NASA Contractor Report 185171

A Trajectory Generation and System Characterization Model for Cislunar Low-Thrust Spacecraft

Volume I—User's Manual

David J. Korsmeyer, Elfego Pinon III,
Brendan M. O'Connor, and Curt R. Bilby

Large Scale Programs Institute
Austin, Texas

(NASA-CR-185171) A TRAJECTORY GENERATION AND SYSTEM CHARACTERIZATION MODEL FOR CISLUNAR LOW-THRUST SPACECRAFT. VOLUME 1: USER'S MANUAL Final Report (Diskette Supplement) (Large Scale Programs Inst.) 53 p

N93-24536

Unclas

G3/20 0159803

May 1990

Prepared for
Lewis Research Center
Under Grant NAG3-928

National Aeronautics and
Space Administration

TABLE OF CONTENTS

I. THE LOWTHRST COMPUTER PROGRAM

The LOWTHRST computer program was developed at the Large Scale Programs Institute (LSPI) in Austin, Texas during 1989. The program was designed with two goals in mind. First, to provide NASA with an analysis tool for evaluating the impacts of various technologies on low-thrust cislunar spacecraft. Second, to allow some concepts and theories concerning the guidance and control of low-thrust spacecraft between the Earth and the Moon to be fully developed. Both of these goals have been met to a degree with the program LOWTHRST.

The program can be utilized to generate spacecraft system sizing data for cislunar orbital transfer vehicles based on a payload size and a propulsion and power system technology choice. This will be used to develop the sizes and masses of the rest of the spacecraft's supporting systems.

The spacecraft sizing portion of the program can also be bypassed to allow trajectory generation between the Earth and the Moon based on idealized or specific spacecraft characteristics. These characteristics include the total spacecraft mass, propulsion system propellant usage, spacecraft exhaust velocity, and final spacecraft mass. System sizing data and the trajectory data is saved after each run to allow additional analysis to be performed.

The operation and theory of the LOWTHRST computer program is compiled in two volumes. This portion is Volume I - User's Manual, and it covers the basic operation of the program, some of the core background concepts, and an example run of the program that showcases the programs capabilities. Volume II - Technical Manual is a more extensive analysis of the program methodology and solution formulation. The present state of the research on cislunar low-thrust spacecraft guidance and control is outlined and the algorithms and methods utilized in LOWTHRST are explained.

II. USING LOWTHRST

LOWTHRST is a stand alone program designed to size and characterize the systems of a low-thrust spacecraft and to generate it's transfer trajectory between orbits about the Earth and the Moon. The program allows the creation of functional trajectories dependent upon the generated and user-supplied spacecraft characteristics. The trajectory generation is a user iterative process, with the intent that the program user will modify the necessary control values until a satisfactory trajectory has been created.

LOWTHRST can be run on IBM AT's or compatibles with a graphics display without recompilation. The program is written in Microsoft FORTRAN 5.0 for MS-DOS. Two versions of the program's source code are provided on the distribution disk to allow recompiling the program on other computer systems. The first source code file, LOWTHRST.MSF, contains the program code with Microsoft extensions to support the graphical output of PC's. The second source code file, LOWTHRST.FOR, contains only standard FORTRAN 77 code without Microsoft language extensions. This means the program will not support any graphical output devices without additional routines or third party products. LOWTHRST can be easily installed on a PC with a Hard Disk by simply copying the file LOWTHRST.EXE from the distribution disk to a subdirectory on the Hard Disk, for example, C:\LT.

The program is conceptually broken into two models. The first model is the vehicle system sizing model. The second is the trajectory generation model. Both of these models are closely coupled in the LOWTHRST program. The vehicle system sizing model will be run first to allow the characterization of the spacecraft. Then the trajectory generation model will run to create the cislunar transfer trajectory.

The program can be started by typing "LOWTHRST" at command prompt, C:\LT>. After starting the program, a title screen appears indicating that LOWTHRST has started. The spacecraft characteristics can be modified or generated by the user at the beginning of each program run. LOWTHRST initially prompts the user to run the vehicle system sizing model or to enter customized spacecraft characteristics. If the system sizing model is run, the user is prompted to size a spacecraft system for a one-or two-way flight. Next, the program prompts the user to enter the payload masses to be delivered, and returned if applicable. The user is then prompted to choose one of five possible technologies for low-thrust propulsion systems or enter the characteristics of a user-defined propulsion system. The predefined technologies are:

1) Ion Thrusters, 50 cm diameter (Xenon Propellant),
2) Ion Thrusters, 50 cm diameter (Krypton Propellant),
3) Ion Thrusters, 50 cm diameter (Argon Propellant),
4) Applied-Field Magnetoplasmadynamic (MPD) Thrusters (Hydrogen Propellant), and
5) Arcjet (Hydrogen Propellant).

If the user selects one of the five predefined technologies the next questions prompt for the desired specific impulse, number of thrusters used on the spacecraft, and the power input to each thruster. The program provides ranges of values for which the model is valid. The next task for the user is to choose one of six possible nuclear power generation and conversion systems. The possible choices are,[1,2]

(1) Liquid Metal Reactor using Rankine Cycle Conversion (1.5 kWe - 50 MWe),

(2) Liquid Metal Reactor- NERVA Derivative using Closed Brayton Cycle Conversion (1.5 kWe - 50 MWe),

(3) Solid Core Reactor using In-Core Thermionic Conversion (10 kWe - 50 MWe),

(4) Liquid Metal Reactor using AMTEC Thermoelectric Conversion (1 kWe - 50 MWe),

(5) NERVA Derivative Reactor using Magnetohydrodynamic Conversion

(100 kWe - 100 MWe), and

(6) SP-100 reactor with Thermionic conversion (100 kWe - 500 kWe).

The program then calculates the masses of the vehicle systems and estimates the mass of the propellant required for the mission. The vehicle characteristics for the mission are written to a data file, LOWTHRST.DAT, for later analysis by the user if desired.

In the case of a two-way flight, the program only passes the values needed for the first half of the flight to the trajectory generation subroutines. The propellant mass, propellant tank mass, and reaction control system propellant mass for the second half of the flight are written to a data file, TWOWAY.DAT. The user may then use this information to rerun the program for the second half of the flight (i.e. return trip).

After the vehicle systems have been specified, the program prompts the user for the direction, either Earth to Moon or Moon to Earth, of the trajectory generation. This sets the direction flags for the rest of the program. LOWTHRST then prompts the user for the final desired altitude about the target body (i.e. the Earth or the Moon). This altitude is used to define a circular orbit in the Earth-Moon plane about the target body.

The trajectory generation model requires an initial orbit and a starting time for the spacecraft. The orbit is specified by the user in classical orbital elements (a, e, i, M_0, ω, Ω) referenced to the Earth's equatorial plane. When the program is run, the user is prompted on-screen for the elements in the following order:

a - Semi-major axis (in km)

e - Eccentricity

i - Inclination (in degrees)

M_0 - Mean anomaly at the start date (in degrees)

ω - Argument of the perigee (in degrees)

Ω - Longitude of the ascending node (in degrees)

Two additional inputs are required, they are the starting date and time and the switches for the J_2 (gravitational term for the oblate Earth) and atmospheric drag effects. The date is prompted for as the month, day and year in numeric format and the time is prompted for in Greenwich Mean Time (GMT). The time is used primarily for determining the relative positions of the spacecraft and the Moon about the Earth.

The J_2 and atmospheric drag switches control whether or not the J_2 and atmospheric drag perturbations are used. The user is prompted for a zero (0) or a one (1). Zero turns the perturbation off and one turns it on for the calculations. These perturbations are accounted for only while they are of the same or higher order of magnitude perturbation than the spacecraft's thrust.

The program uses this initial information to generate the Earth escape trajectory and align the spacecraft's orbital plane of motion with that of the Moon's. After LOWTHRST has finished the orbital plane alignment of the spacecraft's orbit, there exists only one guidance control for the user to modify. This control is the value of the Jacobian constant required for the spacecraft to obtain prior to leaving the spiral escape orbit (see Vol. II Section 3.2.2). The Jacobian constant is the sum of the kinetic, potential, and angular energy of the spacecraft in the Earth-Moon system. Typical values of the Jacobian will range from 10 and higher while the spacecraft is in low Earth orbit, to 2.6 - 2.9 during a cislunar transfer. The program begins the midcourse portion of the trajectory generation by prompting the user for a value of this control and giving a range of possible values. The trajectory generation and system characterization information generated up to this point is saved. The spacecraft's trajectory is output to the screen in a graphical representation of the Earth-Moon system. The distance of the spacecraft away from the target body is shown along with the spacecraft's mass, acceleration, and elapsed time, and the spacecraft's Jacobian constant. LOWTHRST will prompt the user after a set period of integration to determine whether to continue the trajectory or restart the trajectory at the midcourse phase. The program then presents the option of modifying the guidance control value at the beginning of the midcourse phase and rerunning the trajectory. This portion of the trajectory is capable of being repeated until the user is satisfied. The midcourse and capture phases of the trajectory can be rerun until a satisfactory trajectory has been achieved.

LOWTHRST will end when the orbit of the spacecraft achieves the desired final altitude about the target body. The final position and velocity of the spacecraft is recorded with the spacecraft's characteristics in LOWTHRST.DAT. A final system calculation is performed to determine the actual amount of propellant used and estimate the appropriate modifications in the spacecraft's payload size. A complete listing of the trajectory is output to TRAJECT.DAT. The three columns in the file are the X, Y, and Z components of the spacecraft's trajectory in the nondimensionalized rotating restricted three-body coordinate system.

---

[1] Advanced Space Analysis Office - Sverdrup/NASA-LERC, "Evaluation of Advanced Propulsion/Power Concepts," presented to Advanced Space Propulsion Workshop, April 12-13, 1988.

[2] English, Robert E., "Power Generation from Nuclear Reactors in Aerospace Applications," NRC Symposium on Advanced Compact Reactors, Washington, D. C., November 15-17, 1982.

III. PROGRAM CONSIDERATIONS

The trajectory determination methods for impulsive and low-thrust spacecraft differ considerably. Electric propulsion systems need to thrust continuously for long periods of time in order to achieve a significant energy change. Chemical, or impulsive, propulsion systems can create a near instantaneous change in the spacecraft's velocity. Where an impulsive thrusting spacecraft could use two short powerful thrusts to transfer between orbits, a low-thrust orbital transfer would be accomplished as a very slow outward spiral to the desired altitude. This is because the fractional increase of the orbital radius per revolution is very small for low-thrust spacecraft. There are no general closed form analytic solutions to the low-thrust orbit change problem. This complicates the calculation of trajectories for low-thrust vehicles by requiring numerical integration of trajectories to determine spacecraft characteristics for orbit transfer such as trip time and propellant used. Contrarily, impulsive propulsion spacecraft can be simply characterized through the use of a few analytic solutions.

3.1 Vehicle Systems

During the conceptualization and design stage of spacecraft development, the required behavior of the spacecraft during its flight will impact on the vehicle's characteristics. However, the characteristics of the low-thrust spacecraft will also play an important role in determining the type of trajectory that can be flown. The interdependency of the spacecraft characteristics with the spacecraft's trajectory have made it difficult to adequately model low-thrust spacecraft. While the program does not let the user specify a complete vehicle design it does allow the user to choose two of the most important vehicle systems, the propulsion and power systems. These two systems will have a considerable impact on the vehicle characteristics and therefore on the type of trajectories possible.

The power and propulsion systems for low-thrust spacecraft are intimately coupled. The propulsion system in a low-thrust orbital transfer vehicle (OTV) will be the major drain on the power system. The amount of power the propulsion system needs can be determined from the propulsion system efficiency. The propulsion system efficiency is the fraction of electrical power that is converted to exhaust kinetic energy. This yields,

$$\eta = \frac{\dot{m}\left(I_{sp} * g\right)^2}{2P_o}$$

where η is the thruster system efficiency, $\dot{m}$ is the mass flow rate of the thrusters, and P_o is the electrical power input to the propulsion system.[1]

Currently, there are many different low-thrust electric propulsion systems under investigation. The ion engine, magnetoplasmadynamic (MPD) thruster, and arcjet are a few of the leading candidates. All of these engines will require continuous high power to be able to perform competitively against chemical propulsion. The propulsion system choices available in LOWTHRST include ion, MPD, and arcjet. In addition, the user may specify the characteristics of a custom propulsion system.

Once the propulsion and power systems have been chosen, the program uses parametric equations to size the remaining critical vehicle systems, such as the thermal management and reaction control systems, and estimates the total amount of propellant required for either a one-way or two-way flight, as specified by the user. Although data is generated for the two-way flight, when specified, only the values needed for the first half of the flight are passed to the trajectory generation subroutines. The data needed for the second half of the flight is written to data files, so the program may be run using these values to calculate better estimates of the propellant and tank masses needed for the mission. One important assumption made for computing the two-way flight data is that the vehicle does not refuel when it reaches its first destination. Also, it is assumed that empty propellant tanks are not discarded.

3.2 Trajectory Generation

With aerobraking ruled out, the guidance scheme employed to determine a trajectory must use only low-thrust to capture the OTV into Earth orbit. A low-thrust OTV is limited in the range of thrust available to drive the vehicle to the desired orbit. Another restriction for the trajectories of nuclear-powered OTVs is the proposed nuclear safe orbit (NSO)[2]. This would be a designated altitude below which the nuclear powered spacecraft would be prohibited. The spacecraft would be prohibited from descending below the restricted altitude at any point during the trajectory.

In the development of trajectories for low-thrust cislunar OTVs, little attention has been directed at the guidance and control of the spacecraft. The premise that the guidance of the vehicle and the determination of the appropriate trajectory are unrelated is false. Rather, guidance and trajectory determination are closely related problems which, by necessity, must be treated with equal importance[3]. To adequately understand the dynamics of motion of the low-thrust spacecraft, the gravitational effects of the Earth and the Moon on the spacecraft must be included for the full duration of the trajectory. The thrusting acceleration for low-thrust OTVs in high Earth orbit is the same magnitude as the perturbing force due to the Moon. Similarly the Earth will be a major perturbation of a spacecraft about the Moon. Therefore both the Moon's and Earth's gravitational pull must be accounted for during the entire trajectory generation.

The trajectory generation problem can be broken into three distinct segments all of which have equal importance. The spiral orbit and plane alignment; the orientation of the escape trajectory; and the capture and circularization about the target planet. The first segment is concerned with aligning the spacecraft's plane of orbital motion with that of the Moon's. These two planes (see Figure 1) can have different Ω's and i's with respect to the coordinate frame. This can cause the common angle between the two planes, i', to become quite large. The goal during this portion of the trajectory generation is to drive the spacecraft into the plane of motion of the Moon while spiralling out from the initial orbit. These two maneuvers require the implementation of a coupled control algorithm to raise the orbit as well as to change the Ω and i to that of the Moon's orbital plane.

The second segment involves orienting the escape trajectory of the spacecraft toward the target body. This portion of the guidance occurs after the motion of the Moon and the spacecraft is in the same plane. To model the trajectory in the Earth-Moon system with the necessary accuracy and achieve computational efficiency, the restricted three-body formulation of the dynamical equations is utilized as the governing equations of motion. The ability of the spacecraft to achieve a cislunar transfer is indicated by the Jacobian integral of the spacecraft. This integral (see Vol. II, Section 2.2.3) is a combination of the spacecraft's kinetic, potential, and angular energy. For each value of the Jacobian integral, the spacecraft's motion will be bound by a zero-velocity curve (Vol. II, Section 2.2.3). The guidance control for the spacecraft drives the Jacobian integral to have an appropriate value for the zero-velocity curve to allow cislunar transfer (Figure 2) when the spacecraft is aligned to escape.

The final portion of the trajectory generation is the capture and circularization about the target body to the desired orbit. This is accomplished through the use of two guidance algorithms. The first drives the spacecraft's radial and tangential velocity components relative to the target body to match the velocity profile of an ideal spiral capture. The second circularizes the orbit by lowering the orbit's apoapsis and raising its periapsis until a desired eccentricity is reached.

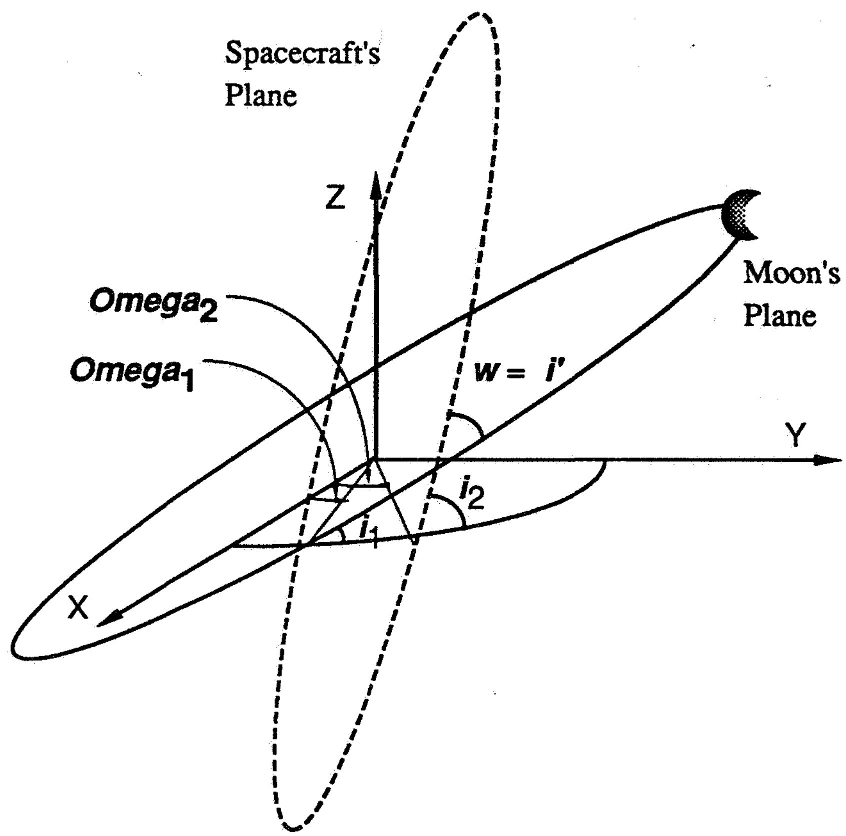

Figure 1 - Two Orbital Planes

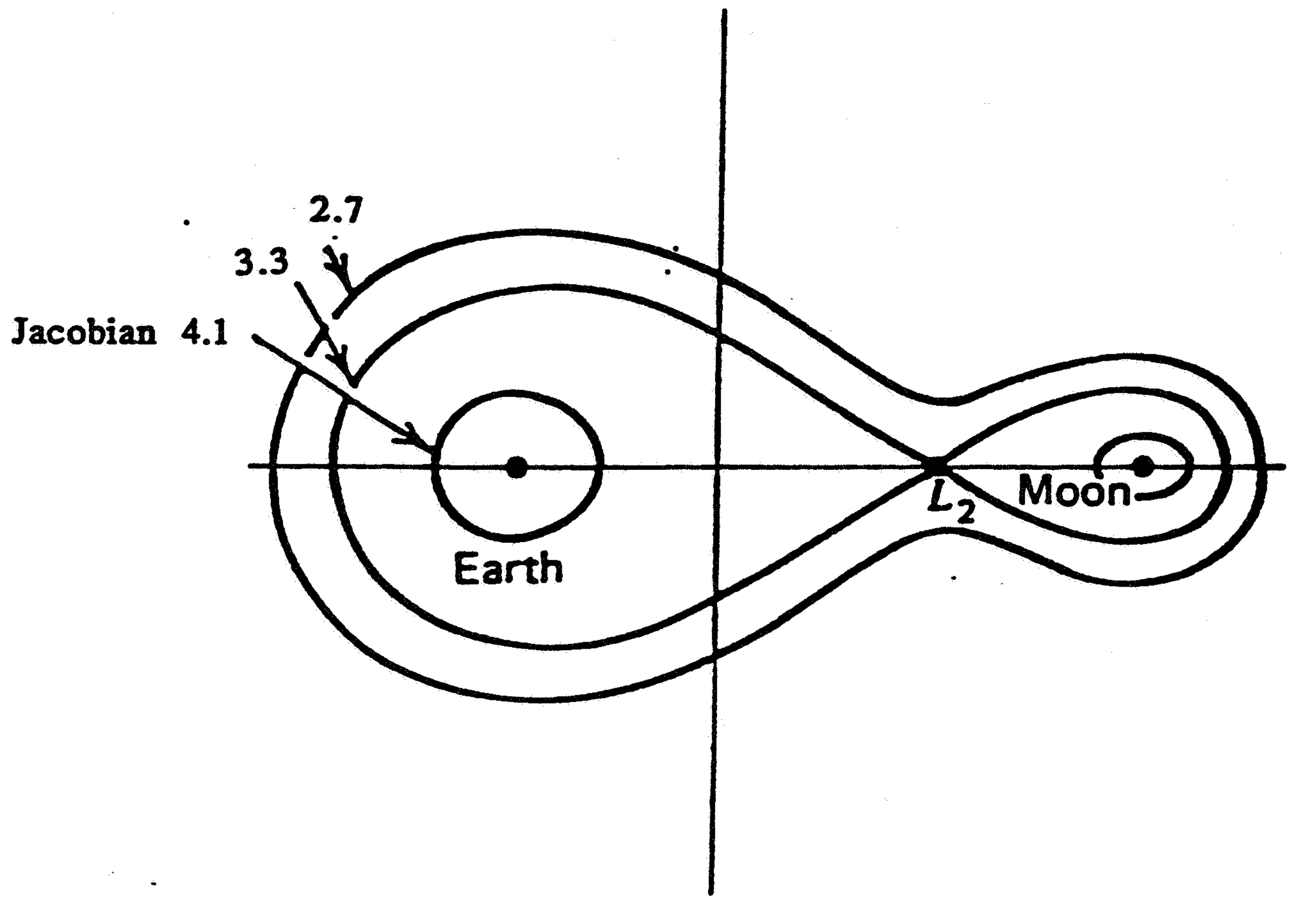

Figure 2 - Midcourse Targeting Zero Velocity

[1] Hill, P. G., and Peterson, C. R., Mechanics and Thermodynamics of Propulsion, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1965, Page 336.

[2] Galecki, Diane L., and Patterson, Micheal J., "Nuclear Powered Mars Cargo Transport Mission Utilizing Advanced Ion Propulsion," AIAA/SAE/ASME/ASEE 23rd Joint Propulsion Conference, San Diego, Ca., 1987, AIAA-87-1903.

[3] Battin, R. H., and Miller, J. S.,"Trajectories and Guidance Theory for a Continuous Low-Thrust Lunar Reconnaissance Vehicle," 6th Symposium on Ballistic Missile and Aerospace Technology, 1961.

IV. EXAMPLE RUN

The following is an example run of LOWTHRST. To start the program on PC, simply type LOWTHRST at the DOS command line. All of the commands typed in by the user will be shown in bold characters.

```
C:\ LOWTHRST

*********************************************************
*                      LOWTHRST                         *
*-------------------------------------------------------*
* TRAJECTORY DETERMINATION AND SYSTEM CHARACTERIZATION  *
*                                                       *
* Created for                                           *
*       NASA Lewis Research Center                      *
*   by the                                              *
*       Large Scale Programs Institute - October 1989   *
*********************************************************

DO YOU WISH TO RUN THE SPACECRAFT SYSTEM SIZING
PORTION OF THE MODEL? (Y/N)
Y

DO YOU WISH TO RUN THE MODEL FOR A ONE-WAY
TRIP (1) OR TWO-WAY TRIP (2) ?
1

ENTER THE PAYLOAD MASS (kg) TO BE DELIVERED:
20000

PLEASE CHOOSE ONE OF THE FOLLOWING  PROPULSION
SYSTEM TECHNOLOGIES:
(1) 50cm-DIA. ION THRUSTERS (XENON PROPELLANT)
    Isp RANGE: 3500s TO 5300s
(2) 50cm-DIA. ION THRUSTERS (KRYPTON PROPELLANT)
    Isp RANGE: 4500s TO 7000s
(3) 50cm-DIA. ION THRUSTERS (ARGON PROPELLANT)
    Isp RANGE: 6500s TO 9500s
(4) APPLIED-FIELD MAGNETOPLASMADYNAMIC (MPD)
    HYDROGEN PROPELLANT
(5) ACRCJET
    HYDROGEN PROPELLANT
(6) USER SPECIFIED
PLEASE ENTER CHOICE:
4
```

```
APPLIED-FIELD MPD PROPULSION SYSTEM
   Isp RANGE: 500s TO 10000s
PLEASE ENTER THE Isp OF THE PROPULSION SYSTEM:
3000

ENTER THE NUMBER OF THRUSTERS:
5

ENTER THE POWER INPUT TO EACH THRUSTER (kWe):
 RANGE: 100kWe TO 10000kWe
1000

THE PROPULSION SYSTEM CHOSEN REQUIRES:     5000.00 kWe
PLEASE CHOOSE ONE OF THE FOLOWING POWER SYSTEMS:
(1) LIQUID METAL REACTOR
     RANKINE CYCLE CONVERSION
     POWER RANGE: 1.5kWe TO 50MWe
(2) LIQUID METAL REACTOR - NERVA DERIVATIVE
     CLOSED BRAYTON CYCLE CONVERSION
     POWER RANGE: 1.5kWe TO 50MWe
(3) SOLID CORE REACTOR
     IN-CORE THERMIONIC CONVERSION
     POWER RANGE: 10kWe TO 50MWe
(4) LIQUID METAL REACTOR
     AMTEC THERMOELECTRIC CONVERSION
     POWER RANGE: 1kWe TO 50MWe
(5) NERVA DERIVATIVE REACTOR
     MHD CONVERSION
     POWER RANGE: 100kWe TO 100MWe
(6) SP-100 SPACE REACTOR
     THERMIONIC CONVERSION
     POWER RANGE: 100kWe TO 500kWe
PLEASE ENTER CHOICE:
2

CALCULATING ESTIMATE OF PROPELLANT REQUIRED...

INPUT DIRECTION OF TRAJECTORY GENERATION
 "0" FOR EARTH TO MOON, "1" FOR MOON TO EARTH
0

EARTH TO MOON TRAJECTORY GENERATION CHOSEN
INPUT DESIRED FINAL ALTITUDE ABOVE MOON (KM)
100

DO YOU WISH TO WRITE THE TRAJECTORY TO A FILE?
TYPE 1 FOR YES, 0 FOR NO.
0

GENERATING REFERENCE CAPTURE VELOCITIES
```

Here the screen will show the radius and the velocity components as the reference spiral is generated.

```
INPUT THE SPACECRAFT INITIAL ORBITAL ELEMENTS:
(CLASSICAL EQUATORIAL ELEMENTS)
SEMI-MAJOR AXIS (KM)  :
8000


ECCENTRICITY :
0.0001

INCLINATION (DEGREES) :
28.5

MEAN ANOMALY AT START DATE (DEG) :
0

ARGUMENT OF PERIGEE (DEG) :
0

LOMG. OF THE ASCENDING NODE (DEG) :
0

ENTER THE STARTING DATE (M/D/Y) AND TIME
MONTH (i.e. 9):
2

DAY
27

YEAR
90

STARTING YEAR WILL BE: 1990
TIME (Greenwich Mean Time in hours, i.e. 12.00)
12

INCLUDE J2 EFFECTS? (1=YES, 0=NO) :
0

INCLUDE ATMOSPHERIC DRAG EFFECTS? (1=YES, 0=NO) :
0

NOW INTEGRATING TRAJECTORY PLANE ALIGNMENT...
R= 8000.1    I =  3.8247...

DO YOU WISH TO SET THE JACOBIAN CONTROL VARIABLE? (Y/N)
N
```

When this question is answered "No" the model uses an empirical value for the Jacobian Control Value. The screen shows the following, with the values printed underneath, as the trajectory generates.

```
S/C JACOBIAN CONSTANT, TOTAL JACOBIAN, JACOBIAN CONTROL
```

The screen will show a graphical representation of the trajectory as it is being generated. The radius from the Moon will be shown as will the non-dimensional time elapsed. The program will prompt,

```
DO YOU WISH TO CONTINUE THIS TRAJECTORY? (Y/N)
```

after the non-dimensional time has reached 20, 25, and 30. If this question is answered "Y" the integration will continue until the final conditions are met. If this question is answered "N" the program will ask,

```
DO YOU WISH TO RUN MIDCOURSE AGAIN? (Y/N)
```

If this question is answered "N" the program will terminate. For this example case the question is answered "Y". Then the program will prompt,

```
DO YOU WISH TO SET THE JACOBIAN CONTROL VARIABLE? (Y/N)
Y
```

```
ENTER VALUE OF JACOBIAN CONTROL
.2
```

The screen will the show the trajectory being generated again. When the program asks,

```
DO YOU WISH TO CONTINUE THIS TRAJECTORY? (Y/N)
```

type

Y

Then the desired Earth to Moon trajectory is generated and the final conditions are met. The Program asks again,

```
DO YOU WISH TO RUN MIDCOURSE AGAIN? (Y/N)
N
```

The program has completed. Data on the spacecraft characteristics has been output in LOWTHRST.DAT

V. SUBROUTINE DICTIONARY

Subroutine List:

ARCJET - Sizing routine for the arcjet propulsion system.

ATM76 - Standard Jacchia 1976 atmospheric model.

CAPTURE - Capture and Circularization algorithms.

CLEQU - Converts from classical elements equinoctial elements.

COORE - Converts from equinoctial elements to geocentric coordinates and velocities.

CURVE - Generates Logarithmic, Exponential, Power, and Linear curve fits to data.

DER3BG - Restricted 3-body derivative and control routine.

DERIV1 - Derivative routine for the integration of 2-body orbits for SPIRAL.

DIANA - Calculates the geocentric coordinates of the Moon from a Julian date.

ECLEQ - Converts ecliptic Cartesian coordinates into mean equatorial coordinates.

EPRTLS - Calculates the partial derivatives of the equinoctial elements.

EQ2LP - Converts from equatorial position and velocity to non-rotating 3-body position and velocity at a given Julian date.

EQDERIV - Derivative routine for the integration of equinoctial elements.

EQUCL - Converts from equinoctial elements to classical elements.

EQUIN - Control Program for Earth orbit plane alignment.

GETOE - Input routine to get initial orbital elements.

INPUT0 - Input routine for the System Sizing Model.

INPUT1 - Input routine for the spacecraft payload mass.

INRK78 - Initialization routine for Runge-Kutta 7/8 Subroutine.

ION - Sizing routine for the Ion propulsion system.

JACBI3 - Calculates the Jacobi constant in the geocentric 3-body coordinate system.

JULDAY - Calculates the Julian date from the M/D/Y and UT.
KEPLER - Solves for Eccentric Anomaly from classical elements.
KEPLRE - Solves for Eccentric Longitude from the equinoctial elements.
LP2R3B - Converts from geocentric non-rotating 3-body coordinates to barycentric rotating 3-body coordinates.
LTHRST - Calculates the thrust vector for the spacecraft to align the orbital planes.
MPD - Sizing routine for the MDP propulsion system.
OE2OE3 - Converts equatorial orbital elements to non-rotating geocentric 3-body elements.
OETORC - Calculates position and velocity vectors from classical orbital elements.
OUTPUT - Outputs the spacecraft system characteristics to a file.
PAYLCHK - Recalculates System sizes based on the trajectory generation.
PERTUR - Generates perturbation effects for the EQDERIV routine.
POLYFIT - Generates Polynomial curve fits for data.
PROPMOD - Propulsion System Sizing model.
PRPEST - Propellant estimation routine.
PWRMOD - Selection and sizing routine for the power system.
QCK - Quadrant check for angles.
R3BGEN - Generates Midcourse portion of the Trajectory.
R3BGEN2 - Generates Capture portion of the Trajectory.
RCS - Sizing routine for the RCS system.
RCTOOE - Calculate classical orbital elements from position and velocity vectors.
RK78 - Runge-Kutta 7/8 Integrator.
SELENE - Calculates classical orbital elements for the Moon based on the Julian date.
SPIRAL - Generates Parametric Capture Velocities.
SYSMOD - System Sizing model for the orbital transfer vehicle.
THERM - Sizing routine for the thermal control system.
USER1 - Routine for user defined propulsion system input.

Main Program: LOWTHRST

Subroutines Directly Called:

INRK78 - Initialization routine for Runge-Kutta 7/8 Subroutine.
INPUT0 - Input routine for the System Sizing Model.
SPIRAL - Generates Parametric Capture Velocities.
EQUIN - Performs Earth Orbital Plane Alignment.
R3BGEN - Generates Midcourse portion of the Trajectory.
R3BGEN2 - Generates Capture portion of the Trajectory.
PAYLCHK - Recalculates System sizes based on the trajectory generation.

Common Blocks:

/INTAP/DT1, NEQ1, TOL1
/INTSP/DT, NEQ, TOL
/THRST/THR, PHI, GME, RE, RM, T0, EMD, XSWITCH, GMI
/SC/CA
/RKCOM/CH(13), AL(13), B(13,12)
/ELF/SCF, PAYM1, PROPM1

Input Variables:

NONE

Output Variables:

NONE

Important Internal Variables:

ALTF - Final altitude above target body (i.e. Earth or Moon) (km).

CA() - Six variables describing the Spacecraft's characteristics:
- CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg),
- CA(2) - Total Propulsion System Mass Flow Rate (kg/s),
- CA(3) - Specific Impulse of Propulsion System (seconds),
- CA(4) - Gravitational Constant of the Earth (km/sec^2),
- CA(5) - Final Vehicle Mass ("Dry Mass") (kg), and
- CA(6) - Direction of Trajectory Generation (0 = Earth to Moon, 1 = Moon to Earth).

DT - Initial step size for integration of 4th order equations (dimensional equations).

DT1 - Initial step size for integration of 6th order equations (non-dimensional equations).

EMD - Earth-Moon distance (km).

GME - Gravitational Parameter of the Earth (km^3/sec^2).

GMM - Gravitational Parameter of the Moon (km^3/sec^2).

IPRTFLAG - Flag indicating whether to print trajectory to a file (1=yes, 0=no).
NEQ - Number of equations for integrator (=4).
NEQ1 - Number of equations for integrator (=6).
PI - The value of pi.
RE - Radius of the Earth (km).
RL2 - Distance from the central body that SPIRAL will generate the parametric reference velocities (km).
SCF - Spacecraft Final Mass (kg).
TA - Time of integration variable (non-dimensional).
TND - Time conversion factor, dimensional to non-dimensional.
TOL - Tolerance of variance for integrator using NEQ.
TOL1 - Tolerance of variance for integrator using NEQ1.
X() - The six Cartesian position and velocity components for the spacecraft (either km and km/s or non-dimensional).
XB() - The Array of position and velocity of the spacecraft after EQUIN subroutine, seven elements: position, velocity and time of integration (non-dimensional).
XE - Distance from the barycenter of Earth-Moon system to the Moon's Center (km).
XF() - A holding Array for the state vector.
XM - Distance from the barycenter of Earth-Moon system to the Earth's Center (km).
XSWITCH - Distance from initial body that the capture algorithms begin (km).

Subroutine: R3BGEN
Generates midcourse portion of the Trajectory
Subroutines Directly Called:
RK78 - Runge-Kutta 7/8 Integrator.
Common Blocks:
/R3B/XE, XM, TOM
/INTAP/DT1, NEQ1, TOL1
/THRST/THR, PHI, GME, RE, RM, T0, EMD, XSWITCH, GMI
/SC/CA
/R3BG/IGDE
Input Variables:
XIS() - Initial State Vector of the Spacecraft (non-dimensional) six elements.

TS - Non-dimensional time of integration.

Output Variables:

XFS() - Final State Vector of the Spacecraft (non-dimensional) six elements.

Important Internal Variables:

ACCND - Conversion factor for accelerations, non-dimensional to dimensional).

ANGLE - Orbital Quadrant angle about departure planet.

C0 - Jacobian constant of the spacecraft in the Restricted 3-body formulation.

CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg).

CA(2) - Total Propulsion System Mass Flow Rate (kg/s).

CA(6) - Direction of Trajectory Generation (0 - Earth to Moon, 1 - Moon to Earth).

CAP - Midcourse control Value.

CAPL - Parametric Midcourse control Value.

DT3 - Integrator fixed step size (non-dimensional).

IFLAG - Flag for printing control variable to screen.

IFLAG2 - Flag for turning 1=on/0=off the midcourse control.

IGDE - Thrusting algorithm control flag.

OMEGA - Potential in the Restricted 3-body formulation.

R1 - Distance from the spacecraft to the Earth (non-dim).

R2 - Distance from the spacecraft to the Moon (non-dim).

T - Non-dimensional time.

TND - Time conversion factor, dimensional to non-dimensional.

TOM - Non-dimensional spacecraft acceleration due to the propulsion system.

V0 - Non-dimensional velocity in Restricted 3-body rotating barycentric coordinate system.

XAXIS - X coordinate distance from the center of the departure planet (i.e. Earth or Moon).

XCUE1 - Control Angle.

XCUE2 - Control Angle.

XE - Distance from the barycenter of Earth-Moon system to the Moon's Center (km).

XM - Distance from the barycenter of Earth-Moon system to the Earth's Center (km).

XMASS - Current Spacecraft Mass (kg).

Subroutine: R3BGEN2

Generates capture portion of the Trajectory

Subroutines Directly Called:

RK78 - Runge-Kutta 7/8 Integrator.

Common Blocks:

/R3B/XE, XM, TOM
/INTAP/DT1, NEQ1, TOL1
/THRST/THR, PHI, GME, RE, RM, T0, EMD, XSWITCH, GMI
/SC/CA
/R3BG/IGDE
/TIME/T0R, TF, TNODE

Input Variables:

XIS() - Initial State Vector of the Spacecraft (non-dimensional) six elements.

TS - Non-dimensional time of integration.

ALTF - Final altitude above target body (i.e. Earth or Moon) (km).

Output Variables:

XIS() - Final State Vector of the Spacecraft (non-dimensional) six elements.

Important Internal Variables:

ACCND - Conversion factor for accelerations, non-dimensional to dimensional).

AM - Classical Semimajor Axis of the spacecraft's orbit about the Target planet (km).

C0 - Jacobian constant of the spacecraft in the Restricted 3-body formulation.

CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg).

CA(2) - Total Propulsion System Mass Flow Rate (kg/s).

CA(6) - Direction of Trajectory Generation (0 - Earth to Moon, 1 - Moon to Earth).

CAPF - Jacobian constant Capture control value.

CAPRAD - Distance from initial body that the circularization algorithms begin (km).

DT3 - Integrator fixed step size (non-dimensional).

EM - Classical Eccentricity of the spacecraft's orbit about the Target planet.

ENGM - Keplerian Energy of the spacecraft about the Target planet (km^2/sec^2).

GMI - Gravitational parameter of the Target planet (km^3/sec^2).
IGDE - Thrusting algorithm control flag.
OMEGA - Potential in the Restricted 3-body formulation.
R1 - Distance from the spacecraft to the Earth (non-dim).
R2 - Distance from the spacecraft to the Moon (non-dim).
RF - Radius of the Target planet (km).
RI - Distance of the spacecraft from the Target planet (non-dimensional).
T - Non-dimensional time.
TND - Time conversion factor, dimensional to non-dimensional.
TOM - Non-dimensional spacecraft acceleration due to the propulsion system.
V0 - Non-dimensional velocity in Restricted 3-body rotating barycentric coordinate system.
VELND - Conversion factor for non-dimensional velocity to dimensional velocity.
XE - Distance from the barycenter of Earth-Moon system to the Moon's Center (km).
XM - Distance from the barycenter of Earth-Moon system to the Earth's Center (km).
XMASS - Current Spacecraft Mass (kg).
XSWITCH - Distance from initial body that the capture algorithms begin (km).

Subroutine: INPUT0

Input routine for the System Sizing Model

Subroutines Directly Called:

SYSMOD - System Sizing Model for the orbital transfer vehicle

Common Blocks:

/SC/CA

Input Variables:

NONE

Output Variables:

CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg),
CA(2) - Total Propulsion System Mass Flow Rate (kg/s),
CA(3) - Specific Impulse of Propulsion System (seconds),
CA(4) - Gravitational Constant of the Earth (km/sec^2),
CA(5) - Final Vehicle Mass ("Dry Mass") (kg), and

CA(6) - Direction of Trajectory Generation (0 = Earth to Moon, 1 = Moon to Earth).

Subroutine: SYSMOD

System Sizing Model for the orbital transfer vehicle

Subroutines Directly Called:

INPUT1 - Input routine for the spacecraft payload mass
PROPMOD - Propulsion system sizing model
PWRMOD - Selection and sizing routine for the power system
THERM - Sizing routine for the thermal control system
RCS - Sizing routine for the RCS system
PRPEST - Propellant estimation routine
OUTPUT - Outputs the spacecraft system characteristics to a file

Common Blocks:

/ELF/SCF, PAYM1, PROPM1

Input Variables:

NONE

Output Variables:

CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg),
CA(2) - Total Propulsion System Mass Flow Rate (kg/s),
CA(3) - Specific Impulse of Propulsion System (seconds),
CA(4) - Gravitational Constant of the Earth (km/sec^2),
CA(5) - Final Vehicle Mass ("Dry Mass") (kg), and
CA(6) - (Not used by System Sizing Model).

Important Internal Variables:

EFFPWR - Efficiency of power system
EFFTHR - Efficiency of thrusters
FLAG - Flag indicating one (1) or two (2) way trip
N1 - Flag indicating which propulsion system was chosen
N2 - Flag indicating which power system was chosen
NTHR - Number of thrusters
PAYM - Total payload mass (kg)
PAYM1 - Payload Mass 1 (delivered) (kg)
PAYM2 - Payload Mass 2 (delivered) (kg)
PM1 - Payload1 + Payload structural support mass (kg)
PM2 - Payload2 + Payload structural support mass (kg)
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)

PRSYSM - Propulsion system mass (kg)
PROPM1 - Propellant mass for delivery trip (kg)
PROPM2 - Propellant mass for return trip (kg)
PWRSM - Power system mass (kg)
RCSM - RCS system mass (kg)
RCSPM1 - RCS propellant mass for delivery trip (kg)
RCSPM2 - RCS propellant mass for return trip (kg)
RCSTHM - RCS system thrust mass (kg)
RCSTM - RCS thruster+structural support mass (kg)
REACM - Reactor mass (kg)
RISP - Specific Impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
RMDOT2 - Total mass flow rate (kg/s)
RMI - initial vehicle "wet" mass (kg)
STRUCM - Estimate of structural mass of vehicle (kg)
TB1 - Estimated burn time for delivery trip (s)
TB2 - Estimated burn time for return trip (s)
TB - Estimated total burn time (s)
TEXIT - power system exit temperature (K)
THCOM - Thermal control system mass (kg)
TM1 - Tank mass for delivery trip (kg)
TM2 - Tank mass for return trip (kg)
TREJ - Power system rejection temperature (K)
VBM - Vehicle base mass (kg) (composed of propulsion, power, and thermal control system masses and associated structural mass)

Subroutine: INPUT1

Input routine for the spacecraft payload mass

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

FLAG - Flag for one-way (1) or two-way (2) trip

Output Variables:

PAYM - Total payload mass (kg)
PAYM1 - Payload mass to be delivered (kg)
PAYM2 - Payload mass to be returned (kg)

Subroutine: PROPMOD

Propulsion System Sizing Model

Subroutines Directly Called:

ION - Sizing routine for the Ion propulsion system
MPD - Sizing routine for the MDP propulsion system
ARCJET - Sizing routine for the arcjet propulsion system
USER1 - Routine for user defined propulsion system input

Common Blocks:

NONE

Input Variables:

NONE

Output Variables:

N1 - Flag indicating one (1) or two (2) way trip
RISP - Specific impulse (s)
PPT - Power input per thruster (kWe)
POWREQ - Total power required (MWe)
PRSYSM - Propulsion system mass (kg)
RMDOT - Mass flow rate (kg/s) per thruster
NTHR - Number of thrusters
EFFTHR - Efficiency of thrusters

Subroutine: ION

Sizing routine for the Ion propulsion system

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

N1

Output Variables:

PPT - Power input per thruster (MWe)
POWREQ - Total power required (MWe)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
NTHR - Number of thrusters
EFFTHR - Efficiency of thrusters

Important Internal Variables:

C - Exhaust velocity (m/s)
EFFTHR - Efficiency of thrusters
GE - Gravitational acceleration constant of earth (m/s^2)
GIMBM - Mass of gimbal system for each thruster (kg)
HSTRM - Thruster system housing structure mass (kg)
NTHR - Number of thrusters
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)
PPTM - Propulsion system mass (kg)
RMDOT - Mass flow rate (kg/s) per thruster
STRTM - Total mass of thruster structure (kg)
TGTM - Total thruster/gimbal mass (kg)
THRM - Mass of each 50cm dia. thruster (kg)
TSCM - Mass of thrust system controller (kg)

Subroutine: MPD

Sizing routine for the MDP propulsion system

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

NONE

Output Variables:

PPT - Power input per thruster (MWe)
POWREQ - Total power required (MWe)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
NTHR - Number of thrusters
EFFTHR - Efficiency of thrusters

Important Internal Variables:

ALPHA - Specific mass of propulsion system (kg/kWe)
C - Exhaust velocity (m/s)
EFFTHR - Efficiency of thrusters
NTHR - Number of thrusters
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)

PPTM - Propulsion system mass (kg)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
THRM - Mass of each 50cm dia. thruster (kg)
UA - ALFVEN critical velocity (m/s)

Subroutine: ARCJET
Sizing routine for the arcjet propulsion system
Subroutines Directly Called:
NONE
Common Blocks:
NONE
Input Variables:
NONE
Output Variables:
PPT - Power input per thruster (MWe)
POWREQ - Total power required (MWe)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
NTHR - Number of thrusters
EFFTHR - Efficiency of thrusters
Important Internal Variables:
ALPHA - Specific mass of propulsion system (kg/kWe)
C - Exhaust velocity (m/s)
EFFTHR - Efficiency of thrusters
GE - Gravitational acceleration constant of earth (m/s^2)
NTHR - Number of thrusters
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)
PPTM - Propulsion system mass (kg)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
THRM - Mass of each 50cm dia. thruster (kg)
UA - ALFVEN critical velocity (m/s)

Subroutine: USER1

Routine for user defined propulsion system input

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

NONE

Output Variables:

PPT - Power input per thruster (MWe)
POWREQ - Total power required (MWe)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster

Important Internal Variables:

EFFTHR - Efficiency of thrusters
NTHR - Number of thrusters
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)
PRSYSM - Propulsion system mass (kg)
RISP - Specific impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster

Subroutine: PWRMOD

Selection and sizing routine for the power system

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

POWREQ

Output Variables:

TOP -
EFFPWR - Efficiency of power system
PWRSM - Power system mass (kg)

Important Internal Variables:

AMTECM - Amtec mass (kg)
CNTRLM - Control/misc. mass (kg)
EFFPWR - Efficiency of power system

HTM - Heat transport system mass (kg)
HTPCM - Heat transport mass + power conversion system. mass (kg)
N2 - Flag indicating which power system was chosen
PKW - Total power required (kWe)
POWREQ - Total power required (MWe)
PW - Total power required (We)
PWRSM - Power system mass (kg)
REACM - Reactor mass (kg)
SPECM - Specific mass (kg/kWe)
TOP - Operating temperature (K)

Subroutine: THERM

Sizing routine for the thermal control system

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

POWREQ - Total power required (MWe)
EFFPWR - Efficiency of power system

Output Variables:

THCOM - Thermal control system mass (kg)

Important Internal Variables:

RSM - Radiator specific mass (kg/kWt)
THMPWR - Thermal power to reject (kWt)

Subroutine: RCS

Sizing routine for the RCS system

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

PAYM - Total payload mass (kg)
VBM - Vehicle base mass (kg)

Output Variables:

RCSTHM - RCS system thruster mass (kg)

RCSPRM - RCS propellant mass (kg)

Important Internal Variables:

PRPM - Estimate of main propulsion system propellant mass (kg)

RATIO - Ratio of mass of RCS propellant to vehicle wet mass

VMEST - Estimate of vehicle "wet" mass (kg)

Subroutine: PRPEST

Propellant estimation routine

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

PAYM - Total payload mass (kg)

RISP - Specific impulse (s)

NTHR - Number of thrusters

RMDOT - Mass flow rate (kg/s) per thruster

RCSM - RCS system mass (kg)

Output Variables:

PROPM - Propellant mass (kg)

RMI - Initial vehicle mass "wet" (kg)

TB - Burn time (s)

Important Internal Variables:

DMP - Propellant mass increment (kg)

DMT - Tank mass increment (kg)

DVLT - Low-thrust Delta V required for Mission (m/s)

DVM - Velocity of vehicle (m/s)

NTHR - Number of thrusters

PF - Propellant factor

RMDOT - Mass flow rate (kg/s) per thruster

RMDOT2 - total mass flow rate (kg/s)

UEQ - Exhaust velocity (m/s)

VBM - Base vehicle mass (i.e. No. propellant, payload) (kg)

Subroutine: OUTPUT

Outputs the spacecraft system characteristics to a file

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

EFFTHR - Efficiency of thrusters
FLAG - Flag indicating one (1) or two (2) way trip
N1 - Flag indicating which propulsion system was chosen
N2 - Flag indicating which power system was chosen
NTHR - Number of thrusters
PAYM1 - Payload Mass 1 (delivered) (kg)
PAYM2 - Payload Mass 2 (delivered) (kg)
POWREQ - Total power required (MWe)
PPT - Power input per thruster (kWe)
PROPM - Propulsion system mas (kg)
PRSYSM - Propulsion system mass (kg)
PWRSM - Power system mass (kg)
RCSM - RCS system mass (kg)
RCSPM1 - RCS propellant mass for delivery trip (kg)
RCSPM2 - RCS propellant mass for return trip (kg)
RISP - Specific Impulse (s)
RMDOT - Mass flow rate (kg/s) per thruster
RMI - initial vehicle "wet" mass (kg)
SPECM - Specific Mass of power system
STRUCM - Estimate of structural mass of vehicle (kg)
TB - Estimated total burn time (s)
THCOM - Thermal control system mass (kg)
TM - Tank mass for delivery trip (kg)
TOP - Power system operating temperature (K)

Subroutine: PAYLCHK

Recalculates system sizes based on the trajectory generation

Subroutines Directly Called:

NONE

Common Blocks:

/ELF/SCF, PAYM1, PROPM1
/SC/CA

Input Variables:

CA(1) - Initial Spacecraft Mass ("Wet Mass") (kg)
CA(2) - Total Propulsion System Mass Flow Rate (kg/s)

CA(3) - Specific Impulse of Propulsion System (seconds)
CA(4) - Gravitational Constant of the Earth (km/sec^2)
CA(5) - Final Vehicle Mass ("Dry Mass") (kg)
CA(6) - (Not used by this subroutine
PAYM1 - Desired payload entered by user (kg)
PROPM1 - Mass of Propellant for delivery flight (kg)
SCF - True final spacecraft mass (kg)

Output Variables:

TPAYM - True payload mass (kg)

Subroutine: DER3BG

Restricted 3-body derivative and control routine

Subroutines Directly Called:

CAPTURE - Capture and Circularization algorithms

Common Blocks:

/R3B/XE,XM,TOM
/R3BG/IGDE
/SC/CA

Input Variables:

T - Non-dimensional time
X() - State vector of the spacecraft (non-dim)

Output Variables:

DX() - Vector of the differentials of the state for the integrator

Important Internal Variables:

ADX() - Array (3) containing the acceleration components due to the low-thrust (non-dim)
DX() - Array (6) of the differential equations of motion for the restricted 3-body formulation (non-dim)
RAD1 - Distance from the departure planet (non-dim)
RAD2 - Distance from the target planet (non-dim)

Subroutine: CAPTURE

Capture and circularization algorithms

Subroutines Directly Called:

NONE

Common Blocks:

/R3B/XE,XM,TOM
/CAPT/VR, VT, HR

/THRST/THR, PHIP, GME, GMM, RE, RM, PI, T0, EMD, XSWITCH, GMI
/R3BG/IGDE
/SC/CA

Input Variables:

RADD -Radius from the target body (non-dim)
XH() - State vector of the spacecraft (non-dim)

Output Variables:

ADX() - Array (3) containing the acceleration components due to the low-thrust (non-dim)

Important Internal Variables:

ACR - Nominal spacecraft radial acceleration (km/s^2)
ACS - Nominal spacecraft transverse acceleration (km/s^2)
AE() - Calculated spacecraft acceleration vector in RSW coords (km/s^2)
AER - Desired spacecraft radial acceleration (km/s^2)
AES - Desired spacecraft transverse acceleration (km/s^2)
AEW - Desired spacecraft normal acceleration (km/s^2)
GMI - Gravitational parameter of the target planet (km^3/s^2)
IGDE - Guidance flag
RTN() - Vector of the spacecraft's radial,transverse, and normal components (km/s)
VMAG - Velocity magnitude (km/s)
VRR - Radial reference velocity (km/s)
VTT - Transverse reference velocity (km/s)
X() - Centric state vector of the spacecraft

Subroutine: SPIRAL

Generates parametric Capture velocities

Subroutines Directly Called:

RK78 - Runge-Kutta 7/8 integrator
CURVE - Generates Logarithmic, Exponential, Power, and Linear curve fits to data
POLYFIT - Generates polynomial curve fits for data

Common Blocks:

/INTSP/DT, NEQ, TOL
/THRST/THR, PHI, GME, GMM, RE, RM, PI, T0, EMD, XSWITCH, GMI
/CAPT/VR, VT, HR

/SC/CA

Input Variables:

AINC - Inclination of final orbit about the moon

ALTF - The altitude of the final orbit

Output Variables:

NONE

Important Internal Variables:

CA(5) - Spacecraft FINAL mass

CA(2) - Mass flow rate

CA(3) - Isp

OMEGA - Angle between the velocity vector and the thrust

GM - Gravitational constant for the moon

SCM - Spacecraft mass

RVRVST - Array of the nominal capture guidance elements

Subroutine: DERIV1

Derivative routine for the integration of 2-body orbits for SPIRAL

Subroutines Directly Called:

NONE

Common Blocks:

/THRST/THR, PHI, GME, GMM, RE, RM, PI, T0, EMD, XSWITCH, GMI

/SC/CA

Input Variables:

T - Time

X() - State vector of the spacecraft (km and km/s)

Output Variables:

DX() - Vector of the differentials of the state for the integrator

Subroutine: EQUIN

Control program for earth orbit plane alignment

Subroutines Directly Called:

CLEQU - Converts from classical elements equinoctial elements

COORE - Converts from equinoctial elements to geocentric coordinates and velocities

EQUCL - Converts from equinoctial elements to classical elements

EQ2LP - Converts from equatorial position and velocity to non-rotating 3-body position and velocity at a given Julian date
GETOE - Input routine to get initial orbital elements
INRK78 - Initialization routine for Runge-Kutta 7/8 Subroutine
LP2R3B - Converts from geocentric non-rotating 3-body coordinates to barycentric rotating 3-body coordinates
OETORC - Calculates position and velocity vectors from classical orbital elements
RCTOOE - Calculate classical orbital elements from position and velocity vectors
RK78 - Runge-Kutta 7/8 Integrator
SELENE - Calculates classical orbital elements for the Moon based on the Julian dates

Common Blocks:

/THRST/THR, PHI, GME, GMM, RE, RM, PI, T0, EMD, XSWITCH, GMI
/MU/RMU, UMU
/LTHRST1/TI, TG, XN, ITSW, DILAST, WEDGE, NTHRSHFTS
/DBG/FN, FNMAX, V, DI, TA
/SC/CA, DF

Input Variables:

ELCL(1) - a (in earth-moon distances)
ELCL(2) - eccentricity
ELCL(3) - inclination (degrees)
ELCL(4) - Mean anomaly at t-0 (degrees)
ELCL(5) - Cap omega (degrees)
ELCL(6) - Omega (degrees)
ELCL(7) - Time (seconds)

Output Variables:

X(7) - State vector of spacecraft in Restricted 3-body coords (non-dim)

Important Internal Variables:

ELCL() - Vector of classical orbit elements
ELEM() - Vector of Equinoctial coordinate elements
ITSW - Switch for which perturbations will be included
OEI - Input orbit elements
TAM - True anomaly of the Moon (rad)
TG - Time guess for escape (seconds)
WEDGE - Initial inclination of the orbit with the Moon's orbit (rad)

XDEG - Cut off degrees for plane alignment (rad)

Subroutine: EQDERIV
Derivative routine for the integration of equinoctial elements
Subroutines Directly Called:
COORE - Converts from equinoctial elements to geocentric coordinates and velocities
EPRTLS - Calculates the partial derivatives of the equinoctial elements
PERTUR - Generates perturbation effects for the EQDERIV routine
Common Blocks:
/MU/RMU, UMU
Input Variables:
T - Time (non-dim)
ELEM() - Vector of Equinoctial coordinate elements
Output Variables:
F() - Vector of the derivatives of the equinoctial elements

Subroutine: PERTUR
Generates perturbation effects for the EQDERIV routine
Subroutines Directly Called:
EQUCL - Converts from equinoctial elements to classical elements
LTHRST - Calculates the thrust vector for the spacecraft to align the orbital planes
ATM76 - Standard Jacchia 1976 atmospheric model
Common Blocks:
/THRST/THR, PHI, GME, GMM, RE, RM, PI, T0, EMD, XSWITCH, GMI
/SWITCH/ISW
/LTHRST1/TI, TG, XN, ITSW, DILAST, WEDGE, NTHRSHFTS
Input Variables:
T - Time (non-dim)
ELEM() - Vector of Equinoctial coordinate elements
X() - State vector of the spacecraft (km and km/s)
RMU - Gravitational parameter of the Earth
Output Variables:

P() - Vector of the RSW perturbations

Subroutine: ATM76

Standard Jacchia 1976 atmospheric model

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

H - Height above the surface of the Earth

Output Variables:

RHO - Density of the atmosphere

Subroutine: COORE

Converts from equinoctial elements to geocentric coordinates and velocities

Subroutines Directly Called:

KEPLRE - Solves for Eccentric Longitude from the equinoctial elements

EQUCL - Converts from equinoctial elements to classical elements

Common Blocks:

/COM1/VF, VG, X1, Y1, X2, Y2, N, B, RA, RB, D, SF, CF, R, ECCAN, ML

Input Variables:

UMU - Gravitational parameter of the central body

EQEL() - Vector of equinoctial elements

IT - Number of iterations for the KEPLRE subroutine

Output Variables:

COOR() - Vector of centric coordinates

Subroutine: KEPLRE

Solves for Eccentric Longitude from the equinoctial elements

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

RML - Mean Longitude
RH - h, the equinoctial element
RK - k, the equinoctial element
EPS - Arbitrarily small number
IT - Number of iterations for the KEPLRE subroutine
MAX - Maximum number of iterations for the KEPLRE subroutine

Output Variables:

ECAN - Eccentric longitude

Subroutine: CLEQU

Converts from classical elements equinoctial elements

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

CLEL() - Vector of classical orbit elements

Output Variables:

EQEL() - Vector of equinoctial elements

Subroutine: EQUCL

Converts from equinoctial elements to classical elements

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

EQEL() - Vector of equinoctial elements

Output Variables:

CLEL() - Vector of classical orbit elements

Subroutine: EPRTLS

Calculates the partial derivatives of the equinoctial elements

Subroutines Directly Called:

NONE

Common Blocks:
/COM1/VF, VG, X1, Y1, X2, Y2, N, B, RA, RB, D, SF, CF, R, ECCAN, RML

Input Variables:
EQEL() - Vector of equinoctial elements
COOR() - Vector of centric coordinates

Output Variables:
DPDXD() - Vector of the partial of the equinoctial elements

Subroutine: JACBI3
Calculates the Julian date from the M/D/Y and UT

Subroutines Directly Called:
NONE

Common Blocks:
NONE

Input Variables:
UMU - Gravitational parameter of the Moon
RMU - Gravitational parameter of the Earth
X() - State vector of the spacecraft in geocentric coordinates

Output Variables:
CON - Value of the Jacobi Constant

Subroutine: OE2OE3
Converts equatorial orbital elements to non-rotating geocentric 3-body orbital elements

Subroutines Directly Called:
NONE

Common Blocks:
NONE

Input Variables:
ELCL() - Vector of classical orbit elements
O3 - Longitude of the ascending node (rad)
XI3 - Inclination (rad)
DIST - Earth-Moon distance
TIME

Output Variables:
ELCL3() - Vector of non-rotating three body orbital elements

Subroutine: KEPLER

Solves for eccentric anomaly from classical elements

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

M - Mean anomaly (rad)

EC - Eccentricity

Output Variables:

E - Eccentric anomaly (rad)

Subroutine: SELENE

Calculates classical orbital elements for the Moon based on the Julian date

Subroutines Directly Called:

DIANA - Calculates the geocentric coordinates of the Moon from a Julian date

ECLEQ - Converts ecliptic Cartesian coordinates into mean equatorial coordinates

RCTOOE - Calculate classical orbital elements from position and velocity vectors

KEPLER - Solves for Eccentric Anomaly from classical elements

Common Blocks:

NONE

Input Variables:

XJD - Julian date

Output Variables:

TAM - True anomaly of the Moon (rad)

XIM - Inclination of the Moon (rad)

CAPOM - Longitude of the ascending node of the Moon's Orbit (rad)

Subroutine: LTHRST

Calculates the thrust vector for the spacecraft to align the orbital planes

Subroutines Directly Called:

NONE

Common Blocks:

/COM1/VF, VG, X1, Y1, X2, Y2, YN, B, RA, RB, D, SF, CF, YR, ECCLON, YML
/MU/RMU, UMU
/LTHRST1/TI, TG, XN, ITSW, DILAST, WEDGE, NTHRSHFTS
/THRST/THR, PHI, GME, GMM, RE, RM, PI, TO, EMD, XSWITCH, GMI
/SC/CA, DF
/DBG/FN, FNMAX, V, DI, TA

Input Variables:

ELCL() - Vector of classical orbit elements
X() - State vector of the spacecraft

Output Variables:

PT() - Vector of perturbing acceleration for plane alignment (km/s^2)

Subroutine: GETOE

Input routine to get initial orbital plane alignment

Subroutines Directly Called:

JULDAY - Calculates the Julian date from the M/D/Y and UT

Common Blocks:

/THRST/THR, PHI, GME, GMM, RE, RM, PI, TO, EMD, XSWITCH, GMI
/SWITCH/ISW

Input Variables:

NONE

Output Variables:

XJD - Julian date
OEI() - Initial classical orbit elements

Subroutine: LP2R3B

Converts from geocentric non-rotating 3-body coordinates to barycentric rotating 3-body coordinates

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

X() - State vector of the spacecraft in geocentric non-rotating coords

TAM - True anomaly of the Moon (rad)

TAMORIG - True anomaly of the Moon at initial date (rad)

RMU - Gravitational parameter of the Earth

VN - Rotation rate of the Earth-Moon system (rad/second)

Output Variables:

X() - State vector of the spacecraft in restricted 3-body coords

Subroutine: RCTOOE

Calculate classical orbital elements from position and velocity vectors

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

POS - Position vector, X, Y, Z

VEL - Velocity vector XDOT, YDOT, ZDOT

MU - Gravitational parameter

Output Variables:

ESETC - Array of orbital elements

MANOM - Mean Anomaly of satellite

P - Orbital period

RP - Distance at perigee

RA - Distance at apogee

ERRFLAG - 0 = O.K.; - 1 = Some classical Keplarian elements are undefined

Important Internal Variables:

H - Magnitude of angular momentum vector of orbiting object

HX - Component of angular momentum vector in X direction

HY - Component of angular momentum vector in Y direction

HZ - Component of angular momentum vector in Z direction

POSMAG - Magnitude of position vector

VELMAG - Magnitude of velocity vector

SPE - Specific mechanical energy of orbit

F - True anomaly

FARG - True anomaly argument to acos function

E - Eccentric anomaly
U - Argument of latitude
RDOTV - Position dotted with velocity
SINU - SINE of ARG of Latitude - U
COSU - COSINE of ARG of Latitude - U
SINO - SINE of Longitude of ascending node - Omega
COSO - COSINE of Longitude of ascending node - Omega
ENU - Mean Motion
A - Semi-major axis
ECC - Eccentricity
INCL - Inclination
ECCARG - Eccentricity argument of sort function
BIGO - Longitude of ascending node, cap omega
TOL - Tolerance used to check ECC and INCL close to zero
TAU - Time of periapsis passage
PI - Const. 3.14159...

Subroutine: OETORC
Calculate position and velocity vectors from classical orbital elements
Subroutines Directly Called:
QCK - Quadrant check for angles
Common Blocks:
NONE
Input Variables:
ESETC() - Vector of the classical orbit elements
Output Variables:
X() - Vector of the X, Y, and Z coords (km)
XDOT() - Vector of the velocities (km/s)

Subroutine: QCK
Quadrant check for angles
Subroutines Directly Called:
NONE
Common Blocks:
NONE
Input Variables:
ANGLE - Angle to be evaluated (rad)

Output Variables:
ANGLE - Angle to be evaluated (rad)

Subroutine: EQ2LP
Converts from equatorial position and velocity to non-rotating 3-body position and velocity at a given Julian date
Subroutines Directly Called:
DIANA - Calculates the geocentric coordinates of the Moon from a Julian date
ECLEQ - Converts ecliptic Cartesian coordinates into mean equatorial coordinates
RCTOOE - Calculate classical orbital elements from position and velocity vectors
KEPLER - Solves for Eccentric Anomaly from classical elements
Common Blocks:
NONE
Input Variables:
X - Equatorial XYZ coordinates (Km/sec)
XDOT - Equatorial velocity components (Km/sec)
XJD - Julian date
Output Variables:
X - Non-rotating lunar plane XYZ coordinates (Km/sec)
XDOT - Non-rotating lunar plane velocity components (km/sec)

Subroutine: JULDAY
Calculates the Julian date from the M/D/Y and UT
Subroutines Directly Called:
NONE
Common Blocks:
NONE
Input Variables:
M - Month
D - Day
Y - Year
UT - Universal time
Output Variables:
JD - Julian Date

Subroutine: DIANA

Calculates the geocentric coordinates of the Moon from a Julian date

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

DATE - Julian Date

EXACT - Flag for degree of precision

Output Variables:

XE - Position of the Moon w.r.t. ecliptic system

ANGLE - Angle the Moon makes with the ecliptic plane

Subroutine: ECLEQ

Converts ecliptic Cartesian coordinates into mean equatorial coordinates

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

X() - Vector of spacecraft in Cartesian coords

TJD - Julian date

Output Variables:

X() - Vector of spacecraft in mean equatorial coords

Subroutine: RK78

Runge-Kutta 7/8 integrator

Subroutines Directly Called:

DERIV

Common Blocks:

/RKCOM/CH, AL, B

Input Variables:

DERIV - Name of the derivative subroutine

T - Time of integration

X() - State vector to be integrated

DT - Time step for integration
TOL - Tolerance of the integration
N - Number of derivative equations

Output Variables:

T - Time of integration
X() - State vector to be integrated
DT - Time step for integration

Subroutine: INRK78

Initialization routine for Runge-Kutta 7/8 subroutine

Subroutines Directly Called:

NONE

Common Blocks:

/RKCOM/CH, AL, B

Input Variables:

NONE

Output Variables:

NONE

Subroutine: CURVE

Generates Logarithmic, Exponential, Power, and Linear curve fits to data

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

X - Data to be fit to the curve
Y - Independent values
N - Number of data points

Output Variables:

CV - Coefficients of the curve equation
HR - Type of equation chosen

Subroutine: POLYFIT

Generates Polynomial curve fits for data

Subroutines Directly Called:

NONE

Common Blocks:

NONE

Input Variables:

PR() - Array of values for curve fit

PV() - Array of dependent values to be curve fit

N - Number of datum

DEGREE - Degree of polynomial fit

Output Variables:

VR() - Vector of the polynomial curve coefficients

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-185171		

4. Title and Subtitle	5. Report Date
A Trajectory Generation and System Characterization Model for Cislunar Low-Thrust Spacecraft Volume I—User's Manual	May 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
David J. Korsmeyer, Elfego Pinon III, Brendan M. O'Connor, and Curt R. Bilby	LSPI-902
	10. Work Unit No. 506-64-12
9. Performing Organization Name and Address Large Scale Programs Institute 2815 San Gabriel Austin, Texas 78705	11. Contract or Grant No. NAG3-928
	13. Type of Report and Period Covered Grant Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	14. Sponsoring Agency Code

15. Supplementary Notes

Final Report, User's Manual. Project Manager, D. Schultz, Mission Assessment and Applications Branch, NASA Lewis Research Center, Cleveland, Ohio 44135.

16. Abstract

The documentation of the Trajectory Generation and System Characterization Model for the Cislunar Low-Thrust Spacecraft is presented in Technical and User's Manuals. The system characteristics and trajectories of low-thrust nuclear electric propulsion spacecraft can be generated through the use of multiple system technology models coupled with a high-fidelity trajectory generation routine. The Earth to Moon trajectories utilize near Earth orbital plane alignment, midcourse control dependent upon the spacecraft's Jacobian constant, and capture to target orbit utilizing velocity matching algorithms. The trajectory generation is performed in a perturbed two-body equinoctial formulation and the restricted three-body formulation. A single control is determined by the user for the interactive midcourse portion of the trajectory. The full spacecraft system characteristics and trajectory are provided as output.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Earth-moon trajectory; Low-thrust spacecraft; Systems characterization; Trajectory model; Low-thrust guidance; Three dimensional trajectory	Unclassified – Unlimited Subject Category 20

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	52	A04

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161