORUS LANDING SYSTEM

This section describes the computer programs developed to predict landing loads and motions and the efficient structural design of crushable torus configurations. Studies performed using these programs leading to a selected crushable torus configuration are also presented in this section.

5.1 COMPUTER PROGRAMS

5.1.1 Crushable Torus Structural Design Program. - This program, which is discussed in detail in Appendix A, provides the capability for either establishing crushable torus configurations meeting design constraints or for evaluating existing configurations. Operation of this program requires initial selection of a payload, payload-attenuator overlap, attenuator material, and desired velocity capability.

To allow maximum flexibility the program was written in two parts. The configuration design part of this program is used to determine required attenuator dimensions providing the desired flat and end landing velocity capabilities. If it is desired to check the velocity capability of a configuration for any attitude between flat and end, the omnidirectional loads part of this program is used. This program is limited to constant attitude crushing.

Dimensional variables used to represent the payload and attenuator are shown in Figure 5.1-1. The payload is a disk with elliptically shaped circumferential surfaces. The attenuator is assumed to be oriented with its principal crushing axis perpendicular to the payload surface. Attenuator shape is

CRUSHABLE TORUS GEOMETRY

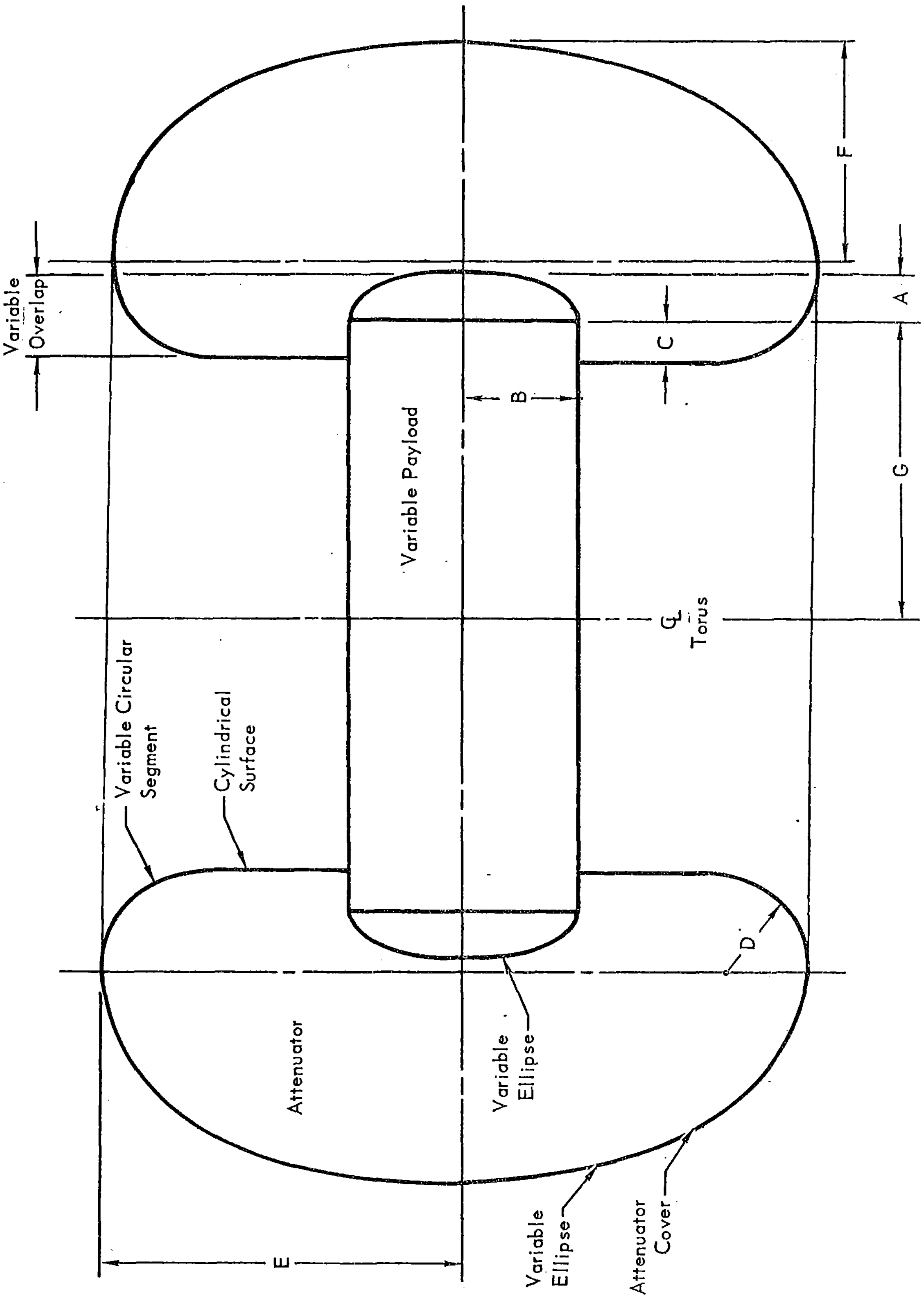


Figure 5.1-1

defined by elliptical, circular, and cylindrical shaped surfaces. By appropriate dimensional variation, crushable spheres can be investigated as well as various torus configurations.

Besides providing the ability to vary payload shape the program allows variation of the following parameters: attenuator material properties; rock diameter; friction coefficient; desired flat and end velocity capability; and inclination of the contact angle between the planet surface and the vehicle axes (constant angle throughout stroke). Load factor can be minimized by parametrically varying input payload shape, attenuator-payload overlap, and/or attenuator material until the actual internal crush force approaches the allowable internal crush force. Attenuator material properties used in the program are density and radial, shear, and transverse crush stress. In addition, the allowable shear stress between the payload and attenuator is input. An input exponential factor is used in the interaction equation relating shear and radial stress.

Output data include: required attenuator dimensions; payload weight; landing system weight (cover, adhesive, and attenuator material); and maximum load factor and velocity capability for flat landing, end landing, and landing at the input contact angle. Data output as a function of stroke for flat, end, and oblique landing include: footprint area; normal crushing force and friction force (input coefficient of friction assumed acting throughout stroke); normal crushing force for zero coefficient of friction; and maximum allowable friction force (with no normal force acting on the crush plane). With friction acting, a smaller normal force will crush the attenuator and a larger volume of attenuator material is required. Load factor is generally a maximum when there is no friction acting. Internal attenuator crushing (payload cannon balling) may be determined by comparing calculated values for actual internal crushing force and allowable internal crushing force (both flat and end landings).

5.1.2 Crushable Torus Landing Loads and Motions Program. - This program is used to predict time histories of landing loads and spatial motions for an intermediate landing vehicle utilizing a crushable torus landing system. The program was written to provide six degrees of freedom motions for the lander defined by the Structural Design Program discussed in Section 5.1.1. Variations in geometrical proportions of landing system and payload, identical to those defined in Section 5.1.1, are permissible. Any energy absorption material, having known mechanical and physical properties, such as honeycomb or foam, may be used. Strain energy stored in the attenuator, which can be recovered in the form of springback, is modeled also. Any density may be selected for the payload, and the program calculates the weight and inertias of the vehicle using the payload and attenuator densities input. The planet's surface conditions may be varied by selecting any rock diameter, any coefficient of friction, and any surface slope (positive or negative) less than 80°.

A very important feature of the program is its spatial motion capability. It can consider translational and rotational motions in any direction arising from the input initial conditions or the external loads and body forces on it. Program running time may be reduced by selectively locking out as many of the six degrees of freedom as desired by input indicator. These motions can be referenced to any of three coordinate systems, one in the vehicle and two on the surface, which are provided so that motions can be referenced to the most meaningful coordinate system, or, in some cases, to more than one system so that data can be more easily understood.

Any of three integration techniques can be selected for the numerical integration of the equations of motion: fourth order Runge - Kutta and fixed and variable step modified Adams - Moulton Predictor-Corrector. Three choices were provided because each technique has different advantages and disadvantages, and no one is optimum for all cases.

The program prints out three types of data; input parameters, time invariant values, and time histories. The time histories are printed out in blocks, one for each time point printed. The size of these blocks and the variables in them can be changed by input parameters. Most of the calculated values in the program are stored in the program common block (COMINT) (see Appendix E), and any variable in COMINT can be printed out.

The program has multiple case capability in a single run. It can run as many cases as desired, with only the changes between cases being read in for each succeeding case after the first.

Program termination occurs when one of up to eight different time varying parameters (simultaneously checked while the program is running) exceeds the input limit value.

The program is written to make modification and improvements easier. It is divided into subroutines so that calculations are in functional groups, and can be modified and improved separately. Additional subroutines, not used in the present program, are in the program deck and are provided for later inclusion, if desired, of aerodynamic forces and further environmental parameters. Capability is built into the equations of motion and the inertia subroutine to handle time varying vehicle moments and cross products of inertia, even though the program is not presently set up to use them since they are assumed to have negligible change. Finally, comment cards have been used liberally in the program to make it more understandable.

Landing conditions causing high torques and large rotational rates during attenuator crushing may create large rotations of the lander. For these conditions, the crush plane at any time may intersect a previously crushed plane. This violates the assumption mentioned in Section B.2.2 that the crush plane always contact virgin material. An analysis of such a case, presented in Section 5.7, showed that crush plane intersection occurred after the maximum stroke point had been reached as the lander was rebounding and rotating. During the period when the crush planes intersect, the program used a zero force although a small crushing force should have been acting. The impulse imparted to the lander by this force acting over a short time period would not appear to materially affect the motions of the lander. If the program is used to study landers and landing conditions where intersecting crush planes are anticipated, vehicle orientation relative to stroke should be checked to determine the extent of overlap and resulting accuracy of loads and motions.

5.2 PAYLOAD SHAPE COMPARISON - Payload and attenuator dimensions for the crushable torus are shown in Figure 5.1-1. There are two payload shape ratios which influence landing load factor and landing system weight: the ratio of the radius of the cylindrical portion of the payload to the payload half-height (G/B); and the ratio of the major to minor radii of the outer elliptical shaped portion of the payload (B/A). The influence of these ratios on landing system weight, end landing load factor, and flat landing load factor are shown in Figures 5.2-1, 5.2-2, and 5.2-3 respectively. These figures are for a constant payload weight of 500 pounds, velocity of 85 feet per second, 5.0 inch rock diameter, and coefficient of friction of 0.3.

As the parameter G/B is increased, landing system weight and end landing load factor are decreased, but flat landing load factor is increased (see curves for 4.5 lb/ft^3 styrofoam and B/A = 2.0). Since G/B ratios larger than 4, begin to exceed the allowable dimensions for stowage in the aeroshell, no larger G/B ratios than 4 were investigated. Internal crushing for all points on these curves was not critical and attenuator/payload overlap was minimal (C = 0.50 inch constant) minimizing the influence of this parameter. Use of a spherical shaped lander results in the lowest flat landing load factors due to its relatively small footprint area while crushing. Use of toroidal landers with a G/B ratio of 3 to 4 results in the lowest landing system weights and best payload exposure after landing.

INFLUENCE OF PAYLOAD SHAPE PARAMETERS ON LANDING SYSTEM WEIGHT

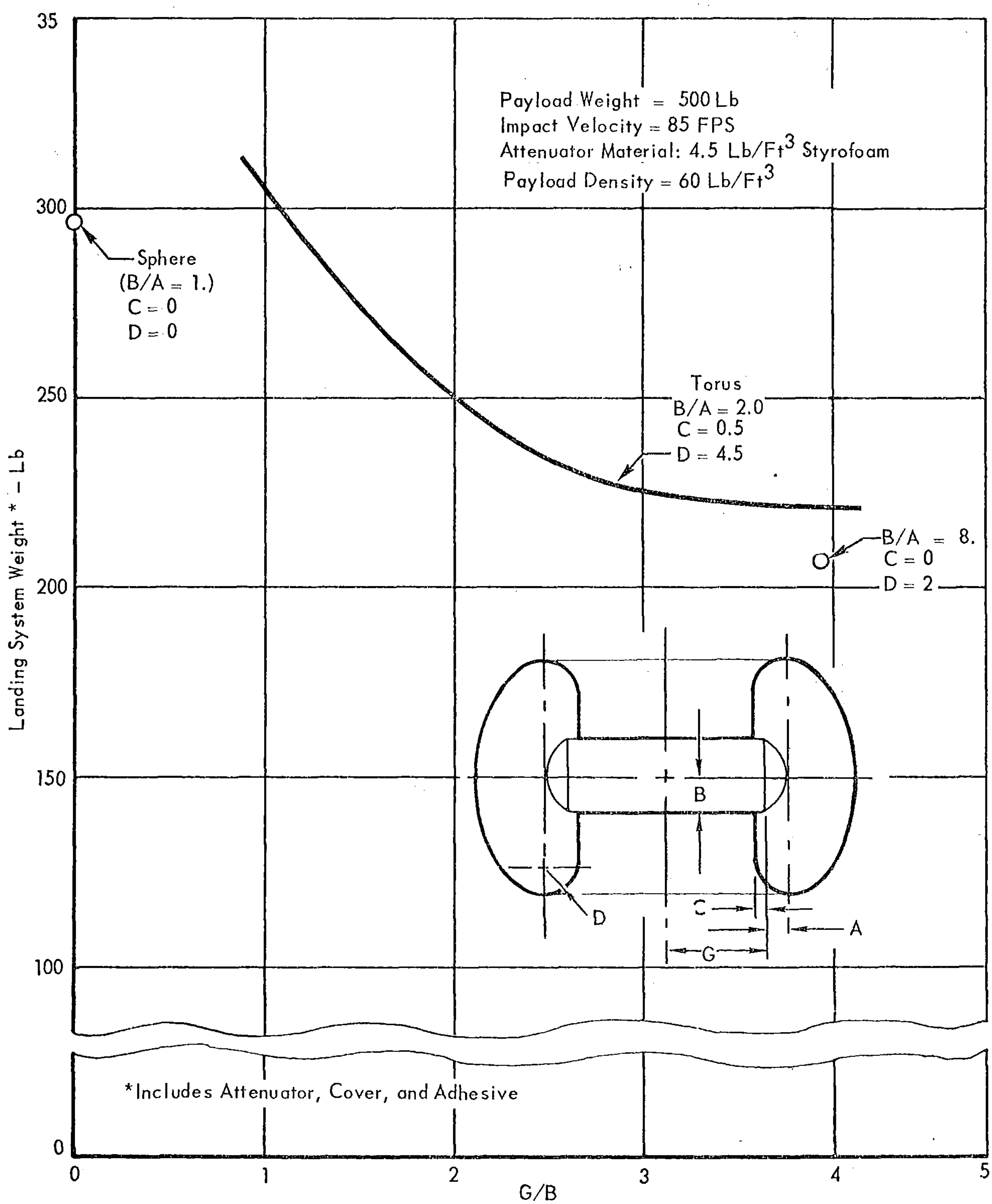


Figure 5.2-1

INFLUENCE OF PAYLOAD SHAPE PARAMETERS ON END LANDING LOAD FACTOR

Payload Weight = 500 Lb
Impact Velocity = 85 FPS
Attenuator Material: 4.5 Lb/Ft^3 Styrofoam
Payload Density = 60 Lb/Ft^3

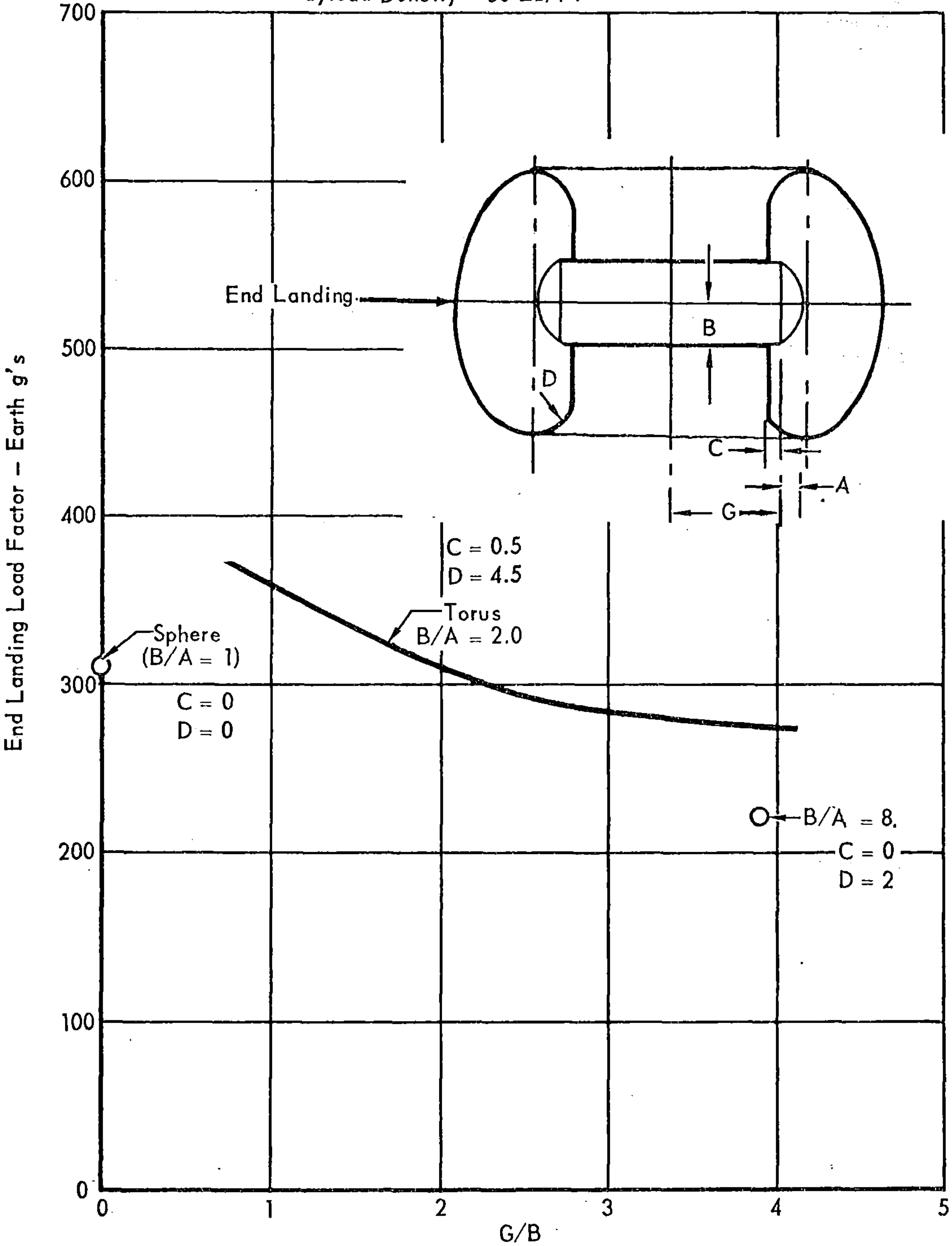


Figure 5.2-2

INFLUENCE OF PAYLOAD SHAPE PARAMETERS ON FLAT LANDING LOAD FACTOR

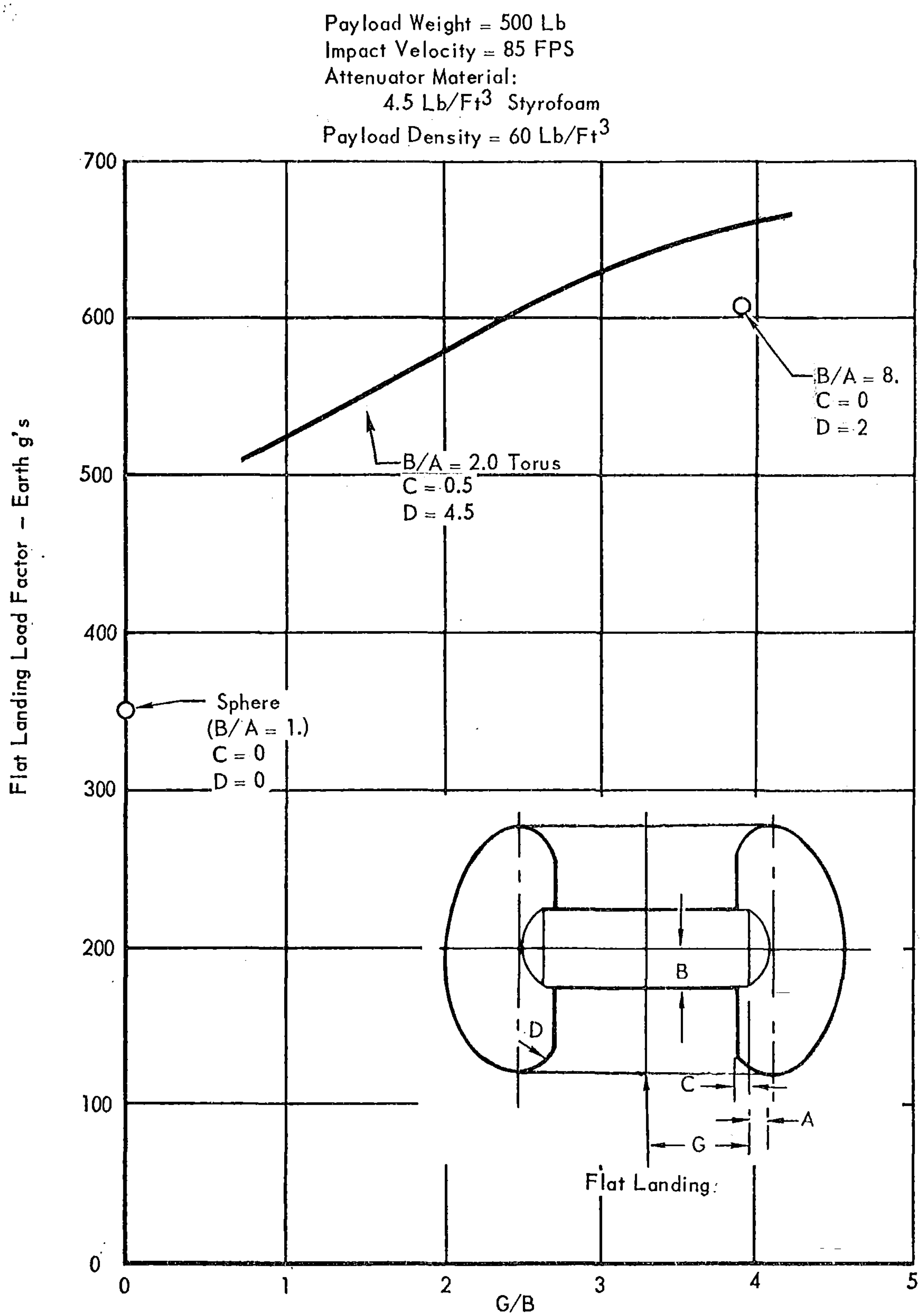


Figure 5.2-3

An increase in the payload shape parameter B/A causes a decrease in flat and end load factors and landing system weight (see curves for 4.5 lb/ft^3 styrofoam and G/B = 3.9). A B/A of 4.0 was selected as optimum. A payload with a flatter elliptical end may experience internal crushing, unless additional overlap (dimension C) is provided. For a given payload shape the load factor can be minimized by reducing dimension C until the actual internal crush force approaches the allowable internal crush force. Attenuator materials with low transverse crush stress are most prone to internal crushing since most of the loads from the payload and upper half of the attenuator must be carried through the lower half of the payload/attenuator bond.

5.3 ATTENUATOR MATERIAL COMPARISON - Candidate crushable attenuator materials for planetary landing systems are balsa wood, steel and aluminum honeycomb, aluminum trussgrid, fiberglass honeycomb, thermoplastic foams, and thermosetting foams. The significant material property affecting load factor is crushing stress, while the material property influencing landing system weight is specific energy.

To achieve a crushable landing system which meets the intermediate lander specified load factor range (50 to 300 earth g's) requires use of materials with relatively low crushing stress. For spherical shaped landers crushing stress should be less than about 250 psi, while for torus landers with a G/B = 3.9 and B/A = 4.0 the crush stress should be less than about 40 psi (see Figure 5.3-1). Thus very low density materials such as aluminum honeycombs, aluminum trussgrid, and foams are required for the intermediate lander load factor range since low density is the best means of achieving

EFFECT OF ATTENUATOR MATERIAL
CRUSH STRESS ON FLAT LANDING LOAD FACTOR

B/A = 4.0 G/B = 3.9 Torus

Payload Weight = 500 Lb
Impact Velocity = 85. FPS

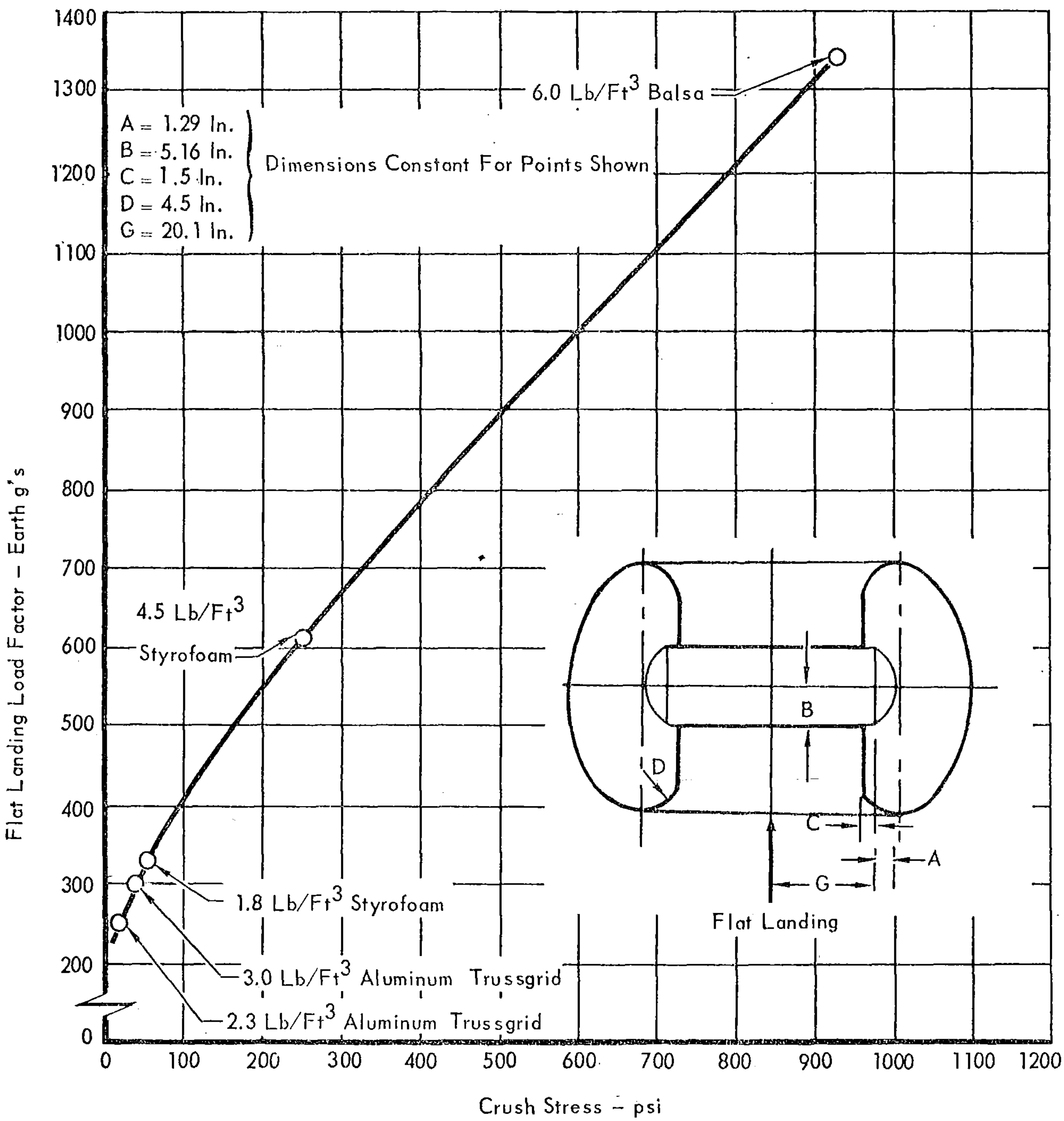


Figure 5.3-1

low crush stress.

Internal attenuator crushing (payload cannonballing) can be a critical factor for the low crush stress attenuator materials. Materials with little or no transverse strength require more payload/attenuator overlap since all load from the payload and upper attenuator must be carried by the lower attenuator/payload bond. Consequently, use of aluminum trussgrid and styrofoam, which have transverse crush strengths equal to the radial crush strength, require less overlap and generally minimize internal crushing problems.

Attenuator allowable shear stress associated with radial and transverse crush stress can influence load factor and landing system weight if design coefficient of friction is large. Application of shear to the attenuator reduces the stress at which normal crushing occurs, reduces load factor, but increases the amount of attenuator required to absorb the landing impact. Throughout this study a coefficient of friction of 0.3 was assumed to be acting constantly through the stroke. When friction was considered in the analysis, flat landing load factor for a typical case (3.0 lb/ft^3 aluminum trussgrid with allowable shear stress equal to 90 percent of the radial crush stress) was reduced by 5.7 percent while landing system weight was increased by 8.4 percent compared to the analysis with no friction. Load factor used for design would be the largest value obtained utilizing a specified coefficient of friction or a friction coefficient of zero.

Landing system weight as a function of attenuator material specific energy (XKW) is presented in Figure 5.3-2 for torus landers with a G/B of 3.9 and a B/A of 4.0. The balsa lander is shown to be the lightest of those

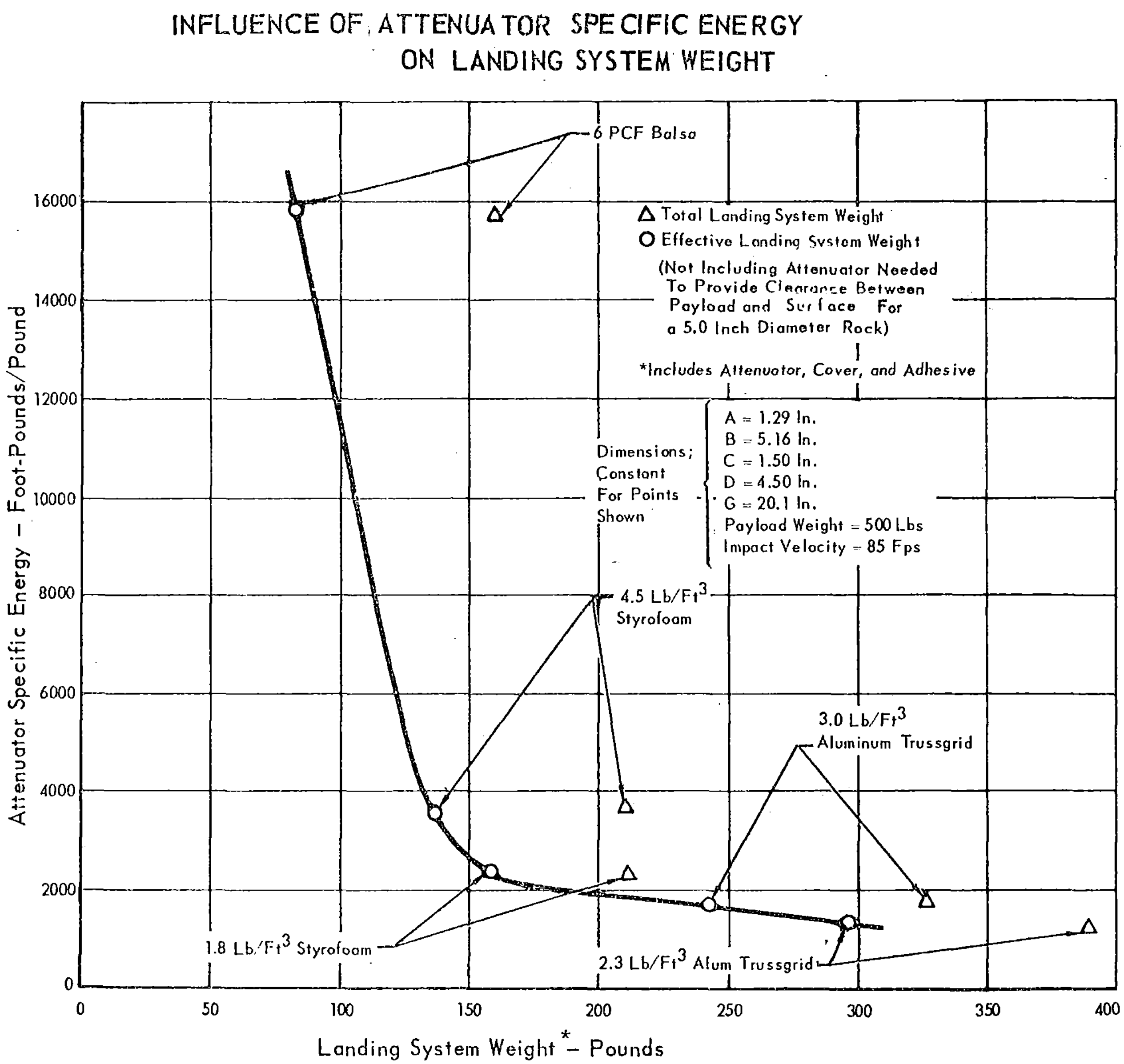

INFLUENCE OF ATTENUATOR SPECIFIC ENERGY
ON LANDING SYSTEM WEIGHT
6 PCF Balsa
△ Total Landing System Weight
O Effective Landing System Weight
(Not Including Attenuator Needed
To Provide Clearance Between
Payload and Surface For
a 5.0 Inch Diameter Rock)
*Includes Attenuator, Cover, and Adhesive
Dimensions;
Constant
For Points
Shown
A = 1.29 In.
B = 5.16 In.
C = 1.50 In.
D = 4.50 In.
G = 20.1 In.
Payload Weight = 500 Lbs
Impact Velocity = 85 Fps
4.5 Lb/Ft3
Styrofoam
3.0 Lb/Ft3
Aluminum Trussgrid
1.8 Lb/Ft3 Styrofoam
2.3 Lb/Ft3 Alum Trussgrid
Attenuator Specific Energy – Foot-Pounds/Pound
0
2000
4000
6000
8000
10000
12000
14000
16000
0
50
100
150
200
250
300
350
400
Landing System Weight* – Pounds

Figure 5.3-2

compared since it has a relatively high specific energy. A portion of the attenuator, required to provide clearance between the landing surface and payload, is not available to absorb energy. Weight of this portion is a function of attenuator density and volume providing the needed clearance. Therefore, both material density and specific energy influence landing system weight. The effect of density on weight can be seen by comparing the total and effective landing system weight for the 1.8 lb/ft^3 and 4.5 lb/ft^3 styrofoam (Figure 5.3-2).

5.4 SELECTED CONFIGURATION - The crushable torus concept was chosen because it offers efficient equipment packaging, flexible experiment packaging, easy experiment deployment, and relatively low landing system weight. A payload with shape parameters G/B = 3.9 and B/A = 4.0 optimizes these concept attributes within the aeroshell compatibility requirement (lander must stow in a 9 foot base diameter 120° blunted cone aeroshell). Of the low density attenuator materials studied, the 3.0 lb/ft^3 density aluminum truss-grid results in a lander design which has the lowest flat landing load factor without experiencing internal crushing, meets the total weight constraints, and stows well in the aeroshell. Use of a 1.8 lb/ft^3 styrofoam results in a lighter design, but slightly exceeds the desired load factor and is too large to stow well in a 9 foot base diameter aeroshell. Thus, the selected baseline configuration, detailed in Figure 5.4-1, meets the specified intermediate lander constraints although it is at the upper limit for load factor (300 earth g's).

CRUSHABLE TORUS BASELINE DESIGN

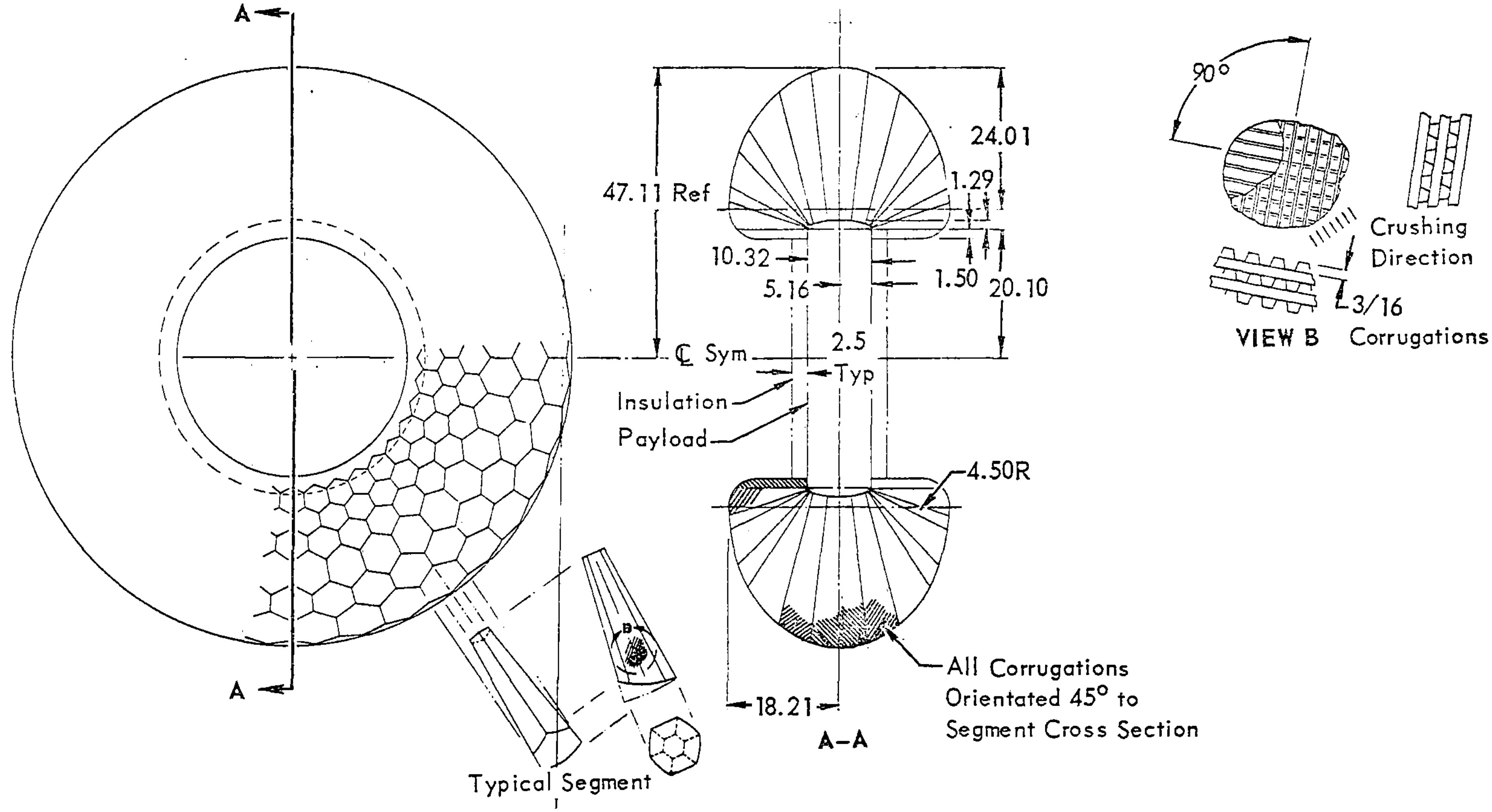


Figure 5.4-1

Notes:

Landing System Weight = 325.3 Lb
Payload Weight = 500 Lb
Total Weight = 825.3 Lb
Corrugation Density is 3.0 Lb/Ft^3
Fabricate Segments from 3/16 Corrugations, 0.002 Foil (5052)

To obtain lower landing load factors would require: reduction of landing velocity from 85 fps; use of attenuator materials with a lower crushing stress than 30 psi without violating internal crushing criteria; an increase in aeroshell base diameter; use of spherical shapes; and/or increase in the payload density by reducing payload volume or reducing payload size. All of the aforementioned means of reducing load factor either violate the specified criteria or result in undesirable payload packaging, exposure, and deployment.

The attenuator materials investigated in this study were those currently available and considered most promising within the specified constraints. Additional studies of low crushing stress materials would be desirable, particularly if the intermediate lander aeroshell base diameter is increased to accomodate a larger lander or if the payload is reduced in size.

The orientation of the aluminum trussgrid ribbon for a typical tapered segment is shown in Figure 5.4-1. This orientation provides maximum resistance to internal crushing for a flat landing. The tapered segments are bonded to each other and to the payload. A nomex cover is bonded on the outer surface of the attenuator to prevent shredding.

5.5 AEROSHELL COMPATIBILITY - An aeroshell installation drawing was made to determine the compatibility of the baseline crushable torus lander when installed in a 9 foot diameter aeroshell. This installation is shown in Figure 5.5-1. Major components of the assembly are a 9 foot aeroshell, the baseline crushable torus lander, parachute canister support truss, de-

CRUSHABLE TORUS AEROSHELL INSTALLATION
9.0 FT DIAMETER AEROSHELL

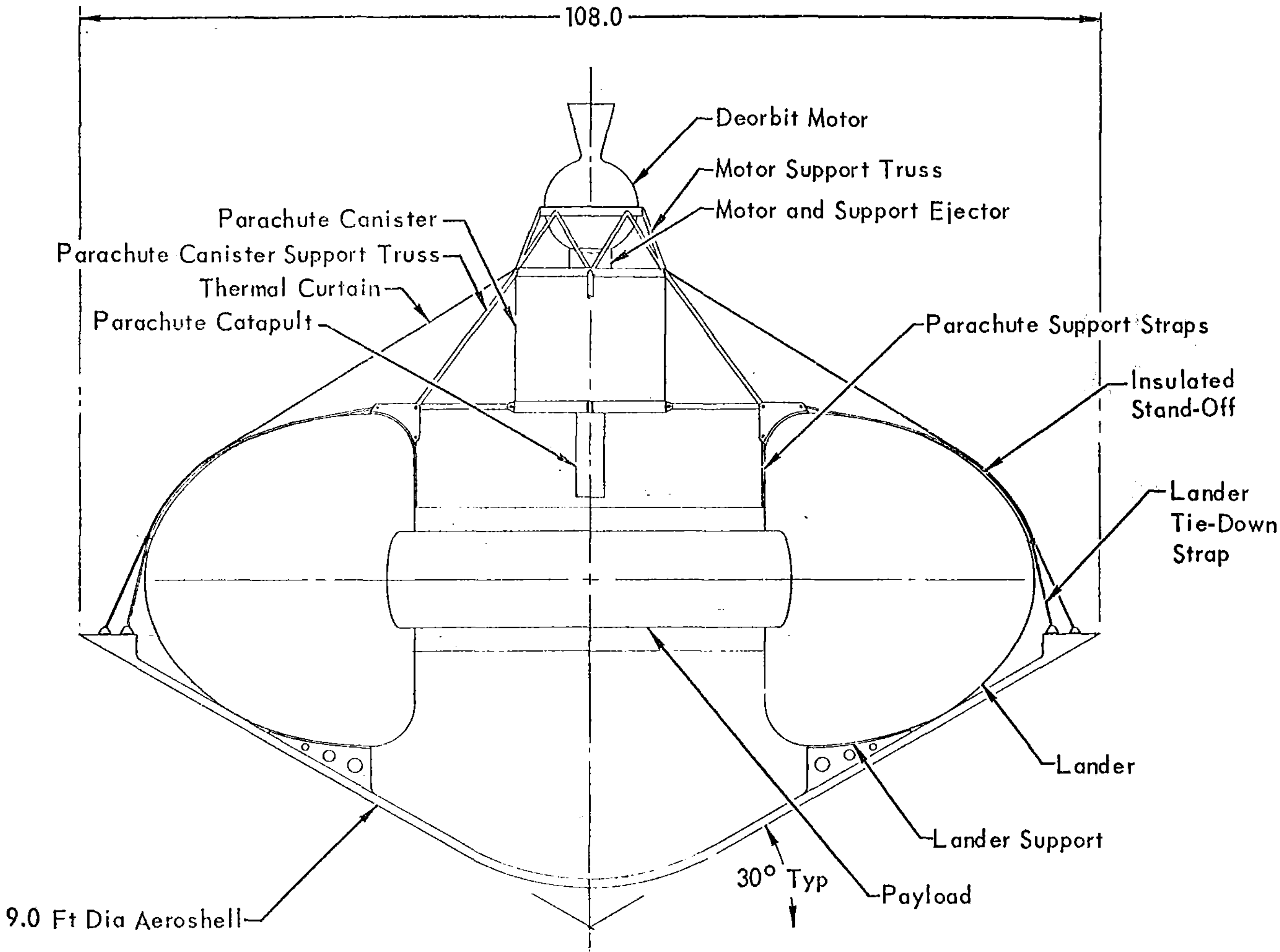


Figure 5.5-1

orbit motor support truss, and a de-orbit motor.

The lander is positioned with the major axis of the torus perpendicular to the centerline of the aeroshell. The lander is cradled in the aeroshell by support brackets attached to the aeroshell structure.

The parachute canister support truss straddles the torus and provides structural carry-thru for the lander tie-down straps in addition to suspending the lander during the parachute phase of the landing. The truss also provides a load path for the thrust from the de-orbit motor.

A thermal curtain encloses the lander and parachute canister to provide protection from aerodynamic heating during entry. The curtain is released with the aeroshell separation.

5.6 EXPERIMENT DEPLOYMENT - Figure 5.6-1 shows the baseline crushable torus with the exposed payload area available for experiment deployment. Lander is shown both with no crushing and with maximum crush plane on one side.

Since the torus lander is bi-directional, experiment instrument deployment is accomplished by one of the following arrangements:

(1) Installing dual instruments and equipment that must be deployed.

(2) Provide a flip-over mechanism to right the lander in case it comes to rest upside down.

(3) Provide a gimbal for those instruments and experiments that must be orientated prior to deployment. The latter concept was used for the baseline payload studies.

PAYLOAD EXPOSURE FOR EXPERIMENT DEPLOYMENT
CRUSHABLE BASELINE DESIGN

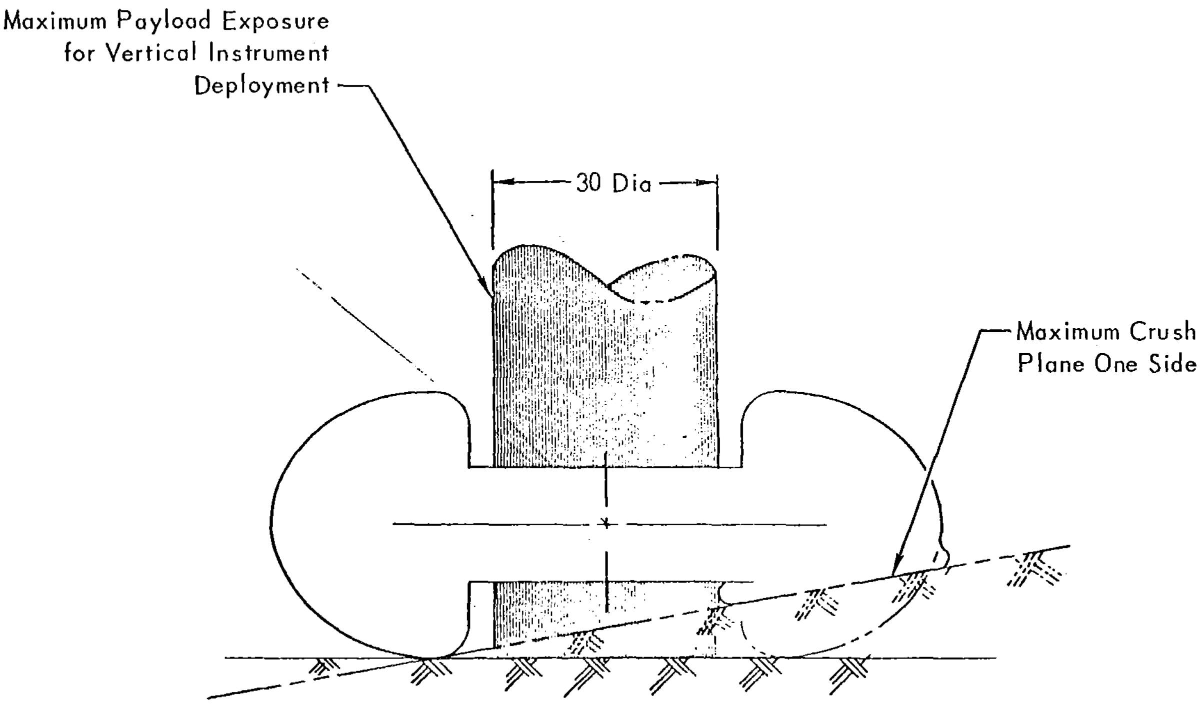


Figure 5.6-1

A typical payload equipment list is shown in Figure 5.6-2 for the 501.7 pound payload used in the baseline torus studies. Science instruments included are a facsimile camera; gas chromatograph/mass spectrometer; life detector, soil acquisition mechanism; meteorology including atmospheric humidity, temperature, pressure, and wind sensor; subsurface probe; and a bound water instrument.

5.7 LANDING LOADS AND MOTIONS - To evaluate the landing characteristics of the selected crushable torus design, three landing conditions were investigated with the Crushable Torus Landing Loads and Motions Program (Appendix B). The three cases were exactly the same except for the heading of their initial horizontal velocity component. The initial horizontal velocity components were directed (1) along the Y_f axis (down slope), (2) 45^o from the Y_f axis (45^o down slope), and (3) 90^o from the Y_f axis (cross slope). All three resultant initial velocities were 85 ft/sec oriented on a 30^o cone about the Z_f axis.

Initially, the lander X axis was parallel and in the same direction as the Z_f axis. The lander Z axis was parallel and in the same direction as the Y_f axis. The surface slope was 34^o, the coefficient of friction was .3, and the elastic stroke rebound was 0.4 inches. The lander geometry is described in Section 5.4.

TYPICAL CRUSHABLE LANDER EQUIPMENT BASELINE DESIGN

ITEM	VOLUME ($IN.^3$)	WEIGHT (LB)
Science	1387	52.7
Deployment & Orientation	280	30.0
Communications	1263	72.0
Sequencer	310	13.0
Radar	95	5.0
Electrical Power	1500	77.0
Wiring and Miscellaneous	1415	75.0
Subtotal Equipment	6250	324.7
Structure	1220	120.7
Thermal Control	6247	56.3
Total Payload	13 717	501.7

Figure 5.6-2

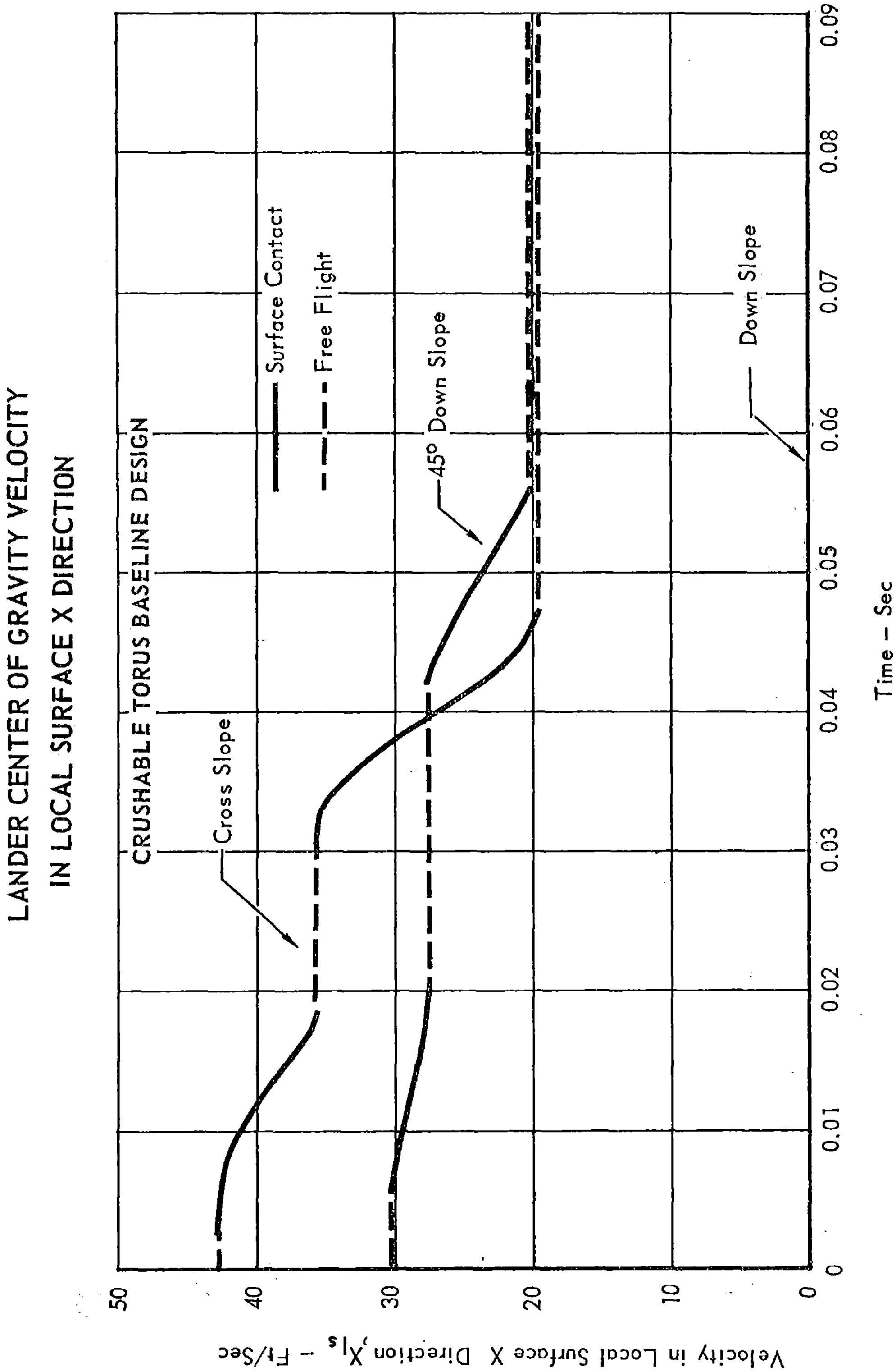
The translational and angular velocity time histories are shown in Figures 5.7-1 thru 5.7-6. From Figures 5.7-4, 5, and 6, it is seen that the cross slope condition resulted in the highest angular velocities. The cross slope case experienced the two impacts and rebounded in a much shorter time than the other two cases. The down slope and 45 degree down slope conditions underwent the two impacts in about the same period of time. The latter case left the ground a little earlier for the second rebound than did the down slope configuration.

The trajectory of the lander center of gravity is shown in Figures 5.7-7 and 5.7-8. This information is presented as the projections of the center of gravity motion on the plane of the landing surface and the Y_{1s} - Z_{1s} plane. It can be seen from Figure 5.7-8 that the two impacts shown for each of these three cases occur before the center of gravity rebounds. The two impacts are caused by one edge of the attenuator impacting first, causing lander rotation, then the other edge impacting, reducing or reversing the rotation (see Figure 5.7-4 thru 5.7-6).

Time histories of the forces and moments acting at the lander center of gravity are shown in Figure 5.7-9 and 5.7-10 for the down slope case. The loads are referenced to the lander coordinate system.

As mentioned in Section 5.1.2, a basic assumption to the program is that crush planes do not contact previously crushed planes. These three cases do violate this assumption to some extent. As an example of the problem, the motions of the crush plane are illustrated for the down slope case in Figure 5.7-11. Crush plane intersection, which is indicated by the shaded area in this figure, occurs during the rebound portion of the initial impact. A small force should be applied to the lander during the short period when the crush planes intersect, however the program has conservatively used zero force. The impulse imparted to the lander by the small force acting over a short time period would not appear to materially affect the motions of the lander.

In the cases studied where intersecting crush planes occur, the intersection occurs after the maximum attenuator stroke is reached. The relationship between lander rotational rate and the rate of change of stroke determines if the crush planes intersect.



LANDER CENTER OF GRAVITY VELOCITY
IN LOCAL SURFACE X DIRECTION
CRUSHABLE TORUS BASELINE DESIGN
Surface Contact
Free Flight
Cross Slope
45° Down Slope
Down Slope
Time – Sec
0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
Velocity in Local Surface X Direction, $\dot{X}_{ls}$ – Ft/Sec
0 10 20 30 40 50

Figure 5.7-1

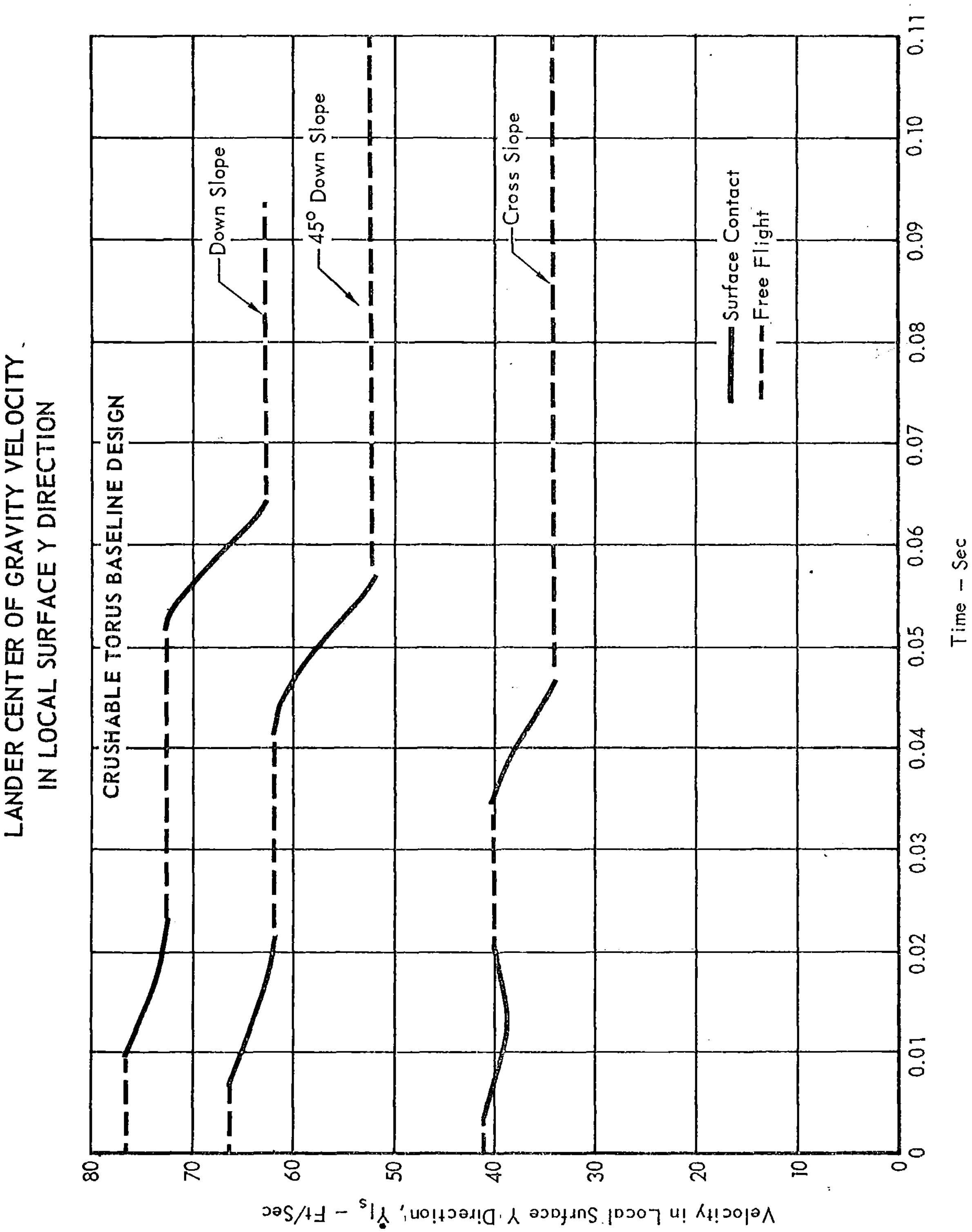

LANDER CENTER OF GRAVITY VELOCITY
IN LOCAL SURFACE Y DIRECTION
CRUSHABLE TORUS BASELINE DESIGN
Down Slope
45° Down Slope
Cross Slope
Surface Contact
Free Flight
Velocity in Local Surface Y Direction, $\dot{Y}_{ls}$ – Ft/Sec
Time – Sec
0
10
20
30
40
50
60
70
80
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11

Figure 5.7-2

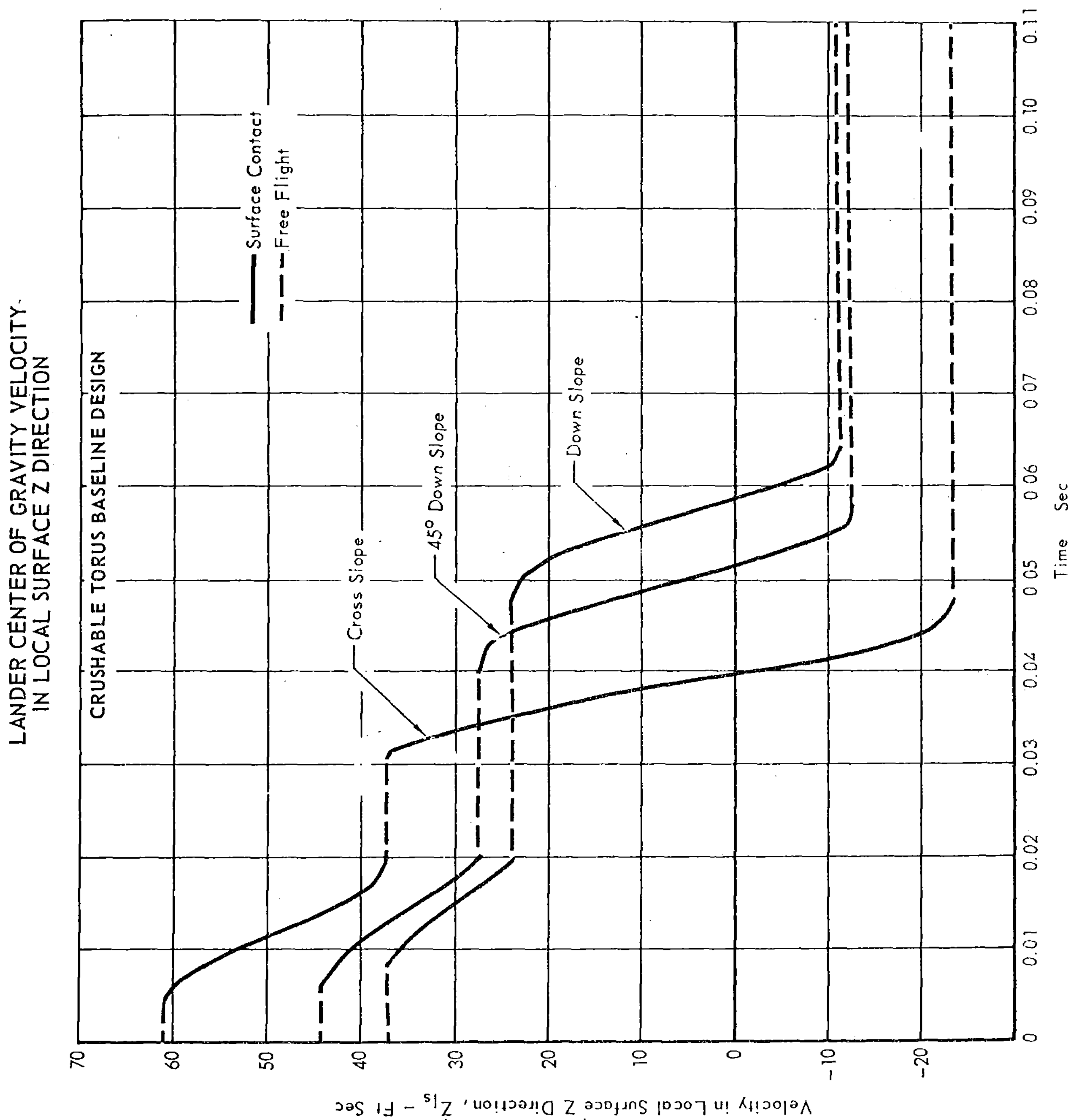

LANDER CENTER OF GRAVITY VELOCITY-
IN LOCAL SURFACE Z DIRECTION
CRUSHABLE TORUS BASELINE DESIGN
Surface Contact
Free Flight
Cross Slope
45° Down Slope
Down Slope
70
60
50
40
30
20
10
0
-10
-20
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
Time Sec
Velocity in Local Surface Z Direction, $\dot{Z}_{ls}$ – Ft Sec

Figure 5.7-3

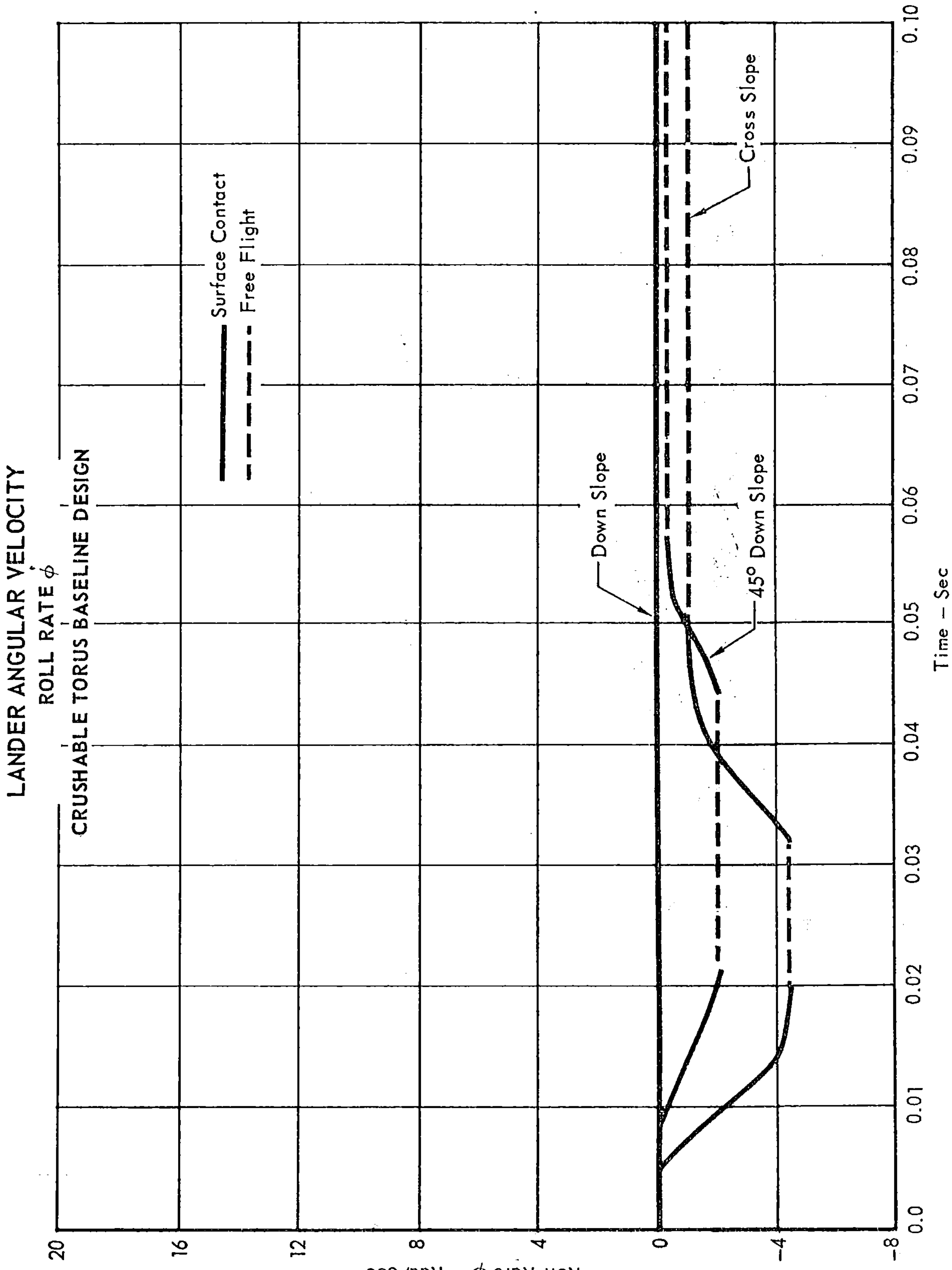
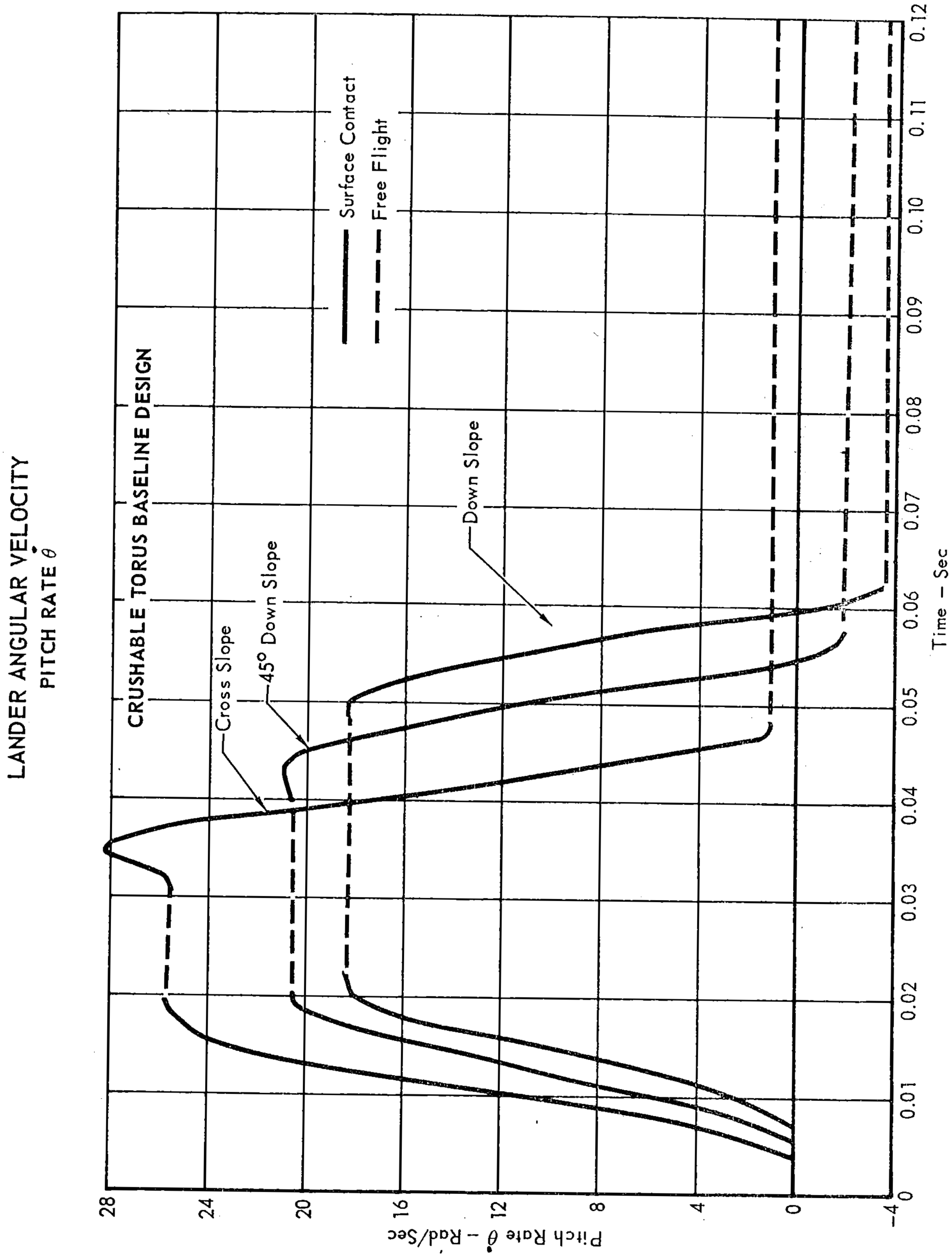

LANDER ANGULAR VELOCITY
ROLL RATE $\dot{\phi}$
CRUSHABLE TORUS BASELINE DESIGN
Surface Contact
Free Flight
Down Slope
45° Down Slope
Cross Slope
Roll Rate $\dot{\phi}$ – Rad/Sec
Time – Sec
20
16
12
8
4
0
–4
–8
0.0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10

Figure 5.7-4



LANDER ANGULAR VELOCITY
PITCH RATE $\dot{\theta}$
CRUSHABLE TORUS BASELINE DESIGN
Surface Contact
Free Flight
Cross Slope
45° Down Slope
Down Slope
Pitch Rate $\dot{\theta}$ – Rad/Sec
Time – Sec
28
24
20
16
12
8
4
0
-4
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
0.12

Figure 5.7-5

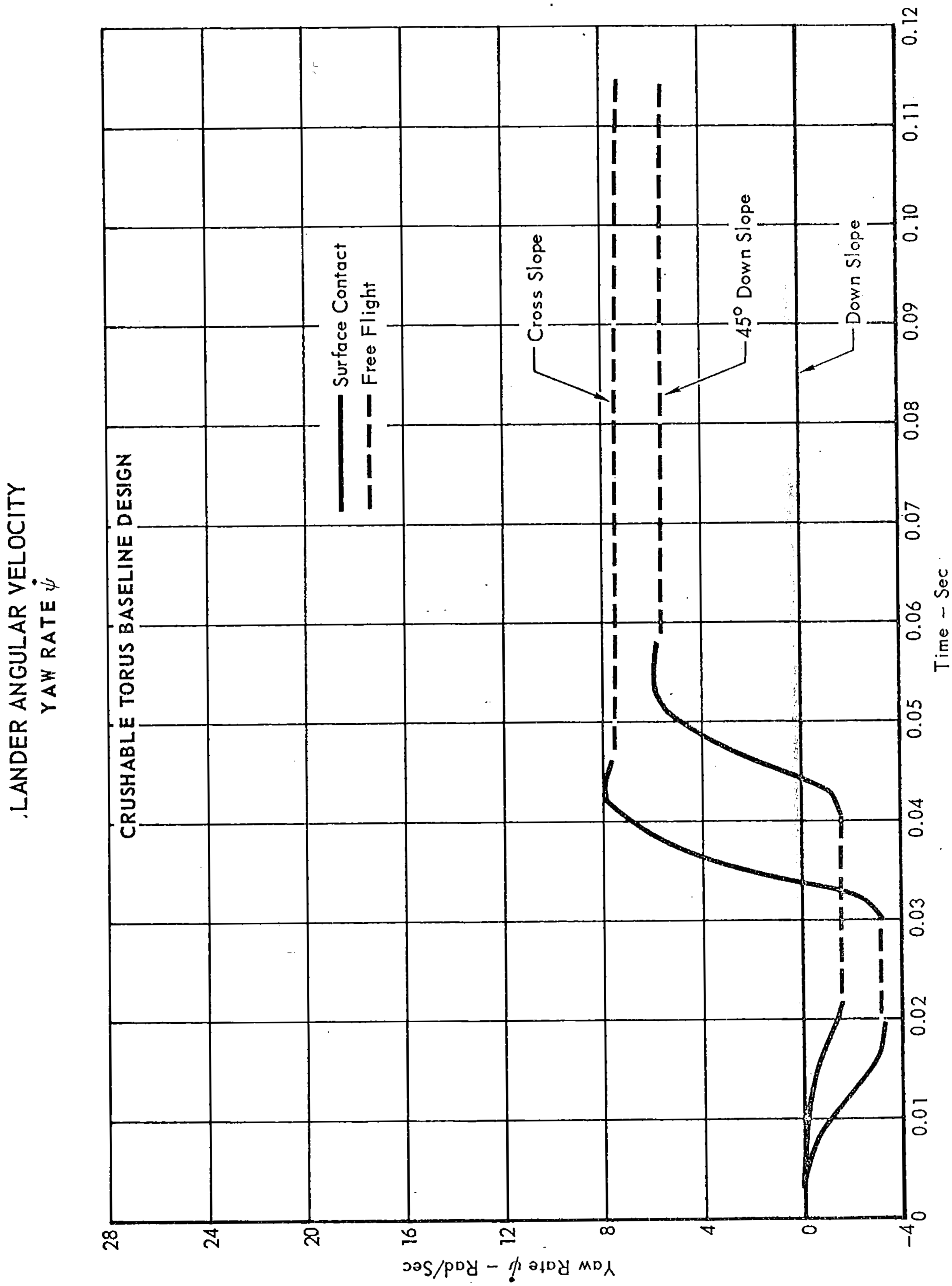

LANDER ANGULAR VELOCITY
YAW RATE ψ̇
CRUSHABLE TORUS BASELINE DESIGN
Surface Contact
Free Flight
Cross Slope
45° Down Slope
Down Slope
Yaw Rate ψ̇ – Rad/Sec
28
24
20
16
12
8
4
0
-4
Time – Sec
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
0.12

Figure 5.7-6

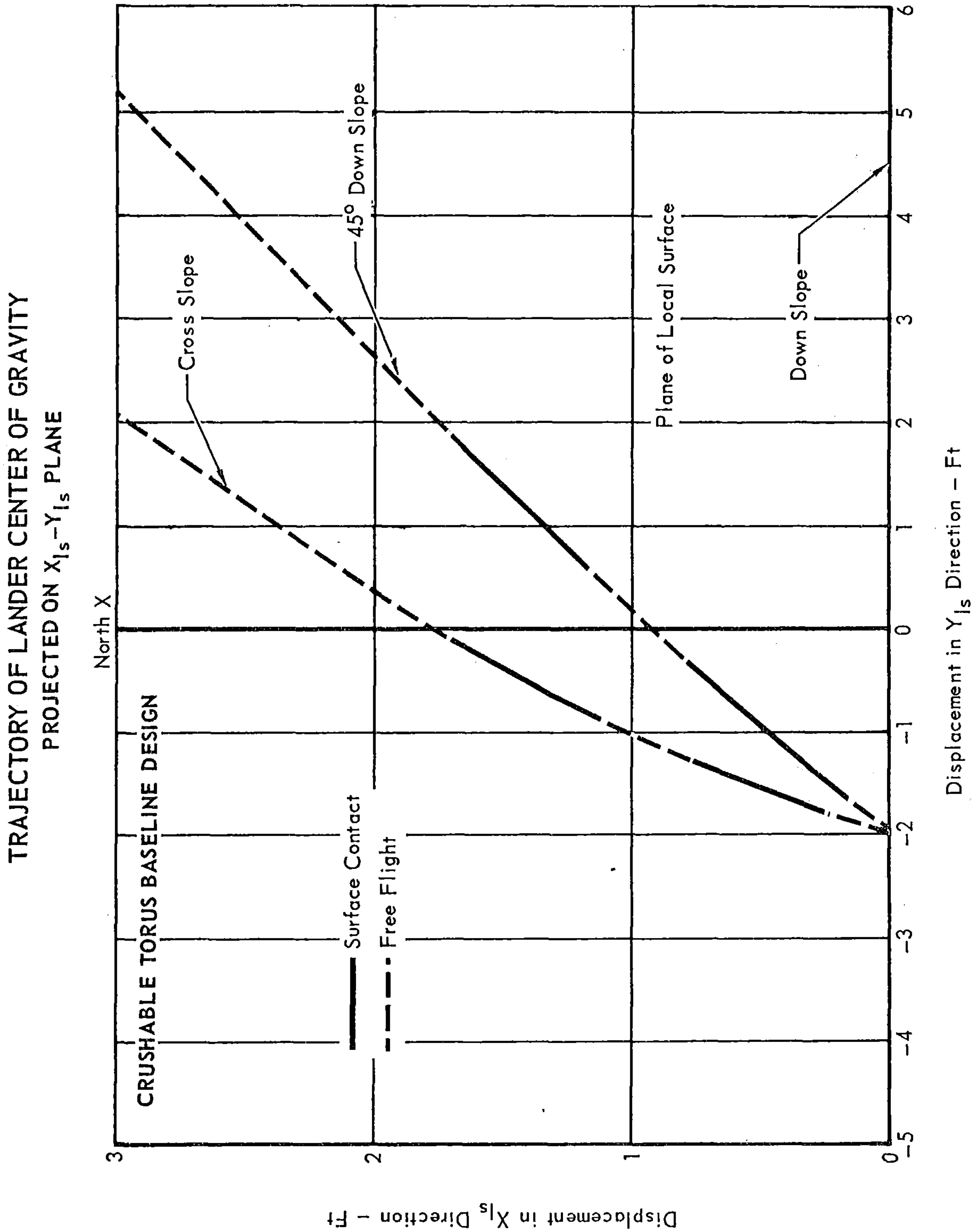

TRAJECTORY OF LANDER CENTER OF GRAVITY
PROJECTED ON $X_{Is}-Y_{Is}$ PLANE
CRUSHABLE TORUS BASELINE DESIGN
North X
Cross Slope
45° Down Slope
Plane of Local Surface
Down Slope
Surface Contact
Free Flight
Displacement in Y_{Is} Direction – Ft
Displacement in X_{Is} Direction – Ft
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
0
1
2
3

Figure 5.7-7

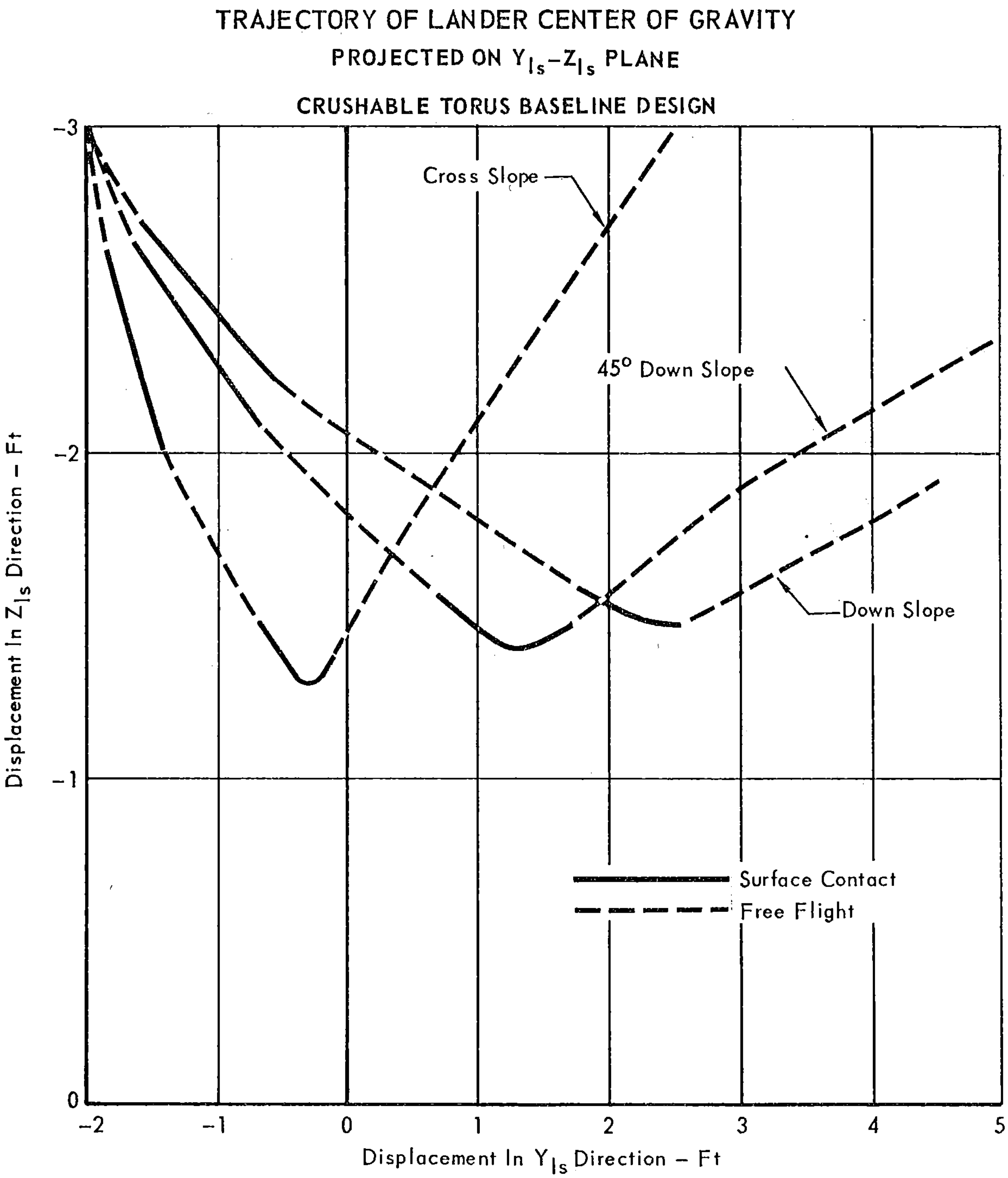

TRAJECTORY OF LANDER CENTER OF GRAVITY
PROJECTED ON $Y_{ls}-Z_{ls}$ PLANE
CRUSHABLE TORUS BASELINE DESIGN
Cross Slope
45° Down Slope
Down Slope
Surface Contact
Free Flight
Displacement In Z_{ls} Direction – Ft
Displacement In Y_{ls} Direction – Ft
-3
-2
-1
0
-2
-1
0
1
2
3
4
5

Figure 5.7-8

FORCES AT
LANDER CENTER OF GRAVITY
DOWN SLOPE CASE

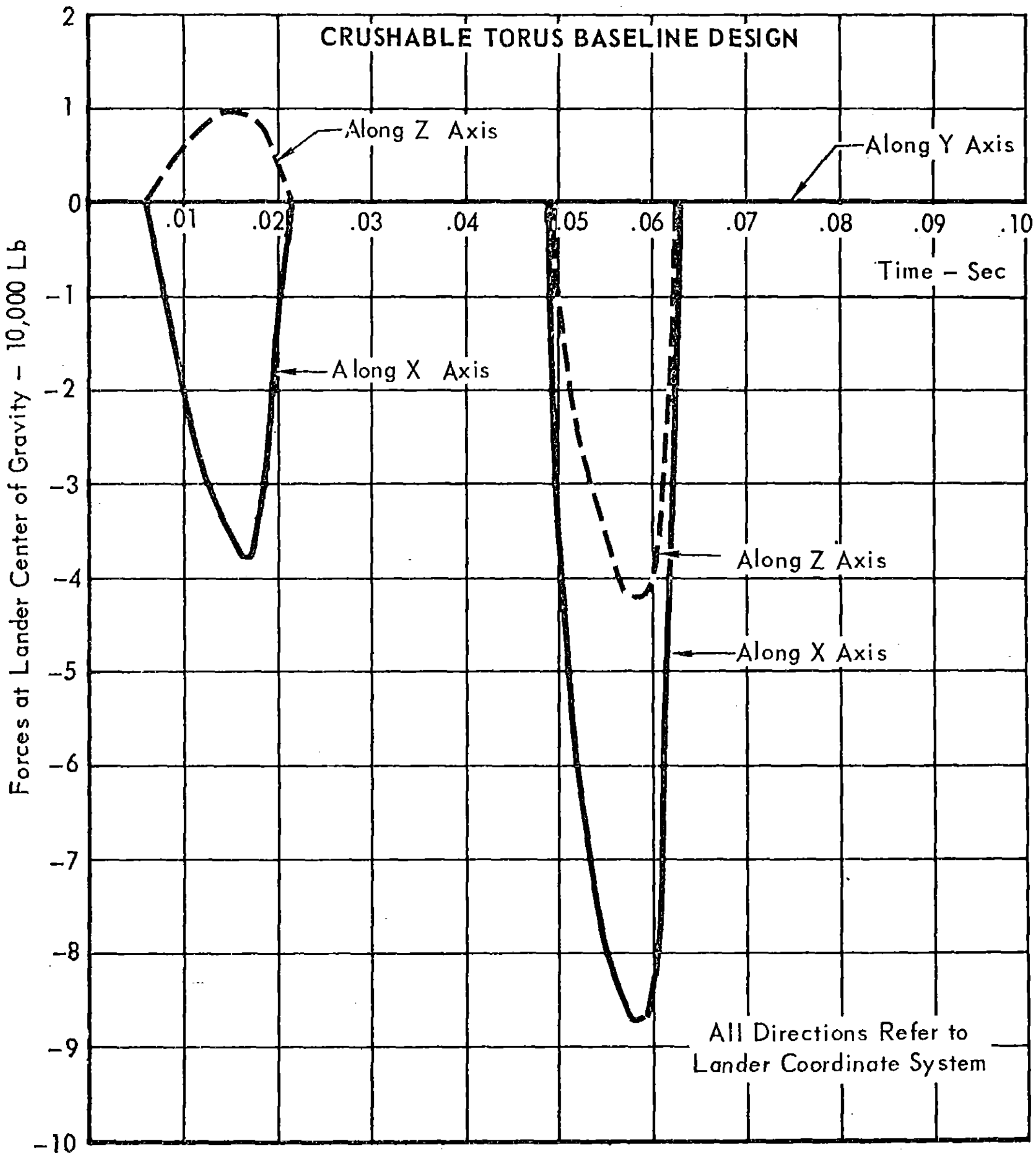


Figure 5.7-9

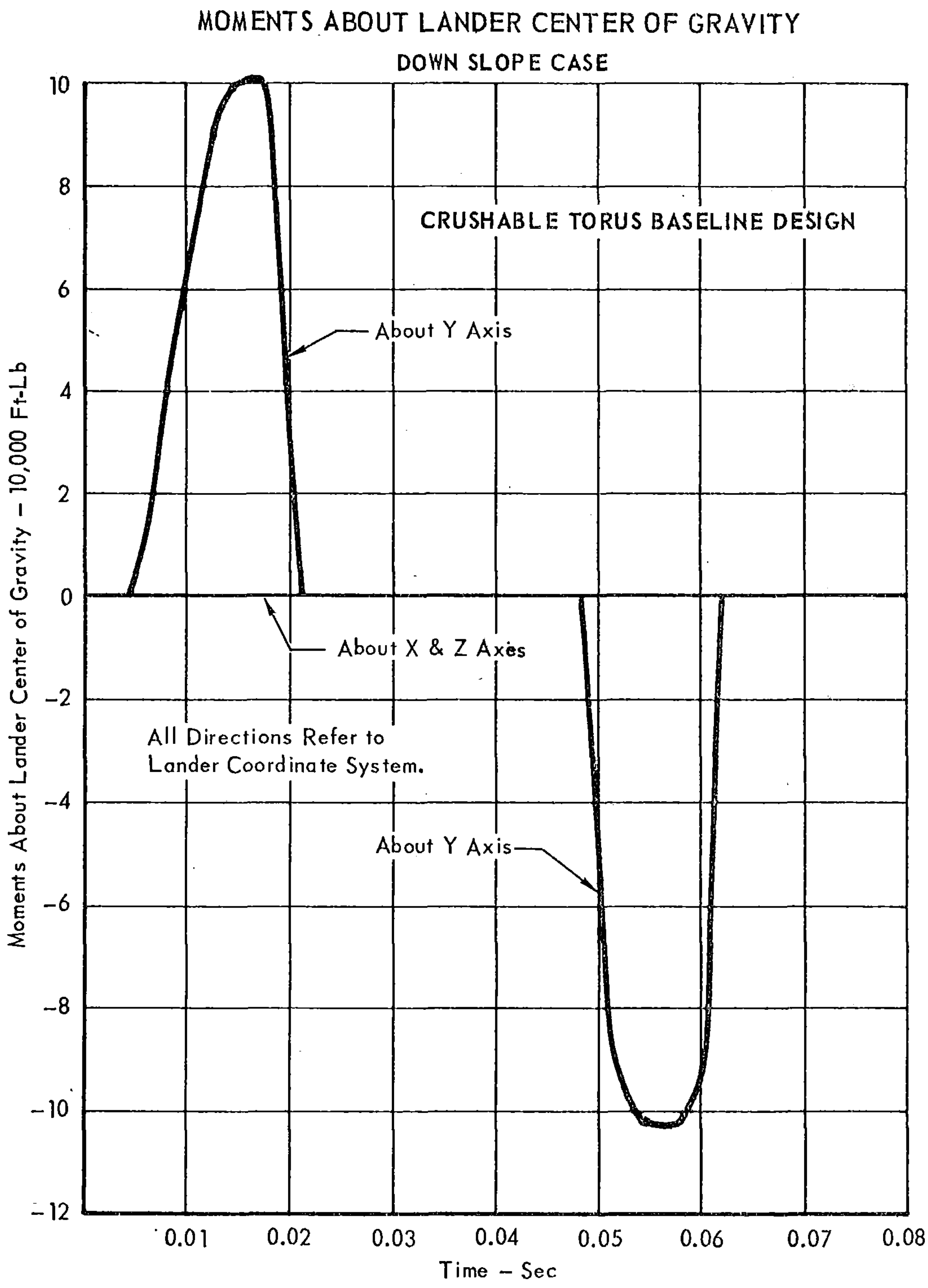

MOMENTS ABOUT LANDER CENTER OF GRAVITY
DOWN SLOPE CASE
CRUSHABLE TORUS BASELINE DESIGN
About Y Axis
About X & Z Axes
All Directions Refer to Lander Coordinate System.
About Y Axis
Moments About Lander Center of Gravity – 10,000 Ft-Lb
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Time – Sec

Figure 5.7-10

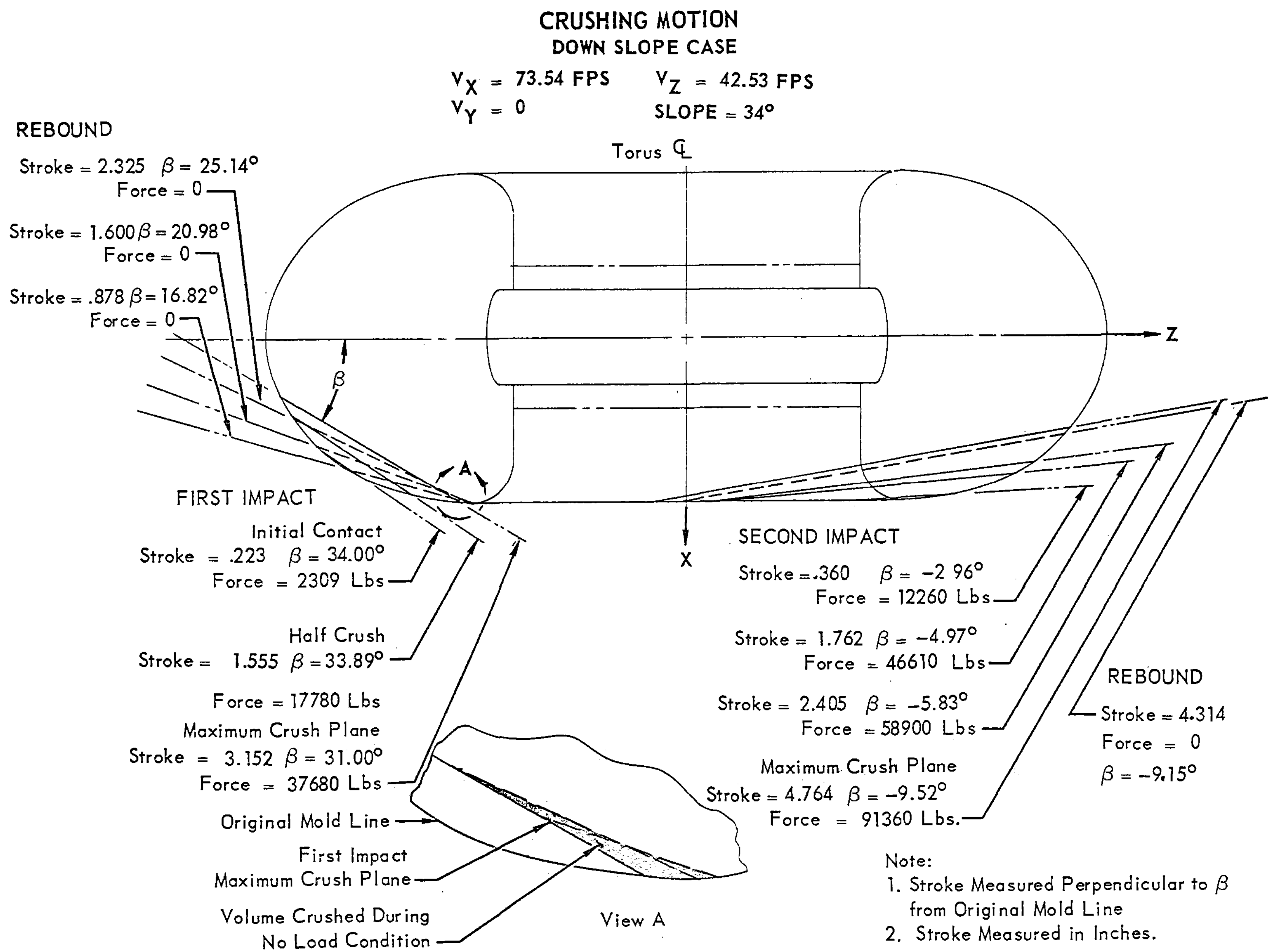

CRUSHING MOTION
DOWN SLOPE CASE
V_X = 73.54 FPS
V_Y = 0
V_Z = 42.53 FPS
SLOPE = 34°
Torus ℄
REBOUND
Stroke = 2.325 β = 25.14°
Force = 0
Stroke = 1.600 β = 20.98°
Force = 0
Stroke = .878 β = 16.82°
Force = 0
β
A
FIRST IMPACT
Initial Contact
Stroke = .223 β = 34.00°
Force = 2309 Lbs
Half Crush
Stroke = 1.555 β = 33.89°
Force = 17780 Lbs
Maximum Crush Plane
Stroke = 3.152 β = 31.00°
Force = 37680 Lbs
Original Mold Line
First Impact
Maximum Crush Plane
Volume Crushed During
No Load Condition
View A
Z
X
SECOND IMPACT
Stroke = .360 β = −2 96°
Force = 12260 Lbs
Stroke = 1.762 β = −4.97°
Force = 46610 Lbs
Stroke = 2.405 β = −5.83°
Force = 58900 Lbs
Maximum Crush Plane
Stroke = 4.764 β = −9.52°
Force = 91360 Lbs.
REBOUND
Stroke = 4.314
Force = 0
β = −9.15°
Note:
1. Stroke Measured Perpendicular to β from Original Mold Line
2. Stroke Measured in Inches.

Figure 5.7-11

6. INFLATABLE TORUS LANDING SYSTEM

This section describes the computer programs developed to predict landing loads and motions and the efficient structural design of inflatable torus configurations. Studies performed using these programs leading to a selected inflatable torus configuration are also presented in this section.

6.1 COMPUTER PROGRAMS

6.1.1 Inflatable Torus Structural Design Program. - This program, which is discussed in detail in Appendix C, provides the capability for establishing inflatable torus configurations meeting specific design constraints. Operation of this program requires initial selection of a payload, bag material, desired velocity capability, and desired load factor. The program then determines torus dimensions, internal pressures, and bag thicknesses providing the required velocity capability and load factor for flat landing. Velocity capability, load factor, and bag stresses for end landing are also determined, and if a bag thickness increase is necessary, the program automatically reruns both flat and end landing calculations with an updated bag thickness and weight. Normally the outer 120° segment of the bag is critical for end landing.

The program will also determine load factor and velocity capability for any desired input unidirectional landing attitude (constant attitude stroking). Flat landing has been found to be critical from both a velocity capability and load factor standpoint. The velocity capability increases and the load factor decreases as landing attitude (BET) is increased from 0° (Flat Landing) to 90° (End Landing).

Dimensional variables used to represent the payload and attenuator are shown in Figures 6.1-1 and 6.1-2. The payload is a cylinder attached to the torus attenuator by a gimbal ring. Thickness of the torus attenuator is determined by the program in integer number of plys at three locations around the circumference. Properties of a single ply of the bag material as well as desired minimum number of plys at the three locations are input quantities. Any elastomer used to seal the bag, which is a function of the number of plys, should be included in the weight of one ply. Weight of any scuff protection for the bag, which is a function only of surface area, is a separate input.

Deflected torus shape has been used as the analytical basis for calculation of footprint area, volume change, and torus stresses. Pressure rise is determined based on volume change, assuming a polytropic process. A two spring/mass dynamic model is utilized for a flat landing, and a single spring/mass dynamic model is utilized for landings other than flat. The program allows the user to select the desired minimum number of plys at three locations, and internally calculates the torus thickness required in integer number of plys if minimum ply strength is exceeded.

An inflatable torus drop test program to determine the effects of landing velocity, payload weight, and torus pressure on payload stroke and acceleration under simulated Mars atmospheric pressure was conducted by McDonnell Douglas Astronautics Company, Eastern Division, under NASA Contract NAS-1-7977

INFLATABLE TORUS GEOMETRY

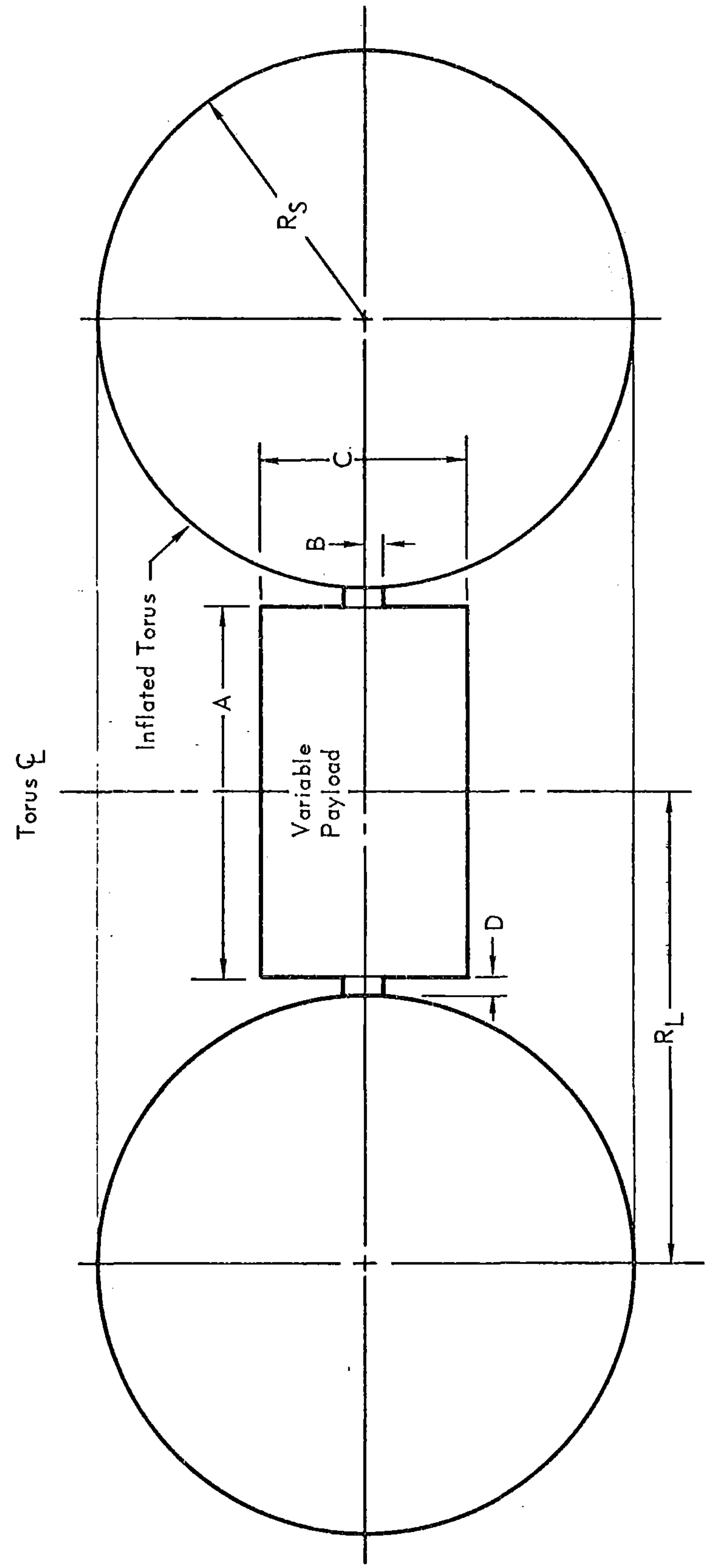


Figure 6.1-1

VARIATION OF TORUS MATERIAL THICKNESS

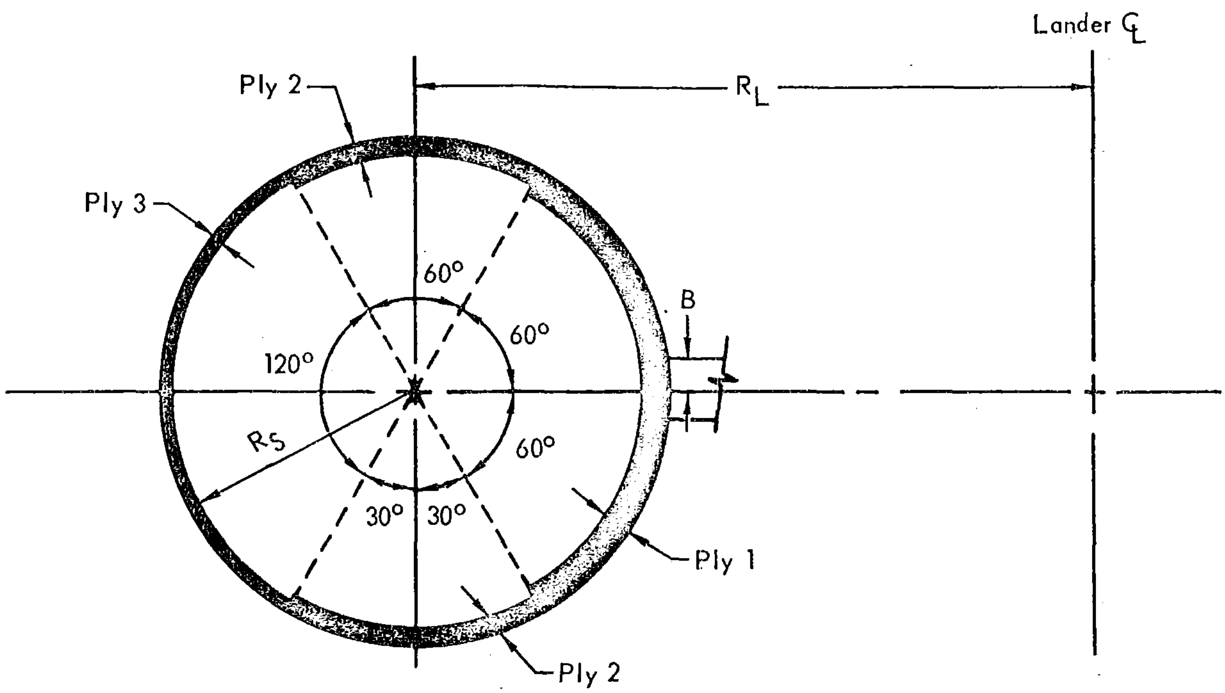


Figure 6.1-2

for Langley Research Center. Test results for the drop test program, presented in Reference 2, substantiate the analytical models chosen for the Inflatable Torus Structural Design Computer Program.

Output data include: bag radius (R_S); torus radius (R_L); inflation pressure; maximum pressure; payload weight; landing system weight (torus fabric, elastomer, scuff protection, and gas); inflation system weight (gas and tank); required number of plys at three locations; maximum running load at three points on the bag; and maximum load factor and velocity capability for flat landing, end landing, and landing at input contact angle. Data output as a function of stroke include: total force normal to the surface; internal bag pressure; footprint area; and velocity capability.

6.1.2 Inflatable Torus Landing Loads and Motions Program. - The Inflatable Torus Landing Loads and Motions Program determines the positions, velocities, and accelerations of a given inflatable torus lander as a function of time. The lander's configuration may be established with the Inflatable Torus Structural Design Program (Section 6.1.1 and Appendix C), or a lander design may be available from some other source. Operating instructions, together with a discussion of the methods of analysis and two example runs, are presented in Appendix D.

The torus lander is assumed to have the geometric properties shown in Figure 6.1-1. The lander is comprised of two main parts; the inflated landing system and the payload. The payload consists of a cylindrical shaped package mounted in the center of the inflated torus impact bag. A gimbal ring supports the payload structure which allows the required payload alignment following landing. During landing, this gimbal ring is locked so that it provides rigid support for the payload. The payload package and gimbal ring are assumed to have a uniform weight density.

The inflatable torus is constructed of a suitable fabric coated with an elastomer to provide gas containment and scuff resistance. To provide the required torus strength properties with a minimum weight, the thickness of the torus material may be varied over three sections. An input quantity is available to account for the hysteresis effects of the torus material. The torus is considered to be an unvented, uncompartmented structure.

The analysis developed in this study was correlated with the inflatable torus drop test program results presented in Reference 2. The Inflatable Torus Landing Loads and Motions Program provides adequate predictions of the results obtained during this test program.

Features incorporated in this program include the ability to: simulate spatial motions; suppress degrees of freedom to conserve computational time; vary the torus hysteresis factor, lander geometry, surface slope, coefficient of friction, and rock diameter; select variable or constant step Predictor-Corrector or Runge-Kutta integration methods; select values for as many as eight independent time history variables for problem termination; and to revise the form of the program output with indicators set through the input data. The computed lander time history variables may be referenced to any of three coordinate systems. Two of these are fixed in the planet's surface and the other moves with the lander's center of gravity. In addition, the program has a multiple case capability.

Major analytical considerations are grouped in subroutines which allow modifications and improvements to be made with relative ease. Also, additional skeleton subroutines are provided with which the effects of aerodynamic and environmental conditions can be included in the analysis.

6.2 PAYLOAD SHAPE COMPARISON - Payload and attenuator dimensions for the inflatable torus are shown in Figure 6.1-1. The influence of the payload shape parameter A/C, which is the cylindrical payload diameter divided by payload height, on landing system weight, bag radius, and absolute inflation pressure is shown in Figures 6.2-1, 6.2-2, and 6.2-3, respectively. Ambient pressure for all studies was 0.0725 psi. These figures are for a 500 pound payload, a velocity of 85 feet per second, and a 5 inch diameter rock. Separate curves are shown for 50, 150, and 300 earth g load factor designs. The Inflatable Torus Structural Design Program allows the tailoring of bag radius, bag thickness, and inflation pressure to provide desired load factor for flat landings. Inflatable systems allow considerably more flexibility than crushable systems in achieving desired load factor in the intermediate landing category. Basically, this is because crushable materials are only available with specific properties, and do not allow continuous variation as does bag inflation pressure for the inflatable landing systems.

Selection of an A/C ratio of between 3.5 and 5.0 depending on desired load factor results in a minimum weight landing system (see Figure 6.2-1).

INFLATABLE TORUS
INFLUENCE OF PAYLOAD SHAPE PARAMETER ON LANDING SYSTEM WEIGHT

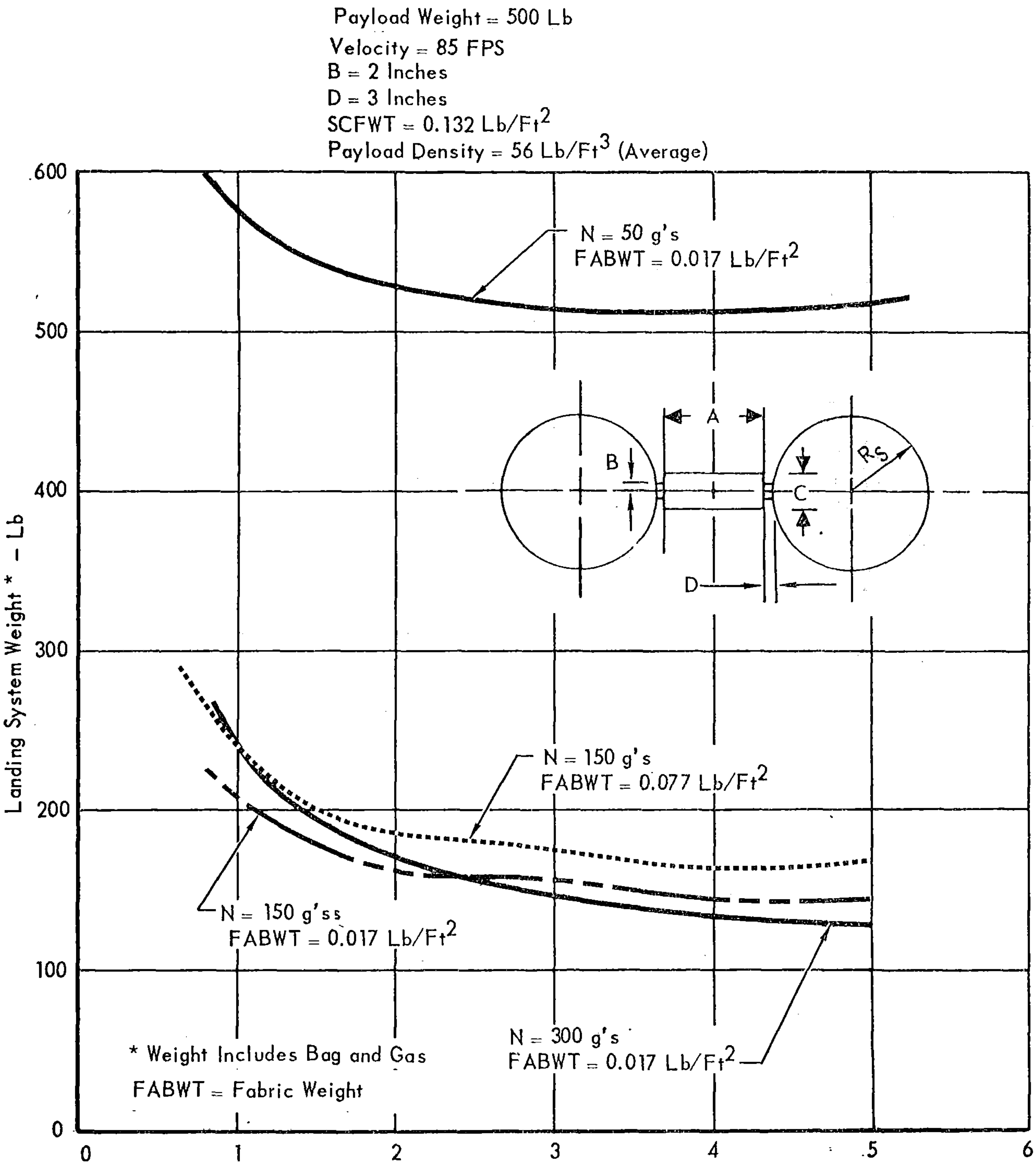


Figure 6.2-1

INFLATABLE TORUS
INFLUENCE OF PAYLOAD SHAPE PARAMETER ON BAG RADIUS

Payload Weight = 500 Lb
Velocity = 85 FPS
B = 2 Inches
D = 3 Inches
SCFWT = 0.132 Lb/Ft^2
Payload Density = 56 Lb/Ft^3 (Average)

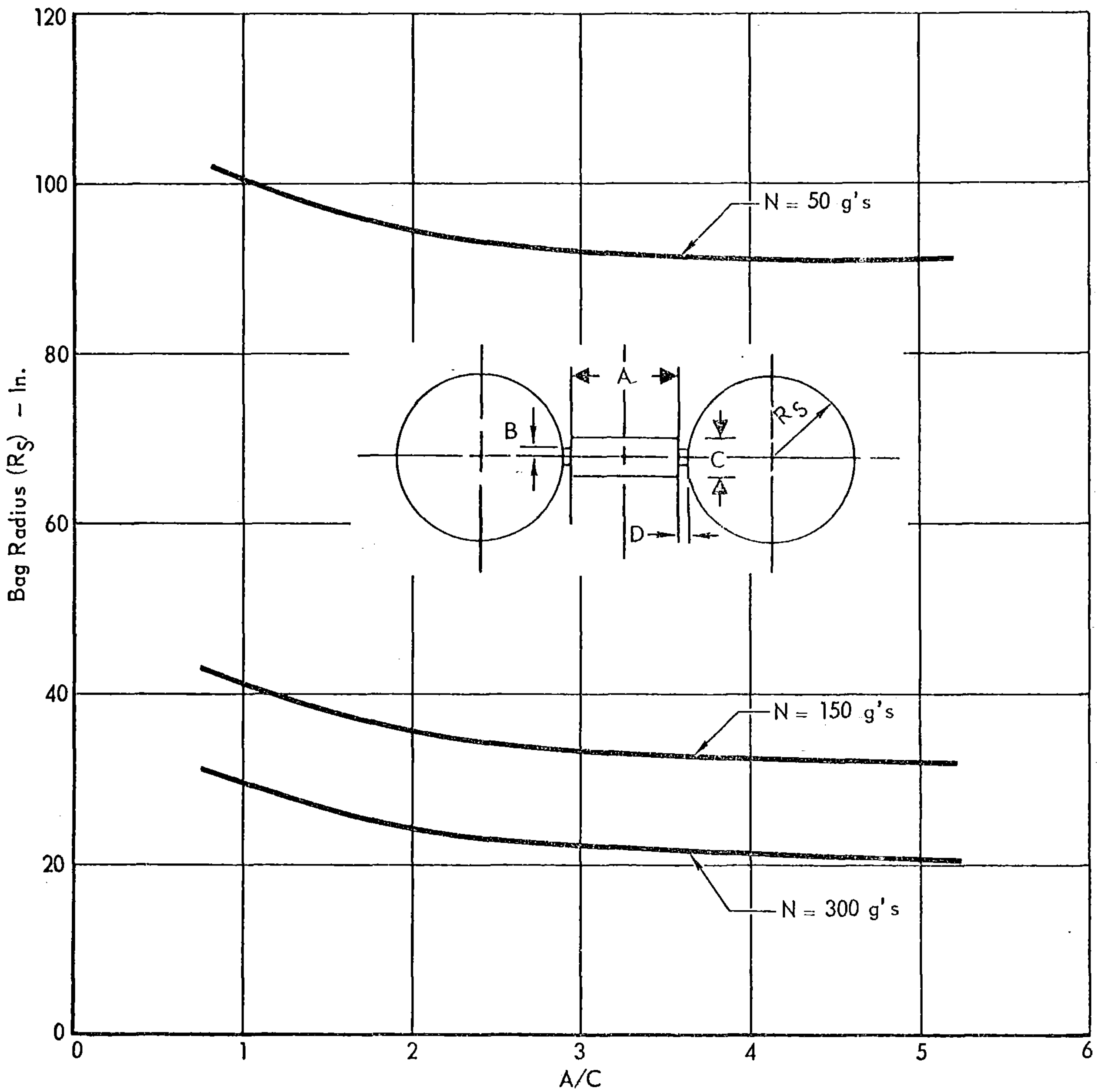


Figure 6.2-2

INFLUENCE OF PAYLOAD SHAPE
PARAMETER ON REQUIRED INFLATION PRESSURE

Payload Weight = 500 Lb
Velocity = 85 FPS
B = 2 Inches
D = 3 Inches
SCFWT = 0.132 Lb/Ft^2
Payload Density = 56 Lb/Ft^3 (Average)

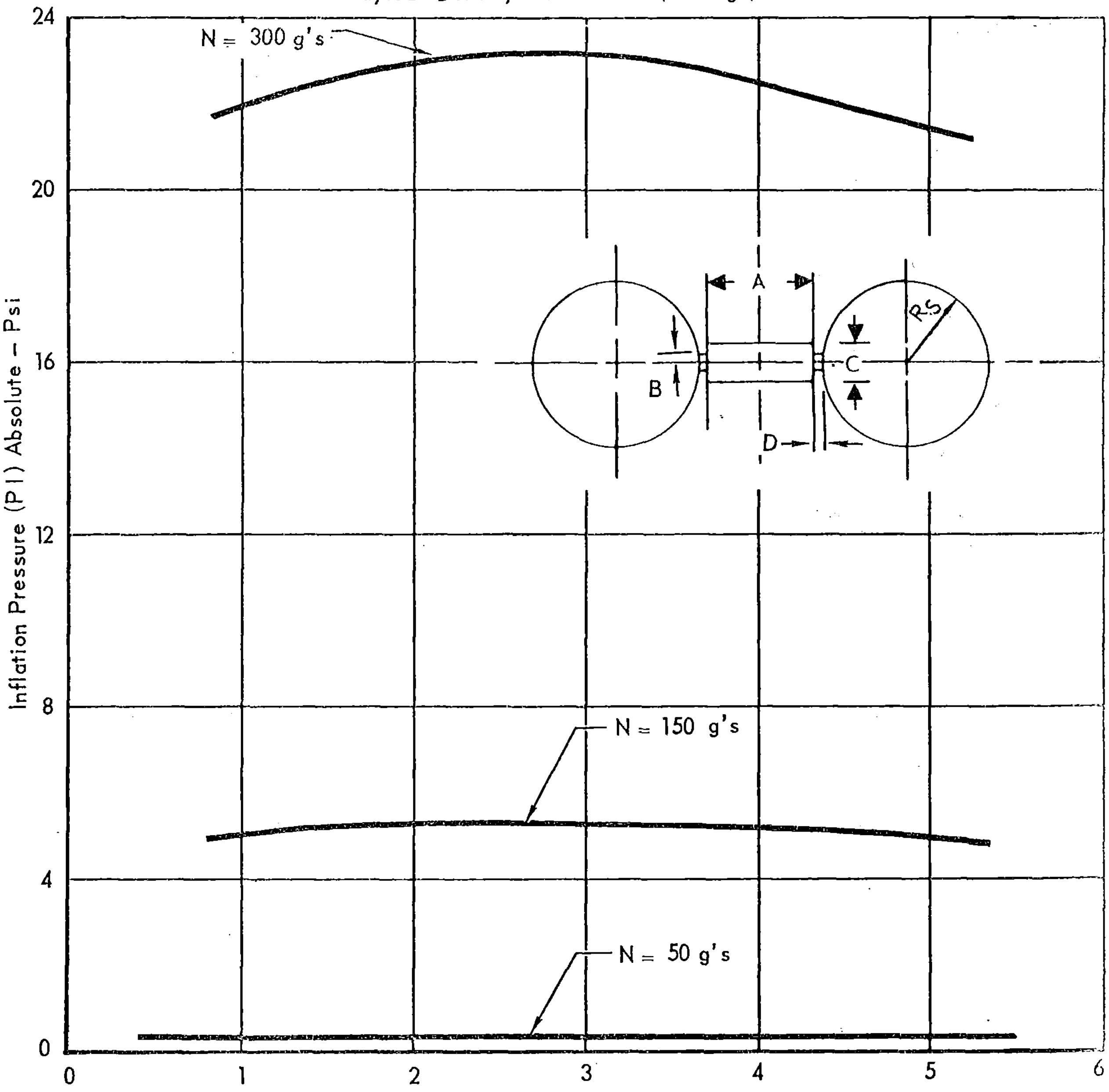


Figure 6.2-3

Values of A/C larger than 5.0 result in heavier landing systems because of minimum bag thickness (ply) restrictions. Inflation pressure is not greatly affected by payload shape parameter A/C, particularly for 50 and 150 g load factor curves (see Figure 6.2-3). Required bag radius is slightly reduced as payload shape parameter A/C is increased (see Figure 6.2-2). Landing system weight and required bag radius are shown to decrease, and inflation pressure is shown to increase as design load factor increases in Figure 6.2-4 (for constant A/C = 4.0 and fabric weight = .017 lb/ft^2).

6.3 BAG MATERIAL COMPARISON - The effect of fabric weight (thickness) on landing system weight is also shown in Figure 6.2-1 for the 150 g load factor curves. Thin fabric (fabric weight = .017 lb/ft^2) allows a design closer to optimum than a relatively thick fabric (fabric weight = .077 lb/ft^2) and consequently allows a 12.7% weight savings for A/C = 4.0. The thin fabric is not as desirable from a manufacturing standpoint, however, since 14 plys are required at the payload attach for the thin fabric, whereas only 3 plys are required for the thick fabric (for A/C = 4.0). There is little difference in either bag radius or inflation pressure for the two fabrics.

6.4 SELECTED CONFIGURATION - The inflatable single torus was chosen because it offers efficient equipment packing, flexible experiment packaging, easy experiment deployment, and lowest landing system weight. A payload shape parameter A/C of 4.0 optimizes these concept attributes within the specified constraints of Section 3. Selection of a baseline design allowing flat landing at 150 earth g's of a 500 pound payload was made since this was near the

INFLATABLE TORUS
INFLUENCE OF LOAD FACTOR ON WEIGHT, BAG RADIUS, AND INFLATION PRESSURE

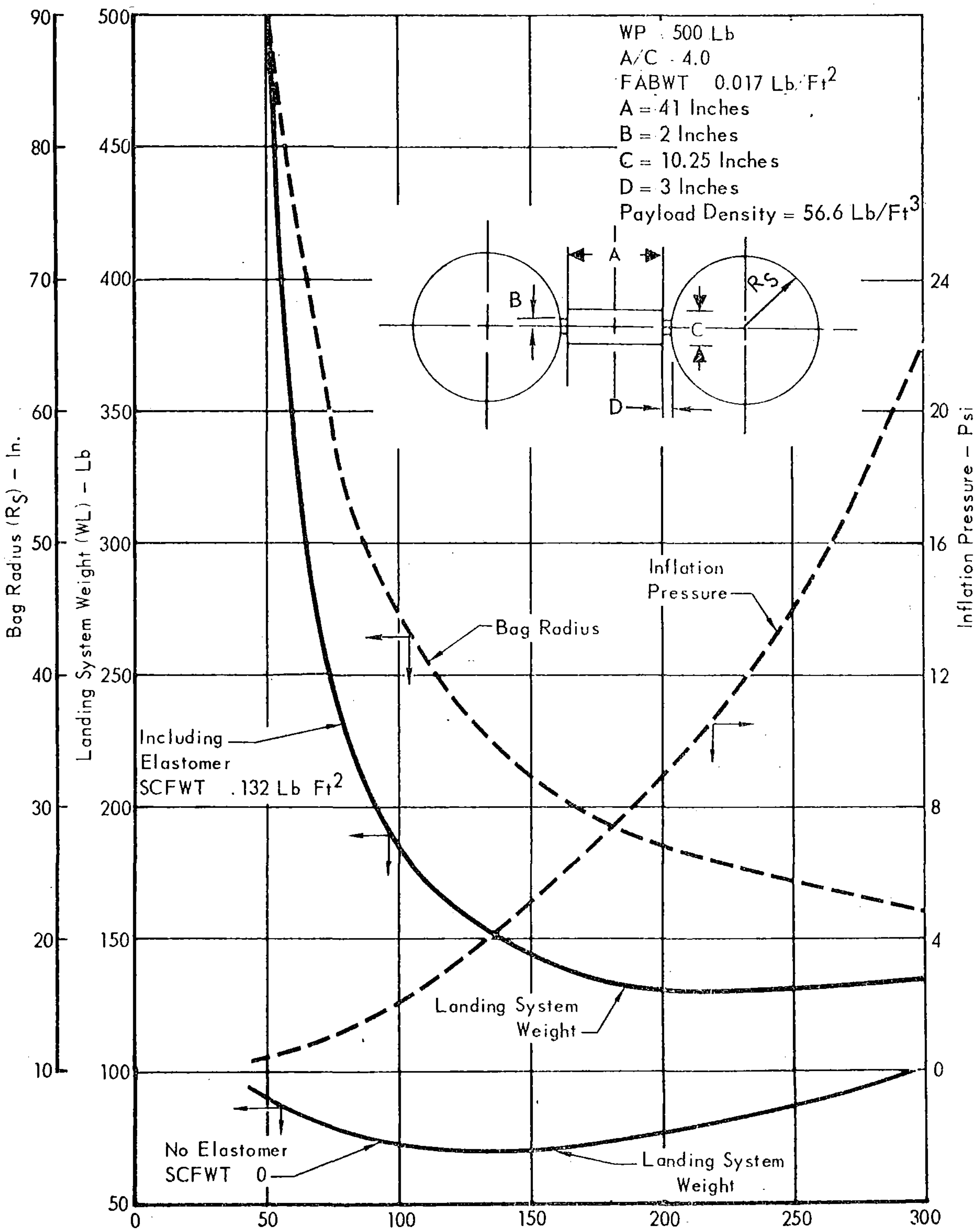


Figure 6.2-4

center of the load factor range. End landing load factor is 71 earth g's for the baseline.

The baseline inflatable torus is shown in Figure 6.4-1. The bag is constructed of 2 plys of HT-1 (Nomex) on the outer 120° portion of the bag and 3 plys of HT-1 on the inner 240°. One ply weighs .077 lb/ft^2 and has a strength of 534 lb/in. An 85% seam efficiency factor reduces the allowable running load to 454 lbs/in. The bag is composed of 20 gores per ply bonded together and attached as shown in Figure 6.4-1, to the gimbal ring. Use of this material (.077 lb/ft^2) rather than the thinner Nomex (.017 lb/ft^2) allows reduction in the number of plys from 14 to 3, while adding 21 pounds to the landing system weight. Abrasion and puncture protection and torus seal are obtained by uniformly coating the torus with an elastomer (silicone rubber compound of the methyl-phenyl type). A weight of .132 lb/ft^2 was included for this material.

Required inflation pressure is 5.31 psi absolute (.0725 psi ambient pressure) with 2.7 pounds of nitrogen gas being supplied from a 20.7 lb. tank, which is not landed (tank is removed by the parachute when it is released prior to lander impact). The ideal gas constant of 1.40 for nitrogen was used in the computer run (program assumes a polytropic process). Maximum limit pressure during impact is 6.25 psi.

6.5 AEROSHELL COMPATIBILITY - Aeroshell installation of the inflatable torus is quite similar to the crushable torus. Since, the inflatable torus is stowed in the deflated condition, the payload package will sit lower in

INFLATABLE TORUS BASELINE DESIGN

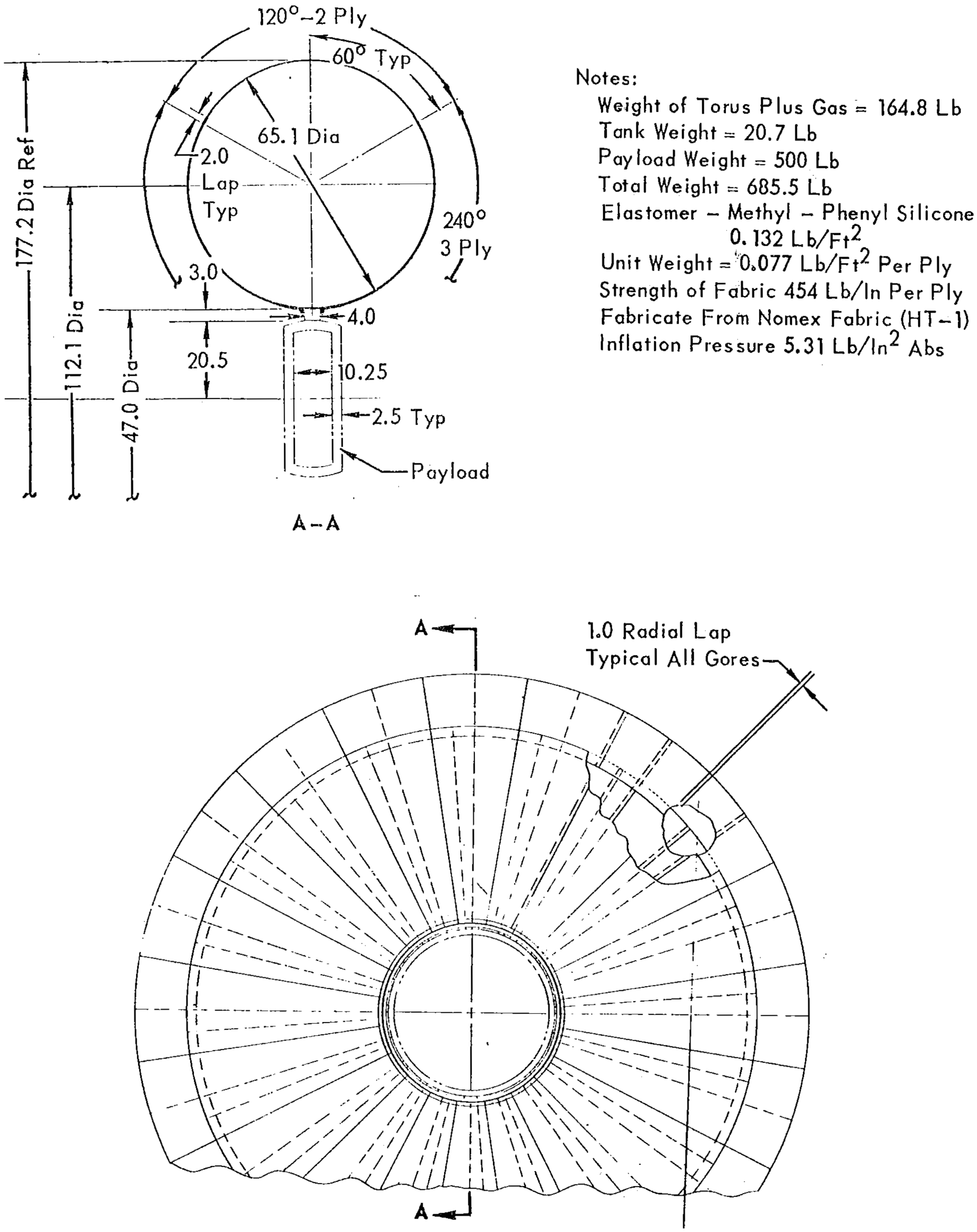


Figure 6.4-1

the aeroshell. This will yield a center of gravity of the entry vehicle a little closer to the apex of the aeroshell cone than was obtained with the crushable torus installation (see Figure 5.5-1).

6.6 EXPERIMENT DEPLOYMENT - The exposed payload area available for experiment deployment of the baseline inflatable lander is shown in Figure 6.6-1. The lander is shown on both a flat horizontal surface and a 34 degree slope. The shaded area shows the maximum envelope available for instrument deployment without interference with the bag.

Design layout of the payload package was prepared for the list of equipment shown in Figure 6.6-2. Equipment and structural arrangements of the payload package are shown in Figure 6.6-3.

The complete payload package is supported by a single gimbal ring which allows the payload to right itself after the lander comes to rest.

6.7 LANDING LOADS AND MOTIONS - To evaluate the landing characteristics of the selected inflatable torus design, a number of landing conditions were investigated with the Inflatable Torus Landing Loads and Motions Program (Appendix D). The lander had an initial attitude oriented so that the X, Y, and Z axes were parallel to and in the same direction as the planet's Z_f, X_f, and Y_f axes, respectively. A local surface slope of 34 degrees with a 0.3 coefficient of friction was investigated. Various initial lander velocities with a resultant of 85 ft/sec oriented on a 30 degree cone about the gravity vector were considered. This resulted in the lander having an initial

PAYLOAD EXPOSURE FOR EXPERIMENT DEPLOYMENT
INFLATABLE BASELINE DESIGN

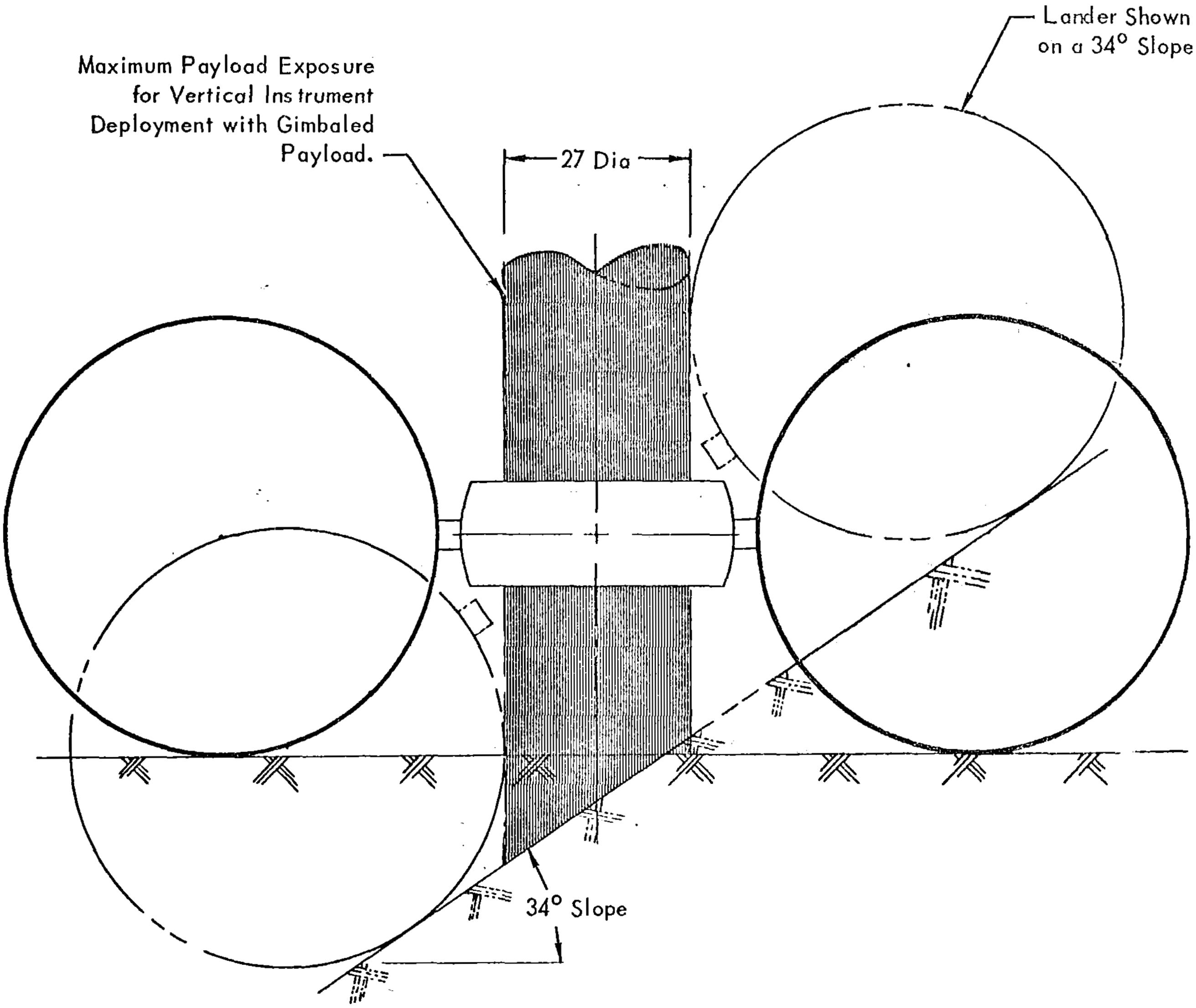


Figure 6.6-1

TYPICAL INFLATABLE LANDER EQUIPMENT BASELINE DESIGN

ITEM	VOLUME ($IN.^3$)	WEIGHT (LB)
Science	1387	52.7
Deployment and Orientation	373	40.0
Communication	1263	72.0
Sequencer	310	13.0
Radar	95	5.0
Electrical Power	1500	77.0
Wiring and Miscellaneous	1415	75.0
Subtotal Equipment	6343	343.7
Structure	1020	99.8
Thermal Control	6970	62.8
Total Payload	14 333	497.3

Figure 6.6-2

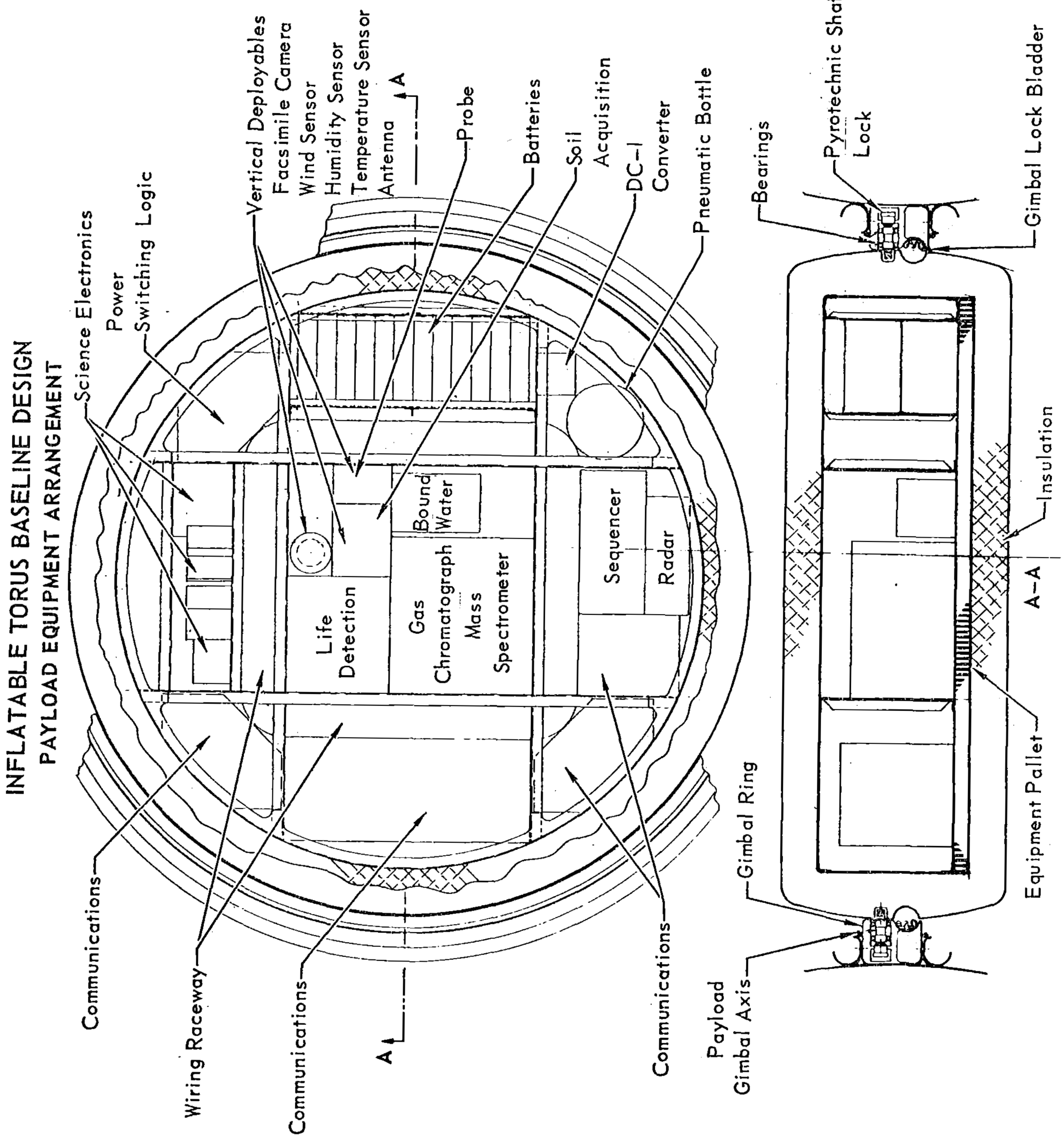

INFLATABLE TORUS BASELINE DESIGN
PAYLOAD EQUIPMENT ARRANGEMENT
Science Electronics
Power
Switching Logic
Vertical Deployables
Facsimile Camera
Wind Sensor
Humidity Sensor
Temperature Sensor
Antenna
A
Probe
Batteries
Soil
Acquisition
DC-I
Converter
Pneumatic Bottle
Bound
Water
Life
Detection
Gas
Chromatograph
Mass
Spectrometer
Sequencer
Radar
Communications
Wiring Raceway
Communications
A
Communications
Payload
Gimbal Axis
Gimbal Ring
Bearings
Pyrotechnic Shaft
Lock
Gimbal Lock Bladder
Insulation
A-A
Equipment Pallet

Figure 6.6-3

velocity with components along the Z_f axis and in the X_f-Y_f plane. The lander geometry considered is given in Section 6.4.

The following figures show the translational and angular lander velocity time histories and lander center of gravity trajectories for three of the cases. In these cases, the direction of the initial velocity's component in the X_f-Y_f plane was referenced to the Y_f axis. The three cases consist of this velocity component directed (1) along the Y_f axis (down slope), (2) 45 degrees from the Y_f axis (45 degrees down slope), and (3) 90 degrees from the Y_f axis (cross slope). Also, additional load time history information is presented for the down slope condition.

The translational and angular velocity time histories are shown in Figures 6.7-1 through 6.7-4. From Figure 6.7-4, it is seen that the cross slope condition resulted in the highest angular velocities. In addition, this case experienced only one impact in the time interval considered. This first impact was much longer in time duration than the first impact for the other two cases. The initial impact for the down slope and the 45 degree down slope cases, ended at about the same time. However, the latter case hits the ground sooner for the second impact.

The trajectory of the lander center of gravity is shown in Figure 6.7-5. This information is presented as the projections of the center of gravity motion on the plane of the landing surface and the Y_{ls} -Z_{ls} plane. It can be seen, that in both the 45 degree down slope and cross slope cases, the lander center of gravity experienced quite a bit of down hill motion relative to the direction of the initial velocity.

Additional information for the down slope case is shown in Figure 6.7-6. Here, the time histories of the forces and moments acting at the lander center of gravity are shown. The loads are referenced to the lander coordinate system.

LANDER CENTER OF GRAVITY VELOCITY IN LOCAL SURFACE X DIRECTION

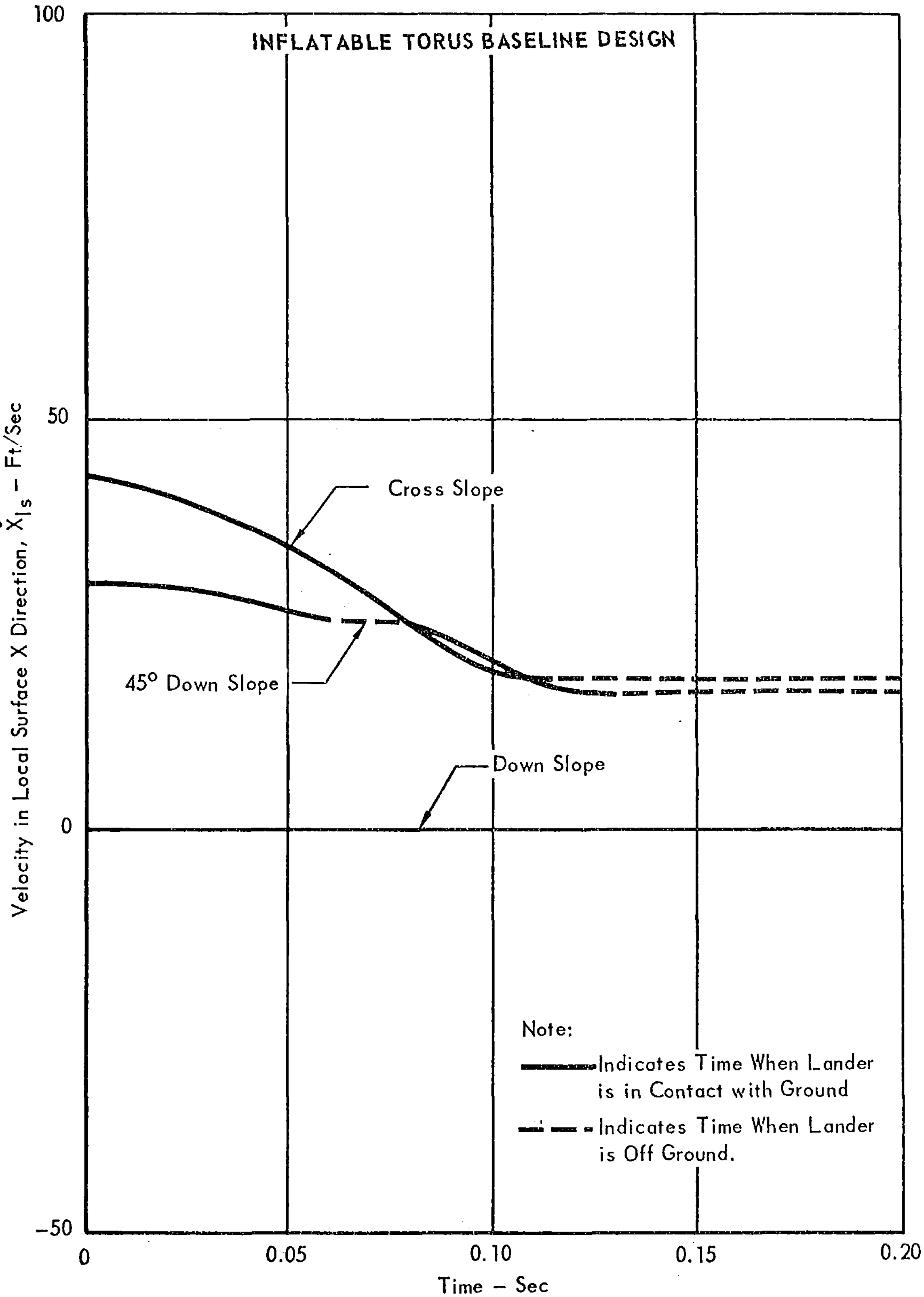


Figure 6.7-1

LANDER CENTER OF GRAVITY VELOCITY IN LOCAL SURFACE Y DIRECTION

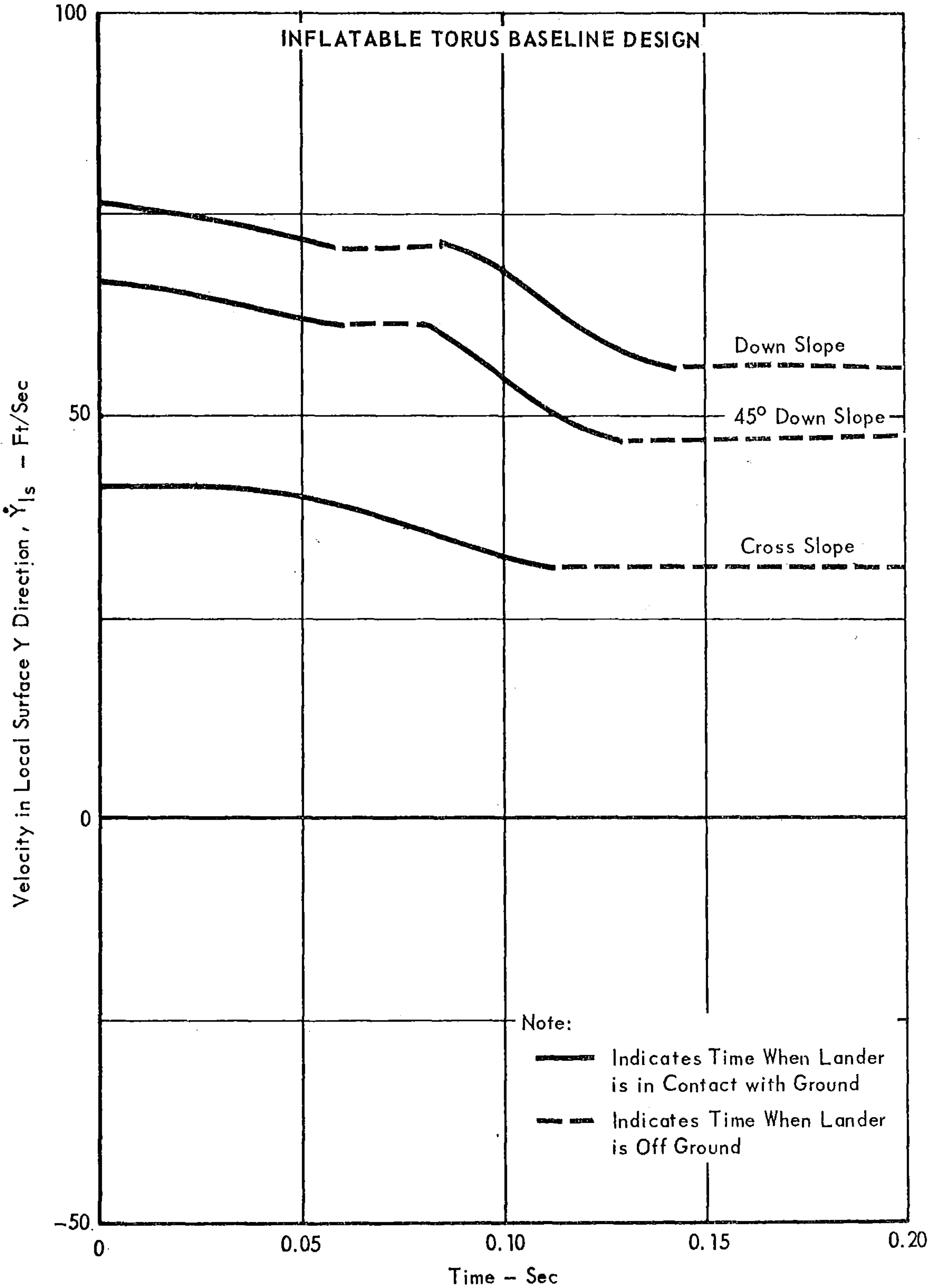


Figure 6.7-2

LANDER CENTER OF GRAVITY VELOCITY IN LOCAL SURFACE Z DIRECTION

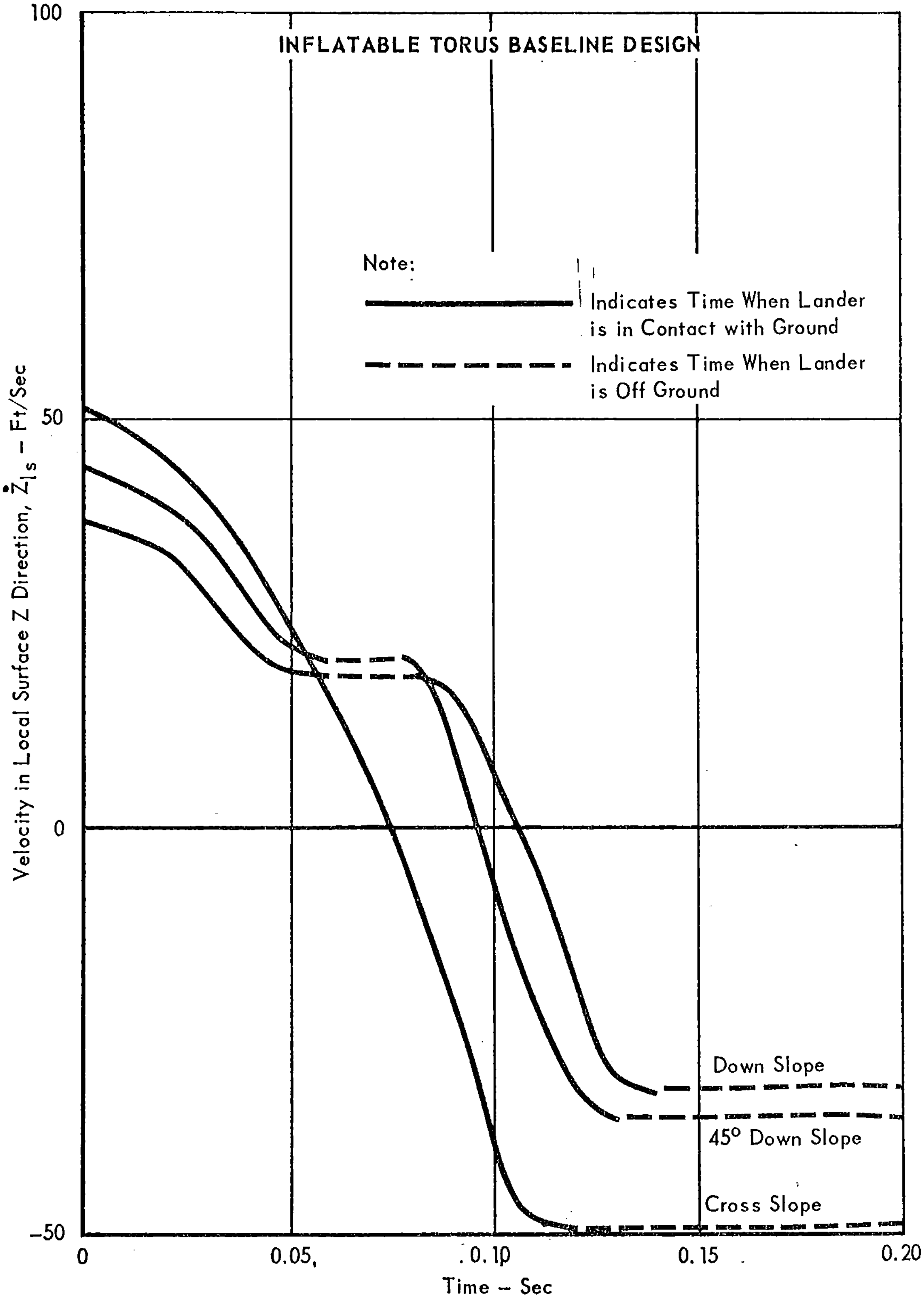


Figure 6.7-3

ANGULAR VELOCITIES ABOUT LANDER
CENTER OF GRAVITY
INFLATABLE TORUS BASELINE DESIGN

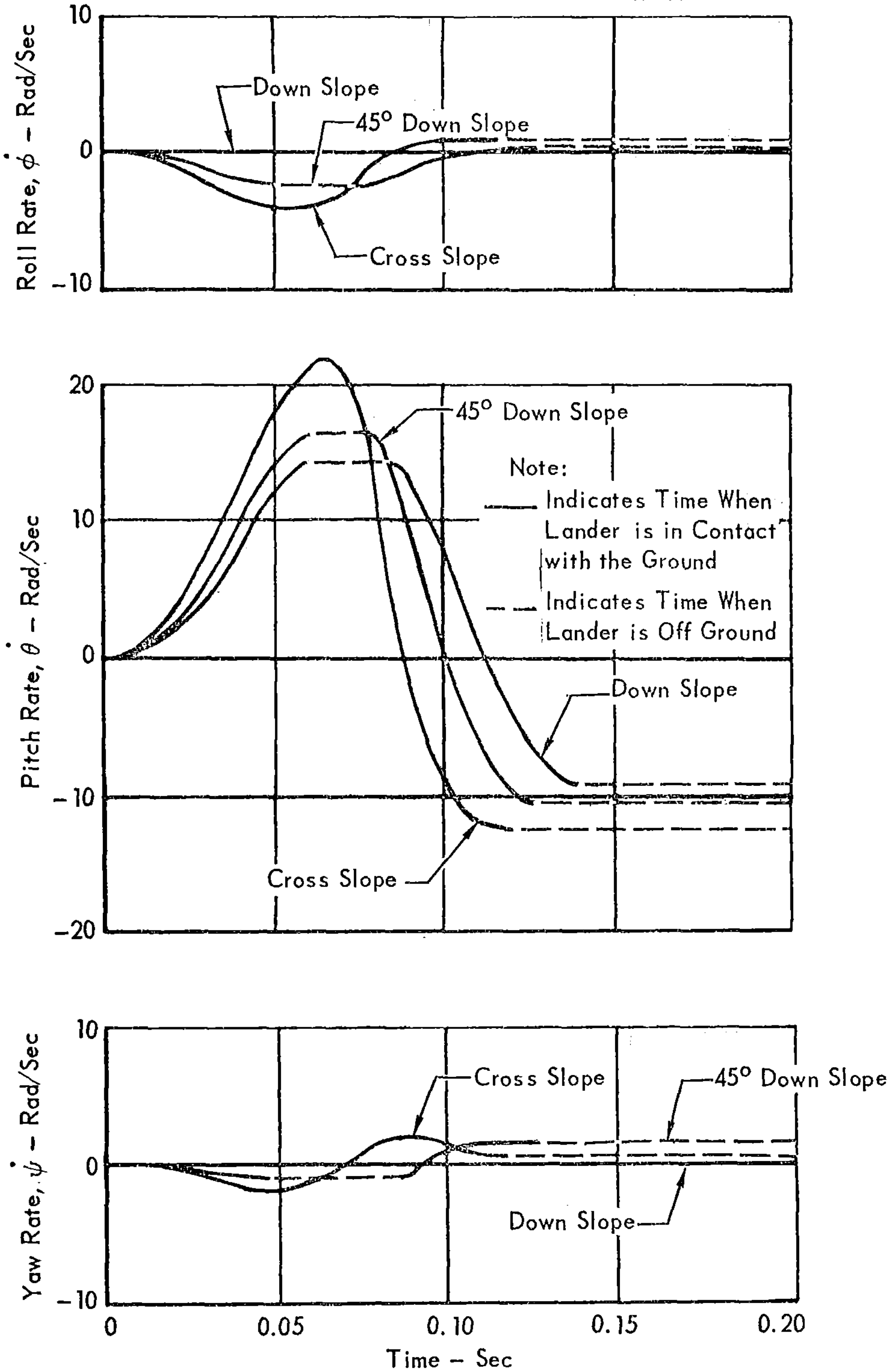


Figure 6.7-4

LANDER CENTER OF GRAVITY TRAJECTORIES IN LOCAL SURFACE COORDINATES

INFLATABLE TORUS BASELINE DESIGN

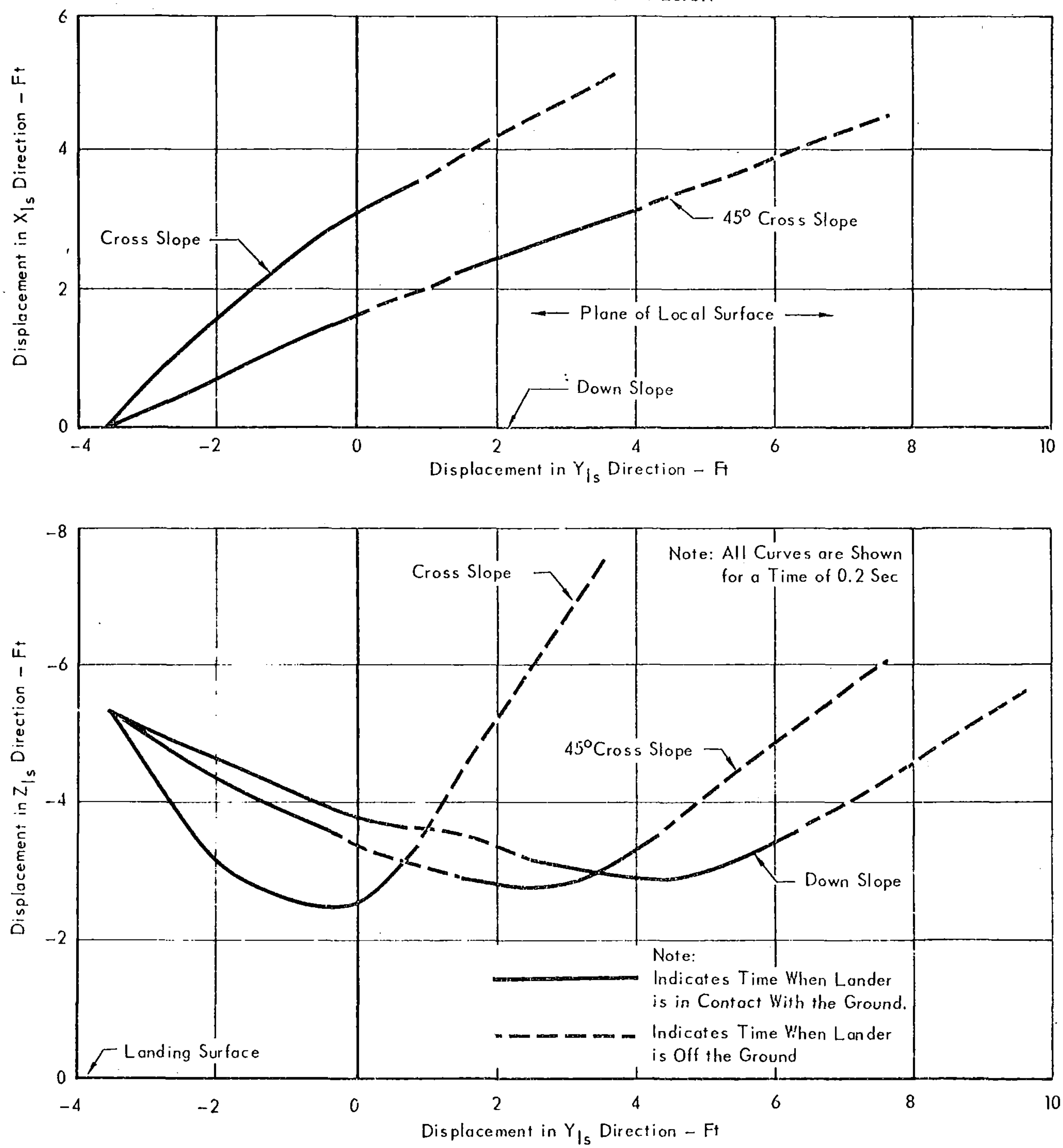
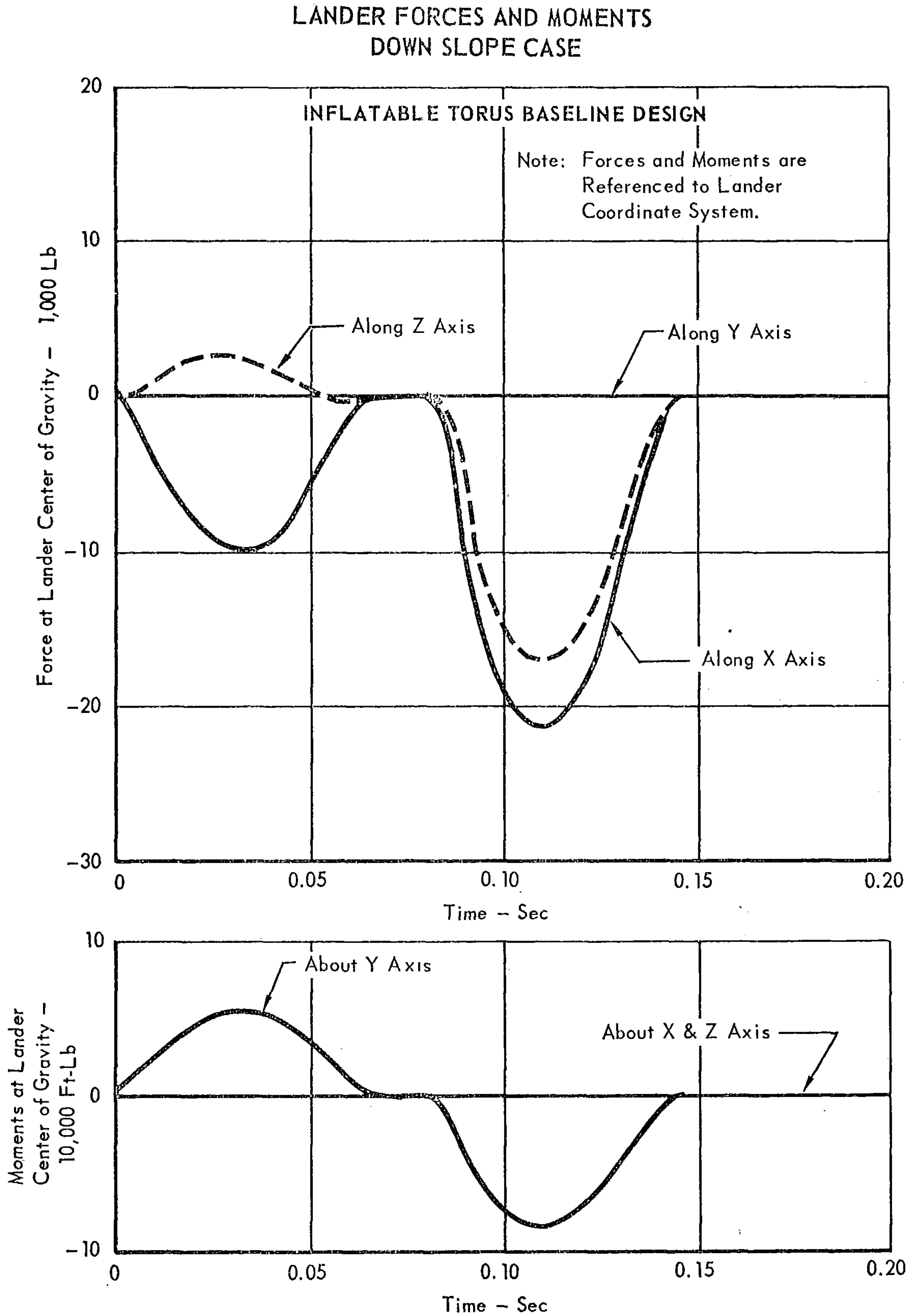


Figure 6.7-5



LANDER FORCES AND MOMENTS
DOWN SLOPE CASE
INFLATABLE TORUS BASELINE DESIGN
Note: Forces and Moments are Referenced to Lander Coordinate System.
Along Z Axis
Along Y Axis
Along X Axis
Force at Lander Center of Gravity – 1,000 Lb
20
10
0
-10
-20
-30
0
0.05
0.10
0.15
0.20
Time – Sec
About Y Axis
About X & Z Axis
Moments at Lander Center of Gravity – 10,000 Ft-Lb
10
0
-10
0
0.05
0.10
0.15
0.20
Time – Sec

Figure 6.7-6

7. CONCLUSIONS

Both the Crushable Torus and the Inflatable Torus Structural Design Programs developed in this study have been shown to be capable of satisfactorily establishing configurations meeting the criteria specified in this report. Utilizing the baseline configurations established with these programs, the capability of the Crushable Torus and the Inflatable Torus Landing Loads and Motions Programs to predict six degrees of freedom motions was demonstrated.

The crushable torus configuration detailed in Section 5.4 has been analyzed using the computer programs developed in this study and has been shown to satisfactorily land a 15.5 slug payload at 85 feet per second with a maximum load factor of 300 earth g's and within the other constraints stipulated in Section 3. Similarly the inflatable torus configuration detailed in Section 6.4 has been shown to satisfactorily land the same payload under the same conditions with a maximum load factor of 150 earth g's.

The inflatable torus may be tailored to provide any desired maximum load factor and velocity capability with minimum landing system weight by varying bag radius, inflation pressure, and payload shape. The baseline torus employs a payload shape parameter A/C, which is the cylindrical payload diameter divided by the payload height, of 4. This was found to be optimum, resulting in the lightest inflatable landing system weight for the specific design constraints studied.

Use of a spherical crushable lander results in the lowest design landing load factors, while use of a crushable torus employing a relatively flat payload results in the lowest landing system weight and best payload accessability of the crushable concepts investigated. The baseline torus employs payload shape with $G/B = 3.9$ and $B/A = 4.0$, which was found to result in the lightest crushable landing system weight for the specific design constraints studied. The ratio G/B is the radius of the cylindrical portion of the payload divided by the payload half height, while B/A is the ratio of the major to minor radii of the outer elliptical portion of the payload. Aluminum trussgrid with a density of 3.0 lb/ft^3 was selected for the baseline crushable attenuator material because its low crush stress (31 psi) results in low load factors. Also, it possesses good transverse strength which minimizes problems of internal attenuator crushing (payload cannon balling) encountered when materials with little or no transverse crush strength were used. Plastic foams also appear to have the desired material properties to achieve landing load factors in the intermediate category, but the specific foams investigated generally produced a lander which was too large for stowage in the aeroshell.

Several assumptions were required in developing the analytical models for the four computer programs. Future program refinement or expansion could be directed toward elimination of the more restrictive assumptions discussed in the following paragraphs.

An important assumption made in the crushable torus programs is that the crush plane always contacts virgin material. This assumption is violated if successive crush planes intersect and may occur when the lander initial conditions are such that high rotational rates are imparted to the vehicle after attenuator stroke has reached a maximum. This did not lead to appreciable errors for the many cases studied (see discussion in Section 5.7). Elimination of this assumption, requiring computer memory of deformed attenuator geometry, is necessary to be analytically precise.

Both inflatable torus programs employ a semi-empirical technique for predicting the footprint area of the deformed torus. This procedure was based on static tests conducted on a small inflatable torus model. Further testing and analysis could refine this semi-empirical approach for the prediction of footprint area.

Results from the inflatable torus model drop test program indicated there is interaction between the payload and the torus material during landing. Use of a multiple degree of freedom dynamic model would yield an improved prediction of these landing motions. A multi-mass/multi-spring model was initially investigated in this study. However, it was not selected because it could not be developed and programmed within the constraints defined for this study. As was demonstrated in Section 6.7, the dynamic model chosen gave quite satisfactory results.

8. REFERENCES

1. Statement of Work for Task Order One, of Master Agreement Contract NAS1-8137 "Analyses and Evaluation of Payload and Landing System Structures for Intermediate Type Planetary Lander."

2. "Mars Lander Capsule Study (Entry from Orbit) - Analysis of Inflatable Landing Systems," NASA CR 66665-9, April 1969.

3. "The Mechanical Characteristics of Pneumatic Tyres," R. Hadekel, British S & T Memo No. 5/50, March 1950.

4. "Mechanical Properties of Pneumatic Tires with Special Reference to Modern Aircraft Tires," R. F. Smiley & W. B. Horne, NASA TR R-64.

5. "An Analysis of the Impact Motion of an Inflated Sphere Landing Vehicle," E. Dale Martin and John T. Howe, NASA TN D-314, April 1960.

6. "Voyager Impact Attenuation System Study," Goodyear Aerospace Corporation, Report GER-12838, Rev. B, March 8, 1967. (This document presents the results of the fabrication, testing, and analysis of a model inflatable torus conducted by Goodyear Aerospace Corporation for McDonnell Douglas Astronautics Company.)

APPENDIX A

OPERATING INSTRUCTIONS
FOR THE CRUSHABLE TORUS
STRUCTURAL DESIGN PROGRAM

APPENDIX A

TABLE OF CONTENTS

APPENDIX A

A.1 INTRODUCTION

This program, which has been written for machine computation on the CDC 6400/6600 computer, provides the capability for either establishing crushable torus configurations meeting design constraints or for evaluating existing configurations. Operation of this program requires initial selection of a payload, payload-attenuator overlap, attenuator material, and desired flat and end velocity capability. The configuration design portion of this program then determines the required attenuator dimensions providing these velocity capabilities. If it is desired to check the velocity capability for any constant attitude between flat and end, the omnidirectional loads portion of this program may be used with the output geometry from the configuration design portion. This program was divided into two parts to allow maximum flexibility, since it may not always be necessary to use the omnidirectional loads portion.

The procedure used to determine the load-stroke relationship in the omnidirectional loads portion of this program is analytically identical to the load-stroke subroutine of the Crushable Torus Landing Loads and Motions Program. This subroutine, which calculates normal crush force as a function of stroke and lander attitude, employs all of the assumptions and derivations found in this Appendix except that it is of course not restricted to constant attitude crushing. Additionally a variable ramp elastic stroke recovery, simulating stored elastic energy producing rebound, has been added to the subroutine (See Appendix B).

APPENDIX A

Dimensional variables used to represent the payload and attenuator are shown in Figure A-1. The payload is composed of a cylinder with elliptical shaped sides. The attenuator is assumed to be oriented with its principal crushing axis perpendicular to the payload surface. Attenuator shape is defined by elliptical, circular, and cylindrical shaped surfaces. By appropriate dimensional variation, crushable spheres can be investigated, as well as various torus configurations.

Besides providing the ability to vary payload shape, the program allows variation of the following parameters: attenuator material properties; rock diameter; friction coefficient; desired flat and end velocity capability; and inclination of the contact angle between the planet surface and the vehicle axes (constant angle throughout stroke). Load factor can be minimized by parametrically varying input payload shape, attenuator-payload overlap, and/or attenuator material until the actual internal crush force approaches the allowable internal crush force. Attenuator material properties used in the program are: radial, shear, and transverse crush stress; density; and attenuator-payload bond shear stress. An input exponential factor determines the interaction between shear and radial stress.

Output data includes: required attenuator dimensions; payload weight; landing system weight; and maximum load factor and velocity capability for flat landing, end landing, and landing at the input contact angle. Data output as a function of stroke for flat landing, end landing, and landing at the input contact angle includes: footprint area; normal crushing force and friction force for the input coefficient of friction (assumed acting

APPENDIX A

CRUSHABLE TORUS GEOMETRY

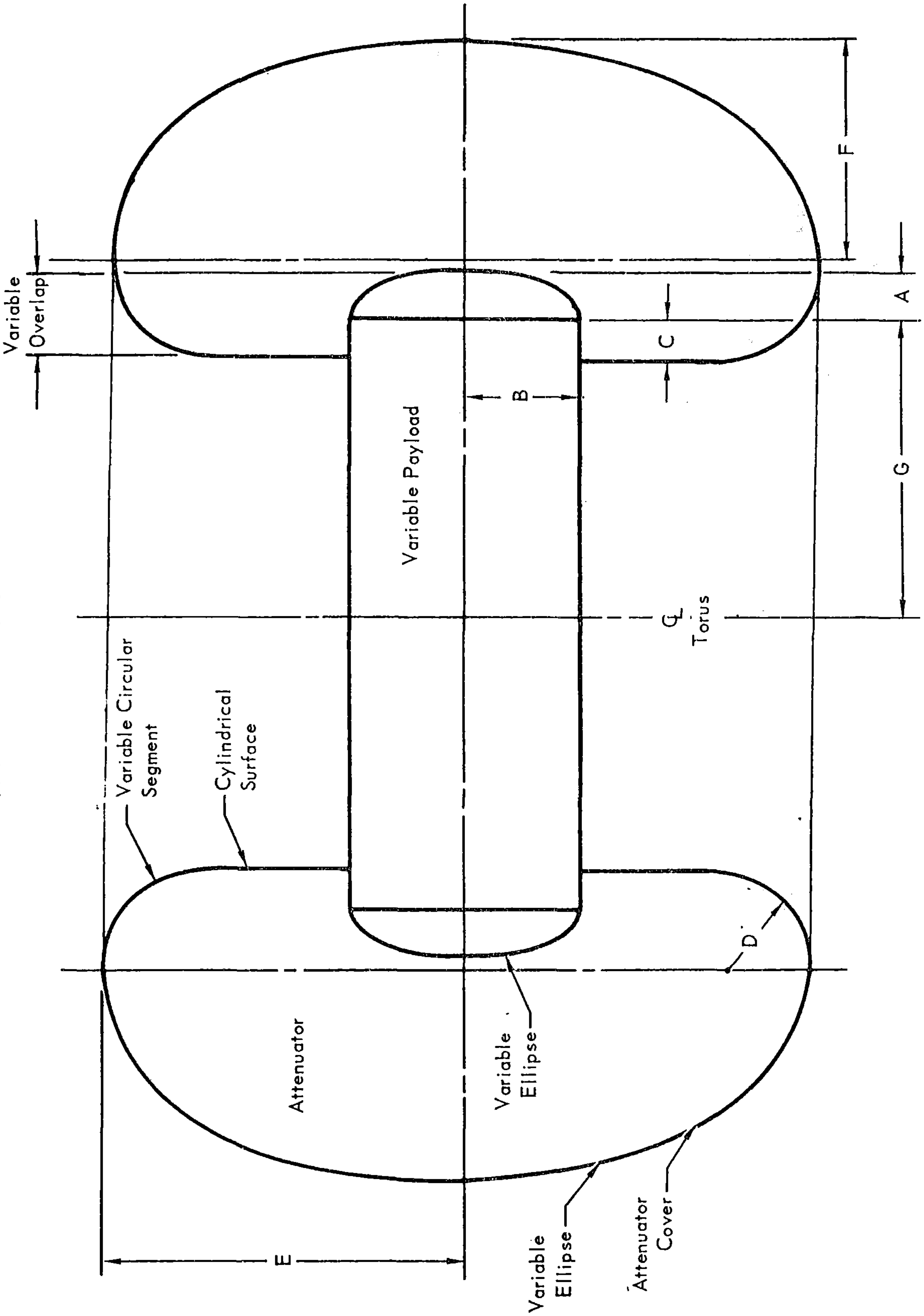


Figure A-1

throughout stroke); normal crushing force for zero coefficient of friction; and maximum allowable force parallel to the crush plane (with no normal force acting on the crush plane). With friction acting a smaller normal force will crush the attenuator and a larger volume of attenuator material is required. Load factor is generally a maximum when there is no friction acting, so that both cases must be investigated. Internal attenuator crushing (payload cannon balling) may be determined by comparing calculated values for actual internal crushing force and allowable internal crushing force (both flat and end landings).

APPENDIX A

A.2 ANALYTICAL PROCEDURES

The cylindrical coordinate system used in this program utilizes X and Y axes which rotate with the integration angle θ (see Figure A-2). Normal crush force acting on the crush plane is calculated by double integration; of DX and Dθ for flat landings, and of DY and Dθ for end landings. For comparison, the coordinate system (X, Y, Z) used in the Landing Loads and Motions Program is also shown.

Symbols, units, recommended range, and definition of required input parameters are given in Figures A-3 and A-4. Output parameters are defined in Figure A-5 and A-6. Format for machine input is discussed in Section A.3.

Major assumptions employed in the analytical model derivation are:

1. Attenuator oriented with principal crush axis perpendicular to the payload surface (radial orientation).
2. Stroke for ultimate velocity is equal to minimum of available height minus solid height, or available height minus rock diameter.
3. Energy absorbed is equal to area under the crush force vs. stroke curve (no shock wave energy dissipation).
4. Infinitely rigid surface - no proturberances or depressions.
5. Rigid body payload.
6. No internal crushing - although program prints out a check on this.
7. Crushed material is located at surface contact plane.
8. Elemental crush stress is constant with stroke.
9. Shear and radial crush stress are combined by an input exponential interaction equation.

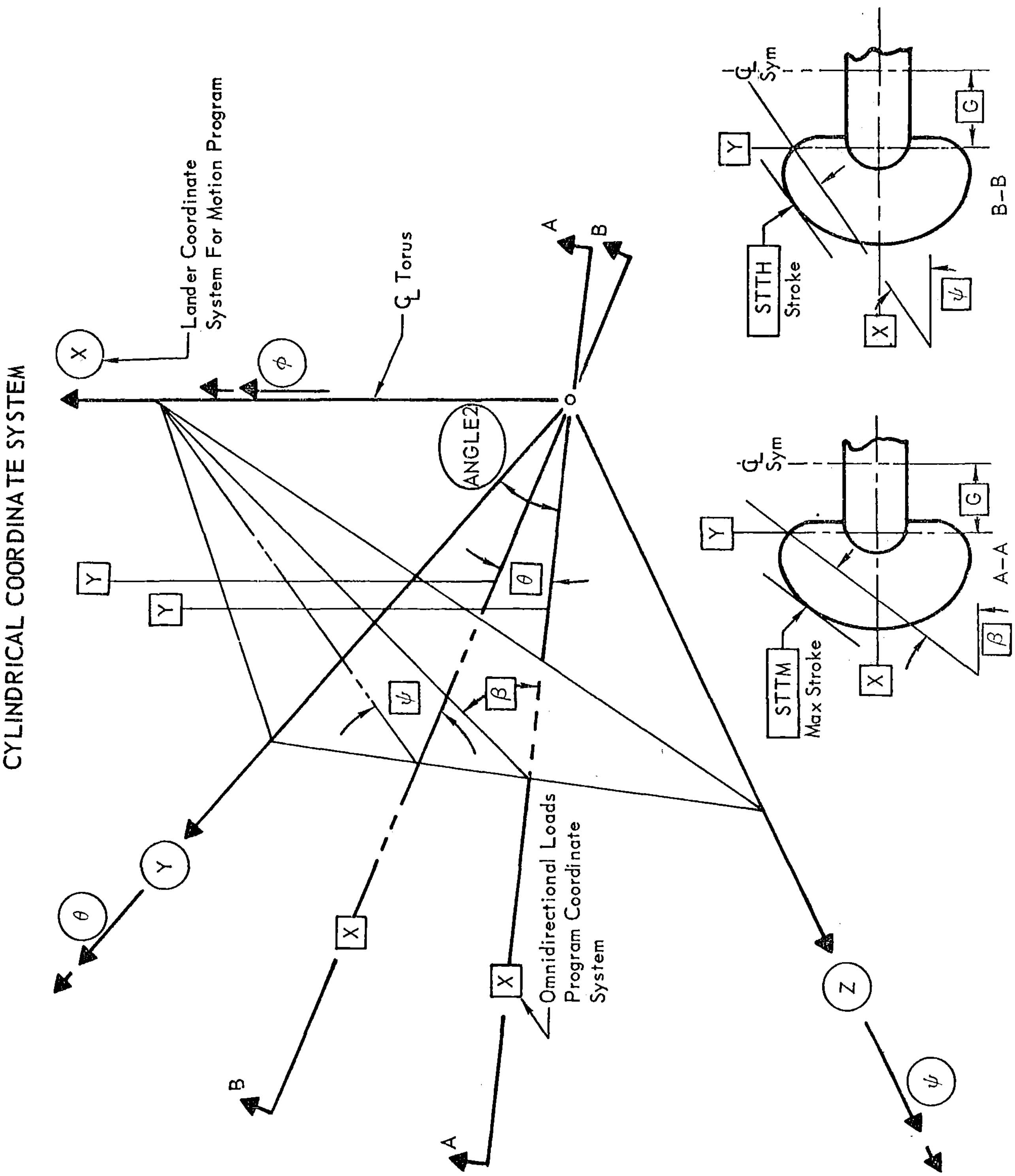

CYLINDRICAL COORDINATE SYSTEM
Lander Coordinate System For Motion Program
℄ Torus
ANGLE2
Omnidirectional Loads Program Coordinate System
X
Y
Z
A
B
STTH
Stroke
STTM
Max Stroke
℄ Sym
G
B–B
A–A

Figure A-2

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM
INPUT PARAMETERS – PART A

SYMBOL	UNITS	RECOMMENDED RANGE	DEFINITION
A,B,G	Inches	–	Payload Dimensions – See Figure A–1
RHOP	Lb/Ft^3	40 → 80	Payload Density
C,D	Inches	–	Attenuator Overlap – See Figure A–1
RHOL	Lb/Ft^3	See Figure A–8	Attenuator Density
XKW	Ft-Lb/Lb	See Figure A–8	Attenuator Specific Energy
RDIA	Inches	0 → 5	Rock Diameter
VE	Ft/Sec	–	Desired Edge Velocity
VF	Ft/Sec	–	Desired Flat Velocity
E1, F1 E2, F2 E3, F3	Inches	–	Assumed Attenuator Dimensions for Interpolation (Program Automatically Determines These if No Values are Entered)
RNORM	Psi/Psi	0 → 1.0	Ratio of Attenuator Transverse to Radial Crush Stress
RSHR	Psi/Psi	0 → 0.7	Ratio of Payload Bond Shear to Radial Crush Stress of Attenuator
STREF	In./In.	0.5 → 0.8	Stroke Efficiency (Available Stroke Divided by Uncrushed Height)
RSHCR	Psi/Psi	0.1 → 0.7	Ratio of Attenuator Maximum Allowable Shear Stress to Radial Crush Stress
AMU	–	0.0 → 1.0	Coefficient of Friction (Constant Throughout Stroke)
POW	–	1 → 3	Exponent Used in Interaction Formula $\left(\frac{FONTT}{FONTTM}\right)^{POW} + \left(\frac{FOFTT}{FOFTTM}\right)^{POW} = 1.0$
HND	–	1 0	If Evaluation Given Design E1, F1 If Establishing New Design

Figure A-3

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM (OMNIDIRECTIONAL LOADS)
INPUT PARAMETERS – PART B

SYMBOL	UNITS	RECOMMENDED RANGE	DEFINITION
A,B,G	Inches	–	Payload Dimensions – See Figure A–1
WT	Pounds	–	Total Weight of Payload and Landing System (From Part A)
C,D,E,F	Inches	–	Attenuator Dimensions – See Figure A–1 (E & F From Part A)
RHOL	Lb/Ft^3	See Figure A–8	Attenuator Density
XKW	Ft-Lb/Lb	See Figure A–8	Attenuator Specific Energy
BET	Radians	0 → + 1.57	Angle Between Crush Plane and X Axis (Constant Throughout Stroke) See Figure 2
RDIA	Inches	0 → 5	Rock Diameter
DX	Inches	0.05*A → 0.02*A	Incremental X
DTHE	Radians	0.005 → 0.015	Incremental Angle THE (Plan View)
RNORM	Psi/Psi	0 → 1.0	Ratio of Attenuator Transverse to Radial Crush Stress
CN	–	5 → 20	Number of Steps Desired in Incrementing Stroke
RSHCR	Psi/Psi	0.1 → 0.7	Ratio of Attenuator Maximum Allowable Shear Stress to Radial Crush Stress
STREF	In/In.	0.2 → 0.5	Stroke Efficiency (Ratio of Attenuator Stroke Allowable to Precrushed Height)
AMU	–	0 → 1.0	Coefficient of Friction (Constant Throughout Stroke)
POW	–	1 → 3	Exponent Used In Interaction Formula $\left(\frac{FONTT}{FONTTM}\right)^{POW} + \left(\frac{FOFTT}{FOFTTM}\right)^{POW} = 1.0$

Figure A-4

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM
OUTPUT PARAMETERS – PART A

SYMBOL	UNITS	DEFINITION
E,F	Inches	Attenuator Dimensions – See Figure A-1
WP	Pounds	Weight of Payload
WL	Pounds	Weight of Landing System
WT	Pounds	Total Weight (Landing System and Payload)
FLOADF	Earth g's	Maximum Load Factor for Flat Landings
ELOADF	Earth g's	Maximum Load Factor for End Landings
VELE	Ft/Sec	Actual Velocity Capability, End Landing
VELF	Ft/Sec	Actual Velocity Capability, Flat Landing
SCR	Inches	Stroke at Which FLOADF Occurs
XMT	Inches	Stroke at Which ELOADF Occurs
FIQ	Pounds	Force at End of Stroke (End Landing)
FIFO	Pounds	Force at End of Stroke (Flat Landing)
WL 1,2,3	Pounds	Weight of Landing System for Dimension E1, F1; E2, F2; E3, F3
V1E, V2E V3E	Ft/Sec	Velocity Capability (End) for Dimension E1, F1; E2, F2; E3, F3
V1F, V2F, V3F	Ft/Sec	Velocity Capability (Flat) for Dimension E1, F1; E2, F2; E3, F3
V4E, V4F	Ft/Sec	Velocity Capability (End Flat) for Dimensions E4, F4
E4, F4	Inches	Curve Fit Attenuator Dimensions Used for Final Iteration
FCR	Pounds	Internal Crush Force Acting (Maximum Flat)
ECR	Pounds	Internal Crush Force Acting (Maximum End)
FINCR	Pounds	Allowable Internal Crush Force (Flat)
EINCR	Pounds	Allowable Internal Crush Force (End)
S	Inches	Flat Stroke
T	Inches	End Stroke
FONTT	Pounds	Crush Force Normal to Crush Plane with Friction Acting
FOFTT	Pounds	Friction Force Acting with FONTT and Equal to AMU Times FONTT
FONTTM	Pounds	Crush Force Normal to Crush Plane with AMU = 0.0
FOFTTM	Pounds	Maximum Allowable Shear Force with Normal Force = 0.0
AREA	In^2	Footprint Area
FFLODF	Earth g's	Maximum Load Factor (Flat) with AMU = 0 and Velocity VF
EELODF	Earth g's	Maximum Load Factor (End) with AMU = 0 and Velocity VE
VELNE	Ft/Sec	Velocity Capability (End) at Stroke T with AMU = 0
VELNF	Ft/Sec	Velocity Capability (Flat) at Stroke S with AMU = 0

Figure A-5

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM (OMNIDIRECTIONAL LOADS)

OUTPUT PARAMETERS – PART B

SYMBOL	UNITS	DEFINITION
U	Inch-Pounds	Total Energy Absorbed in Crushing
CLEAR	Inches	Clearance Between Payload and Crush Plane
ALCL	Inches	Allowable Clearance Between Payload and Crush Plane Allowing 11.8% Stroke Margin for Ultimate Velocity Capability
VELBT	Ft/Sec	Velocity Capability at Angle BET
STTM	Inches	Maximum Stroke
FONTT	Pounds	Crush Force Normal to Crush Plane Acting with FOFTT
TORTT	Inch-Pounds	Total Torque Due to FONTT and FOFTT About Torus Center
ARTSU	In^2	Footprint Area
ARMOM	In^3	ARTSU Times Y Distance to Centroid
FOFTT	Pounds	Friction Force Acting With FONTT
FONTTM	Pounds	Crush Force Normal to Crush Plane (With AMU = 0)
TORTNM	Inch Pounds	Torque Due to FONTTM About Torus Center
FOFTTM	Pounds	Maximum Allowable Shear Force with Normal Force 0
TORTFM	Inch Pounds	Torque Due to FOFTTM About Torus Center

Figure A-6

10. Transverse crush stress is expressed as an input ratio to radial crush stress.

11. Attenuator cover does not contribute to crush force or allowable force parallel to the crush plane.

12. Frictional force equal to input coefficient of friction times normal force acts throughout stroke.

The elemental analytical model used to determine crushing force acting normal to the crush plane is shown in Figure A-7. Numerical integration of the force acting on incremental areas (DX x Dθ) is then performed between integration limits determined in the program. Flat and end landings are particular cases of the general omnidirectional landing attitude shown in Figure A-7. Energy absorbed is determined by numerical integration of the crush force versus stroke curve. Allowable clearance for ultimate velocity capability is calculated in both programs as the maximum of either the rock diameter (RDIA) or the available stroke times (1-STREF), where STREF is the ratio of the allowable stroke to the original height as determined from elemental tests. The allowable clearance is then increased to provide a 25% margin on design kinetic energy.

Typical attenuator material properties needed for input to both programs are shown in Figure A-8. These values are intended simply as a guide and values used in any specific application should reflect the actual material used. Attenuator density (RHOL) has been increased by 17% and specific energy has been reduced by 14% in Figure A-8 to account for the attenuator cover and adhesive.

ELEMENTAL ANALYTICAL MODEL

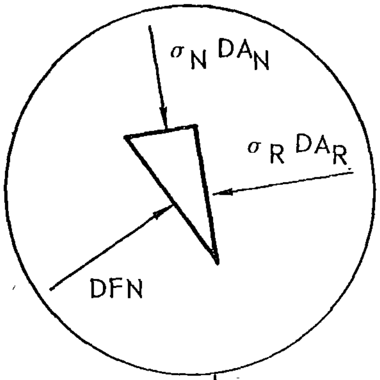
$$DFN = \sigma_R * DA_R * COS(\phi - \psi) + \sigma_N * DA_N * SIN(\phi - \psi)$$

$$FO = \sum_{\phi_{IL}}^{\phi_{OL}} DFN$$

$$FONTT = 2 * \sum_{\theta = 0}^{\theta max} FO * COS\,\beta / COS\,\psi$$

$$TAN\,\psi = TAN\,\beta * COS\,\theta$$

$$STTH = f(\beta, STTM, C, D, E, F, G)$$



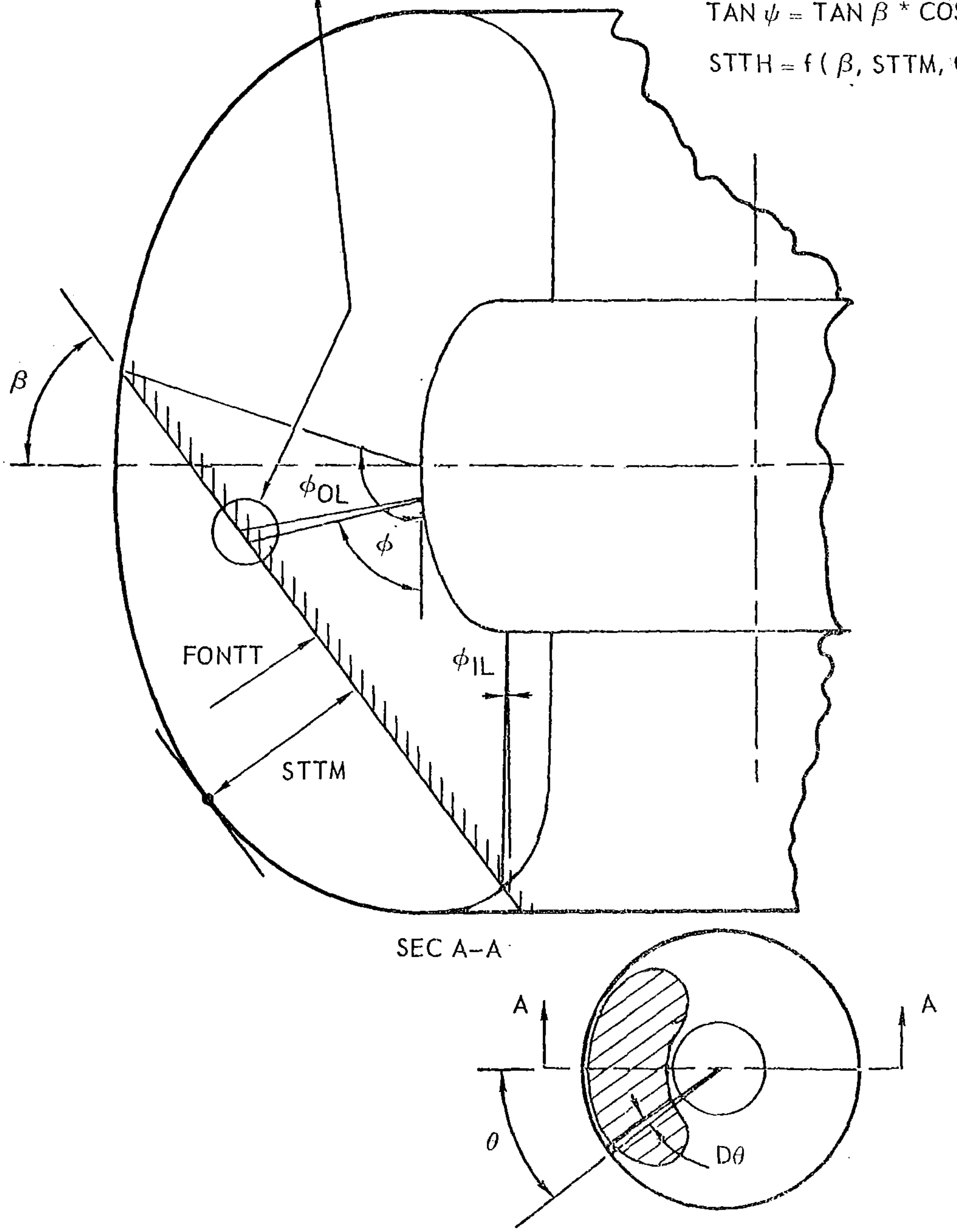


Figure A-7

TYPICAL ATTENUATOR PROPERTIES

ATTENUATOR MATERIAL	DENSITY RHOL LB/FT^3	SPECIFIC ENERGY XKW FT-LB/LB	STROKE EFFICIENCY STREF IN./IN.	TRANSVERSE STRESS RATIO RNORM PSI / PSI	SHEAR STRESS RATIO RSHCR PSI/PSI
Maraging Steel Honeycomb					
5 Lb/Ft^3	5.85	2720	0.90	0.0	1.0
20 Lb/Ft^3	23.40	9650	0.80	0.0	0.6
5052 Aluminum Honeycomb					
2 Lb/Ft^3	2.34	3540	0.80	0.0	1.2
15 Lb/Ft^3	17.56	8770	0.70	0.0	0.7
3003 Aluminum Honeycomb					
2 Lb/Ft^3	2.34	1560	0.80	0.0	1.0
15 Lb/Ft^3	17.56	3360	0.70	0.0	1.0
2.3 Lb/Ft^3	2.69	1335	0.80	1.0	0.9
Trussgrid (Aluminum)					
5 Lb/Ft^3	5.85	3470	0.80	1.0	0.9
7 Lb/Ft^3	8.20	4600	0.80	1.0	0.9
3. Lb/Ft^3	3.51	1720	0.80	1.0	0.9
Fiberglass (Phenolic) Honeycomb					
7.5 Lb/Ft^3	8.77	3530	0.80	0.0	0.6
11 Lb/Ft^3	12.87	7650	0.78	0.0	0.6
Balsa (Parallel To Grain)					
6 Lb/Ft^3	7.01	15400	0.80	0.0	0.2
16 Lb/Ft^3	18.70	16350	0.80	0.0	0.2
Perpendicular To Grain					
6 Lb/Ft^3	7.01	1470	0.80	0.0	2.0
Styrofoam					
1.8 Lb/Ft^3	2.10	2270	0.70	1.0	1.0
4.5 Lb/Ft^3	5.26	3440	0.70	1.0	0.8
Thermosetting Foams					
Lock Foam 2.44 Lb/Ft^3	2.85	374	0.60	1.0	1.0
Epoxy Foam 4.0 Lb/Ft^3	4.68	1570	0.50	1.0	0.8
Ecco Foam 10.0 Lb/Ft^3	11.70	1823	0.50	1.0	0.6

Figure A-8

APPENDIX A

Radial crush stress (sigma) acting on each element is determined in the program from the equation:

$$\text{Sigma} = \text{XKW} \times \text{RHOL}/(\text{STREF} \times 144.\frac{\text{in}^2}{\text{ft}^2})$$

Tangential crush stress acting with radial crush stress is found by multiplying sigma by RNORM, where RNORM may be any value between 0.0 and 1.0. Maximum allowable shear stress is found by multiplying sigma by RSHCR, where RSHCR must be greater than 0.0. Interaction between shear and radial crush force is determined in the program by the equation:

$$\left(\frac{\text{FONTT}}{\text{FONTTM}}\right)^{\text{Pow}} + \left(\frac{\text{FOFTT}}{\text{FOFTTM}}\right)^{\text{Pow}} = 1.$$

FONTT = Crush force normal to crush plane with friction acting

FOFTT = Force parallel to crush plane equal to FONTT times AMU

FONTTM = Crush force normal to crush plane with no friction acting

FOFTTM = Maximum allowable shear force parallel to crush plane

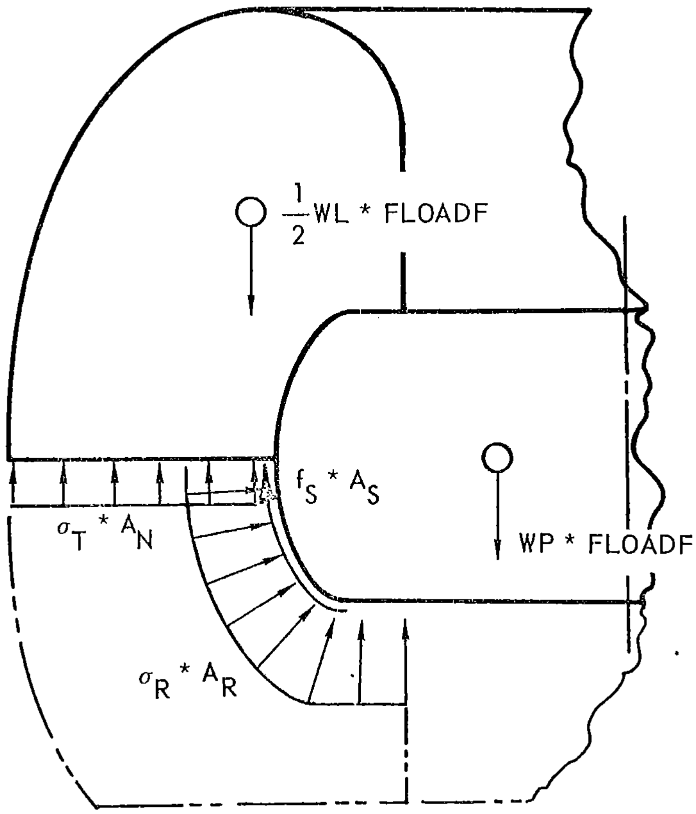
AMU = Surface coefficient of friction

POW = Exponential power describing interaction properties of specific attenuator material when used in above equation.

Both programs assume that the friction force, equal to AMU times FONTT, acts throughout the stroke.

The analytical model for internal crushing during flat landing is shown in Figure A-9. Actual and allowable internal crush forces are determined to ascertain whether or not internal crushing occurs. Acceleration is minimized when actual internal crush force approaches the allowable internal crush force. End landing employs a similar analytical model. RSHEAR

INTERNAL CRUSH ANALYTICAL MODEL



Allowable Internal Crush Force

$$F_{ALLOW} = F_R + F_S + F_T$$

Radial Crush

$$F_R = \pi\,\sigma_R(2AG + A^2 + 2GC - C^2)$$

Shear

$$F_S = \frac{\pi}{2} f_s(4BG + \pi AB)$$

Tangential Crush

$$F_T = \pi\,\sigma_T[(F + D - C + G)^2 - (G + A)^2]$$

Force Acting to Cause Internal Crushing:

$$F_{ACTUAL} = F_{LOAD\ F} * (WP + \frac{1}{2} WL)$$

To Prevent Internal Crushing:

$$F_{ALLOW} \geq F_{ACTUAL}$$

Figure A-9

is used in the structural design program to determine the shear strength of the payload to attenuator bond. The allowable shear stress is equal to RSHEAR times sigma. The value entered for RSHEAR should reflect the lower of either the adhesive bond shear strength or the attenuator shear strength.

Determination of the 45^o attitude (angle BET) velocity capability for torus designs employing a very flat payload ($G/B > 3$), using the omnidirectional loads portion of this program, results in calculated values which are too low. This is because the program assumes a constant attitude throughout stroking. In the actual case, the normal crush force does not act through the lander centroid and causes rotation of the lander towards a flat attitude. This rotation causes the footprint area and normal crush force to increase, thereby dissipating greater kinetic energy. For the baseline lander ($G/B = 3.9$), presented in Section 5.4, a velocity capability of 86% of the design velocity (85 fps) was calculated by the program for a 45^o attitude. The baseline lander was shown to be capable of landing successfully at a vertical velocity of 85 fps with an initial attitude of 45^o by exercising the Crushable Torus Landing Loads and Motions Program. As the payload shape parameter G/B approaches zero (sphere), the omnidirectional loads portion gives results which are more nearly correct because the assumption of no rotation becomes more representative.

A.3 PROGRAM OPERATION

Input format for the configuration design portion of this program is shown in Figure A-10. Data may be located anywhere within the eight spaces indicated for each parameter. Dimensions E1, F1; E2, F2; and E3, F3 may be any reasonable estimate of required attenuator dimensions. The program interpolates for final geometry based on solutions for these values. If a 0.0 value is entered for E2 the program will internally choose appropriate values for E1, F1; E2, F2; and E3, F3 for interpolation. Input data cards are placed behind the program as shown in Section A.4. If a specific lander configuration is to be evaluated without exercising the interpolation routine, an integer value of 1 is entered for HND and the desired geometry entered in E1, F1. Input format for the omnidirectional loads portion of this program is shown in Figure A-11, and the relative position of the input data cards in the program is shown in Section A.4.

Operation of the omnidirectional loads portion of this program requires certain inputs (dimensions E, F, and total weight WT) which are outputs of the configuration design portion of program. Angle of crush (BET) may be any value between 0.0 and +1.57 radians, with the specified value being maintained constant throughout stroke. The value entered for DX should be divisible into dimension A an integer number of times, with a value of DX equal to .02 or .05 times dimension A being recommended. Smaller values of DX will require longer machine times. DTHE also affects running time, and a range of values from .005 radians to .015 radians is suggested. The input variable

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM
EXAMPLE INPUT DATA PART A

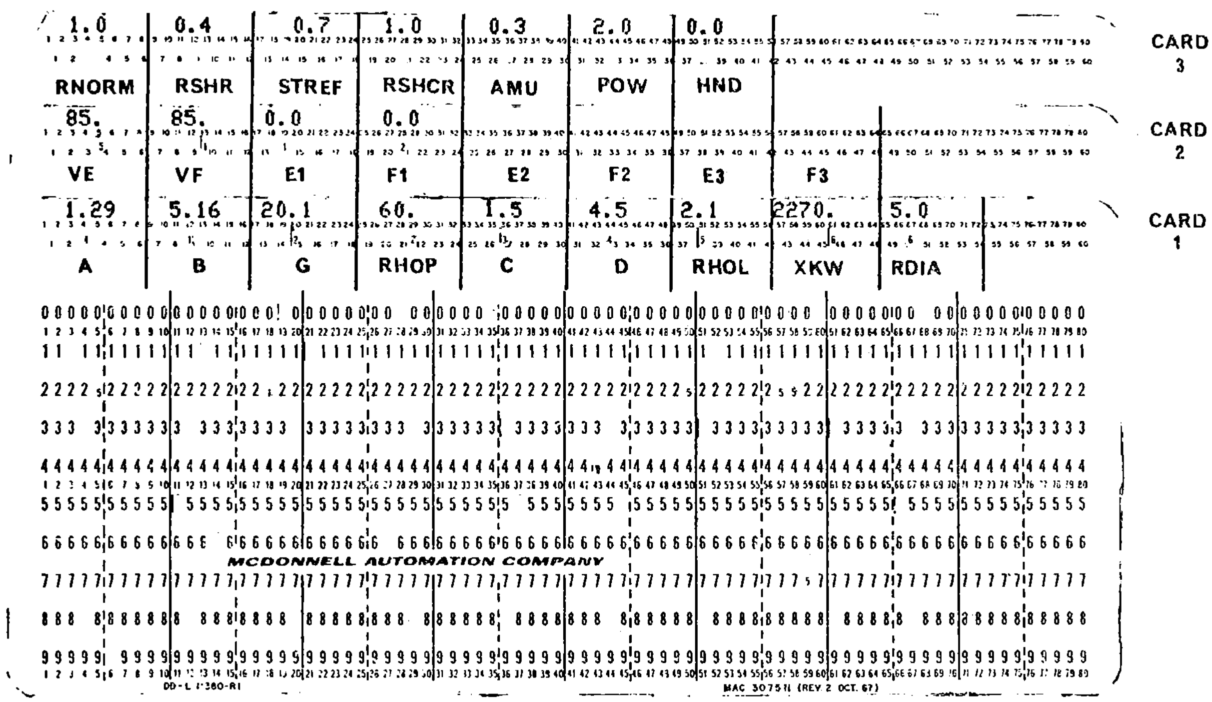


Figure A-10

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM
(OMNIDIRECTIONAL LOADS)
EXAMPLE INPUT DATA – PART B

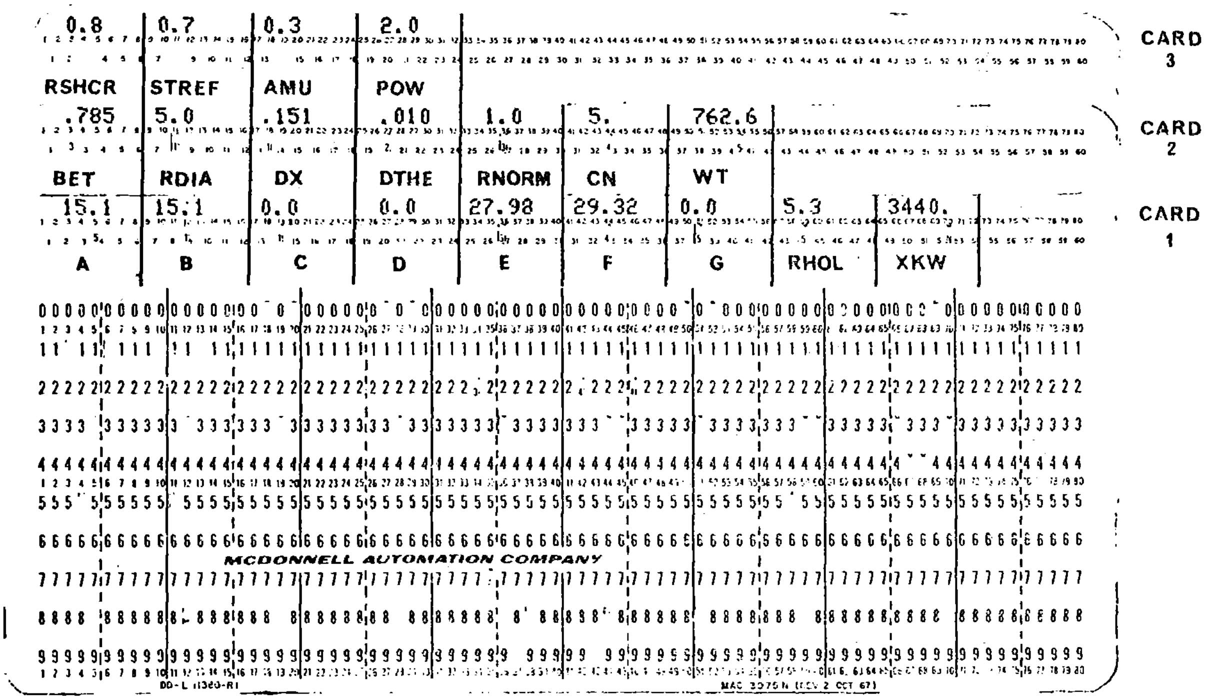


Figure A-11

CN determines the number of steps into which the stroke will be incremented. A range of 5 to 20 is suggested for CN.

Sample output data for both are shown in Figures A-12 and A-13. Refer to Figures A-5 and A-6 or comment cards in the program for nomenclature and units of output.

Program termination for the configuration design portion is automatic when attenuator dimensions E and F have been established for the desired capabilities and all necessary output data calculated. The omnidirectional loads portion then is used to determine velocity capability for constant input attitude (angle BET). Typical machine times are 50 seconds central processor and 5 seconds peripheral processor for the omnidirectional loads portion, and 200 seconds central processor and 5 seconds peripheral processor for the configuration design portion. Core size required is 50K (octal) for compilation of both the configuration design and the omnidirectional loads portions of this program.

APPENDIX A.
CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM.
Example Output Data - Part A

```
CRUSHABLE TORUS LANDER

                    INPUT PARAMETERS

A= 1.20   B= 5.16   C= 1.50   D= 4.50   G= 20.10   RHOP= 60.0   RHOL= 2.1
RDIA= 5.00      VE= 85.0   VF= 85.0      RNORM= 1.000   RSHR= .400   STREF= .700
RSHCR= 1.000   AMU= .300   POW= 2.00   XKW= 2270.0
E1= 17.62   F1= 21.56      E2= 19.49   F2= 26.12      E3= 21.35   F3= 30.69

                    OUTPUT DATA

E4= 17.55   F4= 26.58      E= 17.19   F= 27.38      WP= 501.7   WL= 215.2   WT= 716.8
FLOADF=   335.7   ELOADF=   130.2      FCR=   204507.6   FINCR=   339961.0      ECR=   79310.2   EINCR=   113920.0
FFLODF=   343.7   EELODF=   136.0
VELE= 82.6   VELF= 87.0      SCR= 6.29   XMT= 18.21      FIQ= 93315.1   FTFO= 240620.1
VIE= 71.6   VIF= 85.7      V2E= 83.8   V2F= 104.0      V3E= 95.2   V3F= 121.1      V4E= 81.7   V4F= 88.4
S= 6.28   T= 18.20      WL1= 166.8   WL2= 231.2   WL3= 308.9
```

LOAD STROKE FOR FLAT LANDING

S=	.314	AREA=	1071.91	FONTT=	51919.2	FOFTT=	15575.8	FONTTM=	54558.4	FOFTTM=	50692.5	VELNF=	8.0
S=	.628	AREA=	1548.74	FONTT=	75604.6	FOFTT=	22681.4	FONTTM=	79513.3	FOFTTM=	73242.3	VELNF=	14.9
S=	.942	AREA=	1923.52	FONTT=	90524.3	FOFTT=	27157.3	FONTTM=	94649.8	FOFTTM=	90966.5	VELNF=	20.6
S=	1.256	AREA=	2212.54	FONTT=	108620.9	FOFTT=	32586.3	FONTTM=	114305.4	FOFTTM=	104634.7	VELNF=	25.9
S=	1.570	AREA=	2496.41	FONTT=	121526.8	FOFTT=	36458.0	FONTTM=	127771.8	FOFTTM=	118059.3	VELNF=	30.9
S=	1.884	AREA=	2753.55	FONTT=	133224.0	FOFTT=	39967.2	FONTTM=	139980.1	FOFTTM=	130220.0	VELNF=	35.6
S=	2.198	AREA=	2989.64	FONTT=	144316.5	FOFTT=	43294.9	FONTTM=	151599.2	FOFTTM=	141384.8	VELNF=	40.2
S=	2.512	AREA=	3208.35	FONTT=	154303.7	FOFTT=	46291.1	FONTTM=	162028.8	FOFTTM=	151728.3	VELNF=	44.5
S=	2.826	AREA=	3412.26	FONTT=	163629.8	FOFTT=	49088.9	FONTTM=	171770.3	FOFTTM=	161371.4	VELNF=	48.7
S=	3.140	AREA=	3603.21	FONTT=	172380.5	FOFTT=	51714.2	FONTTM=	180913.0	FOFTTM=	170401.6	VELNF=	52.8
S=	3.455	AREA=	3782.58	FONTT=	180620.5	FOFTT=	54186.1	FONTTM=	189524.6	FOFTTM=	178884.4	VELNF=	56.8
S=	3.769	AREA=	3951.42	FONTT=	188935.8	FOFTT=	56680.7	FONTTM=	198276.7	FOFTTM=	186869.4	VELNF=	60.6
S=	4.083	AREA=	4110.56	FONTT=	196337.2	FOFTT=	58901.2	FONTTM=	206021.9	FOFTTM=	194395.3	VELNF=	64.4
S=	4.397	AREA=	4260.62	FONTT=	203355.1	FOFTT=	61006.5	FONTTM=	213370.2	FOFTTM=	201491.8	VELNF=	68.1
S=	4.711	AREA=	4402.66	FONTT=	210042.6	FOFTT=	63012.8	FONTTM=	220377.3	FOFTTM=	208209.3	VELNF=	71.8
S=	5.025	AREA=	4538.92	FONTT=	216506.4	FOFTT=	64951.9	FONTTM=	227155.3	FOFTTM=	214653.0	VELNF=	75.4
S=	5.339	AREA=	4669.84	FONTT=	222774.4	FOFTT=	66832.3	FONTTM=	233733.9	FOFTTM=	220844.5	VELNF=	78.9
S=	5.653	AREA=	4795.70	FONTT=	228867.7	FOFTT=	68660.3	FONTTM=	240136.6	FOFTTM=	226796.7	VELNF=	82.3
S=	5.967	AREA=	4916.74	FONTT=	234808.5	FOFTT=	70442.5	FONTTM=	246387.2	FOFTTM=	232520.6	VELNF=	85.7
S=	6.281	AREA=	5033.15	FONTT=	240620.1	FOFTT=	72186.0	FONTTM=	252512.3	FOFTTM=	238025.9	VELNF=	0.0

Figure A-12

APPENDIX A

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM

Example Output Data - Part A
(Continued)

LOAD STROKE FOR EDGE LANDING

T=	.910	AREA=	134.06	FONTT=	5476.6	FOFTT=	1643.0	FONTTM=	5670.3	FOFTTM=	6339.8	VELNE=	4.4
T=	1.820	AREA=	263.53	FONTT=	11289.4	FOFTT=	3386.8	FONTTM=	11730.9	FOFTTM=	12462.6	VELNE=	8.9
T=	2.731	AREA=	389.88	FONTT=	16932.9	FOFTT=	5079.9	FONTTM=	17614.6	FOFTTM=	18438.0	VELNE=	13.4
T=	3.641	AREA=	512.86	FONTT=	22213.8	FOFTT=	6664.1	FONTTM=	23103.0	FOFTTM=	24253.9	VELNE=	17.8
T=	4.551	AREA=	632.42	FONTT=	27470.2	FOFTT=	8241.1	FONTTM=	28576.4	FOFTTM=	29908.2	VELNE=	22.2
T=	5.461	AREA=	748.52	FONTT=	32766.3	FOFTT=	9829.9	FONTTM=	34107.7	FOFTTM=	35398.9	VELNE=	26.6
T=	6.371	AREA=	862.05	FONTT=	38100.5	FOFTT=	11430.2	FONTTM=	39692.5	FOFTTM=	40768.0	VELNE=	30.9
T=	7.281	AREA=	971.90	FONTT=	42785.4	FOFTT=	12835.6	FONTTM=	44558.2	FOFTTM=	45962.7	VELNE=	35.3
T=	8.192	AREA=	1078.97	FONTT=	47663.7	FOFTT=	14299.1	FONTTM=	49653.1	FOFTTM=	51026.2	VELNE=	39.6
T=	9.102	AREA=	1182.07	FONTT=	52225.7	FOFTT=	15667.7	FONTTM=	54406.2	FOFTTM=	55902.2	VELNE=	43.8
T=	10.012	AREA=	1283.23	FONTT=	57144.7	FOFTT=	17143.4	FONTTM=	59571.1	FOFTTM=	60686.0	VELNE=	48.0
T=	10.922	AREA=	1380.44	FONTT=	60584.4	FOFTT=	18175.3	FONTTM=	63078.3	FOFTTM=	65283.3	VELNE=	52.2
T=	11.832	AREA=	1474.55	FONTT=	65612.9	FOFTT=	19683.9	FONTTM=	68394.2	FOFTTM=	69733.9	VELNE=	56.3
T=	12.742	AREA=	1566.10	FONTT=	70263.3	FOFTT=	21079.0	FONTTM=	73294.5	FOFTTM=	74063.3	VELNE=	60.4
T=	13.653	AREA=	1653.85	FONTT=	73467.3	FOFTT=	22040.2	FONTTM=	76570.3	FOFTTM=	78213.4	VELNE=	64.5
T=	14.563	AREA=	1738.39	FONTT=	76013.7	FOFTT=	22804.1	FONTTM=	79118.3	FOFTTM=	82211.1	VELNE=	68.5
T=	15.473	AREA=	1820.00	FONTT=	78100.1	FOFTT=	23430.0	FONTTM=	81165.3	FOFTTM=	86070.7	VELNE=	72.4
T=	16.383	AREA=	1898.78	FONTT=	85553.8	FOFTT=	25666.1	FONTTM=	89278.4	FOFTTM=	89796.3	VELNE=	76.3
T=	17.293	AREA=	1974.70	FONTT=	90191.1	FOFTT=	27057.3	FONTTM=	94233.0	FOFTTM=	93386.8	VELNE=	80.3
T=	18.203	AREA=	2047.59	FONTT=	93315.1	FOFTT=	27994.5	FONTTM=	97477.4	FOFTTM=	96834.1	VELNE=	84.2

Figure A-12 (continued)

APPENDIX A

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM
(OMNIDIRECTIONAL LOADS)

Example Output Data - Part B

```
OMNIDIRECTIONAL LOADS CRUSHABLE TORUS

             INPUT PARAMETERS

A= 15.10  B= 15.10  C=  0.00  D=  0.00  E= 27.98  F= 29.32  G=  0.00

RHOL=  5.3   XKW=  3400.0   RNORM=1.000   RSHCR= .800   STREF= .700   AMU= .300   POW=2.000

BET= .785   RDIA= 5.00   DX= .1510   DTHE= .0100   WT=    762.6

             OUTPUT DATA

U=   1042695.0   CLEAR=     5.00   ALCL=     5.00   VELBT=    85.61

STTM= 0.000    FONTT=       0.0   FOFTT=      0.0   TORTT=         0.0   ARTSU=   0.00   ARMOM=      0.0
               FONTTM=      0.0   TORTNM=         0.0   FOFTTM=       0.0   TORTFM=        0.0

STTM= 1.715    FONTT=   50933.2   FOFTT=  15280.0   TORTT=    533409.3   ARTSU=  308.79   ARMOM=   5550.9
               FONTTM=  54201.0   TORTNM=    129542.5   FOFTTM=   44661.7   TORTFM=  1203827.1

STTM= 3.430    FONTT=   97105.9   FOFTT=  29131.8   TORTT=   1060094.5   ARTSU=  605.98   ARMOM=  10141.7
               FONTTM= 102954.0   TORTNM=    344754.8   FOFTTM=   87684.7   TORTFM=  2212067.4

STTM= 5.144    FONTT=  157435.4   FOFTT=  47230.6   TORTT=   2070134.8   ARTSU= 1005.34   ARMOM=  14037.9
               FONTTM= 166452.4   TORTNM=   1014559.2   FOFTTM=  145474.2   TORTFM=  3420541.8

STTM= 6.858    FONTT=  193584.9   FOFTT=  58075.5   TORTT=   2562722.0   ARTSU= 1283.74   ARMOM=  16752.1
               FONTTM= 203801.5   TORTNM=   1365181.7   FOFTTM=  185754.4   TORTFM=  4049230.0

STTM= 8.572    FONTT=  218368.5   FOFTT=  65510.6   TORTT=   2953691.0   ARTSU= 1488.46   ARMOM=  18370.8
               FONTTM= 229229.7   TORTNM=   1719384.0   FOFTTM=  215377.4   TORTFM=  4325834.5
```

Figure A-13

APPENDIX A

A.4 PROGRAM DESCRIPTION

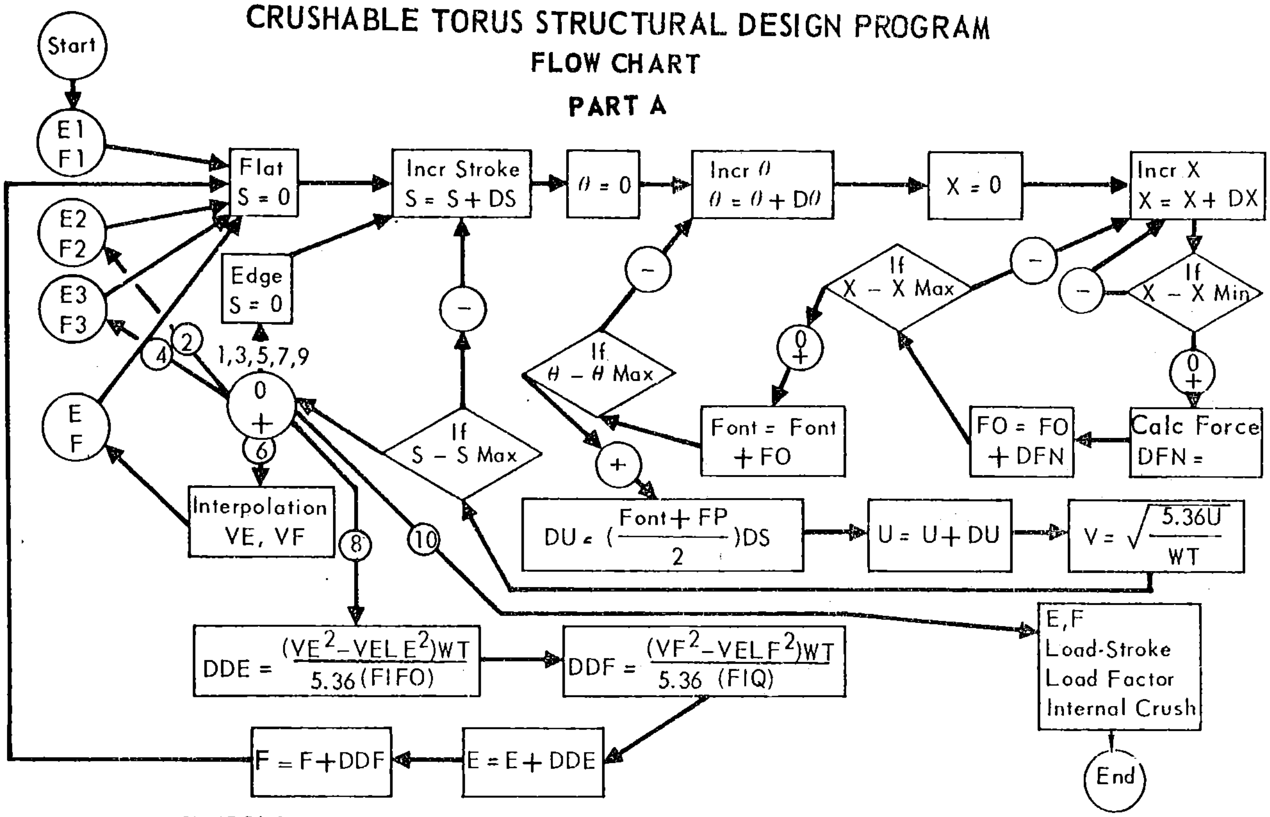
The flow chart shown in Figure A-14 shows major events in the configuration design program. Comment cards in the program additionally aid in identifying key portions of the program. Similarly, Figure A-15 shows the flow chart for the omnidirectional loads program.

The listings for the configuration design and omnidirectional loads portions of the Crushable Torus Structural Design Program are presented in Sections A.4.1 and A.4.2 respectively.

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM

FLOW CHART

PART A



SYMBOLS

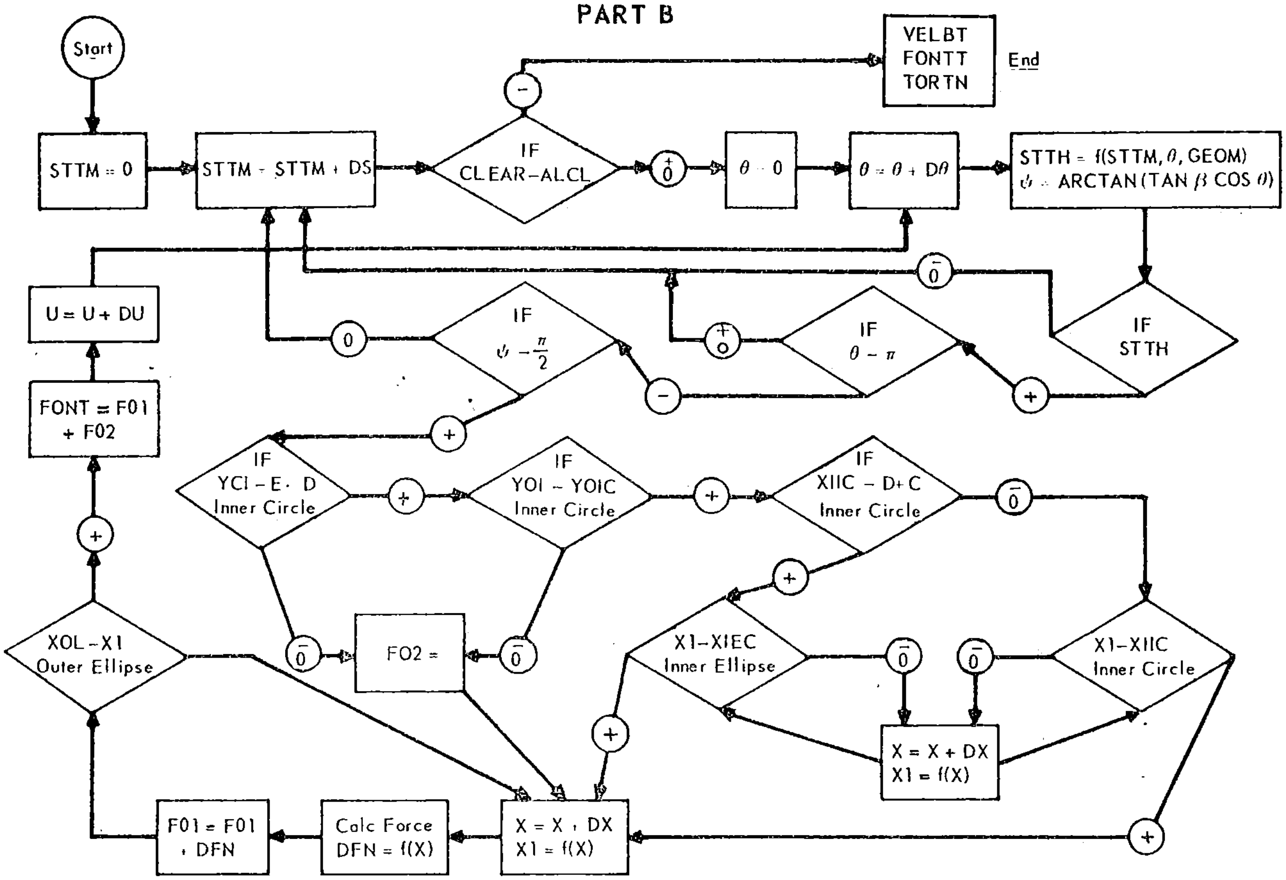
DDE	Final Adjustment to Dimension E to Obtain Desired Velocity VF
DDF	Final Adjustment to Dimension F to Obtain Velocity VE
DFN	Incremental Force Normal to Crush Plane on $D\theta$ by DX Area
DX	Incremental X
DS	Incremental Stroke
$D\theta$	Incremental Angle θ
DU	Incremental Energy Associated with DS
E1, 2, 3	Initial Estimates for Dimension E for Interpolation
F1, 2, 3	Initial Estimates for Dimension F for Interpolation
FIFO	Force at the End of the Flat Landing Stroke
FIQ	Force at the End of the End Landing Stroke
FP	Force at Preceeding Stroke Normal to Crush Plane
FO	Force Normal to Crush Plane at Given Stroke on $D\theta$ Strip
FONT	Total Force Normal to Crush Plane
S	Stroke
SMAX	Maximum Stroke Allowable
U	Energy Under Load Stroke Curve
V	Velocity Capability
VE	Desired Velocity Capability (End Landing)
VELE	Actual Velocity Capability (End Landing)
VF	Desired Velocity Capability (Flat Landing)
VELF	Actual Velocity Capability (Flat Landing)
WT	Total Weight
X	Dimension From Y Axis Locating Ray on Elliptical Payload
X_{MAX}	Integration Limit for Dimension X (Outer)
X_{MIN}	Inner Integration Limit for Dimension X
θ	Rotational Angle THE (Plan View)
θ_{MAX}	Integration Limit on Angle θ

Figure A-14

APPENDIX A

CRUSHABLE TORUS STRUCTURAL DESIGN PROGRAM (OMNIDIRECTIONAL LOADS)

FLOW CHART
PART B



SYMBOLS

ALCL	Allowable Clearance
CLEAR	Clearance Between Payload and Crush Plane
DFN	Incremental Force Normal to Crush Plane on Dθ by DX Area
DS	Incremental Stroke
DX	Incremental X
Dθ	Incremental Angle θ
FONT	Total Force on Dθ Strip Normal to Crush Plane Resolved to θ 0
FONTT	Total Force Normal to Crush Plane
F01	Normal Force on Dθ Strip (Build Up of DFN)
F02	Normal Force on Dθ Strip Inboard of Y Axis
STTM	Maximum Stroke at θ 0
STTH	Stroke at Angle θ
TORTN	Total Torque at Torus c.g.
X	Dimension from Y Axis Locating Ray on Elliptical Payload
X1	X Dimension to Intersection of Crush Plane and Ray from Payload
XIEC	X Dimension to Inboard Intersection of Crush Plane and Attenuator Ellipse
XIIC	X Dimension to Intersection of Crush Plane and Attenuator Circular Segment D
XOL	X Dimension to Outboard Intersection of Attenuator Ellipse and Crush Plane
YCI	Y Dimension to Intersection of Crush Plane and Vertical Line Y C
YOI	Y Dimension to Intersection of Crush Plane and X Axis
YOIC	Y Dimension Associated with XIIC
U	Energy Under Load-Stroke Curve
VELBT	Velocity Capability for Crushing at Constant Angle BET
θ	Rotational Angle THE (Plan View)
ψ	Angle PSI of Crush Plane With Horizontal at Angle θ

Figure A-15

APPENDIX A

A.4.1 CONFIGURATION DESIGN PROGRAM LISTING

```
      PROGRAM DORR    (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)             D3R  10
C     PROGRAM CALCULATES ATTENUATOR DIMENSIONS FOR DESIRED, VELOCITY      D3R  20
C     CAPABILITY (EDGE AND FLAT DIRECTIONS) OF GIVEN PAYLOAD              D3R  30
C     INPUT PARAMETERS                                                    D3R  40
C        A,B,G        PAYLOAD DIMENSIONS IN INCHES                        D3R  50
C        RHOP         PAYLOAD DENSITY IN POUNDS/CUBIC FOOT                D3R  60
C        C,D          ATTENUATOR OVERLAP DIMENSIONS                       D3R  70
C        RHOL         ATTENUATOR DENSITY IN POUNDS/CUBIC FOOT             D3R  80
C        XKW          ATTENUATOR SPECIFIC ENERGY IN FOOT-POUNDS/POUND     D3R  90
C        RDIA         ROCK DIAMETER IN INCHES                             D3R 100
C        VE           DESIRED EDGE VELOCITY CAPABILITY IN FEET/SECOND     D3R 110
C        VF           DESIRED FLAT VELOCITY CAPABILITY IN FEET/SECOND     D3R 120
C        F1,E1        ATTENUATOR DIMENSIONS ASSUMED (FOR CURVE FIT) INCHESD3R 130
C        F2,E2        ATTENUATOR DIMENSIONS ASSUMED (FOR CURVE FIT) INCHESD3R 140
C        F3,E3        ATTENUATOR DIMENSIONS ASSUMED (FOR CURVE FIT) INCHESD3R 150
C        RNORM        RATIO OF NORMAL TO RADIAL CRUSH STRESS              D3R 160
C        RSHR         RATIO OF PAYLOAD BOND SHEAR TO RADIAL CRUSH STRESS  D3R 170
C        STREF        STROKE EFFICIENCY (STROKE/INITIAL HEIGHT)           D3R 180
C        RSHCR        RATIO OF ATTENVATOR SHEAR ALLOWABLE TO RADIAL CRUSH D3R 190
C                     STRESS                                              D3R 191
C        AMU          COEFFICIENT OF FRICTION                             D3R 200
C        POW          EXPONENT USED IN INTERACTION OF ATTENVATOR SHEAR    D3R 210
C                     AND AXIAL STRESS                                    D3R 211
C        HND          INDICATOR 1.IF EVALUATING GIVEN DESIGN E1,F1        D3R 220
C                               0 IF ESTABLISHING NEW DESIGN              D3R 230
C     OUTPUT DATA                                                         D3R 240
C        E4,F4        CURVE FIT ATTENVATOR DIMENSIONS INITIAL SOLUTION    D3R 250
C                     (INCHES)                                            D3R 251
C        E,F          FINAL REQUIRED ATTENUATOR DIMENSIONS IN INCHES      D3R 260
C        WP           WEIGHT OF PAYLOAD IN POUNDS                         D3R 270
C        WL           WEIGHT OF LANDING SYSTEM IN POUNDS                  D3R 280
C        WT           TOTAL WEIGHT OF LANDING SYSTEM AND PAYLOAD IN POUNDSD3R 290
C        FLOADF       MAX LOAD FACTOR FOR FLAT LANDING AT VELF IN G*S     D3R 300
C        ELOADF       MAX LOAD FACTOR FOR EDGE LANDING AT VELE IN G*S     D3R 310
C        FCR          INTERNAL CRUSHING FORCE ACTING (MAX.FLAT) IN POUNDS D3R 320
C        FINCR        ALLOWABLE INTERNAL CRUSH FORCE (FLAT) IN POUNDS     D3R 330
C        ECR          INTERNAL CRUSHING FORCE ACTING (MAX.EDGE) IN POUNDS D3R 340
C        EINCR        ALLOWABLE INTERNAL CRUSH FORCE (EDGE) IN POUNDS     D3R 350
C        FFLODF       MAX LOAD FACTOR (FLAT) WITH AMU=0 AND VELOCITY VF   D3R 360
C                     (G'S)                                               D3R 361
C        EELODF       MAX LOAD FACTOR (EDGE) WITH AMU=0 AND VELOCITY VE   D3R 370
C                     (G'S)                                               D3R 371
C        VELE         ACTUAL VELOCITY CAPABILITY EDGE LANDING (FEET/SEC.) D3R 380
C        VELF         ACTUAL VELOCITY CAPABILITY FLAT LANDING (FEET/SEC.) D3R 390
C        SCR          STROKE AT WHICH LOAD PEAKS (FLAT) IN INCHES         D3R 400
C        XMT          STROKE AT WHICH LOAD PEAKS (EDGE) IN INCHES         D3R 410
C        FIQ          FORCE AT END OF STROKE (EDGE) IN POUNDS             D3R 420
C        FIFO         FORCE AT END OF STROKE (FLAT) IN POUNDS             D3R 430
C        V1E,V1F      VELOCITY CAPABILITY (EDGE,FLAT) FOR E1,F1 DIM(FT/SECD3R 440
C        V2E,V2F      VELOCITY CAPABILITY (EDGE,FLAT) FOR E2,F2 DIM(FT/SECD3R 450
C        V3E,V3F      VELOCITY CAPABILITY (EDGE,FLAT) FOR E3,F3 DIM(FT/SECD3R 460
C        V4E,V4F      VELOCITY CAPABILITY (EDGE,FLAT) FOR E4,F4 DIM(FT/SECD3R 470
C        WL1          WEIGHT OF LANDING SYSTEM FOR DIM.E1.F1 (POUNDS)     D3R 480
C        WL2          WEIGHT OF LANDING SYSTEM FOR DIM.E2,F2 (POUNDS)     D3R 490
C        WL3          WEIGHT OF LANDING SYSTEM FOR DIM.E3,F3 (POUNDS)     D3R 500
C        S            FLAT STROKE IN INCHES                               D3R 510
```

```
C        T             EDGE STROKE IN INCHES
C        AREA          FOOTPRINT AREA (SQUARE INCHES)
C        FONTT         FORCE NORMAL TO CRUSH PLANE ACTING WITH FOFTT (POUND
C        FOFTT         FRICTION FORCE = AMU * FONTT (POUNDS)
C        FONTTM        MAX FORCE NORMAL TO CRUSH PLANE WITH AMU=0 (POUNDS)
C        FOFTTM        MAXIMUM FORCE PARALLEL TO CRUSH PLANE (POUNDS)
C        VELNE         VELOCITY CAPABILITY (EDGE) AT STROKE T WITH AMU=0
C                      (FT/SEC)
C        VELNF         VELOCITY CAPABILITY (FLAT) AT STROKE S WITH AMU=0
C                      (FT/SEC)
      DIMENSION ASAVE(100),BSAVE(100),CSAVE(100),DSAVE(100),ESAVE(100),
     1          FSAVE(100),GSAVE(100)
      DIMENSION RSAVE(100),SSAVE(100),TSAVE(100),USAVE(100),VSAVE(100),
     1          WSAVE(100),XSAVE(100)
 10   WRITE (6,580)
      READ (5,590) A,B,G,RHOP,C,D,RHOL,XKW,RDIA
      READ (5,590) VE,VF,F1,E1,F2,E2,F3,E3
      READ (5,590) RNORM,RSHR,STREF,RSHCR,AMU,POW,HND
      ASQAR=A*A
      BSQAR=B*B
      CSQAR=C*C
      DSQAR=D*D
      GSQAR=G*G
      DO 20 I=1,1400
 20   ASAVE=0.0
      SIGMA=XKW*RHOL/(STREF*144.)
      ROUNT=0.0
      BOUNT=-1.0
      XN=20.0
      XNE=20.0
C
C     PAYLOAD VOLUME AND WEIGHT
      VPA=6.28*GSQAR*B
      VPB=9.87*A*B*(G+0.424*A)
      VP=VPA+VPB
      WP=RHOP*VP/1728.0
      IF (HND.EQ.1.) GO TO 70
      IF (E2.GT.0.0) GO TO 70
      IND1=0
      WPK=.125
      IF (G.EQ.0.0) GO TO 40
      IF (XKW.LE.(VF**2/8.05)) GO TO 540
 30   SQFA=SQRT(WPK*WP/(RHOL*(64.4*WPK*XKW-VF**2)))
      T1E=B+RDIA+11.73*VF*SQFA/SQRT(G+D-C)
      T2E=B+11.73*VF*SQFA/(STREF*SQRT(G+D-C))
      TE=AMAX1(T1E,T2E)
      WPK=0.50-(B+RDIA)/(2.*TE)
      IND1=IND1+1
      IF (IND1.EQ.1) GO TO 30
      T1F=C+A+RDIA-D+22.5*VE*SQFA/SQRT(TE)
      T2F=C+A-D+22.5*VE*SQFA/(STREF*SQRT(TE))
      TF=AMAX1(T1F,T2F)
      E1=TE-0.20*(TE-B-RDIA)
      E2=TE
      E3=TE+0.20*(TE-B-RDIA)
      F1=TF-0.20*(TF-C-A-RDIA+D)
```

```
      F2=TF
      F3=TF+0.20*(TF-C-A-RDIA+D)
      GO TO 70
40    IF (XKW.LE.(VF**2/2.68)) GO TO 540
      TEC=WP*VF**2/(28.*RHOL*(2.68*XKW-VF**2))*1728.+B**3
      TE=TEC**.333
      IF (TE.LT.(1.2*(B+RDIA))) GO TO 60
50    E1=TE-.20*(TE-B-RDIA)
      F1=E1
      E2=TE
      F2=E2
      E3=TE+.20*(TE-B-RDIA)
      F3=E3
      GO TO 70
60    TE=1.25*(B+RDIA)
      GO TO 50
70    E=E1
      F=F1
      GO TO 100
80    E=E2
      F=F2
      GO TO 100
90    E=E3
      F=F3
100   FMAX=0.0
      EMAX=0.0
C
C     LIMITER VOLUME AND WEIGHT
      VA=4.93*E*F*(G-C+D+0.424*F)
      VB=4.93*DSQAR*(G-C+.576*D)
      VC=6.28*D*(E-D)*(G-C+0.5*D)
      VD=4.93*A*B*(G+0.424*A)
      VLE=6.28*B*C*(G-0.5*C)
      VL=(VA+VB+VC-VD-VLE)*2.0
      WL=RHOL*VL/1728.0
C
C     TOTAL WEIGHT
      WT=WP+WL
C
C     FLAT LANDING CASE
      IF (C.LT.(2.*D)) GO TO 110
      S1=0.
      GO TO 120
110   S1=D-SQRT((2.*D-C)*C)
120   S2=D
      SMAX1=STREF*(E-B)
      SMAX2=E-B-RDIA
      SMAX=AMIN1(SMAX1,SMAX2)
      SMAX=SMAX/1.118
      IF (ROUNT.GT.0.0) GO TO 130
      DS=SMAX/10.0-0.001
      GO TO 140
130   DS=SMAX/20.0-0.0005
      XN=50.0
140   FO=0.0
      FP=0.0
```

```
      DX=A/XN                                                           D3R1610
      COUNT=0.0                                                         D3R1620
      S=0.0                                                             D3R1630
      UNF=0.                                                            D3R1640
      UNFM=WT*VF**2/5.36                                                D3R1650
      FPNF=0.                                                           D3R1660
      SCR=SMAX                                                          D3R1670
      I=0                                                               D3R1680
      U=0.0                                                             D3R1690
  150 S=S+DS                                                            D3R1700
      IF (S-SMAX) 160,270,270                                           D3R1710
  160 FO1=0.0                                                           D3R1720
      FO2=0.0                                                           D3R1730
      W=0.0                                                             D3R1740
      X=0.0                                                             D3R1750
      X1MAX=D-C+F/E*SQRT(2.0*E*S-S**2)                                  D3R1760
  170 IF ((ASQAR-X*X).LE.0.0) GO TO 220                                 D3R1770
      Y=B/A*SQRT(ASQAR-X*X)                                             D3R1780
      PHI=ATAN(B*X/(A*SQRT(ASQAR-X*X)))                                 D3R1790
      X1=X+(E-S-Y)*BSQAR*X/(ASQAR*Y)                                    D3R1800
      IF (S-S1) 180,190,190                                             D3R1810
  180 X1MIN=D-C-SQRT((2.*D-S)*S)                                        D3R1820
      IF (X1.LT.X1MIN) GO TO 200                                        D3R1830
  190 DF=SIGMA*6.28*(G+X+(E-S-Y)*BSQAR*X/(ASQAR*Y))*(1.+(E-S-Y)*ASQAR*BSD3R1840
     1QAR*BSQAR/(Y*(ASQAR*ASQAR*Y*Y+BSQAR*BSQAR*X*X)))*DX*(1.+RNORM*TAN(D3R1850
     2PHI)**2)                                                          D3R1860
      FO1=FO1+DF                                                        D3R1870
      IF (X1-X1MAX) 200,220,210                                         D3R1880
  200 X=X+DX                                                            D3R1890
      GO TO 170                                                         D3R1900
  210 DNF=6.28*(G+(X1MAX+X1)/2.0)*(X1-X1MAX)*SIGMA*(COS(PHI)**2+SIN(PHI)D3R1910
     1**2*RNORM)                                                        D3R1920
      FO1=FO1-DNF                                                       D3R1930
  220 IF (S.LT.S1) GO TO 250                                            D3R1940
      IF (S.GE.S2) GO TO 230                                            D3R1950
      W=C-D+SQRT((2.*D-S)*S)                                            D3R1960
      GO TO 240                                                         D3R1970
  230 W=C                                                               D3R1980
  240 FO2=SIGMA*6.28*(G-0.5*W)*W                                        D3R1990
  250 FO=FO1+FO2                                                        D3R2000
      AREA=(X1MAX-X1MIN)*6.28*(G+(X1MIN+X1MAX)/2.)+6.28*(G-.5*W)*W      D3R2010
      FOFTTM=AREA*RSHCR*SIGMA                                           D3R2020
      FONTTM=FC                                                         D3R2030
      RFONTT=(FO*FOFTTM)**POW/(FOFTTM**POW+(FO*AMU)**POW)               D3R2040
      FONTT=RFONTT**(1./POW)                                            D3R2050
      FOFTT=AMU*FONTT                                                   D3R2060
      FO=FONTT                                                          D3R2070
      I=I+1                                                             D3R2080
      ASAVE(I)=S                                                        D3R2090
      BSAVE(I)=AREA                                                     D3R2100
      CSAVE(I)=FONTT                                                    D3R2110
      DSAVE(I)=FOFTT                                                    D3R2120
      ESAVE(I)=FONTTM                                                   D3R2130
      FSAVE(I)=FOFTTM                                                   D3R2140
      IF (COUNT.GT.0.0) GO TO 260                                       D3R2150
      IF (FP.LE.FO) GO TO 260                                           D3R2160
```

```
      COUNT=1.0                                                         D3R2170
      FMAX=FP                                                           D3R2180
      SCR=S-DS                                                          D3R2190
260   DU=(FO+FP)*0.5*DS                                                 D3R2200
      U=U+DU                                                            D3R2210
      FP=FO                                                             D3R2220
      IF (UNF.GE.UNFM) GO TO 150                                        D3R2230
      DUNF=(FONTTM+FPNF)*.5*DS                                          D3R2240
      UNF=UNF+DUNF                                                      D3R2250
      FPNF=FONTTM                                                       D3R2260
      FFLODF=FONTTM/WT                                                  D3R2270
      VELNF=SQRT(5.36*UNF/WT)                                           D3R2280
      GSAVE(I)=VELNF                                                    D3R2290
      GO TO 150                                                         D3R2300
270   DU=FO*(DS-S+SMAX)                                                 D3R2310
      U=U+DU                                                            D3R2320
      S=S-DS                                                            D3R2330
      VELF=SQRT(5.36*U/WT)                                              D3R2340
      FIFO=FO                                                           D3R2350
      FMAX=AMAX1(FIFO,FMAX)                                             D3R2360
      FLOADF=FMAX/WT                                                    D3R2370
      FNCF=RNORM*SIGMA*3.14*((F+D-C+G)**2-(G+A)**2)                     D3R2380
      FINCR=3.14*SIGMA*(2.0*A*G+A**2+2.0*G*C-C**2)+1.57*RSHR*SIGMA*(4.0*D3R2390
     1B*G+3.14*A*B)+FNCF                                                D3R2400
      FCR=FLOADF*(WP+0.5*WL)                                            D3R2410
C                                                                       D3R2420
C     EDGE LANDING CASE                                                 D3R2430
      TMAX1=STREF*(F+D-C-A)                                             D3R2440
      TMAX2=F+D-C-A-RDIA                                                D3R2450
      TMAX=AMIN1(TMAX1,TMAX2)                                           D3R2460
      TMAX=TMAX/1.118                                                   D3R2461
      DT=TMAX/10.0-0.001                                                D3R2470
      IF (ROUNT.GT..0) DT=DT/2.                                         D3R2480
      XMT=TMAX                                                          D3R2490
      T=0.0                                                             D3R2500
      J=0                                                               D3R2510
      FP=0.0                                                            D3R2520
      U=0.0                                                             D3R2530
      UNE=0.                                                            D3R2540
      FPNE=0.                                                           D3R2550
      UNEM=WT*VE**2/5.36                                                D3R2560
      EOUNT=0.0                                                         D3R2570
      H=G+D+F-C                                                         D3R2580
      DTHE=0.005                                                        D3R2590
      DY=B/XNE                                                          D3R2600
280   T=T+DT                                                            D3R2610
      IF (T-TMAX) 290,380,380                                           D3R2620
290   THEM=ACOS(1.0-T/H)                                                D3R2630
      THE=0.0                                                           D3R2640
      FO=0.0                                                            D3R2650
      AREA=0.0                                                          D3R2660
300   IF ((H-(H-T)/COS(THE)).GE.F) GO TO 320                            D3R2670
310   QY=F**2-(F-H+(H-T)/COS(THE))**2                                   D3R2680
      IF (QY.LE.0.0) GO TO 360                                          D3R2690
      Y1MAX=E/F*SQRT(QY)                                                D3R2700
      GO TO 340                                                         D3R2710
```

```
320   XH=(H-T)/COS(THE)                                                    D3R2720
      IF (XH-H+F) 330,310,310                                              D3R2730
330   QY=D**2-(H-F-XH)**2                                                  D3R2740
      IF (QY.LE.0.0) GO TO 360                                             D3R2750
      Y1MAX=E-D+SQRT(QY)                                                   D3R2760
340   Y=0.0                                                                D3R2770
      DAREA=4.*Y1MAX*DTHE*(H-T)/COS(THE)**2                                D3R2780
      AREA=AREA+DAREA                                                      D3R2790
350   Y=Y+DY                                                               D3R2800
      IF ((B**2-Y**2).LE.0.0) GO TO 360                                    D3R2810
      X=A/B*SQRT(B**2-Y**2)                                                D3R2820
      Y1=((H-T)/COS(THE)-G-X)*A**2*Y/(B**2*X)+Y                            D3R2830
      IF (Y1.GE.Y1MAX) GO TO 360                                           D3R2840
      CF3I=B*X/(A*SQRT(ASQAR-X*X))                                         D3R2850
      PHI=ATAN(CF3I)                                                       D3R2860
      CPHI=1.5708-PHI                                                      D3R2870
      DF=SIGMA*(H-T)*DTHE*DY*(1.0+((H-T)/COS(THE)-G-X)*B**2*A**4/(X*(B**D3R2880
     14*X**2+A**4*Y**2)))*(1.0+RNORM*(TAN(CPHI)**2+TAN(THE)**2/COS(CPHI)D3R2890
     2))                                                                   D3R2900
      FO=FO+DF                                                             D3R2910
      DNFP=SIGMA*(Y1MAX-Y1)*(H-T)*DTHE*(COS(CPHI)**2+RNORM*(SIN(CPHI)**2D3R2920
     1+COS(CPHI)*TAN(THE)**2))                                             D3R2930
      GO TO 350                                                            D3R2940
360   THE=THE+DTHE                                                         D3R2950
      IF (THE.LT.THEM) GO TO 300                                           D3R2960
      FC=4.0*(FO+DNFP)                                                     D3R2970
      FOFTTM=AREA*RSHCR*SIGMA                                              D3R2980
      FONTTM=FO                                                            D3R2990
      RFONTT=(FO*FOFTTM)**POW/(FOFTTM**POW+(FO*AMU)**POW)                  D3R3000
      FONTT=RFONTT**(1./POW)                                               D3R3010
      FOFTT=AMU*FONTT                                                      D3R3020
      FO=FONTT                                                             D3R3030
      J=J+1                                                                D3R3040
      RSAVE(J)=T                                                           D3R3050
      SSAVE(J)=AREA                                                        D3R3060
      TSAVE(J)=FONTT                                                       D3R3070
      USAVE(J)=FOFTT                                                       D3R3080
      VSAVE(J)=FONTTM                                                      D3R3090
      WSAVE(J)=FOFTTM                                                      D3R3100
      IF (EOUNT.GT.0.0) GO TO 370                                          D3R3110
      IF (FP.LE.FO) GO TO 370                                              D3R3120
      EOUNT=1.0                                                            D3R3130
      EMAX=FP                                                              D3R3140
      XMT=T-DT                                                             D3R3150
370   DU=(FO+FP)*0.5*DT                                                    D3R3160
      U=U+DU                                                               D3R3170
      FP=FO                                                                D3R3180
      IF (UNE.GE.UNEM) GO TO 280                                           D3R3190
      DUNE=(FONTTM+FPNE)*.5*DT                                             D3R3200
      UNE=UNE+DUNE                                                         D3R3210
      FPNE=FONTTM                                                          D3R3220
      EELODF=FONTTM/WT                                                     D3R3230
      VELNE=SQRT(5.36*UNE/WT)                                              D3R3240
      XSAVE(J)=VELNE                                                       D3R3250
      GO TO 280                                                            D3R3260
380   DU=FO*(DT-T+TMAX)                                                    D3R3270
```

```
      U=U+DU
      T=T-DT
      VELE=SQRT(5.36*U/WT)
      FIQ=FO
      EMAX=AMAX1(FIQ,EMAX)
      ELOADF=EMAX/WT
      ENCF=RNORM*SIGMA*3.14*(E*F+D**2+1.273*D*(E-D)-1.273*D*C-A*B)
      EINCR=SIGMA*B*(4.0*G+3.14*A)+4.0*RSHR*SIGMA*C*(G+C/2.0)+1.57*RSHR*
     1SIGMA*(3.0*B**2+A**2)*(G+0.424*A)/B+ENCF
      ECR=ELOADF*(WP+0.5*WL)
      IF (HND.EQ.1.) GO TO 550
      IF (ROUNT.GT.0.0) GO TO 530
      IF (BOUNT) 390,400,410
 390  BOUNT=0.0
      V1E=VELE
      V1F=VELF
      WL1=WL
      GO TO 80
 400  BOUNT=1.0
      V2E=VELE
      V2F=VELF
      WL2=WL
      GO TO 90
 410  ROUNT=1.0
      V3E=VELE
      V3F=VELF
      WL3=WL
      DAKE=V1E*(F2-F3)-V2E*(F1-F3)+V3E*(F1-F2)
      IF (DAKE.EQ.0.0) GO TO 450
      AKE=(V1E*(F2*F2-F3*F3)-V2E*(F1*F1-F3*F3)+V3E*(F1*F1-F2*F2))/(2.*DA
     1KE)
      AME=((F1-AKE)**2-(F2-AKE)**2)/(V1E-V2E)
      AHE=V1E-(F1-AKE)**2/AME
      AAME=AME*(VE-AHE)
      IF (AAME.LT.0.0) GO TO 420
      IF (AME) 440,450,430
 420  DAKE=F1*(V2E-V3E)-F2*(V1E-V3E)+F3*(V1E-V2E)
      IF (DAKE.EQ.0.0) GO TO 450
      AKE=(F1*(V2E**2-V3E**2)-F2*(V1E**2-V3E**2)+F3*(V1E**2-V2E**2))/(2.
     1*DAKE)
      AME=((V1E-AKE)**2-(V2E-AKE)**2)/(F1-F2)
      AHE=F1-(V1E-AKE)**2/AME
      F4=AHE+(VE-AKE)**2/AME
      GO TO 460
 430  F4=AKE+SQRT(AAME)
      GO TO 460
 440  F4=AKE-SQRT(AAME)
      GO TO 460
 450  F4=F1+(F1-F3)*(VE-V1E)/(V1E-V3E)
 460  F=F4
      DAKF=V1F*(E2-E3)-V2F*(E1-E3)+V3F*(E1-E2)
      IF (DAKF) 470,510,470
 470  AKF=(V1F*(E2**2-E3**2)-V2F*(E1**2-E3**2)+V3F*(E1**2-E2**2))/(2*DAK
     1F)
      AMF=((E1-AKF)**2-(E2-AKF)**2)/(V1F-V2F)
      AHF=V1F-(E1-AKF)**2/AMF
```

```
      AAMF=AMF*(VF-AHF)
      IF (AAMF.LT.0.0) GO TO 480
      IF (AMF) 500,510,490
480   DAKF=E1*(V2F-V3F)-E2*(V1F-V3F)+E3*(V1F-V2F)
      IF (DAKF.EQ.0.0) GO TO 510
      AKF=(E1*(V2F**2-V3F**2)-E2*(V1F**2-V3F**2)+E3*(V1F**2-V2F**2))/(2.
     1*DAKF)
      AMF=((V1F-AKF)**2-(V2F-AKF)**2)/(E1-E2)
      AHF=E1-(V1F-AKF)**2/AMF
      E4=AHF+(VF-AKF)**2/AMF
      GO TO 520
490   E4=AKF+SQRT(AAMF)
      GO TO 520
500   E4=AKF-SQRT(AAMF)
      GO TO 520
510   E4=E1+(E1-E3)*(VF-V1F)/(V1F-V3F)
520   E=E4
      GO TO 100
530   IF (ROUNT.GT.1.0) GO TO 550
      V4E=VELE
      V4F=VELF
      DDE=(VF**2-VELF**2)*WT/(5.36*FIFO)
      E=E+DDE
      DDF=(VE**2-VELE**2)*WT/(5.36*FIQ)
      F=F+DDF
      ROUNT=2.0
      GO TO 100
540   WRITE (6,760)
550   CONTINUE
      WRITE (6,600)
      WRITE (6,610) A,B,C,D,G,RHOP,RHOL
      WRITE (6,620) RDIA,VE,VF,RNORM,RSHR,STREF
      WRITE (6,630) RSHCR,AMU,POW,XKW
      WRITE (6,640) E1,F1,E2,F2,E3,F3
      WRITE (6,650)
      WRITE (6,660) E4,F4,E,F,WP,WL,WT
      WRITE (6,670) FLOADF,ELOADF,FCR,FINCR,ECR,EINCR
      WRITE (6,680) FFLODF,EELODF
      WRITE (6,690) VELE,VELF,SCR,XMT,FIQ,FIFO
      WRITE (6,700) V1E,V1F,V2E,V2F,V3E,V3F,V4E,V4F
      WRITE (6,710) S,T,WL1,WL2,WL3
      WRITE (6,720)
      DO 560 L=1,I
560   WRITE (6,730) ASAVE(L),BSAVE(L),CSAVE(L),DSAVE(L),ESAVE(L),FSAVE(L
     1),GSAVE(L)
      WRITE (6,740)
      DO 570 N=1,J
570   WRITE (6,750) RSAVE(N),SSAVE(N),TSAVE(N),USAVE(N),VSAVE(N),WSAVE(N
     1),XSAVE(N)
      GO TO 10
C
580   FORMAT (1H1,4X,22HCRUSHABLE TORUS LANDER////)
590   FORMAT (9E8.1)
600   FORMAT (20X,16HINPUT PARAMETERS//)
610   FORMAT (1X,2HA=F6.2,3X,2HB=F6.2,3X,2HC=F6.2,3X,2HD=F6.2,3X,2HG=F6.
     12,3X,5HRHOP=F5.1,3X,5HRHOL=F5.1/)
```

APPENDIX A

```
620   FORMAT (1X,5HRDIA=F6.2,8X,3HVE=F6.1,3X,3HVF=F6.1,8X,6HRNORM=F6.3,3
     1X,5HRSHR=F6.3,3X,6HSTREF=F6.3/)
630   FORMAT (1X,6HRSHCR=F6.3,3X,4HAMU=F6.3,3X,4HPOW=F6.2,3X,4HXKW=F8.1/
     1)
640   FORMAT (1X,3HE1=F6.2,3X,3HF1=F6.2,8X,3HE2=F6.2,3X,3HF2=F6.2,8X,3HE
     13=F6.2,3X,3HF3=F6.2///)
650   FORMAT (20X,11HOUTPUT DATA//)
660   FORMAT (1X,3HE4=F6.2,3X,3HF4=F6.2,8X,2HE=F6.2,3X,2HF=F6.2,8X,3HWP=
     1F6.1,3X,3HWL=F6.1,3X,3HWT=F6.1/)
670   FORMAT (1X,7HFLOADF=F8.1,3X,7HELOADF=F8.1,8X,4HFCR=F12.1,3X,6HFINC
     1R=F12.1,8X,4HECR=F12.1,3X,6HEINCR=F12.1/)
680   FORMAT (1X,7HFFLODF=F9.1,3X,7HEELODF=F9.1/)
690   FORMAT (1X,5HVELE=F6.1,3X,5HVELF=F6.1,8X,4HSCR=F6.2,3X,4HXMT=F6.2,
     18X,4HFIQ=F9.1,3X,5HFIFO=F9.1/)
700   FORMAT (1X,4HV1E=F6.1,3X,4HV1F=F6.1,8X,4HV2E=F6.1,3X,4HV2F=F6.1,8X
     1,4HV3E=F6.1,3X,4HV3F=F6.1,8X,4HV4E=F6.1,3X,4HV4F=F6.1/)
710   FORMAT (1X,2HS=F6.2,3X,2HT=F6.2,8X,4HWL1=F6.1,3X,4HWL2=F6.1,3X,4HW
     1L3=F6.1////)
720   FORMAT (20X,28HLOAD STROKE FOR FLAT LANDING//)
730   FORMAT (1X,2HS=F6.3,5X,5HAREA=F9.2,5X,6HFONTT=F10.1,5X,6HFOFTT=F10
     1.1,5X,7HFONTTM=F10.1,5X,7HFOFTTM=F10.1,5X,6HVELNF=F6.1)
740   FORMAT (1H1,20X,28HLOAD STROKE FOR EDGE LANDING//)
750   FORMAT (1X,2HT=F6.3,5X,5HAREA=F9.2,5X,6HFONTT=F10.1,5X,6HFOFTT=F10
     1.1,5X,7HFONTTM=F10.1,5X,7HFOFTTM=F10.1,5X,6HVELNE=F6.1)
760   FORMAT (1H1,20X,43HVF IS TOO GREAT FOR ATTENUATOR XKW SELECTED///)
      END

  1.29      5.16     20.1      60.       1.5       4.5      2.1     2270.     5.0
  85.       85.      0.0       0.0
  1.0       0.4       0.7      1.0       0.3       2.0      0.0
```

A.4.2 OMNIDIRECTIONAL LOADS PROGRAM LISTING

```
      PROGRAM DORR (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION ASAVE(100),BSAVE(100),CSAVE(100),DSAVE(100),ESAVE(100),
     1FSAVE(100),GSAVE(100),HSAVE(100),SSAVE(100),TSAVE(100)
   10 WRITE (6,460)
      READ (5,470) A,B,C,D,E,F,G,RHOL,XKW
      READ (5,470) BET,RDIA,DX,DTHE,RNORM,CN,WT
      READ (5,470) RSHCR,STREF,AMU,POW
C
C     PROGRAM DETERMINES LOAD VS STROKE AND VELOCITY CAPABILITY FOR ANY
C     CRUSH ATTITUDE (CONSTANT THROUGHOUT STROKE) FOR A GIVEN CRUSHABLE
C     TORUS
C     INPUT PARAMETERS
C        A,B,C,D,E,F,G        CRUSHABLE TORUS DIMENSIONS (INCHES)
C        XKW                  ATTENUATOR SPECIFIC ENERGY (FT-LBS/LB)
C        RHOL                 ATTENUATOR DENSITY (LBS/CUBICFOOT)
C        BET                  ANGLE BETWEEN CRUSH PLANE AND X AXIS (RADIANS)
C        RDIA                 ROCK DIAMETER (INCHES)
C        DX                   INCREMENTAL X WHICH BUILD UP TO DIM.A (INCHES)
C        DTHE                 INCREMENTAL ANGLE THE (PLAN VIEW) IN RADIANS
C        RNORM                RATIO OF NORMAL TO RADIAL CRUSH STRESS
C        CN                   NUMBER OF STEPS DESIRED IN INCREMENTING STROKE
C        WT                   TOTAL WEIGHT OF PAYLOAD AND LANDING SYSTEM
C                             IN POUNDS
C        RSHCR                RATIO OF MAX. ALLOW SHEAR TO RADIAL CRUSH
C                             STRESS
C        STREF                STROKE EFFICIENCY - ALLOW. STROKE/ORIGINAL
C                             HEIGHT
C        AMU                  COEFFICIENT OF FRICTION
C        POW                  EXPONENT USED IN INTERACTION OF SHEAR AND
C                             AXIAL STRESS
C     OUTPUT DATA
C        U                    TOTAL ENERGY ABSORBED IN CRUSHING (INCH-POUNDS)
C        VELBT                VELOCITY CAPABILITY AT ANGLE BET (FT/SEC)
C        ALCL                 ALLOWABLE CLEARANCE BETWEEN PAYLOAD AND CRUSH
C                             PLANE IN INCHES ALLOWING 11.8( STROKE MARGIN
C        CLEAR                CLEARANCE BETWEEN PAYLOAD AND CRUSH PLANE (IN)
C        STTM                 STROKE (INCHES)
C        FONTT                CRUSH FORCE NORMAL TO CRUSH PLANE ACTING WITH F
C                             FOFTT IN POUNDS
C        FOFTT                FRICTION FORCE = AMU*FONTT  IN POUNDS
C        TORTT                TOTAL TORQUE DUE TO FONTT AND FOFTT ABOUT
C                             TORUS CENTER (INCH-POUNDS)
C        ARTSU                FOOTPRINT AREA (SQUARE INCHES)
C        ARMOM                ARTSU * Y DISTANCE TO CENTROID (CUBIC INCHES)
C        FONTTM               CRUSH FORCE NORMAL TO CRUSH PLANE (MAX WITH
C                             AMU=0) IN POUNDS
C        TORTNM               TORQUE DUE TO FONTTM ABOUT TORUS CENTER (INCH-
C                             POUNDS)
C        FOFTTM               MAX. ALLOW. FRICTION FORCE WITH NORMAL FORCE
C                             = 0  (POUNDS)
C        TORTFM               TORQUE DUE TO FOFTTM ABOUT TORUS CENTER (INCH-
C                             POUNDS)
      DO 20 I=1,1000
   20 ASAVE(I)=0.0
      I=0.0
      FPNTT=0.0
```

APPENDIX A

```
      U=0.0                                                             OMI 460
      VELBT=0.0                                                         OMI 470
      STTM=0.0                                                          OMI 480
      SPTM=0.0                                                          OMI 490
      FONTTM=0.0                                                        OMI 500
      FOFTTM=0.0                                                        OMI 510
30    FONTT=0.0                                                         OMI 520
      FOFTT=0.0                                                         OMI 530
      TORTN=0.0                                                         OMI 540
      DAREA=0.0                                                         OMI 550
      ARTSU=0.0                                                         OMI 560
      ARMOM=0.0                                                         OMI 570
      ALCL=0.0                                                          OMI 580
      SIGMA=XKW*RHOL/(144.*STREF)                                       OMI 590
      CF1=SQRT(E**2+F**2*(TAN(BET)**2))                                 OMI 600
      YOI=CF1-STTM/COS(BET)+(D-C)*TAN(BET)                              OMI 610
C                                                                       OMI 620
C     CRUSH PLANE/PAYLOAD CLEARANCE AND ALLOWABLE CLEARANCE LIMIT       OMI 630
      IF (BET-0.01) 40,40,50                                            OMI 640
40    CLFAR=E-D-STTM                                                    OMI 650
      GO TO 70                                                          OMI 660
50    IF (BET.LT.1.56) GO TO 60                                         OMI 670
      IF (BET.GT.1.58) GO TO 60                                         OMI 680
      CLEAR=F+D-C-A-STTM                                                OMI 690
      GO TO 70                                                          OMI 700
60    CLEAR=(YOI-SQRT(G**2+A**2*TAN(BET)**2))/(SIN(BET)*(1.0+TAN(BET)**2OMI 710
     1))*TAN(BET)                                                       OMI 720
70    ALCL1=(1.0-STREF)*(STTM+CLEAR)                                    OMI 730
      ALCL2=RDIA                                                        OMI 740
      ALCL=AMAX1(ALCL1,ALCL2)                                           OMI 750
      ALCL=0.894*ALCL+0.106*CLEAR                                       OMI 751
      STINC=(STTM+CLEAR-ALCL-0.005)/CN                                  OMI 760
      IF (ALCL-CLEAR) 80,80,440                                         OMI 770
80    THE=0.0                                                           OMI 780
C                                                                       OMI 790
C     INCREMENT THE, CALCULATE PSI,STTH                                 OMI 800
90    THE=THE+DTHE                                                      OMI 810
      X=0.0                                                             OMI 820
      X1=0.0                                                            OMI 830
      MOUNT=1                                                           OMI 840
      PPSI=TAN(BET)*COS(THE)                                            OMI 850
      PSI=ATAN(PPSI)                                                    OMI 860
      IF (PSI.LT.1.46) GO TO 100                                        OMI 870
      IF (PSI.GT.1.58) GO TO 100                                        OMI 880
      FONT=0.0                                                          OMI 890
      GO TO 320                                                         OMI 900
100   STTH=COS(PSI)*STTM/COS(BET)+COS(PSI)*SQRT(E**2+F**2*(TAN(PSI)**2))OMI 910
     1-COS(PSI)*SQRT(E**2+F**2*(TAN(BET)**2))+(G+D-C)*COS(PSI)*TAN(BET)*OMI 920
     2(COS(THE)-1.0)                                                    OMI 930
C                                                                       OMI 940
C     INTEGRATION LIMITS (STTH=0 OR THE=180.DEGREES)                    OMI 950
      IF (STTH) 400,400,110                                             OMI 960
110   IF (THE-3.14) 120,120,400                                         OMI 970
120   FO1=0.0                                                           OMI 980
      FO2=0.0                                                           OMI 990
      FONT=0.0                                                          OMI1000
```

```
      TORKN=0.0
      TOR2N=0.0
      ARSU=0.0
      ARMOO=0.0
      ARS2=0.0
      ARM2=0.0
      CF1=SQRT(E**2+F**2*(TAN(PSI)**2))
      YOI=CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)
      YCI=YOI+C*TAN(PSI)
      EIC=2.0*E**2*(D-C)/F**2+2.0*YOI*TAN(PSI)
      EIR=YOI**2-E**2+E**2*(D-C)**2/F**2
      EID=CF1**2/F**2
      IF ((EIC**2-4.0*EID*EIR).LT.0.0) GO TO 420
      XOL=(EIC+SQRT(EIC**2-4.0*EID*EIR))/(2.0*EID)
      YOL=YOI-XOL*TAN(PSI)
      IF (YCI-E+D) 130,130,140
C
C     CALCULATE FORCE ON ELEMENTS INBOARD OF G
130   FO2=SIGMA*C*DTHE*(G-C/2.0)*(COS(PSI)+RNORM*TAN(PSI)*SIN(PSI))
      TOR2N=FO2*(G-C/2.0)*COS(PSI)-FO2*SIN(PSI)*(YOI+YCI)/2.0
      ARS2=C*(G-C/2.0)*DTHE/COS(BET)
      ARM2=ARS2*(YCI+YOI)/2.0
      GO TO 200
140   YQI=YOI-(D-C)*TAN(PSI)
      IF (YQI-E) 150,150,280
150   SXI=D**2*(1.0+(TAN(PSI))**2)-(E-D-YOI+(D-C)*TAN(PSI))**2
      IF (SXI.LT.0.0) GO TO 420
      XIIC=((E-D-YOI+(D-C)*TAN(PSI))*TAN(PSI)+SQRT(SXI))/(1.0+TAN(PSI)**
     12)-D+C
      YOIC=YOI+XIIC*TAN(PSI)
      IF (YOI-YOIC) 160,160,170
160   FO2=SIGMA*XIIC*DTHE*(G-XIIC/2.0)*(COS(PSI)+RNORM*TAN(PSI)*SIN(PSI)
     1)
      TOR2N=FO2*(G-XIIC/2.0)*COS(PSI)-FO2*SIN(PSI)*(YOI+XIIC*TAN(PSI))
      ARS2=XIIC*(G-XIIC/2.0)*DTHE/COS(BET)
      ARM2=ARS2*(YOIC+YOI)/2.0
      GO TO 200
170   IF (XIIC+D-C) 280,200,180
C
C     INCREMENT X UNTIL CRUSH PLANE/ATTENVATOR SUFACE IS REACHED
C     )INBOARD*
180   X=X+DX
      IF (X.GE.A) GO TO 430
      CF2=(B/A-A/B)*SQRT(A**2-X**2)
      CF3=A*SQRT(A**2-X**2)/(B*X)
      X1=(CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)-CF2)/(CF3+TAN(PSI))
      Y1=CF2+CF3*X1
      IF (X1+XIIC) 180,200,190
190   CF3=A*SQRT(A**2-X**2)/(B*X)
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      FO2=SIGMA*(X1+XIIC)*(G+(X1-XIIC)/2.0)*DTHE*COS(PHI-PSI)*(COS(PHI-P
     1SI)/COS(PSI)+SIN(PHI-PSI)*TAN(PHI-PSI)*RNORM/COS(PSI))
      TOR2N=FO2*(G+(X1-XIIC)/2.0)*COS(PSI)-FO2*SIN(PSI)*(YOIC+Y1)/2.0
      ARS2=(G+(X1-XIIC)/2.0)*(X1+XIIC)*DTHE/COS(BET)
      ARM2=ARS2*(YOIC+Y1)/2.0
```

```
C
C     INCREMENT X AND SUM FORCE UNTIL OUTBOARD INTERSECTION IS REACHED
200   X=X+DX
      IF (X.GE.A) GO TO 430
      CF2=(B/A-A/B)*SQRT(A**2-X**2)
      CF3=A*SQRT(A**2-X**2)/(B*X)
      X1=(CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)-CF2)/(CF3+TAN(PSI))
      Y1=CF2+CF3*X1
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
210   RAD1=(X1-X)/SIN(PHI)
      RAD2=G+X1
      DPHI=(A**3*B*DX)/((A**2*(A**2-X**2)+B**2*X**2)*SQRT(A**2-X**2))
      DA=RAD2*(DX/COS(PHI)+RAD1*DPHI)*DTHE
      IF ( DA .LT. ABS(Y1*RAD2*DTHE)) GO TO 220
      DA=0.0
220   DFN=SIGMA*DA*(COS(PHI-PSI)+RNORM*TAN(PHI-PSI)*SIN(PHI-PSI))
      FO1=FO1+DFN
      DAS=2.0*DA/COS(PHI-PSI)*COS(PSI)/COS(BET)
      ARSU=ARSU+DAS
      ARMO=DAS*Y1
      ARMOO=ARMOO+ARMO
      DTN=DFN*RAD2*COS(PSI)-DFN*SIN(PSI)*Y1
      TORKN=TORKN+DTN
      IF (MOUNT.EQ.2) GO TO 250
      IF (YOL.GT.0.0) GO TO 240
      IF ((X+1.01*DX).LT.A) GO TO 200
230   X=A
      X1=F+D-C
      PHI=1.57
      DA=RAD2*Y1*DTHE
      DFN=SIGMA*DA*(COS(PHI-PSI)+RNORM*TAN(PHI-PSI)*SIN(PHI-PSI))
      FO1=FO1+DFN
      DAS=2.0*DA/SIN(PSI)*COS(PSI)/COS(BET)
      ARSU=ARSU+DAS
      ARMO=DAS*Y1/2.0
      ARMOO=ARMOO+ARMO
      DTN=DFN*RAD2*COS(PSI)-DFN*SIN(PSI)*Y1
      TORKN=TORKN+DTN
      IF (YOL.GE.0.0) GO TO 270
C
C     INTEGRATION BEYOND PHI=90DEGREES (YOL IS NEGATIVE)
      X=A-DX
      CF2=(A/B-B/A)*SQRT(A**2-X**2)
      CF3=-A*SQRT(A**2-X**2)/(B*X)
      X1=(CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)-CF2)/(CF3+TAN(PSI))
      Y1=CF2+CF3*X1
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      PHI=ABS(PHI)
      RAD1=(X1-X)/SIN(PHI)
      RAD2=G+X1
      DA=-RAD2*Y1*DTHE
      DFN=SIGMA*DA*(COS(PHI-PSI)+RNORM*TAN(PHI-PSI)*SIN(PHI-PSI))
      FO1=FO1+DFN
      DAS=2.0*DA/SIN(PSI)*COS(PSI)/COS(BET)
```

```
      ARSU=ARSU+DAS                                                     OMI2120
      ARMO=DAS*Y1                                                       OMI2130
      ARMOO=ARMOO+ARMO                                                  OMI2140
      DTN=DFN*RAD2*COS(PSI)-DFN*SIN(PSI)*Y1                             OMI2150
      TORKN=TORKN+DTN                                                   OMI2160
      MOUNT=2                                                           OMI2170
      IF (XOL-X1) 270,270,260                                           OMI2180
240   IF ((X+1.01*DX).GE.A) GO TO 230                                   OMI2190
      IF (XOL-X1) 270,270,200                                           OMI2200
250   IF (XOL-X1) 270,270,260                                           OMI2210
260   X=X-DX                                                            OMI2220
      IF (X.GE.A) GO TO 430                                             OMI2230
      CF2=(A/B-B/A)*SQRT(A**2-X**2)                                     OMI2240
      CF3=-A*SQRT(A**2-X**2)/(B*X)                                      OMI2250
      X1=(CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)-CF2)/(CF3+TAN(PSI))          OMI2260
      Y1=CF2+CF3*X1                                                     OMI2270
      CF3I=1.0/CF3                                                      OMI2280
      PHI=ATAN(CF3I)                                                    OMI2290
      PHI=ABS(PHI)                                                      OMI2300
      GO TO 210                                                         OMI2310
C                                                                       OMI2320
C     ADDITION OF FORCES ON DTHE STRIPS                                 OMI2330
270   DNFN=SIGMA*(X1-XOL)*(G+(X1+XOL)/2.0)*DTHE*COS(PHI-PSI)*(COS(PHI-PSOMI2340
     1I)/COS(PSI)+SIN(PHI-PSI)*TAN(PHI-PSI)*RNORM/COS(PSI))             OMI2350
      ARNSU=(X1-XOL)*RAD2*DTHE*2.0/COS(BET)                             OMI2360
      ARNOO=ARNSU*(YOL+Y1)/2.0                                          OMI2370
      ARMOO=ARMOO-ARNOO+2.0*ARM2                                        OMI2380
      ARSU=ARSU-ARNSU+2.0*ARS2                                          OMI2390
      FO1=FO1-DNFN                                                      OMI2400
      FONT=2.0*(FO1+FO2)*COS(BET)/COS(PSI)                              OMI2410
      TORNN=DNFN*RAD2*COS(PSI)-DNFN*Y1*SIN(PSI)                         OMI2420
      TORKN=TORKN-TORNN                                                 OMI2430
      TORKN=2.0*(TORKN+TOR2N)*COS(THE)                                  OMI2440
      TORTN=TORTN+TORKN                                                 OMI2450
      ARTSU=ARTSU+ARSU                                                  OMI2460
      ARMOM=ARMOM+ARMOO                                                 OMI2470
      FONTT=FONTT+FONT                                                  OMI2480
      ARM=(G+D-C)*SIN(BET)+COS(BET)*SQRT(E**2+F**2*TAN(BET)**2)-STTH    OMI2490
      GO TO 90                                                          OMI2500
C                                                                       OMI2510
C     INBOARD INTERSECTION IS ON ATTENUATOR ELLIPSE                     OMI2520
280   EIC=2.0*E**2*(D-C)/F**2+2.0*YOI*TAN(PSI)                          OMI2530
      EIR=YOI**2-E**2+E**2*(D-C)**2/F**2                                OMI2540
      EID=CF1**2/F**2                                                   OMI2550
      IF ((EIC**2-4.0*EID*EIR).LT.0.0) GO TO 420                        OMI2560
      XIEC=(EIC-SQRT(EIC**2-4.0*EID*EIR))/(2.0*EID)                     OMI2570
290   IF (X1-XIEC) 310,200,300                                          OMI2580
300   XIIC=-XIEC                                                        OMI2590
      YIEC=YOI-XIEC*TAN(PSI)                                            OMI2600
      YOIC=YIEC                                                         OMI2610
      GO TO 190                                                         OMI2620
310   X=X+DX                                                            OMI2630
      IF (X.GE.A) GO TO 430                                             OMI2640
      CF2=(B/A-A/B)*SQRT(A**2-X**2)                                     OMI2650
      CF3=A*SQRT(A**2-X**2)/(B*X)                                       OMI2660
      X1=(CF1-STTH/COS(PSI)+(D-C)*TAN(PSI)-CF2)/(CF3+TAN(PSI))          OMI2670
```

```
      Y1=CF2+CF3*X1
      GO TO 290
C
C     EDGE CASE (BET=90.DEGREES)
320   H=G+D+F-C
      DY=DX
      STTH=H-(H-STTM)/COS(THE)
      IF (STTH-F) 330,340,340
330   QY=2.*F*STTH-STTH**2
      IF (QY.LE.0.0) GO TO 390
      Y1MAX=E/F*SQRT(QY)
      GO TO 360
340   XH=(H-STTM)/COS(THE)
      IF (XH-H+F) 350,330,330
350   QY=D**2-(H-F-XH)**2
      IF (QY.LE.0.0) GO TO 390
      Y1MAX=E-C+SQRT(QY)
360   Y=0.0
      DAREA=4.*Y1MAX*DTHE*(H-STTM)/(COS(THE))**2
      ARTSU=ARTSU+DAREA
370   Y=Y+DY
      X=A/B*SQRT(B**2-Y**2)
      Y1=((H-STTM)/COS(THE)-G-X)*A**2*Y/(B**2*X)+Y
      IF (Y1-Y1MAX) 380,390,390
380   CF3I=B*X/(A*SQRT(A**2-X**2))
      PHI=ATAN(CF3I)
      CPHI=1.5708-PHI
      DFN=SIGMA*(H-STTM)*DTHE*DY*(1.+((H-STTM)/COS(THE)-G-X)*B**2*A**4/(
     1X*(B**4*X**2+A**4*Y**2)))*(1.+RNORM*(TAN(CPHI)**2+TAN(THE)**2/COS(
     2CPHI)))
      DNFP=SIGMA*(Y1MAX-Y1)*(H-STTM)*DTHE*(COS(CPHI)**2+RNORM*(SIN(CPHI)
     1**2+COS(CPHI)*TAN(THE)**2))*4.
      FONT=FONT+DFN*4.0
      GO TO 370
390   THEM=ACOS(1.0-STTM/H)
      FONTT=FONTT+FONT+DNFP
      IF (THE-THEM) 90,400,400
C
C     CALCULATE ENERGY ABSORBED, INCREMENT STROKE, AND GO TO BEGINNING
400   IF (FONTT.LE.0.0) GO TO 410
      IF (ARTSU.LE.0.0) GO TO 410
      FONTTM=FONTT
      TORTNM=TORTN
      FOFTTM=SIGMA*RSHCR*ARTSU
      TORTFM=FOFTTM*ARM
      RFONTT=(FONTTM*FOFTTM)**POW/(FOFTTM**POW+(FONTTM*AMU)**POW)
      FONTT=RFONTT**(1./POW)
      FOFTT=AMU*FONTT
      TORTNQ=TORTN*FONTT/FONTTM
      TORTFQ=FOFTT*ARM
      TORTT=TORTNQ+TORTFQ
410   CONTINUE
      IF (STINC.EQ.0.0) GO TO 440
      DU=(STTM-SPTM)*(FONTT+FPNTT)/2.0
      U=U+DU
      I=I+1
```

```
      ASAVE(I)=FONTTM
      BSAVE(I)=FONTT
      CSAVE(I)=TORTNM
      DSAVE(I)=FOFTT
      ESAVE(I)=ARTSU
      FSAVE(I)=ARMOM
      GSAVE(I)=FOFTTM
      HSAVE(I)=TORTFM
      SSAVE(I)=STTM
      TSAVE(I)=TORTT
      FPNTT=FONTT
      SPTM=STTM
      STTM=STTM+STINC
      GO TO 30
  420 CONTINUE
  430 CONTINUE
      GO TO 400
  440 DU=FONTT*(STINC+CLEAR-ALCL)
      U=U+DU
      VELBT=SQRT(5.36*U/WT)
      CLEAR=CLEAR+STINC
      WRITE (6,480)
      WRITE (6,490) A,B,C,D,E,F,G
      WRITE (6,500) RHOL,XKW,RNORM,RSHCR,STREF,AMU,POW
      WRITE (6,510) BET,RDIA,DX,DTHE,WT
      WRITE (6,520)
      WRITE (6,530) U,CLEAR,ALCL,VELBT
      N=I
      DO 450 L=1,N
      WRITE (6,540) SSAVE(L),BSAVE(L),DSAVE(L),TSAVE(L),ESAVE(L),FSAVE(L
     1)
  450 WRITE (6,550) ASAVE(L),CSAVE(L),GSAVE(L),HSAVE(L)
      GO TO 10
C
  460 FORMAT (1H1,4X,37HOMNIDIRECTIONAL LOADS CRUSHABLE TORUS////)
  470 FORMAT (9E8.1)
  480 FORMAT (20X,16HINPUT PARAMETERS//)
  490 FORMAT (1X,2HA=F6.2,3X,2HB=F6.2,3X,2HC=F6.2,3X,2HD=F6.2,3X,2HE=F6.
     12,3X,2HF=F6.2,3X,2HG=F6.2/)
  500 FORMAT (1X,5HRHOL=F5.1,3X,4HXKW=F8.1,3X,6HRNORM=F5.3,3X,6HRSHCR=F5
     1.3,3X,6HSTREF=F5.3,3X,4HAMU=F5.3,3X,4HPOW=F5.3/)
  510 FORMAT (1X,4HBET=F5.3,3X,5HRDIA=F5.2,3X,3HDX=F6.4,3X,5HDTHE=F6.4,3
     1X,3HWT=F9.1///)
  520 FORMAT (20X,11HOUTPUT DATA//)
  530 FORMAT (1X,2HU=F12.1,3X,6HCLEAR=F9.2,3X,5HALCL=F9.2,3X,6HVELBT=F9.
     12///)
  540 FORMAT (1X,5HSTTM=F6.3,5X,6HFONTT=F10.1,5X,6HFOFTT=F10.1,5X,6HTORT
     1T=F12.1,5X,6HARTSU=F9.2,5X,6HARMOM=F9.1)
  550 FORMAT (17X,7HFONTTM=F10.1,5X,7HTORTNM=F12.1,5X,7HFOFTTM=F10.1,5X,
     17HTORTFM=F12.1/)
      END
```

```
 15.1    15.1    0.0     0.0     27.98   29.32   0.0     5.3     3440.
 .785    5.0     .151    .010    1.0     5.      762.6
 0.8     0.7     0.3     2.0
```

APPENDIX B

OPERATING INSTRUCTIONS

FOR THE CRUSHABLE TORUS LANDING

LOADS AND MOTIONS PROGRAM

APPENDIX B

TABLE OF CONTENTS

APPENDIX B

B.1 INTRODUCTION

The Crushable Torus Landing Loads and Motions Program determines impact loads and resulting spatial positions, velocities, and accelerations as a function of time for an intermediate landing vehicle with a crushable torus impact attenuator. The lander configuration may be established with the Crushable Torus Structural Design Program (see Appendix A), or a lander design may be available from some other source.

Features incorporated in this program include the ability to: select up to six degrees of freedom thereby allowing simulation of spatial motion; vary crushable attenuator elastic recovery, lander geometry, surface slope, coefficient of friction, and rock diameter; select values for as many as eight independent parameters used to stop machine computation; and select variable or constant step Predictor-Corrector or Runge-Kutta integration methods. Input data to this program includes initial lander attitude, position, linear and rotational velocities, lander geometric and inertial properties, and surface conditions such as ground slope, coefficient of friction, and rock diameter.

Output data consists of the body forces, attenuator and frictional forces and resulting moments, and lander translational and angular positions, velocities, and accelerations, as a function of time.

The Crushable Torus Landing Loads and Motions Program was developed for a landing vehicle as shown in Figure B-1. The lander is composed of two main parts; the crushable attenuator and the payload.

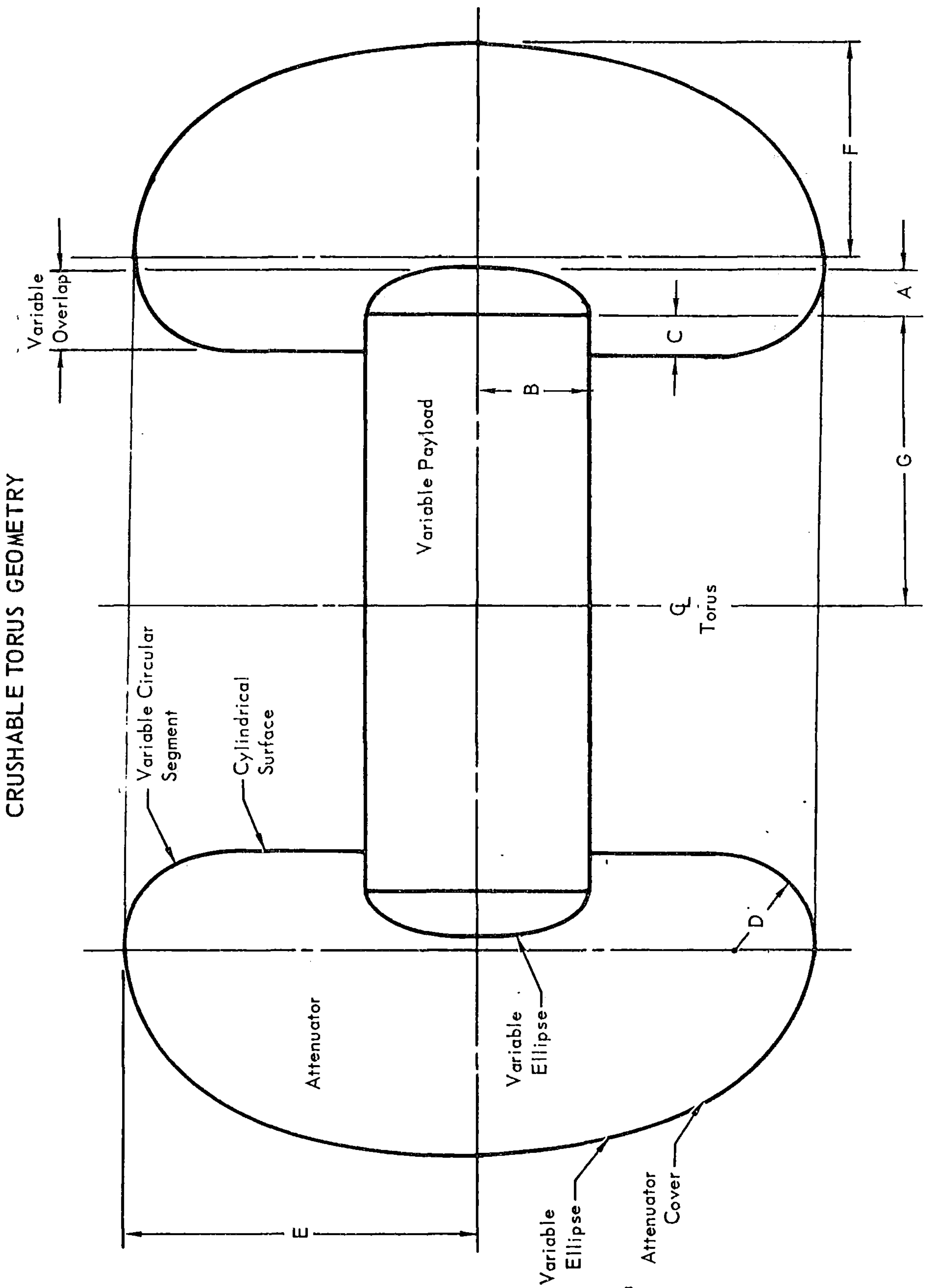

CRUSHABLE TORUS GEOMETRY
Variable Overlap
A
C
B
F
G
Variable Payload
℄ Torus
Variable Circular Segment
Cylindrical Surface
D
Attenuator
Variable Ellipse
Variable Ellipse
Attenuator Cover
E

Figure B-1

APPENDIX B

The attenuator can be shaped as a hollow sphere, a hollow ellipsoid, or a torus with a circular, elliptical, or a combination circular, elliptical, and cylindrical cross section. Any type of attenuator material, such as balsa wood, foams, and honeycombs may be considered. The payload can be spherical, ellipsoidal, or cylindrical with hemispherical or ellipsoidal edge. The dimensions for these shapes can also be varied.

APPENDIX B

B.2 ANALYTICAL PROCEDURES

B.2.1 COORDINATE SYSTEMS - Three coordinate systems used to define the motion of the lander as a function of time are shown in Figure B-2. All three systems are right-handed and each consists of three orthogonal axes. These coordinate systems are defined as follows:

- A coordinate system moving with the lander and fixed at its center of gravity (X, Y, Z). This system is referred to as the lander coordinate system. The X axis is the axis of symmetry of the landing vehicle. Roll, pitch, and yaw axes coincide with the reference X, Y, and Z axes respectively. In defining the signs of rotation, the right-hand rule is used. Since the vehicle is symmetrical about any plane containing the X axis direction at the origin, the Y and Z directions are chosen arbitrarily.
- A coordinate system fixed in the planet and aligned with the gravity vector (X_f, Y_f, Z_f). This system is referred to as the gravity coordinate system. The Z_f axis is directed toward the center of the planet and the positive X_f and Y_f axes are directed toward the north and east respectively.
- A coordinate system fixed in the planet and oriented with respect to the slope of the local surface (X_{ls}, Y_{ls}, Z_{ls}). This system is referred to as the surface coordinate system and differs from the gravity coordinate system by the rotation α about the X_f axis.

The coordinate systems are related by the following expression where TR and TRL are matrices of direction cosines:

$$\begin{Bmatrix} X_f \\ Y_f \\ Z_f \end{Bmatrix} = \begin{bmatrix} \mathrm{TR}\ (I,\ J) \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

and

$$\begin{Bmatrix} X_{ls} \\ Y_{ls} \\ Z_{ls} \end{Bmatrix} = \begin{bmatrix} \mathrm{TRL}\ (I,\ J) \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

These transformations are used for relating forces, accelerations, velocities, and displacements between the various coordinate systems.

During the integration of the equations of motion, one or more of the direction cosines may become slightly greater than one. This is a result of the lander experiencing high angular velocities and is due to the finite step nature of the numerical integration methods employed. When this occurs, the program prints out a message, recomputes the direction cosines such that the direction vector is normalized, and continues the integration.

B.2.2 ASSUMPTIONS - During the analytical model development, several assumptions were made to reduce program complexity consistent with the scope of the study. A summary of the basic assumptions follows:

a. Only rigid body motion is considered.

b. A constant friction coefficient is assumed. (A mechanism is provided in the program for reducing motion oscillation caused by friction forces when the contact surface is approaching zero velocity.)

COORDINATE SYSTEMS

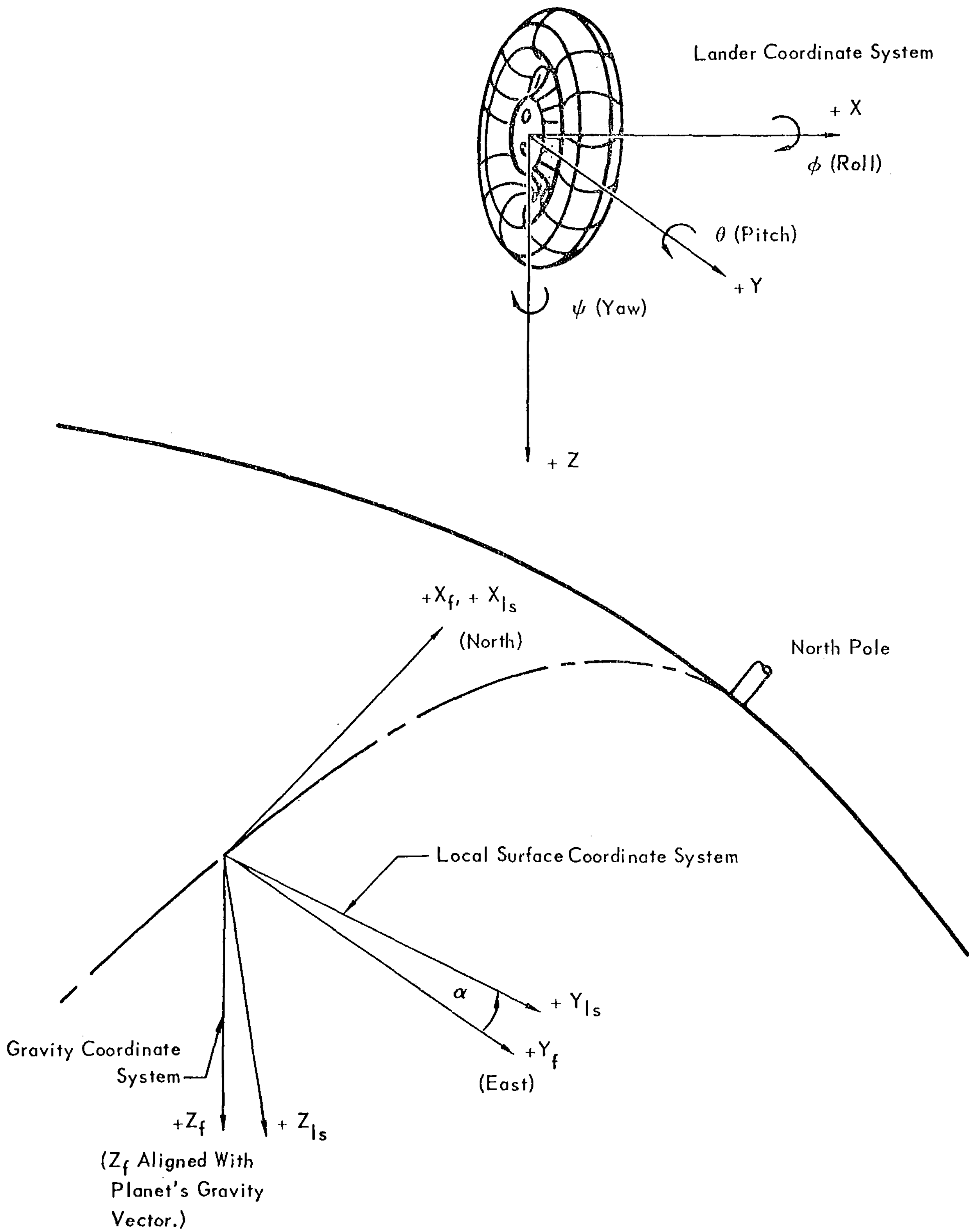


Figure B-2

c. Aerodynamic forces are negligible.

d. Mass moment of inertia changes due to crushing the attenuator are negligible.

e. The general geometric shape is assumed (see Figure B-1).

f. The crush plane always contacts virgin material. This assumption would be violated if there is a second bounce on the same attenuator crush plane or if successive crush planes intersect (see Figure B-3).

g. Load-stroke subroutine assumptions are identical to those for the load-stroke portion of the Crushable Torus Structural Design Program, Appendix A.

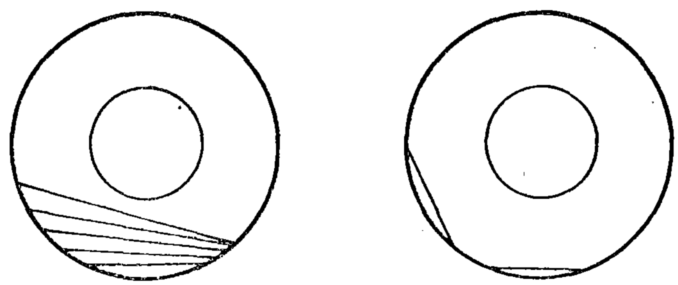
h. The crushable attenuator and the payload are each assumed to have a constant density.

B.2.3 METHODS OF ANALYSIS

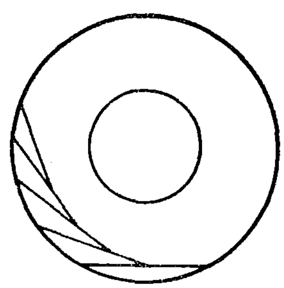
B.2.3.1 Loads Calculation. - The external loads on the vehicle are the crushing force and moments, the frictional force and moments, and the gravitational force.

The crushing force is the resultant force normal to the contact area (footprint) necessary to crush the attenuator. A vehicle moment results if this force does not act through the lander center of gravity. This force and moment are calculated in the LDSTR subroutine (which is an adaptation of part of the Crushable Torus Structural Design Program) and the analyses and assumptions used to determine them are defined in Appendix A.

POTENTIAL ATTENUATOR CRUSH PLANE POSITIONS



NON-INTERSECTING CRUSH PLANES
(Program Assumption)



INTERSECTING CRUSH
PLANES

Figure B-3

APPENDIX B

The location of the crush plane, needed for the crushing force calculation, is defined by the maximum attenuator stroke (at each time point) and the angle (β) between the crush plane and the vehicle Y-Z plane.

The angle (β) is calculated from the direction cosine matrix used to describe the vehicle orientation (see Section B.2.1).

The maximum stroke equation, for β less than 90°, is

$$\text{STROKE} = -\text{ZLS} + \left[(G+D-C)\sin\beta + (E^2+F^2\tan^2\beta)\right]^{1/2}\cos\beta$$

for $\beta = 90°$

$$\text{STROKE} = -\text{ZLS} + G - C + D + F$$

These symbols are defined in Figure B-4.

The frictional forces are calculated assuming a constant coefficient of friction, and their direction is opposite to the motion of the centroid of the footprint area. The frictional forces are reduced when their magnitude is greater than that required to stop the vehicle footprint centroid within one integration time interval and to resist the forces and/or moments tending to move it. This is done to prevent oscillatory motion (instability of the calculations) as the footprint centroid motion stops. A frictional resisting moment is also generated by pure rotation of the contact (footprint) area of the attenuator. These calculations are made in the FORM subroutine.

B.2.3.2 Equations of Motion. - Vehicle motions and displacements as a function of time are obtained by numerically integrating the vehicle differential equations of motion. The equations of motion, in vehicle coordinates are:

APPENDIX B

$$
m \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix} - m \begin{bmatrix} 0 & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & 0 & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & 0 \end{bmatrix} \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix} + m \begin{Bmatrix} g_X \\ g_Y \\ g_Z \end{Bmatrix}
$$

and

$$
\begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{Bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix} = \begin{Bmatrix} T_X \\ T_Y \\ T_Z \end{Bmatrix} - \begin{bmatrix} \dot{I}_{XX} & -\dot{I}_{XY} & -\dot{I}_{XZ} \\ -\dot{I}_{XY} & \dot{I}_{YY} & -\dot{I}_{YZ} \\ -\dot{I}_{XZ} & -\dot{I}_{YZ} & \dot{I}_{ZZ} \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}
$$

$$
- \begin{bmatrix} 0 & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & 0 & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}
$$

Selection of the number of degrees of freedom of motion allowed (up to six) is an input to the program; if less than six degrees of freedom are allowed, some calculations in the program are by-passed and the program running time is reduced.

Three numerical integration methods for integrating the equations of motion are available; fixed step fourth order Runge - Kutta technique and fixed or variable step Adams-Moulton Predictor-Corrector techniques. Their selection is controlled by input indicators (see Figure B-5).

APPENDIX B

SYMBOL DEFINITION

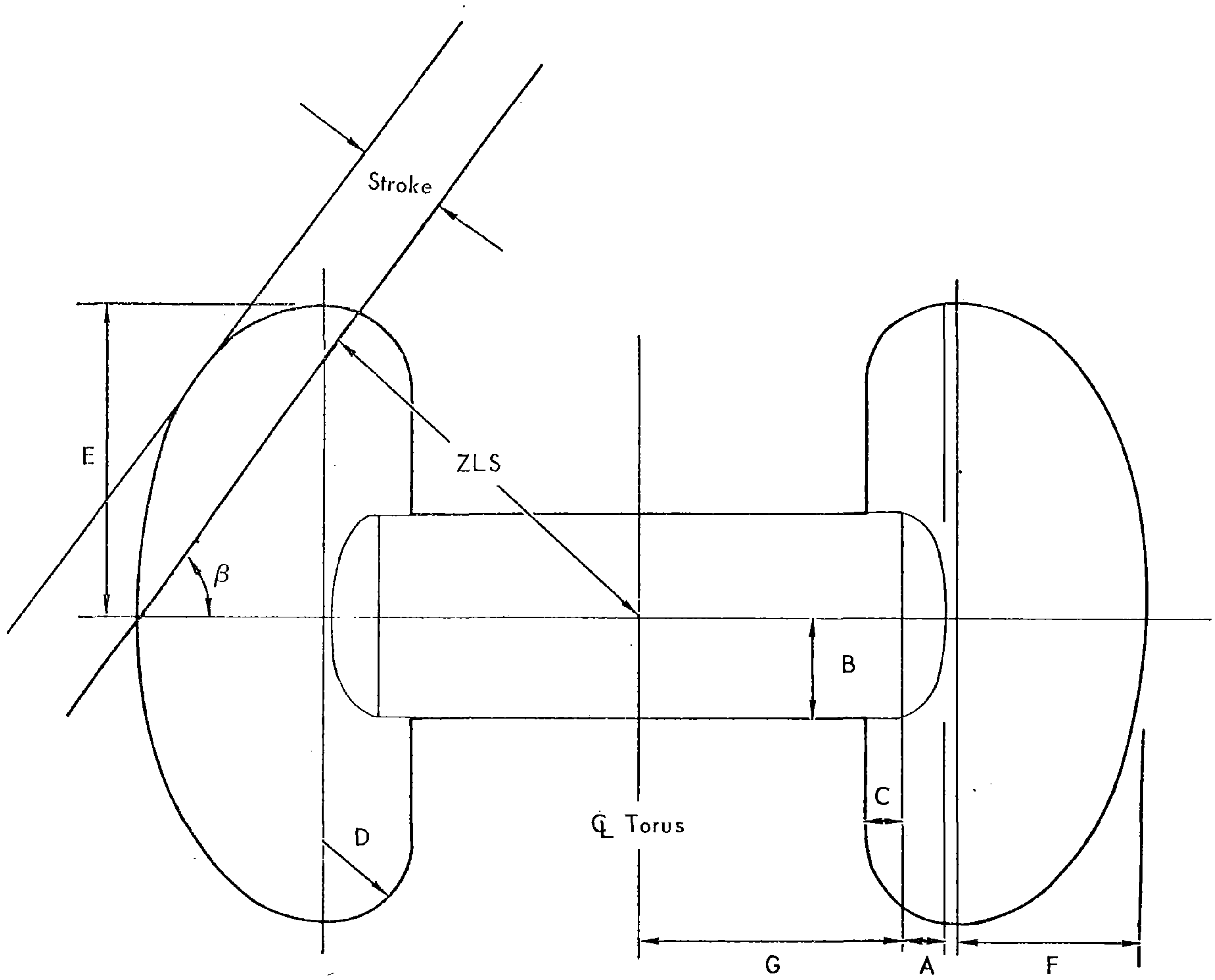


Figure B-4

APPENDIX B

B.3 PROGRAM OPERATION

B.3.1 INPUT DATA

B.3.1.1 Input Quantities. - Information describing the geometric and mass properties of the specific lander to be studied, the lander's initial positions and velocities, the planet surface conditions, and a number of indicators to initialize the integration routines are required as input data by the program. This section discusses the required input quantities while the mechanics of setting up the input data, required data card format, input position definition, and the additional option to modify the output quantities through input indicators is discussed in Section B.3.1.2.

Input parameters are defined in Figure B-5. Many of these quantities are adequately explained in this figure, but a number of them require additional comments.

The program initialization routine assumes that the initial rotations of the lander are carried out in the order of yaw (ψ), pitch (θ), and roll (ϕ). This point must be considered in determining the magnitudes of these rotations to locate the lander at the desired initial angular orientation.

When any of the following indicators are read in as zero, they are reset internally in the program with the nominal values indicated below. This is to guarantee successful initialization of the program routines.

INPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

PROGRAM VARIABLE	LOCATION INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
IP.	001	–	–	Quantity Used to Determine Initial Integration Time Interval. $\Delta t = \Delta t_{max}/2^{+IP}$ (i.e. HZ = HMAX * 2. ** (– IP))
IVARH	002	–	–	Integration Interval Indicator IVARH = 0 – Variable Interval Integration IVARH = 1 – Fixed Interval Integration
IMTH	003	–	–	Integration Method Indicator IMTH = 0 – Predictor-Corrector IMTH = 1 – Runge-Kutta
EMAX	004	–	–	Maximum Integration Accuracy Required for Variable Step Predictor-Corrector Integration (See Appendix F)
EMIN	005	–	–	Minimum Integration Accuracy Required for Variable Step Predictor-Corrector Integration (See Appendix F) .
HMIN	006	Δt_{min}	Sec	Minimum Integration Time Interval. (Variable Step Predictor-Corrector Integration)
NTX	007	–	–	Indicator to Suppress Degree of Freedom Along Lander X Axis. NTX = 0 – Allow Degree of Freedom NTX = 1 – Suppress Degree of Freedom
NTY	008	–	–	Indicator to Suppress Degree of Freedom Along Lander Y Axis (See NTX).
NTZ	009	–	–	Indicator to Suppress Degree of Freedom Along Lander Z Axis (See NTX).
NRX	010	–	–	Indicator to Suppress Degree of Freedom About Lander X Axis. NRX = 0 – Allows Degree of Freedom NRX = 1 – Suppress Degree of Freedom
NRY	011	–	–	Indicator to Suppress Degree of Freedom About Lander Y Axis (See NRX).
NRZ	012	–	–	Indicator to Suppress Degree of Freedom About Lander Z Axis (See NRX).
MMIC	013	–	–	Mass Moments of Inertia Indicator MMIC = 0 – Constant Inertia MMIC = 1 – Variable Inertias (Not Used)
NCUT	014	–	–	Number of Cutoff Variables Considered (Maximum of 8)
IERPRT	015	–	–	Time History Print Format Indicator IERPRT = 0; Three Lines Per Print Time (Option 1) IERPRT = 1; Six Lines Per Print Time (Option 2)
XS(J)	019	–	Sec	Real Time Cutoff Limit (T).

Figure B-5

INPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
IND(J)	020	–	–	Limit Direction Indicator for T. IND(J) = 0 – Limit on Increase IND(J) = 1 – Limit on Decrease
XS(J)	025	–	Ft/Sec	Total Surface Velocity Cutoff Limit (TOTSV)
IND(J)	026	–	–	Limit Direction Indicator for TOTSV.
XS(J)	031	–	Ft	Surface Range Cutoff Limit (RANGE)
IND(J)	032	–	–	Limit Direction Indicator for RANGE
XS(J)	037	–	Ft/Sec	Velocity Parallel to Surface Cutoff Limit (TLSV)
IND(J)	038	–	–	Limit Direction Indicator for TLSV
XS(J)	043	–	Ft	Distance Normal to Surface Cutoff Limit (ZLS)
IND(J)	044	–	–	Limit Direction Indicator for ZDLS
XS(J)	049	–	Rad/Sec	Roll Rate Cutoff Limit (PHID)
IND(J)	050	–	–	Limit Direction Indicator for PHID
XS(J)	055	–	Rad/Sec	Pitch Rate Cutoff Limit (THETAD)
IND(J)	056	–	–	Limit Direction Indicator for THETAD
XS(J)	061	–	Rad/Sec	Yaw Rate Cutoff Limit (PSID)
IND(J)	062	–	–	Limit Direction Indicator for PSID
T	067	t_o	Sec	Initial Value of Time at Start of Integration
HMAX	068	Δt_{max}	Sec	Maximum Integration Time Interval
KOUNT2	069	–	–	Number of Integration Intervals Between Print Times
NMBNCS	070	–	–	Number of Lander Bounces Desired.
XF	073	X_f	Ft	Initial Lander c.g. Location Along X_f Axis
XVF(1)	074	X_f	Ft/Sec	Initial Lander c.g. Velocity Along X_f Axis
YF	075	Y_f	Ft	Initial Lander c.g. Location Along Y_f Axis
XVF(2)	076	Y_f	Ft/Sec	Initial Lander c.g. Velocity Along Y_f Axis
ZF	077	Z_f	Ft	Initial Lander c.g. Location Along Z_f Axis
XVF(3)	078	Z_f	Ft/Sec	Initial Lander c.g. Velocity Along Z_f Axis
PHI	079	ϕ	Deg	Initial Lander Angular Position About X Axis Following Initial Yaw (ψ) and Pitch (θ)
XD(1,4)	080	$\dot{\phi}$	Rad/Sec	Initial Lander Angular Velocity About X Axis
THETA	081	θ	Deg	Initial Lander Angular Position About Y Axis Following Initial Yaw (ψ)
XD(1,5)	082	$\dot{\theta}$	Rad/Sec	Initial Lander Angular Velocity About Y Axis
PSI	083	ψ	Deg	Initial Lander Angular Position About Z Axis

Figure B-5 (continued)

INPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
XD(1, 6)	084	$\dot{\psi}$	Rad/Sec	Initial Lander Angular Velocity About Z Axis
RANGE	085	–	Ft	Initial Arc Distance Traveled in Local Surface at Start of the Integration
SLOPE	086	α	Deg	Angle Between Y_{l_s} Axis and Y_f Axis. Measured Positive About X_f Axis in the Negative Right-Hand Direction.
STREF	098			Stroke Efficiency (Stroke/Initial Height) (See Appendix A)
A	100		In.	Geometric Dimension (See Figure B–1)
B	101		In.	Geometric Dimension (See Figure B–1)
C	102		In.	Geometric Dimension (See Figure B–1)
D	103		In.	Geometric Dimension (See Figure B–1)
E	104		In.	Geometric Dimension (See Figure B–1)
F	105		In.	Geometric Dimension (See Figure B–1)
G	106		In.	Geometric Dimension (See Figure B–1)
RHOL	107		Lb/Ft^3	Attenuator Density
RHOP	108		Lb/Ft^3	Payload Density
POW	109			Exponent Used in Interaction Equation for Shear and Radial Stress (See Appendix A)
RSHCR	110			Ratio of Allowable Shearing Stress to Radial Crushing Stress of Attenuator (See Appendix A)
VMIN	111		Rad/Sec	Angular Velocity at Which Footprint Torque Due to Friction Decreases Towards Zero
RDIA	112		In.	Surface Rock Diameter
DX	113		In.	Incremental Value of X Used in the LDSTR Subroutine Recommended Value – 0.02 to 0.05 Times A
DTHE	114		Rad	Incremental Value of θ in LDSTR Subroutine Recommend Value – 0.005 End, 0.030 Flat; Proportional in Between (Larger Values Reduce Run Time and Accuracy of Load Calculation)
RNORM	115			Ratio of Transverse Crush Strength to Radial Strength of the Attenuator (See Appendix A)

Figure B-5 (continued)

INPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
ELST	117		In.	Elastic Attenuator Stroke Recovery
AMU	119			Coefficient of Friction
XKW	120		Ft Lbs/Lb	Attenuator Specific Energy (See Appendix A).
PLTMAS	130		Slugs	Planet Mass (Used to Calculate the Acceleration of Gravity)
PLTRAD	131		Ft	Planet Radius (Distance from Planet C.G. to Origin of Fixed Surface Coordinates). (Used to Calculate the Acceleration of Gravity).

Note – Last Data Card of Type 2 Must Contain a 1 in Column 2.
The following information is required on each data card requesting a nonstandard *output* variable.
(See Section B.3.1.2)

VNAME Name printed out to identify the *nonstandard* variable.
LCOMN Subscript of the common *array* COMINT element in which the desired nonstandard output variable is located.
LPT Print position in *time history* block in which desired output variable will be printed.

Note – Last data card in *a data* set *must* contain EOD in Columns 2 – 5.

Figure B-5 (continued)

EMAX = 0	EMAX set equal to 1×10^{-4} (see Appendix F)
EMIN = 0	EMIN set equal to 1×10^{-6} (see Appendix F)
HMIN = 0	HMIN set equal to HMAX * 2^{-16} seconds
NMBNCS = 0	NMBNCS set equal to 100
DTHE = 0	DTHE set equal to 0.03 radians

In addition, if either NCUT or VMIN is initially zero, the program will terminate with an error message.

The program considers a lander whose moments of inertia are constant with time, although capability for subsequent inclusion of variable inertias was provided. Therefore, the mass moments of inertia indicator, MMIC, must be set equal to zero.

In order for the program to use the variable inertia option, the time rates of change of the moments and cross products of inertia would have to be calculated.

Several methods for obtaining computational termination for a particular lander case are provided in the program. One feature results in termination when any one of up to eight variables reaches a preset cutoff value. The eight cutoff variables are as follows:

- Real time (T)
- Total surface velocity (TOTSV)
- Surface range (RANGE)
- Velocity parallel to surface (TLSV)

- o Distance normal to surface (ZLS)
- o Roll rate (PHID)
- o Pitch rate (THETAD)
- o Yaw rate (PSID)

XS (J) is the array which contains the cutoff value for each cutoff variable. The array IND(J) contains indicators which define whether the cutoff variables are increasing or decreasing toward the cutoff values. These cutoff indicators are set as follows:

IND(J) = 0 - Cutoff variable increasing to cutoff limit.

IND(J) = 1 - Cutoff variable decreasing to cutoff limit.

In these two arrays, the subscript J indicates the order in which these quantities are read into the program. The time history variable to which a specific cutoff value applies is governed by its location in the data field.

The number of lander impacts to be considered in a particular case is governed by the input quantity NMBNCS. The program terminates when the lander leaves the ground at the end of the last impact of interest. In addition, a specific case is terminated if the crushable attenuator is displaced into the payload clearance envelope (indicating inadequate attenuator dimensions).

The quantities PLTRAD and PLTMAS are used to determine the planet's acceleration of gravity as a function of altitude. This relationship is given as:

$$GZ = g = \frac{-G * \text{PLTMAS}}{(\text{PLTRAD} \quad -Z_f)^2}$$

In this expression:

$GZ = g$ = acceleration of gravity (ft/sec^2).

G = Universal Gravity Constant (1.0684×10^{-9} ft^3/lb-sec^2).

PLTMAS = Planet's mass (lb-sec^2/ft).

PLTRAD = Planet's radius (ft).

Z_f = Position of lander center of gravity in local surface Z_{ls} axis.

An example of the required data setup is given in Section B.3.3. Also shown are the various options which are available for the output data format.

B.3.1.2 Input Format. - All of the data cards required for the execution of any one case are referred to as a data set. As a data set is read, the values of the input variable with appropriate labels and messages are printed for further reference when evaluating the resulting lander time history output. A data set must be terminated by a data card containing EOD in columns 2-4 with columns 5-11 blank. There is no program limit to the number of data sets that can be run during any one job.

Figure B-6 shows the required format for the input data cards. The input data can be thought of as a one dimensional array whose elements contain the values of the various input quantities. The input routine is completely compatible with the Inflatable Torus Landing Loads and Motions Programs (Appendix D) and this fact accounts for some of the apparent blanks in the input format. Many of the locations in the input field are used to input quantities required with the analysis of the inflatable system. These input positions are noted in Figure B-6.

FORMAT FOR INPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

(1)	(I)	DATA (I)	DATA (I + 1)	DATA (I + 2)	DATA (I + 3)	DATA (I + 4)	DATA (I + 5)
	ANY	DESCRIPTIVE	COMMENTS OR	LABELS OR	DATES		
	ANY	DESCRIPTIVE	COMMENTS				
	ANY	DESCRIPTIVE	COMMENTS				
	001	IP	IVARH	IMTH	EMAX	EMIN	HMIN
	007	NTX	NTY	NTZ	NRX	NRY	NRZ
	013	MMIC	NCUT	IERPRT			
	019	XS(J)	IND(J)				
	025	XS(J)	IND(J)				
	031	XS(J)	IND(J)				
	037	XS(J)	IND(J)				
	043	XS(J)	IND(J)				
	049	XS(J)	IND(J)				
	055	XS(J)	IND(J)				
	061	XS(J)	IND(J)				
	067	T	HMAX	KOUNT2	NMBNES		
	073	XF	XVF(1)	YF	XVF(2)	ZF	XVF(3)
	079	PHI	XD(1,4)	THETA	XD(1,5)	PSI	XD(1,6)
	085	RANGE	SLOPE				
	091						
	097		STREF	(2)	A	B	C
	103	D	E	F	G	RHOL	RHOP
	109	POW	RSHCR	VMIN	RDIA	DX	DTHE
	115	RNORM		ELST		AMU	XKW
	121	(2)	(2)	(2)	(2)	(2)	(2)
1	127	(2)	(2)	(2)	PLTMAS	PLTRAD	(2)
VNAME		LCOMN	LPT				
VNAME		LCOMN	LPT				
VNAME		LCOMN	LPT				
EOD							

1. A "1" in Column 2 of any data card identifies that card as the last one containing data for the DATA matrix for that case.
2. These spaces are used for data for the Inflatable Torus Landing Loads and Motion Program.

Figure B-6

APPENDIX B

A data set for the Crushable Torus Landing Loads and Motions Program may consist of three different types of data cards. These three types of cards are described below in the order they are required within a data set.

Type 1:

In any one data set the first three cards are always used as a description of the case to be run. These cards have the following format.

Column 1:	Must be left blank
Columns 2 - 55:	May contain any descriptive comments which will be printed at the beginning of the input data listing.
Columns 56 - 80:	Are not read by the program and may be used for comments or identification statements.

Type 2:

The assignment of initial values to the program variables by input data is accomplished with this format. All data values read from these cards are first stored in the array DATA. The various input variables are then assigned their respective initial values by equating the variables to a specified DATA element. Since the DATA elements are initially set equal to zero, an input variable will have an initial value of zero unless the corresponding DATA element is changed by card input. The few exceptions to this are the indicators discussed in Section B.3.1.1.

The input quantities contained in the six data fields of a data card are stored consecutively in the DATA array. Each card contains a subscript which governs the data position (in the array DATA) of each variable on that card. The data position of an input variable is obtained by assigning the first variable on a card the data position equal to the card subscript. Each following variable, moving from left to right on the card, has a data position one increment larger. For example (Figure B-6), the coefficient of friction (AMU) has a data position of 119. Thus, a particular data card needed to define a given variable is identified by the card subscript.

In the instance of multiple data sets, all of the previous data set is retained except those quantities changed through reading the cards of Type 2. However, each data set must contain at least one card of Type 2.

The following describes the required format for the Type 2 input cards.

Column 2:	Read as an integer number with an I1 format. If the value is zero, another card is read in the same format. If the value is not zero, the card represents the last card of Type 2 in the data set. The last Type 2 card in a data set must contain a non-zero value in this column.
Columns 5 - 7:	Read as an integer number with an I3 format. This field represents the data card subscript for the particular card.
Columns 10-19, 20-29, 30-39, 40-49, 50-59, and 60-69:	Read as a real variable with an E10.5 format. All data values must be input as floating point numbers. Integer variables are converted from floating point quantities by the program.
Columns 1, 8-9, and 70-80:	These are not read by the program and may be used for comments and identification statements.

Type 3:

Information on these cards is used when the optional output routine is desired. The option allows a different variable to temporarily replace a standard output variable in that variable's output postion.

There may be as many as 72 input cards of this type. Each card contains an identification name (one to ten characters), a subscript locating the desired output variable in the program array COMINT, and an indicator giving the desired print position of the variable.

Columns 2-11:	The ten characters in those columns are used as the identification name for the variable to be printed. (VNAME)
Columns 15-19:	Read as an integer number with an I5 format. This value is the subscript of the common array COMINT element in which the desired output variable is located (see Appendix E). (LCOMN)
Columns 21-25:	Read as an integer number with an I5 format. This value is used to specify the print position in which the desired output variable will be printed. The output routine's print positions are defined in Section B.3.2. (LPT)
Columns 1, 12-14, 20, and 26-80:	These positions are not read by the program and may be used for comments and identification statements.

For a multi-case computer run, if the output format is modified in a data set, the succeeding cases will have the same modification unless changed by their data set. New runs revert to the standard format unless they are modified by their data set.

B.3.2 OUTPUT DATA

B.3.2.1 Output Quantities. - At specified times during the integration routine, various time varying quantities defining the lander's positions, velocities, and accelerations; applied forces and moments; and other items of interest such as attenuator crush stroke and lander distance traveled are printed. In addition, an output option is available whereby the standard output variables may be replaced by other quantities of interest. This procedure is discussed in detail as Type 3 in Section B.3.1.2. The output information is printed by a call to the PRINT subroutine. The output format resulting from this subroutine is discussed more fully in the following section, Section B.3.2.2.

Initial output from the program presents the input data read in for the specific case being considered. Following this information are the weight and inertia properties of the lander. The time histories describing the lander's motion are then given.

Figure B-7 shows the form of the standard heading which appears at the top of each page of time history output. The values of the output variables appear as blocks of data following this heading and are printed in the same

STANDARD OUTPUT HEADING

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

LOCAL SURFACE COORDINATE SYSTEM						LANDER COORDINATE SYSTEM					VEHICLE ORIENTATION		
XLS	XDLS	XDDLS	FLT. ANGLE	FONTT	*	XD	XDD	PHID	PHIDD	*	ANG(1,1)	ANG (1,2)	ANG (1,3)
YLS	YDLS	YDDLS	VER. ANGLE	TORTN	*	YD	YDD	THETAD	THETADD	*	ANG(2,1)	ANG (2,2)	ANG (2,3)
ZLS	ZDLS	ZDDLS	STROKE	FRICT	*	ZD	ZDD	PSID	PSIDD	*	ANG (3,1)	ANG (3,2)	ANG (3,3)
RANGE	XCF (1)	T VELOCITY	XN	FS (1)	*	XPF (1)	FF (1)	FF (4)	TFC (1)	*	ANGLD	BLANK	BLANK
ACLR	XCF (2)	YLGRAV	YN	FS (2)	*	XPF (2)	FF (2)	FF (5)	TFC (2)	*	ANGLD2	BLANK	BLANK
GZ	XCF (3)	ZLGRAV	ZN	ANGLD3	*	XPF (3)	FF (3)	FF (6)	TFC (3)	*	BLANK	BLANK	BLANK

Figure B-7

format as the heading. At the top of each block of output, a line of information giving the Central Processor (CP) time since the start of the job, the real time, and the present integration step size is printed. Following these is the block of output quantities corresponding to the printed value of real time.

Figure B-8 defines the standard output parameters which are available in the program. Also shown are the appropriate units of the variable and its respective print position. The top line of the output consists of print positions 1-12, the second line consists of print positions 13-24, and so forth to line six which consists of positions 61-72.

Many of the output parameters are completely defined in Figure B-8, but some require further comment.

The STROKE parameter is the maximum attenuator crush distance measured normal to the crush plane. The resultant normal crushing force (FONTT) is assumed to act only when STROKE is positive and increasing and, on a linearly reducing scale, during the elastic rebound distance (ELST). This would be typically 1% to 5% of the total stroke. FONTT goes from maximum stroke value to zero over the ELST distance. When STROKE is negative, it is the distance normal to the local surface of the uncrushed attenuator's closest point.

The variables XPF (I), XCD (I), ANGLD2, ANGLD3, FS (I), and TFC (I) are not calculated when FONTT is zero, and are printed out as zero.

STANDARD OUTPUT DATA – CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

PROGRAM VARIABLES	UNITS	VARIABLE DEFINITION
LANDER WEIGHT AND INERTIA PROPERTIES		
WTERTH	Lbs	Lander Weight on Earth
WTPLNT	Lbs	Lander Weight on Planet
XXI	Lb Ft Sec^2	Lander Moment of Inertia About X Axis
YYI	Lb Ft Sec^2	Lander Moment of Inertia About Y Axis
ZZI	Lb Ft Sec^2	Lander Moment of Inertia About Z Axis
XYI	Lb Ft Sec^2	Lander XY Product of Inertia
XZI	Lb Ft Sec^2	Lander XZ Product of Inertia
YZI	Lb Ft Sec^2	Lander YZ Product of Inertia
XXID	Lb Ft Sec	Rate of Change of XXI (Not Used)
YYID	Lb Ft Sec	Rate of Change of YYI (Not Used)
ZZID	Lb Ft Sec	Rate of Change of ZZI (Not Used)
XYID	Lb Ft Sec	Rate of Change of XYI (Not Used)
XZID	Lb Ft Sec	Rate of Change of XZI (Not Used)
YZID	Lb Ft Sec	Rate of Change of YZI (Not Used)

PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
INTEGRATION TIME HISTORY PARAMETERS				
TIMECP		CP Time	Sec	Computer Central Processor Elapsed Time
T			Sec	Real Time Elapsed (Starts at Initial Input Time)
HZ		DT	Sec	Integration Step Time Interval
GROUND REFERENCED TIME HISTORY PARAMETERS, 3 LINE FORMAT OPTION (Local Surface Referenced Unless Otherwise Stated)				
XLS	1		Ft	Lander Center of Gravity (C.G.) X_{ls} Location
XDLS	2		Ft/Sec	Lander C.G. Velocity, X_{ls} Component
XDDLS	3		Ft/Sec^2	Lander C.G. Acceleration, X_{ls} Component
YLS	13		Ft	Lander C.G. Y_{ls} Location
YDLS	14		Ft Sec	Lander C.G. Velocity, Y_{ls} Component
YDDLS	15		Ft/Sec^2	Lander C.G. Acceleration, Y_{ls} Component
ZLS	25		Ft	Lander C.G. Z_{ls} Location
ZDLS	26		Ft/Sec	Lander C.G. Velocity, Z_{ls} Component
ZDDLS	27		Ft/Sec^2	Lander C.G. Acceleration, Z_{ls} Component
SIGMA	4	Flt Ang.	Deg	Heading Angle From North Toward East (On X_f-Y_f Plane)
GAMA	16	Ver Ang	Deg	Resultant C.G. Velocity Angle (From the X_f-Y_f Plane)!
STROKE	28		In.	Maximum Attenuator Stroke Normal to Crush Plane.
FONTT	5		Lb.	Resultant Crush Force On Attenuator (Normal to Crush Plane)
TORTN	17		Ft Lb	Moment About the Lander C.G. Caused by FONTT.
FRICT	29		Lb	Resultant Frictional Force on Attenuator Footprint.
VEHICLE COORDINATE REFERENCED TIME HISTORY PARAMETERS, 3 LINE FORMAT OPTION				
XD (1,1)	6	XD	Ft/Sec	Lander C.G. Velocity, X Component
XDD (1,1)	7	XDD	Ft/Sec^2	Lander C.G. Acceleration, X Component
XD (1,2)	18	YD	Ft/Sec	Lander C.G. Velocity, Y Component
XDD (1,2)	19	YDD	Ft/Sec^2	Lander C.G. Acceleration, Y Component
XD (1,3)	30	ZD	Ft/Sec	Lander C.G. Velocity, Z Component
XDD (1,3)	31	ZDD	Ft/Sec^2	Lander C.G. Acceleration, Z Component
XD (1,4)	8	PHID	Rad/Sec	Rotational Velocity About X Axis
XDD (1,4)	9	PHIDD	Rad/Sec^2	Rotational Acceleration About X Axis
XD (1,5)	20	THETAD	Rad/Sec	Rotational Velocity About Y Axis
XDD (1,5)	21	THETADD	Rad/Sec^2	Rotational Acceleration About Y Axis
XD (1,6)	32	PSID	Rad/Sec	Rotational Velocity About Z Axis
XDD (1,6)	33	PSIDD	Rad/Sec^2	Rotational Acceleration About Z Axis

Figure B-8

STANDARD OUTPUT DATA– CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Cont'd)				
PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNIT	VARIABLE DEFINITION
VEHICLE ORIENTATION TIME HISTORY PARAMETERS, 3 LINE FORMAT OPTION (With Respect to the Local Surface)				
ANG (1,1)	10		Deg	Direction Angle Between Local Surface X Axis (X_{ls}) and Lander X Axis (X).
ANG (2,1)	22		Deg	Direction Angle Between Y_{ls} and X
ANG (3,1)	34		Deg	Direction Angle Between Z_{ls} and X
ANG (1,2)	11		Deg	Direction Angle Between X_{ls} and Y
ANG (2,2)	23		Deg	Direction Angle Between Y_{ls} and Y
ANG (3,2)	35		Deg	Direction Angle Between Z_{ls} and Y
ANG (1,3)	12		Deg	Direction Angle Between X_{ls} and Z
ANG (2,3)	24		Deg	Direction Angle Between Y_{ls} and Z
ANG (3,3)	36		Deg	Direction Angle Between Z_{ls} and Z
ADDITIONAL GROUND REFERENCED TIME HISTORY PARAMETERS, 6 LINE FORMAT OPTION (Local Surface Referenced Unless Otherwise Stated)				
RANGE	37		Ft	Distance Traveled Across Local Surface (Arc Length) (Initial Value is Input)
ACLR	49		In.	Remaining allowable attenuator stroke
GZ	61		Ft/Sec^2	Acceleration of Gravity at Lander Altitude
XCF (1)	38		Ft/Sec	Attenuator Footprint Centroid Velocity X_{ls} Component
XCF (2)	50		Ft/Sec	Attenuator Footprint Centroid Velocity Y_{ls} Component
XCF (3)	62		Ft/Sec	Attenuator Footprint Centroid Velocity Z_{ls} Component
TOTGV	39	TOT VEL	Ft/Sec	Resultant Lander C.G. Velocity
YLGRAV	51		Ft/Sec^2	Local Surface Component of the Acceleration of Gravity in Y Direction
ZLGRAV	63		Ft/Sec^2	Local Surface Component of the Acceleration of Gravity Z_{ls} Direction
XN	40		Planet g's	Load Factor X_{ls} Component
YN	52		Planet g's	Load Factor Y_{ls} Component
ZN	64		Planet g's	Load Factor Z_{ls} Component
FS (1)	41		Lb	Attenuator Footprint Frictional Force, X Component
FS(2)	53		Lb	Attenuator Footprint Frictional Force, Y Component
ANGLE3	65		Deg	Angle Between Friction Force Direction and Local Surface X_{ls} Axis.
ADDITIONAL VEHICLE COORDINATE REFERENCED TIME HISTORY PARAMETERS, 6 LINE FORMAT OPTION				
XPF (1)	42		Ft	Location on Crush Plane of Resultant Normal Crush Force (FONTT), X Coordinate
XPF (2)	54		Ft	Location on Crush Plane of FONTT, Y Coordinate.
XPF (3)	66		Ft	Location on Crush Plane of FONTT, Z Coordinate
FF (1)	43		Lb	Equivalent Applied Force at Lander C.G., X Component
FF (2)	55		Lb	Equivalent Applied Force at Lander C.G. Y Component
FF (3)	67		Lb	Equivalent Applied Force at Lander C.G., Z Component
FF (4)	44		Ft Lb	Equivalent Applied Moment, Component About X Axis.
FF (5)	56		Ft Lb	Equivalent Applied Moment, Component About Y Axis
FF (6)	68		Ft Lb	Equivalent Applied Moment, Component About Z Axis.
TFC (1)	45		Ft Lb	Frictional Torque Due to Pure Rotation of the Attenuator Footprint, Area, X Component
TFC (2)	57		Ft Lb	Footprint Frictional Torque, Y Component
TFC (3)	69		Ft Lb	Footprint Frictional Torque, Z Component
ADDITIONAL VEHICLE ORIENTATION TIME HISTORY PARAMETERS, 6 LINE FORMAT OPTION				
ANGLD	46		Deg	Dihedral Angle Between Lander Coordinate YZ Plane and Local Surface Coordinate XY Plane.
ANGLD2	58		Deg	Dihedral Angle Between Lander XY Plane and Plane Defined by Vehicle X Axis and Local Surface Z Axis (Maximum Attenuator Stroke Occurs in this Plane).

Figure B-8 (continued)

B.3.2.2 Output Format. - At the end of specified integration intervals, the various time history quantities of interest are printed. The print times are governed by the input indicator KOUNT2. Two formats are available in the output routine: three lines of output containing 36 variables (Option 1) or six lines containing 72 variables (Option 2). Option 1 prints the top three lines of output defined on Figure B-7, while Option 2 prints all six lines as shown in this figure. The output format is governed by the input quantity IERPRT.

IERPRT	=	0	Print Option 1
IERPRT	=	1	Print Option 2

Internally the subroutine PRINT uses the time history variables' program name and the subscript defining the variable's location in the common array COMINT to print the variable. It is important to note that only real variables in the common array COMINT can be printed.

When more than one cutoff variable (NCUT > 1) is employed with the variable step Predictor-Corrector integration routine, the time history data must be carefully evaluated. In this situation, if the integration error tolerance is exceeded and the integration step size reduced, the output over a portion of the previous real time is repeated. The last time history quantities listed for a specific real time (T) should be used since they are the most accurate. This does not occur when only one cutoff parameter is specified or with the Runge-Kutta or fixed step Predictor-Corrector integration routines. Program termination for number of bounces (NMBNCS) or payload clearance is separate and is not counted in NCUT.

Examples of the output format for the various options are shown in Section B.3.3.

B.3.3 EXAMPLE OF PROGRAM OPERATION - In order to further illustrate the operation of the program, the program setup for the following example case will be shown:

1. Toroidal aluminum trussgrid crushable attenuator (density 3.5 lb/ft^3).
2. Cylindrical payload (density 60 lb/ft^3).
3. Lander dimensions and initial orientation as shown in Figure B-9.
4. Surface gravitational acceleration of 12.3 ft/sec^2.
5. Slope of -34°.
6. Surface rock diameter of 5.0 inches.
7. Initial resultant velocity of 85 ft/sec.
8. Initial c.g. velocity vector in the gravity axes Y_f-Z_f plane, 30° from the positive Z_f axes in the direction of the Y_f axis (see Figure B-9).

The data will be set up to run two cases in the same run. The first case will be as described above and use the standard output parameters with the six line format. The integration will be variable step Predictor-Corrector. The second case will be the same as the first except that the initial center of gravity velocity vector will be in the gravity axes X_f-Z_f plane, the three line output format will be used, the local surface displacements, velocities, and accelerations will be replaced in the printout by their corresponding gravity axis values, and fixed step Predictor-Corrector

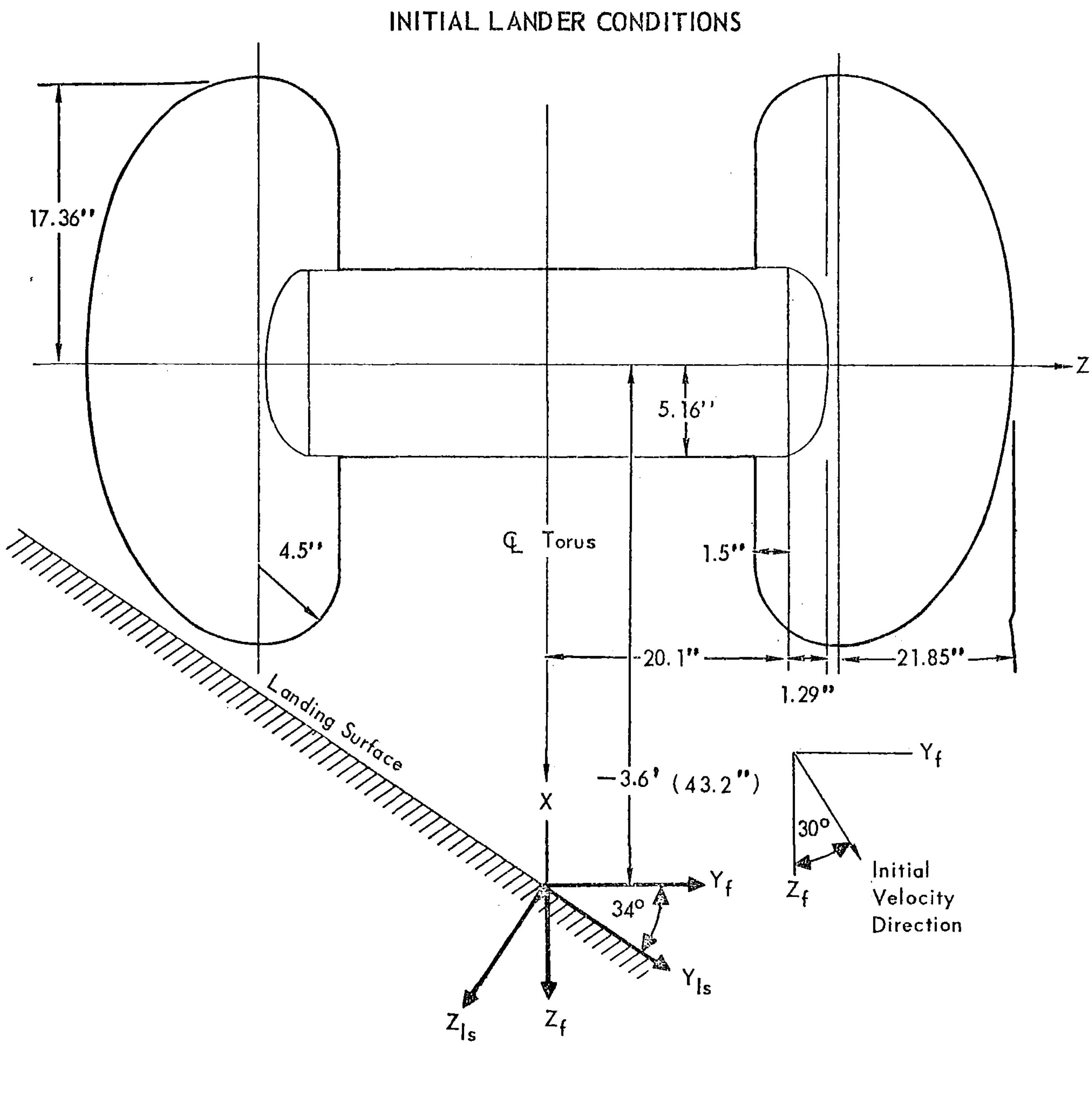

INITIAL LANDER CONDITIONS
17.36"
5.16"
4.5"
℄ Torus
1.5"
20.1"
1.29"
21.85"
Z
Landing Surface
X
−3.6' (43.2")
Y_f
30^o
Z_f
Initial
Velocity
Direction
Y_f
34^o
Y_{ls}
Z_{ls}
Z_f

Figure B-9

integration will be used. The cutoff variable for both cases will be real time (one second, increasing limit). The program will also be terminated after one bounce.

The input cards for this run are shown in Figure B-10, and program output, in the order in which it occurs, is shown in Figures B-11 through B-14.

Central processor computing time for one case involves two components; compilation time and running time. The compilation time is approximately 50 seconds. The running time varies considerably with the type of case run, especially with respect to the input incremental parameters DX and DTHE used in the LDSTR subroutines (larger values reduce running time). Other factors affecting running time are: impact attitude (flat landing cases run longer), the type integration used (variable step Predictor-Corrector fastest, Runge-Kutta slowest), and the number of impacts or bounces allowed (more bounces increase time). Nominal running times for one case are two to five minutes.

APPENDIX B

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

EXAMPLE INPUT DATA

```
EOD
ZDDF              584    27
ZDF               505    26
ZF                388    25
YDDF              583    15
YDF               504    14
YF                387    13
XDDF              582    3
XDF               503    2
XF                386    1
1   073    0.          42.5     0.        0.        -3.6      73.5
    013    0.          1.       0.
    007    0.          0.       0.        0.        0.        0.
    001    4.          1.       0.        .01       .001      0.
DATE
TOROIDAL ALUMINUM TRUSSGRID ATTENUATOR - CROSS SLOPE
SAMPLE CASE 2
EOD
1   127                                   9.3478E24  1.1074E7
    115    1.                   .4                  .3        1720.
    109    2.          .9       1.        5.        .0516     .04
    103    4.5         17.36    21.85     28.10     3.5       60.
    097                .8                 1.29      5.16      1.5
    085    0.          -34.
    079    -90.        0.       -90.      0.        0.        0.
    073    0.          0.       0.        42.5      -3.6      73.5
    067    0.          .004     2.        1.
    019    1.          0.
    013    0.          1.       1.
    007    0.          0.       0.        0.        0.        0.
    001    4.          0.       0.        .01       .001      0.
DATE
TOROIDAL ALUMINUM TRUSSGRID ATTENUATOR - DOWNSLOPE
SAMPLE CASE 1
```

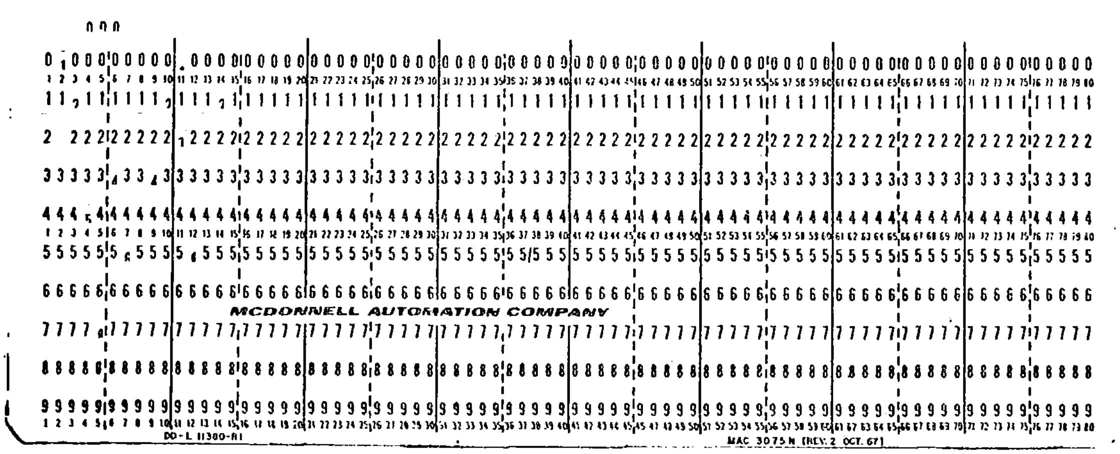


Figure B-10

APPENDIX B

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
Print Out of Input Data
(Example 1)

```
                    CRUSHABLE TORUS LOADS AND MOTIONS PROGRAM
             MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1
           MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION

                                  INPUT DATA

SAMPLE CASE 1
TOROIDAL ALUMINUM TRUSSGRID ATTENUATOR - DOWNSLOPE
DATE

DATA SUBSCRIPT      IP            IVARH         IMTH          EMAX          EMIN          HMIN
       1        4.0000000E+00 0.            0.            1.0000000E-02 1.0000000E-03 0.

DATA SUBSCRIPT      NTX           NTY           NTZ           NRX           NRY           NRZ
       7        0.            0.            0.            0.            0.            0.

DATA SUBSCRIPT      MMIC          NCUT          IERPRT
      13        0.            1.0000000E+00 1.0000000E+00-0.            -0.           -0.

DATA SUBSCRIPT      T             T
                    XS(I)         IND(I)
      19        1.0000000E+00 0.            -0.           -0.           -0.           -0.

DATA SUBSCRIPT      T             HMAX          KOUNT2        NMBNCS
      67        0.            4.0000000E-03 2.0000000E+00 1.0000000E+00-0.            -0.

DATA SUBSCRIPT      XF            XVF(1)        YF            XVF(2)        ZF            XVF(3)
      73        0.            0.            0.            4.2500000E+01-3.4000000E+00 7.3500000E+01

DATA SUBSCRIPT      PHI           PHID          THETA         THETAD        PSI           PSID
      79       -9.0000000E+01 0.           -9.0000000E+01 0.            0.            0.

DATA SUBSCRIPT      RANGE         SLOPE
      85        0.           -3.4000000E+01-0.            -0.           -0.           -0.

DATA SUBSCRIPT                    STREF         SCFWT         A             B             C
      97       -0.            8.0000000E-01-0.            1.2900000E+00 5.1600000E+00 1.5000000E+00

DATA SUBSCRIPT      D             E             F             G             RHOL          RHOP
     103        4.5000000E+00 1.7360000E+01 2.1850000E+01 2.0100000E+01 3.5000000E+00 6.0000000E+01

DATA SUBSCRIPT      POW           RSHCR         VMIN          RDIA          DX            DTHE
     109        2.0000000E+00 9.0000000E-01 1.0000000E+00 5.0000000E+00 5.1600000E-02 4.0000000E-02

DATA SUBSCRIPT      RNORM                       ELST                        AMU           XKW
     115        1.0000000E+00-0.            4.0000000E-01-0.            3.0000000E-01 1.7200000E+03

DATA SUBSCRIPT      HYST          PI            PA            PLTMAS        PLTRAD        GASCNT
     127       -0.           -0.           -0.            9.3478000E+24 1.1074000E+07-0.

EOD             END OF DATA LIST


 VEHICLE WEIGHT, ON EARTH =  7.7890674E+02, ON PLANET =  2.9803543E+02

                MOMENTS AND PRODUCTS OF INERTIA

        XXI               XYI               XXID              XYID
        YYI               XZI               YYID              XZID
        ZZI               YZI               ZZID              YZID

   8.6341990E+01  0.                0.                0.
   4.5923428E+01  0.                0.                0.
   4.5923428E+01  0.                0.                0.
```

Figure B-11

APPENDIX B .

CRUSHABLE TORUS LANDING LOAD AND MOTIONS PROGRAM

Typical Time History Output Page
Print Option 2
(Example 1)

LOCAL SURFACE COORDINATE SYSTEM					LANDER COORDINATE SYSTEM				VEHICLE ORIENTATION		
XLS	XDLS	XDDLS	FLT. ANGLE	FONTT	XD	XDD	PHID	PHIDD	ANG(1,1)	ANG(1,2)	ANG(1,3)
YLS	YDLS	YDDLS	VER. ANGLE	TORIN	YD	YDD	THETAD	THETADD	ANG(2,1)	ANG(2,2)	ANG(2,3)
ZLS	ZDLS	ZDDLS	STROKE	FRICT	ZD	ZDD	PSID	PSIDD	ANG(3,1)	ANG(3,2)	ANG(3,3)
RANGE	XCF(1)	I VELOCITY	XN	FS(1)	XPF(1)	FF(1)	FF(4)	TFC(1)	ANGLD	BLANK	BLANK
RCLR	XCF(2)	YLGRA	YN	FS(2)	XPF(2)	FF(2)	FF(5)	TFC(2)	ANGLD2	BLANK	BLANK
GZ	XCF(3)	ZLGRAV	ZN	ANGLD3	XPF(3)	FF(3)	FF(6)	TFC(3)	BLANK	BLANK	BLANK
CP TIME =	53.797 T	= 1.175E-02 DT		= 1.250E-04							
2.777E-15	3.159E-12	2.450E-09	9.000E+01	1.280E+04	7.292E+01	-5.575E+02	2.904E-12	2.216E-09	9.000E+01	4.071E-13	9.000E+01
-1.116E+00	7.621E+01	-1.140E+02	-5.965E+01	3.067E+05	-1.155E-11	2.839E-09	1.004E+00	7.696E+02	5.605E+01	9.000E+01	3.395E+01
-2.548E+00	3.600E+01	-5.940E+02	1.202E+00	3.840E+03	4.278E+01	2.369E+02	5.823E-13	4.147E-10	3.395E+01	9.000E+01	1.239E+02
8.970E-01	1.161E-11	6.454E+01	-1.639E+01	5.877E-08	9.985E-01	-1.276E+04	1.913E-07	0.	3.395E+01	0.	0.
1.001E+01	7.365E+01	6.076E+00	6.597E-11	-3.840E+03	-3.746E-12	6.696E-04	3.534E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	5.066E+00	2.700E+02	-3.079E+00	3.943E+03	1.904E-08	0.	0.	0.	0.
CP TIME =	54.308 T	= 1.200E-02 DT		= 1.250E-04							
3.644E-15	3.780E-12	2.629E-09	9.000E+01	1.371E+04	7.278E+01	-6.036E+02	3.471E-12	2.371E-09	9.000E+01	4.830E-06	9.000E+01
-1.097E+00	7.617E+01	-1.191E+02	-5.958E+01	3.287E+05	-1.092E-11	3.054E-09	1.203E+00	8.240E+02	5.607E+01	9.000E+01	3.393E+01
-2.539E+00	3.647E+01	-6.475E+02	1.305E+00	4.114E+03	4.284E+01	2.626E+02	6.879E-13	4.345E-10	3.393E+01	9.000E+01	1.239E+02
9.161E-01	1.364E-11	6.445E+01	-1.755E+01	6.296E-08	9.912E-01	-1.368E+04	2.047E-07	0.	3.393E+01	0.	0.
9.902E+00	7.312E+01	6.076E+00	9.211E-11	-4.114E+03	-3.741E-12	7.174E-08	3.784E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	5.446E+00	2.700E+02	-3.075E+00	4.242E+03	1.995E-08	0.	0.	0.	0.
CP TIME =	54.846 T	= 1.225E-02 DT		= 1.250E-04							
4.671E-15	4.453E-12	2.844E-09	9.000E+01	1.508E+04	7.262E+01	-6.691E+02	4.084E-12	2.691E-09	9.000E+01	4.830E-06	9.000E+01
-1.078E+00	7.613E+01	-1.204E+02	-5.951E+01	3.609E+05	-1.003E-11	3.367E-09	1.414E+00	9.041E+02	5.609E+01	9.000E+01	3.391E+01
-2.530E+00	3.632E+01	-7.202E+02	1.407E+00	4.524E+03	4.291E+01	2.952E+02	8.000E-13	4.747E-10	3.391E+01	9.000E+01	1.239E+02
9.351E-01	1.624E-11	6.435E+01	-1.931E+01	6.923E-08	9.864E-01	-1.504E+04	2.245E-07	0.	3.391E+01	0.	0.
9.798E+00	7.255E+01	6.076E+00	1.013E-10	-4.524E+03	-3.731E-12	7.889E-08	4.152E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	5.982E+00	2.700E+02	-3.067E+00	4.659E+03	2.171E-08	0.	0.	0.	0.
CP TIME =	55.323 T	= 1.250E-02 DT		= 1.250E-04							
5.075E-15	5.183E-12	3.049E-09	9.000E+01	1.607E+04	7.245E+01	-7.200E+02	4.748E-12	2.709E-09	9.000E+01	4.071E-13	9.000E+01
-1.059E+00	7.608E+01	-1.325E+02	-5.942E+01	3.851E+05	-9.163E-12	3.602E-09	1.647E+00	9.634E+02	5.611E+01	9.000E+01	3.389E+01
-2.571E+00	3.617E+01	-7.764E+02	1.506E+00	4.821E+03	4.299E+01	3.240E+02	9.194E-13	4.940E-10	3.389E+01	9.000E+01	1.239E+02
9.541E-01	1.808E-11	6.424E+01	-2.058E+01	7.378E-08	9.786E-01	-1.603E+04	2.391E-07	0.	3.389E+01	0.	0.
9.694E+00	7.193E+01	6.076E+00	1.080E-10	-4.821E+03	-3.727E-12	8.408E-08	4.424E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	6.367E+00	2.700E+02	-3.063E+00	4.959E+03	2.256E-08	0.	0.	0.	0.
CP TIME =	55.856 T	= 1.275E-02 DT		= 2.500E-04							
7.267E-15	5.958E-12	3.263E-09	9.000E+01	1.695E+04	7.226E+01	-7.669E+02	5.453E-12	2.914E-09	9.000E+01	4.071E-13	9.000E+01
-1.040E+00	7.603E+01	-1.346E+02	-5.934E+01	4.064E+05	-8.296E-12	3.813E-09	1.892E+00	1.015E+03	5.613E+01	9.000E+01	3.387E+01
-2.511E+00	3.600E+01	-8.331E+02	1.604E+00	5.064E+03	4.307E+01	3.523E+02	1.044E-12	4.902E-10	3.387E+01	9.000E+01	1.239E+02
9.731E-01	2.160E-11	6.413E+01	-2.170E+01	7.780E-08	9.717E-01	-1.690E+04	2.516E-07	0.	3.357E+01	0.	0.
9.592E+00	7.128E+01	6.076E+00	1.138E-10	-5.084E+03	-3.765E-12	8.868E-08	4.663E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	6.705E+00	2.700E+02	-3.059E+00	5.222E+03	2.251E-08	0.	0.	0.	0.
CP TIME =	57.367 T	= 1.325E-02 DT		= 2.500E-04							
1.066E-14	7.655E-12	3.713E-09	9.000E+01	1.923E+04	7.185E+01	-8.845E+02	6.991E-12	3.275E-09	9.000E+01	4.830E-06	9.000E+01
-1.002E+00	7.593E+01	-1.446E+02	-5.914E+01	4.607E+05	-6.221E-12	4.356E-09	2.428E+00	1.149E+03	5.619E+01	9.000E+01	3.381E+01
-2.494E+00	3.564E+01	-9.676E+02	1.794E+00	5.768E+03	4.326E+01	4.182E+02	1.301E-12	5.390E-10	3.381E+01	9.000E+01	1.238E+02
1.011E+00	2.753E-11	6.367E+01	-2.463E+01	8.826E-08	9.509E-01	-1.918E+04	2.845E-07	0.	3.381E+01	0.	0.
9.392E+00	6.987E+01	6.078E+00	1.292E-10	-5.768E+03	-3.750E-12	1.006E-07	5.277E+04	0.	-9.000E+01	0.	0.
1.230E+01	0.	1.020E+01	7.580E+00	2.700E+02	-3.047E+00	5.904E+03	2.475E-08	0.	0.	0.	0.

Figure B-12

APPENDIX B

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
Print Out of Input Data
(Example 2)

```
                    CRUSHABLE TORUS LOADS AND MOTIONS PROGRAM
              MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1
             MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION

                                INPUT DATA

 SAMPLE CASE 2
 TOROIDAL ALUMINUM TRUSSGRID ATTENUATOR - CROSS SLOPE
 DATE

 DATA SUBSCRIPT     IP             IVARH          IMIN         EMAX           EMIN           WMIN
        1       4.0000000E+00 1.0000000E+00 0.            1.0000000E-02 1.0000000E-03 0.

 DATA SUBSCRIPT     NTX            NTY            NTZ          NRX            NRY            NRZ
        7       0.             0.             0.           0.             0.             0.

 DATA SUBSCRIPT     MMIC           NCUT           IERPRT
       13       0.             1.0000000E+00 0.           -0.            -0.            -0.

 DATA SUBSCRIPT     XF             XVF(1)         YF           XVF(2)         ZF             XVF(3)
       73       0.             4.2500000E+01 0.           0.             -3.6000000E+00 7.3500000E+01

 PRINT POSITION    1 WILL CONTAIN THE VALUE FROM COMINT(  386) WITH THE LABEL XF
 PRINT POSITION    2 WILL CONTAIN THE VALUE FROM COMINT(  503) WITH THE LABEL XDF
 PRINT POSITION    3 WILL CONTAIN THE VALUE FROM COMINT(  582) WITH THE LABEL XDDF
 PRINT POSITION   13 WILL CONTAIN THE VALUE FROM COMINT(  387) WITH THE LABEL YF
 PRINT POSITION   14 WILL CONTAIN THE VALUE FROM COMINT(  504) WITH THE LABEL YDF
 PRINT POSITION   15 WILL CONTAIN THE VALUE FROM COMINT(  583) WITH THE LABEL YDDF
 PRINT POSITION   25 WILL CONTAIN THE VALUE FROM COMINT(  388) WITH THE LABEL ZF
 PRINT POSITION   26 WILL CONTAIN THE VALUE FROM COMINT(  505) WITH THE LABEL ZDF
 PRINT POSITION   27 WILL CONTAIN THE VALUE FROM COMINT(  584) WITH THE LABEL ZDDF
 EOD            END OF DATA LIST

 ---------------------

  VEHICLE WEIGHT, ON EARTH =  7.7890674E+02, ON PLANET =  2.9803543E+02
                  MOMENTS AND PRODUCTS OF INERTIA

        XXI              XYI            XXID           XYID
        YYI              XZI            YYID           XZID
        ZZI              YZI            ZZID           YZID

   8.6341990E+01  0.             0.             0.
   4.5923428E+01  0.             0.             0.
   4.5923428E+01  0.             0.             0.
```

Figure B-13

APPENDIX B

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
Typical Time History Output Page
Print Option 1
(Example 2)

LOCAL SURFACE COORDINATE SYSTEM					* LANDER COORDINATE SYSTEM				* VEHICLE ORIENTATION		
XF	XDF	XDDF	FLT. ANGLE	FONTT *	XD	XDD	PHID	PHIDD *	ANG(1,1)	ANG(1,2)	ANG(1,3)
YF	YDF	YDDF	VER. ANGLE	TORTN *	YD	YDD	THETAD	THETADD *	ANG(2,1)	ANG(2,2)	ANG(2,3)
ZF	ZDF	ZDDF	STROKE	FRICT *	ZD	ZDD	PSID	PSIDD *	ANG(3,1)	ANG(3,2)	ANG(3,3)
CP TIME = 88.425 T = 0. DT = 2.500E-04											
0.	4.250E+01	1.791E-19	-1.827E-20	0.	7.350E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
0.	-1.355E-20	2.351E-38	-5.996E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.600E+00	7.350E+01	1.230E+01	-6.018E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.532 T = 2.500E-04 DT = 2.500E-04											
1.062E-02	4.250E+01	-1.434E-36	-1.827E-20	0.	7.350E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.931E-23	-1.355E-20	0.	-5.996E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.582E+00	7.350E+01	1.230E+01	-3.835E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.613 T = 5.000E-04 DT = 2.500E-04											
2.125E-02	4.250E+01	-1.434E-36	-8.136E-21	0.	7.351E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.592E-23	-6.776E-21	0.	-5.996E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.563E+00	7.351E+01	1.230E+01	-3.652E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.693 T = 7.500E-04 DT = 2.500E-04											
3.187E-02	4.250E+01	-1.434E-36	-4.568E-21	0.	7.351E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.366E-23	-3.388E-21	0.	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.545E+00	7.351E+01	1.230E+01	-3.470E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.750 T = 1.000E-03 DT = 2.500E-04											
4.250E-02	4.250E+01	-1.434E-36	-2.284E-21	0.	7.351E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.282E-23	-1.694E-21	-1.175E-38	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.526E+00	7.351E+01	1.230E+01	-3.287E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.807 T = 1.250E-03 DT = 2.500E-04											
5.312E-02	4.250E+01	-1.434E-36	-1.142E-21	0.	7.352E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.244E-23	-8.472E-22	0.	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.508E+00	7.352E+01	1.230E+01	-3.104E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.864 T = 1.500E-03 DT = 2.500E-04											
5.375E-02	4.250E+01	-1.434E-36	-1.142E-21	0.	7.352E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
6.218E-23	-8.472E-22	0.	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.490E+00	7.352E+01	1.230E+01	-2.921E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.921 T = 1.750E-03 DT = 2.500E-04											
7.437E-02	4.250E+01	-1.434E-36	-5.712E-22	0.	7.352E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.195E-23	-4.237E-22	0.	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.471E+00	7.352E+01	1.230E+01	-2.738E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 88.977 T = 2.000E-03 DT = 2.500E-04											
8.500E-02	4.250E+01	-1.434E-36	-2.858E-22	0.	7.352E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.188E-23	-2.120E-22	-1.175E-38	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.453E+00	7.352E+01	1.230E+01	-2.555E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02
CP TIME = 89.034 T = 2.250E-03 DT = 2.500E-04											
9.562E-02	4.250E+01	-1.434E-36	-1.430E-22	0.	7.353E+01	1.230E+01	0.	0.	9.000E+01	4.071E-13	9.000E+01
4.183E-23	-1.061E-22	-1.175E-38	-5.997E+01	0.	4.250E+01	-6.010E-12	0.	0.	5.600E+01	9.000E+01	3.400E+01
-3.435E+00	7.353E+01	1.230E+01	-2.372E+00	0.	-2.076E-11	2.936E-24	0.	0.	3.400E+01	9.000E+01	1.240E+02

Figure B-14

APPENDIX B

B.4 PROGRAM DESCRIPTION

B.4.1 SUBROUTINES - The program consists of the main program and the subroutines SOLVE, AERO, ENVIR, PHYS, INERT, LDSTR, FORM, PCCUT, and PRINT.

The subroutines AERO and ENVIR are not used in the present program, but are provided for future growth and improvement in the program. The AERO subroutine would be used for calculation of aerodynamic forces on the vehicle, and the ENVIR subroutine would be used for environmental effects, such as local winds.

A brief summary of the subroutine and multiple entry names and functions (of those presently used) is given in Figure B-15.

B.4.1.1 Main Program. - The main program reads in the input data, initializes the parameters used in the program, and sets up the data used in the subroutines. It also initializes and updates the direction cosine matrices, calculates the vehicle accelerations to be integrated (using the equations of motion), and controls the number of degrees of freedom of motion as specified in the input data.

B.4.1.2 SOLVE Subroutine. - This is a matrix inversion subroutine used to solve the angular equations of motion for the highest derivative (vehicle accelerations) so that they can be integrated. It is also used to invert a matrix in the FORM subroutine.

B.4.1.3 PHYS Subroutine. - This subroutine has two entries, PHYS and PHYS1. The first entry is called only once per case and is used to initialize

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
COMPUTER PROGRAM ORGANIZATION

MAIN	Input, Initialization, Set Up Equations of Motion
SOLVE	Matrix Inversion
PHYS	Initialization, Mass Properties
INERT	Weight, Moments and Cross Products of Inertia
PHYS1	Calculation of Forces and Moments
LDSTR	Crushing Force and Resulting Torque
FORM	Friction Force, Equivalent Forces and Moments at Vehicle C.G.
PCCUT	Numerical Integration Subroutine
LOC	Cutoff Variable Definition and Storage
INUPD	Parameter List of Integrated Variables
SETUP	Initialization of PCCUT Subroutine
INTEG	Integrate Equations of Motion
CUT	Check Cutoff Limits
UPDAT	Update Variable Lists, Modify Integration Interval
PRINT	Output

Figure B-15

parameters and call the INERT (inertia) subroutine. The second entry is called in each integration time interval and is used to locate the crush plane (maximum attenuator stroke and angle from vehicle Y-Z plane to local surface) and then call subroutines LDSTR and FORM, in that order.

B.4.1.4 INERT Subroutine. - The mass and inertias for the equations of motion are supplied by the INERT subroutine. The subroutine uses the geometry defined in Figure B-1 and divides the vehicle into 8 sections to calculate the inertias. The individual sections weight, center of gravity, and inertias are calculated and then the sections are combined to determine the total weight, center of gravity, and inertias. This approach makes possible the addition or removal of sections during the running of the program. It is this capability that makes variable center of gravities and inertias possible. This capability is not presently used, since inertia changes are small and program running time would be increased.

B.4.1.5 LDSTR Subroutine. - The LDSTR subroutine supplies the resultant normal force caused by crushing the attenuator, and the location of this force and of the footprint centroid. The maximum stroke and the angle between the X-Z and X_{1_S}-Y_{1_S} planes are input to this subroutine from PHYS.

B4.1.6 FORM Subroutine. - The magnitude and direction of frictional force and moments and the total equivalent forces and moments along and about the vehicle axes are calculated in the FORM subroutine.

B.4.1.7 PCCUT Subroutine. - This is the numerical integration subroutine. It has six entries: LOC, for locating and storing program cutoff parameters;

INUPD, for setting up a list of the variables to be integrated; SETUP, for initializing the subroutine parameters; INTEG, for calculating the integrated values; CUT, for checking program cutoff parameters against their limit values; and UPDAT, for updating the integrated variable list with the newly integrated values and, in the variable step Predictor-Corrector, for checking calculation accuracy and modifying step size accordingly.

B.4.1.8 PRINT Subroutine. - This subroutine prints the output data.

B.4.2 FLOW DIAGRAM - The general flow of calculations through the program is shown in Figure B-16.

B.4.3 PROGRAM LISTING - A listing of the program is given on the following pages.

APPENDIX B

FLOW DIAGRAM
CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

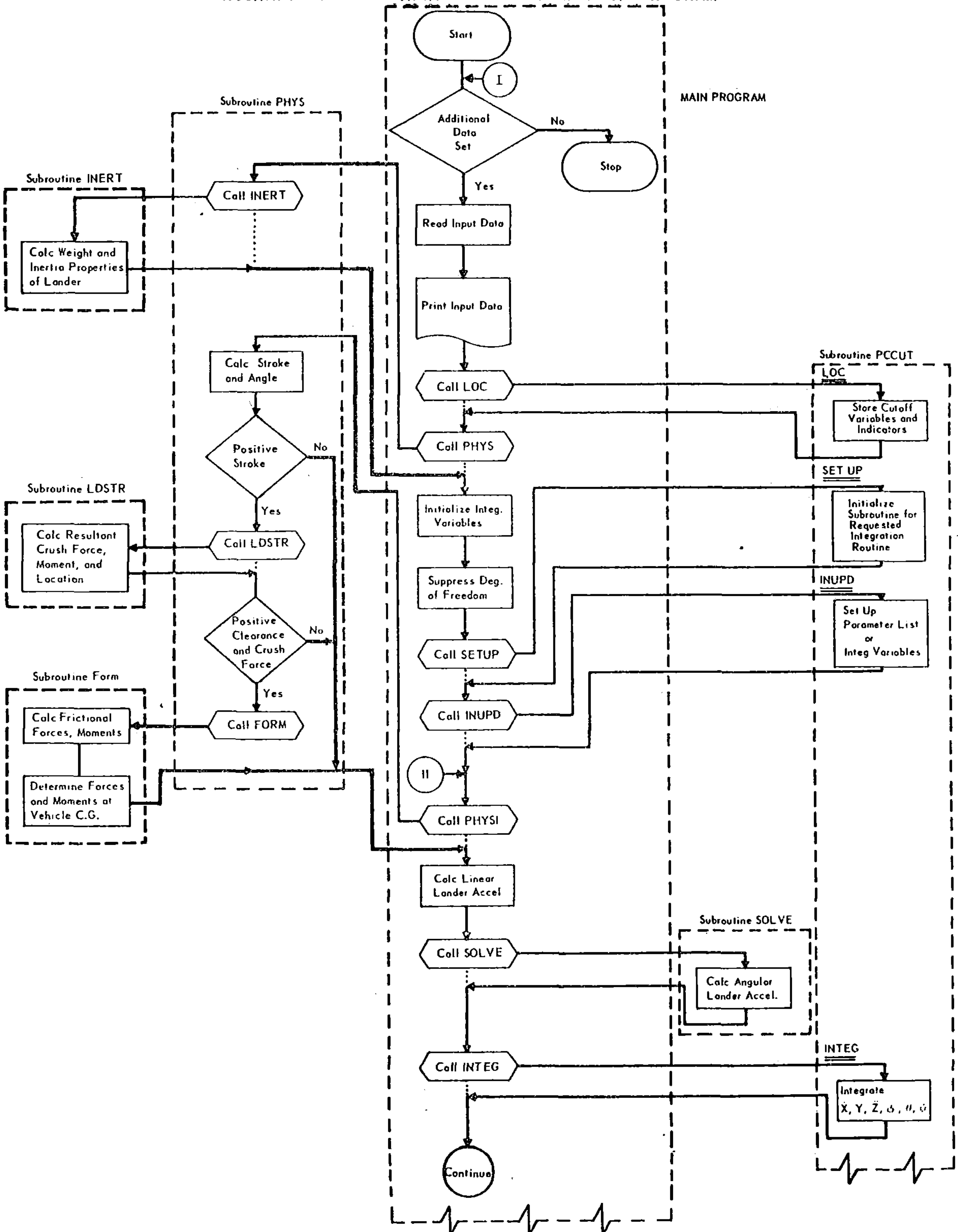


Figure B-16

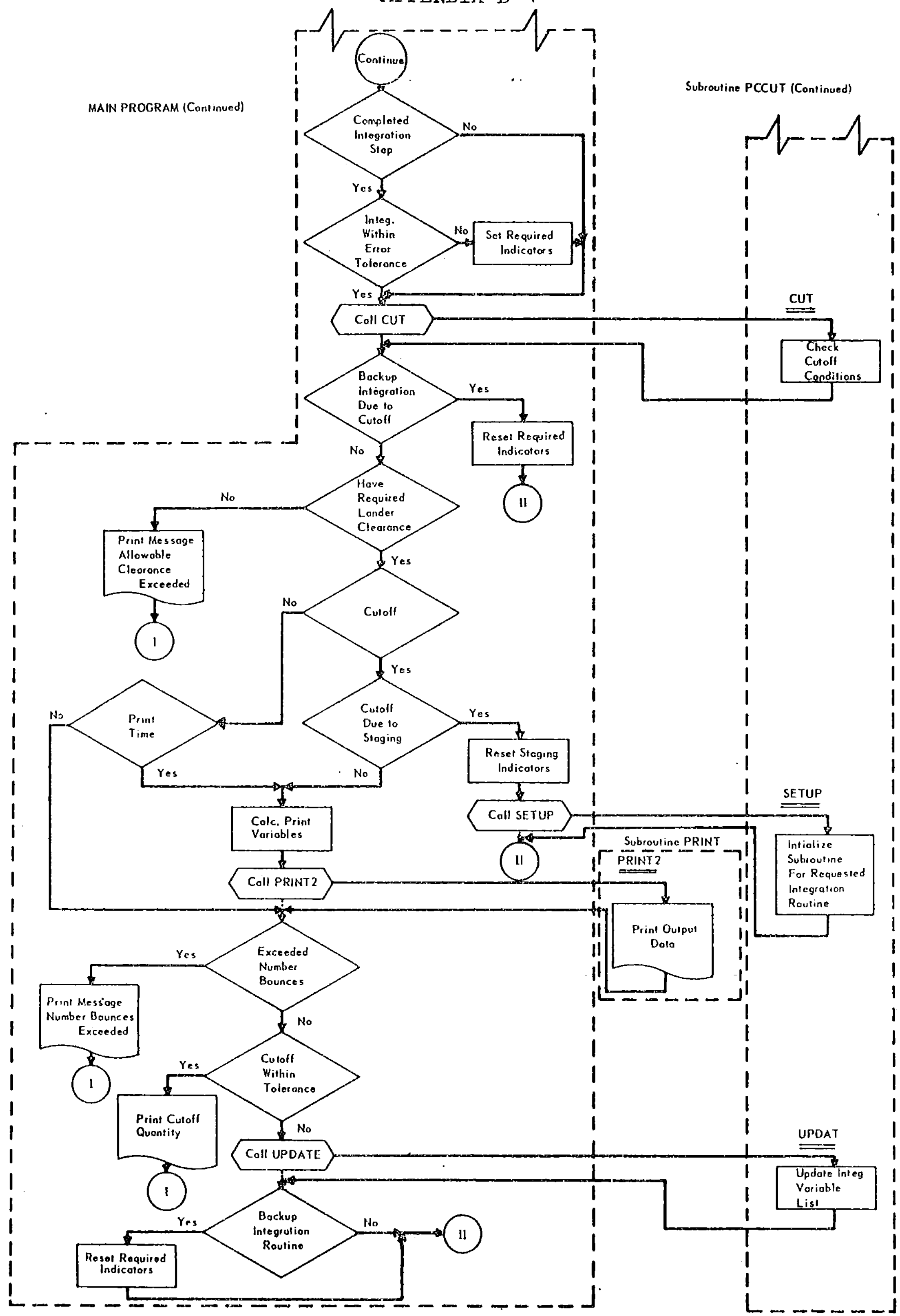


Figure B-16 (continued)

APPENDIX B

CRUSHABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION XCF(3)
      DIMENSION TFC(3)
      DIMENSION NDATA(300)
          DIMENSION XPF(3), XCD(3), FS(3)
      DIMENSION  ANG(3,3), XLPF(3)
      DIMENSION XS(8),IND(8),XDD(7,6),XD(7,19),X(7,19)
      DIMENSION STORE(3), GACC(3), AA(3,3), ASV(3,4), STOR(3,3), NSAV(8)
     1          ,  DC(19) ,       TEMP(3,3) , SUMA(3)
      DIMENSION XVF(3), TR(3,3), TRL(3,3), FF(19),DATA(300)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT(  1 ), T         )
      EQUIVALENCE ( COMINT(  2 ), HMAX      )
      EQUIVALENCE ( COMINT(  3 ), EMIN      )
      EQUIVALENCE ( COMINT(  4 ), EMAX      )
      EQUIVALENCE ( COMINT(  5 ), HZ        )
      EQUIVALENCE ( COMINT(  6 ), IP        )
      EQUIVALENCE ( COMINT(  7 ), IVARH     )
      EQUIVALENCE ( COMINT(  8 ), IMTH      )
      EQUIVALENCE ( COMINT(  9 ), IPRNT     )
      EQUIVALENCE ( COMINT( 10 ), IFIN      )
      EQUIVALENCE ( COMINT( 11 ), IVAL      )
      EQUIVALENCE ( COMINT( 12 ), IPTOTL    )
      EQUIVALENCE ( COMINT( 13 ), IPTATL    )
      EQUIVALENCE ( COMINT(  14), XS        )
      EQUIVALENCE ( COMINT(  23), IND       )
      EQUIVALENCE ( COMINT( 32 ), X         )
      EQUIVALENCE ( COMINT(165 ), XD        )
      EQUIVALENCE ( COMINT(298 ), XDD       )
      EQUIVALENCE ( COMINT( 340), TOTSV     )
      EQUIVALENCE ( COMINT(341), PSIL    )
      EQUIVALENCE ( COMINT( 342), TLSV      )
      EQUIVALENCE ( COMINT( 344), ZLS       )
      EQUIVALENCE ( COMINT( 345), PHIL      )
      EQUIVALENCE ( COMINT( 346), THETAL    )
      EQUIVALENCE ( COMINT( 347), HMIN      )
      EQUIVALENCE ( COMINT( 348), STROKE    )
      EQUIVALENCE ( COMINT( 349), SAFS      )
      EQUIVALENCE ( COMINT( 349), SPELST    )
      EQUIVALENCE ( COMINT( 356), SSSAVE    )
      EQUIVALENCE ( COMINT( 356), SPSTTM    )
      EQUIVALENCE ( COMINT( 363), SFSAVE    )
      EQUIVALENCE ( COMINT( 363), SAVST2    )
      EQUIVALENCE ( COMINT( 370), SVSAVE    )
      EQUIVALENCE ( COMINT( 370), SAVST1    )
      EQUIVALENCE ( COMINT( 377), SKODUP    )
      EQUIVALENCE ( COMINT( 384), CUTERR    )
      EQUIVALENCE ( COMINT( 385), JCUT      )
C         END OF VARIABLE ASSIGNMENT NEEDED FOR PCCUT
C               NOTICE XS AND IND ARE DIMENSIONIAL ARRAYS OF LENGTH 9
C               NOTICE ALL INTEGRATION AND TIME HISTORY VARIABLES MUST
C                     HAVE 7 COMMON LOCATIONS ALLOTED
      EQUIVALENCE ( COMINT( 386), XF        )
      EQUIVALENCE ( COMINT( 387), YF        )
      EQUIVALENCE ( COMINT( 388), ZF        )
      EQUIVALENCE ( COMINT( 389), XW        )
```

```
      EQUIVALENCE ( COMINT( 390), YW       )                            MAN 570
      EQUIVALENCE ( COMINT( 391), ZW       )                            MAN 580
      EQUIVALENCE ( COMINT( 392), SCFWT    )                            MAN 590
      EQUIVALENCE ( COMINT( 393), GZ       )                            MAN 600
      EQUIVALENCE ( COMINT( 394), XAG      )                            MAN 610
      EQUIVALENCE ( COMINT( 395), YAG      )                            MAN 620
      EQUIVALENCE ( COMINT( 396), ZAG      )                            MAN 630
      EQUIVALENCE ( COMINT( 397), DENS     )                            MAN 640
      EQUIVALENCE ( COMINT( 398), XWND     )                            MAN 650
      EQUIVALENCE ( COMINT( 399), YWND     )                            MAN 660
      EQUIVALENCE ( COMINT( 400), ZWND     )                            MAN 670
      EQUIVALENCE ( COMINT( 401), ALPHA    )                            MAN 680
      EQUIVALENCE ( COMINT( 402), GDOT     )                            MAN 690
      EQUIVALENCE ( COMINT( 403), SDOT     )                            MAN 700
      EQUIVALENCE ( COMINT( 404), BETA     )                            MAN 710
      EQUIVALENCE ( COMINT( 405), ALT      )                            MAN 720
      EQUIVALENCE ( COMINT( 406), RANGE    )                            MAN 730
      EQUIVALENCE ( COMINT( 407), THETA    )                            MAN 740
      EQUIVALENCE ( COMINT( 408), PSI      )                            MAN 750
      EQUIVALENCE ( COMINT( 409), PHI      )                            MAN 760
      EQUIVALENCE ( COMINT( 410), XXI      )                            MAN 770
      EQUIVALENCE ( COMINT( 411), YYI      )                            MAN 780
      EQUIVALENCE ( COMINT( 412), ZZI      )                            MAN 790
      EQUIVALENCE ( COMINT( 413), XN       )                            MAN 800
      EQUIVALENCE ( COMINT( 414), YN       )                            MAN 810
      EQUIVALENCE ( COMINT( 415), ZN       )                            MAN 820
      EQUIVALENCE ( COMINT( 416), XMASST   )                            MAN 830
      EQUIVALENCE ( COMINT( 417), THRST    )                            MAN 840
      EQUIVALENCE ( COMINT( 418), XYI      )                            MAN 850
      EQUIVALENCE ( COMINT( 419), XZI      )                            MAN 860
      EQUIVALENCE ( COMINT( 420), YZI      )                            MAN 870
      EQUIVALENCE ( COMINT( 421), DELTA    )                            MAN 880
      EQUIVALENCE ( COMINT( 422), XXID     )                            MAN 890
      EQUIVALENCE ( COMINT( 423), YYID     )                            MAN 900
      EQUIVALENCE ( COMINT( 424), ZZID     )                            MAN 910
      EQUIVALENCE ( COMINT( 425), XYID     )                            MAN 920
      EQUIVALENCE ( COMINT( 426), XZID     )                            MAN 930
      EQUIVALENCE ( COMINT( 427), YZID     )                            MAN 940
      EQUIVALENCE ( COMINT( 428), XLS      )                            MAN 950
      EQUIVALENCE ( COMINT( 429), YLS      )                            MAN 960
      EQUIVALENCE ( COMINT( 430), XDLS     )                            MAN 970
      EQUIVALENCE ( COMINT( 431), YDLS     )                            MAN 980
      EQUIVALENCE ( COMINT( 432), ZDLS     )                            MAN 990
      EQUIVALENCE ( COMINT( 433), XDDLS    )                            MAN1000
      EQUIVALENCE ( COMINT( 434), YDDLS    )                            MAN1010
      EQUIVALENCE ( COMINT( 435), ZDDLS    )                            MAN1020
      EQUIVALENCE ( COMINT( 436), XLGRAV   )                            MAN1030
      EQUIVALENCE ( COMINT( 437), YLGRAV   )                            MAN1040
      EQUIVALENCE ( COMINT( 438), ZLGRAV   )                            MAN1050
      EQUIVALENCE ( COMINT( 439), NTX      )                            MAN1060
      EQUIVALENCE ( COMINT( 440), NTY      )                            MAN1070
      EQUIVALENCE ( COMINT( 441), NTZ      )                            MAN1080
      EQUIVALENCE ( COMINT( 442), NRX      )                            MAN1090
      EQUIVALENCE ( COMINT( 443), NRY      )                            MAN1100
      EQUIVALENCE ( COMINT( 444), NRZ      )                            MAN1110
      EQUIVALENCE ( COMINT( 445), A        )                            MAN1120
```

```
      EQUIVALENCE ( COMINT( 446), B        )                              MAN1130
      EQUIVALENCE ( COMINT( 447), C        )                              MAN1140
      EQUIVALENCE ( COMINT( 448), D        )                              MAN1150
      EQUIVALENCE ( COMINT( 449), E        )                              MAN1160
      EQUIVALENCE ( COMINT( 450), F        )                              MAN1170
      EQUIVALENCE ( COMINT( 451), G        )                              MAN1180
      EQUIVALENCE ( COMINT( 452), RHOL     )                              MAN1190
      EQUIVALENCE ( COMINT( 453), RHOP     )                              MAN1200
      EQUIVALENCE ( COMINT( 454), TOTGV    )                              MAN1210
      EQUIVALENCE ( COMINT( 455), FONTT )                                 MAN1220
      EQUIVALENCE ( COMINT( 455), FORCE    )                              MAN1230
      EQUIVALENCE ( COMINT( 456), TORTN    )                              MAN1240
      EQUIVALENCE ( COMINT( 457), ARTSU    )                              MAN1250
      EQUIVALENCE ( COMINT( 458), PSTTM    )                              MAN1260
      EQUIVALENCE ( COMINT( 459), ANGLE    )                              MAN1270
      EQUIVALENCE ( COMINT( 460), ANGLE1   )                              MAN1280
      EQUIVALENCE ( COMINT( 461), ANGLE2   )                              MAN1290
      EQUIVALENCE ( COMINT( 462), ANGLE3   )                              MAN1300
      EQUIVALENCE ( COMINT( 463), ARMOM    )                              MAN1310
      EQUIVALENCE ( COMINT( 464), ELST     )                              MAN1320
      EQUIVALENCE ( COMINT( 465), RDIA     )                              MAN1330
      EQUIVALENCE ( COMINT( 466), AMU      )                              MAN1340
      EQUIVALENCE ( COMINT( 467), XKW      )                              MAN1350
      EQUIVALENCE ( COMINT( 468), DX       )                              MAN1360
      EQUIVALENCE ( COMINT( 469), DTHE     )                              MAN1370
      EQUIVALENCE ( COMINT( 470), ACLR     )                              MAN1380
      EQUIVALENCE ( COMINT( 471), RNORM    )                              MAN1390
      EQUIVALENCE ( COMINT( 472), LDSTR1   )                              MAN1400
      EQUIVALENCE ( COMINT(473),IERPRT)                                   MAN1410
      EQUIVALENCE ( COMINT( 474), RSXCF    )                              MAN1420
      EQUIVALENCE ( COMINT( 475), PELST    )                              MAN1430
      EQUIVALENCE ( COMINT( 476), SLOPE    )                              MAN1440
      EQUIVALENCE ( COMINT( 477), VOL      )                              MAN1450
      EQUIVALENCE ( COMINT( 478), R1       )                              MAN1460
      EQUIVALENCE ( COMINT( 479), R2       )                              MAN1470
      EQUIVALENCE ( COMINT( 480), R3       )                              MAN1480
      EQUIVALENCE ( COMINT( 481), THETAT   )                              MAN1490
      EQUIVALENCE ( COMINT( 482), RS       )                              MAN1500
      EQUIVALENCE ( COMINT( 483), RL       )                              MAN1510
      EQUIVALENCE ( COMINT( 484), FABWT    )                              MAN1520
      EQUIVALENCE ( COMINT( 485), PLY1     )                              MAN1530
      EQUIVALENCE ( COMINT( 486), PLY2     )                              MAN1540
      EQUIVALENCE ( COMINT( 487), PLY3     )                              MAN1550
      EQUIVALENCE ( COMINT( 488), AF       )                              MAN1560
      EQUIVALENCE ( COMINT( 489), HYST     )                              MAN1570
      EQUIVALENCE ( COMINT( 490), PI       )                              MAN1580
      EQUIVALENCE ( COMINT( 491), PA       )                              MAN1590
      EQUIVALENCE ( COMINT( 492), FRIRA    )                              MAN1600
      EQUIVALENCE ( COMINT( 493), DISTX    )                              MAN1610
      EQUIVALENCE ( COMINT( 494), DISTY    )                              MAN1620
      EQUIVALENCE ( COMINT( 495), PRESS    )                              MAN1630
      EQUIVALENCE ( COMINT( 496), AFS      )                              MAN1640
      EQUIVALENCE ( COMINT( 497), SSAVE    )                              MAN1650
      EQUIVALENCE ( COMINT( 498), FSAVE    )                              MAN1660
      EQUIVALENCE ( COMINT( 499), VSAVE    )                              MAN1670
      EQUIVALENCE ( COMINT( 500), KODEUP   )                              MAN1680
```

```
      EQUIVALENCE ( COMINT( 501), GASCNT )
      EQUIVALENCE ( COMINT( 502), IFLAT  )
      EQUIVALENCE ( COMINT( 503), XVF    )
C         DIMENSIONED XVF(3)
      EQUIVALENCE ( COMINT( 506), TR     )
C         DIMENSIONED TR(3,3)
      EQUIVALENCE ( COMINT( 515), TRL    )
C         DIMENSIONED TRL(3,3)
      EQUIVALENCE ( COMINT( 524), FF     )
C         DIMENSIONED FF(19)
      EQUIVALENCE ( COMINT( 543), XPF    )
C         DIMENSIONED XPF(3)
      EQUIVALENCE ( COMINT( 546), XCD    )
C         DIMENSIONED XCD(3)
      EQUIVALENCE ( COMINT( 549), FS     )
C         DIMENSION FS(3)
      EQUIVALENCE ( COMINT( 552), Q      )
      EQUIVALENCE ( COMINT( 553), XMACH  )
      EQUIVALENCE ( COMINT( 554), SOUND  )
      EQUIVALENCE ( COMINT( 555), FIFLAT )
      EQUIVALENCE ( COMINT( 556), MMIC   )
      EQUIVALENCE ( COMINT( 557), SIGMA  )
      EQUIVALENCE ( COMINT( 558), GAMA   )
      EQUIVALENCE ( COMINT( 559), SAVSTB )
      EQUIVALENCE ( COMINT( 594), POW    )
      EQUIVALENCE ( COMINT( 595), RSHCR  )
      EQUIVALENCE ( COMINT( 560), SAVSTA )
      EQUIVALENCE ( COMINT( 561), DEGRAD )
      EQUIVALENCE ( COMINT( 562), ANGLD1 )
      EQUIVALENCE ( COMINT( 563), ANGLD2 )
      EQUIVALENCE ( COMINT( 564), ANGLD3 )
      EQUIVALENCE ( COMINT( 565), PLTMAS )
      EQUIVALENCE ( COMINT( 566), PLTRAD )
      EQUIVALENCE ( COMINT( 567), NMBNCS )
      EQUIVALENCE ( COMINT( 568), KOUNT2 )
      EQUIVALENCE ( COMINT( 569), XLPF   )
C         DIMENSIONED XLPF(3)
      EQUIVALENCE ( COMINT( 572), ANG    )
C         DIMENSIONED ANG(3,3)
      EQUIVALENCE ( COMINT( 581), VMIN   )
      EQUIVALENCE ( COMINT( 582), GACC   )
C         DIMENSIONED GACC(3)
      EQUIVALENCE ( COMINT( 585), STREF  )
      EQUIVALENCE ( COMINT( 586), TFC    )
C         DIMENSIONED TFC(3)
      EQUIVALENCE ( COMINT( 589), RAV    )
      EQUIVALENCE ( COMINT( 590),FRICT   )
      EQUIVALENCE ( COMINT( 591), XCF    )
C         DIMENSIONED XCF(3)
      DATA EOD / 10HEOD          /
      DATA  DATA    / 300*0.0 /
      DATA (NDATA(I),I=1,90) /
     1 6HIP     , 6HIVARH , 6HIMTH   , 6HEMAX   , 6HEMIN   , 6HHMIN   ,
     2 6HNTX    , 6HNTY   , 6HNTZ    , 6HNRX    , 6HNRY    , 6HNRZ    ,
     3 6HMMIC   , 6HNCUT  , 6HIERPRT, 6H       , 6H       , 6H       ,
     4 6HT      , 6HT     , 6H       , 6H       , 6H       , 6H       ,
```

```
     5 6HTOTSV , 6HTOTSV , 6H       , 6H       , 6H       , 6H       ,
     6 6HXD(,19, 6HXD(,19, 6H       , 6H       , 6H       , 6H       ,
     7 6HTLSV  , 6HTLSV  , 6H       , 6H       , 6H       , 6H       ,
     8 6HZLS   , 6HZLS   , 6H       , 6H       , 6H       , 6H       ,
     9 6HPHIL  , 6HPHIL  , 6H       , 6H       , 6H       , 6H       ,
     $ 6HTHETAL, 6HTHETAL, 6H       , 6H       , 6H       , 6H       ,
     $ 6HPSIL  , 6HPSIL  , 6H       , 6H       , 6H       , 6H       ,
     $ 6HT     , 6HHMAX  , 6HKOUNT2, 6HNMBNCS, 6H       , 6H       ,
     $ 6HXF    , 6HXVF(1), 6HYF     , 6HXVF(2), 6HZF     , 6HXVF(3),
     $ 6HPHI   , 6HPHID  , 6HTHETA  , 6HTHETAD, 6HPSI    , 6HPSID   ,
     $ 6HRANGE , 6HSLOPE , 6H       , 6H       , 6H       , 6H       /
      DATA (NDATA(I),I=91,132) /
     1 6H      , 6H      , 6H       , 6H       , 6H       , 6H       ,
     2 6H      , 6HSTREF , 6HSCFWT  , 6HA      , 6H B     , 6HC      ,
     3 6HD     , 6HE     , 6HF      , 6HG      , 6HRHOL   , 6HRHOP   ,
     4 6HPOW   , 6HRSHCR , 6HVMIN   , 6HRDIA   , 6HDX     , 6HDTHE   ,
     5 6HRNORM , 6H      , 6HELST   , 6H       , 6HAMU    , 6HXKW    ,
     6 6HFABWT , 6HPLY1  , 6HPLY2   , 6HPLY3   , 6HRL     , 6HRS     ,
     7 6HHYST  , 6HPI    , 6HPA     , 6HPLTMAS, 6HPLTRAD, 6HGASCNT /
C     ALL DATA READ AND MANUPULATION PLUS INITIALIZATION AND DEFINED
C     INITIAL COND. HERE
      DO 10 I=1,8
10    NSAV(I)=0
      KXPN=8
20    WRITE (6,1190)
      DO 30 I=1,600
30    COMINT(I)=0.0
      NN=0
      IF (KXPN) 40,70,40
40    DO 60 I=1,KXPN
      IF (NSAV(I)) 50,60,50
50    NN=NN+1
60    CONTINUE
70    NGO=0
      DO 90 I=1,3
C          LEAVE COLUMN ONE BLANK
      READ (5,1200)
      IF (EOF,5) 80,90
80    WRITE (6,1210)
      STOP
90    WRITE (6,1200)
100   READ (5,1220) NSTP,NLOC,ZD1,ZD2,ZD3,ZD4,ZD5,ZD6
      NLOC6=NLOC+5
      WRITE (6,1230) (NDATA(I),I=NLOC,NLOC6)
      IF ((NLOC-13.LE.0).OR.(NLOC-61.GT.0)) GO TO 120
      WRITE (6,1240)
      DO 110 I=1,8
      IF (NLOC.EQ.NSAV(I)) GO TO 120
110   CONTINUE
      NN=NN+1
      NSAV(NN)=NLOC
120   WRITE (6,1250) NLOC,ZD1,ZD2,ZD3,ZD4,ZD5,ZD6
      DATA(NLOC)=ZD1
      DATA(NLOC+1)=ZD2
      DATA(NLOC+2)=ZD3
      DATA(NLOC+3)=ZD4
```

```
      DATA(NLOC+4)=ZD5                                                  MAN2810
      DATA(NLOC+5)=ZD6                                                  MAN2820
      IF (NSTP.EQ.0) GO TO 100                                          MAN2830
      NCUT=DATA(14)+.1                                                  MAN2840
      IF (NCUT.NE.0) GO TO 130                                          MAN2850
      WRITE (6,1260)                                                    MAN2860
      STOP                                                              MAN2870
C          INPUT  CHANGE OF PRINT VARIABLES                             MAN2880
C                ANY VARIABLE IN COMMON CAN BE PRINTED BY CARD OPTION   MAN2890
130   READ (5,1270) VNAME,LCOMON,LPRINT                                 MAN2900
C          COLUMN ONE OF DATA CARD IS USED FOR SPACING CONTROL          MAN2910
      IF (VNAME.EQ.EOD) GO TO 140                                       MAN2920
      WRITE (6,1280) LPRINT,LCOMON,VNAME                                MAN2930
      CALL PRINT (VNAME,LCOMON,LPRINT)                                  MAN2940
      GO TO 130                                                         MAN2950
140   WRITE (6,1290) VNAME                                              MAN2960
      IP=DATA(1)+.1                                                     MAN2970
      HMAX=DATA(68)                                                     MAN2980
      HMIN=DATA(6)                                                      MAN2990
      IF (HMIN.EQ.0.0) HMIN=HMAX*2.**(-16)                              MAN3000
      DEGRAD=57.2957795131                                              MAN3010
      IPTATL=0                                                          MAN3020
      IPTOTL=0                                                          MAN3030
      INDACL=0                                                          MAN3040
      CUTERR=.0001                                                      MAN3050
      NCONST=1                                                          MAN3060
      INDBAC=0                                                          MAN3070
      ACLR=0.0                                                          MAN3080
      LDSTR1=0                                                          MAN3090
      IERPRT=DATA(15)                                                   MAN3100
      NMBNCS=DATA(70)                                                   MAN3110
      IF (NMBNCS.EQ.0) NMBNCS=100                                       MAN3120
      NMBNC=0                                                           MAN3130
      STREF=DATA(98)                                                    MAN3140
      SCFWT=DATA(99)                                                    MAN3150
      A=DATA(100)                                                       MAN3160
      B=DATA(101)                                                       MAN3170
      C=DATA(102)                                                       MAN3180
      D=DATA(103)                                                       MAN3190
      E=DATA(104)                                                       MAN3200
      F=DATA(105)                                                       MAN3210
      G=DATA(106)                                                       MAN3220
      RHOL=DATA(107)                                                    MAN3230
      RHOP=DATA(108)                                                    MAN3240
      IF (DATA(111).NE.0) GO TO 150                                     MAN3250
      WRITE (6,1300)                                                    MAN3260
      STOP                                                              MAN3270
150   VMIN=DATA(111)                                                    MAN3280
      POW=DATA(109)                                                     MAN3290
      RSHCR=DATA(110)                                                   MAN3300
      RDIA=DATA(112)                                                    MAN3310
      DX=DATA(113)                                                      MAN3320
      DTHE=DATA(114)                                                    MAN3330
      IF (DTHE.LE.0.0) DTHE=.03                                         MAN3340
      RNORM=DATA(115)                                                   MAN3350
      ELST=DATA(117)                                                    MAN3360
```

```
      AMU=DATA(119)                                                     MAN3370
      XKW=DATA(120)                                                     MAN3380
C     INFLATABLE DATA ITEMS                                             MAN3390
      FABWT=DATA(121)                                                   MAN3400
      PLY1=DATA(122)                                                    MAN3410
      PLY2=DATA(123)                                                    MAN3420
      PLY3=DATA(124)                                                    MAN3430
      RL=DATA(125)                                                      MAN3440
      RS=DATA(126)                                                      MAN3450
      HYST=DATA(127)                                                    MAN3460
      PI=DATA(128)                                                      MAN3470
      PA=DATA(129)                                                      MAN3480
      PLTMAS=DATA(130)                                                  MAN3490
      PLTRAD=DATA(131)                                                  MAN3500
      GASCNT=DATA(132)                                                  MAN3510
      IVARH=DATA(2)+.1                                                  MAN3520
      IMTH=DATA(3)+.1                                                   MAN3530
      EMAX=DATA(4)                                                      MAN3540
      EMIN=DATA(5)                                                      MAN3550
      IF (EMAX.EQ.0.0) EMAX=1.E-4                                       MAN3560
      IF (EMIN.EQ.0.0) EMIN=1.E-6                                       MAN3570
      NTX=DATA(7)+.1                                                    MAN3580
      NTY=DATA(8)+.1                                                    MAN3590
      NTZ=DATA(9)+.1                                                    MAN3600
      NRX=DATA(10)+.1                                                   MAN3610
      NRY=DATA(11)+.1                                                   MAN3620
      NRZ=DATA(12)+.1                                                   MAN3630
      MMIC=DATA(13)+.1                                                  MAN3640
      T=DATA(67)                                                        MAN3650
      KOUNT2=DATA(69)+.1                                                MAN3660
      KOUNT1=KOUNT2-1                                                   MAN3670
      XF=DATA(73)                                                       MAN3680
      XVF(1)=DATA(74)                                                   MAN3690
      YF=DATA(75)                                                       MAN3700
      XVF(2)=DATA(76)                                                   MAN3710
      ZF=DATA(77)                                                       MAN3720
      XVF(3)=DATA(78)                                                   MAN3730
      RANGE=DATA(85)                                                    MAN3740
      SLOPE=DATA(86)                                                    MAN3750
      PHI=DATA(79)                                                      MAN3760
      PHIP=PHI                                                          MAN3770
      XD(1,4)=DATA(80)                                                  MAN3780
      THETA=DATA(81)                                                    MAN3790
      XD(1,5)=DATA(82)                                                  MAN3800
      PSI=DATA(83)                                                      MAN3810
      PSIP=PSI                                                          MAN3820
      XD(1,6)=DATA(84)                                                  MAN3830
C         DEFINE CUTOFF VARIABLES BY CALL TO LOC                        MAN3840
C               THE FIRST EIGHT CUTOFF VARIABLES ARE DEFINED BY INPUT   MAN3850
C               STROKE IS USED FOR STAGING AT TOUCH DOWN  AND LIFT OFF  MAN3860
      DO 250 J=1,8                                                      MAN3870
      IF (NSAV(J)) 160,250,160                                          MAN3880
160   LOCA=(NSAV(J)-13)/6                                               MAN3890
      GO TO (170,180,190,200,210,220,230,240), LOCA                     MAN3900
170   CALL LOC (1,6HT     ,DATA(19),DATA(20))                           MAN3910
      GO TO 250                                                         MAN3920
```

```
180   CALL LOC (340,6HTOTSV ,DATA(25),DATA(26))
      GO TO 250
190   CALL LOC (291,6HXD(19 ,DATA(31),DATA(32))
      GO TO 250
200   CALL LOC (342,6HTLSV  ,DATA(37),DATA(38))
      GO TO 250
210   CALL LOC (344,6HZLS   ,DATA(43),DATA(44))
      GO TO 250
220   CALL LOC (186,6HPHID  ,DATA(49),DATA(50))
      GO TO 250
230   CALL LOC (193,6HTHETAD,DATA(55),DATA(56))
      GO TO 250
240   CALL LOC (200,6HPSID  ,DATA(61),DATA(62))
250   CONTINUE
      CALL LOC (348,6HSTROKE,0.0,0)
      LOCSTR=IPTATL
      KXPN=I
C     IF IV=1CONSTANT INTEGRATION INTEGRATION, IV=0 VARIBLE INTERVAL
      GZ=16.137E-11*PLTMAS/(-ZF+PLTRAD)**2
      CALL PRINT1
      CALL PHYS
      WTERTH=XMASST*32.147
      WTPLNT=XMASST*GZ
      WRITE (6,1310) WTERTH,WTPLNT
      WRITE (6,1320)
      WRITE (6,1330) XXI,XYI,XXID,XYID,YYI,XZI,YYID,XZID,ZZI,YZI,ZZID,YZ
     1ID
      CALL AERO
      CALL ENVIR
      X(1,4)=PHI/DEGRAD
      X(1,5)=THETA/DEGRAD
      X(1,6)=PSI/DEGRAD
      CS=COS(X(1,6))
      CT=COS(X(1,5))
      CP=COS(X(1,4))
      SP=SIN(X(1,4))
      ST=SIN(X(1,5))
      SS=SIN(X(1,6))
      CSLP=COS(SLOPE/DEGRAD)
      SSLP=SIN(SLOPE/DEGRAD)
      IF (ABS(X(1,4)-1.5707963268).GE.1.E-8) GO TO 260
      CP=0.0
260   IF (ABS(X(1,5)-1.5707963268).GE.1.E-8) GO TO 270
      CT=0.0
270   IF (ABS(X(1,6)-1.5707963268).GE.1.E-8) GO TO 280
      CS=0.0
280   CONTINUE
      TR(1,1)=CS*CT
      TR(2,1)=CT*SS
      TR(3,1)=-ST
      TR(1,2)=-SS*CP+CS*ST*SP
      TR(2,2)=CS*CP+SS*SP*ST
      TR(3,2)=CT*SP
      TR(1,3)=CS*ST*CP+SS*SP
      TR(2,3)=SS*ST*CP-CS*SP
      TR(3,3)=CT*CP
```

```
      TRL(1,1)=TR(1,1)
      TRL(1,2)=TR(1,2)
      TRL(1,3)=TR(1,3)
      TRL(2,1)=TR(2,1)*CSLP-TR(3,1)*SSLP
      TRL(2,2)=TR(2,2)*CSLP-TR(3,2)*SSLP
      TRL(2,3)=TR(2,3)*CSLP-TR(3,3)*SSLP
      TRL(3,1)=TR(2,1)*SSLP+TR(3,1)*CSLP
      TRL(3,2)=TR(2,2)*SSLP+TR(3,2)*CSLP
      TRL(3,3)=TR(2,3)*SSLP+TR(3,3)*CSLP
      XLS=XF
      YLS=YF*CSLP-ZF*SSLP
      ZLS=YF*SSLP+ZF*CSLP
      XDLS=XVF(1)
      YDLS=XVF(2)*CSLP-XVF(3)*SSLP
      ZDLS=XVF(2)*SSLP+XVF(3)*CSLP
      XLGRAV=0.
      YLGRAV=-GZ*SSLP
      ZLGRAV=GZ*CSLP
      DO 300 I=1,3
      X(1,I)=0.
      SUMAA=0.
      DO 290 J=1,3
  290 SUMAA=SUMAA+TR(J,I)*XVF(J)
  300 XD(1,I)=SUMAA
      X(1,7)=TR(1,1)
      X(1,8)=TR(2,1)
      X(1,9)=TR(3,1)
      X(1,10)=TR(1,2)
      X(1,11)=TR(2,2)
      X(1,12)=TR(3,2)
      X(1,13)=TR(1,3)
      X(1,14)=TR(2,3)
      X(1,15)=TR(3,3)
      X(1,16)=XF
      X(1,17)=YF
      X(1,18)=ZF
      X(1,19)=RANGE
      IF (NTX.NE.0) XD(1,1)=0.0
      IF (NTY.NE.0) XD(1,2)=0.0
      IF (NTZ.NE.0) XD(1,3)=0.0
      IF (NRX.NE.0) XD(1,4)=0.0
      IF (NRY.NE.0) XD(1,5)=0.0
      IF (NRZ.NE.0) XD(1,6)=0.0
      ALT=-ZF
      XNUMB=TRL(3,1)
      IF (ABS(XNUMB)-1.) 320,320,310
  310 XNUMB=XNUMB/ABS(XNUMB)
  320 THETAL=-ASIN(XNUMB)*DEGRAD
      IF (ABS(TRL(2,1))+ABS(TRL(1,1))+ABS(SLOPE).NE.0.0) GO TO 330
      PSIL=PSI
      PHIL=PHI
      GO TO 340
  330 PSIL=ATAN2(TRL(2,1),TRL(1,1))*DEGRAD
      PHIL=ATAN2(TRL(3,2),TRL(3,3))*DEGRAD
  340 TOTSV=SQRT(XDLS**2+YDLS**2+ZDLS**2)
      TOTGV=SQRT(XVF(1)**2+XVF(2)**2+XVF(3)**2)
```

```
      CON1=TOTGV+1.E-10
      GAMA=ASIN(-XVF(3)/CON1)*DEGRAD
      IF (XVF(2)) 370,350,370
350   IF (XVF(1)) 370,360,370
360   SIGMA=0.
      GO TO 380
370   SIGMA=ATAN2(XVF(2),XVF(1))*DEGRAD
380   CALL SETUP
      NN=6
      DO 400 I=1,6
      NN=NN+1
      IF (DATA(NN)) 400,390,400
390   IRLM=(I-1)*7
      CALL INUPD (165+IRLM)
      CALL INUPD (298+IRLM)
400   CONTINUE
      DO 410 I=7,19
      IRLM=(I-1)*7
      CALL INUPD (32+IRLM)
      CALL INUPD (165+IRLM)
410   CONTINUE
C         PLACE  TIME HISTORY VARIABLES IN INTEGRATION PARAMETER LIST
C                PURPOSE TO CONTROL VARIABLES WHEN INTEGRATION BACKS UP
      CALL INUPD (349)
      CALL INUPD (356)
      CALL INUPD (363)
      CALL INUPD (370)
C
C
C     CALCULATE ALL HIGHEST DERIVATIVES FOR EACH EQUATION
420   DO 430 I=1,6
430   FF(I)=0.0
      GZ=16.137E-11*PLTMAS/(-ZF+PLTRAD)**2
      XAG=X(1,9)*GZ
      YAG=X(1,12)*GZ
      ZAG=X(1,15)*GZ
      CALL PHYS1
      CALL AERO1
      XN=FF(1)/(XMASST*32.147)
      YN=FF(2)/(XMASST*32.147)
      ZN=FF(3)/(XMASST*32.147)
      XDD(1,1)=FF(1)/XMASST+XD(1,2)*XD(1,6)-XD(1,3)*XD(1,5)+XAG
      XDD(1,2)=FF(2)/XMASST-XD(1,1)*XD(1,6)+XD(1,3)*XD(1,4)+YAG
      XDD(1,3)=FF(3)/XMASST+XD(1,1)*XD(1,5)-XD(1,4)*XD(1,2)+ZAG
      GO TO (440,490), NCONST
440   IF (MMIC) 460,450,460
450   NCONST=2
      GO TO 470
460   TEMP(1,1)=XXID
      TEMP(1,2)=-XYID
      TEMP(1,3)=-XZID
      TEMP(2,1)=-XYID
      TEMP(2,2)=YYID
      TEMP(2,3)=-YZID
      TEMP(3,1)=-XZID
      TEMP(3,2)=-YZID
```

```
      TEMP(3,3)=ZZID
470   AA(1,1)=XXI
      AA(1,2)=-XYI
      AA(1,3)=-XZI
      AA(2,1)=-XYI
      AA(2,2)=YYI
      AA(2,3)=-YZI
      AA(3,1)=-XZI
      AA(3,2)=-YZI
      AA(3,3)=ZZI
      DO 480 I=1,3
      DO 480 J=1,3
480   ASV(I,J)=AA(I,J)
      CALL SOLVE (ASV,STOR,SUM1,3,0)
490   XDD(1,4)=ASV(1,1)*FF(4)+ASV(1,2)*FF(5)+ASV(1,3)*FF(6)
      XDD(1,5)=ASV(2,1)*FF(4)+ASV(2,2)*FF(5)+ASV(2,3)*FF(6)
      XDD(1,6)=ASV(3,1)*FF(4)+ASV(3,2)*FF(5)+ASV(3,3)*FF(6)
      IF (MMIC) 500,540,500
500   DO 520 I=1,3
      DO 510 J=1,3
      SUMA(J)=0.
      DO 510 K=1,3
510   SUMA(J)=SUMA(J)+ASV(J,K)*TEMP(K,I)
      DO 520 L=1,3
520   TEMP(L,I)=SUMA(L)
      N=3
      DO 530 I=1,3
      N=N+1
      DO 530 J=1,3
530   XDD(1,N)=XDD(1,N)-TEMP(I,J)*XD(1,J+3)
540   DO 550 I=1,3
550   TEMP(I,I)=0.
      TEMP(1,2)=-XD(1,6)
      TEMP(1,3)=XD(1,5)
      TEMP(2,1)=XD(1,6)
      TEMP(2,3)=-XD(1,4)
      TEMP(3,1)=-XD(1,5)
      TEMP(3,2)=XD(1,4)
      DO 570 I=1,3
      DO 560 J=1,3
      SUMA(J)=0.
      DO 560 K=1,3
560   SUMA(J)=SUMA(J)+ASV(J,K)*TEMP(K,I)
      DO 570 L=1,3
570   TEMP(L,I)=SUMA(L)
      DO 590 I=1,3
      DO 580 J=1,3
      SUMA(J)=0.
      DO 580 K=1,3
580   SUMA(J)=SUMA(J)*TEMP(I,K)*AA(K,J)
      DO 590 L=1,3
590   TEMP(I,L)=SUMA(L)
      N=3
      DO 600 I=1,3
      N=N+1
      DO 600 J=1,3
```

```
600   XDD(1,N)=XDD(1,N)-TEMP(I,J)*XD(1,J+3)
      DO 690 I=1,6
      GO TO (610,620,630,640,650,660), I
610   IF (NTX.EQ.0) 670,680
620   IF (NTY.EQ.0) 670,680
630   IF (NTZ.EQ.0) 670,680
640   IF (NRX.EQ.0) 670,680
650   IF (NRY.EQ.0) 670,680
660   IF (NRZ.EQ.0) 670,680
670   IRLM=(I-1)*7
      CALL INTEG (298+IRLM,165+IRLM)
      GO TO 690
680   XD(1,I)=0.0
      XDD(1,I)=0.0
690   CONTINUE
      XD(1,7)=XD(1,6)*X(1,10)-XD(1,5)*X(1,13)
      XD(1,8)=XD(1,6)*X(1,11)-XD(1,5)*X(1,14)
      XD(1,9)=XD(1,6)*X(1,12)-XD(1,5)*X(1,15)
      XD(1,10)=XD(1,4)*X(1,13)-XD(1,6)*X(1,7)
      XD(1,11)=XD(1,4)*X(1,14)-XD(1,6)*X(1,8)
      XD(1,12)=XD(1,4)*X(1,15)-XD(1,6)*X(1,9)
      XD(1,13)=XD(1,5)*X(1,7)-XD(1,4)*X(1,10)
      XD(1,14)=XD(1,5)*X(1,8)-XD(1,4)*X(1,11)
      XD(1,15)=XD(1,5)*X(1,9)-XD(1,4)*X(1,12)
      DO 700 I=7,15
      IRLM=(I-1)*7
      CALL INTEG (165+IRLM,32+IRLM)
700   CONTINUE
      IF (IFIN) 740,710,740
710   IF (IVAL) 740,720,720
720   DO 730 I=7,15
      N=I-6
730   DC(N)=X(1,I)
      DELT1=DC(1)*DC(5)*DC(9)+DC(3)*DC(4)*DC(8)+DC(2)*DC(6)*DC(7)-DC(3)*
     1DC(5)*DC(7)-DC(2)*DC(4)*DC(9)-DC(1)*DC(6)*DC(8)
      X(1,7)=.5*(DC(1)+(DC(5)*DC(9)-DC(8)*DC(6))/DELT1)
      X(1,8)=.5*(DC(2)+(DC(7)*DC(6)-DC(4)*DC(9))/DELT1)
      X(1,9)=.5*(DC(3)+(DC(4)*DC(8)-DC(7)*DC(5))/DELT1)
      X(1,10)=.5*(DC(4)+(DC(8)*DC(3)-DC(2)*DC(9))/DELT1)
      X(1,11)=.5*(DC(5)+(DC(1)*DC(9)-DC(7)*DC(3))/DELT1)
      X(1,12)=.5*(DC(6)+(DC(7)*DC(2)-DC(8)*DC(1))/DELT1)
      X(1,13)=.5*(DC(7)+(DC(2)*DC(6)-DC(5)*DC(3))/DELT1)
      X(1,14)=.5*(DC(8)+(DC(4)*DC(3)-DC(1)*DC(6))/DELT1)
      X(1,15)=.5*(DC(9)+(DC(1)*DC(5)-DC(4)*DC(2))/DELT1)
740   TR(1,1)=X(1,7)
      TR(2,1)=X(1,8)
      TR(3,1)=X(1,9)
      TR(1,2)=X(1,10)
      TR(2,2)=X(1,11)
      TR(3,2)=X(1,12)
      TR(1,3)=X(1,13)
      TR(2,3)=X(1,14)
      TR(3,3)=X(1,15)
      TRL(1,1)=TR(1,1)
      TRL(1,2)=TR(1,2)
      TRL(1,3)=TR(1,3)
```

```
      TRL(2,1)=TR(2,1)*CSLP-TR(3,1)*SSLP                                MAN6730
      TRL(2,2)=TR(2,2)*CSLP-TR(3,2)*SSLP                                MAN6740
      TRL(2,3)=TR(2,3)*CSLP-TR(3,3)*SSLP                                MAN6750
      TRL(3,1)=TR(2,1)*SSLP+TR(3,1)*CSLP                                MAN6760
      TRL(3,2)=TR(2,2)*SSLP+TR(3,2)*CSLP                                MAN6770
      TRL(3,3)=TR(2,3)*SSLP+TR(3,3)*CSLP                                MAN6780
      DO 770 J=1,3                                                      MAN6790
      CONT1=0.0                                                         MAN6800
      CONT2=0.0                                                         MAN6810
      DO 760 I=1,3                                                      MAN6820
      CONT1=CONT1+TR(I,J)*TR(I,J)                                       MAN6830
      CONT2=CONT2+TRL(I,J)*TRL(I,J)                                     MAN6840
      IF (ABS(TR(I,J)).LE.1.) GO TO 750                                 MAN6850
      WRITE (6,1340) I,J,TR(I,J)                                        MAN6860
750   IF (ABS(TRL(I,J)).LE.1.) GO TO 760                                MAN6870
      WRITE (6,1350) I,J,TRL(I,J)                                       MAN6880
760   CONTINUE                                                          MAN6890
      CONT1=SQRT(CONT1)                                                 MAN6900
      CONT2=SQRT(CONT2)                                                 MAN6910
      DO 770 K=1,3                                                      MAN6920
      TR(K,J)=TR(K,J)/CONT1                                             MAN6930
770   TRL(K,J)=TRL(K,J)/CONT2                                           MAN6940
      NX=15                                                             MAN6950
      DO 790 I=1,3                                                      MAN6960
      NX=NX+1                                                           MAN6970
      XD(1,NX)=0.                                                       MAN6980
      DO 780 J=1,3                                                      MAN6990
780   XD(1,NX)=XD(1,NX)+TR(I,J)*XD(1,J)                                 MAN7000
790   XVF(I)=XD(1,NX)                                                   MAN7010
      TOTGV=SQRT(XVF(1)**2+XVF(2)**2+XVF(3)**2)                         MAN7020
      XDLS=XVF(1)                                                       MAN7030
      YDLS=XVF(2)*CSLP-XVF(3)*SSLP                                      MAN7040
      ZDLS=XVF(2)*SSLP+XVF(3)*CSLP                                      MAN7050
      TOTSV=SQRT(XDLS**2+YDLS**2+ZDLS**2)                               MAN7060
      CON1=TOTSV+1.E-10                                                 MAN7070
      ANG1=ASIN(-ZDLS/CON1)                                             MAN7080
      XD(1,19)=TOTSV*COS(ANG1)                                          MAN7090
      IF (INDBAC.NE.0) GO TO 810                                        MAN7100
      DO 800 I=16,19                                                    MAN7110
      IRLM=(I-1)*7                                                      MAN7120
800   CALL INTEG (165+IRLM,32+IRLM)                                     MAN7130
810   CONTINUE                                                          MAN7140
      DO 820 I=1,3                                                      MAN7150
      GACC(I)=0.                                                        MAN7160
      DO 820 J=1,3                                                      MAN7170
820   GACC(I)=GACC(I)+XDD(1,J)*TR(I,J)                                  MAN7180
      XDDLS=GACC(1)                                                     MAN7190
      YDDLS=GACC(2)*CSLP-GACC(3)*SSLP                                   MAN7200
      ZDDLS=GACC(2)*SSLP+GACC(3)*CSLP                                   MAN7210
      XNUMB=TR(3,1)                                                     MAN7220
      XNUMBL=TRL(3,1)                                                   MAN7230
      IF (ABS(XNUMB)-1.) 840,840,830                                    MAN7240
830   XNUMB=XNUMB/ABS(XNUMB)                                            MAN7250
840   THETA=-ASIN(XNUMB)*DEGRAD                                         MAN7260
      IF (ABS(XNUMBL)-1.) 860,860,850                                   MAN7270
850   XNUMBL=XNUMBL/ABS(XNUMBL)                                         MAN7280
```

```
860   THETAL=-ASIN(XNUMBL)*DEGRAD
      IF (ABS(TR(2,1))+ABS(TR(1,1)).NE.0.0) GO TO 870
      ICONT=PHIP/70.
      PHI=90.*ICONT
      ICONT=PSIP/70.
      PSI=90.*ICONT
      IF (SLOPE.NE.0.0) GO TO 880
      PSIL=PSI
      PHIL=PHI
      GO TO 890
870   PSI=ATAN2(TR(2,1),TR(1,1))*DEGRAD
      PHI=ATAN2(TR(3,2),TR(3,3))*DEGRAD
880   PSIL=ATAN2(TRL(2,1),TRL(1,1))*DEGRAD
      PHIL=ATAN2(TRL(3,2),TRL(3,3))*DEGRAD
890   CON1=TOTGV+1.E-10
      PSIP=PSI
      PHIP=PHI
      GAMA=ASIN(-XVF(3)/CON1)*DEGRAD
      IF (XVF(2)) 920,900,920
900   IF (XVF(1)) 920,910,920
910   SIGMA=0.
      GO TO 930
920   SIGMA=ATAN2(XVF(2),XVF(1))*DEGRAD
930   ALT=-X(1,18)
      RANGE=X(1,19)
      XF=X(1,16)
      YF=X(1,17)
      ZF=X(1,18)
      XLS=XF
      YLS=YF*CSLP-ZF*SSLP
      ZLS=YF*SSLP+ZF*CSLP
      TLSV=SQRT(XLS**2+YLS**2)
      IF (INDBAC.EQ.0) GO TO 940
C         RESET SAVED TIME HISTORY VARIABLES FOR STROKE,
C                    PSTTM, AND PELST
      PELST=SPELST
      PSTTM=SPSTTM
      SAVSTB=SAVST2
      SAVSTA=SAVST1
      INDBAC=0
      GO TO 420
940   IF (IFIN) 980,950,980
C         IVAL .LT. 0 IMPLIES INTEGRATION BACKUP ON VARIABLE STEP
950   IF (IVAL) 970,960,960
960   KOUNT1=KOUNT1+1
      INDACL=1
      GO TO 980
970   INDBAC=1
980   CALL CUT
      IF (JCUT.LT.0) GO TO 1170
      IF (INDACL.EQ.0) GO TO 1000
      INDACL=0
      IF (ACLR.GE.0.0) GO TO 990
      WRITE (6,1360) ACLR
      GO TO 20
990   CONTINUE
```

```
1000  IF (JCUT.GT.0) GO TO 1020                                          MAN7850
      IF (KOUNT1-KOUNT2) 1180,1010,1180                                  MAN7860
1010  KOUNT1=0                                                           MAN7870
C     INTEGRATED VALUES OF VARIABLES ARE READY FOR                       MAN7880
C     PRINTING - ACCELERATIONS ARE XDD(1,J), VELOCITIES ARE              MAN7890
C     XD(1,J), AND DISPLACEMENTS ARE X(1,J)-J=1 TO MAX                   MAN7900
C     NUMBER OF VARIABLES.                                               MAN7910
1020  IF (JCUT.EQ.LOCSTR) GO TO 1140                                     MAN7920
      DO 1030 I=1,3                                                      MAN7930
      XLPF(I)=0.0                                                        MAN7940
      DO 1030 J=1,3                                                      MAN7950
      XLPF(I)=XLPF(I)+TRL(I,J)*XPF(J)                                    MAN7960
1030  ANG(I,J)=ACOS(TRL(I,J))*DEGRAD                                     MAN7970
      CALL PRINT2                                                        MAN7980
      IF (NMBNC.EQ.NMBNCS) GO TO 1120                                    MAN7990
      IF (JCUT) 1170,1180,1040                                           MAN8000
1040  LOCA=(NSAV(JCUT)-13)/6                                             MAN8010
      GO TO (1050,1060,1070,1080,1090,1095,1100,1110), LOCA              MAN8020
1050  WRITE (6,1380) T                                                   MAN8030
      GO TO 1130                                                         MAN8040
1060  WRITE (6,1390) X(1,19)                                             MAN8050
      GO TO 1130                                                         MAN8060
1070  WRITE (6,1400) XD(1,18)                                            MAN8070
      GO TO 1130                                                         MAN8080
1080  WRITE (6,1410) XD(1,16)                                            MAN8090
      GO TO 1130                                                         MAN8100
1090  WRITE (6,1420) XD(1,17)                                            MAN8110
      GO TO 1130                                                         MAN8120
 1095 WRITE (6,1430) XD(1,4)                                             MAN8130
      GO TO 1130                                                         MAN8140
1100  WRITE (6,1440) XD(1,5)                                             MAN8150
      GO TO 1130                                                         MAN8160
1110  WRITE (6,1450) XD(1,6)                                             MAN8170
      GO TO 1130                                                         MAN8180
1120  WRITE (6,1460) NMBNC                                               MAN8190
1130  WRITE (6,1370)                                                     MAN8200
      GO TO 20                                                           MAN8210
1140  IF (IND(LOCSTR).NE.0) GO TO 1150                                   MAN8220
C         TOUCH-DOWN                                                     MAN8230
C               INTEGRATION CUTOFF ON STROKE = 0, RESET INDICATORS TO    MAN8240
C                 CUT OFF INTEGRATION WITH TORUS LIFTS-OFF               MAN8250
      IND(LOCSTR)=1                                                      MAN8260
      XS(LOCSTR)=-3.*CUTERR                                              MAN8270
      GO TO 1160                                                         MAN8280
C         LIFT-OFF                                                       MAN8290
C               INTEGRATION CUTOFF ON STROKE .GT. 0, RESET INDICATORS TO MAN8300
C                 CUT OFF INTEGRATION WITH TORUS TOUCH-DOWN              MAN8310
1150  IND(LOCSTR)=0                                                      MAN8320
      NMBNC=NMBNC+1                                                      MAN8330
      XS(LOCSTR)=0.0                                                     MAN8340
1160  CALL SETUP                                                         MAN8350
      KOUNT1=KOUNT2-1                                                    MAN8360
      GO TO 420                                                          MAN8370
C         KOUNT1 = -8200 (ARBITRARY CONSTANT), PREVENTS PRINTING UNTIL   MAN8380
C               CUTOFF                                                   MAN8390
1170  KOUNT1=-8200                                                       MAN8400
```

```
      ACLR=0.0                                                          MAN8410
      INDBAC=1                                                          MAN8420
      GO TO 740                                                         MAN8430
 1180 CALL UPDAT                                                        MAN8440
C         INTEGRATION BACKUP REQUIRES A RETURN TO 2 FOR VARIABLE SETUP  MAN8450
      IF (INDBAC.NE.0) GO TO 740                                        MAN8460
      GO TO 420                                                         MAN8470
C                                                                       MAN8480
 1190 FORMAT (*1*,                                                      MAN8490
     *          44X* CRUSHABLE TORUS LOADS AND MOTIONS PROGRAM*        /MAN8500
     *  39X*MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1*    /MAN8510
     * 38X*MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION*/   MAN8511
     * //61X*INPUT DATA*// )                                            MAN8512
 1200 FORMAT (55H                                                       MAN8520
     1)                                                                 MAN8530
 1210 FORMAT (* *,*NO MORE DATA, END OF JOB *)                          MAN8540
 1220 FORMAT (1X,I1,2X,I3,2X,6E10.5)                                    MAN8550
 1230 FORMAT (*0*,*DATA SUBSCRIPT*,2X6(4XA6,4X))                        MAN8560
 1240 FORMAT (* *,20X*XS(I)*,9X*IND(I)* )                               MAN8570
 1250 FORMAT (4X,I5,8X,6E14.7)                                          MAN8580
 1260 FORMAT (1H1,10X,60HNO. OF CUT-OFF VALUES ENTERED AS ZERO-PROBLEM CMAN8590
     1ANNOT PROCEED)                                                    MAN8600
 1270 FORMAT (1XA10,3XI5,1XI5)                                          MAN8610
 1280 FORMAT (*0*,*PRINT POSITION *,I4,* WILL CONTAIN THE VALUE FROM *, MAN8620
     *  *COMINT(*,I5,*) WITH THE LABEL *,A10 )                          MAN8630
 1290 FORMAT(*0*,A10,5X*END OF DATA LIST*)                              MAN8640
 1300 FORMAT (* *,*----------ERROR----------VMIN = DATA)111* = 0*  )    MAN8650
 1310 FORMAT (*1*,* VEHICLE WEIGHT, ON EARTH = *,E14.7,*, ON PLANET = * MAN8660
     1                                        ,E14.7                   )MAN8661
 1320 FORMAT (*0*,16X*MOMENTS AND PRODUCTS OF INERTIA*               /  MAN8670
     *          *0*,7X*XXI*,13X*XYI*,12X*XXID*, 12X*XYID*/              MAN8680
     1          * *,7X*YYI*,13X*XZI*,12X*YYID*, 12X*XZID*/              MAN8690
     1          * *,7X*ZZI*,13X*YZI*,12X*ZZID*, 12X*YZID*          )    MAN8721
 1330 FORMAT (*0*,4(1XE14.7)/* *,4(1XE14.7)/* *,4(1XE14.7))             MAN8700
 1340 FORMAT (* *,*----------ERROR----------TR (*,I2,*,*,I2,*)= *,E20.13MAN8710
     1,VECTOR TAKES UNIT VECTOR FORM*)                                  MAN8720
 1350 FORMAT (* *,*----------ERROR----------TRL(*,I2,*,*,I2,*)= *,E20.13MAN8730
     1     ,*.  VECTOR TAKES UNIT VECTOR FORM* )                        MAN8740
 1360 FORMAT (//,1X,41HCUT OFF,CLEARANCE LESS THAN ALLOWABLE BY ,F10.4) MAN8750
 1370 FORMAT (1H1)                                                      MAN8760
 1380 FORMAT (//,1X,16HCUT OFF ON TIME=,F10.4,1X,7HSECONDS)             MAN8770
 1390 FORMAT (//,1X,38HCUT OFF ON TOTAL VEL RELATIVE TO SURF=,F10.4,1X,6MAN8780
     1HFT/SEC)                                                          MAN8790
 1400 FORMAT (//,1X,25HCUT OFF ON SURFACE RANGE=,F10.4,1X,2HFT)         MAN8800
 1410 FORMAT (//,1X,35HCUT OFF ON PARALLEL TO SURFACE VEL=,F10.4,1X,6HFTMAN8810
     1/SEC)                                                             MAN8820
 1420 FORMAT (//,1X,38HCUT OFF ON DISTANCE NORMAL TO SURFACE=,F10.4,1X, MAN8830
     12HFT)                                                             MAN8840
 1430 FORMAT (//,1X,28HCUT OFF ON LANDER ROLL RATE=,F10.4,1X,7HRAD/SEC) MAN8850
 1440 FORMAT (//,1X,29HCUT OFF ON LANDER PITCH RATE=,F10.4,1X,7HRAD/SEC)MAN8860
 1450 FORMAT (//,1X,27HCUT OFF ON LANDER YAW RATE=,F10.4,1X,7HRAD/SEC)  MAN8870
 1460 FORMAT (//* CUTOFF ON NUMBER OF BOUNCES, NUMBER = *,I5)           MAN8880
      END                                                               MAN8890-
```

APPENDIX B

```
SUBROUTINE PCCUT (IYD,IY,CUTVAL,IDIRT)
DIMENSION X(9), XMN1(9)
DIMENSION XS(9), IND(9), IAD(9), NAMBCD(9), LOCNAM(50)
COMMON COMINT (600)
EQUIVALENCE ( COMINT(  1 ), T         )
EQUIVALENCE ( COMINT(  2 ), HMAX      )
EQUIVALENCE ( COMINT(  3 ), EMIN      )
EQUIVALENCE ( COMINT(  4 ), LMAX      )
EQUIVALENCE ( COMINT(  5 ), HZ        )
EQUIVALENCE ( COMINT(  6 ), IP        )
EQUIVALENCE ( COMINT(  7 ), IVARH     )
EQUIVALENCE ( COMINT(  8 ), IMTH      )
EQUIVALENCE ( COMINT(  9 ), IPRNT     )
EQUIVALENCE ( COMINT( 10 ), IFIN      )
EQUIVALENCE ( COMINT( 11 ), IVAL      )
EQUIVALENCE ( COMINT( 12 ), IPTOTL    )
EQUIVALENCE ( COMINT( 13 ), IPTATL    )
EQUIVALENCE ( COMINT(  14), XS        )
EQUIVALENCE ( COMINT(  23), IND       )
EQUIVALENCE ( COMINT( 347), HMIN      )
EQUIVALENCE ( COMINT( 384), CUTERR    )
EQUIVALENCE ( COMINT( 385), J         )
ENTRY LOC
IPTAIL=IPTATL+1
IF (IPTATL.LE.9) GO TO 10
WRITE (6,870)
STOP
NAMBCD(IPTATL)=IY
IAD(IPTATL)=IYD
XS(IPTATL)=CUTVAL
IND(IPTATL)=IDIRT
RETURN
ENTRY INUPD
IPTOTL=IPTOTL+1
IF (IPTOTL.LE.50) GO TO 20
WRITE (6,880)
STOP
LOCNAM(IPTOTL)=IYD
RETURN
ENTRY SETUP
IERROR=0
ISTEP=1
I=0
IVAL=0
HZ=HMAX*2.**(-IP)
IPT2=2**IP
IPT1=0
IPRNT=0
IFIN=0
INDRH=1
IB5=1
IB1=1
IALP=4
LIST=0
INUPD=0
IF (IMTH) 70,30,70
```

```
30     IB2=1                                                            PCT 570
       JJ=7                                                             PCT 580
       IB3=2                                                            PCT 590
       ISCNT=0                                                          PCT 600
       IBETA=3                                                          PCT 610
       IGAM=-1                                                          PCT 620
       IF (IVARH) 40,50,40                                              PCT 630
40     IB4=1                                                            PCT 640
       IB6=2                                                            PCT 650
       GO TO 60                                                         PCT 660
50     IB4=2                                                            PCT 670
       IB6=1                                                            PCT 680
60     IB7=1                                                            PCT 690
       GO TO 80                                                         PCT 700
70     IB2=2                                                            PCT 710
       JJ=6                                                             PCT 720
       IB3=1                                                            PCT 730
       IB7=2                                                            PCT 740
80     HD2=HZ/2.                                                        PCT 750
       H=HD2                                                            PCT 760
       A1111=HZ/24.                                                     PCT 770
       A2222=19./270.                                                   PCT 780
       RKTME=T                                                          PCT 790
       RETURN                                                           PCT 800
       ENTRY INTEG                                                      PCT 810
       JJR1=IY+JJ-1                                                     PCT 820
       GO TO (90,130,140,150), IB5                                      PCT 830
90     GO TO (100,110,120,100), ISTEP                                   PCT 840
100    COMINT(JJR1)=COMINT(IY)+H*COMINT(IYD)                            PCT 850
       RETURN                                                           PCT 860
110    COMINT(JJR1)=COMINT(IY+1)+H*COMINT(IYD)                          PCT 870
       RETURN                                                           PCT 880
120    COMINT(JJR1)=COMINT(IY+2)+H*COMINT(IYD)                          PCT 890
       RETURN                                                           PCT 900
130    COMINT(JJR1)=COMINT(IY+3)+H/6.*(COMINT(IYD+3)+2.*(COMINT(IYD+2)+COPCT 910
      1MINT(IYD+1))+COMINT(IYD))                                        PCT 920
       RETURN                                                           PCT 930
140    CONTINUE                                                         PCT 940
       COMINT(IY+5)=COMINT(IY)+A1111*(55.*COMINT(IYD)-59.*COMINT(IYD+1)+3PCT 950
      17.*COMINT(IYD+2)-9.*COMINT(IYD+3))                               PCT 960
       RETURN                                                           PCT 970
150    CN1=COMINT(IY+1)+A1111*(9.*COMINT(IYD)+19.*COMINT(IYD+1)-5.*COMINTPCT 980
      1(IYD+2)+COMINT(IYD+3))                                           PCT 990
       XM=ABS(COMINT(IY)-CN1)                                           PCT1000
       COMINT(IY)=CN1+A2222*(COMINT(IY)-CN1)                            PCT1010
       IF (IB6-2) 170,160,170                                           PCT1020
160    RETURN                                                           PCT1030
170    IF (CN1) 180,200,180                                             PCT1040
180    XM1=ABS(XM/CN1)                                                  PCT1050
       IF (XM-XM1) 200,200,190                                          PCT1060
190    XM=XM1                                                           PCT1070
200    IF (XM-EMAX) 220,210,210                                         PCT1080
210    IF (HZ.GT.HMIN) IVAL=-8300                                       PCT1090
       IVAL=IVAL+1                                                      PCT1100
       RETURN                                                           PCT1110
220    IF (XM-EMIN) 240,230,230                                         PCT1120
```

```
230    IVAL=IVAL+1                                   PCT1130
240    RETURN                                        PCT1140
       ENTRY UPDAT                                   PCT1150
       IFIN=1                                        PCT1160
       IF (IPRNT) 260,250,260                        PCT1170
250    IPT1=IPT2                                     PCT1180
260    IF (IB1-2) 270,500,270                        PCT1190
270    IF (IALP-1) 390,280,390                       PCT1200
280    I=I+1                                         PCT1210
       IPT1=IPT1-1                                   PCT1220
       GO TO (360,300), IB2                          PCT1230
290    IBETA=IBETA-1                                 PCT1240
300    IALP=4                                        PCT1250
       ISTEP=1                                       PCT1260
       H=H/2.                                        PCT1270
       IFIN=0                                        PCT1280
       IB5=1                                         PCT1290
       GO TO (330,310), IB3                          PCT1300
310    DO 320 IMVER=1,IPTOTL                         PCT1310
       KMVER=LOCNAM(IMVER)                           PCT1320
       COMINT(KMVER+1)=COMINT(KMVER+3)               PCT1330
       COMINT(KMVER+2)=COMINT(KMVER+4)               PCT1340
       COMINT(KMVER+3)=COMINT(KMVER+5)               PCT1350
       COMINT(KMVER+4)=COMINT(KMVER+5)               PCT1360
320    COMINT(KMVER)=COMINT(KMVER+6)                 PCT1370
       GO TO 350                                     PCT1380
330    DO 340 IMVER=1,IPTOTL                         PCT1390
       KMVER=LOCNAM(IMVER)                           PCT1400
       COMINT(KMVER)=COMINT(KMVER+5)                 PCT1410
340    COMINT(KMVER+1)=COMINT(KMVER+3)               PCT1420
350    IPRNT=IPT1                                    PCT1430
       RETURN                                        PCT1440
360    IF (IBETA-1) 290,370,290                      PCT1450
370    IB1=2                                         PCT1460
       IB5=3                                         PCT1470
       IFIN=0                                        PCT1480
       IF (IVARH) 380,310,380                        PCT1490
380    IB6=2                                         PCT1500
       GO TO 310                                     PCT1510
390    IALP=IALP-1                                   PCT1520
       IF (IALP-1) 400,410,400                       PCT1530
400    ISTEP=ISTEP+1                                 PCT1540
       GO TO 420                                     PCT1550
410    IB5=2                                         PCT1560
420    IF (IALP-2) 430,440,430                       PCT1570
430    T=T+HD2                                       PCT1580
       GO TO 450                                     PCT1590
440    H=HZ                                          PCT1600
450    GO TO (460,480), IB7                          PCT1610
460    DO 470 IMVER=1,IPTOTL                         PCT1620
       KMVER=LOCNAM(IMVER)                           PCT1630
       COMINT(KMVER+5)=COMINT(KMVER+4)               PCT1640
       COMINT(KMVER+4)=COMINT(KMVER+3)               PCT1650
       COMINT(KMVER+3)=COMINT(KMVER+2)               PCT1660
       COMINT(KMVER+2)=COMINT(KMVER+1)               PCT1670
       COMINT(KMVER+1)=COMINT(KMVER)                 PCT1680
```

```
470   COMINT(KMVER)=COMINT(KMVER+6)
      GO TO 350
480   DO 490 IMVER=1,IPTOTL
      KMVER=LOCNAM(IMVER)
      COMINT(KMVER+3)=COMINT(KMVER+2)
      COMINT(KMVER+2)=COMINT(KMVER+1)
      COMINT(KMVER+1)=COMINT(KMVER)
490   COMINT(KMVER)=COMINT(KMVER+5)
      GO TO 350
500   IGAM=-IGAM
      IF (IGAM) 520,520,510
510   IB5=4
      T=T+H
      GO TO 480
520   IB5=3
      GO TO (530,540), IB4
530   IPT1=IPT1-1
      IFIN=0
      GO TO 570
540   IF (IVAL) 550,580,550
550   ISCNT=0
      IF (IVAL) 630,630,560
560   INDRH=1
      IPT1=IPT1-1
      I=I+1
      IFIN=0
570   IVAL=0
      GO TO 350
580   IF (ISCNT-2) 590,590,600
590   ISCNT=ISCNT+1
      GO TO 560
600   IF (2*(IPT1/2).EQ.IPT1) GO TO 560
      IF (H-HMAX) 610,560,560
610   IPT1=IPT1/2
      ISCNT=0
      IBETA=3
      IALP=4
      IB1=1
      IB5=1
      ISTEP=1
      INDRH=0
      IPT2=IPT2/2
      RKTME=T
      HD2=H
      HZ=2.*H
      IFIN=0
620   I=0
      A1111=HZ/24.
      GO TO 570
630   IF (IPT1) 560,640,640
640   IBETA=3
      IALP=4
      ISTEP=1
      IB1=1
      IB5=1
      IF (I-3) 650,670,650
```

APPENSIX B

```
650   T=T-H                                              PCT2250
      RKTME=T                                            PCT2260
      IPT1=2*IPT1                                        PCT2270
      DO 660 IMVER=1,IPTOTL                              PCT2280
      KMVER=LOCNAM(IMVER)                                PCT2290
660   COMINT(KMVER)=COMINT(KMVER+1)                      PCT2300
      GO TO 690                                          PCT2310
670   T=RKTME                                            PCT2320
      IPT1=2*(IPT1+3)                                    PCT2330
      DO 680 IMVER=1,IPTOTL                              PCT2340
      KMVER=LOCNAM(IMVER)                                PCT2350
680   COMINT(KMVER)=COMINT(KMVER+4)                      PCT2360
690   HZ=H/2.                                            PCT2370
      IF (HZ.LT.HMIN) HZ=HMIN                            PCT2380
      HD2=HZ/2.                                          PCT2390
      H=HD2                                              PCT2400
      IPT2=2*IPT2                                        PCT2410
      INDRH=-1                                           PCT2420
      GO TO 620                                          PCT2430
      ENTRY CUT                                          PCT2440
      IF (IFIN) 700,710,700                              PCT2450
700   J=0                                                PCT2460
      IERROR=1                                           PCT2470
      RETURN                                             PCT2480
710   K=1                                                PCT2490
720   IF (K.LE.IPTATL) GO TO 750                         PCT2500
      IF (K-1) 730,700,730                               PCT2510
730   IK=K-1                                             PCT2520
      DO 740 I=1,IK                                      PCT2530
      KMVER=IAD(I)                                       PCT2540
      XMN1(I)=COMINT(KMVER)                              PCT2550
740   CONTINUE                                           PCT2560
      GO TO 700                                          PCT2570
750   KMVER=IAD(K)                                       PCT2580
      X(K)=COMINT(KMVER)                                 PCT2590
      XU=XS(K)+(ABS(XS(K))+1.)*CUTERR                    PCT2600
      XL=XS(K)-(ABS(XS(K))+1.)*CUTERR                    PCT2610
      IF (IND(K)) 760,780,760                            PCT2620
760   IF (X(K)-XU) 770,790,790                           PCT2630
770   IF (X(K)-XL) 820,820,810                           PCT2640
780   IF (X(K)-XL) 790,790,800                           PCT2650
790   K=K+1                                              PCT2660
      IF (K-10) 720,730,730                              PCT2670
800   IF (X(K)-XU) 810,820,820                           PCT2680
810   J=K                                                PCT2690
      IERROR=1                                           PCT2700
      RETURN                                             PCT2710
820   T=T-HZ                                             PCT2720
      HZ=(HZ*(XS(K)-XMN1(K))/(X(K)-XMN1(K)))/2.          PCT2730
      IF (INDRH) 840,830,840                             PCT2740
830   HZ=HZ/2.                                           PCT2750
840   I=0                                                PCT2760
      HD2=HZ/2.                                          PCT2770
      H=HD2                                              PCT2780
      IF (IERROR.NE.0) GO TO 850                         PCT2790
      KMVER=IAD(K)                                       PCT2800
```

```
      WRITE (6,890) NAMBCD(K),COMINT(KMVER)
      STOP
  850 CONTINUE
      A1111=HZ/24.
      ISCNT=0
      IBETA=3
      IALP=4
      ISTEP=1
      IB1=1
      IB2=2
      IB3=1
      IB5=1
      IB7=2
      JJ=6
      IPT1=1000
      IPT2=1000
      IPRNT=1000
      RKTME=T
      DO 860 IMVER=1,IPTOTL
      KMVER=LOCNAM(IMVER)
  860 COMINT(KMVER)=COMINT(KMVER+1)
      IFIN=1
      J=-1
      IERROR=1
      RETURN
C
  870 FORMAT (* *,*-----------JOB TERMINATED, MORE THAN NINE CALLS TO LOC
     1----------* )
  880 FORMAT(* *,*-----------JOB TERMINATED, MORE THAN FIFTY CALLS TO *
     1  ,*INUPD----------* )
  890 FORMAT (18H CUTOFF PASSED BY ,A6,3H = ,E14.7,27H ON THE INITIAL CA
     1LL TO CUT)
      END
```

APPENDIX B

```
      SUBROUTINE AERO
      COMMON COMINT(600)
      EQUIVALENCE ( COMINT( 552), Q       )
      EQUIVALENCE ( COMINT( 553), XMACH   )
      EQUIVALENCE ( COMINT( 404), BETA    )
      EQUIVALENCE ( COMINT( 401), ALPHA   )
C     PROVIDE ALL DATA HERE NECESSARY TO DEFINE
C     AERODYNAMIC FORCES ACTING ON THE VEHICLE.
      RETURN
      ENTRY AERO1
C     DEFINE AERO FORCES ON VEHICLE AT TIME=T
      CALL ENVIR1
      XMACH=0.
      Q=0.
      ALPHA=0.
      BETA=0.
      RETURN
      END
```

```
      SUBROUTINE ENVIR
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 554), SOUND   )
      EQUIVALENCE ( COMINT( 398), XWND    )
      EQUIVALENCE ( COMINT( 399), YWND    )
      EQUIVALENCE ( COMINT( 400), ZWND    )
      EQUIVALENCE ( COMINT( 389), XW      )
      EQUIVALENCE ( COMINT( 390), YW      )
      EQUIVALENCE ( COMINT( 391), ZW      )
C     READ ENVIRONMENT DATA
      RETURN
      ENTRY ENVIR1
C     DEFINE GROUND REF, ENVIRONMENT AT TIME=T
C     XW=NORTH WIND YW=EAST WIND ZW=VERTICAL WIND
      SOUND=0.
      XWND=0.
      YWND=0.
      ZWND=0.
      XW=0.
      YW=0.
      ZW=0.
      RETURN
      END
```

APPENDIX B

```
      SUBROUTINE PRINT (VN,LC,LP)
      DIMENSION TITLEG(72), ISUBCG(72), PNTLST(72)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 473), IERPRT )
      EQUIVALENCE ( COMINT(   1), T      )
      EQUIVALENCE ( COMINT(   5), HZ     )
      DATA ISUBCG /
     1     428, 430, 433, 557, 455, 165, 298, 186, 319, 572, 575, 578,
     2     429, 431, 434, 558, 456, 172, 305, 193, 326, 573, 576, 579,
     3     344, 432, 435, 348, 590, 179, 312, 200, 333, 574, 577, 580,
     4     406, 591, 454, 413, 549, 543, 524, 527, 586, 562, 600, 600,
     5     470, 592, 437, 414, 550, 544, 525, 528, 587, 563, 600, 600,
     6     393, 593, 438, 415, 564, 545, 526, 529, 588, 600, 600, 600  /
      DATA TITLEG /
     1 10H    XLS    , 10H   XDLS    , 10H   XDDLS   , 10HFLT. ANGLE,
     2 10H   FONTT   , 10H    XD     , 10H    XDD    , 10H   PHID    ,
     3 10H   PHIDD   , 10H ANG(1,1)  , 10H ANG(1,2)  , 10H ANG(1,3)  ,
     4 10H    YLS    , 10H   YDLS    , 10H   YDDLS   , 10HVER. ANGLE,
     5 10H   TORTN   , 10H    YD     , 10H    YDD    , 10H  THETAD   ,
     6 10H  THETADD  , 10H ANG(2,1)  , 10H ANG(2,2)  , 10H ANG(2,3)  ,
     7 10H    ZLS    , 10H   ZDLS    , 10H   ZDDLS   , 10H  STROKE   ,
     8 10H   FRICT   , 10H    ZD     , 10H    ZDD    , 10H   PSID    ,
     9 10H   PSIDD   , 10H ANG(3,1)  , 10H ANG(3,2)  , 10H ANG(3,3)  ,
     $ 10H   RANGE   , 10H  XCF(1)   , 10HT VELOCITY, 10H    XN     ,
     $ 10H   FS(1)   , 10H  XPF(1)   , 10H   FF(1)   , 10H   FF(4)   ,
     $ 10H  TFC(1)   , 10H  ANGLD    , 10H   BLANK   , 10H   BLANK   ,
     $ 10H   ACLR    , 10H  XCF(2)   , 10H  YLGRAV   , 10H    YN     ,
     $ 10H   FS(2)   , 10H  XPF(2)   , 10H   FF(2)   , 10H   FF(5)   ,
     $ 10H  TFC(2)   , 10H  ANGLD2   , 10H   BLANK   , 10H   BLANK   ,
     $ 10H    GZ     , 10H  XCF(3)   , 10H  ZLGRAV   , 10H    ZN     ,
     $ 10H  ANGLD3   , 10H  XPF(3)   , 10H   FF(3)   , 10H   FF(6)   ,
     $ 10H  TFC(3)   , 10H   BLANK   , 10H   BLANK   , 10H   BLANK    /
      IF ((LP.LE.72).AND.(LP.GT.0)) GO TO 10
      WRITE (6,100) LP
      STOP
   10 TITLEG(LP)=VN
      IF (LC.GT.0) GO TO 20
      WRITE (6,110) LC
      STOP
   20 ISUBCG(LP)=LC
      RETURN
      ENTRY PRINT1
      IF (IERPRT.NE.0) GO TO 30
      ICONTP=10
      IPRINT=10
      GO TO 40
   30 ICONTP=6
      IPRINT=6
   40 RETURN
      ENTRY PRINT2
      CALL SECOND (TIMECP)
      IF (IPRINT.NE.ICONTP) GO TO 60
      IPRINT=0
      WRITE (6,120)
      WRITE (6,130) (TITLEG(I),I=1,36)
      IF (IERPRT.EQ.0) GO TO 50
```

```
      WRITE (6,130) (TITLEG(I),I=37,72)                                 PRT 570
50    CONTINUE                                                          PRT 580
60    WRITE (6,150) TIMECP,T,HZ                                         PRT 590
      DO 70 I=1,36                                                      PRT 600
      LKI=ISUBCG(I)                                                     PRT 610
70    PNTLST(I)=COMINT(LKI)                                             PRT 620
      WRITE (6,140) (PNTLST(I),I=1,36)                                  PRT 630
      IF (IERPRT.EQ.0) GO TO 90                                         PRT 640
      DO 80 I=37,72                                                     PRT 650
      LKI=ISUBCG(I)                                                     PRT 660
80    PNTLST(I)=COMINT(LKI)                                             PRT 670
      WRITE (6,140) (PNTLST(I),I=37,72)                                 PRT 680
90    CONTINUE                                                          PRT 690
      IPRINT=IPRINT+1                                                   PRT 700
      RETURN                                                            PRT 710
C                                                                       PRT 720
100   FORMAT (* *,*-----------ERROR----------LPRINT = *,I5,*, RANGE IS ONPRT 730
     *LY 1-72* )                                                        PRT 740
110   FORMAT (* *,*-----------ERROR----------LCOMON = *,I5,*, RANGE .GT. PRT 750
     *0 * )                                                             PRT 751
120   FORMAT(*1*,11X*LOCAL SURFACE COORDINATE SYSTEM*,                  PRT 760
     *12X1H*,9X*LANDER COORDINATE SYSTEM*,10X1H*,6X*VEHICLE ORIENTATION*PRT 770
     *)                                                                 PRT 771
130   FORMAT (5(1XA10),1H*,A10,3(1XA10),1H*,A10,2(1XA10))               PRT 780
140   FORMAT (12(1XE10.3))                                              PRT 790
150   FORMAT (*0*,*CP TIME   = *,F10.3,* T        = *,E10.3,            PRT 800
     1 * DT        = *,E10.3)                                           PRT 801
      END                                                               PRT 810-
```

APPENDIX B

```
      SUBROUTINE SOLVE (A,G,SUM,N,M)
      DIMENSION A(3,4),G(3,3),H(3,3)
C     THIS IS A SUBROUTINE FOR DETERMINING THE VALUE OF C IN THE
C     MATRIX EQUATION A*C=G.  THE VALUES OF A AND G ARE PROVIDED
C     AT THE TIME OF CALLING.  N IS THE ORDER OF A (MUST BE
C     A SQUARE MATRIX) AND M IS THE NUMBER OF COLUMNS IN
C     G(A AND G MUST HAVE SAME NUMBER OF ROWS).  THE VALUE
C     OF C IS STORED IN G LOCATION AT RETURN.  IF THE INVERSE
C     OF A IS REQUIRED, MAKE M NEGATIVE.  IF ONLY THE INVERSE
C     OF A IS REQUIRED(G MATRIX DOES NOT EXIST), ENTER M=0.
C     DETERMINANT OF A IS STORED IN LOCATION SUM AT RETURN.
C     IF THE INVERSE OF A IS COMPUTED, IT IS STORED
C     IN LOCATION A AT RETURN TO THE CALLING PROGRAM.
C     IF IT IS DESIRED TO MAKE THIS A DOUBLE PRECISION
C     SUBROUTINE, THE FOLLOWING VARIABLES MUST BE TYPED
C     DOUBLE PRECISION  A,G,H, AND SUM.
C     WHEN PROVIDING DIMENSIONING INFORMATION FOR
C     THE VARIABLE A ,( A(I,J) ), THE VALUE OF J MUST
C     BE ONE GREATER THAN I, IE., J=I+1.
      NGO=1
      NST=1
      IF (M) 10,20,50
10    M=-M
      NST=2
20    NGO=2
      DO 40 I=1,N
      DO 40 J=1,N
      H(I,J)=0.
      IF (I-J) 40,30,40
30    H(I,J)=1.
40    CONTINUE
50    N1=N+1
      N2=N-1
      KPT=2
      DO 300 IP=1,NST
      IF (M) 70,60,70
60    IP=2
70    GO TO (80,90), IP
80    NSP=M
      GO TO 100
90    NSP=N
100   DO 300 JP=1,NSP
      DO 130 I=1,N
      GO TO (110,120), IP
110   A(I,N1)=G(I,JP)
      GO TO 130
120   A(I,N1)=H(I,JP)
130   CONTINUE
      DO 140 I=KPT,N1
140   A(1,I)=A(1,I)/A(1,1)
      DO 200 I=2,N
      NN=I-1
      DO 200 J=KPT,N1
      NM=J-1
      SUM=0.
      IF (NM-NN) 150,150,160
```

```
150   KK=NM
      GO TO 170
160   KK=NN
170   DO 180 L=1,KK
180   SUM=SUM+A(L,J)*A(I,L)
      A(I,J)=A(I,J)-SUM
      IF (J-I) 200,200,190
190   A(I,J)=A(I,J)/A(I,I)
200   CONTINUE
      GO TO (210,220), IP
210   G(N,JP)=A(N,N1)
      GO TO 230
220   H(N,JP)=A(N,N1)
230   DO 290 I=1,N2
      NI=N-I
      GO TO (240,250), IP
240   G(NI,JP)=A(NI,N1)
      GO TO 260
250   H(NI,JP)=A(NI,N1)
260   DO 290 J=1,I
      NJ=N1-J
      GO TO (270,280), IP
270   G(NI,JP)=G(NI,JP)-A(NI,NJ)*G(NJ,JP)
      GO TO 290
280   H(NI,JP)=H(NI,JP)-A(NI,NJ)*H(NJ,JP)
290   CONTINUE
      KPT=N1
300   CONTINUE
      SUM=1.
      DO 310 I=1,N
310   SUM=SUM*A(I,I)
      GO TO (340,320), NGO
320   DO 330 I=1,N
      DO 330 J=1,N
330   A(I,J)=H(I,J)
340   RETURN
      END
```

APPENDIX B

```
      SUBROUTINE PHYS
      DIMENSION XYZI(15)
      DIMENSION BX(7,19)
      DIMENSION X(7,19), XD(7,19), XDD(7,6)
      DIMENSION XPF(3), XCD(3), FS(3), TFC(3)
      DIMENSION XVF(3), TR(3,3), TRL(3,3), FF(19)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 32 ), X         )
      EQUIVALENCE ( COMINT(165 ), XD        )
      EQUIVALENCE ( COMINT(298 ), XDD       )
      EQUIVALENCE ( COMINT (344), ZLS       )
      EQUIVALENCE ( COMINT( 348), STROKE    )
      EQUIVALENCE ( COMINT( 410), XXI       )
      EQUIVALENCE ( COMINT( 411), YYI       )
      EQUIVALENCE ( COMINT( 412), ZZI       )
      EQUIVALENCE ( COMINT( 416), XMASST    )
      EQUIVALENCE ( COMINT( 418), XYI       )
      EQUIVALENCE ( COMINT( 419), XZI       )
      EQUIVALENCE ( COMINT( 420), YZI       )
      EQUIVALENCE ( COMINT( 422), XXID      )
      EQUIVALENCE ( COMINT( 423), YYID      )
      EQUIVALENCE ( COMINT( 424), ZZID      )
      EQUIVALENCE ( COMINT( 425), XYID      )
      EQUIVALENCE ( COMINT( 426), XZID      )
      EQUIVALENCE ( COMINT( 427), YZID      )
      EQUIVALENCE ( COMINT( 445), A         )
      EQUIVALENCE ( COMINT( 446), B         )
      EQUIVALENCE ( COMINT( 447), C         )
      EQUIVALENCE ( COMINT( 448), D         )
      EQUIVALENCE ( COMINT( 449), E         )
      EQUIVALENCE ( COMINT( 450), F         )
      EQUIVALENCE ( COMINT( 451), G         )
      EQUIVALENCE ( COMINT( 452), RHOL      )
      EQUIVALENCE ( COMINT( 453), RHOP      )
      EQUIVALENCE ( COMINT( 455), FONTT     )
      EQUIVALENCE ( COMINT( 456), TORTN     )
      EQUIVALENCE ( COMINT( 457), ARTSH     )
      EQUIVALENCE ( COMINT( 459), ANGLE     )
      EQUIVALENCE ( COMINT( 460), ANGLE1    )
      EQUIVALENCE ( COMINT( 461), ANGLE2    )
      EQUIVALENCE ( COMINT( 462), ANGLE3    )
      EQUIVALENCE ( COMINT( 463), ARMOM     )
      EQUIVALENCE ( COMINT( 464), ELST      )
      EQUIVALENCE ( COMINT( 465), RDIA      )
      EQUIVALENCE ( COMINT( 466), AMU       )
      EQUIVALENCE ( COMINT( 467), XKW       )
      EQUIVALENCE ( COMINT( 468), DX        )
      EQUIVALENCE ( COMINT( 469), DTHE      )
      EQUIVALENCE ( COMINT( 470), ACLR      )
      EQUIVALENCE ( COMINT( 474), RSXCF     )
      EQUIVALENCE ( COMINT( 543), XPF       )
      EQUIVALENCE ( COMINT( 546), XCD       )
      EQUIVALENCE ( COMINT( 549), FS        )
      EQUIVALENCE ( COMINT( 503), XVF       )
      EQUIVALENCE ( COMINT( 506), TR        )
      EQUIVALENCE ( COMINT( 515), TRL       )
```

```
      EQUIVALENCE ( COMINT( 524), FF      )
      EQUIVALENCE ( COMINT( 561), DEGRAD  )
      EQUIVALENCE ( COMINT( 562), ANGLD1  )
      EQUIVALENCE ( COMINT( 563), ANGLD2  )
      EQUIVALENCE ( COMINT( 564), ANGLD3  )
      EQUIVALENCE ( COMINT( 586), TFC     )
      EQUIVALENCE ( COMINT( 590), FRICT   )
      DO 10 I=1,15
   10 XYZI(I)=0.0
      CALL INERT (XYZI)
      XXI=XYZI(1)
      YYI=XYZI(2)
      ZZI=XYZI(3)
      XYI=XYZI(4)
      XZI=XYZI(5)
      YZI=XYZI(6)
      XXID=XYZI(7)
      YYID=XYZI(8)
      ZZID=XYZI(9)
      XYID=XYZI(10)
      XZID=XYZI(11)
      YZID=XYZI(12)
      RETURN
      ENTRY PHYS1
      AT=0.
      BETA=TRL(3,1)
      ANGLE=ACOS(ABS(TRL(3,1)))
      ANGLD1=ANGLE*DEGRAD
      ANGLE1=ACOS(BETA)
      IF (ANGLE.LT.1.56.OR.ANGLE.GT.1.58) GO TO 20
      STROKE=ZLS*12.+(G-C+D+F)
      GO TO 30
   20 TN=ABS(TAN(ANGLE))
      FFFF=F*F
      STROKE=ZLS*12.+((G+D-C)*SIN(ANGLE)+(E*E+FFFF*TN*TN)**.5*COS(ANGLE)
     1)
   30 IF (STROKE.LE.0.0) GO TO 40
      CALL LDSTR
      IF (ACLR.LT.0.0) RETURN
      IF (FONTT.LE.1.E-14) GO TO 40
      CALL FORM
      IF (ARTSU.EQ.0.) GO TO 40
      GO TO 60
   40 CONTINUE
      FONTT=0.0
      TORTN=0.0
      FRICT=0.0
      DO 50 I=1,3
      FS(I)=0.0
      XPF(I)=0.0
      TFC(I)=0.0
   50 XCD(I)=0.0
      ARTSU=0.0
      ANGLE2=0.0
      ANGLE3=0.0
      ANGLD2=0.
```

APPENDIX B

```
      ANGLD3=0.                                                        PHY1130
60    CONTINUE                                                         PHY1140
      RETURN                                                           PHY1150
      END                                                              PHY1160-
```

APPENDIX B

```
      SUBROUTINE FORM
      DIMENSION  XD(7,19),             XC(3), XCF(3)
      DIMENSION XYZI(3),FRF(3),FSS(3)
      DIMENSION A(3,4),AFRF(3,3),XPFD(3),XPFX(3),FNT(3),ACELTN(3)
      DIMENSION ACTN(3)
      DIMENSION XVF(3), TR(3,3), TRL(3,3), FF(19)
      DIMENSION XPF(3), XCD(3), FS(3), TFC(3)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 5 ), HZ      )
      EQUIVALENCE ( COMINT( 165), XD      )
      EQUIVALENCE ( COMINT(344), ZLS     )
      EQUIVALENCE ( COMINT( 410), XYZI    )
      EQUIVALENCE ( COMINT( 416), XMASST  )
      EQUIVALENCE ( COMINT( 436), XLGRAV  )
      EQUIVALENCE ( COMINT( 437), YLGRAV  )
      EQUIVALENCE ( COMINT( 438), ZLGRAV  )
      EQUIVALENCE ( COMINT( 439), NTX     )
      EQUIVALENCE ( COMINT( 440), NTY     )
      EQUIVALENCE ( COMINT( 441), NTZ     )
      EQUIVALENCE ( COMINT( 442), NRX     )
      EQUIVALENCE ( COMINT( 443), NRY     )
      EQUIVALENCE ( COMINT( 444), NRZ     )
      EQUIVALENCE ( COMINT( 455), FORTT   )
      EQUIVALENCE ( COMINT( 456), TORTN   )
      EQUIVALENCE ( COMINT( 457), ARTSU   )
      EQUIVALENCE ( COMINT( 459), ANGLE   )
      EQUIVALENCE ( COMINT( 460), ANGLE1  )
      EQUIVALENCE ( COMINT( 461), ANGLE2  )
      EQUIVALENCE ( COMINT( 462), ANGLE3  )
      EQUIVALENCE ( COMINT( 463), ARMOM   )
      EQUIVALENCE ( COMINT( 466), AMU     )
      EQUIVALENCE ( COMINT( 589), RAV     )
      EQUIVALENCE ( COMINT( 543), XPF     )
      EQUIVALENCE ( COMINT( 546), XCD     )
      EQUIVALENCE ( COMINT( 549), FS      )
      EQUIVALENCE ( COMINT( 503), XVF     )
      EQUIVALENCE ( COMINT( 506), TR      )
      EQUIVALENCE ( COMINT( 515), TRL     )
      EQUIVALENCE ( COMINT( 524), FF      )
      EQUIVALENCE ( COMINT( 561), DEGRAD  )
      EQUIVALENCE ( COMINT( 562), ANGLD1  )
      EQUIVALENCE ( COMINT( 563), ANGLD2  )
      EQUIVALENCE ( COMINT( 564), ANGLD3  )
      EQUIVALENCE ( COMINT( 581), VMIN    )
      EQUIVALENCE ( COMINT( 586), TFC     )
      EQUIVALENCE ( COMINT( 590),FRICT  )
      EQUIVALENCE ( COMINT( 591), XCF     )
      DO 10 I=1,3
      XCD(I)=0.0
      XC(I)=0.
      FS(I)=0.
      FF(I+3)=0.0
10    FF(I)=0.0
      CONT1=1.570796-ANGLE1
      OMEG3=0.0
      DO 20 J=1,3
```

```
20    OMEG3=TRL(3,J)*XD(1,J+3)+OMEG3
      IF (ABS(OMEG3).GT.1.E-6) GO TO 30
      TF=0.0
      GO TO 40
30    TF=FONTT*AMU*RAV*(-OMEG3/ABS(OMEG3))/12.
      VMIN1=VMIN
      IF (ABS(OMEG3).GT.VMIN1) GO TO 40
      TF=TF/VMIN1*ABS(OMEG3)
40    DO 50 I=1,3
      TFC(I)=TF*TRL(3,I)
50    IF (ABS(TFC(I)).GT.ABS(XD(1,I+3)*XYZI(I)/HZ)) TFC(I)=-XD(1,I+3)*XY
     1ZI(I)/HZ*.9
      IF (ABS(TRL(3,2))+ABS(TRL(3,3)).GT.1.E-10) GO TO 60
      ANGLE2=1.5707963268
      GO TO 70
60    ANGLE2=ATAN2(TRL(3,3),TRL(3,2))
70    IF (ARTSU.EQ.0.) RETURN
      XC(1)=(ARMOM/ARTSU)/12.*SIGN(1.,CONT1)
      IF (ABS(ANGLE).GT.1.E-3) GO TO 80
      R=0.0
      GO TO 90
80    R=ABS(-ZLS/SIN(ANGLE1)-XC(1)/TAN(ANGLE1))
90    XC(2)=R*COS(ANGLE2)
      XC(3)=R*SIN(ANGLE2)
      XCD(1)=XD(1,1)-XD(1,6)*XC(2)+XD(1,5)*XC(3)
      XCD(2)=XD(1,2)+XD(1,6)*XC(1)-XD(1,4)*XC(3)
      XCD(3)=XD(1,3)-XD(1,5)*XC(1)+XD(1,4)*XC(2)
      XPF(1)=(-ZLS*COS(ANGLE)-TORTN/FONTT*SIN(ANGLE)/12.)*(SIGN(1.,CONT1
     1))
      IF (ABS(ANGLE).GT.1.E-5) GO TO 100
      XPF(2)=0.0
      XPF(3)=0.0
      GO TO 110
100   XPF(2)=(-ZLS*SIN(ANGLE)+TORTN/FONTT*COS(ANGLE)/12.)*COS(ANGLE2)
      XPF(3)=(-ZLS*SIN(ANGLE)+TORTN/FONTT*COS(ANGLE)/12.)*SIN(ANGLE2)
110   XPFD(1)=XD(1,1)-XD(1,6)*XPF(2)+XD(1,5)*XPF(3)
      XPFD(2)=XD(1,2)+XD(1,6)*XPF(1)-XD(1,4)*XPF(3)
      XPFD(3)=XD(1,3)-XD(1,5)*XPF(1)+XD(1,4)*XPF(2)
      RSXCF=0.0
      DO 120 I=1,3
      XCF(I)=0.
      XPFX(I)=0.0
      DO 120 J=1,3
      XPFX(I)=TRL(I,J)*XPFD(J)+XPFX(I)
120   XCF(I)=TRL(I,J)*XCD(J)+XCF(I)
      IF ((ABS(XCF(1)).GT.1.E-3).OR.(ABS(XCF(2)).GT.1.E-3)) GO TO 140
      FS(3)=-FONTT
      FS(1)=-XLGRAV*XMASST
      FS(2)=-YLGRAV*XMASST
      DO 130 I=1,2
130   IF (ABS(FS(I)).GT.AMU*FONTT) FS(I)=AMU*FONTT*SIGN(1.,FS(I))
      GO TO 170
140   IF (ABS(XCF(1)).GT.1.E-6) GO TO 150
      ANGLE3=1.5707963268
      IF (XCF(2).GT.0.) ANGLE3=4.7123889804
      GO TO 160
```

```
150   ANGLE3=ATAN2(-XCF(2),-XCF(1))
160   FS(3)=-FONTT
      FS(2)=AMU*FONTT*SIN(ANGLE3)
      FS(1)=AMU*FONTT*COS(ANGLE3)
170   DO 180 I=1,2
      DO 180 J=2,3
      IF (I.EQ.J) GO TO 180
      K=6-I-J
      A(K,K)=1./XMASST+XPF(I)**2/XYZI(J)+XPF(J)**2/XYZI(I)
      A(I,J)=-XPF(I)*XPF(J)/XYZI(K)
180   A(J,I)=A(I,J)
      DO 190 I=1,3
190   FNT(I)=TRL(3,I)*(-FONTT)
      DO 200 I=1,3
      ACELTN(I)=0.0
      DO 200 J=1,3
200   ACELTN(I)=A(I,J)*FNT(J)+ACELTN(I)
      DO 210 I=1,3
      ACTN(I)=0.0
      DO 210 J=1,3
210   ACTN(I)=TRL(I,J)*ACELTN(J)+ACTN(I)
      ACTN(3)=0.
      DO 220 I=1,3
      ACELTN(I)=0.
      DO 220 J=1,3
220   ACELTN(I)=TRL(J,I)*ACTN(J)+ACELTN(I)
      XPFX(3)=0.0
      DO 240 I=1,3
      FRF(I)=0.0
      DO 230 J=1,2
230   FRF(I)=XPFX(J)*TRL(J,I)+FRF(I)
240   AFRF(I,1)=-FRF(I)/HZ-ACELTN(I)
      CALL SOLVE (A,AFRF,SUM,3,1)
      DO 250 I=1,3
      FSS(I)=0.0
      DO 250 J=1,3
250   FSS(I)=TRL(I,J)*AFRF(J,1)+FSS(I)
      DO 260 I=1,2
260   IF (ABS(FS(I)).GT.ABS(FSS(I))) FS(I)=FSS(I)
      IF (NTX.EQ.0) FF(1)=FS(1)*TRL(1,1)+FS(2)*TRL(2,1)+FS(3)*TRL(3,1)
      IF (NTY.EQ.0) FF(2)=FS(1)*TRL(1,2)+FS(2)*TRL(2,2)+FS(3)*TRL(3,2)
      IF (NTZ.EQ.0) FF(3)=FS(1)*TRL(1,3)+FS(2)*TRL(2,3)+FS(3)*TRL(3,3)
      IF (NRX.EQ.0) FF(4)=-FF(2)*XPF(3)+FF(3)*XPF(2)+TFC(1)
      IF (NRY.EQ.0) FF(5)=FF(1)*XPF(3)-FF(3)*XPF(1)+TFC(2)
      IF (NRZ.EQ.0) FF(6)=-FF(1)*XPF(2)+FF(2)*XPF(1)+TFC(3)
      FRICT=SQRT(FS(1)**2+FS(2)**2)
      ANGLD1=ANGLE1*DEGRAD
      ANGLD2=ANGLE2*DEGRAD
      ANGLD3=ANGLE3*DEGRAD
      RETURN
      END
```

APPENDIX B

```
      SUBROUTINE INERT (XYZI)
C     CRUSHABLETORUSINERTIACALCULATION
      DIMENSION W(20),X(20),Y(20),Z(20),WX(20),WY(20),WZ(20),WXX(20),
     1WYY(20),WZZ(20),DIX(20),DIY(20),DIZ(20),WXY(20),WXZ(20),WYZ(20)
      DIMENSION CRD(100), XYZI(15)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 416), XMASST )
      EQUIVALENCE ( COMINT( 445), A      )
      EQUIVALENCE ( COMINT( 446), B      )
      EQUIVALENCE ( COMINT( 447), C      )
      EQUIVALENCE ( COMINT( 448), D      )
      EQUIVALENCE ( COMINT( 449), E      )
      EQUIVALENCE ( COMINT( 450), F      )
      EQUIVALENCE ( COMINT( 451), G      )
      EQUIVALENCE ( COMINT( 452), RHOL   )
      EQUIVALENCE ( COMINT( 453), RHOP   )
      DO 10 I=1,100
10    CRD(I)=0.0
      DO 20 I=1,15
20    XYZI(I)=0.
      CRD(5)=A
      CRD(6)=B
      CRD(7)=C
      CRD(8)=D
      CRD(9)=E
      CRD(10)=F
      CRD(11)=G
      CRD(12)=RHOL
      CRD(13)=RHOP
C     LIMITER VOLUME AND WEIGHT
      VA=4.93*E*F*(G-C+D+0.424*F)
      VB=4.93*D**2*(G-C+0.576*D)
      VC=6.28*D*(E-D)*(G-C+0.5*D)
      VD=4.93*A*B*(G+0.424*A)
      VLE=6.28*B*C*(G-0.5*C)
      VL=(VA+VB+VC-VD-VLE)*2.0
      WL=RHOL*VL/1728.0
C     PAYLOAD VOLUME AND WEIGHT
      VPA=6.28*G**2*B
      VPB=9.87*A*B*(G+0.424*A)
      VP=VPA+VPB
      WP=RHOP*VP/1728.0
      CRD(50)=VA
      CRD(51)=VB
      CRD(52)=VC
      CRD(53)=VD
      CRD(54)=VLE
      CRD(55)=VPA
      CRD(56)=VPB
      DO 30 I=1,20
      W(I)=0.0
      X(I)=0.0
      Y(I)=0.0
      Z(I)=0.0
      WX(I)=0.0
      WY(I)=0.0
```

```
      WZ(I)=0.0
      WXX(I)=0.0
      WYY(I)=0.0
      WZZ(I)=0.0
      DIX(I)=0.0
      DIY(I)=0.0
      WXY(I)=0.0
      WXZ(I)=0.0
      WYZ(I)=0.0
30    DIZ(I)=0.0
      RHOLF=CRD(12)/1728.
      RHOPF=CRD(13)/1728.
      W(1)=RHOLF*CRD(50)*2.
      Z(1)=0.0
      DIZ(1)=((.707*((CRD(11)-CRD(7)+CRD(8)+.828*CRD(10))**2+(CRD(11)-CR
     1D(7)+CRD(8))**2)**.5)**2)*W(1)
      X(1)=0.0
      DIX(1)=((.289*(3.*((CRD(11)-CRD(7)+CRD(8)+.828*CRD(10))**2+(CRD(11
     1)-CRD(7)+CRD(8))**2)+(.424*2.*CRD(9))**2)**.5)**2)*W(1)
      Y(1)=X(1)
      DIY(1)=DIX(1)
      W(2)=RHOLF*CRD(51)
      Z(2)=CRD(9)-.576*CRD(8)
      DIZ(2)=(W(2)/4.)*(4.*(CRD(11)-CRD(7)+.576*CRD(8))**2+3.*(.474*CRD(
     18))**2)
      X(2)=0.0
      DIX(2)=(W(2)/8.)*(4.*(CRD(11)-CRD(7)+.576*CRD(8))**2+5.*(.474*CRD(
     18))**2)
      Y(2)=X(2)
      DIY(2)=DIX(2)
      W(3)=W(2)
      Z(3)=(-1.)*Z(2)
      DIZ(3)=DIZ(2)
      X(3)=0.0
      DIX(3)=DIX(2)
      Y(3)=X(3)
      DIY(3)=DIX(3)
      W(4)=RHOLF*CRD(52)*2.
      Z(4)=0.0
      DIZ(4)=(W(4)/2.)*((CRD(11)-CRD(7)+CRD(8))**2+(CRD(11)-CRD(7))**2)
      X(4)=0.0
      DIX(4)=(W(4)/12.)*(3.*((CRD(11)-CRD(7)+CRD(8))**2+(CRD(11)-CRD(7))
     1**2)+(2.*(CRD(9)-CRD(8)))**2)
      Y(4)=X(4)
      DIY(4)=DIX(4)
      W(5)=RHOLF*CRD(53)*(-2.)
      Z(5)=0.0
      DIZ(5)=(W(5)/2.)*((.848*CRD(5)+CRD(11))**2+CRD(11)**2)
      X(5)=0.0
      DIX(5)=(W(5)/12.)*(3.*((.848*CRD(5)+CRD(11))**2+CRD(11)**2)+(2.*CR
     1D(6))**2)
      Y(5)=X(5)
      DIY(5)=DIX(5)
      W(6)=RHOLF*CRD(54)*(-2.)
      Z(6)=0.0
      DIZ(6)=(W(6)/2.)*(CRD(11)**2+(CRD(11)-CRD(7))**2)
```

```
      X(6)=0.0                                                          INT1130
      DIX(6)=(W(6)/12.)*(3.*(CRD(11)**2+(CRD(11)-CRD(7))**2)+(2.*CRD(6))INT1140
     1**2)                                                              INT1150
      Y(6)=X(6)                                                         INT1160
      DIY(6)=DIX(6)                                                     INT1170
      W(7)=RHOPF*CRD(55)                                                INT1180
      Z(7)=0.0                                                          INT1190
      DIZ(7)=(W(7)/2.)*CRD(11)**2                                       INT1200
      X(7)=0.0                                                          INT1210
      DIX(7)=(W(7)/12.)*(3.*CRD(11)**2+(2.*CRD(6))**2)                  INT1220
      Y(7)=X(7)                                                         INT1230
      DIY(7)=DIX(7)                                                     INT1240
      W(8)=RHOPF*CRD(56)                                                INT1250
      Z(8)=0.0                                                          INT1260
      DIZ(8)=(DIZ(5)/W(5))*W(8)                                         INT1270
      X(8)=0.0                                                          INT1280
      DIX(8)=(DIX(5)/W(5))*W(8)                                         INT1290
      Y(8)=X(8)                                                         INT1300
      DIY(8)=DIX(8)                                                     INT1310
      W(9)=CRD(57)                                                      INT1320
      X(9)=CRD(58)                                                      INT1330
      Y(9)=CRD(59)                                                      INT1340
      Z(9)=CRD(60)                                                      INT1350
      DIX(9)=CRD(61)                                                    INT1360
      DIY(9)=CRD(62)                                                    INT1370
      DIZ(9)=CRD(63)                                                    INT1380
      W(10)=CRD(64)                                                     INT1390
      X(10)=CRD(65)                                                     INT1400
      Y(10)=CRD(66)                                                     INT1410
      Z(10)=CRD(67)                                                     INT1420
      DIX(10)=CRD(68)                                                   INT1430
      DIY(10)=CRD(69)                                                   INT1440
      DIZ(10)=CRD(70)                                                   INT1450
      DO 40 J=1,19                                                      INT1460
      WX(J)=W(J)*X(J)                                                   INT1470
      WY(J)=W(J)*Y(J)                                                   INT1480
      WZ(J)=W(J)*Z(J)                                                   INT1490
      WX(20)=WX(20)+WX(J)                                               INT1500
      WY(20)=WY(20)+WY(J)                                               INT1510
      WZ(20)=WZ(20)+WZ(J)                                               INT1520
      WXX(J)=WX(J)*X(J)                                                 INT1530
      WYY(J)=WY(J)*Y(J)                                                 INT1540
      WZZ(J)=WZ(J)*Z(J)                                                 INT1550
      WXY(J)=WX(J)*Y(J)                                                 INT1560
      WXZ(J)=WX(J)*Z(J)                                                 INT1570
      WYZ(J)=WY(J)*Z(J)                                                 INT1580
      WXY(20)=WXY(20)+WXY(J)                                            INT1590
      WXZ(20)=WXZ(20)+WXZ(J)                                            INT1600
      WYZ(20)=WYZ(20)+WYZ(J)                                            INT1610
      WXX(20)=WXX(20)+WXX(J)                                            INT1620
      WYY(20)=WYY(20)+WYY(J)                                            INT1630
      WZZ(20)=WZZ(20)+WZZ(J)                                            INT1640
      DIX(20)=DIX(20)+DIX(J)                                            INT1650
      DIY(20)=DIY(20)+DIY(J)                                            INT1660
      DIZ(20)=DIZ(20)+DIZ(J)                                            INT1670
   40 W(20)=W(20)+W(J)                                                  INT1680
```

```
      X(20)=WX(20)/W(20)
      Y(20)=WY(20)/W(20)
      Z(20)=WZ(20)/W(20)
      WXB=W(20)*X(20)**2
      WYB=W(20)*Y(20)**2
      WZB=W(20)*Z(20)**2
      DIX(20)=DIX(20)+WYY(20)-WYB+WZZ(20)-WZB
      DIY(20)=DIY(20)+WXX(20)-WXB+WZZ(20)-WZB
      DIZ(20)=DIZ(20)+WYY(20)-WYB+WXX(20)-WXB
      CRD(76)=W(20)
      CRD(77)=X(20)
      CRD(78)=Y(20)
      CRD(79)=Z(20)
      CRD(80)=DIX(20)
      CRD(81)=DIY(20)
      CRD(82)=DIZ(20)
      CRD(83)=WXY(20)
      CRD(84)=WXZ(20)
      CRD(85)=WYZ(20)
      XYZI(1)=DIZ(20)
      XYZI(2)=DIY(20)
      XYZI(3)=DIX(20)
      XYZI(4)=WYZ(20)
      XYZI(5)=WXY(20)
      XYZI(6)=WXZ(20)
      DO 50 I=1,6
50    XYZI(I)=XYZI(I)/(32.174*144.)
      XMASST=W(20)/32.174
      XYZI(13)=Z(20)/12.
      XYZI(14)=Y(20)/12.
      XYZI(15)=X(20)/12.
      RETURN
      END
```

APPENDIX B

```
SUBROUTINE LDSTR
COMMON COMINT( 600)
EQUIVALENCE ( COMINT( 10 ), IFIN   )
EQUIVALENCE ( COMINT( 432), ZDLS   )
EQUIVALENCE ( COMINT( 348), STROKE )
EQUIVALENCE ( COMINT( 349), SPELST )
EQUIVALENCE ( COMINT( 356), SPSTTM )
EQUIVALENCE ( COMINT( 363), SAVST2 )
EQUIVALENCE ( COMINT( 370), SAVST1 )
EQUIVALENCE ( COMINT( 455), FONTT  )
EQUIVALENCE ( COMINT( 456), TORTN  )
EQUIVALENCE ( COMINT( 457), ARTSU  )
EQUIVALENCE ( COMINT( 445), A      )
EQUIVALENCE ( COMINT( 446), B      )
EQUIVALENCE ( COMINT( 447), C      )
EQUIVALENCE ( COMINT( 448), D      )
EQUIVALENCE ( COMINT( 449), E      )
EQUIVALENCE ( COMINT( 450), F      )
EQUIVALENCE ( COMINT( 451), G      )
EQUIVALENCE ( COMINT( 452), RHOL   )
EQUIVALENCE ( COMINT( 453), RHOP   )
EQUIVALENCE ( COMINT( 458), PSTTM  )
EQUIVALENCE ( COMINT( 459), ANGLE  )
EQUIVALENCE ( COMINT( 463), ARMOM  )
EQUIVALENCE ( COMINT( 464), ELST   )
EQUIVALENCE ( COMINT( 465), RDIA   )
EQUIVALENCE ( COMINT( 466), AMU    )
EQUIVALENCE ( COMINT( 467), XKW    )
EQUIVALENCE ( COMINT( 468), DX     )
EQUIVALENCE ( COMINT( 469), DTHE   )
EQUIVALENCE ( COMINT( 470), ACLR   )
EQUIVALENCE ( COMINT( 471), RNORM  )
EQUIVALENCE ( COMINT( 472), LDSTR1 )
EQUIVALENCE ( COMINT( 475), PELST  )
EQUIVALENCE ( COMINT( 559), SAVSTB )
EQUIVALENCE ( COMINT( 560), SAVSTA )
EQUIVALENCE ( COMINT( 585), STREF  )
EQUIVALENCE ( COMINT( 589), RAV    )
EQUIVALENCE ( COMINT( 594), POW    )
EQUIVALENCE ( COMINT( 595), RSHCR  )
STTM=STROKE
BET=ANGLE
TANBET=TAN(BET)
COSBET=COS(BET)
SINBET=SIN(BET)
IF (LDSTR1.GT.0) GO TO 10
SAVSTA=0.0
AAAA=A*A
BBBB=B*B
CCCC=C*C
DDDD=D*D
EEEE=E*E
FFFF=F*F
GGGG=G*G
PSTTM=0.0
PELST=ELST
```

```
      LDSTR1=1
   10 CF1=SQRT(EEEE+FFFF*TANBET*TANBET)
      YOI=CF1-STTM/COSBET+(D-C)*TANBET
      IF (BET.GT..01) GO TO 20
      CLEAR=E-B-STTM
      GO TO 40
   20 IF (BET.LT.1.56) GO TO 30
      IF (BET.GT.1.58) GO TO 30
      CLEAR=F+D-C-A-STTM
      GO TO 40
   30 CLEAR=(YOI*TANBET-TANBET*SQRT(BBBB+AAAA*TANBET*TANBET))/(SINBET*(1
     1.+TANBET*TANBET))
   40 ALCL1=(1.-STREF)*(STTM+CLEAR)
      ACLR=CLEAR-AMAX1(ALCL1,RDIA)
      IF (ACLR.LT.0.0) RETURN
      IF (IFIN.NE.0) GO TO 50
      SPSTTM=PSTTM
      SAVST2=SAVSTA
      SPELST=PELST
      SAVST1=STTM
      SAVSTB=SAVSTA
      SAVSTA=STTM
   50 IF (ABS(STTM-PSTTM).LE.1.E-6) GO TO 60
      IF (STTM.GT.SAVSTB) GO TO 80
C
      IF (PSTTM-STTM.GE.PELST) GO TO 70
      FONTT=(1.-(PSTTM-STTM)/PELST)*PFONTT
      TORTN=(1.-(PSTTM-STTM)/PELST)*PTORTN
      PFONTT=FONTT
      PTORTN=TORTN
      PELST=PELST-PSTTM+STTM
      PSTTM=STTM
      RETURN
   60 FONTT=PFONTT
      TORTN=PTORTN
      RETURN
   70 FONTT=0.0
      TORTN=0.0
      PTORTN=0.0
      PFONTT=0.0
      PSTTM=STTM
      PELST=ELST
      RETURN
   80 FONTT=0.0
      TORTN=0.0
      ARTSU=0.0
      ARMOM=0.0
      I=0.0
      SIGMA=XKW*RHOL/(STREF*144.)
      THE=0.0
   90 THE=THE+DTHE
      COSTHE=COS(THE)
      ARSU=0.0
      FONT=0.0
      X=0.0
      X1=0.0
```

```
      MOUNT=1
      PPSI=TANBET*COSTHE
      PSI=ATAN(PPSI)
      SINPSI=SIN(PSI)
      COSPSI=COS(PSI)
      TANPSI=TAN(PSI)
      IF (BET.GT.1.56) GO TO 270
      STTH=COSPSI*STTM/COSBET+COSPSI*SQRT(EEEE+FFFF*TANPSI*TANPSI)-COSPS
     1I*SQRT(EEEE+FFFF*TANBET*TANBET)+(G+D-C)*COSPSI*TANBET*(COSTHE-1.)
      IF (STTH.LE.0.0) GO TO 330
      IF (THE.GT.3.14) GO TO 330
      FO1=0.0
      FO2=0.0
      TORKN=0.0
      TOR2N=0.0
      ARMOO=0.0
      ARS2=0.0
      ARM2=0.0
      CF1=SQRT(EEEE+FFFF*TANPSI*TANPSI)
      YOI=CF1-STTH/COSPSI+(D-C)*TANPSI
      YCI=YOI+C*TANPSI
      EIC=2.0*EEEE*(D-C)/FFFF+2.*YOI*TANPSI
      EIR=YCI*YOI-EEEE+EEEE*(D-C)**2/FFFF
      EID=CF1*CF1/FFFF
      XOL=(EIC+SQRT(EIC**2-4.0*EID*EIR))/(2.0*EID)
      YOL=YOI-XOL*TANPSI
      IF (THE.GT.DTHE) GO TO 100
      XIL=XOL
  100 IF (YCI-E+D.GT.0.0) GO TO 110
      CONT1=G-C/2.
      FO2=SIGMA*C*DTHE*CONT1*(COSPSI+RNORM*TANPSI*SINPSI)
      TOR2N=FO2*CONT1*COSPSI-FO2*SINPSI*(YOI+YCI)/2.
      ARS2=C*CONT1*DTHE/COSBET
      ARM2=ARS2*(YCI+YOI)/2.
      GO TO 150
  110 YOI=YOI-(D-C)*TANPSI
      IF (YOI.GT.E) GO TO 230
      SXI=DDDD*(1.+TANPSI*TANPSI)-(E-D-YOI+(D-C)*TANPSI)**2
      XIIC=((E-D-YOI+(D-C)*TANPSI)*TANPSI+SQRT(SXI))/(1.+TANPSI**2)-D+C
      YOIC=YOI+XIIC*TANPSI
      IF (YOI.GT.YOIC) GO TO 120
      CONT1=G-XIIC/2.
      FO2=SIGMA*XIIC*DTHE*CONT1*(COSPSI+RNORM*TANPSI*SINPSI)
      TOR2N=FO2*CONT1*COSPSI-FO2*SINPSI*(YOI+XIIC*TANPSI)
      ARS2=XIIC*CONT1*DTHE/COSBET
      ARM2=ARS2*(YOIC+YOI)/2.0
      GO TO 150
  120 IF (XIIC+D-C) 230,150,130
  130 X=X+DX
      IF (X.GE.A) GO TO 330
      CF2=(B/A-A/B)*SQRT(AAAA-X*X)
      CF3=A*SQRT(AAAA-X*X)/(B*X)
      X1=(CF1-STTH/COSPSI+(D-C)*TANPSI-CF2)/(CF3+TANPSI)
      Y1=CF2+CF3*X1
      IF (X1+XIIC) 130,150,140
  140 IF (X.EQ.0.0) GO TO 150
```

```
      CF3=A*SQRT(AAAA-X*X)/(B*X)
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      SINISI=SIN(PHI-PSI)
      COSISI=COS(PHI-PSI)
      TANISI=TAN(PHI-PSI)
      CONT1=G+(X1-XIIC)/2.
      FO2=SIGMA*(X1+XIIC)*CONT1*DTHE*COSISI*(COSISI/COSPSI+SINISI*TANISI
     1*RNORM/COSPSI)
      TOR2N=FO2*CONT1*COSPSI-FO2*SINPSI*(YOIC+Y1)/2.
      ARS2=CONT1*(X1+XIIC)*DTHE/COSBET
      ARM2=ARS2*(YOIC+Y1)/2.0
  150 X=X+DX
      IF (X.GE.A) GO TO 330
      CF2=(B/A-A/B)*SQRT(AAAA-X*X)
      CF3=A*SQRT(AAAA-X*X)/(B*X)
      X1=(CF1-STTH/COSPSI+(D-C)*TANPSI-CF2)/(CF3+TANPSI)
      Y1=CF2+CF3*X1
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      SINISI=SIN(PHI-PSI)
      COSISI=COS(PHI-PSI)
      TANISI=TAN(PHI-PSI)
  160 RAD1=(X1-X)/SIN(PHI)
      RAD2=G+X1
      DPHI=(AAAA*A*B*DX)/((AAAA*(AAAA-X*X)+BBBB*X*X)*SQRT(AAAA-X*X))
      DA=RAD2*(DX/COS(PHI)+RAD1*DPHI)*DTHE
      IF (DA.LT.ABS(Y1*RAD2*DTHE)) GO TO 170
      DA=0.0
  170 DFN=SIGMA*DA*(COSISI+RNORM*TANISI*SINISI)
      FO1=FO1+DFN
      DAS=2.*DA/COSISI*COSPSI/COSBET
      ARSU=ARSU+DAS
      ARMO=DAS*Y1
      ARMOO=ARMOO+ARMO
      DTN=DFN*RAD2*COSPSI-DFN*SINPSI*Y1
      TORKN=TORKN+DTN
      IF (MOUNT.EQ.2) GO TO 200
      IF (YOL.GT.0.0) GO TO 190
      IF ((X+1.01*DX).LT.A) GO TO 150
  180 X=A
      X1=F+D-C
      PHI=1.57
      SINISI=SIN(PHI-PSI)
      COSISI=COS(PHI-PSI)
      TANISI=TAN(PHI-PSI)
      DA=RAD2*Y1*DTHE
      DFN=SIGMA*DA*(COSISI+RNORM*TANISI*SINISI)
      FO1=FO1+DFN
      DAS=2.*DA/SINPSI*COSPSI/COSBET
      ARSU=ARSU+DAS
      ARMO=DAS*Y1/2.0
      ARMOO=ARMOO+ARMO
      DTN=DFN*RAD2*COSPSI-DFN*SINPSI*Y1
      TORKN=TORKN+DTN
      IF (YOL.GE.0.0) GO TO 220
```

APPENDIX B

```
      X=A-DX
      CF2=(A/B-B/A)*SQRT(AAAA-X*X)
      CF3=-A*SQRT(AAAA-X*X)/(B*X)
      X1=(CF1-STTH/COSPSI+(D-C)*TANPSI-CF2)/(CF3+TANPSI)
      Y1=CF2+CF3*X1
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      PHI=PHI+3.1416
      SINISI=SIN(PHI-PSI)
      COSISI=COS(PHI-PSI)
      TANISI=TAN(PHI-PSI)
      RAD1=(X1-X)/SIN(PHI)
      RAD2=G+X1
      DA=-RAD2*Y1*DTHE
      DFN=SIGMA*DA*(COSISI+RNORM*TANISI*SINISI)
      FO1=FO1+DFN
      DAS=2.*DA/SINPSI*COSPSI/COSBET
      ARSU=ARSU+DAS
      ARMO=DAS*Y1
      ARMOO=ARMOO+ARMO
      DTN=DFN*RAD2*COSPSI-DFN*SINPSI*Y1
      TORKN=TORKN+DTN
      MOUNT=2
      IF (XOL-X1) 220,220,210
190   IF ((X+1.01*DX).GE.A) GO TO 180
      IF (XOL-X1) 220,220,150
200   IF (XOL.LE.X1) GO TO 220
210   X=X-DX
      CF2=(A/B-B/A)*SQRT(AAAA-X*X)
      CF3=-A*SQRT(AAAA-X*X)/(B*X)
      X1=(CF1-STTH/COSPSI+(D-C)*TANPSI-CF2)/(CF3+TANPSI)
      Y1=CF2+CF3*X1
      CF3I=1.0/CF3
      PHI=ATAN(CF3I)
      PHI=PHI+3.1416
      SINISI=SIN(PHI-PSI)
      COSISI=COS(PHI-PSI)
      TANISI=TAN(PHI-PSI)
      GO TO 160
220   DNFN=SIGMA*(X1-XOL)*(G+(X1+XOL)/2.0)*DTHE*COSISI*(COSISI/COSPSI+SI
     1NISI*TANISI*RNORM/COSPSI)
      ARNSU=(X1-XOL)*RAD2*DTHE*2./COSBET
      ARNOO=ARNSU*(YOL+Y1)/2.0
      ARMOO=ARMOO-ARNOO+2.0*ARM2
      ARSU=ARSU-ARNSU+2.0*ARS2
      FO1=FO1-DNFN
      FONT=2.*(FO1+FO2)*COSBET/COSPSI
      TORNN=DNFN*RAD2*COSPSI-DNFN*Y1*SINPSI
      TORKN=TORKN-TORNN
      TORKN=2.0*(TORKN+TOR2N)*COSTHE
      TORTN=TORTN+TORKN
      ARTSU=ARTSU+ARSU
      ARMOM=ARMOM+ARMOO
      FONTT=FONTT+FONT
      GO TO 90
230   EIC=2.*EEEE*(D-C)/FFFF+2.*YOI*TANPSI
```

```
      EIR=YOI*YOI-EEEE+EEEE*(D-C)**2/FFFF                              LOD2810
      EID=CF1*CF1/FFFF                                                 LOD2820
      XIEC=(EIC-SQRT(EIC**2-4.0*EID*EIR))/(2.0*EID)                    LOD2830
  240 IF (X1-XIEC) 260,150,250                                         LOD2840
  250 XIIC=-XIEC                                                       LOD2850
      YIEC=YOI-XIEC*TANPSI                                             LOD2860
      YOIC=YIEC                                                        LOD2870
      GO TO 140                                                        LOD2880
  260 X=X+DX                                                           LOD2890
      CF2=(B/A-A/B)*SQRT(AAAA-X*X)                                     LOD2900
      CF3=A*SQRT(AAAA-X*X)/(B*X)                                       LOD2910
      X1=(CF1-STTH/COSPSI+(D-C)*TANPSI-CF2)/(CF3+TANPSI)               LOD2920
      Y1=CF2+CF3*X1                                                    LOD2930
      GO TO 240                                                        LOD2940
  270 H=G+D+F-C                                                        LOD2950
      T=STROKE                                                         LOD2951
      DY=DX                                                            LOD2960
      STTH=H-(H-STTM)/COSTHE                                           LOD2970
      IF (STTH-F) 280,290,290                                          LOD2980
  280 QY=FFFF-(F-H+(H-STTH)/COSTHE)**2                                 LOD2990
      IF (QY.LT.0.0) GO TO 340                                         LOD3000
      Y1MAX=E/F*SQRT(QY)                                               LOD3010
      GO TO 300                                                        LOD3020
  290 XH=(H-STTH)/COSTHE                                               LOD3030
      IF (XH-H+F.GE.0.0) GO TO 280                                     LOD3040
      QY=DDDD-(H-F-XH)**2                                              LOD3050
      IF (QY.LT.0.0) GO TO 340                                         LOD3060
      Y1MAX=E-D+SQRT(QY)                                               LOD3070
  300 Y=0.0                                                            LOD3080
      IF (THE.GT.DTHE) GO TO 310                                       LOD3090
      YIL=Y1MAX                                                        LOD3100
  310 Y=Y+DY                                                           LOD3110
      X=A/B*SQRT(BBBB-Y*Y)                                             LOD3120
      Y1=((H-STTH)/COSTHE-G-X)*AAAA*Y/(BBBB*X)+Y                       LOD3130
      IF (Y1.GE.Y1MAX) GO TO 320                                       LOD3140
      CF3I=B*X/(A*SQRT(AAAA-X*X))                                      LOD3150
      PHI=ATAN(CF3I)                                                   LOD3160
      CPHI=1.5708-PHI                                                  LOD3170
      COSCPI=COS(CPHI)                                                 LOD3180
      DCPHI=BBBB*AAAA*AAAA*DY/(X*(BBBB*BBBB*X*X+AAAA*AAAA*Y*Y))        LOD3190
      RAD1=(H-T)/COSTHE                                                LOD3200
      RAD2=((H-T)/COSTHE-G-X)/COSCPI                                   LOD3210
      DFN=SIGMA*(H-STTH)*DTHE*(DY+((H-STTH)/COSTHE-G-X)*DCPHI)*(1.+RNORMLOD3220
     1*(TAN(CPHI)+TAN(THE)**2/COS(CPHI)))                              LOD3230
      FONT=FONT+DFN*4.0                                                LOD3240
      DAS=4.*RAD1*DTHE*(DY/COSCPI+RAD2*DCPHI)/(COSCPI*COSTHE)          LOD3250
      ARSU=ARSU+DAS                                                    LOD3260
      DNFP=SIGMA*(Y1MAX-Y1)*(H-T)*DTHE*(COS(CPHI)**2+RNORM*(SIN(CPHI)**2LOD3270
     1+COS(CPHI)*TAN(THE)**2))                                         LOD3280
      ARSUP=(Y1MAX-Y1)*(H-STTH)*DTHE*4.0/COSTHE                        LOD3290
      GO TO 310                                                        LOD3300
  320 THEM=ACOS(1.-STTM/H)                                             LOD3310
      FONTT=FONTT+FONT+DNFP                                            LOD3320
      ARTSU=ARTSU+ARSU+ARSUP                                           LOD3330
      IF (THE-THEM) 90,330,330                                         LOD3340
  330 CONTINUE                                                         LOD3350
```

```
      FONTTM=FONTT                                                      LOD3360
      FOFTTM=SIGMA*RSHCR*ARTSU                                          LOD3370
      IF (FONTTM+FOFTTM.EQ.0.0) RETURN                                  LOD3380
      IF ((FONTTM.LT.0.0).OR.(FOFTTM.LT.0.0)) RETURN                    LOD3381
      RFONTT=(FONTTM*FOFTTM)**POW/(FOFTTM**POW+(FONTTM*AMU)**POW)       LOD3390
      FONTT=RFONTT**(1./POW)                                            LOD3400
      TORTN=FONTT*TORTN/FONTTM                                          LOD3410
  340 PFONTT=FONTT                                                      LOD3420
      PTORTN=TORTN                                                      LOD3430
      PELST=ELST                                                        LOD3440
      PSTTM=STTM                                                        LOD3450
      FAC=1.+1.5*(3.14-THE)/3.14                                        LOD3460
      IF (BET.GT.1.56) GO TO 350                                        LOD3470
      IF (BET.GT..04) GO TO 360                                         LOD3480
      RAV=G+XIL/2.0                                                     LOD3490
      GO TO 380                                                         LOD3500
  350 XCA=G+D-C+F-STTM                                                  LOD3510
      R3=YIL/FAC                                                        LOD3520
      GO TO 370                                                         LOD3530
  360 XCA=-ARMOM/(ARTSU*TANBET)+D-C-STTM/SINBET+SQRT(EEEE+FFFF*TANBET*TALOD3540
     1NBET)/TANBET                                                      LOD3550
      R3=(XIL-XCA)/(FAC*COSBET)                                         LOD3560
  370 CONTINUE                                                          LOD3570
      R4=SQRT(GGGG+(G+XCA)**2-2.*G*(G+XCA)*COSTHE)/FAC                  LOD3580
      RAV=(R3+R4)/2.0                                                   LOD3590
  380 CONTINUE                                                          LOD3600
      IF (BET.LT..001) TORTN=0.0                                        LOD3610
      RETURN                                                            LOD3620
      END                                                               LOD3630-
```

APPENDIX C

OPERATING INSTRUCTIONS FOR THE INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM

APPENDIX C

TABLE OF CONTENTS

APPENDIX C

C.1 INTRODUCTION

This program, which has been written for machine computation on the CDC 6400/6600 Computer, provides the capability for establishing inflatable torus configurations meeting specific design constraints. Operation of this program requires initial selection of a payload, bag material, desired velocity capability, and desired load factor. The program then determines torus dimensions, internal pressures, and bag thicknesses providing the required capability and load factor for flat landing. Velocity capability, load factor, and bag stresses for end landing are also determined, and if a bag thickness increase is necessary, the program automatically reruns both flat and end landing calculations with an updated bag thickness and weight. Normally the outer 120° segment of the bag is critical for end landing.

The program will also determine load factor and velocity capability for any desired input unidirectional landing attitude (constant attitude stroking). Flat landing has been found to be critical from both a velocity capability and load factor standpoint. The velocity capability increases and the load factor decreases as landing attitude (BET) is increased from 0° (Flat Landing) to 90° (End Landing).

Dimensional variables used to represent the payload and attenuator are shown in Figure C-1. The payload is a cylinder attached to the torus attenuator by a gimbal ring. Thickness of the torus attenuator is determined by the program in integer number of plys at three locations around the circumference. Properties of a single ply of the bag material as well as desired

APPENDIX C

INFLATABLE TORUS GEOMETRY

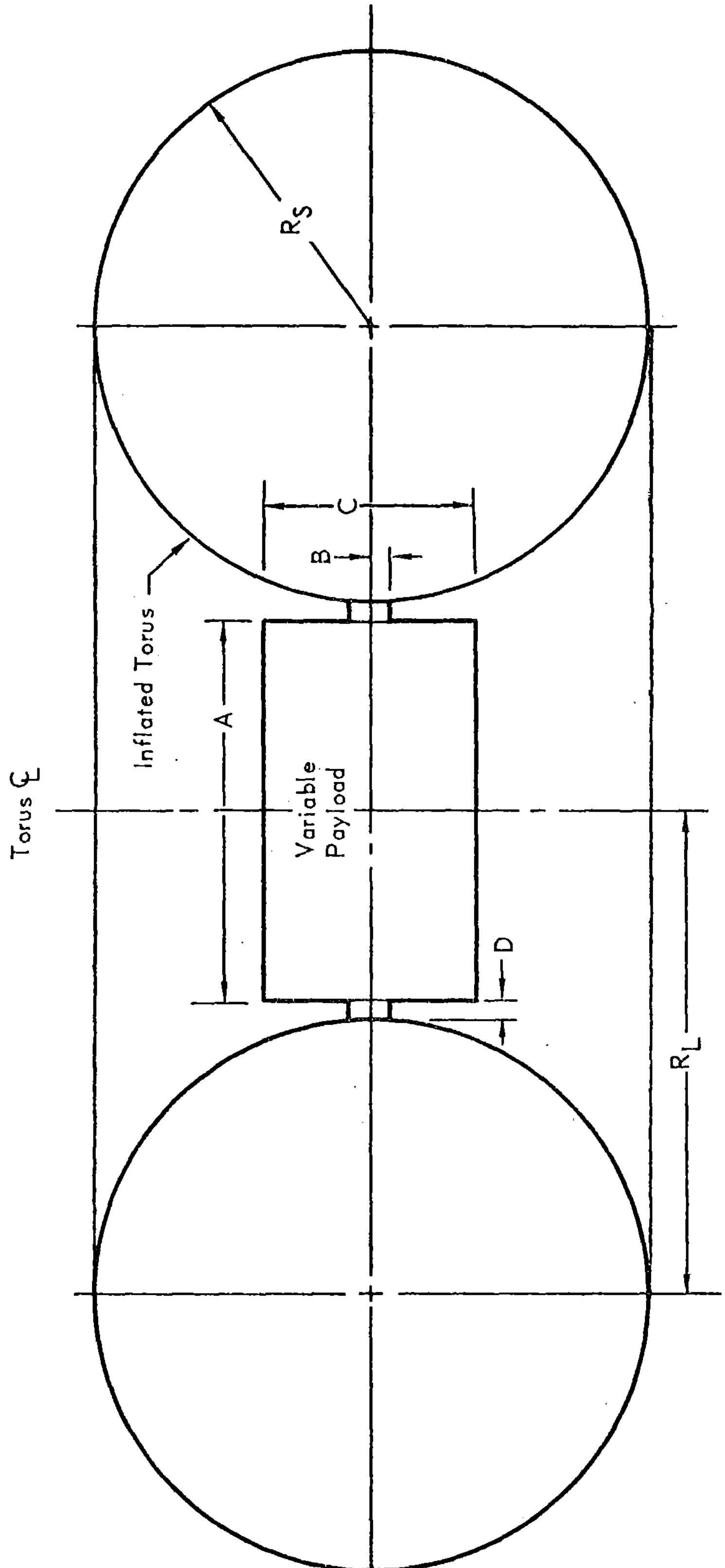


Figure C-1

minimum number of plys at the three locations are input quantities. Any elastomer used to seal the bag which is a function of the number of plys should be included in the weight of one ply. Weight of any scuff protection for the bag, which is a function only of surface area, is a separate input.

Deflected torus shape has been used as the analytical basis for calculation of footprint area, volume change, and torus stresses. Pressure rise is determined based on volume change, assuming a polytropic process. A two spring/mass dynamic model is utilized for a flat landing, and a single spring/mass dynamic model is utilized for landings other than flat. The program allows the user to select the desired minimum number of plys at three locations, and internally calculates the torus thickness required in integer number of plys if minimum ply strength is exceeded.

An inflatable torus drop test program to determine the effects of landing velocity, payload weight, and torus pressure on payload stroke and acceleration under simulated Mars atmospheric pressure was conducted by McDonnell Douglas Astronautics Company Eastern Division under NASA Contract NAS 1-7977 for Langley Research Center. Test results for the drop test program, presented in Reference 2, substantiate the analytical models chosen for the Inflatable Torus Structural Design Computer Program.

Output data include: bag radius; torus radius; inflation pressure; maximum pressure; payload weight; landing system weight; inflation system weight; required number of plys at three locations; maximum running load at three points on the bag; and maximum load factor and velocity capability for flat landing, end landing, and landing at input contact angle. Data output as a function of stroke include: total force normal to the surface; internal bag pressure; footprint area; and velocity capability.

APPENDIX C

C.2 ANALYTICAL PROCEDURES

For oblique landings the footprint area is calculated by applying a semi-empirical correction factor to the theoretical chordal area determined by the intersection of a flat plane with a torus. The chordal area is calculated by summing the area of $D\theta$ strips. The cylindrical coordinate system used in this program utilizes X and Y axes which rotate with the integration angle θ (see Figure C-2). Pressure rise is determined based on volume change, assuming a polytropic process. Normal force on the bag is calculated as the product of footprint area and pressure. The resulting normal force/stroke relationship is then employed in a single nonlinear spring/mass dynamic model used for the oblique landing calculations, consistent with Reference 2. Flat landing is a special case since material is trapped on the inner portion of the torus creating a different deflection pattern and smaller footprint area and consequently a different analytical model (see Figure C-3). Deflected torus bag shape has been used as the analytical basis for calculation of end landing torus stresses (see Figure C-4).

Symbols, units, recommended range, and definition of required input parameters are given in Figure C-5. Output parameters are defined in Figure C-6. Format for machine input is discussed in Section C.4.

Major assumptions employed in the analytical model derivation are:

1. Deflected bag shape.
2. Linearization of flat landing load-stroke to an effective K for use

CYLINDRICAL COORDINATE SYSTEM

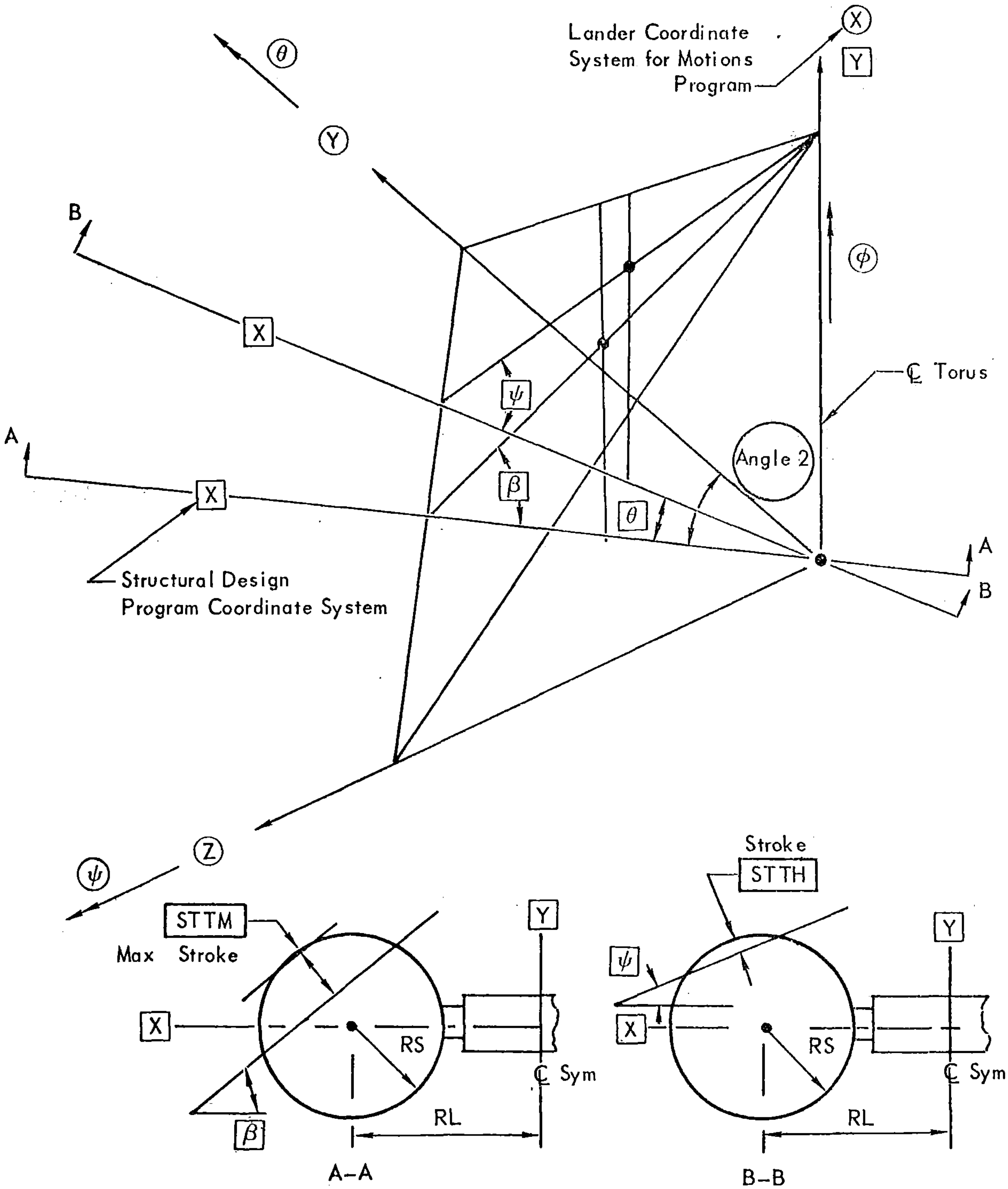
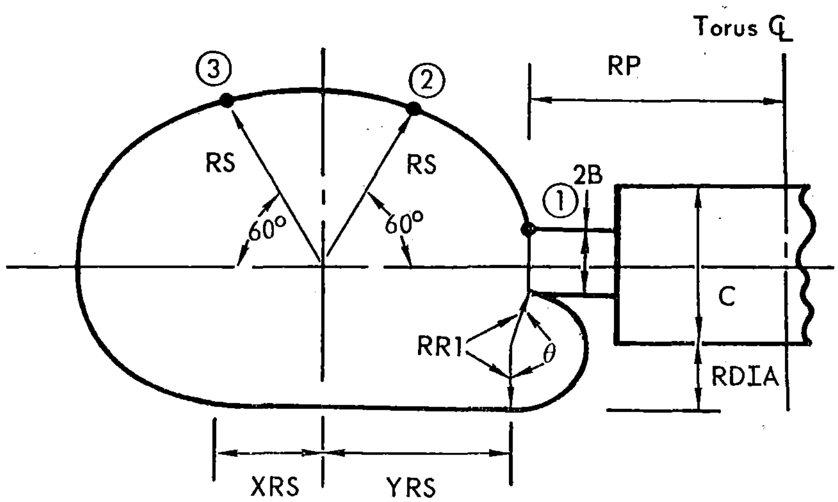


Figure C-2

APPENDIX C

FLAT LANDING ANALYTICAL MODEL



$$XRS = 0.0053 + 0.6901 * (S/RS) - 0.525 * (S/RS)^2$$

$$YRS = 0.0049 + 2.409 * (S/RS) - 1.171 * (S/RS)^2$$

$$VOLR = 0.9983 + 0.0485 * (S/RS) + 0.2399 * (S/RS)^2$$

$$PF = PI * VOLR$$

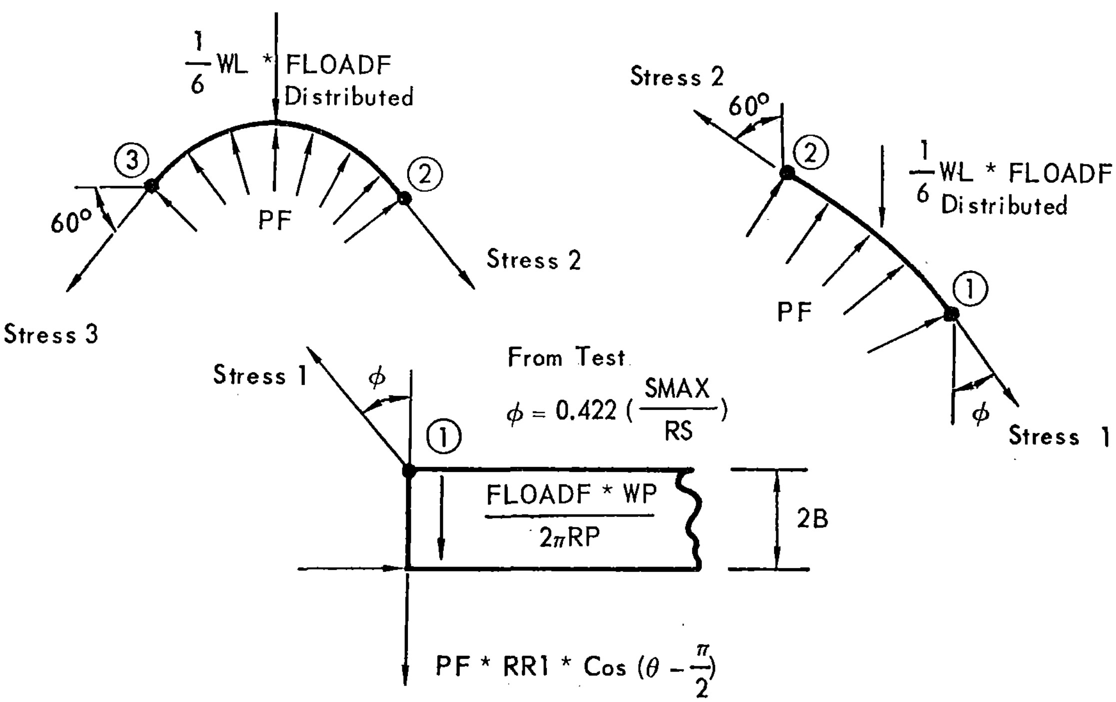
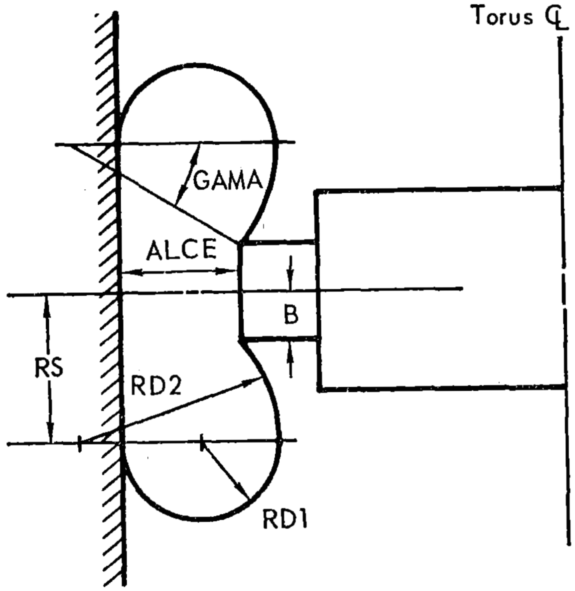


Figure C-3

DEFLECTED TORUS SHAPE – END LANDING TORUS STRESSES

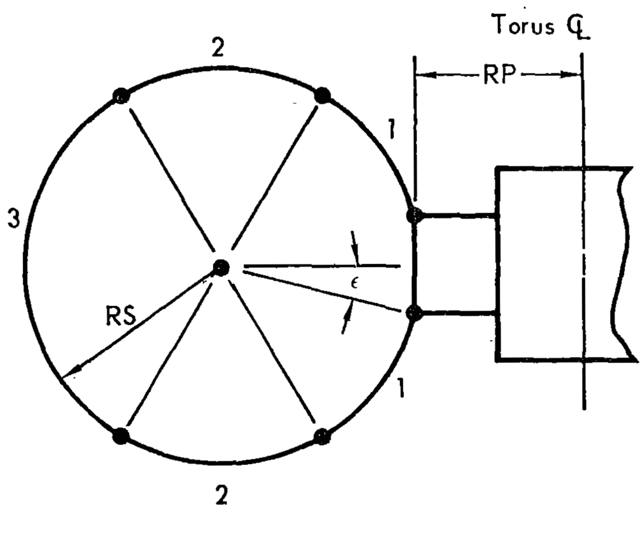


$$RD1 = \frac{(RS-B)\,(1-\mathrm{Cos}\,(GAMA))}{2\,\mathrm{Sin}\,(GAMA)} + \frac{ALCE}{2}$$

$$RD2 = (RS-B)/\mathrm{Sin}\,(GAMA)$$

$$GAMA = \frac{\pi(RS-RD1) - RS - B}{RD2}$$

$$ALCE = 0.212 * RS + 0.894 * RDIA$$
(Limit Velocity)



$$\sigma_{1_B} = \frac{P * RS * (RP + RS * \frac{\mathrm{Cos}\,\epsilon}{2}) * BFS}{RP}$$

$$\sigma_{1_A} = \frac{P * RD2 * (RP + ALCE/2 - RD1) * BFS}{RP}$$

σ_1 = Maximum of σ_{1A} or σ_{1B}

$$\sigma_{2_A} = \frac{P * RD1 * (RP + ALCE - 1.5 * RD1) * BFS}{RP + ALCE - 2.0 * RD1}$$

$$\sigma_{2_B} = \frac{P * RS * (RP + 0.75 * RS) * BFS}{RP + 0.50 * RS}$$

σ_2 = Maximum of σ_{2A} or σ_{2B}

$$\sigma_{3_A} = P * RD1 * \left(1.0 - \frac{RD1}{2.0\,(RP + ALCE)}\right) * BFS$$

$$\sigma_{3_B} = \frac{P * RS * (RP + 1.25 * RS) * BFS}{RP + 1.50\,RS}$$

σ_3 = Maximum of σ_{3A} or σ_{3B}

Subscript :

A – Meridional Stresses, Deformed Portion of Torus
B – Meridional Stresses, Undeformed Portion of Torus

Figure C-4

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM INPUT PARAMETERS

SYMBOL	UNITS	RECOMMENDED RANGE	DEFINITION
A, C	In.	–	Payload Dimensions (See Figure 1)
B, D	In.	–	Payload Gimbal Ring Dimensions (See Figure 1)
RHOP	Lb/Ft^3	–	Payload Density (Including Insulation and Gimbal Ring)
VF	Ft/Sec	–	Desired Limit Velocity Capability
FMAXLF	Earth g's	–	Desired Maximum Limit Load Factor
RDIA	In.	0.0 → 5.0	Rock Diameter
RS1, RS2, RS3	In.	–	3 Different Assumed Values for Bag Radius Used for Interpolation
FABWT	Lb/Sq Ft	–	Bag Material Weight for One Ply
FABST	Lb/In.	–	Bag Material Strength for One Ply
PLY1	–	1.0 → 5.0	Desired Minimum Number of Plys Inner 120° (Payload Attach)
PLY2	–	1.0 → 5.0	Desired Minimum Number of Plys Upper and Lower Center 60°
PLY3	–	1.0 → 5.0	Desired Minimum Number of Plys Outer 120°
BFS	–	1.5 → 3.0	Bag Factor of Safety (Ultimate/Limit)
SCFWT	Lb/Ft^2	0.0 → 0.20	Weight of Scuff Protection (Uniform on Entire Bag)
PA	$Lb/In.^2$	–	Ambient Pressure
BET	Radians	0 → +1.57	Angle Between Crush Plane and X Axis

Figure C-5

APPENDIX C

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM OUTPUT PARAMETERS

SYMBOL	UNITS	DEFINITION
RS	In.	Required Bag Radius
RL	In.	Required Radius from Torus Center to Bag Center
PI	PSI	Required Inflation Pressure (Absolute)
PF	PSI	Limit Bag Pressure at Limit Stroke (Absolute)
FLOADF	Earth g's	Maximum Limit Load Factor at Limit Stroke
WP	Lb	Payload Weight
WL	Lb	Weight of Landing System Including Gas, Excluding Bottle
WIS	Lb	Weight of Inflation System, Bottle and Gas
PLY1F	–	Required Number of Plys Inner 120° (Payload Attach)
PLY2F	–	Required Number of Plys Upper and Lower Center 60°
PLY3F	–	Required Number of Plys Outer 120°
WL1, WL2, WL3	Lb	Landing System Weight, Bag RS1, RS2, and RS3
WIS1, WIS2, WIS3	Lb	Inflation System Weight, Bag RS1, RS2, and RS3
PI1, PI2, PI3	PSI	Inflation Pressure, Bag RS1, RS2, and RS3 (Absolute)
PF1, PF2. PF3	PSI	Pressure at Maximum Limit Stroke, Bag RS1, RS2 and RS3 (Absolute)
FLOAD1, FLOAD2, FLOAD3	Earth g's Earth g's Earth g's	Maximum Limit Load Factor, Bag RS1 Maximum Limit Load Factor, Bag RS2 Maximum Limit Load Factor, Bag RS3
S	In.	Stroke
FORCE	Lb	Total Load Normal to Surface (Limit)
DIST	In.	Y Distance to Centroid of Footprint Area
P	PSI	Limit Bag Pressure at Stroke S (Absolute)
AF	Sq In.	Footprint Area
CLEAR	In.	Payload to Ground Clearance for Impact at Angle BET
ALCL	In.	Allowable Clearance Between Payload and Ground Allowing 11.8% Stroke Margin for Ultimate Velocity Capability (at Angle BET)
U	In.-Lb	Total Energy Absorbed for Landing at Angle BET
VELBT	Ft Sec	Limit Velocity Capability for Landing at Angle BET
CLEARE	In.	Payload to Ground Clearance for End Landing
UE	In.-Lb	Total Energy Absorbed for End Landing
VELE	Ft Sec	Limit Velocity Capability for End Landing
GAMA	Radians	Bag Angle with Local Horizontal at Payload Attach, End Landing
RD1	In.	Outer Radius at Maximum Stroke Point, End Landing
RD2	In.	Radius at Payload Attach Point, End Landing
ALCE	In.	Allowable Clearance Between Payload and Ground, End Landing, Allowing 11.8% Stroke Margin
EGSTR1	Lb/In.	Maximum Running Load for End Landing, Inner 120°
EGSTR2	Lb/In.	Maximum Running Load for End Landing, Upper and Lower Center 60°
EGSTR3	Lb/In.	Maximum Running Load for End Landing, Outer 120°

Figure C-6

in the two spring/mass model. (This assumption is not employed in the non-flat calculations.)

3. Two spring/mass dynamic model for flat landings.
4. One spring/mass dynamic model for non-flat landings.
5. Stroke is provided for ultimate energy (1.25 times limit energy). Limit stroke is equal to ultimate stroke divided by 1.118.
6. Insulation on payload is external to Dimension C (although included in payload weight) and may be violated by a rock for ultimate energy landing, provided payload itself is not contacted.
7. Bag is designed for ultimate pressures (pressure at limit stroke times bag factor of safety) and effect of ultimate load factor (load factor at limit stroke times bag factor of safety).
8. Bag thickness is comprised of an integer number of plys (at three locations) and no less than the input minimum number of plys.
9. Rigid body payload.
10. Gimbal is included in payload weight.
11. Infinitely rigid surface - no protuberances or depressions other than rock diameter.
12. Friction does not influence force normal to footprint area.
13. Bag material is inextensible.

The payload gimbal ring is defined by Dimensions B and D, with the resultant volume being multiplied by payload density (RHOP) to establish gimbal weight, which is considered as part of the payload weight in the program. Payload insulation is assumed to be external to Dimension C (although insulation weight is included in payload weight) and may be violated by a rock for an ultimate energy (1.25 times limit energy) landing, provided the payload itself is not contacted. Limit stroke (maximum) for flat landing is:

SMAX = (RS - C/2 - RDIA)/1.118

RS = Bag Radius

C = Payload Height

RDIA = Rock Diameter

Insulation thickness would range between two to three inches using balsa wood. Optimum payload shape depends on many factors, but generally has been found to occur when A/C is between three and five.

The values selected for fabric weight (FABWT) and fabric strength (FABST) depend on the torus size and manufacturing constraints on the number of plys. A thin material (low FABWT and FABST) will result in a lower weight but may require an excessive number of plys. Values used for FABWT for studies run to date vary between 0.017 lb/ft^2 and 0.077 lb/ft^2 with a corresponding range in FABST of 100 lb/in. to 454 lb/in. respectively. FABST should include only seam efficiency factors, since required factor of safety is input independently (BFS). These values are consistent with HT-1 fabric (Nomex). Abrasion and puncture protection and torus seal are obtained by uniformly coating the torus with an elastomer (silicone rubber compound of the methylphenyl type). Typical weight (SCFWT) is 0.132 lb/ft^2. Bag factor of safety (BFS) is the factor determining the required limit to ultimate strength relationship.

APPENDIX C

The desired minimum number of plys at three locations is input for PLY 1, PLY 2, and PLY 3. The values chosen will depend on fabric selected, expected surface features, manufacturing methods, and payload attachment method.

Ambient planet pressure at the surface (PA) is entered in pounds/square inch. The value specified in Reference 1 for the master agreement of 5 mb is equivalent to .0725 psi. If nitrogen gas is used to inflate the torus, a value of 1.40 should be entered for the constant determining the polytropic gas process (CK). The value of CK for most gases would range between 1.3 and 1.87. If it is desired to simulate an isothermal process for any gas, a value of CK of 1.0 is entered.

C.3 PROGRAM OPERATION

Input format for the structural design portion of the program is shown in Figure C-7. Data may be located anywhere within the eight spaces indicated for each parameter. Dimensions RS1, RS2, and RS3 may be any reasonable estimate of required torus radius. Reference 2 may be used to establish these estimates. The program interpolates for final geometry based on solutions for these values. Input data cards are placed behind the program as shown in Section C.4.

The program calculates velocity capability and load factor for unidirectional input angle BET, which may be any value between 0.0 and 1.57 radians. This angle is maintained constant throughout stroke for these calculations.

Sample output data are shown in Figure C-8. Refer to Figure C-6 or comment cards in the program for nomenclature and units of output.

Program termination is automatic when all output data have been calculated for all input data. The program goes to the input data cards for additional cases until it either finds none or exceeds a specified time limit. Typical machine time for two cases is 35 seconds central processor and 20 seconds peripheral processor. Core size required is 45 K (octal) for compilation of the program.

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM
EXAMPLE INPUT DATA

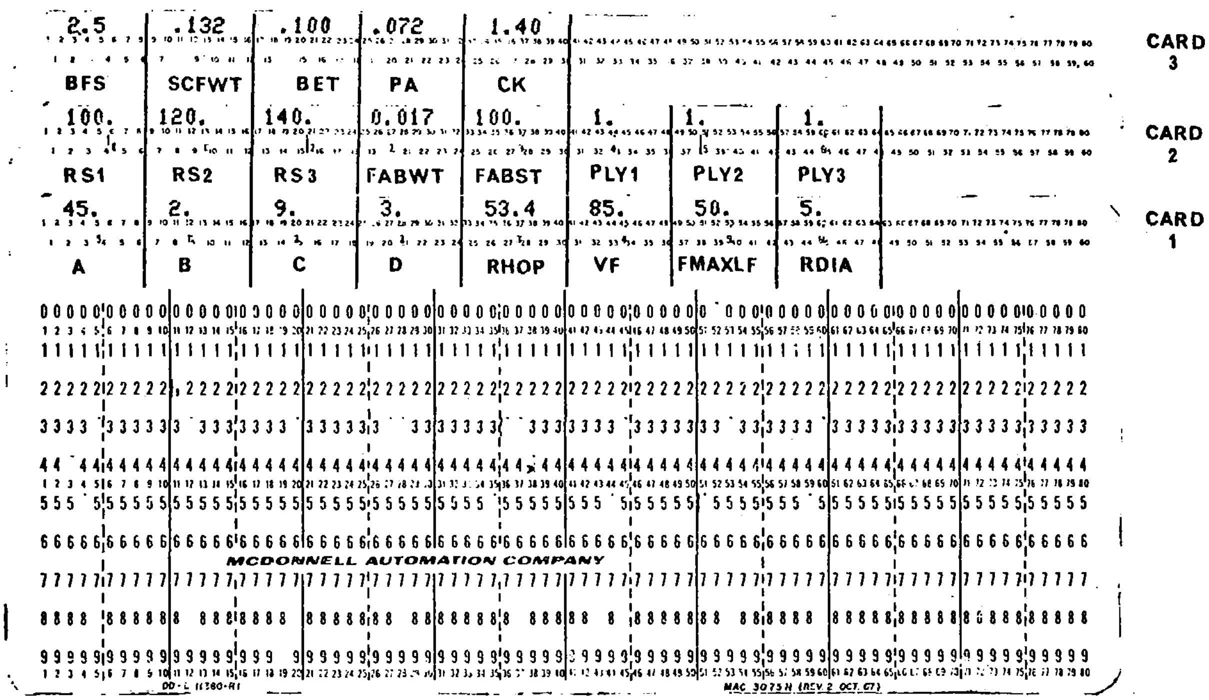


Figure C-7

APPENDIX C

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM
Example Output Data

```
INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM

INPUT PARA ETERS

A=   45.00   H=   2.00   C=   9.00   D=   3.00   RHOP=  53.4   VF=   85.0   FMAXLE=   50.0   RDIA=   5.00
WS1=  100.00   WS2=  120.00   WS3=  140.00   FABWT=   .0170   FABST=   100.0   PLY1= 1.0   PLY2= 1.0   PLY3= 1.0
WFS= 2.500   SCFWT=   .1320   BET=  .100   PA=    .072   CK=1.4000
OUTPUT DATA

WS=   99.99   WL=   116.49   PI=    .37   PF=    .44   FLOADF=    49.9   WP=   498.0   WL=   517.8   WIS=   26.2
PLY1F= 5.0   PLY2F= 3.0   PLY3F= 2.0   STRES1=   411.3   STRES2=   269.2   STRES3=   42.6
WL1=   573.1   PI1=    .30   PF1=    .36   FLOAD1=    46.4   WIS1=   28.2
WL2=   767.9   PI2=    .22   PF2=    .27   FLOAD2=    40.6   WIS2=   34.9
WL3=  1006.1   PI3=    .18   PF2=    .22   FLOAD3=    36.2   WIS3=   43.8
              LOAD STROKE FOR END LANDING (BET = 90 DEGREES)
CLEARE=   23.756   UE=  2276607.4   VELE=  113.98   GAMA=   .7800   RD1=   30.17   RD2=   126.53   ALCE=  23.759
EGSTW1=   337.2   EGSTR2=   160.0   EGSTR3=   104.1
S=   3.955   FORCE=   460.9   DIST=   0.000   P=   .366   AF=   1571.5   VELE=   2.2
S=   7.909   FORCE=   944.7   DIST=   0.000   P=   .366   AF=   3286.6   VELE=   4.4
S=  11.864   FORCE=  1485.4   DIST=   0.000   P=   .366   AF=   5052.7   VELE=   6.7
S=  15.819   FORCE=  2016.4   DIST=   0.000   P=   .367   AF=   6844.2   VELE=   9.1
S=  19.774   FORCE=  2563.1   DIST=   0.000   P=   .368   AF=   8675.6   VELE=  11.4
S=  23.728   FORCE=  3133.9   DIST=   0.000   P=   .369   AF=  10570.4   VELE=  13.8
S=  27.683   FORCE=  3721.6   DIST=   0.000   P=   .370   AF=  12500.6   VELE=  16.3
S=  31.638   FORCE=  4317.2   DIST=   0.000   P=   .372   AF=  14395.7   VELE=  18.7
S=  35.593   FORCE=  4904.5   DIST=   0.000   P=   .373   AF=  16300.5   VELE=  21.2
S=  39.547   FORCE=  5518.1   DIST=   0.000   P=   .375   AF=  18223.8   VELE=  23.7
S=  43.502   FORCE=  6150.1   DIST=   0.000   P=   .377   AF=  20168.4   VELE=  26.2
S=  47.457   FORCE=  6802.3   DIST=   0.000   P=   .380   AF=  22133.8   VELE=  28.8
S=  51.412   FORCE=  7475.8   DIST=   0.000   P=   .382   AF=  24118.0   VELE=  31.3
S=  55.366   FORCE=  8171.5   DIST=   0.000   P=   .385   AF=  26117.7   VELE=  33.9
S=  59.321   FORCE=  8890.0   DIST=   0.000   P=   .389   AF=  28128.4   VELE=  36.6
```

Figure C-8

APPENDIX C

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM

Example Output Data

(Continued)

S=	63.276	FORCE=	9651.3	DIST=	0.000	P=	.392	AF= 30143.7	VELF= 39.3
S=	67.231	FORCE=	10344.9	DIST=	0.000	P=	.396	AF= 32154.7	VELF= 42.0
S=	71.185	FORCE=	11178.3	DIST=	0.000	P=	.400	AF= 34147.6	VELF= 44.7
S=	75.140	FORCE=	11975.4	DIST=	0.000	P=	.404	AF= 36098.1	VELF= 47.5
S=	79.195	FORCE=	12767.0	DIST=	0.000	P=	.409	AF= 37944.2	VELF= 50.2
S=	83.050	FORCE=	13556.5	DIST=	0.000	P=	.414	AF= 39548.9	VELF= 53.1
S=	87.004	FORCE=	144 7.9	DIST=	0.000	P=	.419	AF= 41532.3	VELF= 56.0
S=	90.959	FORCE=	15326.9	DIST=	0.000	P=	.425	AF= 43456.7	VELF= 58.9
S=	94.914	FORCE=	16243.8	DIST=	0.000	P=	.431	AF= 45262.5	VELF= 61.8
S=	98.869	FORCE=	16948.1	DIST=	0.000	P=	.438	AF= 46486.6	VELF= 64.8
S=	102.823	FORCE=	18044.4	DIST=	0.000	P=	.445	AF= 48452.8	VELF= 67.8
S=	106.778	FORCE=	19125.5	DIST=	0.000	P=	.452	AF= 50347.1	VELF= 70.8
S=	110.733	FORCE=	20144.2	DIST=	0.000	P=	.460	AF= 52043.4	VELF= 73.9
S=	114.687	FORCE=	21042.2	DIST=	0.000	P=	.469	AF= 53152.0	VELF= 77.0
S=	118.642	FORCE=	22345.2	DIST=	0.000	P=	.478	AF= 55142.8	VELF= 80.2
S=	122.597	FORCE=	23678.5	DIST=	0.000	P=	.487	AF= 56962.0	VELF= 83.4
S=	126.552	FORCE=	24574.1	DIST=	0.000	P=	.497	AF= 57824.3	VELF= 86.6
S=	130.506	FORCE=	26029.0	DIST=	0.000	P=	.508	AF= 59730.8	VELF= 89.9
S=	134.461	FORCE=	27566.9	DIST=	0.000	P=	.520	AF= 61668.1	VELF= 93.3
S=	138.416	FORCE=	28858.4	DIST=	0.000	P=	.532	AF= 62778.3	VELF= 96.7
S=	142.371	FORCE=	30050.3	DIST=	0.000	P=	.545	AF= 63566.1	VELF= 100.1
S=	146.325	FORCE=	31440.9	DIST=	0.000	P=	.559	AF= 64720.8	VELF= 103.6
S=	150.280	FORCE=	32441.6	DIST=	0.000	P=	.574	AF= 64735.9	VELF= 107.0
S=	154.235	FORCE=	33544.3	DIST=	0.000	P=	.589	AF= 65053.1	VELF= 110.4
S=	158.190	FORCE=	353 8.9	DIST=	0.000	P=	.605	AF= 66332.9	VELF= 114.0

LOAD STROKE FOR IMPACT AT CONSTANT ANGLE BET

CLEAR= 14.633 ALCL= 14.634 U= 1683394.1 VELBT= 100.16

S=	2.031	FORCE=	674.1	DIST=	89.078	P=	.366	AF= 2128.3	VELRT= 1.8
S=	4.063	FORCE=	1243.5	DIST=	87.615	P=	.366	AF= 4374.2	VELRT= 3.7
S=	6.094	FORCE=	1977.3	DIST=	86.168	P=	.366	AF= 6731.9	VELRT= 5.6
S=	8.126	FORCE=	2742.0	DIST=	84.728	P=	.367	AF= 9185.2	VELRT= 7.5

Figure C-8 (continued)

APPENDIX C
INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM
Example Output Data
(Continued)

S=	10.157	FORCE=	3468.0	DIST=	83.311	P=	.367	AF=	11766.3	VELRT=	9.5
S=	12.189	FORCE=	4278.3	DIST=	81.916	P=	.368	AF=	14479.8	VELRT=	11.5
S=	14.220	FORCE=	5138.3	DIST=	80.544	P=	.369	AF=	17338.9	VELRT=	13.6
S=	16.252	FORCE=	6057.6	DIST=	79.201	P=	.370	AF=	20369.0	VELRT=	15.7
S=	18.283	FORCE=	7058.6	DIST=	77.913	P=	.371	AF=	23637.1	VELRT=	17.9
S=	20.314	FORCE=	8166.8	DIST=	76.700	P=	.373	AF=	27218.1	VELRT=	20.1
S=	22.346	FORCE=	9441.3	DIST=	75.646	P=	.374	AF=	31358.8	VELRT=	22.5
S=	24.377	FORCE=	11131.4	DIST=	74.918	P=	.376	AF=	36658.3	VELRT=	25.0
S=	26.409	FORCE=	12450.5	DIST=	73.508	P=	.378	AF=	40703.7	VELRT=	27.6
S=	28.440	FORCE=	13642.5	DIST=	71.872	P=	.381	AF=	44242.9	VELRT=	30.2
S=	30.472	FORCE=	14773.5	DIST=	70.127	P=	.384	AF=	47494.4	VELRT=	32.8
S=	32.503	FORCE=	15870.2	DIST=	68.316	P=	.386	AF=	50544.7	VELRT=	35.4
S=	34.535	FORCE=	16947.6	DIST=	66.462	P=	.390	AF=	53440.7	VELRT=	38.0
S=	36.566	FORCE=	18015.4	DIST=	64.578	P=	.393	AF=	56211.5	VELRT=	40.6
S=	38.597	FORCE=	19080.7	DIST=	62.671	P=	.397	AF=	58876.7	VELRT=	43.2
S=	40.629	FORCE=	20148.9	DIST=	60.747	P=	.400	AF=	61450.6	VELRT=	45.8
S=	42.660	FORCE=	21224.3	DIST=	58.810	P=	.404	AF=	63943.5	VELRT=	48.4
S=	44.692	FORCE=	22310.8	DIST=	56.862	P=	.409	AF=	66363.4	VELRT=	51.0
S=	46.723	FORCE=	23411.6	DIST=	54.906	P=	.413	AF=	68716.4	VELRT=	53.6
S=	48.755	FORCE=	24529.6	DIST=	52.942	P=	.418	AF=	71007.4	VELRT=	56.3
S=	50.786	FORCE=	25667.6	DIST=	50.973	P=	.423	AF=	73240.9	VELRT=	58.9
S=	52.818	FORCE=	26828.1	DIST=	48.999	P=	.428	AF=	75418.1	VELRT=	61.5
S=	54.849	FORCE=	28013.6	DIST=	47.021	P=	.434	AF=	77563.4	VELRT=	64.2
S=	56.880	FORCE=	29226.4	DIST=	45.039	P=	.440	AF=	79618.1	VELRT=	66.8
S=	58.912	FORCE=	30469.0	DIST=	43.054	P=	.446	AF=	81643.8	VELRT=	69.5
S=	60.943	FORCE=	31743.6	DIST=	41.066	P=	.452	AF=	83621.7	VELRT=	72.2
S=	62.975	FORCE=	33052.7	DIST=	39.076	P=	.459	AF=	85552.7	VELRT=	74.9
S=	65.006	FORCE=	34398.7	DIST=	37.084	P=	.466	AF=	87437.7	VELRT=	77.6
S=	67.038	FORCE=	35784.1	DIST=	35.090	P=	.473	AF=	89277.0	VELRT=	80.3
S=	69.069	FORCE=	37211.3	DIST=	33.095	P=	.481	AF=	91071.0	VELRT=	83.1
S=	71.101	FORCE=	38682.9	DIST=	31.098	P=	.489	AF=	92819.9	VELRT=	85.9
S=	73.132	FORCE=	40201.7	DIST=	29.100	P=	.498	AF=	94523.6	VELRT=	88.7
S=	75.163	FORCE=	41770.5	DIST=	27.101	P=	.507	AF=	96182.1	VELRT=	91.5
S=	77.195	FORCE=	43392.1	DIST=	25.101	P=	.516	AF=	97795.3	VELRT=	94.4
S=	79.226	FORCE=	45069.7	DIST=	23.100	P=	.526	AF=	99362.4	VELRT=	97.2
S=	81.258	FORCE=	46806.5	DIST=	21.099	P=	.536	AF=	100883.3	VELRT=	100.2

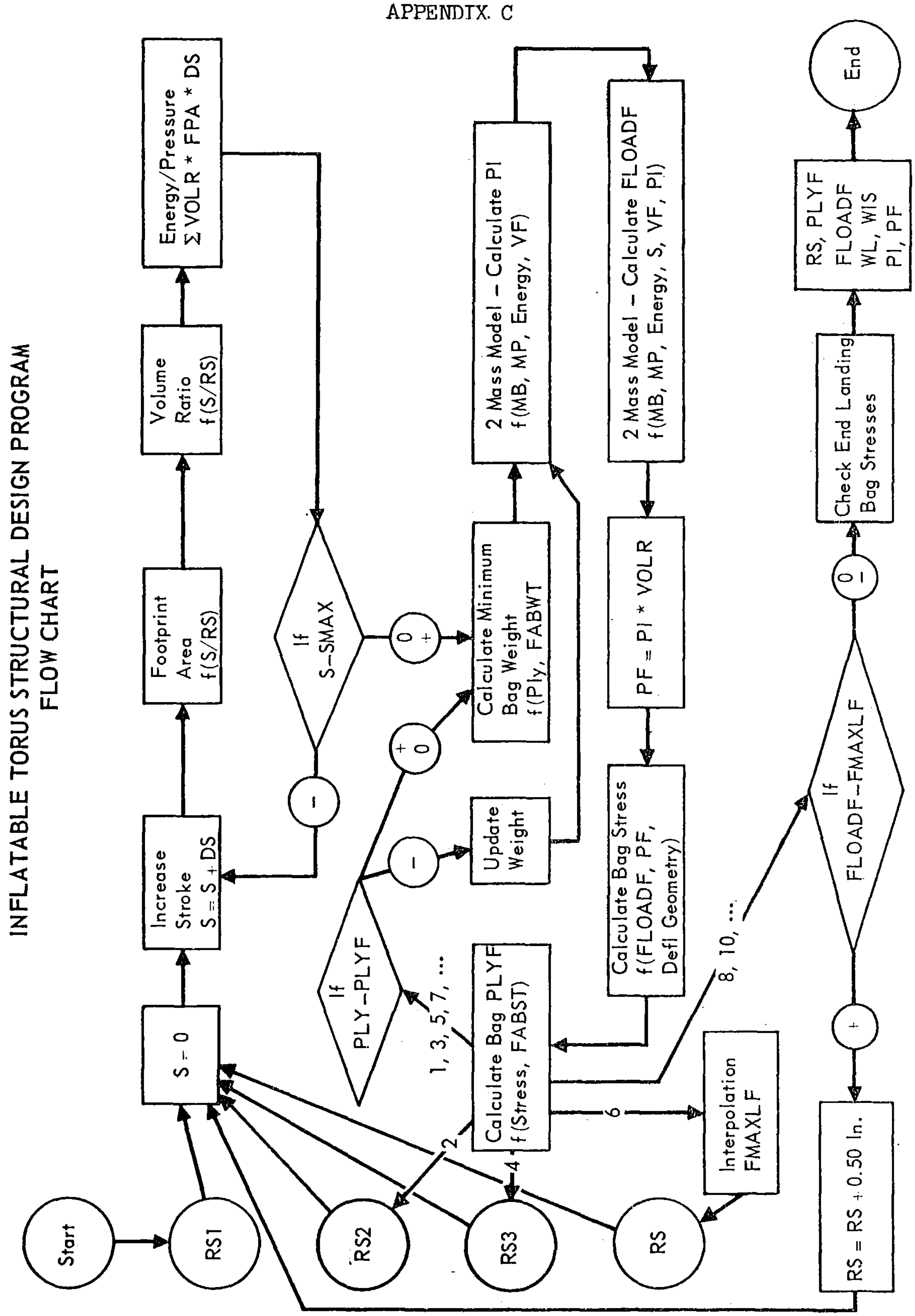
Figure C-8 (continued)

APPENDIX C

C.4 PROGRAM DESCRIPTION

The flow chart shown in Figure C-9 shows major events in the structural design program. Comment cards in the program additionally aid in identifying major events.

The listing for the Inflatable Torus Structural Design Program is presented on the following pages.



INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM
FLOW CHART
Start
RS1
RS2
RS3
RS
S = 0
Increase Stroke S = S + DS
Footprint Area f(S/RS)
Volume Ratio f(S/RS)
Energy/Pressure Σ VOLR * FPA * DS
If S–SMAX
–
0 +
Calculate Minimum Bag Weight f(Ply, FABWT
2 Mass Model – Calculate PI f(MB, MP, Energy, VF)
2 Mass Model – Calculate FLOADF f(MB, MP, Energy, S, VF, PI)
PF = PI * VOLR
Calculate Bag Stress f(FLOADF, PF, Defl Geometry)
Calculate Bag PLYF f(Stress, FABST)
1, 3, 5, 7, ...
2
4
6
8, 10, ...
If PLY–PLYF
+ 0
–
Update Weight
Interpolation FMAXLF
If FLOADF–FMAXLF
+
0 –
RS = RS + 0.50 In.
Check End Landing Bag Stresses
RS, PLYF FLOADF WL, WIS PI, PF
End

Figure C-9

APPENDIX C

INFLATABLE TORUS STRUCTURAL DESIGN PROGRAM

```
      PROGRAM D1R (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION ASAVE(100),BSAVE(100),CSAVE(100),DSAVE(100),ESAVE(100)
     1,SSAVE(100)
      DIMENSION FSAVE(100),GSAVE(100),HSAVE(100),PSAVE(100),RSAVE(100)
     1,QSAVE(100)
10    READ (5,330) A,B,C,D,RHOP,VF,FMAXLF,RDIA
      READ (5,330) RS1,RS2,RS3,FABWT,FABST,PLY1,PLY2,PLY3
      READ (5,330) BFS,SCFWT,BET,PA,CK
      WRITE (6,320)
C
C     PROGRAM DETERMINES REQUIRED TORUS BAG RADIUS AND INFATION
C     PRESSURE FOR A DISIRED VELOCTIY CAPABILITY AND MAXIMUM
C     LOAD FACTOR
C     INPUT PARAMETERS
C        A      PAYLOAD DIAMETER  (IN.)
C        B      PAYLOAD GIMBOL HALF-HEIGHT (IN.)
C        C      PAYLOAD HEIGHT  (IN.)
C        D      PAYLOAD GIMBOL WIDTH (IN.)
C        RHOP   PAYLOAD DENSITY (POUNDS/CUBIC FOOT)
C        VF     DESIRED VELOCITY CAPABILITY (LIMIT) IN FEET/SECOND
C        FMAXLF DESIRED MAXIMUM LOAD FACTOR (LIMIT) IN G*S
C        RDIA   ROCK DIAMETER (INCHES)
C        RS1    ASSUMED BAG RADIUS (FOR CURVE FIT) INCHES
C        RS2    ASSUMED BAG RADIUS (FOR CURVE FIT) INCHES
C        RS3    ASSUMED BAG RADIUS (FOR CURVE FIT) INCHES
C        FABWT  BAG MATERIAL WEIGHT (FOR ONE PLY) LBS/SQ.FOOT
C        FABST  BAG MATERIAL STRENGTH (FOR ONE PLY) POUNDS/INCH
C        PLY1   DESIRED MINIMUM NUMBER OF PLYS INNER 120 DEGREES
C                    (PAYLOAD ATTACH)
C        PLY2   DESIRED MINIMUM NUMBER OF PLYS UPPER AND LOWER CENTER
C               60 DEGREES
C        PLY3   DESIRED MINIMUM NUMBER OF PLYS OUTER 120 DEGREES
C        BFS    BAG FACTOR OF SAFETY (ULTIMATE/LIMIT)
C        SCFWT  WEIGHT OF SCUFF PROTECTION (LBS/SQ.FOOT)
C        PA     AMBIENT PRESSURE (PSI)
C        BET    ANGLE BETWEEN CRUSH PLANE AND X AXIS (RADIANS)
C        CK     POLYTROPIC GAS CONSTANT
C     OUTPUT DATA
C        RS     REQUIRED BAG RADIUS (INCHES)
C        RL     REQUIRED RADIUS FROM TORUS CENTER TO BAG CENTER (IN.)
C        PI     REQUIRED INFLATION PRESSURE ABSOLUTE (PSI)
C        PF     LIMIT BAG PRESSURE AT LIMIT STROKE (POUNDS/SQ.IN.)
C               ABSOLUTE
C      FLOADF MAXIMUM LIMIT LOAD FACTOR AT LIMIT STROKE (G*S)
C        WP     PAYLOAD WEIGHT (POINDS)
C        WL     WEIGHT OF LANDING SYSTEM INCLUDING GAS,EXCLUDING BOTTLE
C               (POUNDS)
C        WIS    WEIGHT OF INFLATION SYSTEM,BOTTLE AND GAS (POUNDS)
C        PLY1F REQUIRED NO. PLYS INNER 120 DEGREES (PAYLOAD ATTACH)
C        PLY2F REQUIRED NO. PLYS UPPER AND LOWER CENTER 60 DEGREES
C        PLY3F REQUIRED NO. PLYS OUTER 120 DEGREES
C      STRES1 MAX. ULT. RUNNING LOAD INNER 120 DEGREES (POUNDS/INCH)
C      STRES2 MAX.RUNNING LOAD UPPER AND LOWER CENTER 60 DEGREES
C               (LB/IN.)
C      STRES3 MAX. RUNNING LOAD OUTER 120 DEGREES (LB./IN.)
C        WL1    LANDING SYSTEM WEIGHT BAG RS1 (POUNDS)
```

```
C        WL2    LANDING SYSTEM WEIGHT BAG RS2 (POUNDS)                       DIR 510
C        WL3    LANDING SYSTEM WEIGHT BAG RS3 (POUNDS)                       DIR 520
C         WIS1  INFLATION SYSTEM WEIGHT (POUNDS) BAG RS1                     DIR 530
C         WIS2  INFLATION SYSTEM WEIGHT (POUNDS) BAG RS2                     DIR 540
C         WIS3  INFLATION SYSTEM WEIGHT (POUNDS) BAG RS3                     DIR 550
C        PI1    INFLATION PRESSURE BAG RS1 (PSI)                             DIR 560
C        PI2    INFLATION PRESSURE BAG RS2 (PSI)                             DIR 570
C        PI3    INFLATION PRESSURE BAG RS3 (PSI)                             DIR 580
C         PF1   PRESSURE AT MAX.LIMIT STROKE BAG RS1 (PSI) ABSOLUTE          DIR 590
C        PF2    PRESSURE AT MAX. STROKE BAG RS2 (PSI)                        DIR 600
C        PF3    PRESSURE AT MAX. STROKE BAG RS3 (PSI)                        DIR 610
C        FLOAD1 MAX.LIMIT LOAD FACTOR BAG RS1 (G*S)                          DIR 620
C        FLOAD2 MAXIMUM LOAD FACTOR BAG RS2 (G*S)                            DIR 630
C        FLOAD3 MAXIMUM LOAD FACTOR BAG RS3 (G*S)                            DIR 640
C         S     STROKE (INCHES)                                              DIR 650
C         FORCE TOTAL LOAD NORMAL TO SURFACE (POUNDS) LIMIT                  DIR 660
C         DIST  Y DISTANCE TO CENTROID OF FOOTPRINT AREA (INCHES)            DIR 670
C         P     LIMIT BAG PRESSURE AT STROKE S (PSI) ABSOLUTE                DIR 680
C         AF    FOOTPRINT AREA (SQUARE INCHES)                               DIR 690
C         CLEAR PAYLOAD/GROUND CLEARANCE FOR IMPACT AT ANGLE BET (INCHES)DIR 700
C         ALCL  ALLOWABLE CLEARANCE BETWEEN PAYLOAD AND GROUND (INCHES) ADIR 710
C               11.8( OF AVAILABLE STROKE FOR ULT. VELOCITY CAPABILITY   DIR 720
C         U     TOTAL ENERGY ABSORBED FOR LANDING AT ANGLE BET              DIR 730
C               (INCH POUNDS)                                                DIR 731
C         VELBT LIMIT VELOCITY CABABILITY FOR LANDING AT ANGLE BET           DIR 740
C               (FEET/SECOND)                                                DIR 741
C        CLEARE PAYLOAD/GROUND CEARANCE FOR END LANDING (BET=90DEGREES) IDIR 750
C         UE    TOTAL ENERGY ABSORBED FOR END LANDING (INCH POUNDS)          DIR 760
C         VELE  LIMIT VELOCITY CAPABILITY FOR END LANDING (FEET/SECOND)      DIR 770
C         GAMA  BAG ANGLE WITH LOCAL HORIZONTAL AT PAYLOAD ATTACH-END        DIR 780
C               LANDING (RADIANS)                                            DIR 781
C         RD1   OUTER RADIUS AT MAX STROKE POINT (INCHES)                    DIR 790
C         RD2   RADIUS AT PAYLOAD ATTACH - END LANDING (INCHES) MAXIMUM      DIR 800
C               STROKE POINT                                                 DIR 801
C         ALCE  ALLOWABLE CLEARANCE BETWEEN PAYLOAD AND GROUND END           DIR 810
C               LANDING ALLOWING 11.8( OF AVAILABLE STROKE FOR ULTIMATE      DIR 820
C               VELOCITY CAPABILITY (INCHES)                                 DIR 821
C         GSTR1 MAX. RUNNING LOAD INNER 120 DEGREES ,POUNDS/INCH)            DIR 830
C               ULTIMATE                                                     DIR 831
C        EGSTR2 MAX. RUNNING LOAD UPPER AND LOWER CENTER 60 DEGREES          DIR 840
C               (LB./IN.) ULTIMATE                                           DIR 841
C        EGSTR3 MAX. RUNNING LOAD OUTER 120 DEGREES (POUNDS/INCH)            DIR 850
C               ULTIMATE                                                     DIR 851
C                                                                            DIR 860
      WRITE (6,340)                                                          DIR 870
      WRITE (6,350) A,B,C,D,RHOP,VF,FMAXLF,RDIA                              DIR 880
      WRITE (6,360) RS1,RS2,RS3,FABWT,FABST,PLY1,PLY2,PLY3                   DIR 890
      WRITE (6,370) BFS,SCFWT,BET,PA,CK                                      DIR 900
      DO 20 I=1,600                                                          DIR 910
20    ASAVE(I)=0.0                                                           DIR 920
      DO 30 J=1,600                                                          DIR 930
30    FSAVE(J)=0.0                                                           DIR 940
      BET1=BET                                                               DIR 950
      RS=RS1                                                                 DIR 960
      WP=(0.785*C*A**2+6.28*D*B*(A+D))*RHOP/1728.                            DIR 970
      PM=WP/386.0                                                            DIR 980
```

```
      KOUNT=1
      GO TO 60
40    RS=RS2
      KOUNT=2
      GO TO 60
50    RS=RS3
      KOUNT=3
60    SMAX=RS-C/2.0-RDIA
      SMAX=SMAX/1.118
      DS=SMAX/20.0+0.0005
      MOUNT=0
      S=0.0
      I=0
      J=0
      IND=0
      FDPP=0.0
      FPP=0.0
      AMENG=0.0
      ENRGY=0.0
      PP=A/2.0+D
      RL=RP+RS
C     INCREMENT STROKE CALCULATE ENERGY (PI UNKNOWN)
70    S=S+DS
      XRS=0.0053+0.6901*(S/RS)-0.525*(S/RS)**2
      YRS=0.0049+2.409*(S/RS)-1.171*(S/RS)**2
      VOLR=0.9983+0.0485*(S/RS)+0.2399*(S/RS)**2
      FPA=3.14*RS**2*(XRS**2-YRS**2+2.0*RL/RS*(XRS+YRS))
      FDPI=FPA*VOLR**CK
      ENRG=(FDPI+FDPP)*DS/2.0
      ENRGY=ENRGY+ENRG
      AMEN=PA*(FPA+FPP)*DS
      AMENG=AMENG+AMEN
      FPP=FPA
      FDPP=FDPI
      IF (S.LT.SMAX) GO TO 70
C
C     CALCULATE MINIMUM BAG WEIGHT
      A1=13.12*RS*(RP+0.172*RS)
      A2=13.12*RS*(RP+RS)
      A3=13.12*RS*(RP+1.828*RS)
      WLM1=FABWT*PLY1*A1/144.0
      WLM2=FABWT*PLY2*A2/144.0
      WLM3=FABWT*PLY3*A3/144.0
      WSKF=SCFWT*(A1+A2+A3)/144.0
      WL=WLM1+WLM2+WLM3+WSKF
C
C     GO TO EQUIVALENT TWO MASS SYSTEM AND FIND PI REQUIRED
      EFFK=2.0*ENRGY/S**2
80    TM=WL/386.0
      OMEGN=SQRT(EFFK/(PM+TM))
      XLAM=0.6*WL/(WP+0.6*WL)
      FFS=0.9993-0.3348*XLAM+0.0995*XLAM**2
      PI=144.0*FFS**2*VF**2/(S**2*OMEGN**2)
      PI=PI+AMENG/(S**2*OMEGN**2*(PM+TM))
      FFN=1.1217-0.2704*XLAM+4.2594*XLAM**2-4.6105*XLAM**3
      FLOADF=FFN*OMEGN*VF*SQRT(PI)/32.2
```

```
      FLOADF=FLOADF*BFS
      PF=BFS*VOLR**CK*(PI-PA)
C     CALCULATE BAG STRESSES, THICKNESSES, AND UPDATE BAG WEIGHT
      RATI=RS*(1.0-YRS)/(1.57*RS-YRS*RS-B)
      THE=0.0
      DTHE=0.04
90    THE=THE+DTHE
      IF (THE.GE.3.1) GO TO 100
      RATO=SIN(THE)/THE
      IF (RATI.LT.RATO) GO TO 90
      RR1=RS*(1.0-YRS)/SIN(THE)
      GO TO 110
100   RR1=0.0
110   PHI=0.422*S/RS
      STRES1=(FLOADF*WP/(6.28*RP)+PF*RR1*COS(THE-1.57))/COS(PHI)
      PLY1F=STRES1/FABST
      IPLY1=PLY1F
      IF ((IPLY1+.001).GT.PLY1F) GO TO 120
      PLY1F=IPLY1+1.
120   IF (PLY1F.GT.PLY1) GO TO 130
      PLY1F=PLY1
130   WLM1=FABWT*PLY1F*A1/144.
      STRES2=(6.28*RP*STRES1*COS(PHI)+0.167*WL*FLOADF-3.14*PF*(RP*RS+0.2
     15*RS**2))/(3.14*(RP+RS/2.0))
      STREH2=PF*RS/2.0
      STRES2=AMAX1(STREH2,STRES2)
      PLY2F=STRES2/FABST
      IPLY2=PLY2F
      IF ((IPLY2+.001).GT.PLY2F) GO TO 140
      PLY2F=IPLY2+1.
140   IF (PLY2F.GT.PLY2) GO TO 150
      PLY2F=PLY2
150   WLM2=FABWT*PLY2F*A2/144.
      STRES3=(6.28*RP*STRES1*COS(PHI)+0.333*WL*FLOADF-3.14*PF*(RP*RS*3.0
     1+2.25*RS**2))/(3.14*(RP+1.5*RS))
      STREH3=PF*RS/2.0
      STRES3=AMAX1(STREH3,STRES3)
      PLY3F=STRES3/FABST
      IPLY3=PLY3F
      IF ((IPLY3+.001).GT.PLY3F) GO TO 160
      PLY3F=IPLY3+1.
160   IF (PLY3F.GT.PLY3) GO TO 170
      PLY3F=PLY3
170   WLM3=FABWT*PLY3F*A3/144.
      WIS=PI*RS**2*RL/13470.
      WL=WLM1+WLM2+WLM3+WSKF+0.1165*WIS
      PF=PI*VOLR
      FLOADF=FLOADF/BFS
      PO=PI-PA
      MOUNT=MOUNT+1
      IF (MOUNT.LT.3) GO TO 80
      IF (KOUNT.GT.1) GO TO 180
      FLOAD1=FLOADF
      WL1=WL
      WIS1=WIS
      PI1=PI
```

```
      PF1=PF
      GO TO 40
180   IF (KOUNT.GT.2) GO TO 190
      FLOAD2=FLOADF
      WL2=WL
      WIS2=WIS
      PI2=PI
      PF2=PF
      GO TO 50
190   IF (KOUNT.GT.3) GO TO 230
      FLOAD3=FLOADF
      WL3=WL
      WIS3=WIS
      PI3=PI
      PF3=PF
      KOUNT=4
      DAKE=FLOAD1*(RS2-RS3)-FLOAD2*(RS1-RS3)+FLOAD3*(RS1-RS2)
      IF (DAKE.EQ.0.0) GO TO 220
      AKE=(FLOAD1*(RS2**2-RS3**2)-FLOAD2*(RS1**2-RS3**2)+FLOAD3*(RS1**2-
     1RS2**2))/(2.0*DAKE)
      AME=((RS1-AKE)**2-(RS2-AKE)**2)/(FLOAD1-FLOAD2)
      IF (AME.EQ.0.0) GO TO 220
      AHE=FLOAD1-(RS1-AKE)**2/AME
      AAME=AME*(FMAXLF-AHE)
      IF (AAME.LT.0.0) GO TO 220
      IF (AME) 200,220,210
200   RS=AKE+SQRT(AAME)
      GO TO 60
210   RS=AKE-SQRT(AAME)
      GO TO 60
220   RS=RS1+(RS1-RS3)*(FMAXLF-FLOAD1)/(FLOAD1-FLOAD3)
      GO TO 60
230   IF (FLOADF.LE.FMAXLF) GO TO 240
      RS=RS+0.25
      GO TO 60
240   CONTINUE
C     DETERMINE VELOCITY CAPABILITY FOR END LANDING (BET=90DEGREES)
      BET=1.57
      FORCP=0.0
      UE=0.0
      S=0.0
      IND=0
      ALCE=0.212*RS+0.694*RDIA
      CLEARE=2.*RS-B*TAN(ALPHA)-S
250   DS=(S+CLEARE-ALCE+0.001)/40.
      S=S+DS
      CLEARE=2.*RS-B*TAN(ALPHA)-S
      CALL LOAD (RS,RL,B,PO,PA,BET,S,IND,FORCE,DIST,P,AF,CK)
      J=J+1
      FSAVE(J)=S
      GSAVE(J)=FORCE
      HSAVE(J)=DIST
      PSAVE(J)=P+PA
      RSAVE(J)=AF
      DUE=DS*(FORCE+FORCP)/2.
      UE=UE+DUE
```

```
      FORCP=FORCE                                                       D1R2670
      EPS=1.-AF*(SCFWT+PLY3F*FABWT)/(144.*WL)                           D1R2680
      VELE=SQRT(5.36*UE/(EPS*WL+WP))                                    D1R2690
      QSAVE(J)=VELE                                                     D1R2700
      IF (ALCE.LT.CLEARE) GO TO 250                                     D1R2710
C     DETERMINE BAG STRESSES FOR END LANDING AND UPDATE BAG PLY IF      D1R2720
C     REQUIRED                                                          D1R2721
      DGAMA=0.01                                                        D1R2730
      GAMA=0.0                                                          D1R2740
      GF1=ALCE-1.362*RS+0.637*B                                         D1R2750
      GF2=0.637*(RS-B)                                                  D1R2760
      GF3=B-RS                                                          D1R2770
260   GAMA=GAMA+DGAMA                                                   D1R2780
      GF4=GF1*SIN(GAMA)+GF2*GAMA                                        D1R2790
      GF5=GF3*(1.-COS(GAMA))                                            D1R2800
      IF (GF4.LT.GF5) GO TO 260                                         D1R2810
      RD1=ALCE/2.-GF5/(2.*SIN(GAMA))                                    D1R2820
      RD2=(RS-B)/SIN(GAMA)                                              D1R2830
      EGSTR1=P*RD2*(RP+ALCE/2.-RD1)/RP*BFS                              D1R2840
      ZETA=ASIN(B/RS)                                                   D1R2850
      EGSTH1=P*RS*(RP+RS*COS(ZETA)/2.)/RP*BFS                           D1R2860
      EGSTR1=AMAX1(EGSTR1,EGSTH1)                                       D1R2870
      EGSTR2=P*RD1*(RP+ALCE-1.5*RD1)/(RP+ALCE-2.*RD1)*BFS               D1R2880
      EGSTH2=P*RS*(RP+0.75*RS)/(RP+0.5*RS)*BFS                          D1R2890
      EGSTR2=AMAX1(EGSTR2,EGSTH2)                                       D1R2900
      EGSTR3=P*RD1*(1.-RD1/(2.*(RP+ALCE)))*BFS                          D1R2910
      EGSTH3=P*RS*(RP+1.25*RS)/(RP+1.5*RS)*BFS                          D1R2920
      EGSTR3=AMAX1(EGSTR3,EGSTH3)                                       D1R2930
      IF (EGSTR3.LE.(FABST*PLY3F)) GO TO 280                            D1R2940
      PLY3=EGSTR3/FABST                                                 D1R2950
      IPLY3=PLY3                                                        D1R2960
      IF ((IPLY3+.001).GT.PLY3) GO TO 270                               D1R2970
      PLY3=IPLY3+1.                                                     D1R2980
270   PLY2=PLY2F                                                        D1R2990
      PLY1=PLY1F                                                        D1R3000
      IF (KOUNT.EQ.5) GO TO 280                                         D1R3010
      KOUNT=5                                                           D1R3020
      GO TO 60                                                          D1R3030
C     DETERMINE VELOCITY CAPABILITY FOR LANDING AT INPUT ANGLE BET      D1R3040
280   FORCP=0.0                                                         D1R3050
      BET=BET1                                                          D1R3060
      IND=0                                                             D1R3070
      U=0.0                                                             D1R3080
      ALPHA=0.50*ASIN(B/(4.*RS))                                        D1R3090
      ALCL=RDIA                                                         D1R3100
      S=0.0                                                             D1R3110
      CLEAR=RS*(1.+SIN(BET))-B*(COS(BET)+SIN(BET)*TAN(ALPHA))-S         D1R3120
      CLEAR2=RS*(1.+SIN(BET))+D*SIN(BET)-C*COS(BET)/2.-S                D1R3130
      CLEAR=AMIN1(CLEAR,CLEAR2)                                         D1R3140
      ALCL=0.894*ALCL+0.106*CLEAR                                       D1R3150
290   DS=(S+CLEAR-ALCL+0.001)/40.                                       D1R3160
      S=S+DS                                                            D1R3170
      CLEAR=RS*(1.+SIN(BET))-B*(COS(BET)+SIN(BET)*TAN(ALPHA))-S         D1R3180
      CLEAR2=RS*(1.+SIN(BET))+D*SIN(BET)-C*COS(BET)/2.-S                D1R3190
      CLEAR=AMIN1(CLEAR,CLEAR2)                                         D1R3200
      CALL LOAD (RS,RL,B,PO,PA,BET,S,IND,FORCE,DIST,P,AF,CK)            D1R3210
```

```
      I=I+1
      ASAVE(I)=S
      BSAVE(I)=FORCE
      CSAVE(I)=DIST
      DSAVE(I)=P+PA
      ESAVE(I)=AF
      DU=DS*(FORCE+FORCP)/2.
      U=U+DU
      FORCP=FORCE
      EPS=1.-AF*(SCFWT+PLY3F*FABWT)/(144.*WL)
      VELBT=SQRT(5.36*U/(EPS*WL+WP))
      SSAVE(I)=VELBT
      IF (ALCL.LT.CLEAR) GO TO 290
      WRITE (6,380)
      WRITE (6,390) RS,RL,PI,PF,FLOADF,WP,WL,WIS
      WRITE (6,400) PLY1F,PLY2F,PLY3F,STRES1,STRES2,STRES3
      WRITE (6,410) WL1,PI1,PF1,FLOAD1,WIS1
      WRITE (6,420) WL2,PI2,PF2,FLOAD2,WIS2
      WRITE (6,430) WL3,PI3,PF3,FLOAD3,WIS3
      WRITE (6,470)
      WRITE (6,480) CLEARL,UE,VELE,GAMA,RD1,RD2,ALCE
      WRITE (6,490) EGSTR1,EGSTR2,EGSTR3
      DO 300 M=1,J
300   WRITE (6,500) FSAVE(M),GSAVE(M),HSAVE(M),PSAVE(M),RSAVE(M),QSAVE(M
     1)
      WRITE (6,440)
      WRITE (6,450) CLEAR,ALCL,U,VELBT
      N=I
      DO 310 L=1,N
310   WRITE (6,460) ASAVE(L),BSAVE(L),CSAVE(L),DSAVE(L),ESAVE(L),SSAVE(L
     1)
      GO TO 10
C
320   FORMAT (1H1,4X,42HINFLATABLE TORUS STRUCTURAL DESIGN PROGRAM////)
330   FORMAT (9F8.1)
340   FORMAT (1X,16HINPUT PARAMETERS//)
350   FORMAT (3X,2HA=F9.2,3X,2HB=F9.2,3X,2HC=F9.2,3X,2HD=F9.2,3X,5HRHOP=
     1F6.1,3X,3HVF=F9.1,3X,7HFMAXLF=F12.1,3X,5HRDIA=F9.2/)
360   FORMAT (3X,4HRS1=F9.2,3X,4HRS2=F9.2,3X,4HRS3=F9.2,3X,6HFABWT=F9.4,
     13X,6HFABST=F9.1,3X,5HPLY1=F4.1,3X,5HPLY2=F4.1,3X5HPLY3=F4.1/)
370   FORMAT (3X,4HBFS=F6.3,3X,6HSCFWT=F9.4,3X,4HBET=F6.3,3X,3HPA=F9.3,3
     1X,3HCK=F6.4/)
380   FORMAT (1X,11HOUTPUT DATA//)
390   FORMAT (3X,3HRS=F9.2,3X,3HRL=F9.2,3X,3HPI=F9.2,3X,3HPF=F9.2,3X,7HF
     1LOADF=F12.1,3X,3HWP=F9.1,3X,3HWL=F9.1,3X,4HWIS=F9.1/)
400   FORMAT (3X,6HPLY1F=F4.1,3X,6HPLY2F=F4.1,3X,6HPLY3F=F4.1,3X,7HSTRES
     11=F9.1,3X,7HSTRES2=F9.1,3X,7HSTRES3=F9.1/)
410   FORMAT (3X,4HWL1=F9.1,3X,4HPI1=F9.2,3X,4HPF1=F9.2,3X,7HFLOAD1=F12.
     11,3X,5HWIS1=F9.1/)
420   FORMAT (3X,4HWL2=F9.1,3X,4HPI2=F9.2,3X,4HPF2=F9.2,3X,7HFLOAD2=F12.
     11,3X,5HWIS2=F9.1/)
430   FORMAT (3X,4HWL3=F9.1,3X,4HPI3=F9.2,3X,4HPF2=F9.2,3X,7HFLOAD3=F12.
     11,3X,5HWIS3=F9.1/)
440   FORMAT (25X,*LOADSTROKEFORIMPACTATCONSTANTANGLEBET*/)
450   FORMAT (3X,6HCLEAR=F9.3,3X,5HALCL=F9.3,3X,2HU=F12.1,3X,6HVELBT=F9.
     12/)
```

```
460   FORMAT (3X,2HS=F9.3,5X,6HFORCE=F10.1,5X,5HDIST=F9.3,5X,2HP=F9.3,5X
     1,3HAF=F10.1,5X,6HVELBT=F6.1/)
470   FORMAT (25X,46HLOAD STROKE FOR END LANDING (BET = 90 DEGREES)/)
480   FORMAT (3X,7HCLEARE=F9.3,3X,3HUE=F12.1,3X,5HVELE=F9.2,3X,5HGAMA=F9
     1.4,3X,4HRD1=F9.2,3X,4HRD2=F9.2,3X,5HALCE=F9.3/)
490   FORMAT (3X,7HEGSTR1=F9.1,3X,7HEGSTR2=F9.1,3X,7HEGSTR3=F9.1/)
500   FORMAT (3X,2HS=F9.3,5X,6HFORCE=F10.1,5X,5HDIST=F9.3,5X,2HP=F9.3,5X
     1,3HAF=F10.1,5X,5HVELE=F6.1/)
      END
```

```
      SUBROUTINE LOAD (RS,RL,B,PO,PA,BET,S,IND,FORCE,DIST,P,AF,CK)
C     THIS SUBROUTINE DETERMINES LOAD,PRESSURE,FOOTPRINT AREA VS STROKE
C     FOR GENERAL LANDING ATTITUDE
      TOL=0.001
      DTHE=.030
      STR=S
      PI=3.14159
      VO=2*PI*PI*RL*RS*RS
      SRATIO=STR/RS
      IF (IND.NE.0) GO TO 10
      IND=1
      SSAVE=0.0
      AFS=0.0
      V=VO
10    THE=0.0
      ACHORD=0.0
      ACMOM=0.0
20    THE=THE+DTHE
      PPSI=TAN(BET)*COS(THE)
      PSI=ATAN(PPSI)
      IF (PSI.LT.1.56) GO TO 40
      IF (PSI.GT.1.58) GO TO 40
      H=RL+RS
      STTH=H-(H-STR)/COS(THE)
      IF (STTH.LE.0.0) GO TO 80
      IF (STTH.GT.RS) GO TO 60
      CH=2*SQRT(2*STTH*RS-STTH**2)
30    ACH=CH*DTHE*(H-STTH)*2
      ACH=ACH/COS(THE)
      ACHORD=ACHORD+ACH
      XDLOD=H-STR
      ACMOM=ACHORD*(H-STR)
      YDLOD=0.0
      ARMOM=0.0
      GO TO 20
40    STTH=COS(PSI)*STR/COS(BET)+RS*COS(PSI)*(SQRT(1.0+TAN(PSI)**2)-SQRT
     1(1.0+TAN(BET)**2))+RL*COS(PSI)*TAN(BET)*(COS(THE)-1.0)
      IF (STTH.LE.0.0) GO TO 80
      IF ((THE-3.14).GT.0.0) GO TO 80
      IF (STTH.GT.RS) GO TO 70
      CH=2*SQRT(2*STTH*RS-STTH**2)
50    RTP=RL+RS*TAN(PSI)/SQRT(1.0+TAN(PSI)**2)
      ACH=CH*DTHE*(RTP-STTH*SIN(PSI))*2.0
      ACH=ACH*COS(PSI)/COS(BET)
      ACHORD=ACHORD+ACH
      ACOM=ACH*(RTP-STTH*SIN(PSI))*COS(THE)
      ACMOM=ACMOM+ACOM
      XDLOD=ACMOM/ACHORD
      YDLOD=RS*SQRT(1+TAN(BET)**2)-STR/COS(BET)+(RL-XDLOD)*TAN(BET)
      ARMOM=ACHORD*YDLOD
      GO TO 20
60    CH=2.*RS
      GO TO 30
70    CH=2.*RS
      GO TO 50
80    CONTINUE
```

```
      IF (SRATIO.GE.1.537) GO TO 90
      AF=ACHORD*(0.499+0.326*SRATIO)
      GO TO 100
90    AF=ACHORD
100   V=V-0.5*(STR-SSAVE)*(AF+AFS)
      DIST=YDLOD
      P=(PO+PA)*(VO/V)**CK-PA
      FORCE=P*AF
      AFS=AF
      SSAVE=STR
      RETURN
      END

  45.     2.      9.      3.     53.4    65.     50.     5.
  100.   120.    140.    0.017   100.    1.      1.      1.
  2.5     .132    .100    .072    1.40
```

APPENDIX D

OPERATING INSTRUCTIONS FOR THE INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

APPENDIX D

TABLE OF CONTENTS

APPENDIX D

D.1 INTRODUCTION

The Inflatable Torus Landing Loads and Motions Program determines the spatial positions, velocities, and accelerations of a given inflatable torus lander as a function of time. These parameters are determined using the normal force due to compressing the torus inflation gas and the friction forces and moments. The lander configuration may be established with the Inflatable Torus Structural Design Program, Appendix C, or a lander design may be available from some other source.

Features incorporated in this program include the ability to: select up to six degrees of freedom thereby allowing simulation of spatial motion; vary load/stroke hysteresis effects on rebound, lander geometry, surface slope, coefficient of friction, rock diameter; select values for as many as eight independent parameters used to stop machine computation; and ability to select variable or constant step Predictor-Corrector or Runge-Kutta integration methods. Input data to this program includes initial lander attitude and position; linear and rotational velocities; lander geometric and inertia properties; and surface conditions such as ground slope, coefficient of friction, and rock diameter. Output data consists of the lander's translational and angular positions, velocities, and accelerations as a function of time.

The Inflatable Torus Landing Loads and Motions Program was developed for a landing vehicle as shown in Figure D-1. The lander is comprised of two main parts; the inflated landing system and the payload. The payload consists of a cylindrical shaped package mounted in the center of the

APPENDIX D

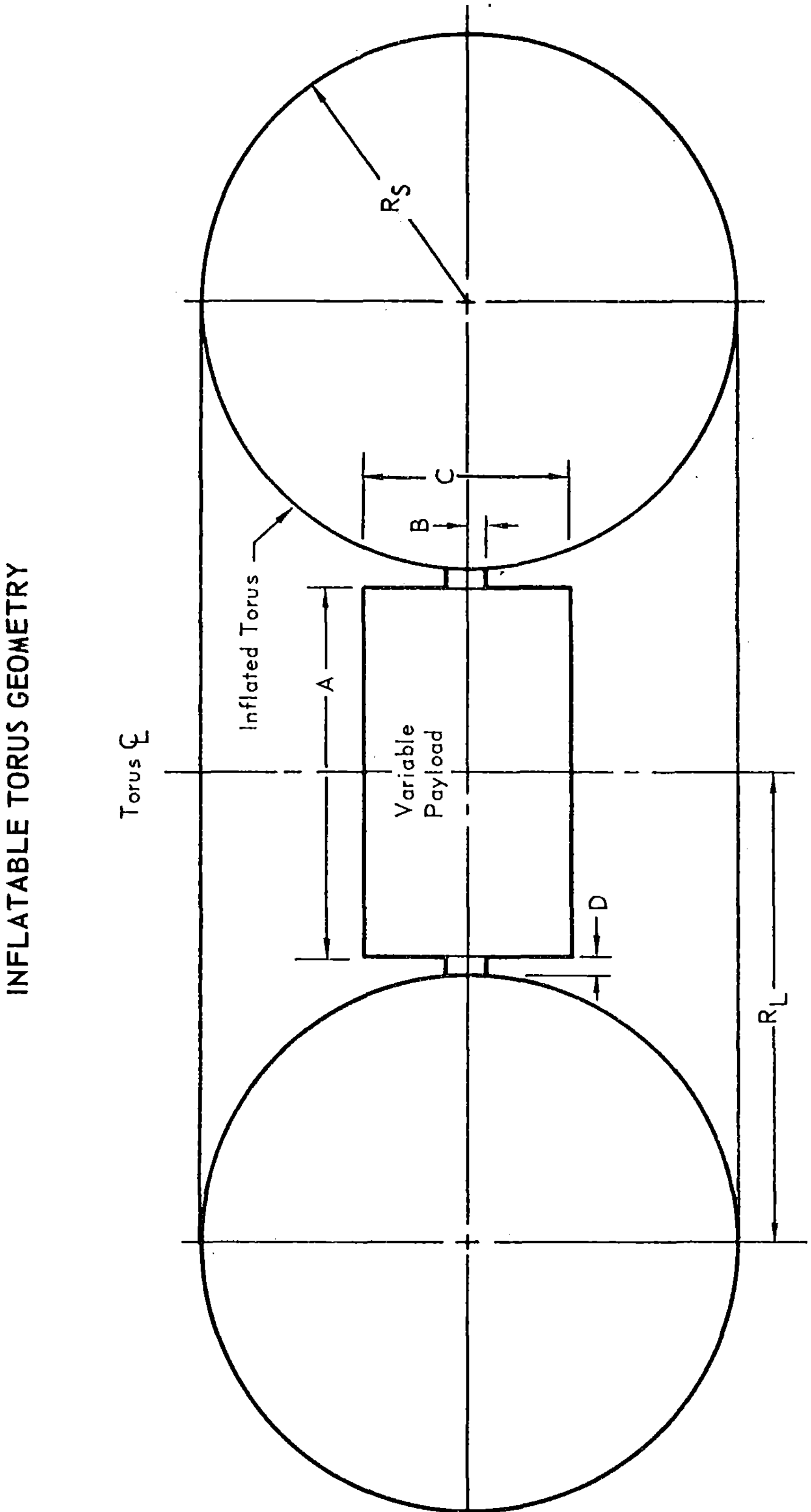


Figure D-1

inflated torus impact bag. A gimbal ring supports the payload structure and allows the required payload alignment to occur following landing. During landing, this gimbal ring is locked so that it provides rigid support for the payload. The payload package and gimbal ring are assumed to have a uniform weight density.

The inflatable torus is constructed of a suitable fabric coated with an elastomer to provide gas containment and scuff resistance. The torus is assumed to be an unvented, uncompartmented structure with material hysteresis effects included in the analysis. To provide the required torus strength with minimum weight, the thickness of the torus material may be changed in three steps as shown in Figure D-2. The symbols PLY1, PLY2, and PLY3 are used to designate the number of material plys in the respective torus sections and FABWT is the weight per square foot of one ply of material. These terms are consistent with the Inflatable Torus Structural Design Program.

VARIATION OF TORUS MATERIAL THICKNESS

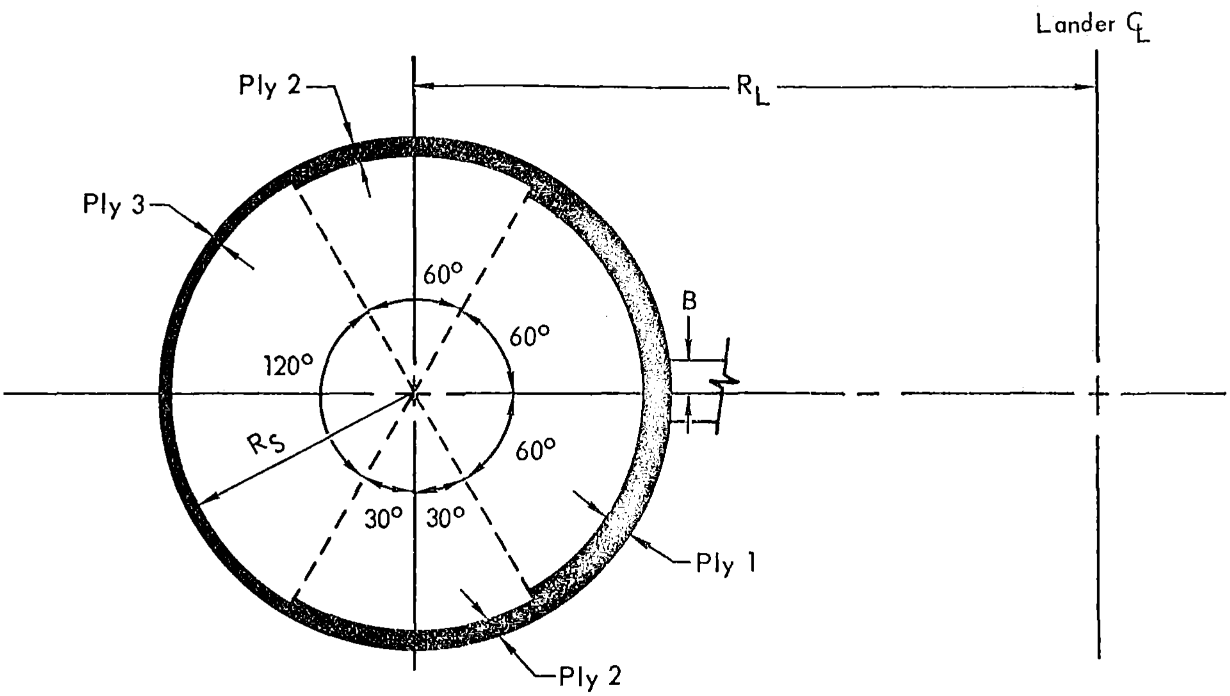


Figure D-2

APPENDIX D

D.2 ANALYTICAL PROCEDURES

D.2.1 COORDINATE SYSTEMS - Three coordinate systems used to define the motion of the lander as a function of time are shown in Figure D-3. All three systems are right-handed and each consists of three orthogonal axes. These coordinate systems are defined as follows:

- A coordinate system moving with the lander and fixed at its center of gravity (X, Y, Z). This system is referred to as the lander coordinate system. The X axis is the axis of symmetry of the landing vehicle and the Y and Z directions are chosen arbitrarily. Roll, pitch, and yaw axes coincide with the reference X, Y, and Z axes respectively. In defining the signs of rotation, the right-hand rule is used.
- A coordinate system fixed in the planet and aligned with the gravity vector (X_f, Y_f, Z_f). This system is referred to as the gravity coordinate system. The Z_f axis is directed toward the center of the planet and the positive X_f and Y_f axes are directed toward the north and east respectively.
- A coordinate system fixed in the planet and oriented with respect to the slope of the local surface (X_{1s}, Y_{1s}, Z_{1s}). This system is referred to as the surface coordinate system and differs from the gravity coordinate system by the rotation α about the X_f axis.

These coordinate systems are related by the following expressions where TR and TRL are the matrices of direction cosines:

$$\begin{Bmatrix} X_f \\ Y_f \\ Z_f \end{Bmatrix} = \begin{bmatrix} \\ TR\ (I,\ J) \\ \\ \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

and

$$\begin{Bmatrix} X_{1s} \\ Y_{1s} \\ Z_{1s} \end{Bmatrix} = \begin{bmatrix} \\ TRL\ (I,J) \\ \\ \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

These transformations are used for relating forces, accelerations, velocities, and displacements between the various coordinate systems.

During the integration of the equations of motion, one or more of the direction cosines may become slightly greater than one. This is a result of the lander experiencing high angular velocities and is due to the finite step nature of the numerical integration methods employed. When this occurs, the program prints out a message, recomputes the direction cosines such that the direction vector is normalized, and continues the integration.

D.2.2 ASSUMPTIONS - It is assumed in the analysis that the lander may be represented as a rigid mass experiencing forces and moments at its center of gravity. These loads are comprised of the normal force due to compressing the torus inflation gas and the friction forces and associated torques due to the motion of the torus material relative to the landing surface. These loads are resolved to the lander's center of gravity and the resulting equations of motions take the form

$$m \begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix} - m \begin{bmatrix} o & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & o & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & o \end{bmatrix} \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{Bmatrix} + m \begin{Bmatrix} g_X \\ g_Y \\ g_Z \end{Bmatrix}$$

APPENDIX D
COORDINATE SYSTEMS

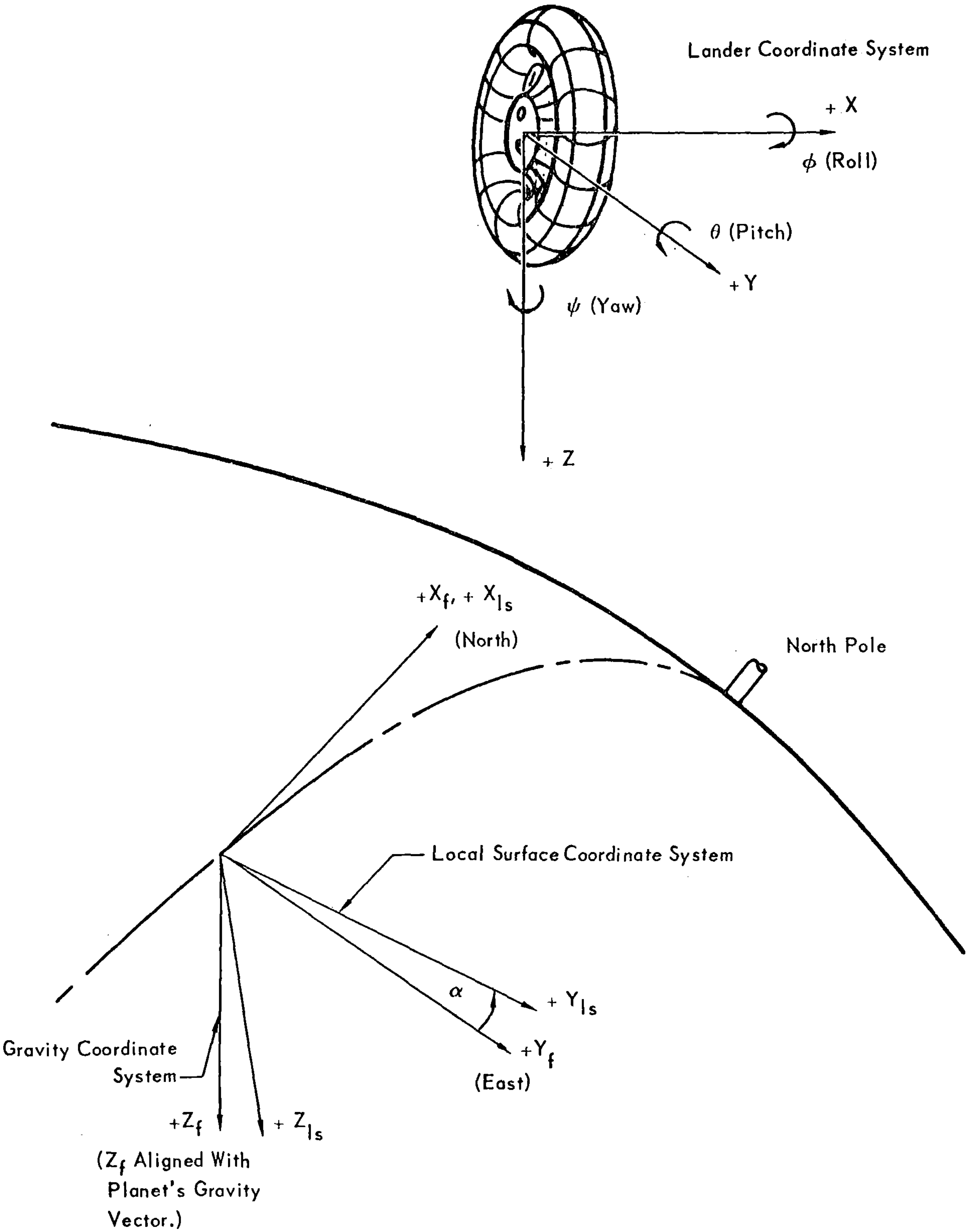


Figure D-3

and

$$\begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{Bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix} = \begin{Bmatrix} T_X \\ T_Y \\ T_Z \end{Bmatrix} - \begin{bmatrix} \dot{I}_{XX} & -\dot{I}_{XY} & -\dot{I}_{XZ} \\ -\dot{I}_{XY} & \dot{I}_{YY} & -\dot{I}_{YZ} \\ -\dot{I}_{XZ} & -\dot{I}_{YZ} & \dot{I}_{ZZ} \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} - \begin{bmatrix} o & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & o & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & o \end{bmatrix} \begin{bmatrix} I_{XX} & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_{YY} & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_{ZZ} \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}$$

Basic assumptions which are reflected in the digital simulation of this study are as follows:

- o Rigid body payload
- o Unyielding landing surface
- o No surface protuberances or depressions other than rocks
- o Aerodynamic forces are negligible
- o Changes in moments of inertia are negligible
- o Normal force unaffected by friction force
- o Normal force calculation based on assumed torus deflection shape
- o Friction force function of normal force
- o Coefficient of friction changes linearly from zero at zero sliding velocity to the constant input value (AMU) over a small (input quantity VMIN) velocity increment
- o Direction of friction force opposite to footprint area centroid velocity
- o Deformations of torus due to friction forces are neglected
- o Ideal gas undergoing polytropic compression process
- o Torus material hysteresis effects are included

D.2.3 METHODS OF ANALYSIS

D.2.3.1 Normal Force. - The analysis required to determine the normal force as a function of the lander's stroke and attitude is presented here. This force is normal to the landing surface and is the result of compressing the torus inflation gas while the lander is in contact with the ground. There are two landing conditions which require slightly different analytical techniques for predicting the normal force; (1) a flat landing where the lander's attitude, β is zero and (2) an oblique landing where β is not equal to zero.

Flat Landing - An assumed torus deflected shape, as shown in Figure D-4, is the basis for predicting the normal force during a flat landing. Assumptions required for determining the flat landing torus deflected shape are:

- o Tangency point (4) directly above (3) .
- o Torus material is tangent to landing surface at points (2) and (3) .
- o Material is tangent to gimbal ring above point (1) .
- o Payload attach point experiences only vertical motion.
- o No stretching of torus material.

A system of six equations are used to describe the deflected shape. From these, the six unknown torus deflection parameters, R_1, R_2, R_3, e, f, and θ_t are determined for a given stroke. These quantities are related as follows:

$$e = \frac{S}{2}\left(\frac{\pi}{\pi+1}\right)$$

$$R_3 = R_S + e$$

TORUS DEFLECTED SHAPE
FLAT LANDING

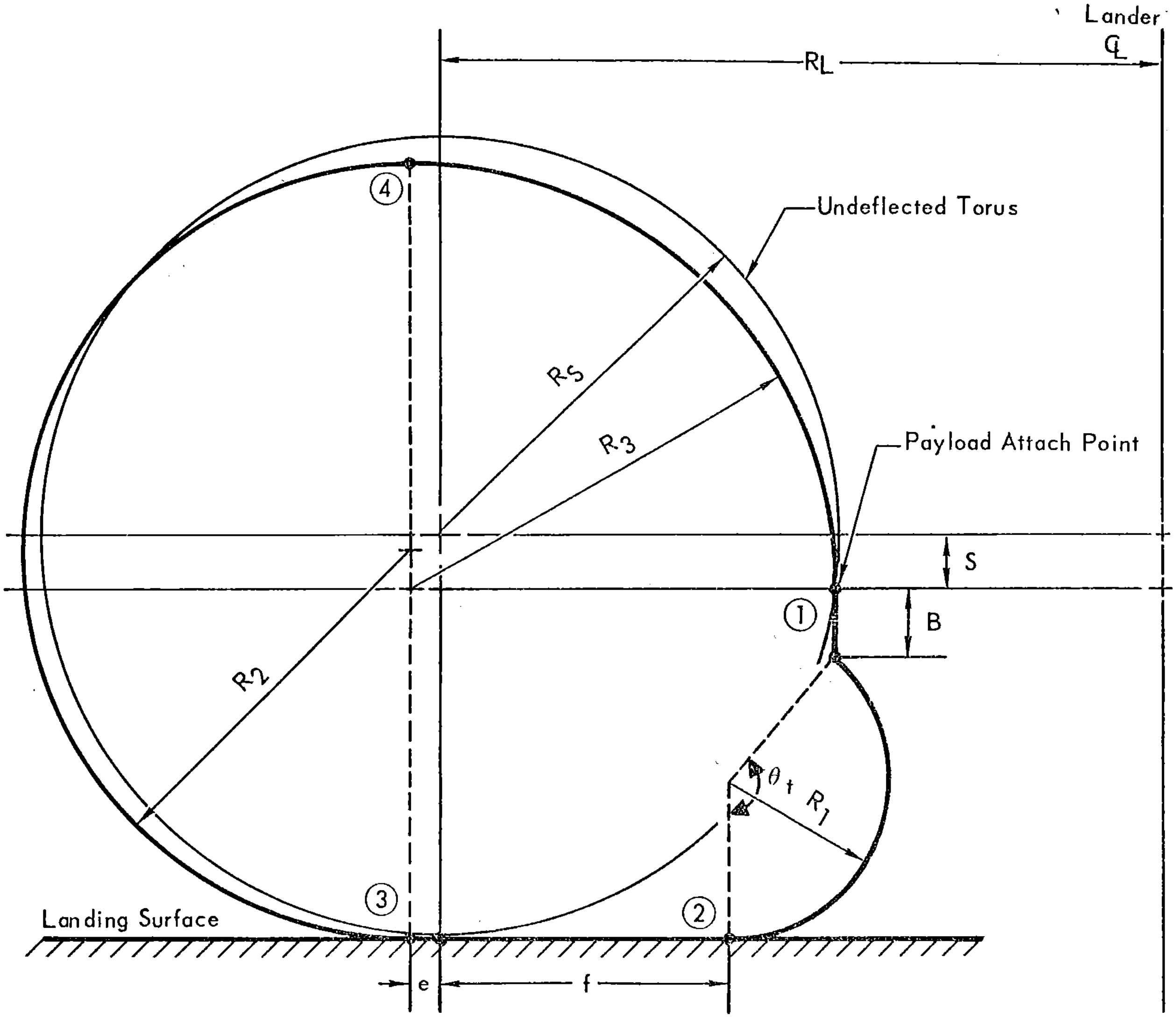


Figure D-4

$$R_2 = R_s - \frac{S}{4}\left(\frac{\pi - 2}{\pi - 1}\right)$$

$$R_1 \theta_t + f = \frac{\pi}{2} R_s - B$$

$$R_1 (1 - \cos \theta_t) = R_s - (S + B)$$

$$f + R_1 \operatorname{Sin} \theta_t = R_s$$

In the above, S is the stroke of the payload center of gravity.

With the deflection shape parameters, the footprint area may be expressed as a function of stroke in the following manner.

$$A_f = \pi (e + f)(2R_L + e - f)$$

The resulting internal torus volume change in terms of the stroke is shown in Figure D-5. The internal bag pressure, assuming an ideal gas undergoing a polytropic process, is expressed as:

$$P = (P_i + P_a)\left(Vo/V\right)^n - P_a$$

where P = torus gage pressure - psig

P_i = initial inflation pressure - psig

P_a = atmospheric pressure - psi

V_o = undeformed torus volume - in^3

V = torus volume - in^3

n = gas constant which determines gas compression process

$n=1$ for isothermal process

$n=k$ for isentropic process - where k is the specific heat ratio for the inflation gas of interest

TORUS VOLUME CHANGE AS A FUNCTION OF STROKE
FLAT LANDING

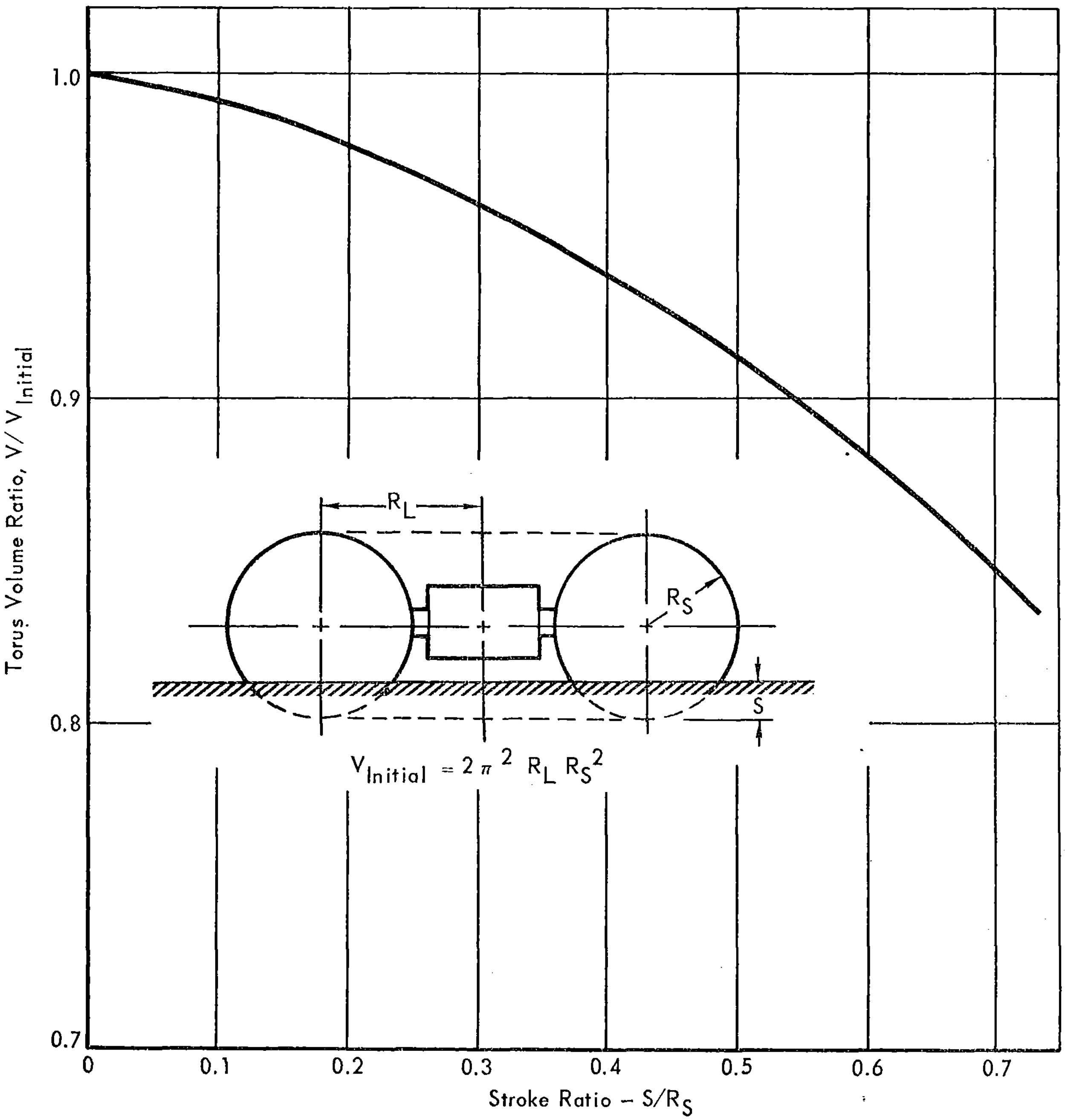


Figure D-5

The normal force acting on the lander can then be expressed as:

$$F = P * A_f$$

where A_f is the torus footprint area. This force acts through the lander's center of gravity and produces no moment about the center of gravity.

Oblique Landing - The torus deflection shape for an oblique landing is more complicated than for a flat landing. An approximate footprint area was determined by intersecting the plane of the landing surface with the undeformed torus. The resulting chordal area was too large, due to roll-up of the torus material relative to the landing surface. To account for this phenomenon, a semi-empirical area reduction factor was applied to the calculated chordal area to predict the torus footprint area.

The semi-empirical area reduction factor was derived from static test data for a inflated torus loaded on end (Reference 2 and 6). These data are presented in Figure D-6. The ratio of the experimentally determined footprinted area (A_f), to the chordal area (ACHORD), is plotted versus the torus stroke ratio (S/R_s). A least squares curve fit was used to obtain the following relationships between the footprint area and chordal area:

$$A_f = \left[.449 + .326 \left(S/R_s\right)^2\right] \text{ ACHORD} \qquad \text{for } S/R_s \leq 1.537$$

$$A_f = \text{ACHORD} \qquad \text{for } S/R_s > 1.537$$

Employing the above expressions, the torus volume at a particular time during the landing was determined from:

$$V_t = V_{t-1} - \frac{1}{2} (S_t - S_{t-1}) \quad (A_{f_t} + A_{f_{t-1}})$$

In the above, the subscripts t and t-1 refer to the current time and the

FOOTPRINT AREA RATIO AS A FUNCTION OF STROKE RATIO

STATIC TEST RESULTS
END LOADING OF TORUS MODEL

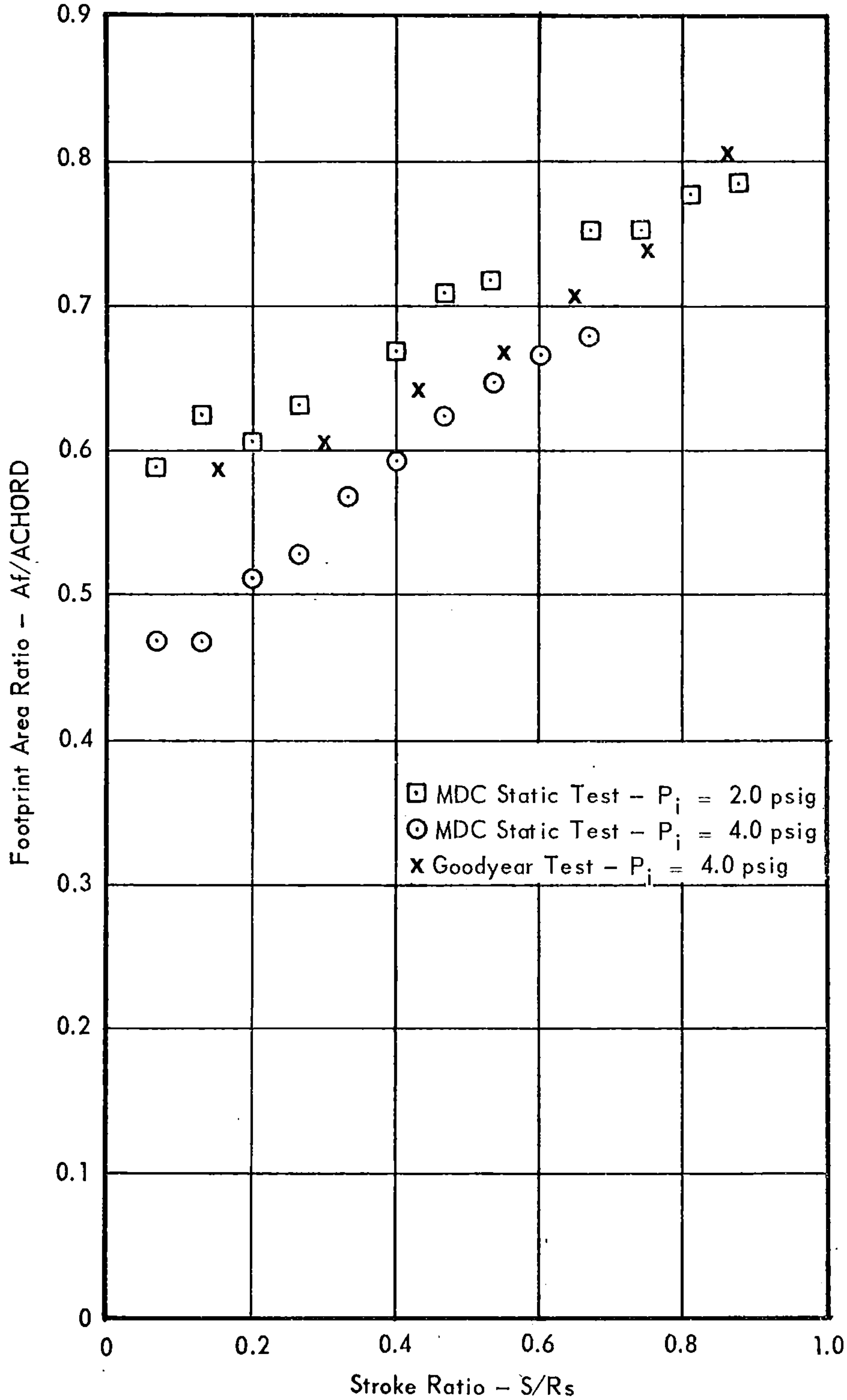


Figure D-6

time at the last integration step. With this expression for the torus volume, the pressure rise and resulting normal force were obtained assuming a polytropic process in a manner similar to that for a flat landing.

For the general oblique landing attitude, (Figure D-7) the footprint area versus stroke was determined in the same manner as for an end landing. The chordal area, obtained for the particular lander attitude of interest was corrected, using the same ratios used for the end landing, to give the torus footprint area. In a manner similar to that of the end landing, the footprint area was used to determine the volume change, pressure rise, and resulting normal force. It was assumed that the normal force acted at the centroid of the footprint area. Figure D-8 indicates the application point of this force relative to the lander coordinate system. This normal force location results in both a moment and force being applied at the lander center of gravity.

A comparison between the analysis and the test data for the static load stroke relationship is shown in Figure D-9. For lander attitudes near a flat landing, the analysis is switched between the flat routine and the oblique analysis whenever the critical lander attitude is reached. This switching is governed by:

$$\beta \text{ (radians)} \leq \frac{\pi}{18} \; (S/R_S) \qquad \text{- flat analysis}$$

$$\beta \text{ (radians)} > \frac{\pi}{18} \; (S/R_S) \qquad \text{- oblique analysis}$$

The methods discussed for determining the normal force require knowledge of an appropriate gas compression process. For static loading

LANDER ATTITUDE

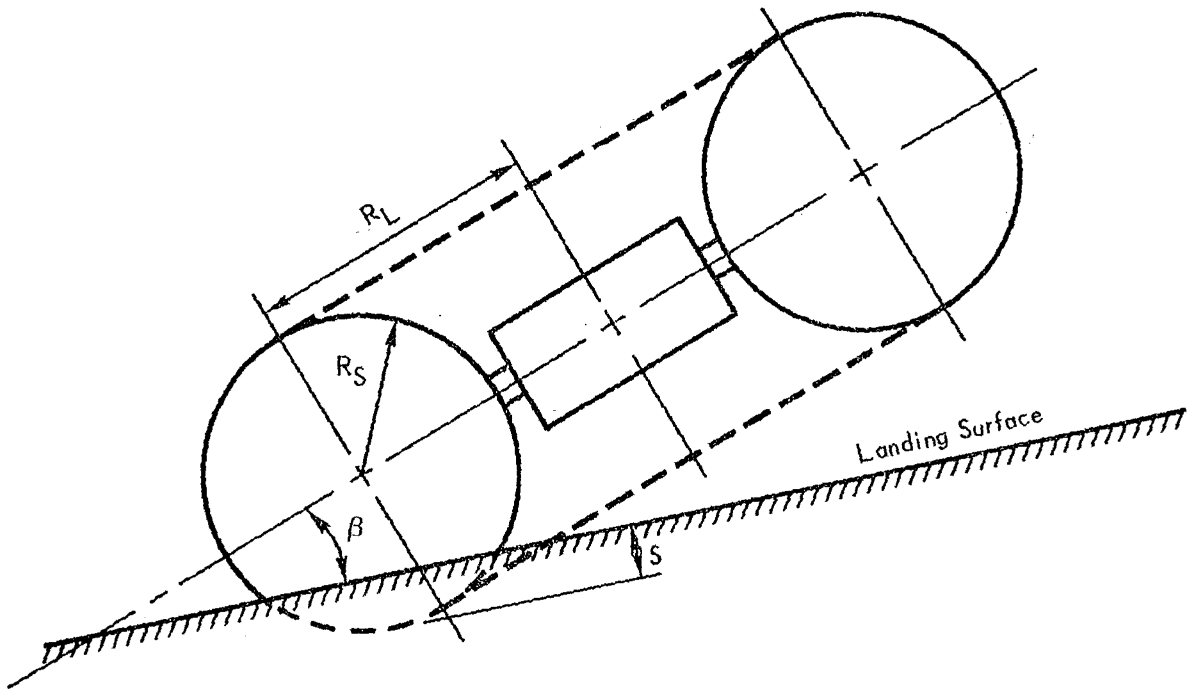


Figure D-7

APPENDIX D

NORMAL FORCE LOCATION

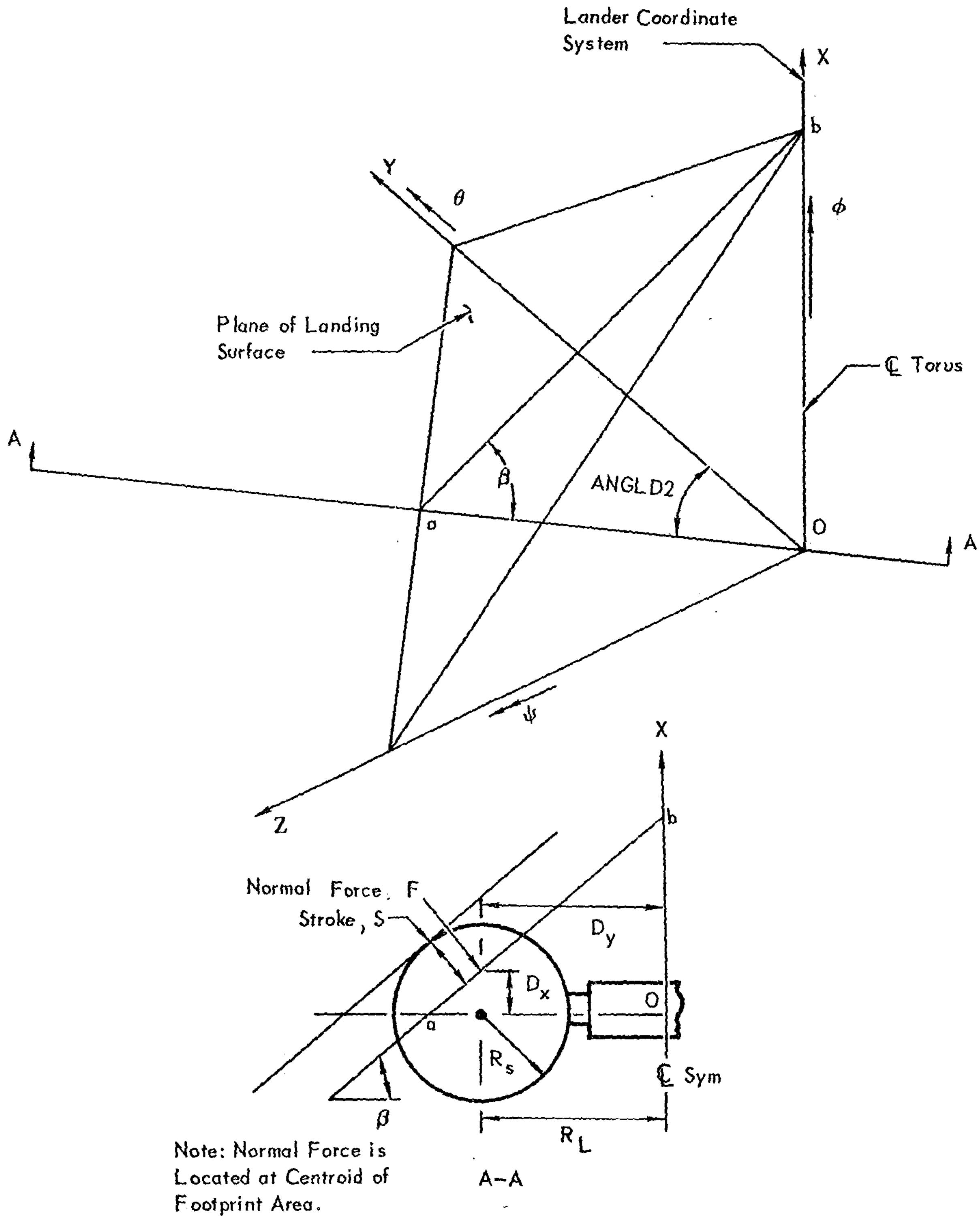


Figure D-8

APPENDIX D

COMPARISON OF PREDICTED WITH MEASURED NORMAL FORCE FOR TORUS MODEL

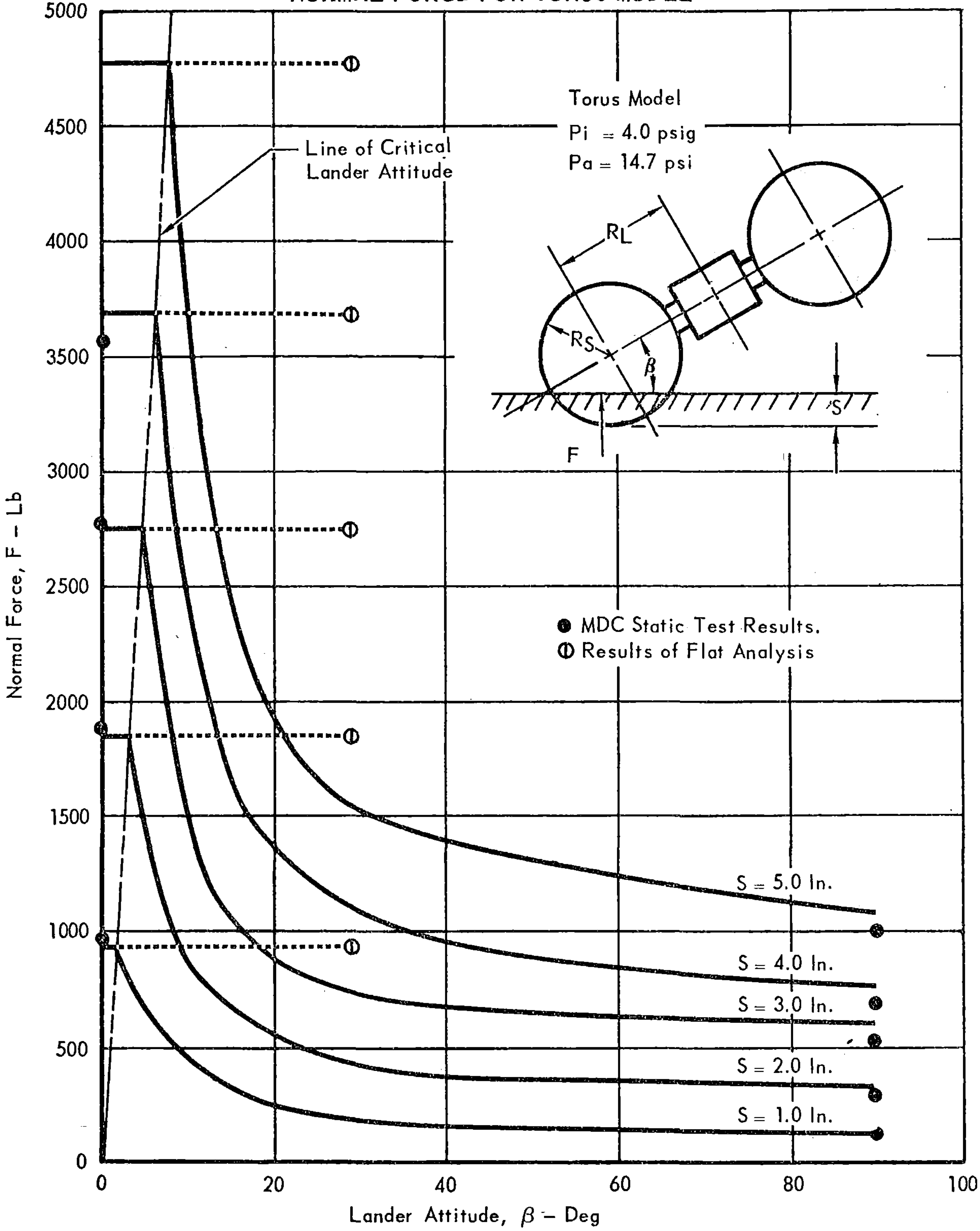


Figure D-9

conditions, an isothermal process ($n = 1$) was found to give good agreement between theory and test. However, in the dynamic case, an isentropic process ($n = k$) was used. This resulted in better agreement with the torus model drop test results (Reference 2) than when an isothermal process was assumed. This trend was indicated during early dynamic testing of aircraft tires (Reference 3 and 4) and analysis of spherical landers (Reference 5).

During the rebound portion of the impact, when the lander is moving away from the surface, the normal force is reduced to account for the hysteresis effects of the bag material. This effect is expressed as

$$F = F - HYST * F_{max} \operatorname{Sin} \frac{\pi S}{S_{max}}$$

where HYST is the bag hysteresis factor and is set with the input data. Data from Reference 2 indicates that a factor of 5 to 10 percent accounts for the hysteresis effects.

The magnitude and location of the normal force, for a given lander stroke and attitude, are determined in the subroutine LOAD (Section D.4.1).

D.2.3.2 Torus Effective Mass. - During the stroking process of a lander impact, the torus material experiences significant motion relative to the payload package. In addition, material is being removed from the dynamic system as the torus flattens against the landing surface. To account for this, an expression for the effective mass of the torus has been formulated. This effective mass is a function of the stroke of the lander payload package and is that portion of the torus mass which is included with the payload mass in the dynamic system's mass.

APPENDIX D

For a flat landing, a Rayleigh energy approach was used to reduce the distributed torus material to an equivalent single mass term. It was assumed that the bag's amplitude of deflection and velocity were of the form shown in Figure D-4. With this assumed velocity distribution, an expression for the material's kinetic energy in terms of the payload stroke was derived, from which the torus effective mass was obtained.

During an end landing, the mass of the torus footprint area was removed from the dynamic system. It was assumed that the remainder of the torus mass is fully effective at the payload center of gravity for this lander attitude. This results in an effective mass which is much larger for an end landing than a flat landing.

For lander attitudes other than flat or end, a mass reduction factor was derived which expressed the effective mass in terms of the total torus mass. This factor was formulated so that the resulting effective mass corresponds with the values obtained for flat and end attitudes. It was assumed that the reduction factor varied proportionately to the change in footprint area between flat and end attitudes.

The calculations required to determine the payload mass and moments of inertia and the torus effective mass are performed in the subroutine MASS, Section D.4.1.

APPENDIX D

D.3 PROGRAM OPERATION

D.3.1 INPUT DATA

D.3.1.1 Input Quantities. - Information describing the geometric and inertia properties of the specific lander to be studied, the landers initial positions and velocities, and a number of indicators to initialize the integration routines are required as input data by the program. This section discusses the required input quantities while the mechanics of setting up the input data, required data card format, input position definition, and the additional option of modifying the output quantities through input indicators is discussed in Section D.3.1.2.

Input parameters are defined in Figure D-10. Many of these quantities are adequately explained in this figure, but a number of them require additional comments.

The program initialization routine assumes that the initial rotations of the lander are carried out in the order of yaw (ψ), pitch (θ), and roll (ϕ). This point must be considered in determining the magnitudes of these rotations to locate the lander at the desired initial angular orientation.

When any of the following indicators are read in as zero, they are reset internally in the program with the nominal values indicated below. This is to guarantee successful initialization of the program routines.

INPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

PROGRAM VARIABLE	LOCATION IN INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
IP	001	–	–	Quantity Used to Determine Initial Integration Time Intervals. $\Delta t = \Delta t_{max}/2.^{+IP}$ (i.e. HZ = HMAX * 2. ** (–IP))
IVARH	002	–	–	Integration Interval Indicator IVARH = 0 – Variable Interval Integration IVARH = 1 – Fixed Interval Integration
IMTH	003	–	–	Integration Method Indicator. IMTH = 0 – Predictor-Corrector IMTH = 1 – Runge-Kutta
EMAX	004	–	–	Maximum Integration Accuracy Required. (Variable Step Predictor-Corrector Integration) See Appendix F.
EMIN	005	–	–	Minimum Integration Accuracy Required. (Variable Step Predictor-Corrector Integration) See Appendix F.
HMIN	006	Δt_{min}	Sec	Minimum Integration Time Interval. (Variable Step Predictor-Corrector Integration)
NTX	007	–	–	Indicator to Suppress Degree of Freedom Along Lander X Axis. NTX = 0 – Allows Degree of Freedom NTX = 1 – Suppresses Degree of Freedom
NTY	008	–	–	Indicator to Suppress Degree of Freedom Along Lander Y Axis (See NTX).
NTZ	009	–	–	Indicator to Suppress Degree of Freedom Along Lander Z Axis (See NTX).
NRX	010	–	–	Indicator to Suppress Degree of Freedom About Lander X Axis. NRX = 0 – Allows Degree of Freedom NRX = 1 – Suppresses Degree of Freedom
NRY	011	–	–	Indicator to Suppress Degree of Freedom About Lander Y Axis (See NRX).
NRZ	012	–	–	Indicator to Suppress Degree of Freedom About Lander Z Axis (See NRX).
MMIC	013	–	–	Mass Moments of Inertia Indicator MMIC = 0 – Constant Inertias MMIC = 1 – Variable Inertias (Not Used)
NCUT	014	–	–	Number of Cutoff Variables Considered (Maximum of 8).
IERPRT	015	–	–	Time History Print Format Indicator IERPRT = 0; Three Lines per Print Time (Option 1). IERPRT = 1; Six Lines per Print Time (Option 2).
XS(J)	019	–	Sec	Real Time Cutoff Limit (T).

Figure D-10

INPUT DATA – INFLATABLE TORIUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION IN INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
IND (J)	020	–	–	Limit Direction Indicator for T. IND (J) = 0 – Limit on Increase. IND (J) = 1 – Limit on Decrease.
XS (J)	025	–	Ft/Sec	Total Surface Velocity Cutoff Limit (TOTSV)
IND (J)	026	–	–	Limit Direction Indicator for TOTSV.
XS (J)	031	–	Ft	Surface Range Cutoff Limit (RANGE).
IND (J)	032	–	–	Limit Direction Indicator for RANGE
XS (J)	037	–	Ft/Sec	Velocity Parallel to Surface Cutoff Limit (TLSV).
IND (J)	038	–	–	Limit Direction Indicator for TLSV.
XS (J)	043	–	Ft	Distance Normal to Surface Cutoff Limit (ZLS).
IND (J)	044	–	–	Limit Direction Indicator for ZDLS.
XS (J)	049	–	Rad/Sec	Roll Rate Cutoff Limit (PHID).
IND (J)	050	–	–	Limit Direction Indicator for PHID.
XS (J)	055	–	Rad/Sec	Pitch Rate Cutoff Limit (THETAD).
IND (J)	056	–	–	Limit Direction Indicator for THSTAD.
XS (J)	061	–	Rad/Sec	Yaw Rate Cutoff Limit (PSID).
IND (J)	062	–	–	Limit Direction Indicator for PSID.
T	067	t_o	Sec	Initial Valve of Time at Start of Integration.
HMAX	068	Δt_{max}	Sec	Maximum Integration Time Interval
KOUNT2	069	–	–	Number of Integration Intervals Between Print Times
NMBNCS	070	–	–	Number of Lander Bounces Desired
XF	073	X_f	Ft	Initial Lander c.g. Location Along X_f Axis
XVF (1)	074	$\dot{X}_f$	Ft/Sec	Initial Lander c.g. Velocity Along X_f Axis
XF	075	Y_f	Ft	Initial Lander c.g. Location Along X_f Axis
XVF (2)	076	$\dot{Y}_f$	Ft/Sec	Initial Lander c.g. Velocity Along X_f Axis
ZF	077	Z_f	Ft	Initial Lander c.g. Location Along Z_f Axis
XYF (3)	078	$\dot{Z}_f$		Initial Lander c.g. Velocity Along Z_f Axis
PHI	079	ϕ	Deg	Initial Lander Angular Position About X Axis following Initial Yaw (ψ) and Pitch (θ)
XD (1,4)	080	$\dot{\phi}$	Rad/Sec	Initial Lander Angular Velocity About X Axis
THETA	081	θ	Deg	Initial Lander Angular Position About Y Axis following Initial Yaw (ψ).
XD (1,5)	082	$\dot{\theta}$	Rad/Sec	Initial Lander Angular Velocity About Y Axis
PSI	083	ψ	Deg	Initial Lander Angular Position About Z Axis
XD (1,6)	084	$\dot{\psi}$	Rad/Sec	Initial Lander Angular Velocity About Z Axis

Figure D-10 (Continued)

INPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION IN INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
RANGE	085	–	Ft	Initial Distance Traveled Over Planet's Surface at Start of the Integration.
SLOPE	086	α	Deg	Angle Between Y_{Is} Axis and Y_f Axis Measured Positive About X_f Axis in the Negative Right-Hand Rotation.
SCFWT	099	–	Lb/Ft^2	Weight per Unit Area of Torus Scuff Material
A	100	A	In.	Payload Geometric Dimension (See Figure D–1).
B	101	B	In.	Payload Geometric Dimension (See Figure D–1).
C	102	C	In.	Payload Geometric Dimension (See Figure D–1).
D	103	D	In.	Payload Geometric Dimension (See Figure D–1).
RHOP	108	–	Lb/Ft^3	Payload Density
VMIN	111	–	Ft/Sec	Sliding Velocity of Footprint Area Below Which the Coefficient of Friction Decreases to Zero.
RDIA	112	D_R	In.	Surface Rook Diameter
DTHE	114	$\Delta\theta$	Rad	Incremental Angle Used in the LOAD Subroutine
AMU	119	μ	–	Coefficient of Friction
FABWT	121	–	Lb/Ft^2	Weight per Unit Area for One Ply of Torus Material
PLY 1	122	PLY 1	–	Number of Torus Material Plys on Inner 120 Degrees (See Figure D–2).
PLY 2	123	PLY 2	–	Number of Torus Material Plys on Upper and Lower 60 Degrees (See Figure D–2).
PLY 3	124	PLY 3	–	Number of Torus Material Plys on Outer 120 Degrees (See Figure D–2).
RL	125	R_L	In.	Radius of the Lander (See Figure D–1).
RS	126	R_S	In.	Radius of Torus (See Figure D–1).
HYST	127	HYST	–	Torus Hysteresis Factor
PI	128	P_i	Psig	Initial Torus Inflation Pressure
PA	129	P_a	Psi	Atmosphere Pressure
PLTMAS	130	–	$Lb\text{-}Sec^2/Ft$	Planet Mass (Used to Calculate Planet's Acceleration of Gravity)
PLTRAD	131	–	Ft	Planet Radius – Distance from Planet c.g. to Origin of Fixed Surface Coordinates. (Used to Calculate Planet's Acceleration of Gravity)
GASCNT	132	n	–	Gas Constant Which Yields Assumed Gas Compression Process: n = 1 – Isothermal Process; n = k – Isentropic Process – Where k is the Specific Heat Ratio for the Particular Inflation Gas of Interest.

Figure D-10 (Continued)

INPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	LOCATION IN INPUT FIELD	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
Note – Last data card of this type must contain a 1 in Column 2.				

The following information is required on each data card requesting a nonstandard output variable (See Section D.3.1).

VNAME	Name printed out to identify the nonstandard variable.
LCOMN	Subscript of the common array COMINT element in which the desired nonstandard output variable is located.
LPT	Print position in time history block in which desired output variable will be printed.

Note – Last data card in a data set must contain EOD in columns 2–5.

Figure D-10 (Continued)

EMAX = 0	EMAX set equal to 1×10^{-4} (see Appendix F)
EMIN = 0	EMIN set equal to 1×10^{-6} (see Appendix F)
HMIN = 0	HMIN set equal to HMAX * 2^{-16} seconds
NMBNCS = 0	NMBNCS set equal to 100
DTHE = 0	DTHE set equal to 0.03 radians

In addition, if either NCUT or VMIN is initially zero, the program will terminate with an error message.

At the present time, the program considers a lander whose moments of inertia are constant with time. Therefore, the mass moments of inertia indicator, MMIC, must be set equal to zero.

Several methods for obtaining computational termination for a particular lander case are provided in the program. One feature results in termination when any one of up to eight variables reach a preset cutoff value. The eight cutoff variables are as follows:

- o Real time (T)
- o Total surface velocity (TOTSV)
- o Surface range (RANGE)
- o Velocity parallel to surface (TLSV)
- o Distance normal to surface (ZLS)
- o Roll rate (PHID)
- o Pitch rate (THETAD)
- o Yaw rate (PSID)

XS (J) is the array which contains the cutoff value for each cutoff variable. The array IND (J) contains indicators which define whether the cutoff variables are increasing or decreasing toward the cutoff values. These indicators are set as follows:

IND (J) = 0 - Cutoff variable increasing to cutoff limit.

IND (J) = 1 - Cutoff variable decreasing to cutoff limit.

In these two arrays, the subscript J indicates the order in which these quantities are read into the program. The time history variable to which a specific cutoff value applies is governed by its location in the data field.

The number of lander impacts to be considered in a particular case is governed by the input quantity NMBNCS. The program terminates when the lander leaves the ground at the end of the last impact of interest. In addition, a specific case is terminated if the clearance between the lander payload and the surface rocks become less than zero.

The quantities PLTRAD and PLTMAS are used to determine the planet's acceleration of gravity as a function of altitude. This relationship is given as:

$$GZ = g = \frac{-G * PLTMAS}{(PLTRAD - Z_f)^2}$$

In this expression:

$GZ = g =$ acceleration of gravity (ft/sec^2).

$G =$ Universal Gravity Constant (1.0684×10^{-9} ft^3/lb-sec^2).

PLTMAS = Planet's mass (lb-sec^2/ft).

PLTRAD = Planet's radius (ft).

Z_f = Position of lander center of gravity in local surface Z axis.

An example of the required date setup is given in Section D.3.3. Also shown are the various options which are available for the output data format.

D.3.1.2 Input Format. - All of the data cards required for the execution of any one case are referred to as a data set. As a data set is read, the values of the input variables with appropriate labels and messages are printed for future reference when evaluating the resulting lander time history output. A data set must be terminated by a data card containing EOD in columns 2-4 with columns 5-11 blank. There is no program limit to the number of data sets that can be run during any one job.

Figure D-11 shows the required format for the input data cards. The input data can be thought of as a one dimensional array whose elements contain the values of the various input quantities. The input routine is completely compatible with the Crushable Torus Landing Loads and Motions Program (Appendix B), and this fact accounts for the apparent blanks in the input format. Many of the locations in the input field are used to input quantities required with the analysis of the crushable system. These input positions are noted in Figure D-11.

A data set for the Inflatable Torus Landing Loads and Motions Program may consist of three different types of data cards. These three groups of cards are described below in the order they are required within a data set.

Type 1:

In any one data set, the first three cards are always used as a description of the case to be run. These cards have the following format:

FORMAT FOR INPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

Card	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
TITLE CARD						
TITLE CARD						
TITLE CARD						
001	IP	IVARH	IMTH	EMAX	EMIN	HMIN
007	NTY	NTY	NTZ	NRX	NRY	NRZ
013	MMIC	NCUT	IERPRT			
019	XS(J)	IND(J)				
025	XS(J)	IND(J)				
031	XS(J)	IND(J)				
037	XS(J)	IND(J)				
043	XS(J)	IND(J)				
049	XS(J)	IND(J)				
055	XS(J)	IND(J)				
061	XS(J)	IND(J)				
067	T	HMAX	KOUNT2	NMBNCS		
073	XF	XVF(1)	YF	XVF(2)	ZF	XVF(3)
079	PHI	XD(1,4)	THETA	XD(1,5)	PSI	XD(1,6)
085	RANGE	SLOPE				
097		(1)	SCFWT	A	B	C
103	D	(1)	(1)	(1)	(1)	RHOP
109	(1)	(1)	VMIN	RDIA	(1)	DTHE
115	(1)		(1)		AMU	(1)
121	FABWT	PLY1	PLY2	PLY3	RL	RS
1 127	HYST	P	PA	PLTMAS	PLTRAD	GASCNT
VNAME	LCOMN	LPT				
EOD						

(1) – These positions may not be used for inflatable torus input data.

AC 7701-69

Figure D-11

Column 1:	Must be left blank
Columns 2-55:	May contain any descriptive comments which will be printed at the beginning of the input data listing.
Columns 56-80:	Are not read by the program and may be used for comments or identification statements.

Type 2:

The assignment of initial values to the program variables by input data is accomplished with this format. All data values read from these cards are first stored in the array DATA. The various input variables are then assigned their respective initial values by equating the variables to a specified DATA element. Since the DATA elements are initially set equal to zero, an input variable will have an initial value of zero unless the corresponding DATA element is changed by card input. The few exceptions to this are the indicators discussed in Section D.3.1.1.

The input quantities contained in the six data fields of a data card are stored consecutively in the DATA array. Each card contains a subscript which governs the data position (in the array DATA) of each variable on that card. The data position of an input variable is obtained by assigning the first variable on a card a data position equal to the card subscript. Each following variable, moving from left to right on the card, has a data position one increment larger. For example, (Figure D-11), the coefficient of friction (AMU) has a data position of 119. Thus, a particular data card needed to define a given variable is identified by the card subscript.

In the instance of multiple data sets, all of the previous data set is retained except those quantities changed through reading in cards of Type 2. However, each data set must contain at least one card of Type 2.

The following describes the required format for the Type 2 input cards.

Column 2:	Read as an integer number with an I1 format. If the value is zero, another card is read in the same format. If the value is not zero, the card represents the last card of Type 2 in the data set. The last Type 2 card in a data set must contain a non-zero value in this column.
Columns 5-7:	Read as an integer number with an I3 format. This field represents the data card subscript for the particular card.
Columns 10-19, 20-29, 30-39, 40-49, 50-59, and 60-69:	Read as a real variable with an E10.5 format. All data values must be input as floating point numbers. Integer variables are converted from floating point quantities by the program.

Columns 1, 8-9, and 70-80:	These are not read by the program and may be used for comments and identification statements.

Type 3:

Information on these cards is used when the optional output routine is desired. This option allows a different variable to temporarily replace a standard output variable in that variable's output position.

There may be as many as 72 input cards of this type. Each card contains an idendification name (one to ten characters), a subscript locating the desired output variable in the program array COMINT, and an indicator giving the desired print position of the variable.

These data cards have the following format.

Columns 2-11:	The ten characters in these columns are used as the identification name for the variable to be printed.
Columns 15-19:	Read as an integer number with an I5 format. This value is the subscript of the common array COMINT location of the desired output variable (see Appendix E).
Columns 21-25:	Read as an integer number with a I5 format. This value is used to specify the print position in which the desired output variable will be

	printed. The output routine's print positions are defined in Section D.3.2.
Columns 1, 12-14, 20, and 26-80:	These positions are not read by the program and may be used for comments and identification statements.

For a multi-case computer run, if the output format is modified in a data set, the succeeding cases will have the same modification unless changed by their data set. New runs revert to the standard format unless they are modified by their data set.

D.3.2 OUTPUT DATA

D.3.2.1 Output Quantities. - At specified times during the integration routine, various time varying quantities defining the lander's positions, velocities, and accelerations; applied forces and moments; and other items of interest such as torus pressure, volume, and footprint area are printed. In addition, an output option is available whereby the standard output variables may be replaced by other quantities of interest. This procedure is discussed in detail as Type 3 in Section D.3.1.2. The output information is printed by a call to the PRINT subroutine. The output format resulting from this subroutine is discussed more fully in the following Section, D.3.2.2.

Initial output from the program presents the input data read in for the specified case being considered. Following this information are the weight and inertia properties of the lander. The time histories describing the lander's motion are then given.

Figure D-12 shows the form of the standard heading which appears at the top of each page of time history output. The values of the output variables appear as blocks of data following this heading and are printed in the same format as the heading. At the top of each block of output, a line of information giving the Central Processor (CP) time since the start of the job, the real time, and the present integration step size is printed. Following these is the block of output quantities corresponding to the printed value of real time.

Figure D-13 defines the standard output parameters which are available in the program. Also shown are the appropriate units of the variable and its respective print position. The top line of the output consists of print positions 1-12, the second line consists of print positions 13-24, and so forth to line six which consists of positions 61-72.

D.3.2.2 Output Format. - At the end of specified integration intervals, the various time history quantities of interest are printed. The print times are governed by the input indicator KOUNT2. Two formats are available in the output routine: three lines of output containing 36 variables (Option 1) or six lines containing 72 variables (Option 2). Option 1 prints the top three lines of output defined in Figure D-12, while Option 2 prints all six lines as shown in this figure. The output format is governed by the input quantity IERPRT.

IERPRT	=	0	Print Option 1
IERPRT	=	1	Print Option 2

STANDARD OUTPUT HEADING

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

LOCAL SURFACE COORDINATE SYSTEM					LANDER COORDINATE SYSTEM								
XLS	XDLS	XDDLS	ANGLD1	*	XD	XN	PHID	PHIDD	*	STROKE	XAG	AF	DISTX
YLS	YDLS	YDDLS	ANGLD2	*	YN	YN	THETAD	THETADD	*	FORCE	YAG	VOL	DISTY
ZLS	ZDLS	ZDDLS	ANGLD3	*	ZD	ZN	PSID	PSIDD	*	PRESS	ZAG	RSXCF	FRIRA
ANG (1,1)	ANG (1,2)	ANG(1,3)	XLPF(1)	*	FF(1)	FF(4)	FS(1)	XPF(1)	*	XCF (1)	ACLR	R1	E
ANG (2,1)	ANG (2,2)	ANG(2,3)	XLPF(2)	*	FF(2)	FF(5)	FS(2)	XPF(2)	*	XCF (2)	XMASST	R2	F
ANG (3,1)	ANG (3,2)	ANG(3,3)	XLPF(3)	*	FF(3)	FF(6)	FS(3)	XPF(3)	*	XCF (3)	FIFLAT	R3	THETAT

Figure D-12

STANDARD OUTPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
		LANDER WEIGHT AND INERTIA PROPERTIES		
WTERTH			Lb	Lander Weight on Earth
WTPLNT			Lb	Lander Weight on Planet
XXI			Lb Ft Sec^2	Lander Moment of Inertia About X Axis
YYI			Lb Ft Sec^2	Lander Moment of Inertia About Y Axis
ZZI			Lb Ft Sec^2	Lander Moment of Inertia About Z Axis
XYI			Lb Ft Sec^2	Lander XY Product of Inertia
XZI			Lb Ft Sec^2	Lander XZ Product of Inertia
YZI			Lb Ft Sec^2	Lander YZ Product of Inertia
XXID			Lb Ft Sec	Rate of Change of XXI (Not Used)
YYID			Lb Ft Sec	Rate of Change of YYI (Not Used)
ZZID			Lb Ft Sec	Rate of Change of ZZI (Not Used)
XYID			Lb Ft Sec	Rate of Change of XYI (Not Used)
XZID			Lb Ft Sec	Rate of Change of XZI (Not Used)
YZID			Lb Ft Sec	Rate of Change of YZI (Not Used)
		INTEGRATION TIME HISTORY PARAMETERS (The Following Times Appear at the Top of Each Block of Output)		
TIMECP		CP TIME	Sec	Computer Central Processor Elapsed Time
T			Sec	Real Time Elapsed (Starts at Initial Input Time)
HZ		ΔT	Sec	Integration Step Time Interval
		LOCAL SURFACE COORDINATE SYSTEM		
XLS	1	X_{ls}	Ft	Lander Center of Gravity (C.G.) X_{ls} Location
XDLS	2	$\dot{X}_{ls}$	Ft/Sec	Lander C.G. Velocity, X_{ls} Component
XDDLS	3	$\ddot{X}_{ls}$	Ft/Sec^2	Lander C.G. Acceleration, X_{ls} Component
YLS	13	Y_{ls}	Ft	Lander C.G. Y_{ls} Location

Figure D-13

STANDARD OUTPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
		LOCAL SURFACE COORDINATE SYSTEM (Continued)		
YDLS	14	$\dot{Y}_{ls}$	Ft/Sec	Lander C.G. Velocity, Y_{ls} Component
YDDLS	15	$\ddot{Y}_{ls}$	Ft/Sec^2	Lander C.G. Acceleration, Y_{ls} Component
ZLS	25	Z_{ls}	Ft	Lander C.G. Z_{ls} Location
ZDLS	26	$\dot{Z}_{ls}$	Ft/Sec	Lander C.G. Velocity, Z_{ls} Component
ZDDLS	27	$\ddot{Z}_{ls}$	Ft/Sec^2	Lander C.G. Acceleration, Z_{ls} Component
ANGLD1	4	–	Deg	Dihedral Angle Between Lander Y-Z Plane and $X_{ls} - Y_{ls}$ Plane
ANGLD2	16	–	Deg	Dihedral Angle Between Lander X-Y Plane and Plane Defined by X and Z_{ls} Axes
ANGLD3	28	–	Deg	Angle Between Friction Force Direction and X_{ls} Axis
ANG (1,1)	37	–	Deg	Direction Angle Between X_{ls} and X Axes.
ANG (1,2)	38	–	Deg	Direction Angle Between X_{ls} and Y Axes.
ANG (1,3)	39	–	Deg	Direction Angle Between X_{ls} and Z Axes.
ANG (2,1)	49	–	Deg	Direction Angle Between Y_{ls} and X Axes.
ANG (2,2)	50	–	Deg	Direction Angle Between Y_{ls} and Y Axes.
ANG (2,3)	51	–	Deg	Direction Angle Between Y_{ls} and Z Axes.
ANG (3,1)	61	–	Deg	Direction Angle Between Z_{ls} and X Axes.
ANG (3,2)	62	–	Deg	Direction Angle Between Z_{ls} and Y Axes.
ANG (3,3)	63	–	Deg	Direction Angle Between Z_{ls} and Z Axes.
XLPF (1)	40	–	Ft	Normal Force Location in X_{ls} Direction
XLPF (2)	52	–	Ft	Normal Force Location in Y_{ls} Direction
XLPF (3)	64	–	Ft	Normal Force Location in Z_{ls} Direction
		LANDER COORDINATE SYSTEM.		
XD	5	$\dot{X}$	Ft/Sec	X Component of Lander C.G. Velocity
YD	17	$\dot{Y}$	Ft/Sec	Y Component of Lander C.G. Velocity
ZD	29	$\dot{Z}$	Ft/Sec	Z Component of Lander C.G. Velocity
XN	6	N_x	Planet g's	X Component of Load Factor at Lander C.G.
YN	18	N_y	Planet g's	Y Component of Load Factor at Lander C.G.
ZN	30	N_z	Planet g's	Z Component of Load Factor at Lander C.G.
PHID	7	$\dot{\phi}$	Rad/Sec	Rotational Velocity About X Axis
PHIDD	8	$\ddot{\phi}$	Rad/Sec^2	Rotational Acceleration About X Axis

Figure D-13 (Continued)

STANDARD OUTPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
LANDER COORDINATE SYSTEM (Continued)				
THETAD	19	$\dot{\theta}$	Rad/Sec	Rotational Velocity About Y Axis
THETADD	20	$\ddot{\theta}$	Rad/Sec^2	Rotational Acceleration About Y Axis
PSID	31	$\dot{\psi}$	Rad/Sec	Rotational Velocity About Z Axis
PSIDD	32	$\ddot{\psi}$	Rad/Sec^2	Rotational Acceleration About Z Axis
FF (1)	41	–	Lb	Force Applied at Lander C.G., X Component
FF(2)	42	–	Lb	Force Applied at Lander C.G., Y Component
FF(3)	53	–	Lb	Force Applied at Lander C.G., Z Component
FF(4)	54	–	Ft-Lb	Moment Applied at Lander C.G., Component About X Axis.
FF(5)	65	–	Ft-Lb	Moment Applied at Lander C.G., Component About Y Axis.
FF(6)	66	–	Ft-Lb	Moment Applied at Lander C.G., Component About Z Axis.
FS(1)	43	–	Lb	X_{ls} Component of Friction Force
FS (2)	55	–	Lb	Y_{ls} Component of Friction Force
FS (3)	67	–	Lb	Normal Force in Z_{ls} Direction
XPF (1)	44	–	Ft	Normal Force Location in X Direction
XPF (2)	56	–	Ft	Normal Force Location in Y Direction
XPF (3)	68	–	Ft	Normal Force Location in Z Direction
ADDITIONAL OUTPUT INFORMATION				
STROKE	9	S	In.	Attenuator Stroke
FORCE	21	F	Lb	Force Normal to Landing Surface due to Compressing Inflatable Torus
PRESS	33	P	Psig	Torus Inflation Pressure
XAG	10	g_x	Ft/Sec^2	Component of Acceleration of Gravity in X Direction
YAG	22	g_y	Ft/Sec^2	Component of Acceleration of Gravity in Y Direction
ZAG	34	g_z	Ft/Sec^2	Component of Accelerator of Gravity Z Direction
AF	11	A_f	$In.^2$	Footprint Area
VOL	23	V	$In.^3$	Torus Internal Volume
RSXCF	35	–	Ft/Sec	Velocity of Footprint Area Centroid Relative to Landing Surface
DISTX	12	D_x	In.	Location of Footprint Area Centroid in X Direction
DISTY	24	D_y	In.	Location of Footprint Area Centroid in Y Direction
FRIRA	76	R_f	In.	Radius Defining Torque Due to Friction During Spinning

Figure D-13 (Continued)

STANDARD OUTPUT DATA – INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM (Continued)

PROGRAM VARIABLE	PRINT POSITION	ANALYSIS SYMBOL	UNITS	VARIABLE DEFINITION
		ADDITIONAL OUTPUT INFORMATION (Continued)		
XCF(1)	45	–	Ft/Sec	X_{ls} Component of Footprint Area Centroid Velocity
XCF(2)	57	–	Ft/Sec	Y_{ls} Component of Footprint Area Centroid Velocity
XCF(3)	69	–	Ft/Sec	Z_{ls} Component of Footprint Area Centroid Velocity
ACLR	46	–	In.	Remaining Lander Clearance
XMASST	58	m	$Lb\text{-}Sec^2/Ft$	Effective Lander Mass
FIFLAT	70	–	–	Lander Attitude Indicator (IFLAT) FIFLAT = 0; Oblique Landing FIFLAT = 1; Flat Landing
R1	47	R_1	In.	Torus Deflection Parameter for Flat Landing (See Figure D-4)
R2	59	R_2	In.	Torus Deflection Parameter for Flat Landing (See Figure D-4)
R3	71	R_3	In.	Torus Deflection Parameter for Flat Landing (See Figure D-4)
E	48	e	In.	Torus Deflection Parameter for Flat Landing (See Figure D-4)
F	60	f	In.	Torus Deflection Parameter for Flat Landing (See Figure D-4)
THETAT	72	θ_t	Rad	Torus Deflection Parameter for Flat Landing (See Figure D-4)

Figure D-13 (Continued)

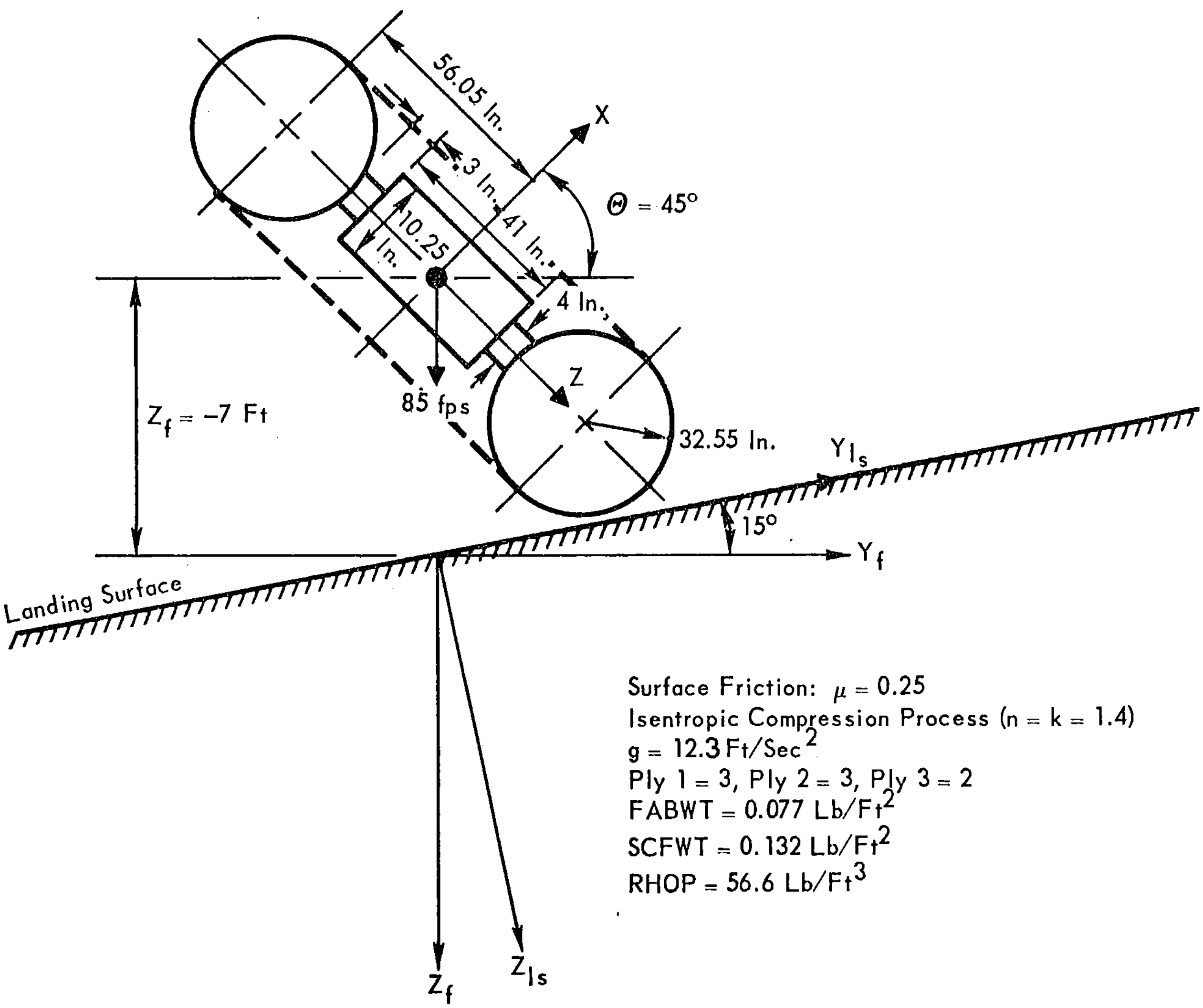
Internally, the subroutine PRINT uses the time history variable's program name and the subscript defining the variable's location in the common array COMINT to print the variable. It is important to note that only real variables equivalenced in the common array COMINT can be printed.

When more than one cutoff variable (NCUT>1) is employed with the variable step Predictor-Corrector integration routine, the time history data must be carefully evaluated. In this situation, if the integration error tolerance is exceeded and the integration step size reduced, the output over a portion of the previous real time is repeated. The last time history quantities listed for a specific real time (T) should be used since they are the most accurate. This does not occur when only one cutoff parameter is specified or with the Runge-Kutta or fixed step Predictor-Corrector integration routines. Program termination for number of bounces (NMBNCS) or payload clearance is separate and is not counted in NCUT.

Examples of the output format for the various options are shown in Section D.3.3.

D.3.3 EXAMPLE OF PROGRAM OPERATION - The required input data card format, resulting print out of the input data, and typical pages of time history output for two data sets are shown in the following figures. In these two cases, the torus geometry and initial conditions are identical, only the requested output format was changed to show the two available print options. The inflatable torus baseline design (Section 6.4) was assumed and the initial conditions for the landing configuration considered are shown in Figure D-14.

EXAMPLE LANDING CONDITION



Surface Friction: $\mu = 0.25$
Isentropic Compression Process (n = k = 1.4)
g = 12.3 Ft/Sec^2
Ply 1 = 3, Ply 2 = 3, Ply 3 = 2
FABWT = 0.077 Lb/Ft^2
SCFWT = 0.132 Lb/Ft^2
RHOP = 56.6 Lb/Ft^3

Figure D-14

Figure D-15 shows the two sets of input data cards. The first case inputs the physical data and requests the print out to be that of Option 2. The second case requests Option 1 and the displacements, velocities, and accelerations of the lander center of gravity, normally expressed in local surface coordinates (X_{1s}, Y_{1s}, Z_{1s}) are replaced by these quantities defined in the gravity coordinate system (X_f, Y_f, Z_f).

It was not necessary to input the physical data again for the second case. All that was required was the resetting of the print indicator (IERPRT) to change the print format from Option 2 to Option 1. In addition, the information required to change the coordinate system output variables is shown.

A fixed step, Predictor-Coordinator integration routine was used with a print out at the end of each integration step. The problem was setup to run as a planar case with the degree of freedom along the lander Y axis and those about the lander X and Z axes being suppressed. A time history for the first lander impact was requested.

Figures D-16 and D-17 present the input data print out and the first page of time history data for Option 2. This same information for Option 1 is given in Figures D-18 and D-19. Both time histories show the initial impact of the lander with the ground.

Approximately 30 seconds were required to compile and load the program. With the fixed step, Predictor-Corrector routine, 15 to 20 seconds of computer time were required to analyze this typical single bounce case. The Runge-Kutta routine requires additional run time while the variable step, Predictor-Corrector procedure is the most economical in terms of computer time.

APPENDIX D

INFLATABLE TORUS LANDING LOADS & MOTIONS PROGRAM

EXAMPLE INPUT DATA

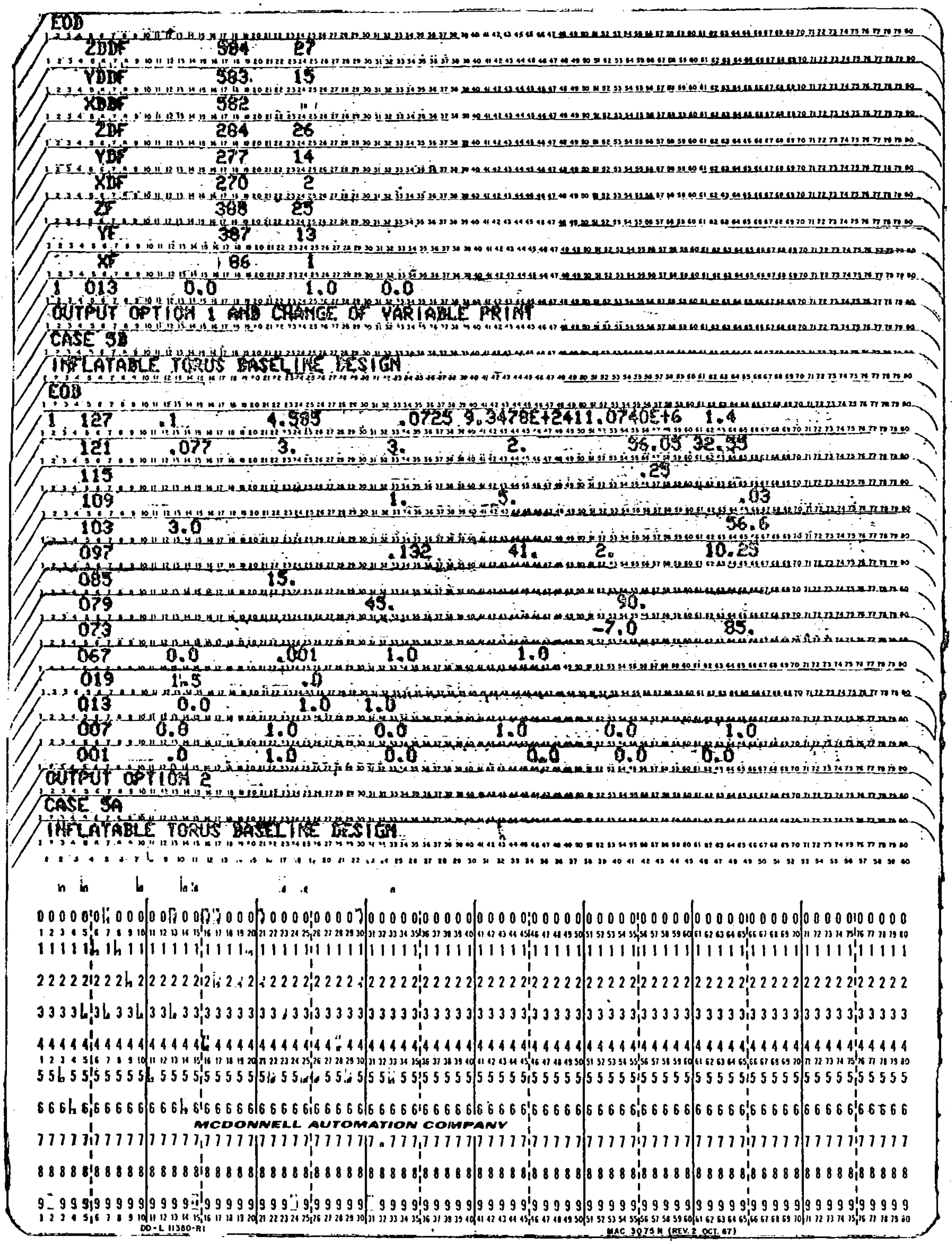


Figure D-15

APPENDIX D

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
Print Out of Input Data
(Example 1)

```
                    INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
                    MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1
                    MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION
                                        INPUT DATA

CASE 5A
INFLATABLE TORUS BASELINE DESIGN
OUTPUT OPTION 2

DATA SUBSCRIPT   IP              IVAR1          IMTH           FMAX           FMIN           HMIN
      1          0.              1.0000000E+00  0.             0.             0.             0.

DATA SUBSCRIPT   NTX             NTY            NTZ            NRX            NRY            NRZ
      7          0.              1.0000000E+00  0.             1.0000000E+00  0.             1.0000000E+00

DATA SUBSCRIPT   MMIC            NCUT           IERPRT
     13          0.              1.0000000E+00  1.0000000E+00  -0.            -0.            -0.

DATA SUBSCRIPT   T               T
                 XS(I)           IND(I)
     19          1.5000000E-01   0.             -0.            -0.            -0.            -0.

DATA SUBSCRIPT   T               HMAX           KOUNT2         NMBNCS
     67          0.              1.0000000E-03  1.0000000E+00  1.0000000E+00  -0.            -0.

DATA SUBSCRIPT   XF              XVF(1)         YF             XVF(2)         ZF             XVF(3)
     73          0.              0.             0.             0.             -7.0000000E+00 8.5000000E+01

DATA SUBSCRIPT   PHI             PHID           THETA          THETAD         PSI            PSID
     79          0.              0.             4.5000000E+01  0.             9.0000000E+01  0.

DATA SUBSCRIPT   RANGE           SLOPE
     65          0.              1.5000000E+01  -0.            -0.            -0.            -0.

DATA SUBSCRIPT                                  SCFWT          A              B              C
     97          -0.             -0.            1.3200000E-01  4.1000000E+01  2.0000000E+00  1.0250000E+01

DATA SUBSCRIPT   D               E              F              G              RHOL           RHOP
    103          3.0000000E+00   -0.            -0.            -0.            -0.            5.6600000E+01

DATA SUBSCRIPT   POW             RSHCP          VMIN           RDIA           DX             DTHE
    109          -0.             -0.            1.0000000E+00  5.0000000E+00  -0.            3.0000000E-02

DATA SUBSCRIPT   RNORM           ACLR           ELST           LDSTRI         AMU            XKW
    115          -0.             -0.            -0.            -0.            2.5000000E-01  -0.

DATA SUBSCRIPT   FABWT           PLY1           PLY2           PLY3           RL             RS
    121          7.7000000E-02   3.0000000E+00  3.0000000E+00  2.0000000E+00  5.6050000E+01  3.2550000E+01

DATA SUBSCRIPT   HYST            PT             PA             PLTMAS         PLTRAD         GASCNT
    127          1.0000000E-01   4.5850000E+00  7.2500000E-02  9.3478000E+24  1.1074000E+07  1.4000000E+00

EOD              END OF DATA LIST

  VEHICLE WEIGHT, ON EARTH =  6.6014778E+02, ON PLANET =  2.5259418E+02

                 MOMENTS AND PRODUCTS OF INERTIA

        XXI             XYI            XXID           XYID
        YYI             XZI            YYID           XZID
        ZZI             YZI            ZZID           YZID

  2.2562788E+02  0.              0.             0.
  1.4476933E+02  0.              0.             0.
  1.4476933E+02  0.              0.             0.
```

Figure D-16

APPENDIX D

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

Typical Time History Output Page
Print Option 2
(Example 1)

LOCAL SURFACE COORDINATE SYSTEM				LANDER COORDINATE SYSTEM							
XLS	XDLS	XDDLS	ANGLD1 *	XD	XN	PHID	PHIDD *	STROKE	XAG	AF	DISTX
YLS	YDLS	YDDLS	ANGLD2 *	YD	YN	THETAD	THETADD *	FORCE	YAG	VOL	DISTY
ZLS	ZDLS	ZDDLS	ANGLD3 *	ZD	ZN	PSID	PSIDD *	PRESS	ZAG	RSXCF	FRIRA
ANG(1,1)	ANG(1,2)	ANG(1,3)	XLPF(1) *	FF(1)	FF(4)	FS(1)	XPF(1) *	XCF(1)	ACLR	R1	E
ANG(2,1)	ANG(2,2)	ANG(2,3)	XLPF(2) *	FF(2)	FF(5)	FS(2)	XPF(2) *	XCF(2)	XMASST	R2	F
ANG(3,1)	ANG(3,2)	ANG(3,3)	XLPF(3) *	FF(3)	FF(6)	FS(3)	XPF(3) *	XCF(3)	FIFLAT	R3	THETAT
CP TIME =	43.999	T =	0.	DT =	1.000E-03						
-0.	0.	0.	0.	-6.010E+01	0.	0.	0.	-4.705E-02	-8.698E+00	0.	0.
1.812E+00	-2.200E+01	-3.184E+00	0.	0.	0.	-0.	0.	0.	0.	0.	0.
-6.761E+00	8.210E+01	1.188E+01	0.	6.010E+01	0.	0.	0.	4.585E+00	8.698E+00	0.	0.
9.000E+01	1.800E+02	9.000E+01	0.	0.	0.	0.	0.	0.	5.473E+01	0.	-0.
3.000E+01	9.000E+01	6.000E+01	0.	0.	0.	0.	0.	0.	2.054E+01	0.	-0.
1.200E+02	9.000E+01	3.000E+01	0.	0.	0.	0.	0.	0.	0.	0.	0.
CP TIME =	44.285	T =	4.775E-05	DT =	1.000E-03						
0.	0.	0.	1.200E+02	-6.010E+01	0.	0.	0.	-3.241E-06	-8.698E+00	1.498E+02	1.598E+01
1.811E+00	-2.200E+01	-3.184E+00	9.000E+01	0.	0.	0.	0.	0.	0.	1.172E+06	8.364E+01
-6.758E+00	8.210E+01	1.188E+01	9.000E+01	6.010E+01	0.	0.	0.	4.585E+00	8.698E+00	2.197E+01	2.439E+00
9.000E+01	1.800E+02	9.000E+01	1.192E-14	0.	0.	0.	-1.321E+00	0.	5.469E+01	0.	-0.
3.000E+01	9.000E+01	6.000E+01	2.331E+00	0.	0.	0.	-1.192E-14	-2.197E+01	2.054E+01	0.	-0.
1.200E+02	9.000E+01	3.000E+01	6.679E+00	0.	0.	0.	6.950E+00	0.	0.	0.	0.
CP TIME =	44.417	T =	1.048E-03	DT =	1.000E-03						
0.	0.	0.	1.200E+02	-6.011E+01	7.886E-01	0.	0.	9.853E-01	-8.698E+00	1.571E+02	1.597E+01
1.789E+00	-2.200E+01	5.294E+00	9.000E+01	0.	0.	4.481E-03	1.990E+01	7.204E+02	0.	1.172E+06	8.361E+01
-6.675E+00	8.211E+01	-2.360E+01	9.000E+01	6.011E+01	-8.156E-01	0.	0.	4.585E+00	8.698E+00	2.197E+01	2.557E+00
9.000E+01	1.800E+02	9.000E+01	1.191E-14	5.162E+02	0.	-9.191E-10	-1.319E+00	0.	5.370E+01	0.	-0.
3.000E+01	9.000E+01	6.000E+01	2.331E+00	0.	2.881E+03	1.801E+02	-1.191E-14	-2.197E+01	2.036E+01	0.	-0.
1.200E+02	9.000E+01	3.000E+01	6.675E+00	-5.338E+02	0.	-7.204E+02	6.947E+00	0.	0.	0.	0.
CP TIME =	44.563	T =	2.048E-03	DT =	1.000E-03						
0.	0.	0.	1.200E+02	-6.009E+01	1.631E+00	0.	0.	1.970E+00	-8.698E+00	3.245E+02	1.564E+01
1.767E+00	-2.199E+01	1.268E+01	9.000E+01	0.	0.	2.954E-02	4.088E+01	1.489E+03	0.	1.172E+06	8.294E+01
-6.593E+00	8.208E+01	-6.192E+01	9.000E+01	6.008E+01	-1.686E+00	0.	0.	4.587E+00	8.698E+00	2.188E+01	3.907E+00
9.000E+01	1.800E+02	9.000E+01	1.179E-14	1.067E+03	0.	-1.899E-09	-1.282E+00	0.	5.272E+01	0.	-0.
3.000E+01	9.000E+01	6.000E+01	2.327E+00	0.	5.918E+03	3.722E+02	-1.179E-14	-2.180E+01	2.035E+01	0.	-0.
1.200E+02	9.000E+01	3.000E+01	6.593E+00	-1.103E+03	0.	-1.489E+03	6.874E+00	0.	0.	0.	0.
CP TIME =	44.717	T =	3.048E-03	DT =	1.000E-03						
0.	0.	0.	1.200E+02	-6.004E+01	2.536E+00	0.	0.	2.953E+00	-8.698E+00	5.062E+02	1.534E+01
1.745E+00	-2.198E+01	1.903E+01	9.000E+01	0.	0.	7.593E-02	6.314E+01	2.314E+03	0.	1.171E+06	8.230E+01
-6.511E+00	8.200E+01	-1.036E+02	9.000E+01	6.003E+01	-2.622E+00	0.	0.	4.589E+00	8.497E+00	2.148E+01	4.789E+00
9.000E+01	1.800E+02	9.000E+01	1.166E-14	1.658E+03	0.	-2.952E-09	-1.244E+00	0.	5.173E+01	0.	-0.
3.000E+01	9.000E+01	6.000E+01	2.323E+00	0.	9.141E+03	5.785E+02	-1.166E-14	-2.148E+01	2.034E+01	0.	-0.
1.200E+02	9.000E+01	3.000E+01	6.511E+00	-1.714E+03	0.	-2.314E+03	6.800E+00	0.	0.	0.	0.
CP TIME =	44.822	T =	4.048E-03	DT =	1.000E-03						
0.	0.	0.	1.200E+02	-5.997E+01	3.413E+00	0.	0.	3.933E+00	-8.699E+00	6.777E+02	1.500E+01
1.723E+00	-2.195E+01	2.342E+01	9.000E+01	0.	0.	1.425E-01	8.442E+01	3.112E+03	0.	1.171E+06	8.163E+01
-6.429E+00	8.190E+01	-1.444E+02	9.000E+01	5.994E+01	-3.529E+00	0.	0.	4.592E+00	8.696E+00	2.103E+01	5.297E+00
9.000E+01	1.800E+02	9.000E+01	1.154E-14	2.230E+03	0.	-3.971E-09	-1.207E+00	0.	5.075E+01	0.	-0.
3.001E+01	9.000E+01	5.999E+01	2.320E+00	0.	1.222E+04	7.781E+02	-1.154E-14	-2.103E+01	2.033E+01	0.	-0.
1.200E+02	9.000E+01	3.001E+01	6.429E+00	-2.306E+03	0.	-3.112E+03	6.728E+00	0.	0.	0.	0.

Figure D-17

APPENDIX D

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

Print Out of Input Data

(Example 2)

```
                         INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
                        MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1
                       MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION

                                        INPUT DATA

CASE 5B
INFLATABLE TORUS BASELINE DESIGN
OUTPUT OPTION 1 AND CHANGE OF VARIABLE PRINT

DATA SUBSCRIPT      MMIC           NCUT          IERRRT
     13           0.          1.0000000E+00 0.          -0.          -0.          -0.

PRINT POSITION     1 WILL CONTAIN THE VALUE FROM COMINT(  386) WITH THE LABEL XF
PRINT POSITION    13 WILL CONTAIN THE VALUE FROM COMINT(  387) WITH THE LABEL YF
PRINT POSITION    25 WILL CONTAIN THE VALUE FROM COMINT(  388) WITH THE LABEL ZF
PRINT POSITION     2 WILL CONTAIN THE VALUE FROM COMINT(  270) WITH THE LABEL XDF
PRINT POSITION    14 WILL CONTAIN THE VALUE FROM COMINT(  277) WITH THE LABEL YDF
PRINT POSITION    26 WILL CONTAIN THE VALUE FROM COMINT(  284) WITH THE LABEL ZDF
PRINT POSITION     3 WILL CONTAIN THE VALUE FROM COMINT(  582) WITH THE LABEL XDDF
PRINT POSITION    15 WILL CONTAIN THE VALUE FROM COMINT(  583) WITH THE LABEL YDDF
PRINT POSITION    27 WILL CONTAIN THE VALUE FROM COMINT(  584) WITH THE LABEL ZDDF
EOD            END OF DATA LIST


 VEHICLE WEIGHT, ON EARTH =  6.6014778E+02, ON PLANET =  2.5259418E+02

                 MOMENTS AND PRODUCTS OF INERTIA

        XXI               XYI              XXID              XYID
        YYI               XZI              YYID              XZID
        ZZI               YZI              ZZID              YZID

  2.2562788E+02  0.                 0.               0.
  1.4476933E+02  0.                 0.               0.
  1.4476933E+02  0.                 0.               0.
```

Figure D-18

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

Typical Time History Output Page
Print Option 1
(Example 2)

LOCAL SURFACE COORDINATE SYSTEM					LANDER COORDINATE SYSTEM						
XF YF ZF	XDF YDF ZDF	XDDF YDDF ZDDF	ANGLD1 * ANGLD2 * ANGLD3 *	XD YD ZD	XN YN ZN	PHID THETAD PSID	PHIDD * THETADD * PSIDD *	STROKE FORCE PRESS	XAG YAG ZAG	AF VOL RSXCF	DISTX DISTY FRIRA
CP TIME =	55.723 T	= 0.	DT	= 1.000E-03							
-0.	0.	0.	0.	-6.010E+01	0.	0.	0.	-4.705E-02	-8.698E+00	0.	0.
-0.	-2.274E-13	-2.842E-14	0.	0.	0.	-0.	0.	0.	0.	0.	0.
-7.000E+00	8.500E+01	1.230E+01	0.	6.010E+01	0.	0.	0.	4.585E+00	8.698E+00	0.	0.
CP TIME =	55.961 T	= 4.775E-05 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-6.010E+01	0.	0.	0.	-3.241E-06	-8.698E+00	1.498E+02	1.599E+01
-1.086E-17	-2.274E-13	0.	9.000E+01	0.	0.	0.	0.	0.	0.	1.172E+06	8.364E+01
-6.996E+00	8.500E+01	1.230E+01	9.000E+01	6.010E+01	0.	0.	0.	4.585E+00	8.698E+00	2.197E+01	2.439E+00
CP TIME =	56.065 T	= 1.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-6.011E+01	7.886E-01	0.	0.	9.853E-01	-8.698E+00	1.571E+02	1.597E+01
-4.588E-08	-1.376E-04	-9.944E-01	9.000E+01	0.	0.	4.481E-03	1.920E+01	7.204E-02	0.	1.172E+06	8.361E+01
-6.911E+00	8.500E+01	-2.416E+01	9.000E+01	6.011E+01	-8.156E-01	0.	0.	4.585E+00	8.698E+00	2.197E+01	2.557E+00
CP TIME =	56.184 T	= 2.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-6.009E+01	1.631E+00	0.	0.	1.970E+00	-8.698E+00	3.255E+02	1.566E+01
-5.438E-07	-9.116E-04	-3.778E+00	9.000E+01	0.	0.	2.954E-02	4.088E+01	1.489E+03	0.	1.172E+06	8.294E+01
-6.826E+00	8.497E+01	-6.309E+01	9.000E+01	6.008E+01	-1.686E+00	0.	0.	4.587E+00	8.698E+00	2.180E+01	3.907E+00
CP TIME =	56.310 T	= 3.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-6.004E+01	2.536E+00	0.	0.	2.953E+00	-8.698E+00	5.042E+02	1.534E+01
-2.148E-06	-2.355E-03	-8.419E+00	9.000E+01	0.	0.	7.593E-02	6.314E+01	2.314E+03	0.	1.171E+06	8.230E+01
-6.741E+00	8.490E+01	-1.050E+02	9.000E+01	6.003E+01	-2.622E+00	0.	0.	4.589E+00	8.697E+00	2.148E+01	4.789E+00
CP TIME =	56.387 T	= 4.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-5.997E+01	3.413E+00	0.	0.	3.933E+00	-8.699E+00	6.777E+02	1.300E+01
-5.581E-06	-4.687E-03	-1.474E+01	9.000E+01	0.	0.	1.425E-01	8.442E+01	3.112E+03	0.	1.171E+06	8.163E+01
-6.656E+00	8.479E+01	-1.455E+02	9.000E+01	5.994E+01	-3.529E+00	0.	0.	4.592E+00	8.696E+00	2.103E+01	5.297E+00
CP TIME =	56.463 T	= 5.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-5.987E+01	4.385E+00	0.	0.	4.910E+00	-8.701E+00	8.692E+02	1.471E+01
-1.172E-05	-7.637E-03	-2.288E+01	9.000E+01	0.	0.	2.301E-01	1.077E+02	3.995E+03	0.	1.170E+06	8.101E+01
-6.571E+00	8.463E+01	-1.904E+02	9.000E+01	5.982E+01	-4.532E+00	0.	0.	4.596E+00	8.695E+00	2.045E+01	6.187E+00
CP TIME =	56.543 T	= 6.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.200E+02	-5.975E+01	5.325E+00	0.	0.	5.882E+00	-8.703E+00	1.053E+03	1.440E+01
-2.117E-05	-1.148E-02	-3.297E+01	9.000E+01	0.	0.	3.415E-01	1.299E+02	4.847E+03	0.	1.169E+06	8.036E+01
-6.487E+00	8.444E+01	-2.338E+02	9.000E+01	5.966E+01	-5.501E+00	0.	0.	4.602E+00	8.692E+00	1.973E+01	6.878E+00
CP TIME =	56.623 T	= 7.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.201E+02	-5.961E+01	6.281E+00	0.	0.	6.848E+00	-8.707E+00	1.239E+03	1.400E+01
-3.486E-05	-1.595E-02	-4.484E+01	9.000E+01	0.	0.	4.746E-01	1.521E+02	5.711E+03	0.	1.168E+06	7.971E+01
-6.403E+00	8.420E+01	-2.778E+02	9.000E+01	5.946E+01	-6.482E+00	0.	0.	4.608E+00	8.689E+00	1.887E+01	7.465E+00
CP TIME =	56.703 T	= 8.048E-03 DT	= 1.000E-03								
0.	0.	0.	1.201E+02	-5.944E+01	7.257E+00	0.	0.	7.807E+00	-8.712E+00	1.428E+03	1.380E+01
-5.336E-05	-2.124E-02	-5.851E+01	9.000E+01	0.	0.	6.301E-01	1.746E+02	6.590E+03	0.	1.167E+06	7.908E+01
-6.319E+00	8.391E+01	-3.227E+02	9.000E+01	5.922E+01	-7.481E+00	0.	0.	4.615E+00	8.684E+00	1.788E+01	7.971E+00

Figure D-19

D.4 PROGRAM DESCRIPTION

D.4.1 SUBROUTINES - To facilitate improvements and modifications to the Inflatable Torus Landing Loads and Motions Program, the program is comprised of several subroutines related through a main driving program. The main program reads and prints the input data, sets up and initializes the equations of motion, and calls the subroutines in the required order.

A listing of the subroutines used in the program is given in Figure D-20. Included in this figure are the subroutine names and a description of the operations performed by the subroutine. Several subroutines have multiple entry points. The purpose of each entry into the subroutine and the name associated with the entry point are given in Figure D-20.

The two subroutines ENVIR and AERO are not used in the current program make up. They are skeleton routines which have been provided for the possible inclusion of wind and aerodynamic effects. These two subroutines are not shown on the program flow chart, Section D.4.2.

The subroutine PCCUT performs the numerical integration of the dynamic equation of motions. Three integration methods are available in this subroutine for solving the equations of motion. These consist of a fourth order Runge-Kutta procedure and a variable or fixed step Adams-Moulton Predictor-Corrector technique. The input variables IVARH and IMTH govern the routine used during a particular run. These two quantities are defined in Figure D-10. All of these integration procedures employ the Runge-Kutta routine to initialize the numerical routine.

SUBROUTINES IN THE INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

SUBROUTINE NAME	SUBROUTINE ENTRY POINT	SUBROUTINE OPERATIONS
ENVIR	ENVIR	Determines wind loads acting on lander. Currently this routine is not used.
AERO	AERO	Determines aerodynamic forces acting on lander. Currently this routine is not used.
PCCUT	–	Numerical Integration Subroutine
	LOC	Stores the name, cutoff value, and direction indicator of each cutoff variable for use in the cutoff routine.
	INUPD	Sets up the parameter list for storage of integrated variables in the common array COMINT.
	SETUP	Initialization of subroutine for requested type of integration.
	INTEG	Performs requested numerical integration method.
	CUT	Monitors the cutoff variables for possible program termination.
	UPDAT	Updates the integrated variable list and integration interval at the end of an integration step.
MASS	MASS	Determines mass and inertia properties of lander.
	MASS 1	Determines torus effective mass for a flat landing.
	MASS 2	Determines torus effective mass for an oblique landing.
LOAD	LOAD	Determines the magnitude and location of the normal force due to compressing the torus inflation gas.
PHYS	PHYS	This entry calls MASS for initial inertia calculations.
	PHYS 1	Determines the friction forces and torques acting on the lander. Relates these plus the normal force to the forces and moments acting at lander C.G.
SOLVE	SOLVE	Solution of simultaneous equations by matrix inversion.
PRINT	PRINT	Sets up print routine for optional output.
	PRINT 1	Initializes the print routime for Option 1 or Option 2.
	PRINT 2	Prints time history variables at specified integration times.

Figure D-20

The procedures PHYS, LOAD, and MASS are related to the specific geometry of an impacting inflatable torus. These subroutines obtain the physical quantities required by the integration routine to solve the dynamic equations. These quantities include the normal force, friction forces and torques, forces and moments at lander center of gravity, and the lander's inertia properties.

The subroutine SOLVE is used to solve a set of simultaneous equations through a matrix inversion procedure. The subroutine PRINT outputs the time history variables at specified times during the integration procedure.

Information is transferred between the main program and the subroutines through the common array COMINT. A table defining all the quantities stored in COMINT is given in Appendix E. This information is useful when using the optional output routine discussed in Section D.3.1.

D.4.2 FLOW DIAGRAM - A flow chart indicating the program's general operation is given in Figure D-21. This diagram is not intended to be a comprehensive programming chart. Rather it shows the general flow of the program logic and indicates the order of operations within the subroutines.

APPENDIX D

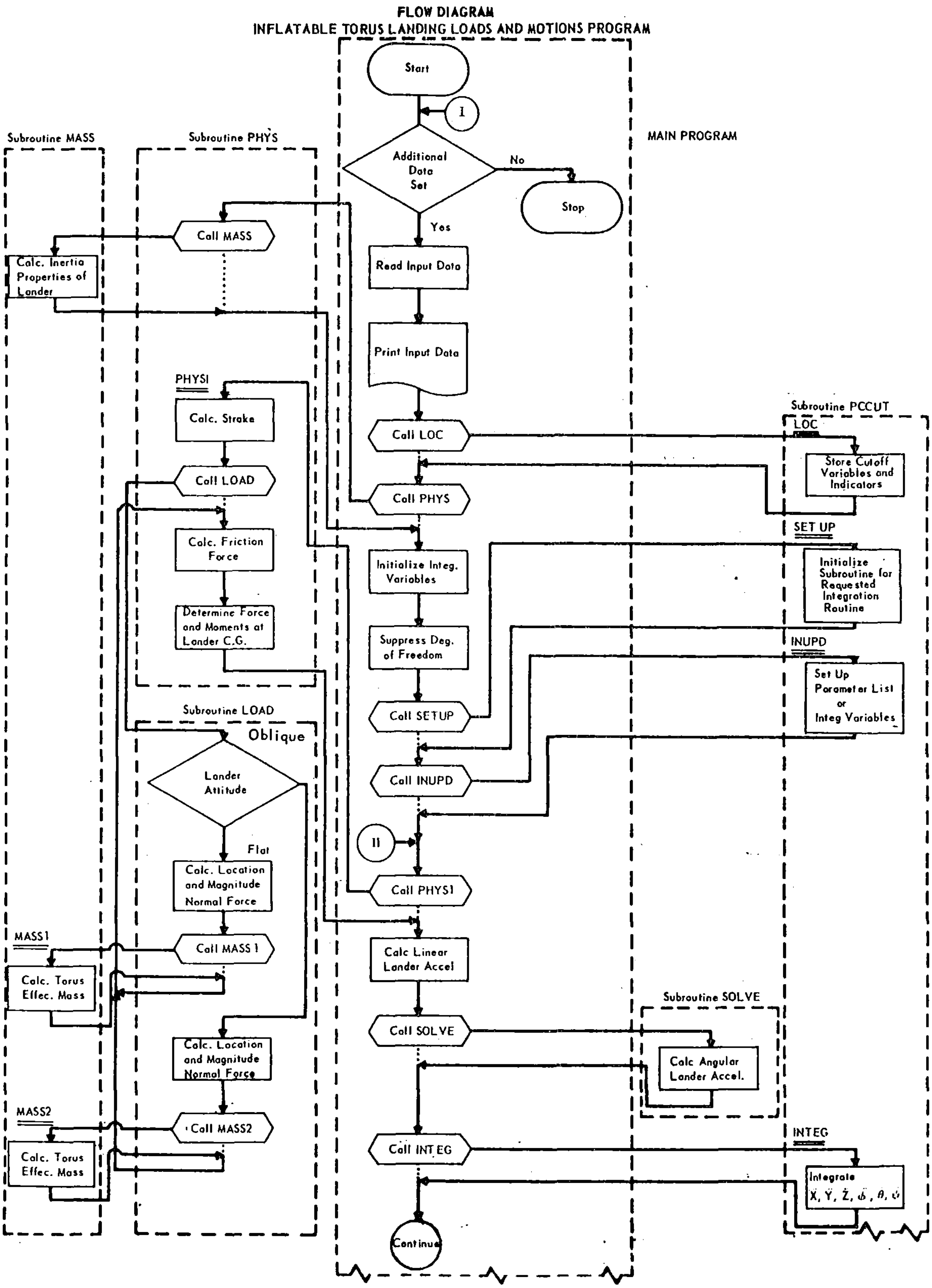


Figure D-21

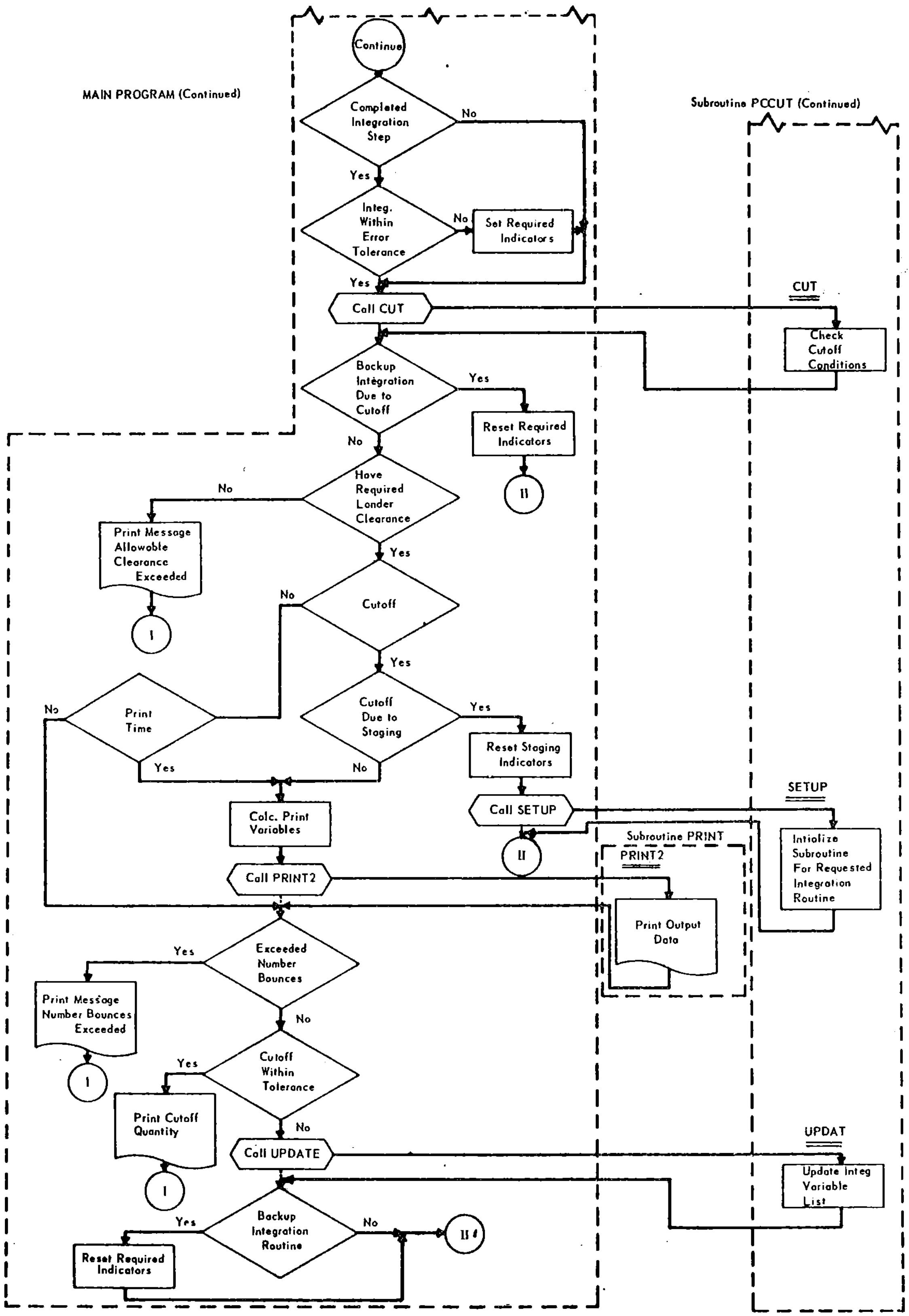


Figure D-21 (Continued)

D.4.3 PROGRAM LISTING - A complete listing of the Inflatable Torus Landing Loads and Motions Program is given on the following pages. This listing is shown in the following order: the main diriving program is presented first, followed by the eight program subroutines. These routines are written in Fortran 2.0 language and were designed for machine computation on the CDC 6400/6600 Computer.

APPENDIX D

INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM

```
      PROGRAM MAIN   (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM
C     MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1
C     MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION
      DIMENSION XCF(3)
      DIMENSION TFC(3)
      DIMENSION NDATA(300)
         DIMENSION XPF(3), XCD(3), FS(3)
      DIMENSION  ANG(3,3), XLPF(3)
      DIMENSION XS(8),IND(8),XDD(7,6),XD(7,19),X(7,19)
      DIMENSION STORE(3), GACC(3), AA(3,3), ASV(3,4), STOR(3,3), NSAV(6)
     1          ,  DC(19) ,       TEMP(3,3) , SUMA(3)
      DIMENSION XVF(3), TR(3,3), TRL(3,3), FF(19),DATA(300)
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT(  1 ), T        )
      EQUIVALENCE ( COMINT(  2 ), HMAX     )
      EQUIVALENCE ( COMINT(  3 ), EMIN     )
      EQUIVALENCE ( COMINT(  4 ), EMAX     )
      EQUIVALENCE ( COMINT(  5 ), HZ       )
      EQUIVALENCE ( COMINT(  6 ), IP       )
      EQUIVALENCE ( COMINT(  7 ), IVARH    )
      EQUIVALENCE ( COMINT(  8 ), IMTH     )
      EQUIVALENCE ( COMINT(  9 ), IPRNT    )
      EQUIVALENCE ( COMINT( 10 ), IFIN     )
      EQUIVALENCE ( COMINT( 11 ), IVAL     )
      EQUIVALENCE ( COMINT( 12 ), IPTOTL   )
      EQUIVALENCE ( COMINT( 13 ), IPTATL   )
      EQUIVALENCE ( COMINT(  14), XS       )
      EQUIVALENCE ( COMINT(  23), IND      )
      EQUIVALENCE ( COMINT( 32 ), X        )
      EQUIVALENCE ( COMINT(165 ), XD       )
      EQUIVALENCE ( COMINT(298 ), XDD      )
      EQUIVALENCE ( COMINT( 340), TOTSV    )
      EQUIVALENCE ( COMINT(341), PSIL    )
      EQUIVALENCE ( COMINT( 342), TLSV     )
      EQUIVALENCE ( COMINT( 344), ZLS      )
      EQUIVALENCE ( COMINT( 345), PHIL     )
      EQUIVALENCE ( COMINT( 346), THETAL   )
      EQUIVALENCE ( COMINT( 347), HMIN     )
      EQUIVALENCE ( COMINT( 348), STROKE   )
      EQUIVALENCE ( COMINT( 349), SAFS     )
      EQUIVALENCE ( COMINT( 349), SPELST   )
      EQUIVALENCE ( COMINT( 356), SSSAVE   )
      EQUIVALENCE ( COMINT( 356), SPSTTM   )
      EQUIVALENCE ( COMINT( 363), SFSAVL   )
      EQUIVALENCE ( COMINT( 363), SAVST2   )
      EQUIVALENCE ( COMINT( 370), SVSAVE   )
      EQUIVALENCE ( COMINT( 370), SAVST1   )
      EQUIVALENCE ( COMINT( 377), SKODUP   )
      EQUIVALENCE ( COMINT( 384), CUTERR   )
      EQUIVALENCE ( COMINT( 385), JCUT     )
C         END OF VARIABLE ASSIGNMENT NEEDED FOR PCCUT
C               NOTICE XS AND IND ARE DIMENSIONIAL ARRAYS OF LENGTH 9
C               NOTICE ALL INTEGRATION AND TIME HISTORY VARIABLES MUST
C                     HAVE 7 COMMON LOCATIONS ALLOTED
      EQUIVALENCE ( COMINT( 386), XF       )
```

```
      EQUIVALENCE ( COMINT( 387), YF      )                          MAN 570
      EQUIVALENCE ( COMINT( 388), ZF      )                          MAN 580
      EQUIVALENCE ( COMINT( 389), XW      )                          MAN 590
      EQUIVALENCE ( COMINT( 390), YW      )                          MAN 600
      EQUIVALENCE ( COMINT( 391), ZW      )                          MAN 610
      EQUIVALENCE ( COMINT( 392), SCFWT   )                          MAN 620
      EQUIVALENCE ( COMINT( 393), GZ      )                          MAN 630
      EQUIVALENCE ( COMINT( 394), XAG     )                          MAN 640
      EQUIVALENCE ( COMINT( 395), YAG     )                          MAN 650
      EQUIVALENCE ( COMINT( 396), ZAG     )                          MAN 660
      EQUIVALENCE ( COMINT( 397), DENS    )                          MAN 670
      EQUIVALENCE ( COMINT( 398), XWND    )                          MAN 680
      EQUIVALENCE ( COMINT( 399), YWND    )                          MAN 690
      EQUIVALENCE ( COMINT( 400), ZWND    )                          MAN 700
      EQUIVALENCE ( COMINT( 401), ALPHA   )                          MAN 710
      EQUIVALENCE ( COMINT( 402), GDOT    )                          MAN 720
      EQUIVALENCE ( COMINT( 403), SDOT    )                          MAN 730
      EQUIVALENCE ( COMINT( 404), BETA    )                          MAN 740
      EQUIVALENCE ( COMINT( 405), ALT     )                          MAN 750
      EQUIVALENCE ( COMINT( 406), RANGE   )                          MAN 760
      EQUIVALENCE ( COMINT( 407), THETA   )                          MAN 770
      EQUIVALENCE ( COMINT( 408), PSI     )                          MAN 780
      EQUIVALENCE ( COMINT( 409), PHI     )                          MAN 790
      EQUIVALENCE ( COMINT( 410), XXI     )                          MAN 800
      EQUIVALENCE ( COMINT( 411), YYI     )                          MAN 810
      EQUIVALENCE ( COMINT( 412), ZZI     )                          MAN 820
      EQUIVALENCE ( COMINT( 413), XN      )                          MAN 830
      EQUIVALENCE ( COMINT( 414), YN      )                          MAN 840
      EQUIVALENCE ( COMINT( 415), ZN      )                          MAN 850
      EQUIVALENCE ( COMINT( 416), XMASST  )                          MAN 860
      EQUIVALENCE ( COMINT( 417), THRST   )                          MAN 870
      EQUIVALENCE ( COMINT( 418), XYI     )                          MAN 880
      EQUIVALENCE ( COMINT( 419), XZI     )                          MAN 890
      EQUIVALENCE ( COMINT( 420), YZI     )                          MAN 900
      EQUIVALENCE ( COMINT( 421), DELTA   )                          MAN 910
      EQUIVALENCE ( COMINT( 422), XXID    )                          MAN 920
      EQUIVALENCE ( COMINT( 423), YYID    )                          MAN 930
      EQUIVALENCE ( COMINT( 424), ZZID    )                          MAN 940
      EQUIVALENCE ( COMINT( 425), XYID    )                          MAN 950
      EQUIVALENCE ( COMINT( 426), XZID    )                          MAN 960
      EQUIVALENCE ( COMINT( 427), YZID    )                          MAN 970
      EQUIVALENCE ( COMINT( 428), XLS     )                          MAN 980
      EQUIVALENCE ( COMINT( 429), YLS     )                          MAN 990
      EQUIVALENCE ( COMINT( 430), XDLS    )                          MAN1000
      EQUIVALENCE ( COMINT( 431), YDLS    )                          MAN1010
      EQUIVALENCE ( COMINT( 432), ZDLS    )                          MAN1020
      EQUIVALENCE ( COMINT( 433), XDDLS   )                          MAN1030
      EQUIVALENCE ( COMINT( 434), YDDLS   )                          MAN1040
      EQUIVALENCE ( COMINT( 435), ZDDLS   )                          MAN1050
      EQUIVALENCE ( COMINT( 436), XLGRAV  )                          MAN1060
      EQUIVALENCE ( COMINT( 437), YLGRAV  )                          MAN1070
      EQUIVALENCE ( COMINT( 438), ZLGRAV  )                          MAN1080
      EQUIVALENCE ( COMINT( 439), NTX     )                          MAN1090
      EQUIVALENCE ( COMINT( 440), NTY     )                          MAN1100
      EQUIVALENCE ( COMINT( 441), NTZ     )                          MAN1110
      EQUIVALENCE ( COMINT( 442), NRX     )                          MAN1120
```

```
      EQUIVALENCE ( COMINT( 443), NRY    )                              MAN1130
      EQUIVALENCE ( COMINT( 444), NRZ    )                              MAN1140
      EQUIVALENCE ( COMINT( 445), A      )                              MAN1150
      EQUIVALENCE ( COMINT( 446), B      )                              MAN1160
      EQUIVALENCE ( COMINT( 447), C      )                              MAN1170
      EQUIVALENCE ( COMINT( 448), D      )                              MAN1180
      EQUIVALENCE ( COMINT( 449), E      )                              MAN1190
      EQUIVALENCE ( COMINT( 450), F      )                              MAN1200
      EQUIVALENCE ( COMINT( 451), G      )                              MAN1210
      EQUIVALENCE ( COMINT( 452), RHOL   )                              MAN1220
      EQUIVALENCE ( COMINT( 453), RHOP   )                              MAN1230
      EQUIVALENCE ( COMINT( 454), TOTGV  )                              MAN1240
      EQUIVALENCE ( COMINT( 455), FONTT )                               MAN1250
      EQUIVALENCE ( COMINT( 455), FORCE  )                              MAN1260
      EQUIVALENCE ( COMINT( 456), TORTN  )                              MAN1270
      EQUIVALENCE ( COMINT( 457), ARTSU  )                              MAN1280
      EQUIVALENCE ( COMINT( 458), PSTTM  )                              MAN1290
      EQUIVALENCE ( COMINT( 459), ANGLE  )                              MAN1300
      EQUIVALENCE ( COMINT( 460), ANGLE1 )                              MAN1310
      EQUIVALENCE ( COMINT( 461), ANGLE2 )                              MAN1320
      EQUIVALENCE ( COMINT( 462), ANGLE3 )                              MAN1330
      EQUIVALENCE ( COMINT( 463), ARMOM  )                              MAN1340
      EQUIVALENCE ( COMINT( 464), ELST   )                              MAN1350
      EQUIVALENCE ( COMINT( 465), RDIA   )                              MAN1360
      EQUIVALENCE ( COMINT( 466), AMU    )                              MAN1370
      EQUIVALENCE ( COMINT( 467), XKW    )                              MAN1380
      EQUIVALENCE ( COMINT( 468), DX     )                              MAN1390
      EQUIVALENCE ( COMINT( 469), DTHE   )                              MAN1400
      EQUIVALENCE ( COMINT( 470), ACLR   )                              MAN1410
      EQUIVALENCE ( COMINT( 471), RNORM  )                              MAN1420
      EQUIVALENCE ( COMINT( 472), LDSTR1 )                              MAN1430
      EQUIVALENCE ( COMINT(473),IERPRT)                                 MAN1440
      EQUIVALENCE ( COMINT( 474), RSXCF  )                              MAN1450
      EQUIVALENCE ( COMINT( 475), PELST  )                              MAN1460
      EQUIVALENCE ( COMINT( 476), SLOPE  )                              MAN1470
      EQUIVALLNCE ( COMINT( 477), VOL    )                              MAN1480
      EQUIVALENCE ( COMINT( 478), R1     )                              MAN1490
      EQUIVALENCE ( COMINT( 479), R2     )                              MAN1500
      EQUIVALENCE ( COMINT( 480), R3     )                              MAN1510
      EQUIVALENCE ( COMINT( 481), THETAT )                              MAN1520
      EQUIVALENCE ( COMINT( 482), RS     )                              MAN1530
      EQUIVALENCE ( COMINT( 483), RL     )                              MAN1540
      EQUIVALENCE ( COMINT( 484), FABWT  )                              MAN1550
      EQUIVALENCE ( COMINT( 485), PLY1   )                              MAN1560
      EQUIVALENCE ( COMINT( 486), PLY2   )                              MAN1570
      EQUIVALENCE ( COMINT( 487), PLY3   )                              MAN1580
      EQUIVALENCE ( COMINT( 488), AF     )                              MAN1590
      EQUIVALENCE ( COMINT( 489), HYST   )                              MAN1600
      EQUIVALENCE ( COMINT( 490), PI     )                              MAN1610
      EQUIVALENCE ( COMINT( 491), PA     )                              MAN1620
      EQUIVALENCE ( COMINT( 492), FRIKA  )                              MAN1630
      EQUIVALENCE ( COMINT( 493), DISTX  )                              MAN1640
      EQUIVALENCE ( COMINT( 494), DISTY  )                              MAN1650
      EQUIVALENCE ( COMINT( 495), PRESS  )                              MAN1660
      EQUIVALENCE ( COMINT( 496), AFS    )                              MAN1670
      EQUIVALENCE ( COMINT( 497), SSAVL  )                              MAN1680
```

```
      EQUIVALENCE ( COMINT( 498), FSAVE  )                              MAN1690
      EQUIVALENCE ( COMINT( 499), VSAVE  )                              MAN1700
      EQUIVALENCE ( COMINT( 500), KODEUP )                              MAN1710
      EQUIVALENCE ( COMINT( 501), GASCNT )                              MAN1720
      EQUIVALENCE ( COMINT( 502), IFLAT  )                              MAN1730
      EQUIVALENCE ( COMINT( 503), XVF    )                              MAN1740
C         DIMENSIONED XVF(3)                                            MAN1750
      EQUIVALENCE ( COMINT( 506), TR     )                              MAN1760
C         DIMENSIONED TR(3,3)                                           MAN1770
      EQUIVALENCE ( COMINT( 515), TRL    )                              MAN1780
C         DIMENSIONED TRL(3,3)                                          MAN1790
      EQUIVALENCE ( COMINT( 524), FF     )                              MAN1800
C         DIMENSIONED FF(19)                                            MAN1810
      EQUIVALENCE ( COMINT( 543), XPF    )                              MAN1820
C         DIMENSIONED XPF(3)                                            MAN1830
      EQUIVALENCE ( COMINT( 546), XCD    )                              MAN1840
C         DIMENSIONED XCD(3)                                            MAN1850
      EQUIVALENCE ( COMINT( 549), FS     )                              MAN1860
C         DIMENSION FS(3)                                               MAN1870
      EQUIVALENCE ( COMINT( 552), Q      )                              MAN1880
      EQUIVALENCE ( COMINT( 553), XMACH  )                              MAN1890
      EQUIVALENCE ( COMINT( 554), SOUND  )                              MAN1900
      EQUIVALENCE ( COMINT( 555), FIFLAT )                              MAN1910
      EQUIVALENCE ( COMINT( 556), MMIC   )                              MAN1920
      EQUIVALENCE ( COMINT( 557), SIGMA  )                              MAN1930
      EQUIVALENCE ( COMINT( 558), GAMA   )                              MAN1940
      EQUIVALENCE ( COMINT( 559), SAVSTB )                              MAN1950
      EQUIVALENCE ( COMINT( 560), SAVSTA )                              MAN1960
      EQUIVALENCE ( COMINT( 561), DEGRAD )                              MAN1970
      EQUIVALENCE ( COMINT( 562), ANGLD1 )                              MAN1980
      EQUIVALENCE ( COMINT( 563), ANGLD2 )                              MAN1990
      EQUIVALENCE ( COMINT( 564), ANGLD3 )                              MAN2000
      EQUIVALENCE ( COMINT( 565), PLTMAS )                              MAN2010
      EQUIVALENCE ( COMINT( 566), PLTRAD )                              MAN2020
      EQUIVALENCE ( COMINT( 567), NMBNCS )                              MAN2030
      EQUIVALENCE ( COMINT( 568), KOUNT2 )                              MAN2040
      EQUIVALENCE ( COMINT( 569), XLPF   )                              MAN2050
C         DIMENSIONED XLPF(3)                                           MAN2060
      EQUIVALENCE ( COMINT( 572), ANG    )                              MAN2070
C         DIMENSIONED ANG(3,3)                                          MAN2080
      EQUIVALENCE ( COMINT( 581), VMIN   )                              MAN2090
      EQUIVALENCE ( COMINT( 582), GACC   )                              MAN2100
C         DIMENSIONED GACC(3)                                           MAN2110
      EQUIVALENCE ( COMINT( 585), STREF  )                              MAN2120
      EQUIVALENCE ( COMINT( 586), TFC    )                              MAN2130
C         DIMENSIONED TFC(3)                                            MAN2140
      EQUIVALENCE ( COMINT( 589), RAV   )                               MAN2150
      EQUIVALENCE ( COMINT( 590), FRICT  )                              MAN2160
      EQUIVALENCE ( COMINT( 591), XCF    )                              MAN2170
C         DIMENSIONED XCF(3)                                            MAN2180
      EQUIVALENCE ( COMINT( 594), POW    )                              MAN2190
      EQUIVALENCE ( COMINT( 595), RSHCR  )                              MAN2200
      DATA EOD / 10HEOD         /                                       MAN2210
      DATA  DATA   / 300*0.0 /                                          MAN2220
      DATA (NDATA(I),I=1,90) /                                          MAN2230
     1 6HIP    , 6HIVARH , 6HIMTH  , 6HEMAX  , 6HEMIN  , 6HHMIN  ,      MAN2240
```

```
      2 6HNTX    , 6HNTY    , 6HNTZ    , 6HNRX    , 6HNRY    , 6HNRZ    ,    MAN2250
      3 6HMMIC   , 6HNCUT   , 6HIERPRT , 6H       , 6H       , 6H       ,    MAN2260
      4 6HT      , 6HT      , 6H       , 6H       , 6H       , 6H       ,    MAN2270
      5 6HTOTSV  , 6HTOTSV  , 6H       , 6H       , 6H       , 6H       ,    MAN2280
      6 6HXD(,19, 6HXD(,19, 6H       , 6H       , 6H       , 6H       ,    MAN2290
      7 6HTLSV   , 6HTLSV   , 6H       , 6H       , 6H       , 6H       ,    MAN2300
      8 6HZLS    , 6HZLS    , 6H       , 6H       , 6H       , 6H       ,    MAN2310
      9 6HPHIL   , 6HPHIL   , 6H       , 6H       , 6H       , 6H       ,    MAN2320
      $ 6HTHETAL , 6HTHETAL , 6H       , 6H       , 6H       , 6H       ,    MAN2330
      $ 6HPSIL   , 6HPSIL   , 6H       , 6H       , 6H       , 6H       ,    MAN2340
      $ 6HT      , 6HHMAX   , 6HKOUNTZ , 6HNMBNCS , 6H       , 6H       ,    MAN2350
      $ 6HXF     , 6HXVF(1) , 6HYF     , 6HXVF(2) , 6HZF     , 6HXVF(3) ,    MAN2360
      $ 6HPHI    , 6HPHID   , 6HTHETA  , 6HTHETAD , 6HPSI    , 6HPSID   ,    MAN2370
      $ 6HRANGL  , 6HSLOPE  , 6H       , 6H       , 6H       , 6H       /    MAN2380
       DATA (NDATA(I),I=91,132) /                                           MAN2390
      1 6H       , 6H       , 6H       , 6H       , 6H       , 6H       ,    MAN2400
      2 6H       , 6H       , 6HSCFWT  , 6HA      , 6H B     , 6HC      ,    MAN2410
      3 6HD      , 6HL      , 6HF      , 6HG      , 6HRHOL   , 6HRHOP   ,    MAN2420
      4 6HPOW    , 6HRSHCR  , 6HVMIN   , 6HKDIA   , 6HDX     , 6HDTHL   ,    MAN2430
      5 6HRNORM  , 6HACLR   , 6HLLST   , 6HLDSTR1 , 6HAMU    , 6HXKW    ,    MAN2440
      6 6HFABWT  , 6HPLY1   , 6HPLY2   , 6HPLY3   , 6HRL     , 6HRS     ,    MAN2450
      7 6HHYST   , 6HPI     , 6HPA     , 6HPLTMAS , 6HPLTRAD , 6HGASCNT /    MAN2460
C      ALL DATA READ AND MANUPULATION PLUS INITIALIZATION AND DEFINED       MAN2470
C      INITIAL COND. HERE                                                   MAN2480
       DO 10 I=1,8                                                          MAN2490
10     NSAV(I)=0                                                            MAN2500
       KXPN=8                                                               MAN2510
20     WRITE (6,1190)                                                       MAN2520
       DO 30 I=1,600                                                        MAN2530
30     COMINT(I)=0.0                                                        MAN2540
       NN=0                                                                 MAN2550
       IF (KXPN) 40,70,40                                                   MAN2560
40     DO 60 I=1,KXPN                                                       MAN2570
       IF (NSAV(I)) 50,60,50                                                MAN2580
50     NN=NN+1                                                              MAN2590
60     CONTINUE                                                             MAN2600
70     NGO=0                                                                MAN2610
       DO 90 I=1,3                                                          MAN2620
C            LEAVE COLUMN ONE BLANK                                         MAN2630
       READ (5,1200)                                                        MAN2640
       IF (EOF,5) 80,90                                                     MAN2650
80     WRITE (6,1210)                                                       MAN2660
       STOP                                                                 MAN2670
90     WRITE (6,1200)                                                       MAN2680
100    READ (5,1220) NSIP,NLOC,ZD1,ZD2,ZD3,ZD4,ZD5,ZD6                      MAN2690
       NLOC6=NLOC+5                                                         MAN2700
       WRITE (6,1230) (NDATA(I),I=NLOC,NLOC6)                               MAN2710
       IF ((NLOC-13.LE.0).OR.(NLOC-61.GT.0)) GO TO 120                      MAN2720
       WRITE (6,1240)                                                       MAN2730
       DO 110 I=1,8                                                         MAN2740
       IF (NLOC.EQ.NSAV(I)) GO TO 120                                       MAN2750
110    CONTINUE                                                             MAN2760
       NN=NN+1                                                              MAN2770
       NSAV(NN)=NLOC                                                        MAN2780
120    WRITE (6,1250) NLOC,ZD1,ZD2,ZD3,ZD4,ZD5,ZD6                          MAN2790
       DATA(NLOC)=ZD1                                                       MAN2800
```

```
      DATA(NLOC+1)=ZD2
      DATA(NLOC+2)=ZD3
      DATA(NLOC+3)=ZD4
      DATA(NLOC+4)=ZD5
      DATA(NLOC+5)=ZD6
      IF (NSTP.EQ.0) GO TO 100
      NCUT=DATA(14)+.1
      IF (NCUT.NE.0) GO TO 130
      WRITE (6,1260)
      STOP
C         INPUT  CHANGE OF PRINT VARIABLES
C              ANY VARIABLE IN COMMON CAN BE PRINTED BY CARD OPTION
130   READ (5,1270) VNAME,LCOMON,LPRINT
C         COLUMN ONE OF DATA CARD IS USED FOR SPACING CONTROL
      IF (VNAME.EQ.EOD) GO TO 140
      WRITE (6,1280) LPRINT,LCOMON,VNAME
      CALL PRINT (VNAME,LCOMON,LPRINT)
      GO TO 130
140   WRITE (6,1290) VNAME
      IP=DATA(1)+.1
      HMAX=DATA(68)
      HMIN=DATA(6)
      IF (HMIN.EQ.0.0) HMIN=HMAX*2.**(-16)
      DEGRAD=57.2957795131
      IPTATL=0
      IPTOTL=0
      INDACL=0
      CUTERR=.0001
      NCONST=1
      INDBAC=0
      ACLR=0.0
      LDSTR1=0
      IERPRT=DATA(15)
      NMBNCS=DATA(70)
      IF (NMBNCS.EQ.0) NMBNCS=100
      NMBNC=0
      STREF=DATA(98)
      SCFWT=DATA(99)
      A=DATA(100)
      B=DATA(101)
      C=DATA(102)
      D=DATA(103)
      E=DATA(104)
      F=DATA(105)
      G=DATA(106)
      RHOL=DATA(107)
      RHOP=DATA(108)
      IF (DATA(111).NE.0) GO TO 150
      WRITE (6,1300)
      STOP
150   POW=DATA(109)
      RSHCR=DATA(110)
      VMIN=DATA(111)
      RDIA=DATA(112)
      DX=DATA(113)
      DTHE=DATA(114)
```

```
      IF (DTHE.LE.0.0) DTHE=.03
      RNORM=DATA(115)
      ELST=DATA(117)
      AMU=DATA(119)
      XKW=DATA(120)
C     INFLATABEL DATA ITEMS
      FABWT=DATA(121)
      PLY1=DATA(122)
      PLY2=DATA(123)
      PLY3=DATA(124)
      RL=DATA(125)
      RS=DATA(126)
      HYST=DATA(127)
      PI=DATA(128)
      PA=DATA(129)
      PLTMAS=DATA(130)
      PLTRAD=DATA(131)
      GASCNT=DATA(132)
      IVARH=DATA(2)+.1
      IMTH=DATA(3)+.1
      EMAX=DATA(4)
      EMIN=DATA(5)
      IF (EMAX.EQ.0.0) EMAX=1.E-4
      IF (EMIN.EQ.0.0) EMIN=1.E-6
      NTX=DATA(7)+.1
      NTY=DATA(8)+.1
      NTZ=DATA(9)+.1
      NRX=DATA(10)+.1
      NRY=DATA(11)+.1
      NRZ=DATA(12)+.1
      MMIC=DATA(13)+.1
      T=DATA(67)
      KOUNT2=DATA(69)+.1
      KOUNT1=KOUNT2-1
      XF=DATA(73)
      XVF(1)=DATA(74)
      YF=DATA(75)
      XVF(2)=DATA(76)
      ZF=DATA(77)
      XVF(3)=DATA(78)
      RANGE=DATA(85)
      SLOPE=DATA(86)
      PHI=DATA(79)
      PHIP=PHI
      XD(1,4)=DATA(80)
      THETA=DATA(81)
      XD(1,5)=DATA(82)
      PSI=DATA(83)
      PSIP=PSI
      XD(1,6)=DATA(84)
C         DEFINE CUTOFF VARIABLES BY CALL TO LOC
C               THE FIRST EIGHT CUTOFF VARIABLES ARE DEFINED BY INPUT
C               STROKE IS USED FOR STAGING AT TOUCH DOWN  AND LIFT OFF
      DO 250 J=1,8
      IF (NSAV(J)) 160,250,160
160   LOCA=(NSAV(J)-13)/6
```

```
      GO TO (170,180,190,200,210,220,230,240), LOCA                        MAN3930
170   CALL LOC (1,6HT      ,DATA(19),DATA(20))                              MAN3940
      GO TO 250                                                             MAN3950
180   CALL LOC (340,6HTOTSV ,DATA(25),DATA(26))                             MAN3960
      GO TO 250                                                             MAN3970
190   CALL LOC (291,6HXD(19 ,DATA(31),DATA(32))                             MAN3980
      GO TO 250                                                             MAN3990
200   CALL LOC (342,6HTLSV  ,DATA(37),DATA(38))                             MAN4000
      GO TO 250                                                             MAN4010
210   CALL LOC (344,6HZLS   ,DATA(43),DATA(44))                             MAN4020
      GO TO 250                                                             MAN4030
220   CALL LOC (186,6HPHID  ,DATA(49),DATA(50))                             MAN4040
      GO TO 250                                                             MAN4050
230   CALL LOC (193,6HTHETAD,DATA(55),DATA(56))                             MAN4060
      GO TO 250                                                             MAN4070
240   CALL LOC (200,6HPSID  ,DATA(61),DATA(62))                             MAN4080
250   CONTINUE                                                              MAN4090
      CALL LOC (348,6HSTROKE,0.0,0)                                         MAN4100
      LOCSTR=IPTATL                                                         MAN4110
      KXPN=I                                                                MAN4120
C     IF IV=1CONSTANT INTERVAL INTEGRATION, IV=0 VARIBLE INTERVAL           MAN4130
      GZ=16.137E-11*PLTMAS/(-ZF+PLTRAD)**2                                  MAN4140
      CALL PRINT1                                                           MAN4150
      CALL PHYS                                                             MAN4160
      WTERTH=XMASST*32.147                                                  MAN4170
      WTPLNT=XMASST*GZ                                                      MAN4180
      WRITE (6,1310) WTERTH,WTPLNT                                          MAN4190
      WRITE (6,1320)                                                        MAN4200
      WRITE (6,1330) XXI,XYI,XXID,XYID,YYI,XZI,YYID,XZID,ZZI,YZI,ZZID,YZMAN4210
     1ID                                                                    MAN4220
      CALL AERO                                                             MAN4230
      CALL ENVIR                                                            MAN4240
      X(1,4)=PHI/DEGRAD                                                     MAN4250
      X(1,5)=THETA/DEGRAD                                                   MAN4260
      X(1,6)=PSI/DEGRAD                                                     MAN4270
      CS=COS(X(1,6))                                                        MAN4280
      CT=COS(X(1,5))                                                        MAN4290
      CP=COS(X(1,4))                                                        MAN4300
      SP=SIN(X(1,4))                                                        MAN4310
      ST=SIN(X(1,5))                                                        MAN4320
      SS=SIN(X(1,6))                                                        MAN4330
      CSLP=COS(SLOPE/DEGRAD)                                                MAN4340
      SSLP=SIN(SLOPE/DEGRAD)                                                MAN4350
      IF (ABS(X(1,4)-1.5707963268).GE.1.E-8) GO TO 260                      MAN4360
      CP=0.0                                                                MAN4370
260   IF (ABS(X(1,5)-1.5707963268).GE.1.E-8) GO TO 270                      MAN4380
      CT=0.0                                                                MAN4390
270   IF (ABS(X(1,6)-1.5707963268).GE.1.E-8) GO TO 280                      MAN4400
      CS=0.0                                                                MAN4410
280   CONTINUE                                                              MAN4420
      TR(1,1)=CS*CT                                                         MAN4430
      TR(2,1)=CT*SS                                                         MAN4440
      TR(3,1)=-ST                                                           MAN4450
      TR(1,2)=-SS*CP+CS*ST*SP                                               MAN4460
      TR(2,2)=CS*CP+SS*SP*ST                                                MAN4470
      TR(3,2)=CT*SP                                                         MAN4480
```

```
      TR(1,3)=CS*ST*CP+SS*SP
      TR(2,3)=SS*ST*CP-CS*SP
      TR(3,3)=CT*CP
      TRL(1,1)=TR(1,1)
      TRL(1,2)=TR(1,2)
      TRL(1,3)=TR(1,3)
      TRL(2,1)=TR(2,1)*CSLP-TR(3,1)*SSLP
      TRL(2,2)=TR(2,2)*CSLP-TR(3,2)*SSLP
      TRL(2,3)=TR(2,3)*CSLP-TR(3,3)*SSLP
      TRL(3,1)=TR(2,1)*SSLP+TR(3,1)*CSLP
      TRL(3,2)=TR(2,2)*SSLP+TR(3,2)*CSLP
      TRL(3,3)=TR(2,3)*SSLP+TR(3,3)*CSLP
      XLS=XF
      YLS=YF*CSLP-ZF*SSLP
      ZLS=YF*SSLP+ZF*CSLP
      XDLS=XVF(1)
      YDLS=XVF(2)*CSLP-XVF(3)*SSLP
      ZDLS=XVF(2)*SSLP+XVF(3)*CSLP
      XLGRAV=0.
      YLGRAV=-GZ*SSLP
      ZLGRAV=GZ*CSLP
      DO 300 I=1,3
      X(1,I)=0.
      SUMAA=0.
      DO 290 J=1,3
  290 SUMAA=SUMAA+TR(J,I)*XVF(J)
  300 XD(1,I)=SUMAA
      X(1,7)=TR(1,1)
      X(1,8)=TR(2,1)
      X(1,9)=TR(3,1)
      X(1,10)=TR(1,2)
      X(1,11)=TR(2,2)
      X(1,12)=TR(3,2)
      X(1,13)=TR(1,3)
      X(1,14)=TR(2,3)
      X(1,15)=TR(3,3)
      X(1,16)=XF
      X(1,17)=YF
      X(1,18)=ZF
      X(1,19)=RANGE
      IF (NTX.NE.0) XD(1,1)=0.0
      IF (NTY.NE.0) XD(1,2)=0.0
      IF (NTZ.NE.0) XD(1,3)=0.0
      IF (NRX.NE.0) XD(1,4)=0.0
      IF (NRY.NE.0) XD(1,5)=0.0
      IF (NRZ.NE.0) XD(1,6)=0.0
      ALT=-ZF
      XNUMB=TRL(3,1)
      IF (ABS(XNUMB)-1.) 320,320,310
  310 XNUMB=XNUMB/ABS(XNUMB)
  320 THETAL=-ASIN(XNUMB)*DEGRAD
      IF (ABS(TRL(2,1))+ABS(TRL(1,1))+ABS(SLOPE).NE.0.0) GO TO 330
      PSIL=PSI
      PHIL=PHI
      GO TO 340
  330 PSIL=ATAN2(TRL(2,1),TRL(1,1))*DEGRAD
```

```
      PHIL=ATAN2(TRL(3,2),TRL(3,3))*DEGRAD
340   TOTSV=SQRT(XDLS**2+YDLS**2+ZDLS**2)
      TOTGV=SQRT(XVF(1)**2+XVF(2)**2+XVF(3)**2)
      CON1=TOTGV+1.E-10
      GAMA=ASIN(-XVF(3)/CON1)*DEGRAD
      IF (XVF(2)) 370,350,370
350   IF (XVF(1)) 370,360,370
360   SIGMA=0.
      GO TO 380
370   SIGMA=ATAN2(XVF(2),XVF(1))*DEGRAD
380   CALL SETUP
      NN=6
      DO 400 I=1,6
      NN=NN+1
      IF (DATA(NN)) 400,390,400
390   IRLM=(I-1)*7
      CALL INUPD (165+IRLM)
      CALL INUPD (298+IRLM)
400   CONTINUE
      DO 410 I=7,19
      IRLM=(I-1)*7
      CALL INUPD (32+IRLM)
      CALL INUPD (165+IRLM)
410   CONTINUE
C          PLACE  TIME HISTORY VARIABLES IN INTEGRATION PARAMETER LIST
C                 PURPOSE TO CONTROL VARIABLES WHEN INTEGRATION BACKS UP
      CALL INUPD (349)
      CALL INUPD (356)
      CALL INUPD (363)
      CALL INUPD (370)
      CALL INUPD (377)
C     CALCULATE ALL HIGHEST DERIVATIVES FOR EACH EQUATION
420   DO 430 I=1,19
430   FF(I)=0.0
      GZ=16.137E-11*PLTMAS/(-ZF+PLTRAD)**2
      XAG=X(1,9)*GZ
      YAG=X(1,12)*GZ
      ZAG=X(1,15)*GZ
      CALL PHYS1
      CALL AERO1
      XN=FF(1)/(XMASST*32.147)
      YN=FF(2)/(XMASST*32.147)
      ZN=FF(3)/(XMASST*32.147)
      XDD(1,1)=FF(1)/XMASST+XD(1,2)*XD(1,6)-XD(1,3)*XD(1,5)+XAG
      XDD(1,2)=FF(2)/XMASST-XD(1,1)*XD(1,6)+XD(1,3)*XD(1,4)+YAG
      XDD(1,3)=FF(3)/XMASST+XD(1,1)*XD(1,5)-XD(1,4)*XD(1,2)+ZAG
      GO TO (440,490), NCONST
440   IF (MMIC) 460,450,460
450   NCONST=2
      GO TO 470
460   TEMP(1,1)=XXID
      TEMP(1,2)=-XYID
      TEMP(1,3)=-XZID
      TEMP(2,1)=-XYID
      TEMP(2,2)=YYID
      TEMP(2,3)=-YZID
```

```
      TEMP(3,1)=-XZID
      TEMP(3,2)=-YZID
      TEMP(3,3)=ZZID
470   AA(1,1)=XXI
      AA(1,2)=-XYI
      AA(1,3)=-XZI
      AA(2,1)=-XYI
      AA(2,2)=YYI
      AA(2,3)=-YZI
      AA(3,1)=-XZI
      AA(3,2)=-YZI
      AA(3,3)=ZZI
      DO 480 I=1,3
      DO 480 J=1,3
480   ASV(I,J)=AA(I,J)
      CALL SOLVE (ASV,STOR,SUM1,3,0)
490   XDD(1,4)=ASV(1,1)*FF(4)+ASV(1,2)*FF(5)+ASV(1,3)*FF(6)
      XDD(1,5)=ASV(2,1)*FF(4)+ASV(2,2)*FF(5)+ASV(2,3)*FF(6)
      XDD(1,6)=ASV(3,1)*FF(4)+ASV(3,2)*FF(5)+ASV(3,3)*FF(6)
      IF (MMIC) 500,540,500
500   DO 520 I=1,3
      DO 510 J=1,3
      SUMA(J)=0.
      DO 510 K=1,3
510   SUMA(J)=SUMA(J)+ASV(J,K)*TEMP(K,I)
      DO 520 L=1,3
520   TEMP(L,I)=SUMA(L)
      N=3
      DO 530 I=1,3
      N=N+1
      DO 530 J=1,3
530   XDD(1,N)=XDD(1,N)-TEMP(I,J)*XD(1,J+3)
540   DO 550 I=1,3
550   TEMP(I,I)=0.
      TEMP(1,2)=-XD(1,6)
      TEMP(1,3)=XD(1,5)
      TEMP(2,1)=XD(1,6)
      TEMP(2,3)=-XD(1,4)
      TEMP(3,1)=-XD(1,5)
      TEMP(3,2)=XD(1,4)
      DO 570 I=1,3
      DO 560 J=1,3
      SUMA(J)=0.
      DO 560 K=1,3
560   SUMA(J)=SUMA(J)+ASV(J,K)*TEMP(K,I)
      DO 570 L=1,3
570   TEMP(L,I)=SUMA(L)
      DO 590 I=1,3
      DO 580 J=1,3
      SUMA(J)=0.
      DO 580 K=1,3
580   SUMA(J)=SUMA(J)*TEMP(I,K)*AA(K,J)
      DO 590 L=1,3
590   TEMP(I,L)=SUMA(L)
      N=3
      DO 600 I=1,3
```

```
      N=N+1                                                             MAN6170
      DO 600 J=1,3                                                      MAN6180
600   XDD(1,N)=XDD(1,N)-TEMP(I,J)*XD(1,J+3)                             MAN6190
      DO 690 I=1,6                                                      MAN6200
      GO TO (610,620,630,640,650,660), I                                MAN6210
610   IF (NTX.EQ.0) 670,680                                             MAN6220
620   IF (NTY.EQ.0) 670,680                                             MAN6230
630   IF (NTZ.EQ.0) 670,680                                             MAN6240
640   IF (NRX.EQ.0) 670,680                                             MAN6250
650   IF (NRY.EQ.0) 670,680                                             MAN6260
660   IF (NRZ.EQ.0) 670,680                                             MAN6270
670   IRLM=(I-1)*7                                                      MAN6280
      CALL INTEG (298+IRLM,165+IRLM)                                    MAN6290
      GO TO 690                                                         MAN6300
680   XD(1,I)=0.0                                                       MAN6310
      XDD(1,I)=0.0                                                      MAN6320
690   CONTINUE                                                          MAN6330
      XD(1,7)=XD(1,6)*X(1,10)-XD(1,5)*X(1,13)                           MAN6340
      XD(1,8)=XD(1,6)*X(1,11)-XD(1,5)*X(1,14)                           MAN6350
      XD(1,9)=XD(1,6)*X(1,12)-XD(1,5)*X(1,15)                           MAN6360
      XD(1,10)=XD(1,4)*X(1,13)-XD(1,6)*X(1,7)                           MAN6370
      XD(1,11)=XD(1,4)*X(1,14)-XD(1,6)*X(1,8)                           MAN6380
      XD(1,12)=XD(1,4)*X(1,15)-XD(1,6)*X(1,9)                           MAN6390
      XD(1,13)=XD(1,5)*X(1,7)-XD(1,4)*X(1,10)                           MAN6400
      XD(1,14)=XD(1,5)*X(1,8)-XD(1,4)*X(1,11)                           MAN6410
      XD(1,15)=XD(1,5)*X(1,9)-XD(1,4)*X(1,12)                           MAN6420
      DO 700 I=7,15                                                     MAN6430
      IRLM=(I-1)*7                                                      MAN6440
      CALL INTEG (165+IRLM,32+IRLM)                                     MAN6450
700   CONTINUE                                                          MAN6460
      IF (IFIN) 740,710,740                                             MAN6470
710   IF (IVAL) 740,720,720                                             MAN6480
720   DO 730 I=7,15                                                     MAN6490
      N=I-6                                                             MAN6500
730   DC(N)=X(1,I)                                                      MAN6510
      DELT1=DC(1)*DC(5)*DC(9)+DC(3)*DC(4)*DC(8)+DC(2)*DC(6)*DC(7)-DC(3)*MAN6520
     1DC(5)*DC(7)-DC(2)*DC(4)*DC(9)-DC(1)*DC(6)*DC(8)                   MAN6530
      X(1,7)=.5*(DC(1)+(DC(5)*DC(9)-DC(8)*DC(6))/DELT1)                 MAN6540
      X(1,8)=.5*(DC(2)+(DC(7)*DC(6)-DC(4)*DC(9))/DELT1)                 MAN6550
      X(1,9)=.5*(DC(3)+(DC(4)*DC(8)-DC(7)*DC(5))/DELT1)                 MAN6560
      X(1,10)=.5*(DC(4)+(DC(8)*DC(3)-DC(2)*DC(9))/DELT1)                MAN6570
      X(1,11)=.5*(DC(5)+(DC(1)*DC(9)-DC(7)*DC(3))/DELT1)                MAN6580
      X(1,12)=.5*(DC(6)+(DC(7)*DC(2)-DC(8)*DC(1))/DELT1)                MAN6590
      X(1,13)=.5*(DC(7)+(DC(2)*DC(6)-DC(5)*DC(3))/DELT1)                MAN6600
      X(1,14)=.5*(DC(8)+(DC(4)*DC(3)-DC(1)*DC(6))/DELT1)                MAN6610
      X(1,15)=.5*(DC(9)+(DC(1)*DC(5)-DC(4)*DC(2))/DELT1)                MAN6620
740   TR(1,1)=X(1,7)                                                    MAN6630
      TR(2,1)=X(1,8)                                                    MAN6640
      TR(3,1)=X(1,9)                                                    MAN6650
      TR(1,2)=X(1,10)                                                   MAN6660
      TR(2,2)=X(1,11)                                                   MAN6670
      TR(3,2)=X(1,12)                                                   MAN6680
      TR(1,3)=X(1,13)                                                   MAN6690
      TR(2,3)=X(1,14)                                                   MAN6700
      TR(3,3)=X(1,15)                                                   MAN6710
      TRL(1,1)=TR(1,1)                                                  MAN6720
```

```
      TRL(1,2)=TR(1,2)
      TRL(1,3)=TR(1,3)
      TRL(2,1)=TR(2,1)*CSLP-TR(3,1)*SSLP
      TRL(2,2)=TR(2,2)*CSLP-TR(3,2)*SSLP
      TRL(2,3)=TR(2,3)*CSLP-TR(3,3)*SSLP
      TRL(3,1)=TR(2,1)*SSLP+TR(3,1)*CSLP
      TRL(3,2)=TR(2,2)*SSLP+TR(3,2)*CSLP
      TRL(3,3)=TR(2,3)*SSLP+TR(3,3)*CSLP
      DO 770 J=1,3
      CONT1=0.0
      CONT2=0.0
      DO 760 I=1,3
      CONT1=CONT1+TR(I,J)*TR(I,J)
      CONT2=CONT2+TRL(I,J)*TRL(I,J)
      IF (ABS(TR(I,J)).LE.1.) GO TO 750
      WRITE (6,1340) I,J,TR(I,J)
  750 IF (ABS(TRL(I,J)).LE.1.) GO TO 760
      WRITE (6,1350) I,J,TRL(I,J)
  760 CONTINUE
      CONT1=SQRT(CONT1)
      CONT2=SQRT(CONT2)
      DO 770 K=1,3
      TR(K,J)=TR(K,J)/CONT1
  770 TRL(K,J)=TRL(K,J)/CONT2
      NX=15
      DO 790 I=1,3
      NX=NX+1
      XD(1,NX)=0.
      DO 780 J=1,3
  780 XD(1,NX)=XD(1,NX)+TR(I,J)*XD(1,J)
  790 XVF(I)=XD(1,NX)
      TOTGV=SQRT(XVF(1)**2+XVF(2)**2+XVF(3)**2)
      XDLS=XVF(1)
      YDLS=XVF(2)*CSLP-XVF(3)*SSLP
      ZDLS=XVF(2)*SSLP+XVF(3)*CSLP
      TOTSV=SQRT(XDLS**2+YDLS**2+ZDLS**2)
      CON1=TOTSV+1.E-10
      ANG1=ASIN(-ZDLS/CON1)
      XD(1,19)=TOTSV*COS(ANG1)
      IF (INDBAC.NE.0) GO TO 810
      DO 800 I=16,19
      IRLM=(I-1)*7
  800 CALL INTEG (165+IRLM,32+IRLM)
  810 CONTINUE
      DO 820 I=1,3
      GACC(I)=0.
      DO 820 J=1,3
  820 GACC(I)=GACC(I)+XDD(1,J)*TR(I,J)
      XDDLS=GACC(1)
      YDDLS=GACC(2)*CSLP-GACC(3)*SSLP
      ZDDLS=GACC(2)*SSLP+GACC(3)*CSLP
      XNUMB=TR(3,1)
      XNUMBL=TRL(3,1)
      IF (ABS(XNUMB)-1.) 840,840,830
  830 XNUMB=XNUMB/ABS(XNUMB)
  840 THETA=-ASIN(XNUMB)*DEGRAD
```

```
      IF (ABS(XNUMBL)-1.) 860,860,850                                   MAN7290
850   XNUMBL=XNUMBL/ABS(XNUMBL)                                         MAN7300
860   THETAL=-ASIN(XNUMBL)*DEGRAD                                       MAN7310
      IF (ABS(TR(2,1))+ABS(TR(1,1)).NE.0.0) GO TO 870                   MAN7320
      ICONT=PHIP/70.                                                    MAN7330
      PHI=90.*ICONT                                                     MAN7340
      ICONT=PSIP/70.                                                    MAN7350
      PSI=90.*ICONT                                                     MAN7360
      IF (SLOPE.NE.0.0) GO TO 880                                       MAN7370
      PSIL=PSI                                                          MAN7380
      PHIL=PHI                                                          MAN7390
      GO TO 890                                                         MAN7400
870   PSI=ATAN2(TR(2,1),TR(1,1))*DEGRAD                                 MAN7410
      PHI=ATAN2(TR(3,2),TR(3,3))*DEGRAD                                 MAN7420
880   PSIL=ATAN2(TRL(2,1),TRL(1,1))*DEGRAD                              MAN7430
      PHIL=ATAN2(TRL(3,2),TRL(3,3))*DEGRAD                              MAN7440
890   CON1=TOTGV+1.E-10                                                 MAN7450
      PSIP=PSI                                                          MAN7460
      PHIP=PHI                                                          MAN7470
      GAMA=ASIN(-XVF(3)/CON1)*DEGRAD                                    MAN7480
      IF (XVF(2)) 920,900,920                                           MAN7490
900   IF (XVF(1)) 920,910,920                                           MAN7500
910   SIGMA=0.                                                          MAN7510
      GO TO 930                                                         MAN7520
920   SIGMA=ATAN2(XVF(2),XVF(1))*DEGRAD                                 MAN7530
930   ALT=-X(1,18)                                                      MAN7540
      RANGE=X(1,19)                                                     MAN7550
      XF=X(1,16)                                                        MAN7560
      YF=X(1,17)                                                        MAN7570
      ZF=X(1,18)                                                        MAN7580
      XLS=XF                                                            MAN7590
      YLS=YF*CSLP-ZF*SSLP                                               MAN7600
      ZLS=YF*SSLP+ZF*CSLP                                               MAN7610
      TLSV=SQRT(XLS**2+YLS**2)                                          MAN7620
      IF (INDBAC.EQ.0) GO TO 940                                        MAN7630
C          RESET SAVED TIME HISTORY VARIABLE FOR AF, STROKE, FORCE, VOL, MAN7640
C                AND KODEUP                                             MAN7650
      AFS=SAFS                                                          MAN7660
      SSAVE=SSSAVE                                                      MAN7670
      FSAVE=SFSAVE                                                      MAN7680
      VSAVE=SVSAVE                                                      MAN7690
      KODEUP=SKODUP                                                     MAN7700
      INDBAC=0                                                          MAN7710
      GO TO 420                                                         MAN7720
940   IF (IFIN) 980,950,980                                             MAN7730
C          IVAL .LT. 0 IMPLIES INTEGRATION BACKUP ON VARIABLE STEP      MAN7740
950   IF (IVAL) 970,960,960                                             MAN7750
960   KOUNT1=KOUNT1+1                                                   MAN7760
      INDACL=1                                                          MAN7770
      GO TO 980                                                         MAN7780
970   INDBAC=1                                                          MAN7790
980   CALL CUT                                                          MAN7800
      IF (JCUT.LT.0) GO TO 1170                                         MAN7810
      IF (INDACL.EQ.0) GO TO 1000                                       MAN7820
      INDACL=0                                                          MAN7830
      IF (ACLR.GE.0.0) GO TO 990                                        MAN7840
```

```
      WRITE (6,1360) ACLR                                          MAN7850
      GO TO 20                                                     MAN7860
990   CONTINUE                                                     MAN7870
1000  IF (JCUT.GT.0) GO TO 1020                                    MAN7880
      IF (KOUNT1-KOUNT2) 1180,1010,1180                            MAN7890
1010  KOUNT1=0                                                     MAN7900
C     INTEGRATED VALUES OF VARIABLES ARE READY FOR                 MAN7910
C     PRINTING - ACCELERATIONS ARE XDD(1,J), VELOCITIES ARE        MAN7920
C     XD(1,J), AND DISPLACEMENTS ARE X(1,J)-J=1 TO MAX             MAN7930
C     NUMBER OF VARIABLES.                                         MAN7940
1020  IF (JCUT.EQ.LOCSTR) GO TO 1140                               MAN7950
      DO 1030 I=1,3                                                MAN7960
      XLPF(I)=0.0                                                  MAN7970
      DO 1030 J=1,3                                                MAN7980
      XLPF(I)=XLPF(I)+TRL(I,J)*XPF(J)                              MAN7990
1030  ANG(I,J)=ACOS(TRL(I,J))*DEGRAD                               MAN8000
      CALL PRINT2                                                  MAN8010
      IF (NMBNC.EQ.NMBNCS) GO TO 1120                              MAN8020
      IF (JCUT) 1170,1180,1040                                     MAN8030
1040  LOCA=(NSAV(JCUT)-13)/6                                       MAN8040
      GO TO (1050,1060,1070,1080,1090,440,1100,1110), LOCA         MAN8050
1050  WRITE (6,1380) T                                             MAN8060
      GO TO 1130                                                   MAN8070
1060  WRITE (6,1390) X(1,19)                                       MAN8080
      GO TO 1130                                                   MAN8090
1070  WRITE (6,1400) XD(1,18)                                      MAN8100
      GO TO 1130                                                   MAN8110
1080  WRITE (6,1410) XD(1,16)                                      MAN8120
      GO TO 1130                                                   MAN8130
1090  WRITE (6,1420) XD(1,17)                                      MAN8140
      GO TO 1130                                                   MAN8150
      WRITE (6,1430) XD(1,4)                                       MAN8160
      GO TO 1130                                                   MAN8170
1100  WRITE (6,1440) XD(1,5)                                       MAN8180
      GO TO 1130                                                   MAN8190
1110  WRITE (6,1450) XD(1,6)                                       MAN8200
      GO TO 1130                                                   MAN8210
1120  WRITE (6,1460) NMBNC                                         MAN8220
1130  WRITE (6,1370)                                               MAN8230
      GO TO 20                                                     MAN8240
1140  IF (IND(LOCSTR).NE.0) GO TO 1150                             MAN8250
C        TOUCH-DOWN                                                MAN8260
C              INTEGRATION CUTOFF ON STROKE = 0, RESET INDICATORS TO  MAN8270
C                 CUT OFF INTEGRATION WITH TORUS LIFTS-OFF         MAN8280
      IND(LOCSTR)=1                                                MAN8290
      XS(LOCSTR)=-3.*CUTERR                                        MAN8300
      GO TO 1160                                                   MAN8310
C        LIFT-OFF                                                  MAN8320
C              INTEGRATION CUTOFF ON STROKE .GT. 0, RESET INDICATORS TO MAN8330
C                 CUT OFF INTEGRATION WITH TORUS TOUCH-DOWN        MAN8340
1150  IND(LOCSTR)=0                                                MAN8350
      NMBNC=NMBNC+1                                                MAN8360
      XS(LOCSTR)=0.0                                               MAN8370
1160  CALL SETUP                                                   MAN8380
      KOUNT1=KOUNT2-1                                              MAN8390
      GO TO 420                                                    MAN8400
```

```
C         KOUNT1 = -8200 (ARBITRARY CONSTANT), PREVENTS PRINTING UNTIL  MAN8410
C              CUTOFF                                                   MAN8420
 1170  KOUNT1=-8200                                                      MAN8430
       ACLR=0.0                                                          MAN8440
       INDBAC=1                                                          MAN8450
       GO TO 740                                                         MAN8460
 1180  CALL UPDAT                                                        MAN8470
C         INTEGRATION BACKUP REQUIRES A RETURN TO 2 FOR VARIABLE SETUP  MAN8480
       IF (INDBAC.NE.0) GO TO 740                                        MAN8490
       GO TO 420                                                         MAN8500
C                                                                        MAN8510
 1190  FORMAT (*1*,                                                      MAN8520
      *      40X*INFLATABLE TORUS LANDING LOADS AND MOTIONS PROGRAM*/    MAN8530
      *  39X*MASTER AGREEMENT, CONTRACT NAS1-8137, TASK ORDER NO. 1*    /MAN8540
      * 38X*MCDONNELL DOUGLAS ASTRONAUTICS COMPANY, EASTERN DIVISION*    MAN8541
      * //61X*INPUT DATA*// )                                            MAN8542
 1200  FORMAT (55H                                                       MAN8550
      1)                                                                 MAN8560
 1210  FORMAT (* *,*NO MORE DATA, END OF JOB *)                          MAN8570
 1220  FORMAT (1X,I1,2X,I3,2X,6E10.5)                                    MAN8580
 1230  FORMAT (*0*,*DATA SUBSCRIPT*,2X6(4XA6,4X))                        MAN8590
 1240  FORMAT (* *,20X*XS(I)*,9X*IND(I)* )                               MAN8600
 1250  FORMAT (4X,I5,8X,6E14.7)                                          MAN8610
 1260  FORMAT (1H1,10X,60HNO. OF CUT-OFF VALUES ENTERED AS ZERO-PROBLEM CMAN8620
      1ANNOT PROCEED)                                                    MAN8630
 1270  FORMAT (1XA10,3XI5,1XI5)                                          MAN8640
 1280  FORMAT (*0*,*PRINT POSITION *,I4,* WILL CONTAIN THE VALUE FROM *, MAN8650
      *  *COMINT(*,I5,*) WITH THE LABEL *,A10 )                          MAN8660
 1290  FORMAT(*0*,A10,5X*END OF DATA LIST*)                              MAN8670
 1300  FORMAT (* *,*----------ERROR----------VMIN = DATA)111* = 0*  )    MAN8680
 1310  FORMAT (*1*,* VEHICLE WEIGHT, ON EARTH = *,E14.7,*, ON PLANET = * MAN8690
      1                                            ,E14.7              ) MAN8691
 1320  FORMAT (*0*,16X*MOMENTS AND PRODUCTS OF INERTIA*            /     MAN8700
      *          *0*,7X*XXI*,13X*XYI*,12X*XXID*, 12X*XYID*/              MAN8710
      1          * *,7X*YYI*,13X*XZI*,12X*YYID*, 12X*XZID*/              MAN8720
      1          * *,7X*ZZI*,13X*YZI*,12X*ZZID*, 12X*YZID*          )    MAN8721
 1330  FORMAT (*0*,4(1XE14.7)/* *,4(1XE14.7)/* *,4(1XE14.7))             MAN8730
 1340  FORMAT (* *,*----------ERROR----------TR (*,I2,*,*,I2,*)= *,E20.13MAN8740
      1,VECTOR TAKES UNIT VECTOR FORM*)                                  MAN8750
 1350  FORMAT (* *,*----------ERROR----------TRL(*,I2,*,*,I2,*)= *,E20.13MAN8760
      1     ,*.  VECTOR TAKES UNIT VECTOR FORM* )                        MAN8770
 1360  FORMAT (//,1X,41HCUT OFF,CLEARANCE LESS THAN ALLOWABLE BY ,F10.4) MAN8780
 1370  FORMAT (1H1)                                                      MAN8790
 1380  FORMAT (//,1X,16HCUT OFF ON TIME=,F10.4,1X,7HSECONDS)             MAN8800
 1390  FORMAT (//,1X,38HCUT OFF ON TOTAL VEL RELATIVE TO SURF=,F10.4,1X,6MAN8810
      1HFT/SEC)                                                          MAN8820
 1400  FORMAT (//,1X,25HCUT OFF ON SURFACE RANGE=,F10.4,1X,2HFT)         MAN8830
 1410  FORMAT (//,1X,35HCUT OFF ON PARALLEL TO SURFACE VEL=,F10.4,1X,6HFTMAN8840
      1/SEC)                                                             MAN8850
 1420  FORMAT (//,1X,38HCUT OFF ON DISTANCE NORMAL TO SURFACE=,F10.4,1X, MAN8860
      12HFT)                                                             MAN8870
 1430  FORMAT (//,1X,28HCUT OFF ON LANDER ROLL RATE=,F10.4,1X,7HRAD/SEC) MAN8880
 1440  FORMAT (//,1X,29HCUT OFF ON LANDER PITCH RATE=,F10.4,1X,7HRAD/SEC)MAN8890
 1450  FORMAT (//,1X,27HCUT OFF ON LANDER YAW RATE=,F10.4,1X,7HRAD/SEC)  MAN8900
 1460  FORMAT (//* CUTOFF ON NUMBER OF BOUNCES, NUMBER = *,I5)           MAN8910
       END                                                               MAN8920
```

```
      SUBROUTINE ENVIR
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 554), SOUND   )
      EQUIVALENCE ( COMINT( 398), XWND    )
      EQUIVALENCE ( COMINT( 399), YWND    )
      EQUIVALENCE ( COMINT( 400), ZWND    )
      EQUIVALENCE ( COMINT( 389), XW      )
      EQUIVALENCE ( COMINT( 390), YW      )
      EQUIVALENCE ( COMINT( 391), ZW      )
C     READ ENVIRONMENT DATA
      RETURN
      ENTRY ENVIR1
C     DEFINE GROUND REF, ENVIRONMENT AT TIME=T
C     XW=NORTH WIND YW=EAST WIND ZW=VERTICAL WIND
      SOUND=0.
      XWND=0.
      YWND=0.
      ZWND=0.
      XW=0.
      YW=0.
      ZW=0.
      RETURN
      END
```

APPENDIX D

```
      SUBROUTINE AERO
      COMMON COMINT(600)
      EQUIVALENCE ( COMINT( 552), Q        )
      EQUIVALENCE ( COMINT( 553), XMACH    )
      EQUIVALENCE ( COMINT( 404), BETA     )
      EQUIVALENCE ( COMINT( 401), ALPHA    )
C     PROVIDE ALL DATA HERE NECESSARY TO DEFINE
C     AERODYNAMIC FORCES ACTING ON THE VEHICLE.
      RETURN
      ENTRY AERO1
C     DEFINE AERO FORCES ON VEHICLE AT TIME=T
      CALL ENVIR1
      XMACH=0.
      Q=0.
      ALPHA=0.
      BETA=0.
      RETURN
      END
```

APPENDIX D

```
      SUBROUTINE PCCUT (IYD,IY,CUTVAL,IDIRT)
      DIMENSION X(9), XMN1(9)
      DIMENSION XS(9), IND(9), IAD(9), NAMBCD(9), LOCNAM(50)
      COMMON COMINT (600)
      EQUIVALENCE ( COMINT(  1 ), T        )
      EQUIVALENCE ( COMINT(  2 ), HMAX     )
      EQUIVALENCE ( COMINT(  3 ), EMIN     )
      EQUIVALENCE ( COMINT(  4 ), EMAX     )
      EQUIVALENCE ( COMINT(  5 ), HZ       )
      EQUIVALENCE ( COMINT(  6 ), IP       )
      EQUIVALENCE ( COMINT(  7 ), IVARH    )
      EQUIVALENCE ( COMINT(  8 ), IMTH     )
      EQUIVALENCE ( COMINT(  9 ), IPRNT    )
      EQUIVALENCE ( COMINT( 10 ), IFIN     )
      EQUIVALENCE ( COMINT( 11 ), IVAL     )
      EQUIVALENCE ( COMINT( 12 ), IPTOTL   )
      EQUIVALENCE ( COMINT( 13 ), IPTATL   )
      EQUIVALENCE ( COMINT(  14), XS       )
      EQUIVALENCE ( COMINT(  23), IND      )
      EQUIVALENCE ( COMINT( 347), HMIN     )
      EQUIVALENCE ( COMINT( 384), CUTERR   )
      EQUIVALENCE ( COMINT( 385), J        )
      ENTRY LOC
      IPTATL=IPTATL+1
      IF (IPTATL.LE.9) GO TO 10
      WRITE (6,870)
      STOP
10    NAMBCD(IPTATL)=IY
      IAD(IPTATL)=IYD
      XS(IPTATL)=CUTVAL
      IND(IPTATL)=IDIRT
      RETURN
      ENTRY INUPD
      IPTOTL=IPTOTL+1
      IF (IPTOTL.LE.50) GO TO 20
      WRITE (6,880)
      STOP
20    LOCNAM(IPTOTL)=IYD
      RETURN
      ENTRY SETUP
      IERROR=0
      ISTEP=1
      I=0
      IVAL=0
      HZ=HMAX*2.**(-IP)
      IPT2=2**IP
      IPT1=0
      IPRNT=0
      IFIN=0
      INDRH=1
      IB5=1
      IB1=1
      IALP=4
      LIST=0
      INUPD=0
      IF (IMTH) 70,30,70
```

```
30    IB2=1
      JJ=7
      IB3=2
      ISCNT=0
      IBETA=3
      IGAM=-1
      IF (IVARH) 40,50,40
40    IB4=1
      IB6=2
      GO TO 60
50    IB4=2
      IB6=1
60    IB7=1
      GO TO 80
70    IB2=2
      JJ=6
      IB3=1
      IB7=2
80    HD2=HZ/2.
      H=HD2
      A1111=HZ/24.
      A2222=19./270.
      RKTME=T
      RETURN
      ENTRY INTEG
      JJR1=IY+JJ-1
      GO TO (90,130,140,150), IB5
90    GO TO (100,110,120,100), ISTEP
100   COMINT(JJR1)=COMINT(IY)+H*COMINT(IYD)
      RETURN
110   COMINT(JJR1)=COMINT(IY+1)+H*COMINT(IYD)
      RETURN
120   COMINT(JJR1)=COMINT(IY+2)+H*COMINT(IYD)
      RETURN
130   COMINT(JJR1)=COMINT(IY+3)+H/6.*(COMINT(IYD+3)+2.*(COMINT(IYD+2)+CO
     1MINT(IYD+1))+COMINT(IYD))
      RETURN
140   CONTINUE
      COMINT(IY+5)=COMINT(IY)+A1111*(55.*COMINT(IYD)-59.*COMINT(IYD+1)+3
     17.*COMINT(IYD+2)-9.*COMINT(IYD+3))
      RETURN
150   CN1=COMINT(IY+1)+A1111*(9.*COMINT(IYD)+19.*COMINT(IYD+1)-5.*COMINT
     1(IYD+2)+COMINT(IYD+3))
      XM=ABS(COMINT(IY)-CN1)
      COMINT(IY)=CN1+A2222*(COMINT(IY)-CN1)
      IF (IB6-2) 170,160,170
160   RETURN
170   IF (CN1) 180,200,180
180   XM1=ABS(XM/CN1)
      IF (XM-XM1) 200,200,190
190   XM=XM1
200   IF (XM-EMAX) 220,210,210
210   IF (HZ.GT.HMIN) IVAL=-8300
      IVAL=IVAL+1
      RETURN
220   IF (XM-EMIN) 240,230,230
```

```
230   IVAL=IVAL+1
240   RETURN
      ENTRY UPDAT
      IFIN=1
      IF (IPRNT) 260,250,260
250   IPT1=IPT2
260   IF (IB1-2) 270,500,270
270   IF (IALP-1) 390,280,390
280   I=I+1
      IPT1=IPT1-1
      GO TO (360,300), IB2
290   IBETA=IBETA-1
300   IALP=4
      ISTEP=1
      H=H/2.
      IFIN=0
      IB5=1
      GO TO (330,310), IB3
310   DO 320 IMVER=1,IPTOTL
      KMVER=LOCNAM(IMVER)
      COMINT(KMVER+1)=COMINT(KMVER+3)
      COMINT(KMVER+2)=COMINT(KMVER+4)
      COMINT(KMVER+3)=COMINT(KMVER+5)
      COMINT(KMVER+4)=COMINT(KMVER+5)
320   COMINT(KMVER)=COMINT(KMVER+6)
      GO TO 350
330   DO 340 IMVER=1,IPTOTL
      KMVER=LOCNAM(IMVER)
      COMINT(KMVER)=COMINT(KMVER+5)
340   COMINT(KMVER+1)=COMINT(KMVER+3)
350   IPRNT=IPT1
      RETURN
360   IF (IBETA-1) 290,370,290
370   IB1=2
      IB5=3
      IFIN=0
      IF (IVARH) 380,310,380
380   IB6=2
      GO TO 310
390   IALP=IALP-1
      IF (IALP-1) 400,410,400
400   ISTEP=ISTEP+1
      GO TO 420
410   IB5=2
420   IF (IALP-2) 430,440,430
430   T=T+HD2
      GO TO 450
440   H=HZ
450   GO TO (460,480), IB7
460   DO 470 IMVER=1,IPTOTL
      KMVER=LOCNAM(IMVER)
      COMINT(KMVER+5)=COMINT(KMVER+4)
      COMINT(KMVER+4)=COMINT(KMVER+3)
      COMINT(KMVER+3)=COMINT(KMVER+2)
      COMINT(KMVER+2)=COMINT(KMVER+1)
      COMINT(KMVER+1)=COMINT(KMVER)
```

```
470   COMINT(KMVER)=COMINT(KMVER+6)                    PCT1690
      GO TO 350                                        PCT1700
480   DO 490 IMVER=1,IPTOTL                            PCT1710
      KMVER=LOCNAM(IMVER)                              PCT1720
      COMINT(KMVER+3)=COMINT(KMVER+2)                  PCT1730
      COMINT(KMVER+2)=COMINT(KMVER+1)                  PCT1740
      COMINT(KMVER+1)=COMINT(KMVER)                    PCT1750
490   COMINT(KMVER)=COMINT(KMVER+5)                    PCT1760
      GO TO 350                                        PCT1770
500   IGAM=-IGAM                                       PCT1780
      IF (IGAM) 520,520,510                            PCT1790
510   IB5=4                                            PCT1800
      T=T+H                                            PCT1810
      GO TO 480                                        PCT1820
520   IB5=3                                            PCT1830
      GO TO (530,540), IB4                             PCT1840
530   IPT1=IPT1-1                                      PCT1850
      IFIN=0                                           PCT1860
      GO TO 570                                        PCT1870
540   IF (IVAL) 550,580,550                            PCT1880
550   ISCNT=0                                          PCT1890
      IF (IVAL) 630,630,560                            PCT1900
560   INDRH=1                                          PCT1910
      IPT1=IPT1-1                                      PCT1920
      I=I+1                                            PCT1930
      IFIN=0                                           PCT1940
570   IVAL=0                                           PCT1950
      GO TO 350                                        PCT1960
580   IF (ISCNT-2) 590,590,600                         PCT1970
590   ISCNT=ISCNT+1                                    PCT1980
      GO TO 560                                        PCT1990
600   IF (2*(IPT1/2).EQ.IPT1) GO TO 560                PCT2000
      IF (H-HMAX) 610,560,560                          PCT2010
610   IPT1=IPT1/2                                      PCT2020
      ISCNT=0                                          PCT2030
      IBETA=3                                          PCT2040
      IALP=4                                           PCT2050
      IB1=1                                            PCT2060
      IB5=1                                            PCT2070
      ISTEP=1                                          PCT2080
      INDRH=0                                          PCT2090
      IPT2=IPT2/2                                      PCT2100
      RKTML=T                                          PCT2110
      HD2=H                                            PCT2120
      HZ=2.*H                                          PCT2130
      IFIN=0                                           PCT2140
620   I=0                                              PCT2150
      A1111=HZ/24.                                     PCT2160
      GO TO 570                                        PCT2170
630   IF (IPT1) 560,640,640                            PCT2180
640   IBETA=3                                          PCT2190
      IALP=4                                           PCT2200
      ISTEP=1                                          PCT2210
      IB1=1                                            PCT2220
      IB5=1                                            PCT2230
      IF (I-3) 650,670,650                             PCT2240
```

```
650    T=T-H
       RKTME=T
       IPT1=2*IPT1
       DO 660 IMVER=1,IPTOTL
       KMVER=LOCNAM(IMVER)
660    COMINT(KMVER)=COMINT(KMVER+1)
       GO TO 690
670    T=RKTME
       IPT1=2*(IPT1+3)
       DO 680 IMVER=1,IPTOTL
       KMVER=LOCNAM(IMVER)
680    COMINT(KMVER)=COMINT(KMVER+4)
690    HZ=H/2.
       IF (HZ.LT.HMIN) HZ=HMIN
       HD2=HZ/2.
       H=HD2
       IPT2=2*IPT2
       INDRH=-1
       GO TO 620
       ENTRY CUT
       IF (IFIN) 700,710,700
700    J=0
       IERROR=1
       RETURN
710    K=1
720    IF (K.LE.IPTATL) GO TO 750
       IF (K-1) 730,700,730
730    IK=K-1
       DO 740 I=1,IK
       KMVER=IAD(I)
       XMN1(I)=COMINT(KMVER)
740    CONTINUE
       GO TO 700
750    KMVER=IAD(K)
       X(K)=COMINT(KMVER)
       XU=XS(K)+(ABS(XS(K))+1.)*CUTERR
       XL=XS(K)-(ABS(XS(K))+1.)*CUTERR
       IF (IND(K)) 760,780,760
760    IF (X(K)-XU) 770,790,790
770    IF (X(K)-XL) 820,820,810
780    IF (X(K)-XL) 790,790,800
790    K=K+1
       IF (K-10) 720,730,730
800    IF (X(K)-XU) 810,820,820
810    J=K
       IERROR=1
       RETURN
820    T=T-HZ
       HZ=(HZ*(XS(K)-XMN1(K))/(X(K)-XMN1(K)))/2.
       IF (INDRH) 840,830,840
830    HZ=HZ/2.
840    I=0
       HD2=HZ/2.
       H=HD2
       IF (IERROR.NE.0) GO TO 850
       KMVER=IAD(K)
```

```
      WRITE (6,890) NAMBCD(K),COMINT(KMVER)                            PCT2810
      STOP                                                              PCT2820
850   CONTINUE                                                          PCT2830
      A1111=B2/24.                                                      PCT2840
      ISCNT=0                                                           PCT2850
      IBETA=3                                                           PCT2860
      IALP=4                                                            PCT2870
      ISTEP=1                                                           PCT2880
      IB1=1                                                             PCT2890
      IB2=2                                                             PCT2900
      IB3=1                                                             PCT2910
      IB5=1                                                             PCT2920
      IB7=2                                                             PCT2930
      JJ=6                                                              PCT2940
      IPT1=1000                                                         PCT2950
      IPT2=1000                                                         PCT2960
      IPRNT=1000                                                        PCT2970
      RKTME=T                                                           PCT2980
      DO 860 IMVER=1,IPTOTL                                             PCT2990
      KMVER=LOCNAM(IMVER)                                               PCT3000
860   COMINT(KMVER)=COMINT(KMVER+1)                                     PCT3010
      IFIN=1                                                            PCT3020
      J=-1                                                              PCT3030
      IERROR=1                                                          PCT3040
      RETURN                                                            PCT3050
C                                                                       PCT3060
870   FORMAT (* *,*-----------JOB TERMINATED, MORE THAN NINE CALLS TO LOCPCT3070
     1----------* )                                                     PCT3080
880   FORMAT(* *,*-----------JOB TERMINATED, MORE THAN FIFTY CALLS TO *  PCT3090
     1   ,*INUPS-----------* )                                          PCT3100
890   FORMAT (18H CUTOFF PASSED BY ,A6,3H = ,E14.7,27H ON THE INITIAL CAPCT3110
     1LL TO CUT)                                                        PCT3120
      END                                                               PCT3130-
```

APPENDIX D

```
      SUBROUTINE MASS
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 392), SCFWT   )
      EQUIVALENCE ( COMINT( 416), XMASST  )
      EQUIVALENCE ( COMINT( 410), XXI     )
      EQUIVALENCE ( COMINT( 411), YYI     )
      EQUIVALENCE ( COMINT( 412), ZZI     )
      EQUIVALENCE ( COMINT( 418), XYI     )
      EQUIVALENCE ( COMINT( 419), XZI     )
      EQUIVALENCE ( COMINT( 420), YZI     )
      EQUIVALENCE ( COMINT( 422), XXID    )
      EQUIVALENCE ( COMINT( 423), YYID    )
      EQUIVALENCE ( COMINT( 424), ZZID    )
      EQUIVALENCE ( COMINT( 425), XYID    )
      EQUIVALENCE ( COMINT( 426), XZID    )
      EQUIVALENCE ( COMINT( 427), YZID    )
      EQUIVALENCE ( COMINT( 459), ANGLE   )
      EQUIVALENCE ( COMINT( 478), R1      )
      EQUIVALENCE ( COMINT( 479), R2      )
      EQUIVALENCE ( COMINT( 480), R3      )
      EQUIVALENCE ( COMINT( 449), L       )
      EQUIVALENCE ( COMINT( 450), F       )
      EQUIVALENCE ( COMINT( 481), THETAT  )
      EQUIVALENCE ( COMINT( 482), RS      )
      EQUIVALENCE ( COMINT( 483), RL      )
      EQUIVALENCE ( COMINT( 484), FABWT   )
      EQUIVALENCE ( COMINT( 485), PLY1    )
      EQUIVALENCE ( COMINT( 486), PLY2    )
      EQUIVALENCE ( COMINT( 487), PLY3    )
      EQUIVALENCE ( COMINT( 453), RHOP    )
      EQUIVALENCE ( COMINT( 445), A       )
      EQUIVALENCE ( COMINT( 446), B       )
      EQUIVALENCE ( COMINT( 447), C       )
      EQUIVALENCE ( COMINT( 448), D       )
      EQUIVALENCE ( COMINT( 488), AF      )
      EQUIVALENCE ( COMINT( 348), STROKE  )
      XYI=0.0
      XZI=0.0
      YZI=0.0
      XXID=0.0
      YYID=0.0
      ZZID=0.0
      XYID=0.0
      XZID=0.0
      YZID=0.0
      PIE=3.14159
C     *  IMPACT BAG  *
      A1=2*PIE*RS*((2*PIE*RL)/3-1.73206*RS)
      A2=(4*PIE*PIE*RS*RL)/3
      A3=2*PIE*RS*((2*PIE*RL)/3+1.73206*RS)
      BMASS1=((FABWT*PLY1+SCFWT)*A1)/(144.*32.147)
      BMASS2=((FABWT*PLY2+SCFWT)*A2)/(144.*32.147)
      BMASS3=((FABWT*PLY3+SCFWT)*A3)/(144.*32.147)
      BAGM=BMASS1+BMASS2+BMASS3
      AA=0.5*RS
      BB=3.0*RS/PIE
```

```
      CC=0.86603*RS
      DD=2.59809*RS/PIE
      XXI1=BMASS1*((RL-DD)**2)/144.0
      YYI1=0.5*BMASS1*(((RL-DD)**2)+2.*CC*CC/3.)/144.0
      ZZI1=YYI1
      XXI2=2.*BMASS2*(RL*RL+AA*AA)/144.0
      YYI2=BMASS2*(RL*RL+AA*AA+2.*BB*BB)/144.0
      ZZI2=YYI2
      XXI3=BMASS3*((RL+DD)**2)/144.0
      YYI3=.5*BMASS3*(((RL+DD)**2)+2.*CC*CC/3.)/144.
      ZZI3=YYI3
      EQFABM=BAGM/(4.*PIE*PIE*RL*RS)
C     *  PAYLOAD PACKAGE  *
      PMASS4=(PIE*A*A*C*RHOP)/(4.*32.147*1728.)
      PMASS5=(PIE*B*RHOP*(((A+2.*D)**2)-A*A))/(2.*32.147*1728.)
      XXI4=(PMASS4*A*A)/(8.*144.)
      YYI4=(PMASS4*(0.75*A*A+C*C))/(12.*144.)
      ZZI4=YYI4
      XXI5=(PMASS5*(((A+2.*D)**2)+A*A))/(8.*144.)
      YYI5=(PMASS5*(0.75*(((A+2.*D)**2)+A*A)+4.*B*B))/(12.*144.)
      ZZI5=YYI5
C     *  TOTAL MASS PROPERTIES  *
      XMASST=PMASS4+PMASS5+BAGM
      XXI=XXI1+XXI2+XXI3+XXI4+XXI5
      YYI=YYI1+YYI2+YYI3+YYI4+YYI5
      ZZI=ZZI1+ZZI2+ZZI3+ZZI4+ZZI5
      RETURN
C     *** FLAT ANALYSIS ***
      ENTRY MASS1
      EFFWT=0.0
      NDIM=10
      TPFAFT=2.*PIE*FABWT/144.0
C     *  ARC A-B  *
      PHIO=F/RS
      PHI1=(R1*THETAT)/RS+PHIO
      IF (PHIO.GE.(30*PIE/180)) GO TO 20
      DPHI=((30*PIE/180)-PHIO)/NDIM
      PHIINT=PHIO+0.5*DPHI
      DO 10 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R1)*(PHI-PHIO)
      SY=STROKE+R1*(1.0-COS(BETA))-RS*(1.0-COS(PHI))
      SX=F+R1*SIN(BETA)-RS*SIN(PHI)
      RM=RL-RS*SIN(PHI)
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
   10 EFFWT=EFFWT+TPFAFT*FACTOR*PLY2*RM*RS*DPHI
      DPHI=(PHI1-(30*PIE/180))/NDIM
      PHIINT=(30*PIE/180)+0.5*DPHI
      GO TO 30
   20 DPHI=(PHI1-PHIO)/NDIM
      PHIINT=PHIO+0.5*DPHI
   30 CONTINUE
      DO 40 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R1)*(PHI-PHIO)
      SY=STROKE+R1*(1.0-COS(BETA))-RS*(1.0-COS(PHI))
```

APPENDIX D

```
      SX=F+R1*SIN(BETA)-RS*SIN(PHI)
      RM=RL-RS*SIN(PHI)
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
40    EFFWT=EFFWT+TPFAFT*FACTOR*PLY1*RM*RS*DPHI
C     *  ARC B*-B  *
      EFFWT=EFFWT+TPFAFT*RS*PLY1*(RL*(PIE/2-PHI1)+RS*COS(PHI1))
C     *  ARC B-T  *
      PHI1=(R3*PIE)/(2*RS)
      DPHI=(60*PIE)/(NDIM*180)
      PHIINT=0.5*DPHI
      DO 50 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R3)*PHI
      SX=E+RS*COS(PHI)-R3*COS(BETA)
      SY=R3*SIN(BETA)-RS*SIN(PHI)
      RM=RL-RS*COS(PHI)
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
50    EFFWT=EFFWT+TPFAFT*FACTOR*PLY1*RM*RS*DPHI
      DPHI=(PHI1-(60*PIE)/180)/NDIM
      PHIINT=(60*PIE)/180+0.5*DPHI
      DO 60 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R3)*PHI
      SX=E+RS*COS(PHI)-R3*COS(BETA)
      SY=R3*SIN(BETA)-RS*SIN(PHI)
      RM=RL-RS*COS(PHI)
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
60    EFFWT=EFFWT+TPFAFT*FACTOR*PLY2*RM*RS*DPHI
C     *  ARC C-T  *
      PHIO=E/RS
      PHI1=PHIO+(R2*PIE)/RS
      IF (PHIO.GE.(30*PIE/180)) GO TO 80
      DPHI=((30*PIE/180)-PHIO)/NDIM
      PHIINT=PHIO+0.5*DPHI
      DO 70 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R2)*(PHI-PHIO)
      SY=STROKE+R2*(1.0-COS(BETA))-RS*(1.0-COS(PHI))
      SX=RS*SIN(PHI)-R2*SIN(BETA)-E
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
      RM=RL+RS*SIN(PHI)
70    EFFWT=EFFWT+TPFAFT*FACTOR*PLY2*RM*RS*DPHI
      DPHI=(120*PIE)/(NDIM*180)
      PHIOP=(30*PIE)/180
      GO TO 90
80    DPHI=((150*PIE/180)-PHIO)/NDIM
      PHIOP=PHIO
90    PHIINT=PHIOP+0.5*DPHI
      DO 100 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R2)*(PHI-PHIO)
      SY=STROKE+R2*(1.0-COS(BETA))-RS*(1.0-COS(PHI))
      SX=RS*SIN(PHI)-R2*SIN(BETA)-E
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
      RM=RL+RS*SIN(PHI)
100   EFFWT=EFFWT+TPFAFT*FACTOR*PLY3*RM*RS*DPHI
```

```
      DPHI=(PHI1-(150*PIE/180))/NDIM
      PHIINT=(150*PIE/180)+0.5*DPHI
      DO 110 I=1,NDIM
      PHI=PHIINT+(I-1)*DPHI
      BETA=(RS/R2)*(PHI-PHIO)
      SY=STROKE+R2*(1.0-COS(BETA))-RS*(1.0-COS(PHI))
      SX=RS*SIN(PHI)-R2*SIN(BETA)-E
      FACTOR=((STROKE-SY)**2+SX*SX)/(STROKE**2)
      RM=RL+RS*SIN(PHI)
110   EFFWT=EFFWT+TPFAFT*FACTOR*PLY2*RM*RS*DPHI
      XMASST=PMASS4+PMASS5+EFFWT/32.147
      RETURN
C     *** OFF ATTITUDE ANALYSIS ***
      ENTRY MASS2
      IF (ANGLE.LE.(25.*PIE/180.)) GO TO 120
      IF (ANGLE.LE.(40.*PIE/180.)) GO TO 130
      EPS=0.946+0.00108*((ANGLE*180./PIE)-40.)
      GO TO 140
120   EPS=0.33+0.027*ANGLE*180./PIE
      GO TO 140
130   EPS=0.84+0.00706*((ANGLE*180./PIE)-25.)
140   CONTINUE
      XMASST=PMASS4+PMASS5+(4.*PIE*PIE*RL*RS-AF)*EPS*EWFABM
      RETURN
      END
```

APPENDIX D

```
      SUBROUTINE LOAD
      COMMON COMINT( 600)
      EQUIVALENCE ( COMINT( 477), VOL    )
      EQUIVALENCE ( COMINT( 459), ANGLE  )
      EQUIVALENCE ( COMINT( 455), FORCE  )
      EQUIVALENCE ( COMINT( 492), FRIRA  )
      EQUIVALENCE ( COMINT( 493), DISTX  )
      EQUIVALENCE ( COMINT( 494), DISTY  )
      EQUIVALENCE ( COMINT( 495), PRESS  )
      EQUIVALENCE ( COMINT( 482), RS     )
      EQUIVALENCE ( COMINT( 483), RL     )
      EQUIVALENCE ( COMINT( 446), B      )
      EQUIVALENCE ( COMINT( 490), PI     )
      EQUIVALENCE ( COMINT( 491), PA     )
      EQUIVALENCE ( COMINT( 489), HYST   )
      EQUIVALENCE ( COMINT( 472), LDSTR1 )
      EQUIVALENCE ( COMINT( 478), R1     )
      EQUIVALENCE ( COMINT( 479), R2     )
      EQUIVALENCE ( COMINT( 480), R3     )
      EQUIVALENCE ( COMINT( 449), E      )
      EQUIVALENCE ( COMINT( 450), F      )
      EQUIVALENCE ( COMINT( 481), THETAT )
      EQUIVALENCE ( COMINT( 488),  AF     )
      EQUIVALENCE ( COMINT( 496), AFS    )
      EQUIVALENCE ( COMINT( 497), SSAVE  )
      EQUIVALENCE ( COMINT( 498), FSAVE  )
      EQUIVALENCE ( COMINT( 499), VSAVE  )
      EQUIVALENCE ( COMINT( 500), KODEUP )
      EQUIVALENCE ( COMINT( 469), DTHE    )
      EQUIVALENCE ( COMINT( 501), GASCNT )
      EQUIVALENCE ( COMINT( 502), IFLAT  )
      EQUIVALENCE (COMINT( 7 ), IVARH )
      EQUIVALENCE (COMINT( 10), IFIN  )
      EQUIVALENCE ( COMINT( 348), STROKE )
      EQUIVALENCE ( COMINT( 349), SAFS   )
      EQUIVALENCE ( COMINT( 356), SSSAVE )
      EQUIVALENCE ( COMINT( 363), SFSAVE )
      EQUIVALENCE ( COMINT( 370), SVSAVE )
      EQUIVALENCE ( COMINT( 377), SKODUP )
      IF (LDSTR1.NE.0) GO TO 10
      PIE=3.1415926535898
      PRESS=PI
      TCL=0.001
      FMAX=0.0
      FSAVE=0.0
      SMAX=0.0
      SSAVE=0.0
      AFS=0.0
      VOLINT=2*PIE*PIE*RL*RS*RS
      VOL=VOLINT
      VSAVE=VOLINT
      LDSTR1=1
      KODEUP=1
      GO TO 20
10    IF (IFIN.EQ.0) GO TO 20
      IF (ABS((STROKE-SSAVE)/RS).LE.0.00001) RETURN
```

```
      GO TO 40
20    CONTINUE
      IF (ANGLE.GT.((PIE*STROKE)/(18.*RS))) GO TO 30
      IFLAT=1
      GO TO 40
30    IFLAT=0
40    CONTINUE
      COSANG=COS(ANGLE)
      TANANG=TAN(ANGLE)
      SRATIO=STROKE/RS
      IF (IFLAT.EQ.0) GO TO 60
C     *  *  *  *  *  *  *
C
C     BEGINNING OF FLAT ROUTINE.
C
C     *  *  *  *  *  *  *
      E=0.5*STROKE*(PIE/(PIE+1))
      R3=RS+E
      R2=RS-0.25*STROKE*((PIE+2)/(PIE+1))
      CONT1=RS-STROKE-B
      CONT2=(RS*(0.5*PIE-1.0)-B)/CONT1
      THETAT=0.5*PIE*(0.9538+2.9618*SRATIO*SRATIO)
50    SINT=SIN(THETAT)
      COST=COS(THETAT)
      THSAV=THETAT
      FTH=THETAT-SINT-CONT2*(1.0-COST)
      FTHD=1.0-COST-CONT2*SINT
      THETAT=THETAT-FTH/FTHD
      IF (ABS(THETAT-THSAV).GT.TOL) GO TO 50
      R1=CONT1/(1.0-COS(THETAT))
      F=RS-R1*SIN(THETAT)
      AF=PIE*(E+F)*(2*RL+E-F)
      ARK1=(0.5*PIE*R2*R2)*((4*R2)/(3*PIE)+E+RL)
      ARK2=(0.25*PIE*R3*R3)*(RL+E-(4*R3)/(3*PIE))
      ARK3=(E+F)*(RS-STROKE)*(RL-RS+R1*SIN(THETAT)+.5*(E+F))
      ARK4=-(0.25*R1*R1*SIN(2*THETAT))*(RL-RS+2.*R1*SIN(THETAT)/3.)
      ARK5=(.5*R1*R1*THETAT)*(RL-RS+R1*SIN(THETAT)-2.*R1*(1-COS(THETAT))
     1/(3.*THETAT))
      ARK6=(E*R1*SIN(THETAT))*((RL-RS)+.5*R1*SIN(THETAT))
      VOL=2*PIE*(ARK1+ARK2+ARK3+ARK4+ARK5+ARK6)
      DISTX=RS-STROKE
      DISTY=0.0
      FRIRA=RL+0.5*(E-F)
      CALL MASS1
      GO TO 200
C     *  *  *  *  *  *  *
C
C     BEGINNING OF GENERAL OFF ATTITUDE ROUTINE
C     * * * * * * *
C
60    THE=0.0
      ACHORD=0.0
      ARMOMX=0.0
      ARMOMY=0.0
      FRIRA=0.0
      KOUNT=0
```

APPENDIX D

```
70     THE=THE+DTHE
       COSTHE=COS(THE)
       TANTHE=TAN(THE)
       KOUNT=KOUNT+1
       IF (ANGLE.GT.1.56) GO TO 80
       PSI=ATAN(TANANG*COSTHE)
       SINPSI=SIN(PSI)
       COSPSI=COS(PSI)
       TANPSI=TAN(PSI)
       GO TO 120
C      *   PURE END LOADING   *
80     STTH=RL+RS-(RL+RS-STROKE)/COSTHE
       IF (STTH.LE.0.0) GO TO 110
       IF (STTH.GE.RS) GO TO 90
       CH=2.*SQRT(2.*RS*STTH-STTH**2)
       GO TO 100
90     CH=2.*RS
100    DISTY=RS+RL-STTH
       DISTZ=DISTY*SINTHE
       WD=(DISTY*DTHE)/COSTHE
       ACH=2.*CH*WD
       ACHORD=ACHORD+ACH
       FRIRA=FRIRA+SQRT((0.25*CH)**2+DISTZ**2)
       GO TO 70
110    DISTY=RS+RL-STROKE
       DISTX=0.0
       GO TO 170
C      *   OFF ATTITUDE LOADING   *
120    STTH=RS-COSPSI*(RL*TANANG*(1.0-COSTHE)+(RS-STROKE)/COSANG)
       IF (STTH.LE.0.0) GO TO 150
       IF ((THE-3.14).GT.0.0) GO TO 150
       IF (STTH.GE.RS) GO TO 130
       CH=2.*SQRT(2.*RS*STTH-STTH**2)
       GO TO 140
130    CH=2.*RS
140    DISTY=RL+(RS-STTH)*SINPSI
       DISTX=(RS-STTH)*COSPSI
       DISTZ=DISTY*SINTHE
       WD=(DISTY*DTHE*COSPSI)/COSANG
       ACH=2.*CH*WD
       ACHORD=ACHORD+ACH
       FRIRA=FRIRA+SQRT((0.25*CH)**2+DISTZ**2)
       ARMOMX=ARMOMX+ACH*DISTX
       ARMOMY=ARMOMY+ACH*DISTY
       GO TO 70
150    IF (ACHORD.LE..0001) GO TO 160
       DISTX=ARMOMX/ACHORD
       DISTY=ARMOMY/ACHORD
       GO TO 170
160    DISTX=0.0
       DISTY=0.0
170    CONTINUE
       FRIRA=FRIRA/KOUNT
       IF (SRATIO.GE.1.537) GO TO 180
       AF=ACHORD*(.499+.326*SRATIO)
       GO TO 190
```

```
180   AF=ACHORD
190   VOL=VSAVE-0.5*(STROKE-SSAVE)*(AF+AFS)
      CALL MASS2
C     *  *  *  *  *  *  *
C
C     HYSTERESIS AND UPDATE ROUTINE.
C
C     *  *  *  *  *  *  *
200   PRESS=(PI+PA)*((VOLINT/VOL)**GASCNT)-PA
      FORCE=PRESS*AF
      IF (KODEUP.EQ.0) FORCE=FORCE-HYST*FMAX*SIN(STROKE*PIE/SMAX)
      IF (IFIN.NE.0) RETURN
      SAFS=AFS
      SSSAVE=SSAVE
      SFSAVE=FSAVE
      SVSAVE=VSAVE
      SKODUP=KODEUP
      IF (KODEUP.EQ.0) GO TO 220
      IF (STROKE.LT.SSAVE) GO TO 210
      GO TO 220
210   KODEUP=0
      FMAX=FSAVE
      SMAX=SSAVE
      FORCE=FORCE-HYST*FMAX*SIN(STROKE*PIE/SMAX)
220   AFS=AF
      SSAVE=STROKE
      FSAVE=FORCE
      VSAVE=VOL
      RETURN
      END
```

APPENDIX D

```
SUBROUTINE PHYS
DIMENSION XD(7,19), XC(3), XCF(3), TFC(3)
DIMENSION XPF(3), XCD(3), FS(3)
DIMENSION XVF(3), TR(3,3), TRL(3,3), FF(19)
COMMON COMINT( 600)
EQUIVALENCE ( COMINT(   5 ), MZ      )
EQUIVALENCE (COMINT(165), XD      )
EQUIVALENCE (COMINT(344), ZLS     )
EQUIVALENCE ( COMINT( 348), STROKE )
EQUIVALENCE ( COMINT( 410), XXI    )
EQUIVALENCE ( COMINT( 411), YYI    )
EQUIVALENCE ( COMINT( 411), ZZI    )
EQUIVALENCE ( COMINT( 439), NTX    )
EQUIVALENCE ( COMINT( 440), NTY    )
EQUIVALENCE ( COMINT( 441), NTZ    )
EQUIVALENCE ( COMINT( 442), NRX    )
EQUIVALENCE ( COMINT( 443), NRY    )
EQUIVALENCE ( COMINT( 444), NRZ    )
EQUIVALENCE ( COMINT( 446), B      )
EQUIVALENCE ( COMINT( 448), D      )
EQUIVALENCE ( COMINT( 455), FORCL  )
EQUIVALENCE ( COMINT( 459), ANGLL  )
EQUIVALENCE ( COMINT( 460), ANGLE1 )
EQUIVALENCE ( COMINT( 461), ANGLE2 )
EQUIVALENCE ( COMINT( 462), ANGLE3 )
EQUIVALENCE ( COMINT( 474), RSXCF  )
EQUIVALENCE ( COMINT( 482), RS     )
EQUIVALENCE ( COMINT( 483), RL     )
EQUIVALENCE ( COMINT( 493), DISTX  )
EQUIVALENCE ( COMINT( 465), RDIA   )
EQUIVALENCE ( COMINT( 466), AMU    )
EQUIVALENCE ( COMINT( 470), ACLR   )
EQUIVALENCE ( COMINT( 472), LDSTR1 )
EQUIVALENCE ( COMINT( 490), PI     )
EQUIVALENCE ( COMINT( 492), FRIRA  )
EQUIVALENCE ( COMINT( 494), DISTY  )
EQUIVALENCE ( COMINT( 495), PRESS  )
EQUIVALENCE ( COMINT( 502), IFLAT )
EQUIVALENCE ( COMINT( 503), XVF    )
EQUIVALENCE ( COMINT( 506), TR     )
EQUIVALENCE ( COMINT( 515), TRL    )
EQUIVALENCE ( COMINT( 524), FF     )
EQUIVALENCE ( COMINT( 543), XPF    )
EQUIVALENCE ( COMINT( 546), XCD    )
EQUIVALENCE ( COMINT( 549), FS     )
EQUIVALENCE ( COMINT( 555), FIFLAT )
EQUIVALENCE ( COMINT( 561), DEGRAD )
EQUIVALENCE ( COMINT( 562), ANGLD1 )
EQUIVALENCE ( COMINT( 563), ANGLD2 )
EQUIVALENCE ( COMINT( 564), ANGLD3 )
EQUIVALENCE ( COMINT( 581), VMIN   )
EQUIVALENCE (COMINT( 591), XCF     )
CALL MASS
RETURN
ENTRY PHYS1
BETA=TRL(3,1)
```

```
      ANGLE=ACOS(ABS(TRL(3,1)))
      ANGLE1=ACOS(BETA)
      STROKE=RS+RL*SIN(ANGLE)-12.*ABS(ZLS)
      CLEAR1=RS+SQRT(RS*RS-B*B)*SIN(ANGLE)-STROKE-B*COS(ANGLE)
      CLEAR2=RS+SIN(ANGLE)*(RS+D)-STROKE-C*COS(ANGLE)/2.
      ACLR=AMIN1(CLEAR1,CLEAR2)-RDTA
      IF (ACLR.LE.0.0) RETURN
      IF (STROKE.LE.0.0) GO TO 10
      CALL LOAD
      FIFLAT=IFLAT
      IF (FORCE.GT.1.0E-05) GO TO 30
10    FORCE=0.0
      DO 20 I=1,3
      XCD(I)=0.0
      XC(I)=0.0
      FS(I)=0.0
      FF(I)=0.0
20    FF(I+3)=0.0
      PRESS=PI
      TF=0.0
      LDSTR1=0
      CALL MASS
      RETURN
30    CONTINUE
      RADUS1=DISTY*COS(ANGLE)-DISTX*SIN(ANGLE)
      CONT1=1.570796-ANGLE1
      OMEG3=0.0
      DO 40 J=1,3
40    OMEG3=TRL(3,J)*XD(1,J+3)+OMEG3
      IF (ABS(OMEG3).GT.1.0E-06) GO TO 50
      TF=0.0
      GO TO 60
50    TF=FORCE*AMU*FRIRA*(-OMEG3/ABS(OMEG3))/12.
      VMIN1=VMIN
      IF (ABS(OMEG3).GT.VMIN1) GO TO 60
      TF=TF/VMIN1*ABS(OMEG3)
60    DO 70 I=1,3
70    TFC(I)=TF*TRL(3,I)
      IF (ABS(TFC(1)).GT.ABS(XD(1,4)*XXI/HZ)) TFC(1)=-XD(1,4)*XXI/HZ*.9
      IF (ABS(TFC(2)).GT.ABS(XD(1,5)*YYI/HZ)) TFC(2)=-XD(1,5)*YYI/HZ*.9
      IF (ABS(TFC(3)).GT.ABS(XD(1,6)*ZZI/HZ)) TFC(3)=-XD(1,6)*ZZI/HZ*.9
      IF (ABS(TRL(3,2))+ABS(TRL(3,3)).GT.1.E-10) GO TO 80
      ANGLE2=1.570796326B
      GO TO 90
80    ANGLE2=ATAN2(TRL(3,3),TRL(3,2))
90    XC(1)=(DISTX/12.0)*SIGN(1.,CONT1)
      IF (ABS(ANGLE1).GT.1.E-5) GO TO 100
      R=0.0
      GO TO 110
100   R=ABS(-ZLS/SIN(ANGLE1)-XC(1)/TAN(ANGLE1))
110   XC(2)=R*COS(ANGLE2)
      XC(3)=R*SIN(ANGLE2)
      XCD(1)=XD(1,1)-XD(1,6)*XC(2)+XD(1,5)*XC(3)
      XCD(2)=XD(1,2)+XD(1,6)*XC(1)-XD(1,4)*XC(3)
      XCD(3)=XD(1,3)-XD(1,5)*XC(1)+XD(1,4)*XC(2)
      RSXCF=0.0
```

```
      DO 120 I=1,2
      XCF(I)=0.
      DO 120 J=1,3
120   XCF(I)=TRL(I,J)*XCD(J)+XCF(I)
      DO 130 I=1,2
130   RSXCF=RSXCF+XCF(I)**2
      IF ((ABS(XCF(1)).GT.1.E-3).OR.(ABS(XCF(2)).GT.1.E-3)) GO TO 140
      FS(3)=-FORCE
      FS(2)=0.0
      FS(1)=0.0
      GO TO 170
140   IF (ABS(XCF(1)).GT.1.E-6) GO TO 150
      ANGLE3=1.5707963268
      IF (XCF(2).GT.0.0) ANGLE3=4.7123889804
      GO TO 160
150   ANGLE3=ATAN2(-XCF(2),-XCF(1))
160   FS(3)=-FORCE
      FS(2)=AMU*FORCE*SIN(ANGLE3)
      FS(1)=AMU*FORCE*COS(ANGLE3)
      RSXCF=RSXCF**.5
      IF (RSXCF.GT.VMIN) GO TO 170
      FS(2)=FS(2)/VMIN*RSXCF
      FS(1)=FS(1)/VMIN*RSXCF
170   CONTINUE
      IF (NTX.EQ.0) FF(1)=FS(1)*TRL(1,1)+FS(2)*TRL(2,1)+FS(3)*TRL(3,1)
      IF (NTY.EQ.0) FF(2)=FS(1)*TRL(1,2)+FS(2)*TRL(2,2)+FS(3)*TRL(3,2)
      IF (NTZ.EQ.0) FF(3)=FS(1)*TRL(1,3)+FS(2)*TRL(2,3)+FS(3)*TRL(3,3)
      XPF(1)=(-ZLS*COS(ANGLE)-RADUS1*SIN(ANGLE)/12.)*SIGN(1.,CONT1)
      IF (ABS(ANGLE).GT.1.E-10) GO TO 180
      XPF(2)=0.0
      XPF(3)=0.0
      GO TO 190
180   XPF(2)=(-ZLS*SIN(ANGLE)+RADUS1*COS(ANGLE)/12.)*COS(ANGLE2)
      XPF(3)=(-ZLS*SIN(ANGLE)+RADUS1*COS(ANGLE)/12.)*SIN(ANGLE2)
190   IF (NRX.EQ.0) FF(4)=-FF(2)*XPF(3)+FF(3)*XPF(2)+TFC(1)
      IF (NRY.EQ.0) FF(5)=FF(1)*XPF(3)-FF(3)*XPF(1)+TFC(2)
      IF (NRZ.EQ.0) FF(6)=-FF(1)*XPF(2)+FF(2)*XPF(1)+TFC(3)
      ANGLD1=ANGLE1*DEGRAD
      ANGLD2=ANGLE2*DEGRAD
      ANGLD3=ANGLE3*DEGRAD
      RETURN
      END
```

APPENDIX D

```
      SUBROUTINE SOLVE (A,G,SUM,N,M)
      DIMENSION A(3,4),G(3,3),H(3,3)
C     THIS IS A SUBROUTINE FOR DETERMINING THE VALUE OF C IN THE
C     MATRIX EQUATION A*C=G.  THE VALUES OF A AND G ARE PROVIDED
C     AT THE TIME OF CALLING.  N IS THE ORDER OF A (MUST BE
C     A SQUARE MATRIX) AND M IS THE NUMBER OF COLUMNS IN
C     G(A AND G MUST HAVE SAME NUMBER OF ROWS).  THE VALUE
C     OF C IS STORED IN G LOCATION AT RETURN.  IF THE INVERSE
C     OF A IS REQUIRED, MAKE M NEGATIVE.  IF ONLY THE INVERSE
C     OF A IS REQUIRED(G MATRIX DOES NOT EXIST), ENTER M=0.
C     DETERMINANT OF A IS STORED IN LOCATION SUM AT RETURN.
C     IF THE INVERSE OF A IS COMPUTED, IT IS STORED
C     IN LOCATION A AT RETURN TO THE CALLING PROGRAM.
C     IF IT IS DESIRED TO MAKE THIS A DOUBLE PRECISION
C     SUBROUTINE, THE FOLLOWING VARIABLES MUST BE TYPED
C     DOUBLE PRECISION  A,G,H, AND SUM.
C     WHEN PROVIDING DIMENSIONING INFORMATION FOR
C     THE VARIABLE A ,( A(I,J) ), THE VALUE OF J MUST
C     BE ONE GREATER THAN I, IE., J=I+1.
      NGO=1
      NST=1
      IF (M) 10,20,50
10    M=-M
      NST=2
20    NGO=2
      DO 40 I=1,N
      DO 40 J=1,N
      H(I,J)=0.
      IF (I-J) 40,30,40
30    H(I,J)=1.
40    CONTINUE
50    N1=N+1
      N2=N-1
      KPT=2
      DO 300 IP=1,NST
      IF (M) 70,60,70
60    IP=2
70    GO TO (80,90), IP
80    NSP=M
      GO TO 100
90    NSP=N
100   DO 300 JP=1,NSP
      DO 130 I=1,N
      GO TO (110,120), IP
110   A(I,N1)=G(I,JP)
      GO TO 130
120   A(I,N1)=H(I,JP)
130   CONTINUE
      DO 140 I=KPT,N1
140   A(1,I)=A(1,I)/A(1,1)
      DO 200 I=2,N
      NN=I-1
      DO 200 J=KPT,N1
      NM=J-1
      SUM=0.
      IF (NM-NN) 150,150,160
```

```
150   KK=NM
      GO TO 170
160   KK=NN
170   DO 180 L=1,KK
180   SUM=SUM+A(L,J)*A(I,L)
      A(I,J)=A(I,J)-SUM
      IF (J-I) 200,200,190
190   A(I,J)=A(I,J)/A(I,I)
200   CONTINUE
      GO TO (210,220), IP
210   G(N,JP)=A(N,N1)
      GO TO 230
220   H(N,JP)=A(N,N1)
230   DO 290 I=1,N2
      NI=N-I
      GO TO (240,250), IP
240   G(NI,JP)=A(NI,N1)
      GO TO 260
250   H(NI,JP)=A(NI,N1)
260   DO 290 J=1,I
      NJ=N1-J
      GO TO (270,280), IP
270   G(NI,JP)=G(NI,JP)-A(NI,NJ)*G(NJ,JP)
      GO TO 290
280   H(NI,JP)=H(NI,JP)-A(NI,NJ)*H(NJ,JP)
290   CONTINUE
      KPT=N1
300   CONTINUE
      SUM=1.
      DO 310 I=1,N
310   SUM=SUM*A(I,I)
      GO TO (340,320), NGO
320   DO 330 I=1,N
      DO 330 J=1,N
330   A(I,J)=H(I,J)
340   RETURN
      END
```

```
      SUBROUTINE PRINT (VN,LC,LP)                                       PRT  10
      DIMENSION TITLEG(72), ISUBCG(72), PNTLST(72)                      PRT  20
      COMMON COMINT( 600)                                               PRT  30
      EQUIVALENCE ( COMINT( 473), IERPRT )                              PRT  40
      EQUIVALENCE ( COMINT(   1), T      )                              PRT  50
      EQUIVALENCE ( COMINT(   5), HZ     )                              PRT  60
      DATA ISUBCG  /                                                    PRT  70
     1     428, 430, 433, 562, 165, 413, 186, 319, 348, 394, 463, 493,  PRT  80
     2     429, 431, 434, 563, 172, 414, 193, 326, 455, 395, 477, 494,  PRT  90
     3     344, 432, 435, 564, 179, 415, 200, 333, 495, 396, 474, 492,  PRT 100
     4     572, 575, 578, 569, 524, 527, 549, 543, 591, 470, 478, 449,  PRT 110
     5     573, 576, 579, 570, 525, 528, 550, 544, 592, 416, 479, 450,  PRT 120
     6     574, 577, 580, 571, 526, 529, 551, 545, 593, 555, 480, 481  /PRT 130
      DATA TITLEG /                                                     PRT 140
     1 10H    XLS   , 10H   XDLS   , 10H   XDDLS  , 10H  ANGLD1  ,      PRT 150
     2 10H    XD    , 10H    XN    , 10H   PHID   , 10H  PHIDD   ,      PRT 160
     3 10H  STROKE  , 10H    XAG   , 10H    AF    , 10H  DISTX   ,      PRT 170
     4 10H    YLS   , 10H   YDLS   , 10H   YDDLS  , 10H  ANGLD2  ,      PRT 180
     5 10H    YD    , 10H    YN    , 10H  THETAD  , 10H  THETADD ,      PRT 190
     6 10H   FORCE  , 10H    YAG   , 10H    VOL   , 10H  DISTY   ,      PRT 200
     7 10H    ZLS   , 10H   ZDLS   , 10H   ZDDLS  , 10H  ANGLD3  ,      PRT 210
     8 10H    ZD    , 10H    ZN    , 10H   PSID   , 10H  PSIDD   ,      PRT 220
     9 10H   PRESS  , 10H    ZAG   , 10H   KSXCF  , 10H  FKIRA   ,      PRT 230
     $ 10H ANG(1,1) , 10H ANG(1,2) , 10H ANG(1,3) , 10H  XLPF(1) ,      PRT 240
     $ 10H   FF(1)  , 10H   FF(4)  , 10H   FS(1)  , 10H  XPF(1)  ,      PRT 250
     $ 10H  XCF(1)  , 10H   ACLR   , 10H    R1    , 10H    E     ,      PRT 260
     $ 10H ANG(2,1) , 10H ANG(2,2) , 10H ANG(2,3) , 10H  XLPF(2) ,      PRT 270
     $ 10H   FF(2)  , 10H   FF(5)  , 10H   FS(2)  , 10H  XPF(2)  ,      PRT 280
     $ 10H  XCF(2)  , 10H  XMASST  , 10H    R2    , 10H    F     ,      PRT 290
     $ 10H ANG(3,1) , 10H ANG(3,2) , 10H ANG(3,3) , 10H  XLPF(3) ,      PRT 300
     $ 10H   FF(3)  , 10H   FF(6)  , 10H   FS(3)  , 10H  XPF(3)  ,      PRT 310
     $ 10H  XCF(3)  , 10H  FIFLAT  , 10H    R3    , 10H  THETAT  /      PRT 320
      IF ((LP.LE.72).AND.(LP.GT.0)) GO TO 10                            PRT 330
      WRITE (6,100) LP                                                  PRT 340
      STOP                                                              PRT 350
10    TITLEG(LP)=VN                                                     PRT 360
      IF (LC.GT.0) GO TO 20                                             PRT 370
      WRITE (6,110) LC                                                  PRT 380
      STOP                                                              PRT 390
20    ISUBCG(LP)=LC                                                     PRT 400
      RETURN                                                            PRT 410
      ENTRY PRINT1                                                      PRT 420
      IF (IERPRT.NE.0) GO TO 30                                         PRT 430
      ICONTP=10                                                         PRT 440
      IPRINT=10                                                         PRT 450
      GO TO 40                                                          PRT 460
30    ICONTP=6                                                          PRT 470
      IPRINT=6                                                          PRT 480
40    RETURN                                                            PRT 490
      ENTRY PRINT2                                                      PRT 500
      CALL SECOND (TIMECP)                                              PRT 510
      IF (IPRINT.NE.ICONTP) GO TO 60                                    PRT 520
      IPRINT=0                                                          PRT 530
      WRITE (6,120)                                                     PRT 540
      WRITE (6,130) (TITLEG(I),I=1,36)                                  PRT 550
      IF (IERPRT.EQ.0) GO TO 50                                         PRT 560
```

```
      WRITE (6,130) (TITLEG(I),I=37,72)
50    CONTINUE
60    WRITE (6,150) TIMECP,T,HZ
      DO 70 I=1,36
      LKI=ISUBCG(I)
70    PNTLST(I)=COMINT(LKI)
      WRITE (6,140) (PNTLST(I),I=1,36)
      IF (IERPRT.EQ.0) GO TO 90
      DO 80 I=37,72
      LKI=ISUBCG(I)
80    PNTLST(I)=COMINT(LKI)
      WRITE (6,140) (PNTLST(I),I=37,72)
90    CONTINUE
      IPRINT=IPRINT+1
      RETURN
C
  100 FORMAT (* *,*-----------ERROR----------LPRINT = *,I5,*, RANGE IS ON
     *LY 1-72* )
  110 FORMAT (* *,*-----------ERROR----------LCOMON = *,I5,*, RANGE .GT.
     *0 * )
  120 FORMAT (*1*,6X*LOCAL SURFACE COORDINATE SYSTEM*,
     *              15X* LANDER COORDINATE SYSTEM* )
130   FORMAT (4(1XA10),1H*,A10,3(1XA10),1H*,A10,3(1XA10))
140   FORMAT (12(1XE10.3))
  150 FORMAT (*0*,*CP TIME  = *,F10.3,* T        = *,E10.3,
     1 * DT       = *,E10.3)
      END
```

APPENDIX E

PROGRAM VARIABLES

LOCATED IN

COMMON ARRAY "COMINT"

APPENDIX E

A brief definition, program name, analysis symbol, and location of all the variables located in the common array COMINT are given in the following figure. This array appears in both the Crushable Torus Landing Loads and Motions Program (Appendix B) and the Inflatable Torus Landing Loads and Motions Program (Appendix D). A great number of these variables apply to both programs; however, some apply to only one or the other of the programs. Applicability of each variable is indicated in this figure.

The array COMINT is used for the transfer of information between the various program subroutines. Also, this array is useful when used in conjunction with the optional output routine available in both motions programs. As described in the two program operating instructions (Sections B.3 and D.3), any variable appearing in COMINT may be printed with the standard time history quantities.

PROGRAM VARIABLES IN COMMON ARRAY "COMINT"

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
T	1	Both	t	Time
HMAX	2	Both	Δt_{max}	Maximum Integration Step Size
EMIN	3	Both	–	Minimum Integration Accuracy Required (Predictor-Corrector)
EMAX	4	Both	–	Maximum Integration Accuracy Required (Predictor-Corrector)
HZ	5	Both	Δt	Integration Step Size
IP	6	Both	–	Quantity Used to Determine Initial Integration Time Interval
IVARH	7	Both	–	Variable or Fixed Step Integration Indicator.
IMTH	8	Both	–	Runge-Kutta or Predictor-Corrector Integration Indicator
IPRNT	9	Both	–	Print Indicator; Ready to Print When 0.
IFIN	10	Both	–	Integration Internal Indicator; End of Integration Step When 0.
IVAL	11	Both	–	Integration Indicator IVAL = "+" – Valid Completion of Integration Step. IVAL = "–" – Integration Step Size Halved. IVAL = 0 Three Times in Row – Internal Doubled.
IPTOTL	12	Both	–	Number of Integration Variables.
IPTATL	13	Both	–	Number of Cutoff Variables.
XS	14–22	Both	–	Cutoff Values of Cutoff Variables.
IND	23–31	Both	–	Direction Indicators for Cutoff Variables.
X	32–164	Both	–	Integration Variables as Follows: X(1,1), X(1,2), X(1,3) – X,Y, & Z Displacements in Lander Coordinates X(1,4), X(1,5), X(1,6) – Angular Displacements About X,Y, & Z Lander Coordinates X(1,7) Through X(1,15)–Values Used in Computing TR (I, J) X(1,16), X(1,17), X(1,18)–X,Y,&Z Displacements in Gravity Coordinates X(1,19) – RANGE X(2,1) Through X(7,19) – Storage Space Required for Above Variables in Integration Routine.
XD	165–297	Both	–	Integration Variables as Follows: XD(1,1), XD (1,2), XD (1,3) – X,Y,&Z Velocities in Lander Coordinates XD (1,4), XD (1,5), XD (1,6) – Angular Velocities About X, Y, & Z Lander Coordinates

Figure E-1

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
XDD	298–339	Both	–	XD (1,7) Through X(1,15) – Values Used in Computing TR(I,J) XD(1,16),XD(1,17),XD(1,18)–X,Y,&Z Velocities in Gravity Coordinates XD (2,1) Through XD(7,18)–Storage Space Required for Above Variables in Integration Routine. Integration Variables as Follows: XDD(1,1),XDD(1,2),XDD(1,3)–X,Y,&Z Accelerations in Lander Coordinates XDD(1,4),XDD(1,5),XDD(1,6)–Angular Accelerations About X,Y,&Z Lander Coordinates XDD(2,1) Through XDD(7,6)–Storage Space Required for Above Variables in Integration Routine.
TOTSV	340	Both	–	Total Surface Velocity of Lander c.g.
PSIL	341	Both	–	Yaw Angle with Respect to Local Surface Coordinates
TLSV	342	Both	–	Resultant Local Surface Straight Line Distance From Reference Point.
	343			Location Not Used.
ZLS	344	Both	–	Position of Lander c.g. in Z_{ls} Ditection
PHIL	345	Both	–	Roll Angle with Respect to Local Surface Coordinates.
THETAL	346	Both	–	Pitch Angle with Respect to Local Surface Coordinates
HMIN	347	Both	Δt_{min}	Minimum Integration Step Size
STROKE	348	Both	S	Attenuator Stroke
SAFS	349–355	Inflatable	–	Past Values of Footprint Area
SPELST	349–355	Crushable	–	Past Values of ELST.
SSSAVE	356–362	Inflatable	–	Past Values of Stroke.
SPSTTM	356–362	Crushable	–	Past Values of Stroke.
SFSAVE	363–369	Inflatable	–	Past Values of Force
SAVST2	363–369	Crushable	–	Past Values of SAVSTB.
SVSAVE	370–376	Inflatable	–	Past Values of Torus Volume
SAVST1	370–376	Crushable	–	Past Values of SAVSTA.
SKODUP	377–383	Inflatable	–	Past Values of Indicator KODEUP
CUTERR	384	Both	–	Error Tolerance to Determine Point of Cutoff
JCUT	385	Both	–	Cutoff Indicator
XF	386	Both	X_f	X Displacement in Gravity Coordinate (X(1,16)).

Figure E-1 (Continued)

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
YF	387	Both	Y_f	Y Displacement in Gravity Coordinate (X (1,17)).
ZF	388	Both	Z_f	Z Displacement in Gravity Coordinate (X (1,18)).
XW	389	(1)	–	X Surface Component of Wind Velocity
YW	390	(1)	–	Y Surface Component of Wind Velocity
ZW	391	(1)	–	Z Surface Component of Wind Velocity
SCFWT	392	Inflatable	–	Weight per Unit Area of Torus Scuff Material.
GZ	393	Both	g	Planet's Acceleration of Gravity
XAG	394	Both	g_x	Component of Acceleration of Gravity in Lander X Axis.
YAG	395	Both	g_y	Component of Acceleration of Gravity in Lander Y Axis.
ZAG	396	Both	g_z	Component of Acceleration of Gravity in Lander Z Axis.
DENS	397	(1)	–	Atmospheric Density
XWND	398	(1)	–	X Lander Component of Wind Velocity
YWND	399	(1)	–	Y Lander Component of Wind Velocity
ZWND	400	(1)	–	Z Lander Component of Wind Velocity
ALPHA	401	(1)	–	Angle of Attack
GDOT	402	(1)	–	Time Derivative of Vertical Flight Path Angle
SDOT	403	(1)	–	Time Derivative of Horizontal Flight Path Angle
BETA	404	(1)	–	Sideslip Angle
ALT	405	Both	–	Altitude Above Planet's Surface.
RANGE	406	Both	–	Distance Traveled Over Planet Surface (Arc Length)
THETA	407	Both	θ	Pitch Angle
PSI	408	Both	ψ	Yaw Angle
PHI	409	Both	ϕ	Roll Angle
XXI	410	Both	I_{xx}	Lander Mass Moment of Inertia
YYI	411	Both	I_{yy}	Lander Mass Moment of Inertia
ZZI	412	Both	I_{zz}	Lander Mass Moment of Inertia
XN	413	Both	N_x	Lander Load Factor in X Direction
YN	414	Both	N_y	Lander Load Factor in Y Direction
ZN	415	Both	N_z	Lander Load Factor in Z Direction
XMASST	416	Both	m	Lander Mass
THRST	417	(1)	–	Thrust Level
XYI	418	Both	I_{xy}	Lander Product of Inertia
XZI	419	Both	I_{xz}	Lander Product of Inertia
YZI	420	Both	I_{yz}	Lander Product of Inertia

Figure E-1 (Continued)

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)				
PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
DELTA	421	(1)	–	Thrust Deflection Angle
XXID	422	Both	$\dot{I}_{xx}$	Time Derivative of Lander Mass Moment of Inertia
YYID	423	Both	$\dot{I}_{yy}$	Time Derivative of Lander Mass Moment of Inertia
ZZID	424	Both	$\dot{I}_{zz}$	Time Derivative of Lander Mass Moment of Inertia
XYID	425	Both	$\dot{I}_{xy}$	Time Derivative of Lander Product of Inertia
XZID	426	Both	$\dot{I}_{xz}$	Time Derivative of Lander Product of Inertia
YZID	427	Both	$\dot{I}_{yz}$	Time Derivative of Lander Product of Inertia
XLS	428	Both	Xl_s	Position of Lander c.g. with Respect to Local Surface X Axis
YLS	429	Both	Yl_s	Position of Lander c.g. with Respect to Local Surface Y Axis.
XDLS	430	Both	$\dot{X}l_s$	Velocity of Lander c.g. with Respect to Local Surface X Axis
YDLS	431	Both	$\dot{Y}l_s$	Velocity of Lander c.g. with Respect to Local Surface Y Axis
ZDLS	432	Both	$\dot{Z}l_s$	Velocity of Lander c.g. with Respect to Local Surface Z Axis
XDDLS	433	Both	$\ddot{X}l_s$	Acceleration of Lander c.g. with Respect to Local Surface X Axis
XDDLS	434	Both	$\ddot{Y}l_s$	Acceleration of Lander c.g. with Respect to Local Surface Y Axis
ZDDLS	435	Both	$\ddot{Z}l_s$	Acceleration of Lander c.g. with Respect to Local Surface Z Axis
XLGRAV	436	Both	–	Component of Acceleration of Gravity in Local Surface X Coordinate
YLGRAV	437	Both	–	Component of Acceleration of Gravity in Local Surface Y Coordinate
ZLGRAV	438	Both	–	Component of Acceleration of Gravity in Local Surface Z Coordinate
NTX	439	Both	–	Indicator to Surpress Degree of Freedom Along Lander X Axis
NTY	440	Both	–	Indicator to Surpress Degree of Freedom Along Lander Y Axis
NTZ	441	Both	–	Indicator to Surpress Degree of Freedom Along Lander Z Axis
NRX	442	Both	–	Indicator to Surpress Degree of Freedom About Lander X Axis (Roll)
NRY	443	Both	–	Indicator to Surpress Degree of Freedom About Lander Y Axis (Pitch)
NRZ	444	Both	–	Indicator to Surpress Degree of Freedom About Lander Z Axis (Yaw)
A	445	Inflatable	A	Payload Geometric Dimension
A	445	Crushable	A	Attenuator Geometric Dimension
B	446	Both	B	Payload Geometric Dimension
C	447	Inflatable	C	Payload Geometric Dimension
C	447	Crushable	C	Attenuator Geometric Dimension
D	448	Inflatable	D	Payload Geometric Dimension
D	448	Crushable	D	Attenuator Geometric Dimension
E	449	Inflatable	E	Torus Deflection Parameter – Flat Landing
E	449	Crushable	E	Attenuator Geometric Dimension
F	450	Inflatable	F	Torus Deflection Parameter – Flat Landing

Figure E-1 (Continued)

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
F	450	Crushable	F	Attenuator Geometric Dimension
G	451	Crushable	G	Payload Geometric Dimension
RHOL	452	Crushable	–	Attenuator Material Density
RHOP	453	Both	–	Payload Package Density
TOTGV	454	Both	–	Resultant Local Surface Velocity of Lander c.g.
FONTT	455	Crushable	F	Force Normal to Local Surface Due to Compressing Attenuator
FORCE	455	Both	F	Force Normal to Local Surface Due to Compressing Attenuator
TORTN	456	Crushable	–	Torque at Lander c.g. Due to Normal Force
ARTSU	457	Crushable	–	Total Crushed Footprint Area
PSTTM	458	Crushable	–	Previous Value of Stroke
ANGLE	459	Both	β	Absolute Value of ANGLE 1
ANGLE 1	460	Both	–	Angle Between Y-Z Lander Plane and Plane of Local Surface
ANGLE 2	461	Both	–	Angle in Y-Z Lander Plane Locating Normal Force
ANGLE 3	462	Both	–	Angle Between Friction Force and Local Surface X Axis.
ARMOM	463	Crushable	–	Moment of Footprint Area with Respect to Lander c.g.
ELST	464	Crushable	–	Elastic Stroke Recovery
RDIA	465	Both	D_R	Surface Rock Diameter
AMU	466	Both	μ	Coefficient of Friction
XKW	467	Crushable	–	Attenuator Specific Energy
DX	468	Crushable	ΔX	Incremental Value of X Used in the LDSTR Subroutine
DTHE	469	Both	$\Delta\theta$	Incremental Value of θ Used in the LDSTR or LOAD Subroutines
ACLR	470	Both	–	Lander Clearance
RNORM	471	Crushable	–	Ratio of Transverse Crush Strength to Radial Crush Strength of Attenuator
LDSTRI	472	Both	–	Indicator Used in LDSTR or LOAD Subroutines
IERPRT	473	Both	–	Print Format Indicator
RSXCF	474	Both	–	Velocity of Attenuator Contact Point Relative to Local Surface
PELST	475	Crushable	–	Previous Valve of ELST
SLOPE	476	Both	α	Ground Slope
VOL	477	Inflatable	V	Internal Volume of Torus
R1	478	Inflatable	R_1	Torus Deflection Parameter – Flat Landing
R2	479	Inflatable	R_2	Torus Deflection Parameter – Flat Landing
R3	480	Inflatable	R_3	Torus Deflection Parameter – Flat Landing
THETAT	481	Inflatable	θ_t	Torus Deflection Parameter – Flat Landing

Figure E-1 (Continued)

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
RS	482	Inflatable	R_S	Radius of Inflatable Torus
RL	483	Inflatable	R_L	Radius of Lander
FABWT	484	Inflatable	–	Torus Material Weight per Unit Area for One Ply
PLY 1	485	Inflatable	PLY1	Number of Torus Material Plys on Inner 120 Degrees
PLY 2	486	Inflatable	PLY2	Number of Torus Material Plys on Upper and Lower 60 Degrees
PLY 3	487	Inflatable	PLY3	Number of Torus Material Plys on Outer 120 Degrees
AF	488	Inflatable	A_f	Footprint Area
HYST	489	Inflatable	HYST	Torus Hysteresis Factor
PI	490	Inflatable	P_i	Initial Torus Inflation Pressure – Gauge
PA	491	Inflatable	P_a	Atmospheric Pressure
FRIRA	492	Inflatable	R_f	Radius Defining Torque Due to Friction During Spinning.
DISTX	493	Inflatable	D_x	Location of Normal Force Relative to Lander c.g.
DISTY	494	Inflatable	D_y	Location of Normal Force Relative to Lander c.g.
PRESS	495	Inflatable	P	Internal Torus Pressure – Gauge
AFS	496	Inflatable	–	Previous Value of Footprint Area
SSAVE	497	Inflatable	–	Previous Value of Stroke
FSAVE	498	Inflatable	–	Previous Value of Normal Force
VSAVE	499	Inflatable	–	Previous Value of Torus Volume
KODEUP	500	Inflatable	–	Indicator to Determine Direction of Stroking Process
GASCNT	501	Inflatable	n	Gas Constant Which Yields Assumed Gas Compression Process.
IFLAT	502	Inflatable	–	Indicator to Distinguish Between Flat and Oblique Landing
XVF	503–505	Both	–	X,Y,Z Components of Velocity in Gravity Coordinates
TR	506–514	Both	–	Direction Cosines Relating Lander Coordinates to Gravity Coordinates
TRL	515–523	Both	–	Direction Cosines Relating Lander Coordinates to Local Surface Coordinates
FF	524–542	Both	–	FF(1),FF(2), FF(3) – Forces at Lander c.g. in X,Y, & Z Axes FF(4), FF(5), FF(6), – Moments at Lander c.g. About X, Y, and Z Axes.
XPF	543–545	Both	–	Location of Normal Force in Lander Coordinates
XCD	546–548	Inflatable	–	Velocity of Footprint Area Centroid in Lander Coordinates
FS	549–551	Both	–	Friction Forces in Local Surface Coordinates
Q	552	(1)	–	Dynamic Pressure
XMACH	553	(1)	–	Mach Number
SOUND	554	(1)	–	Speed of Sound
FIFLAT	555	Inflatable	–	Floating Point Valve of IFLAT Required for Print
MMIC	556	Both	–	Variable Mass Moment of Inertia Indicator
SIGMA	557	Both	–	Horizontal Flight Path Angle (Heading From North)

Figure E-1 (Continued)

PROGRAM VARIABLES IN COMMON ARRAY "COMINT" (Continued)

PROGRAM VARIABLE	COMMON LOCATION	APPLICABLE PROGRAM	ANALYSIS SYMBOL	VARIABLE DEFINITION
GAMA	558	Both	–	Vertical Flight Path Angle
SAVSTB	559	Crushable	–	Stored Value of Stroke Preceeding SAVSTA
SAVSTA	560	Crushable	–	Stored Value of Stroke from Last Integration Point.
DEGRAD	561	Both	–	Conversion Factor from Radians to Degrees
ANGLD1	562	Both	–	ANGLE 1 in Degrees.
ANGLD2	563	Both	–	ANGLE 2 in Degrees.
ANGLD3	564	Both	–	ANGLE 3 in Degrees.
PLTMAS	565	Both	–	Planet Mass
PLTRAD	566	Both	–	Planet Radius
NMBNCS	567	Both	–	Number of Bounces Required for a Particular Data Set
KOUNT2	568	Both	–	Number of Integration Intervals Between Print Times.
XLPF	569–571	Inflatable	–	Location of Normal Force in Local Surface Coordinates
ANG	572–580	Inflatable	–	Direction Angles Obtained from TRL.
VMIN	581	Both	–	Footprint Area Sliding Velocity Below Which Coefficient of Friction Decreases to Zero.
GACC	582–584	Crushable	–	Lander c.g Translational Accelerations in Gravity Coordinate System
STREF	585	Crushable	–	Stroke Efficiency
TFC	586–588	Both	–	Friction Torques About Lander Axes.
RAV	589	Crushable	–	Radius Defining Torque due to Friction During Spinning
FRICT	590	Crushable	–	Resultant Friction Force on Attenuator Footprint
XCF	591–593	Crushable	–	Velocity of Footprint Area Centroid in Lander Coordinates
POW	594	Crushable	–	Shear and Radial Stress Interaction Exponent.
RSHCR	595	Crushable	–	Ratio of Allowable Shearing Stress to Radial Crushing Stress.

Notes: 1. Variables not used in this study.
2. In the column "Applicable Program" the following hold:
   a. Crushable – Variable applies to only the crushable torus lander.
   b. Inflatable – Variable applies to only the inflatable torus lander.
   c. Both – Variable applies to both the crushable and inflatable landers.

Figure E-1 (Continued)

APPENDIX F

ACCURACY INDICATOR

FOR VARIABLE STEP PREDICTOR -

CORRECTOR INTEGRATION METHOD

APPENDIX F

The numerical integration subroutine PCCUT, contains the option for a modified Adams-Moulton variable step Predictor-Corrector method. The following presents a brief discussion of the integration step size control in this variable step routine.

During an integration step, the following pair of parameters are determined for each of the integrated variables being considered in a particular case.

$$D_n^i = \left| P_n^i - C_n^i \right|$$

and

$$R_n^i = \left| \frac{P_n^i - C_n^i}{C_n^i} \right|$$

In the above:

1) P_n^i is the predicted value of the ith variable for the nth step of the integration routine.
2) C_n^i is the corrected value of the ith variable for the nth step of the integration routine.

The smallest value of these two parameters (D_n^i and R_n^i) for each integration variable is retained. The maximum value of all these remaining parameters is defined as the integration accuracy indicator, En. This accuracy indicator is compared with the two input constants, E_{min} (EMIN) and E_{max} (EMAX). The integration step size is then controlled in the following manner:

APPENDIX F

1) If $E_n \leq E_{min}$ for four consecutive integration steps, the step size (HZ) is doubled and the integration restarted at the end of the last integration step. However, if the step size is equal to the input value for the maximum step size (HMAX), it remains the same.

2) If $E_{min} < E_n < E_{max}$, the integration step size remains unchanged.

3) If $E_n \geq E_{max}$, the step size is halved and the last integration point satisfying the specified degree of accuracy is used to restart the integration. However, if the step size is equal to the input value for the minimum step size (HMIN), it remains the same.

If the two input parameters, EMIN and EMAX, are read into the program as zero, they are reset to the following nominal values.

$$EMAX = 1 \times 10^{-4}$$

$$EMIN = 1 \times 10^{-6}$$

Experience has shown that these nominal values result in the satisfactory operation of the variable step routine. However, increasing the magnitudes of these parameters will decrease the computer time required for a given case. Decreasing the magnitude of E_{min} and E_{max}, increases the computer run time.