

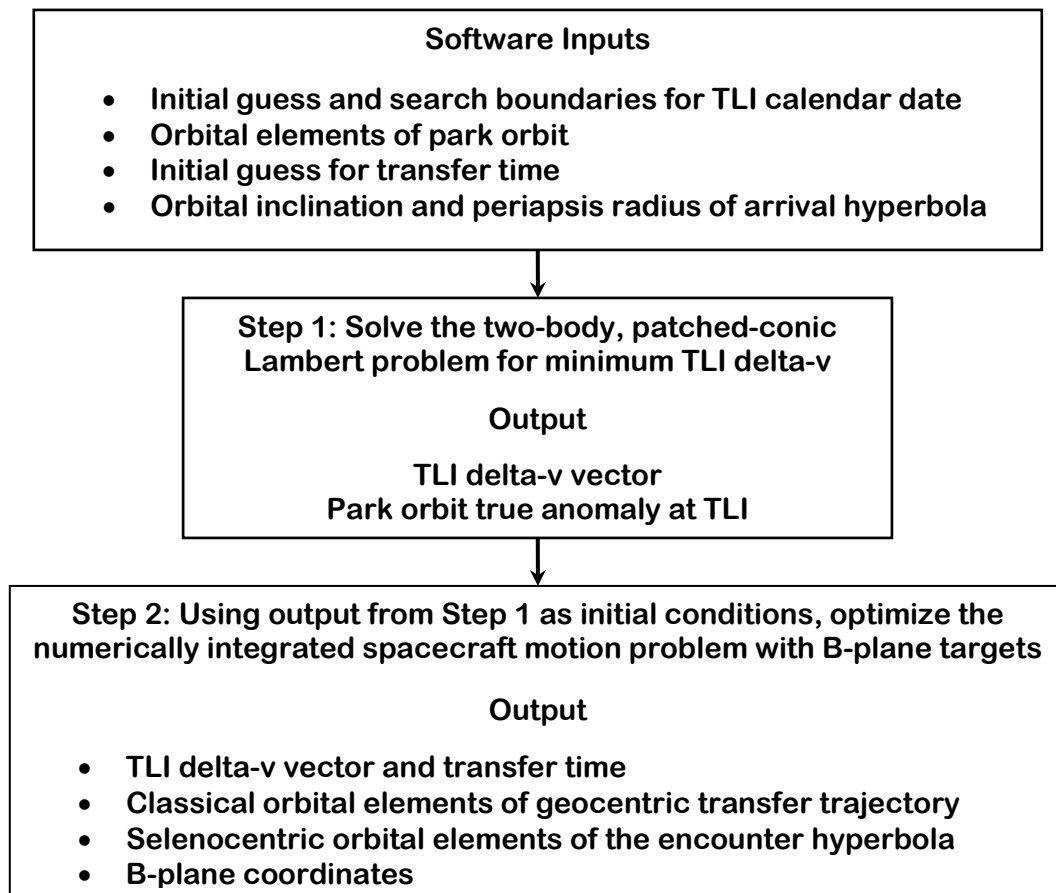
# A MATLAB Script for Preliminary Lunar Mission Design and Analysis

This document is the user's guide for a MATLAB script called `lunar1` that can be used to design preliminary lunar missions from Earth park orbit to B-plane encounter at the moon. The software assumes that trans-lunar injection (TLI) occurs *impulsively* from a circular Earth park orbit. The B-plane coordinates are expressed in a moon-centered (selenocentric) mean equator and IAU node of epoch coordinate system. The results from this scientific simulation can be used as an initial guess for a more sophisticated trajectory optimization program.

The first part of this MATLAB script solves for the minimum TLI delta-v using a two-body Lambert solution for the transfer trajectory from the Earth park orbit to the center of the moon. The second part of the script implements a *shooting* method that attempts to minimize the TLI delta-v while numerically integrating the spacecraft equations of motion and targeting to user-defined constraints.

In the shooting algorithm, the spacecraft motion model includes the Earth's oblate gravity effect and the point-mass perturbations of the sun and moon. The B-plane targets are enforced via a user-defined periapsis radius and orbital inclination of the arrival hyperbola relative to the moon.

The program inputs and major computational steps implemented in this script are as follows:



This MATLAB script uses the SNOPT nonlinear programming (NLP) method for both optimization algorithms of the lunar transfer problem implemented in this script. The lunar coordinates required by this script are computed using the JPL DE421 ephemeris.

## Input data file

The lunar1 MATLAB script is “data-driven” by a simple text file created by the user. This section describes a typical input data file. In the following discussion the actual input file contents are in *courier* font and all explanations are in *times italic* font.

Each data item within an input file is preceded by one or more lines of *annotation* text. Do not delete any of these annotation lines or increase or decrease the number of lines reserved for each comment. However, you may change them to reflect your own explanation. The annotation line also includes the correct units and when appropriate, the valid range of the input.

*The first four lines of any input file are reserved for user comments. These lines are ignored by the software. However the input file must begin with four and only four initial text lines.*

```
*****  
* input file for lunar1.m  
* lunar1.in - April 4, 2005  
*****
```

*The software allows the user to specify an initial guess for the TLI calendar date and lower and upper bounds on the actual date found during both the two-body and numerically integrated TLI delta-v optimization processes. For any guess for the TLI time  $t_{TLI}$  and user-defined lower and upper bounds  $\Delta t_l$  and  $\Delta t_u$ , the actual TLI time  $t$  is constrained as follows:*

$$t_{TLI} - \Delta t_l \leq t \leq t_{TLI} + \Delta t_u$$

*The first five inputs define the initial guesses for the TLI calendar date and the lower and upper bounds, respectively. Be sure to include all four digits of the calendar year. The first set of bounds are used during the two-body optimization, and the second set is used during the numerical integration and b-plane targeting.*

*The TLI calendar date is a control variable in the NLP formulation and must always have a lower and upper bound. For a fixed TLI calendar date, input small values (e.g., plus and minus 1.0e-8) for the bounds.*

```
initial guess for TLI calendar date (month, day, year)  
10,1,2008  
  
lower bound for TLI calendar date search (two-body optimization; hours)  
-240.0  
  
upper bound for TLI calendar date search (two-body optimization; hours)  
+24.0  
  
lower bound for TLI calendar date search (b-plane optimization; hours)  
-12.0  
  
upper bound for TLI calendar date search (b-plane optimization; hours)  
+12.0
```

*The next input is the user’s initial guess for the TLI-to-B-plane transfer time, in hours.*

```
initial guess for transfer time (hours)  
96.0
```

*The next two numbers define the fixed values for park orbit altitude and orbital inclination.*

```
*****  
circular park orbit characteristics  
*****  
  
altitude (kilometers)  
185.32  
  
orbital inclination (degrees)  
28.5
```

*This next integer input defines the type of TLI maneuver to perform. The script uses this indicator to compute the park orbit RAAN.*

```
type of TLI maneuver  
(1 = ascending, 2 = descending)  
2
```

*The next two inputs define the periapsis radius and orbital inclination to use during the numerically integrated solution. These coordinates refer to the selenocentric hyperbola. The orbital inclination should be specified in the lunar equator and equinox of J2000 coordinate system.*

```
*****  
final lunar orbit characteristics  
(lunar equator and equinox of J2000)  
*****  
  
periapsis radius (kilometers)  
1838.0  
  
orbital inclination (degrees)  
90.0
```

*This next integer input defines the type of targeting algorithm to use. The b-plane algorithm is recommended with the orbital elements algorithm a backup in case the software has trouble establishing a hyperbolic orbit encounter during the shooting calculations.*

```
type of targeting  
(1 = b-plane, 2 = orbital elements, 3 = user-defined b-plane targets)  
1
```

*For option 3, the software allows the user to input b-plane targets directly using the following two program inputs.*

```
user-defined b dot r target (kilometers)  
6000.0  
  
user-defined b dot t target (kilometers)  
-10.0
```

*The last three integer inputs define the types of perturbations to include during the numerical integration of the spacecraft's motion. The first option will include the effect of  $J_2$  in the equations of motion, and options 2 and 3 will include the point mass gravity of the moon and sun.*

```
*****  
trajectory perturbations  
*****  
  
include Earth J2 perturbation (1 = yes, 0 = no)  
1
```

```
include lunar perturbation (1 = yes, 0 = no)
1

include solar perturbation (1 = yes, 0 = no)
1
```

## Program example and trajectory graphics

The following is the solution created by the computer program for this example. The output is organized by the following major sections:

- First pass
  1. two body Lambert solution
  2. TLI delta-v vector and magnitude
- Targeting pass
  1. pre-TLI and post-TLI flight conditions
  2. TLI delta-v vector and magnitude
  3. time and conditions at lunar closest approach
  4. classical orbital elements of the lunar transfer trajectory

*The first output section summarizes the two-body Lambert solution. The solution is presented in the Earth mean equator of J2000 coordinate system (EME2000). The trajectory characteristics are given before and after the TLI maneuver.*

```
*****
lunar1 - Matlab version
*****

input data file = lunar1.in

descending TLI maneuver

b-plane targeting

-----
minimum TLI delta-v - two-body Lambert solution
-----

transfer time                96.000000  hours

time and conditions prior to TLI maneuver
(geocentric - EME2000 coordinates)
-----

TLI calendar date           24-Sep-2008

TLI dynamical time         13:11:15.197

      sma (km)          eccentricity          inclination (deg)          argper (deg)
+6.563336300000000e+003  +1.11295133887758e-016  +2.850000000000000e+001  +0.000000000000000e+000

      raan (deg)          true anomaly (deg)          arglat (deg)          period (min)
```

```

+1.77705546439843e+002 +1.80000000000000e+002 +1.80000000000000e+002 +8.81955595103076e+001
      rx (km)              ry (km)              rz (km)              rmag (km)
+6.55807431141161e+003 -2.62763606499976e+002 +3.83529179181631e-013 +6.56333630000000e+003
      vx (kps)              vy (kps)              vz (kps)              vmag (kps)
+2.74186215617689e-001 +6.84316066877278e+000 -3.71851431328934e+000 +7.79303373004450e+000

```

time and conditions after TLI maneuver  
(geocentric - EME2000 coordinates)

```

-----
TLI calendar date      24-Sep-2008
TLI dynamical time    13:11:15.197

```

```

      sma (km)              eccentricity          inclination (deg)      argper (deg)
+2.03941589927772e+005 +9.67817574760753e-001 +2.85000000000000e+001 +1.80051991171913e+002
      raan (deg)           true anomaly (deg)   arglat (deg)          period (min)
+1.77705546439843e+002 +3.59948008828087e+002 +1.80000000000000e+002 +1.52763089934016e+004
      rx (km)              ry (km)              rz (km)              rmag (km)
+6.55807431141161e+003 -2.62763606499976e+002 +3.83529179181631e-013 +6.56333630000000e+003
      vx (kps)              vy (kps)              vz (kps)              vmag (kps)
+3.79750520197056e-001 +9.59970631761018e+000 -5.21629123717111e+000 +1.09319836312054e+001

```

TLI delta-v vector and magnitude  
(geocentric - EME2000 coordinates)

```

-----
x-component of delta-v      105.564305 meters/second
y-component of delta-v      2756.545649 meters/second
z-component of delta-v      -1497.776924 meters/second

total delta-v                3138.952604 meters/second

```

*This section of the program output is created after the B-plane targeting problem has been solved. It includes a summary of the solution, the TLI delta-v vector and magnitude, the final B-plane coordinates and the orbital elements and state vector of the encounter hyperbola.*

```

*****
numerically integrated solution
*****

```

```

transfer time                110.891940 hours

```

time and conditions prior to TLI maneuver  
(geocentric - EME2000 coordinates)

```

-----
TLI calendar date      24-Sep-2008
TLI dynamical time    01:11:15.197

```

```

      sma (km)              eccentricity          inclination (deg)      argper (deg)
+6.56333630000000e+003 +1.11295133887758e-016 +2.85000000000000e+001 +0.00000000000000e+000
      raan (deg)           true anomaly (deg)   arglat (deg)          period (min)
+1.84017239949538e+002 +1.74855546792525e+002 +1.74855546792525e+002 +8.81955595103076e+001
      rx (km)              ry (km)              rz (km)              rmag (km)
+6.55706987643078e+003 -5.79732872022432e+001 +2.80815100150813e+002 +6.56333630000000e+003
      vx (kps)              vy (kps)              vz (kps)              vmag (kps)

```

+2.19200741591069e-001 +6.85325845368647e+000 -3.70353538115570e+000 +7.79303373004449e+000

time and conditions after TLI maneuver  
(geocentric - EME2000 coordinates)

-----  
TLI calendar date 24-Sep-2008

TLI dynamical time 01:11:15.197

sma (km)	eccentricity	inclination (deg)	argper (deg)
+2.02860158648795e+005	+9.67646006296600e-001	+2.84994858145553e+001	+1.74858455278079e+002
raan (deg)	true anomaly (deg)	arglat (deg)	period (min)
+1.84017336967008e+002	+3.59997006253626e+002	+1.74855461531705e+002	+1.51549627883683e+004
rx (km)	ry (km)	rz (km)	rmag (km)
+6.55706987643078e+003	-5.79732872022432e+001	+2.80815100150813e+002	+6.56333630000000e+003
vx (kps)	vy (kps)	vz (kps)	vmag (kps)
+3.07195112464278e-001	+9.61330728710483e+000	-5.19497844897032e+000	+1.09315070744473e+001

c3 -1.96490252 (km/sec)^2

TLI delta-v vector and magnitude  
(geocentric - EME2000 coordinates)

-----  
x-component of delta-v 87.994371 meters/second  
y-component of delta-v 2760.048833 meters/second  
z-component of delta-v -1491.443068 meters/second  
  
total delta-v 3138.473354 meters/second

time and conditions at lunar closest approach  
(selenocentric - lunar equator and IAU node of epoch)

-----  
calendar date 28-Sep-2008

dynamical time 16:04:46.180

sma (km)	eccentricity	inclination (deg)	argper (deg)
-6.00589525113263e+003	+1.30603264088309e+000	+9.00000000887733e+001	+1.25487004202640e+002
raan (deg)	true anomaly (deg)	arglat (deg)	period (min)
+2.68900344358660e+002	+3.59712259978551e-013	+1.25487004202640e+002	+1.66666500000000e+003
rx (km)	ry (km)	rz (km)	rmag (km)
+2.04771149384415e+001	+1.06679608552031e+003	+1.49658636335048e+003	+1.83799998457134e+003
vx (kps)	vy (kps)	vz (kps)	vmag (kps)
+3.87566099375694e-002	+2.01910243853986e+000	-1.43978607183829e+000	+2.48017351505716e+000

flight path angle 2.0004757244e-013 degrees

b-plane coordinates of incoming hyperbola  
(selenocentric - lunar equator and IAU node of epoch)

-----  
b-magnitude 5045.385486 kilometers  
b dot r -5045.385486  
b dot t -0.000008  
b-plane angle 270.000000 degrees

v-infinity	903.510524	meters/second
r-periapsis	1837.999985	kilometers
decl-asymptote	14.480312	degrees
rasc-asymptote	88.900344	degrees

The `lunar1` script can also create graphic displays of the geocentric transfer orbit and the selenocentric hyperbola within the sphere-of-influence of the moon. This section describes the user interaction with this software option and also provides typical graphic displays for this example. After the software has computed the trajectory solution, it will display the following graphics menu:

```
graphics menu
<1> transfer trajectory
<2> motion within the lunar SOI
<3> both flight phases
<4> none
?
```

If the user elects option 1, 2 or 3, the script will request the plot duration with this next menu:

```
plot duration menu
<1> time of closest approach
<2> user-defined duration
?
```

This menu allows the user to create graphics from trans-lunar injection until either the time of closest approach to the moon or a simulation duration defined by the user.

For the second plot duration option, the software will ask for this input with the following request:

```
please input the simulation duration (hours)
?
```

Finally, the MATLAB script will request the plot step size with

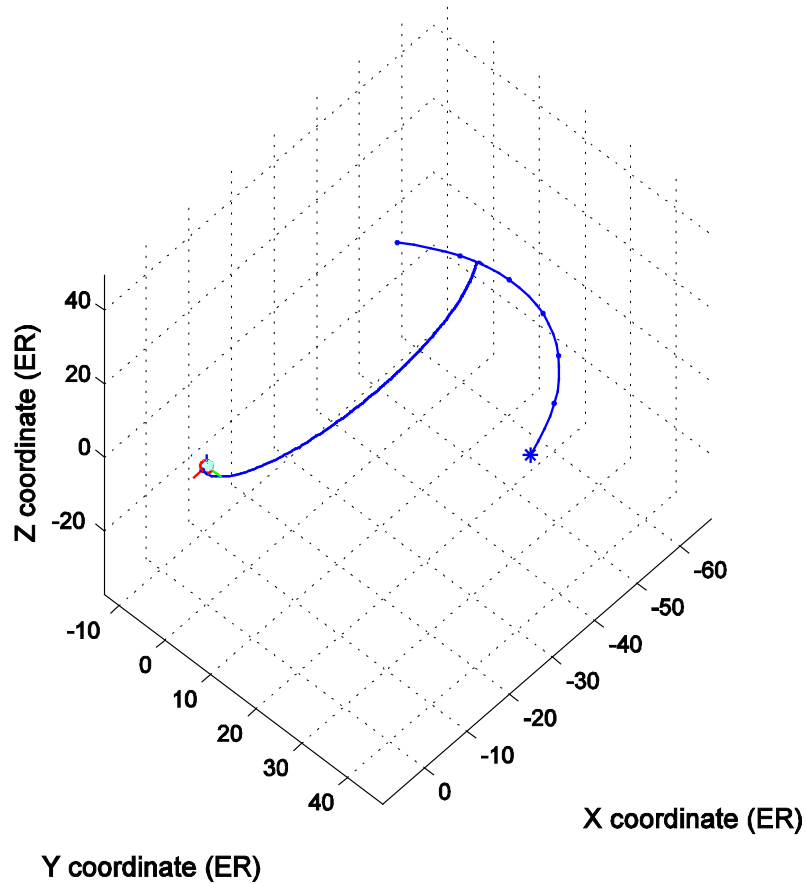
```
please input the plot step size (minutes)
?
```

A plot step size of ten minutes is recommended. However, for zoomed plots of either the geocentric or selenocentric trajectory, smaller plot step sizes will create “smoother” trajectory displays.

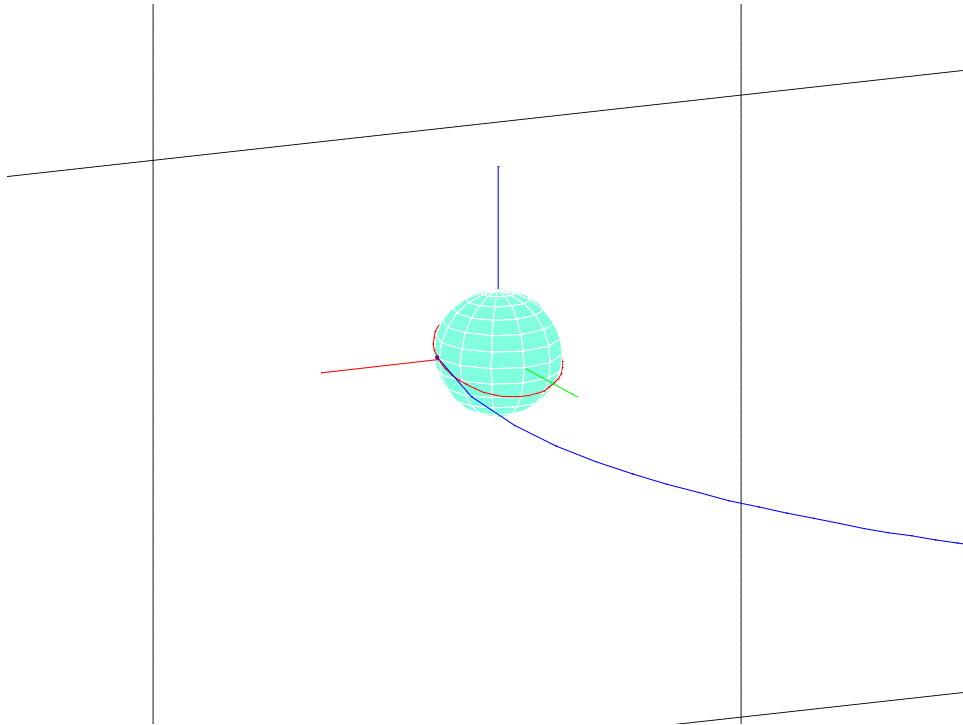
The following is a plot of the geocentric transfer trajectory for this example. Please note that the coordinates are displayed in the units of Earth radii (ER). The asterisk symbol is the position of the moon at the moment of trans-lunar injection and the moon’s orbit is marked with a small blue dot symbol at 24 hour intervals. The park orbit trace is red.

The interactive graphic features of MATLAB will allow the user to rotate and “zoom” the displays in and out. These capabilities allow the user to interactively find the “best” viewpoint as well as verify the basic orbital geometry of the geocentric and selenocentric trajectories.

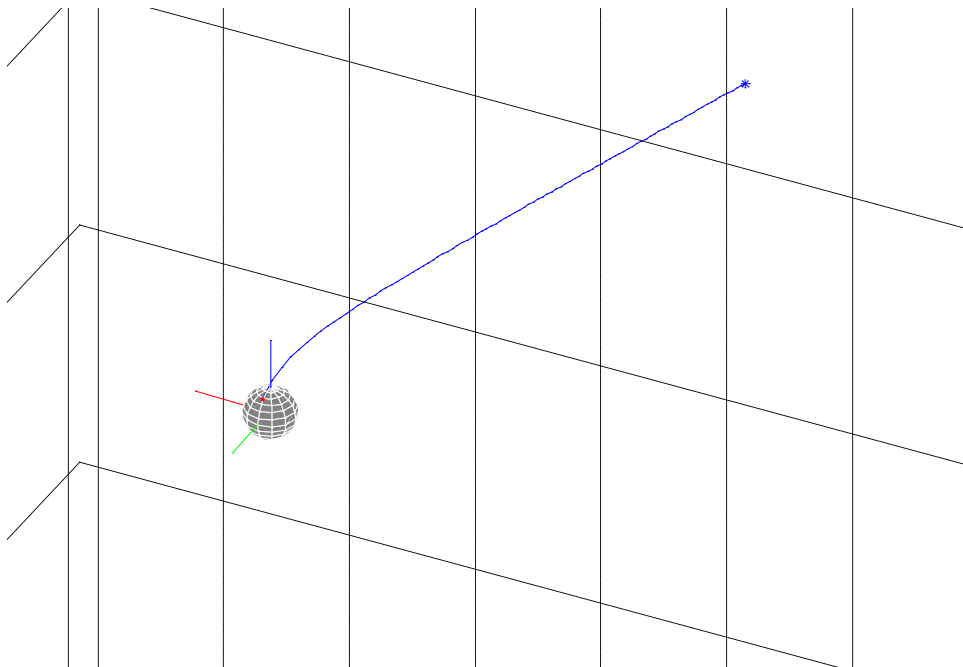
### Geocentric Transfer Trajectory



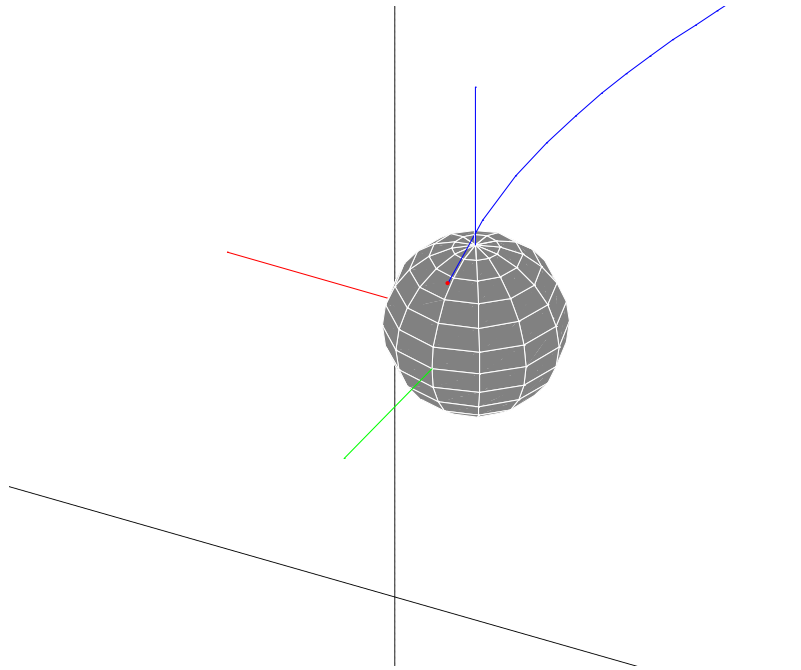
This next plot is a “zoomed” display of the first plot closer to the Earth. The initial park orbit is displayed in red, the transfer trajectory is blue, and an inertial, Earth-centered coordinate system is on the plot. The x-axis of this system is red, the y-axis green and the z-axis blue. The location on the park orbit at which TLI occurs is marked with a small dot symbol.



The following is a plot of the selenocentric hyperbola within the moon's sphere-of-influence. The coordinate units are lunar radii (LR). The entry into the SOI is marked with an asterisk. In this MATLAB script the radius of the moon's SOI is "hardwired" to a value of 64,000 kilometers.



The final plot is a "zoomed" display of the previous plot. This display is labeled with a selenocentric, inertial coordinate system. The x-axis is red, the y-axis green and the z-axis blue. The small red dot is periapsis of the approach hyperbola.



The `lunar1` MATLAB script can also create these types of graphic displays for the two-body solution.

## Technical Discussion

This section provides additional details about the numerical algorithms used in this computer program. The computational methods discussed here include solving the two body Lambert problem, the method used for propagating the spacecraft's geocentric trajectory, the algorithm used for targeting to the B-plane, and the geocentric-to-selenocentric coordinate transformation.

### *Nonlinear programming problem*

A trajectory optimization problem can be described by a system of *dynamic variables*

$$\mathbf{z} = \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{u}(t) \end{bmatrix}$$

consisting of the *state variables*  $\mathbf{y}$  and the *control variables*  $\mathbf{u}$  for any time  $t$ . In this discussion vectors are denoted in bold.

The system dynamics are defined by a vector system of ordinary differential equations called the *state equations* that can be represented as follows:

$$\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t]$$

where  $\mathbf{p}$  is a vector of problem *parameters* that is not time dependent.

The initial dynamic variables at time  $t_0$  are defined by  $\boldsymbol{\psi}_0 \equiv \boldsymbol{\psi}[\mathbf{y}(t_0), \mathbf{u}(t_0), t_0]$  and the terminal conditions at the final time  $t_f$  are defined by  $\boldsymbol{\psi}_f \equiv \boldsymbol{\psi}[\mathbf{y}(t_f), \mathbf{u}(t_f), t_f]$ . These conditions are called the *boundary values* of the trajectory problem.

The problem may also be subject to *path constraints* of the form  $\mathbf{g}[\mathbf{y}(t), \mathbf{u}(t), t] = 0$ .

For any mission time  $t$  there are also simple bounds on the state variables

$$\mathbf{y}_l \leq \mathbf{y}(t) \leq \mathbf{y}_u$$

the control variables

$$\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u$$

and the problem parameters

$$\mathbf{p}_l \leq \mathbf{p}(t) \leq \mathbf{p}_u$$

The basic nonlinear programming problem (NLP) is to determine the control vector history and problem parameters that minimize the scalar performance index or objective function given by

$$J = \phi[\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f, \mathbf{p}]$$

while satisfying all the user-defined mission constraints.

During the two-body trajectory optimization, the control variables are the TLI calendar date and the true anomaly of the TLI maneuver. For the numerical integration optimization, the control variables consist of the TLI calendar date, the RAAN of the park orbit, the true anomaly of the TLI maneuver, and the Cartesian components of the TLI delta-v vector.

For both types of optimization, the objective function or performance index is the scalar magnitude of the TLI delta-v.

In addition to the bounds on the TLI calendar date mentioned earlier, the true anomaly during the two-body optimization is bounded according to

$$-180^\circ \leq \theta \leq +180^\circ$$

During the second part of the trajectory optimization, the RAAN and true anomaly bounds are

$$\Omega_{TB} - 30^\circ \leq \Omega \leq \Omega_{TB} + 30^\circ$$

$$\theta_{TB} - 30^\circ \leq \theta \leq \theta_{TB} + 30^\circ$$

where  $\Omega_{TB}$  and  $\theta_{TB}$  are the RAAN and true anomaly found during the two-body optimization. The bounds on the components of the TLI delta-v are given by

$$-(|\Delta\mathbf{v}| + 0.1|\Delta\mathbf{v}|) \leq \Delta\mathbf{v}_{x,y,z} \leq +(|\Delta\mathbf{v}| + 0.1|\Delta\mathbf{v}|)$$

where  $|\Delta\mathbf{v}|$  is the scalar magnitude of the two-body TLI delta-v vector.

The final boundary conditions are the B-plane coordinates of the incoming selenocentric hyperbola.

### *Solving the two body Lambert problem*

Lambert's problem is concerned with the determination of an orbit that passes between two positions within a specified time-of-flight. This classic astrodynamics problem is also known as the orbital two-point boundary value problem (TPBVP).

The time to traverse a trajectory depends only upon the length of the semimajor axis  $a$  of the transfer trajectory, the sum  $r_i + r_f$  of the distances of the initial and final positions relative to a central body, and the length  $c$  of the chord joining these two positions. This relationship can be stated as follows:

$$tof = tof(r_i + r_f, c, a)$$

From the following form of Kepler's equation

$$t - t_0 = \sqrt{\frac{a^3}{\mu}} (E - e \sin E)$$

we can write

$$t = \sqrt{\frac{a^3}{\mu}} [E - E_0 - e(\sin E - \sin E_0)]$$

where  $E$  is the eccentric anomaly associated with radius  $r$ ,  $E_0$  is the eccentric anomaly at  $r_0$ , and  $t = 0$  when  $r = r_0$ .

At this point we need to introduce the following trigonometric sum and difference identities:

$$\begin{aligned} \sin \alpha - \sin \beta &= 2 \sin \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} \\ \cos \alpha - \cos \beta &= -2 \sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2} \\ \cos \alpha + \cos \beta &= 2 \cos \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} \end{aligned}$$

If we let  $E = \alpha$  and  $E_0 = \beta$  and substitute the first trig identity into the second equation above, we have the following equation:

$$t = \sqrt{\frac{a^3}{\mu}} \left\{ E - E_0 - 2 \sin \frac{E - E_0}{2} \left( e \cos \frac{E + E_0}{2} \right) \right\}$$

With the two substitutions given by

$$e \cos \frac{E + E_0}{2} = \cos \frac{\alpha + \beta}{2}$$

$$\sin \frac{E - E_0}{2} = \sin \frac{\alpha - \beta}{2}$$

the time equation becomes

$$t = \sqrt{\frac{a^3}{\mu}} \left\{ (\alpha - \beta) - 2 \sin \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} \right\}$$

From the elliptic relationships given by

$$r = a(1 - e \cos E)$$

$$x = a(\cos E - e)$$

$$y = a \sin E \sqrt{1 - e^2}$$

and some more manipulation, we have the following equations:

$$\cos \alpha = \left( 1 - \frac{r + r_0}{2a} \right) - \frac{c}{2a} = 1 - \frac{r + r_0 + c}{2a} = 1 - \frac{s}{a}$$

$$\sin \beta = \left( 1 - \frac{r + r_0}{2a} \right) + \frac{c}{2a} = 1 - \frac{r + r_0 - c}{2a} = 1 - \frac{s - c}{a}$$

This part of the derivation makes use of the following three relationships:

$$\cos \frac{\alpha - \beta}{2} \cos \frac{\alpha + \beta}{2} = 1 - \frac{r + r_0}{2}$$

$$\sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2} = \sin \frac{E - E_0}{2} \sqrt{1 - \left( e \cos \frac{E + E_0}{2} \right)^2}$$

$$\left( \sin \frac{\alpha - \beta}{2} \sin \frac{\alpha + \beta}{2} \right)^2 = \left( \frac{x - x_0}{2a} \right)^2 + \left( \frac{y - y_0}{2a} \right)^2 = \left( \frac{c}{2a} \right)^2$$

With the use of the half angle formulas given by

$$\sin \frac{\alpha}{2} = \sqrt{\frac{s}{2a}} \quad \sin \frac{\beta}{2} = \sqrt{\frac{s-c}{2a}}$$

and several additional substitutions, we have the time-of-flight form of Lambert's theorem

$$t = \sqrt{\frac{a^3}{\mu}} [(\alpha - \beta) - (\sin \alpha - \sin \beta)]$$

A discussion about the angles  $\alpha$  and  $\beta$  can be found in "Geometrical Interpretation of the Angles  $\alpha$  and  $\beta$  in Lambert's Problem" by J. E. Prussing, *AIAA Journal of Guidance and Control*, Volume 2, Number 5, Sept.-Oct. 1979, pages 442-443.

The algorithm used in this MATLAB script is based on the method described in "A Procedure for the Solution of Lambert's Orbital Boundary-Value Problem" by R. H. Gooding, *Celestial Mechanics and Dynamical Astronomy* **48**: 145-165, 1990. This iterative solution is valid for elliptic, parabolic and hyperbolic transfer orbits which may be either posigrade or retrograde, and involve one or more revolutions about the central body.

#### *Propagating the spacecraft's trajectory*

This part of the trajectory analysis implements a *special perturbation* technique which numerically integrates the vector system of second-order, nonlinear differential equations of motion of a spacecraft given by

$$\vec{a}(\vec{r}, \vec{v}, t) = \vec{r}''(\vec{r}, \vec{r}', t) = \vec{a}_g(\vec{r}) + \vec{a}_m(\vec{r}, t) + \vec{a}_s(\vec{r}, t)$$

where

$t$  = dynamical time

$\vec{r}$  = inertial position vector of the satellite

$\vec{v}$  = inertial velocity vector of the satellite

$\vec{a}_g$  = acceleration due to the Earth's gravity

$\vec{a}_m$  = acceleration due to the Moon

$\vec{a}_s$  = acceleration due to the Sun

The system of six first-order differential equations is defined by

$$\dot{y}_1 = v_x = y_4$$

$$\dot{y}_2 = v_y = y_5$$

$$\dot{y}_3 = v_z = y_6$$

$$\dot{y}_4 = -\mu \frac{r_x}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 1 - \frac{5 r_z^2}{r^2} \right) \right\}$$

$$\dot{y}_5 = -\mu \frac{r_y}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 1 - \frac{5 r_z^2}{r^2} \right) \right\}$$

$$\dot{y}_6 = -\mu \frac{r_z}{r^3} \left\{ 1 + \frac{3 J_2 r_{eq}^2}{2 r^2} \left( 3 - \frac{5 r_z^2}{r^2} \right) \right\}$$

where  $r = \sqrt{r_x^2 + r_y^2 + r_z^2} = \sqrt{y_1^2 + y_2^2 + y_3^2}$ . In these equations  $\mu$  and  $r_{eq}$  are the gravitational constant and equatorial radius of the Earth, respectively and  $J_2$  is the oblateness gravity coefficient.

The acceleration contribution of the moon represented by a *point mass* is given by

$$\vec{a}_m(\vec{r}, t) = -\mu_m \left( \frac{\vec{r}_{m-b}}{|\vec{r}_{m-b}|^3} + \frac{\vec{r}_{e-m}}{|\vec{r}_{e-m}|^3} \right)$$

where

$\mu_m$  = gravitational constant of the moon

$\vec{r}_{m-b}$  = position vector from the moon to the satellite

$\vec{r}_{e-m}$  = position vector from the Earth to the moon

The acceleration contribution of the sun represented by a *point masses* is given by

$$\vec{a}_s(\vec{r}, t) = -\mu_s \left( \frac{\vec{r}_{s-b}}{|\vec{r}_{s-b}|^3} + \frac{\vec{r}_{e-s}}{|\vec{r}_{e-s}|^3} \right)$$

where

$\mu_s$  = gravitational constant of the sun

$\vec{r}_{s-b}$  = position vector from the sun to the satellite

$\vec{r}_{e-s}$  = position vector from the Earth to the sun

In this MATLAB script the heliocentric coordinates of the sun and moon are based on the JPL Development Ephemeris DE421. These coordinates are provided in the Earth mean equator and equinox of J2000 coordinate system (EME2000).

### *Park orbit RAAN*

For a given TLI injection time, there are two possible locations on the initial park orbit at which to perform the propulsive maneuver. One opportunity occurs during the ascending part of the park orbit and the other during the descending motion. The park orbit RAAN  $\Omega_p$  at these two locations can be

determined from spherical trigonometry relationships involving the park orbit inclination and the right ascension and declination of the moon at encounter.

ascending

$$\Omega_p = -180^\circ + \alpha_m + \sin^{-1} \left( \frac{\tan \delta_m}{\tan i_p} \right)$$

descending

$$\Omega_p = \alpha_m - \sin^{-1} \left( \frac{\tan \delta_m}{\tan i_p} \right)$$

where

$\alpha_m$  = right ascension of the moon at encounter

$\delta_m$  = declination of the moon at encounter

$i_p$  = park orbit inclination

These opportunities are valid whenever  $|\delta_m| \leq i_p$ .

#### *Targeting to a Selenocentric Periapsis Radius and Orbital Inclination*

For this targeting option, the equality constraints enforced by the SNOPT nonlinear programming algorithm are

$$r_p - r_{ca} = 0$$

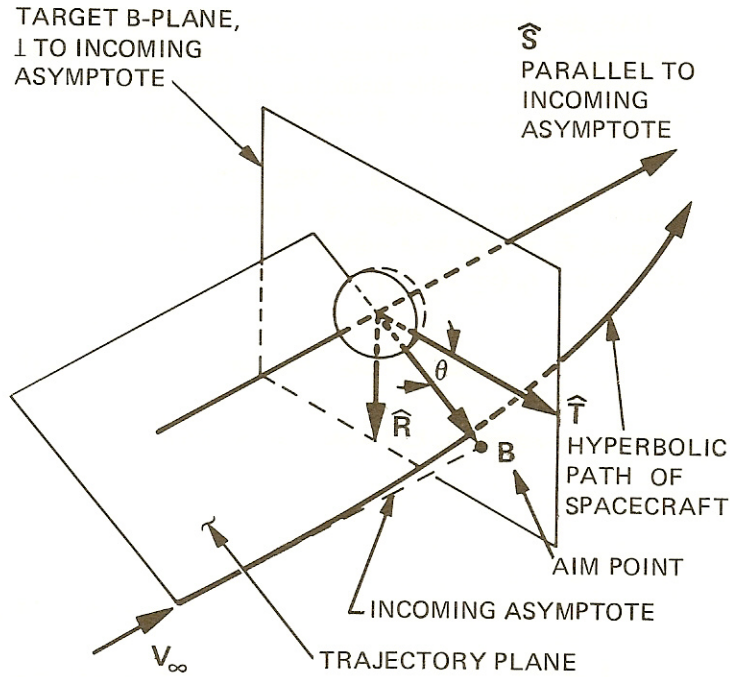
$$\cos i - \hat{\mathbf{h}}_z = 0$$

where  $r_p$  and  $i$  are the user-defined periapsis radius and selenocentric orbital inclination, respectively, and  $\hat{\mathbf{h}}_z$  is the z-component of the unit angular momentum vector at closest approach to the moon.

Closest approach is determined during the numerical integration of the spacecraft equations of motion by finding the time since TLI at which the selenocentric flight path angle is zero.

#### *B-plane targeting*

The derivation of B-plane coordinates is described in the classic JPL reports, “A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories” and “Some Orbital Elements Useful in Space Trajectory Calculations”, both by William Kizner. The following diagram illustrates the fundamental geometry of the B-plane coordinate system.



The software solves the B-plane targeting problem by minimizing the delta-v vector at the TLI while satisfying two nonlinear *equality constraint* equations. These constraint equations are the differences between components of the *required* B-plane and the B-plane components *predicted* by the software.

Given the user-defined closest approach radius  $r_{ca}$  and orbital inclination  $i$ , and the incoming v-infinity magnitude  $v_{\infty}$  and the right ascension  $\alpha_{\infty}$  and declination  $\delta_{\infty}$  of the incoming asymptote vector at the moment of closest approach, the following series of equations can be used to determine the *required* B-plane target components:

$$\mathbf{B} \cdot \mathbf{T} = b_t \cos \theta$$

$$\mathbf{B} \cdot \mathbf{R} = b_t \sin \theta$$

where

$$b_t = \sqrt{\frac{2\mu r_{ca}}{v_{\infty}^2} + r_{ca}^2} = r_{ca} \sqrt{1 + \frac{2\mu}{r_{ca} v_{\infty}^2}}$$

and

$$\cos \theta = \frac{\cos i}{\cos \delta_{\infty}}$$

$$\sin \theta = -\sqrt{1 - \cos^2 \theta}$$

$$\sin \delta_{\infty} = |\hat{\mathbf{s}} \times \hat{\mathbf{z}}| = \sqrt{s_x^2 + s_y^2}$$

$$\hat{\mathbf{z}} = [0 \ 0 \ 1]^T$$

The arrival asymptote unit vector  $\hat{\mathbf{S}}$  is given by

$$\hat{\mathbf{S}} = \begin{Bmatrix} \cos \delta_\infty \cos \alpha_\infty \\ \cos \delta_\infty \sin \alpha_\infty \\ \sin \delta_\infty \end{Bmatrix}$$

where  $\delta_\infty$  and  $\alpha_\infty$  are the declination and right ascension of the asymptote of the incoming hyperbola.

*Important note!!*

This technique only works for lunar orbit inclinations that satisfy

$$|i| > |\delta_\infty|$$

If this inequality is not satisfied, the software will print the following error message

```
b-plane targeting error!!
|inclination| must be > |asymptote declination|
```

It will also display the actual declination of the asymptote and stop. The user should then edit the input file, include a valid orbital inclination and restart the simulation.

The following computational steps summarize the calculation of the *predicted* B-plane vector from a moon-centered position vector  $\mathbf{r}$  and velocity vector  $\mathbf{v}$  at closest approach.

angular momentum vector

$$\mathbf{h} = \mathbf{r} \times \mathbf{v}$$

$$\hat{\mathbf{h}} = \frac{\mathbf{h}}{|\mathbf{h}|}$$

radius rate

$$\dot{r} = \frac{\mathbf{r} \cdot \mathbf{v}}{|\mathbf{r}|}$$

semiparameter

$$p = \frac{h^2}{\mu}$$

semimajor axis

$$a = \frac{r}{\left(2 - \frac{rv^2}{\mu}\right)}$$

orbital eccentricity

$$e = \sqrt{1 - p/a}$$

true anomaly

$$\cos \theta = \frac{p - r}{er}$$

$$\sin \theta = \frac{\dot{r}h}{e\mu}$$

B-plane magnitude

$$B = \sqrt{p|a|}$$

fundamental vectors

$$\hat{\mathbf{z}} = \frac{r\mathbf{v} - \dot{r}\mathbf{r}}{h}$$

$$\hat{\mathbf{p}} = \cos \theta \hat{\mathbf{r}} - \sin \theta \hat{\mathbf{z}}$$

$$\hat{\mathbf{q}} = \sin \theta \hat{\mathbf{r}} + \cos \theta \hat{\mathbf{z}}$$

S vector

$$\mathbf{S} = -\frac{a}{\sqrt{a^2 + b^2}} \hat{\mathbf{p}} + \frac{b}{\sqrt{a^2 + b^2}} \hat{\mathbf{q}}$$

B vector

$$\mathbf{B} = \frac{b^2}{\sqrt{a^2 + b^2}} \hat{\mathbf{p}} + \frac{ab}{\sqrt{a^2 + b^2}} \hat{\mathbf{q}}$$

T vector

$$\mathbf{T} = \frac{(S_y^2, -S_x^2, 0)^T}{\sqrt{S_x^2 + S_y^2}}$$

R vector

$$\mathbf{R} = \mathbf{S} \times \mathbf{T} = (-S_z T_y, S_z T_x, S_x T_y - S_y T_x)^T$$

The mission elapsed time at which the spacecraft reaches closest approach to the moon is predicted using the event prediction capability of the MATLAB `ode45` algorithm. During the numerical integration of the spacecraft's geocentric equations of motion, the `ode45` numerical method searches for the time at which the flight path angle *with respect to the moon* is nearly zero within a small tolerance. This constraint corresponds to closest approach to the moon. The *predicted* B-plane coordinates are based on the selenocentric flight conditions at closest approach. Close approach is predicted with the following *mission constraint*

$$\gamma = \sin^{-1} \left( \frac{\mathbf{r} \cdot \mathbf{v}}{|\mathbf{r} \cdot \mathbf{v}|} \right)$$

where  $\mathbf{r}$  and  $\mathbf{v}$  are the moon-centered position and velocity vectors, respectively.

### *Geocentric-to-selenocentric coordinate transformation*

This section describes the transformation of coordinates between the Earth mean equator and equinox 2000 (EME2000) and lunar mean equator and equinox of 2000 coordinate systems. This transformation is used to compute the B-plane coordinates at encounter.

A unit vector in the direction of the pole of the moon can be determined from

$$\hat{\mathbf{p}}_{Moon} = \begin{bmatrix} \cos \alpha_p \cos \delta_p \\ \sin \alpha_p \cos \delta_p \\ \sin \delta_p \end{bmatrix}$$

The right ascension and declination of the lunar pole in the EME2000 coordinate system are given by the following expressions

$$\begin{aligned} \alpha_p = & 269.9949 + 0.0031T - 3.8787 \sin E1 - 0.1204 \sin E2 \\ & + 0.0700 \sin E3 - 0.0172 \sin E4 + 0.0072 \sin E6 \\ & - 0.0052 \sin E10 + 0.0043 \sin E13 \end{aligned}$$

$$\begin{aligned} \delta_p = & 66.5392 + 0.0130T + 1.5419 \cos E1 + 0.0239 \cos E2 \\ & - 0.0278 \cos E3 + 0.0068 \cos E4 - 0.0029 \cos E6 \\ & + 0.0009 \cos E7 + 0.0008 \cos E10 - 0.0009 \cos E13 \end{aligned}$$

where  $T$  is the time in Julian centuries given by  $T = (JD - 2451545.0)/36525$  and  $JD$  is the TDB Julian Date.

The trigonometric arguments, in degrees, for these equations are

$$\begin{aligned} E1 &= 125.045 - 0.0529921d \\ E2 &= 250.089 - 0.1059842d \\ E3 &= 260.008 + 13.0120009d \\ E4 &= 176.625 + 13.3407154d \\ E6 &= 311.589 + 26.4057084d \\ E7 &= 134.963 + 13.0649930d \\ E10 &= 15.134 - 0.1589763d \\ E13 &= 25.053 + 12.9590088d \end{aligned}$$

where  $d = JD - 2451545$  is the number of days since January 1.5, 2000. These equations are given in “Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000”, *Celestial Mechanics and Dynamical Astronomy*, **82**: 83-110, 2002.

The unit vector in the x-axis direction of this selenocentric coordinate system is given by

$$\hat{\mathbf{x}} = \hat{\mathbf{z}} \times \hat{\mathbf{p}}_{Moon}$$

where  $\hat{\mathbf{z}} = [0 \ 0 \ 1]^T$ . The unit vector in the y-axis direction can be determined using

$$\hat{\mathbf{y}} = \hat{\mathbf{p}}_{Moon} \times \hat{\mathbf{x}}$$

Finally, the components of the matrix that transforms coordinates from the EME2000 system to the moon-centered (selenocentric) mean equator and equinox of 2000 system are as follows:

$$\mathbf{M} = [\hat{\mathbf{x}} \ \hat{\mathbf{y}} \ \hat{\mathbf{p}}_{Moon}]^T$$

## References and Bibliography

“Lunar Trajectories”, NASA TN D-866, August 1961.

“Earth-Moon Trajectories”, JPL Technical Report No. 32-503, May 1, 1964.

“Three-Dimensional Lunar Trajectories”, V. A. Egorov, Mechanics of Space Flight Series, Israel Program for Scientific Translations, Jerusalem 1969.

“Circumlunar Trajectory Calculations”, MIT Instrumentation Laboratory Report R-353, April 1962.

“Optimal Low Thrust Trajectories to the Moon”, John T. Betts and Sven O. Erb, *SIAM Journal on Applied Dynamical Systems*, Vol. 2, No. 2, pp. 144-170, 2003.

“Integrated Algorithm for Lunar Transfer Trajectories Using a Pseudostate Technique”, R. V. Ramanan, *AIAA Journal of Guidance, Control and Dynamics*, Vol. 25, No. 5, September-October 2002, pp. 946-952.