University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Masters Theses

Graduate School

5-2011

DEVELOPMENT AND ANALYSIS OF ONBOARD TRANSLUNAR INJECTION TARGETING ALGORITHMS

Phillippe Lyles Winters Reed
preed5@utk.edu

Recommended Citation

Reed, Phillippe Lyles Winters, "DEVELOPMENT AND ANALYSIS OF ONBOARD TRANSLUNAR INJECTION TARGETING ALGORITHMS. " Master's Thesis, University of Tennessee, 2011.
https://trace.tennessee.edu/utk_gradthes/907

To the Graduate Council:

I am submitting herewith a thesis written by Phillippe Lyles Winters Reed entitled "DEVELOPMENT AND ANALYSIS OF ONBOARD TRANSLUNAR INJECTION TARGETING ALGORITHMS." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aerospace Engineering.

Evans Lyne, Major Professor

We have read this thesis and recommend its acceptance:

Kivanc Ekici, Vasilios Alexiades

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

DEVELOPMENT AND ANALYSIS OF ONBOARD TRANSLUNAR INJECTION TARGETING ALGORITHMS

A Thesis Presented for
the Master of Science
Degree
The University of Tennessee, Knoxville

Phillippe Lyles Winters Reed
May 2011

To my wife, Emily

Non est ad astra mollis e terris via.

– Seneca the Younger, *Hercules Furens*, l. 437

ACKNOWLEDGEMENTS

I would like to thank the members of my thesis committee, Dr. Evans Lyne, Dr. Kivanc Ekici, and Dr. Vasilios Alexiades, for their assistance with my research and encouragement in my educational goals. I also wish to thank Dr. Greg Dukeman of Marshall Space Flight Center for the idea and motivation for this research and for answering my endless barrage of questions. Finally, I wish to express my gratitude to the Mechanical, Aerospace, and Biomedical Engineering Department for my teaching assistantship and funding for my graduate studies.

ABSTRACT

Several targeting algorithms are developed and analyzed for possible future use onboard a spacecraft. Each targeter is designed to determine the appropriate propulsive burn for translunar injection to obtain desired orbital parameters upon arrival at the moon. Primary design objectives are to minimize the computational requirements for each algorithm but also to ensure reasonable accuracy, so that the algorithm's errors do not force the craft to conduct large mid-course corrections. Several levels of accuracy for dynamical models are explored, the convergence range and speed of each algorithm are compared, and the possible benefits of the Broyden and trust-region targeters are evaluated. These targeters provide a proof of concept for the feasibility of a translunar injection targeting algorithm. Anticipating some future improvements, these algorithms could serve as a viable alternative to uploading ground-based targeting solutions and bypass the problems of delays and disruptions in communication, enabling the craft to conduct a translunar injection burn autonomously.

TABLE OF CONTENTS

Introduction 1

Design Requirements and Methodology 2

Dynamical Model 5

Equations of Motion and Reference Frames 8

Ephemeris 11

Assumptions 13

Initial Conditions 16

Predictor 19

Numerical Integration 19

Perilune 27

Sphere of Influence Method 28

Orbital Parameter Calculations 29

Initial Guesses 31

Correctors 37

Results 39

Conclusions and Recommendations 45

References 47

Appendix 49

Vita 53

LIST OF FIGURES

Figure 1. Relative Accelerations on Spacecraft between Earth and the Moon............................... 7

Figure 2. Inclination for Lunar Flybys. 14

Figure 3. Coplanar Lunar Transfer Departure Geometry. 14

Figure 4. Error in Position after One Period as a Function of Computation Time, Fixed Time-Step Solvers. 23

Figure 5. Error in Specific Angular Momentum after One Period as a Function of Computation Time, Fixed Time-Step Solvers. 24

Figure 6. Error in Velocity after One Period as a Function of Computation Time, Fixed Time-Step Solvers. 24

Figure 7. Error in Position after One Period as a Function of Computation Time, Variable Time-Step Solvers. 25

Figure 8. Error in Position after One Period as a Function of Computation Time, Select Solvers. 26

Figure 9. Error in Radius of Perilune for Kepler Assumption as a Function of Approach Distance (Sphere of Influence Radius). 29

Figure 10. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee. 33

Figure 11. Radius of Perilune for Varying ΔV and γ between 106º and 144º, Point-mass Gravitational Model, Departure at Lunar Perigee. 33

Figure 12. Inclination for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee. 34

Figure 13. Perilune Latitude for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee. 35

Figure 14. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Apogee. 36

Figure 15. Nominal Translunar Trajectory for Targeted r_p = 2605.5 km, λ_p = 3° (ECI Frame). ... 37

Figure 16: Nominal Translunar Trajectory for Targeted r_p = 2605.5 km, λ_p = 3° (ME Frame). ... 38

LIST OF TABLES

Table 1. Number of Broyden Targeter Iterations for Deviations in Initial Guess. 40

Table 2. Number of Gauss-Newton Targeter Iterations and Function Evaluations for Deviations in Initial Guess.. 41

Table 3. Number of Levenberg-Marquardt Targeter Iterations and Function Evaluations for Deviations in Initial Guess. 42

Table 4. Number of Trust-Region Targeter Iterations and Function Evaluations for Deviations in Initial Guess.. 43

LIST OF EQUATIONS

(1) 5

(2) 5

(3) 6

(4) 6

(5) 8

(6) 8

(7) 8

(8) 8

(9) 8

(10) 8

(11) 8

(12) 8

(13) 9

(14) 9

(15) 9

(16) 10

(17) 16

(18) 16

(19) 16

(20) 17

(21) 17

(22) 17

(23) 17

(24) 17

(25) 17

(26) 18

(27) 19

(28) 19

(29) 19

(30) 19

(31) 19

(32) 20

(33) 20

(34) 21

(35) 21

(36) 21

(37) 21

(38) 21

(39) 22

(40) 27

(41) 27

(42) 28

(43) 28

(44) 28

(45) 30

(46) 30

(47) 31

INTRODUCTION*

Previous lunar missions have relied heavily on ground-based computers to calculate required propulsive burns for translunar injection (TLI), and these calculations must then be uploaded to a spacecraft's guidance system. For instance, the Apollo missions used the so-called "hypersurface," a large series of polynomial curve-fits, to provide the targeting parameters to the guidance system.[1, 2, 3] Recently though, interest has been expressed in developing an efficient and robust TLI targeting algorithm for use onboard a spacecraft. This would allow the spacecraft to navigate more autonomously and would be particularly useful in case communication with the ground is disrupted. Apollo engineers originally considered an onboard targeting algorithm for TLI, but ultimately rejected it to simplify the requirements for the real-time computer complex.[4] Recently, Marchand et al. have explored similar onboard targeting techniques for the Orion three-maneuver trans-Earth phase.[5]

* Some portions of this paper were presented in, "An Onboard Targeting Algorithm for Lunar Missions" (Phillippe Reed, Greg Dukeman, and Evans Lyne) at the AAS/AIAA Spaceflight Mechanics Meeting in New Orleans, LA, 13-17 Feb. 2011.

DESIGN REQUIREMENTS AND METHODOLOGY

The purpose of this study was to develop and evaluate several variations on simple TLI targeting algorithms. The primary design requirements considered were computational speed, accuracy, and robustness. The algorithm must be simple enough to be used by onboard computers without requiring too much time or power, but accurate enough to require only small subsequent mid-course corrections. The algorithm must also be robust enough to start and operate autonomously for as wide a range of target parameters as possible.

The targeting algorithms were designed to calculate the required ΔV vector to depart from parking orbit around Earth, given the desired orbital parameters upon arrival at the moon (such as radius of perilune, latitude of perilune, inclination, etc.). To this end, a predictor-corrector technique was used. The predictor first assumes a velocity vector after TLI (i.e., a guess for an instantaneous change in velocity, ΔV vector) and the position vector of departure from the parking orbit. It then numerically integrates the equations of motion to determine the spacecraft's trajectory and the resulting orbital parameters about the moon. The calculated errors in orbital parameters from the initial guess constitute the objective function. The corrector then adjusts the initial state vectors to try to minimize the objective function (the errors), and the process is repeated until the required tolerance in each targeted orbital parameter is met.

Ideally, if the spacecraft's exact velocity, position, and time of arrival at the moon were prescribed by mission requirements, the launch time, position, and velocity could simply be determined by running one numerical integration backward in time until the spacecraft reaches the initial orbit location. In general, arrival parameters for a lunar mission need not be this strict, so there is often a wide range of possible departure times and velocities which must be considered. For the purposes of this investigation, the exact velocity and position vectors of the spacecraft and moon at perilune are considered irrelevant, and thus the inverse problem is useless (since there would be infinitely many suitable initial conditions). Further, the inverse problem would still have to target the initial Earth parking orbit parameters.

One major challenge for developing these targeting algorithms is balancing the two typically opposing criteria of accuracy and simplicity. The targeter must be accurate enough that it will not force the craft to expend too much fuel on mid-course corrections, and it must be simple enough that it can operate quickly and without excessive computational power. Several methods were used to address this problem.

Since the iterative corrector scheme must call on the predictor step to perform the numerical integration several times, the speed of the predictor is crucially important. Therefore, several methods to improve the predictor's performance were evaluated: a variety of numerical integration schemes were compared to determine which ones offer the greatest accuracy for the lowest computational time; a method of stopping the numerical integration short and then calculating a subsequent analytical solution was considered to limit the time required for integration; and different levels of sophistication in the gravitational model and in generating planetary ephemeris data for the equations of motion were considered. Similarly, several corrector methods were analyzed to generate an accurate TLI state vector with as few iterations and function evaluations as possible. Analytical and empirical approximations for a "good" initial guess were also used to try to reduce the required number of iterations and improve convergence.

It was intended that any reasonable predictor-corrector combinations would be tested against a high-fidelity gravitational model. This would have ensured that the simplifying assumptions and numerical methods in each algorithm were responsible for only small mid-course corrections, relative to other factors, such as sub-nominal engine burns. Also, it would have been beneficial to compare any targeter's results to optimized trajectories from pre-existing software packages (like NASA's Copernicus program), particularly when considering the ΔV requirement. These tests would have provided another means to compare each algorithm, but unfortunately, no such software packages could be obtained due to governmental restrictions and lack of funding.

DYNAMICAL MODEL

Several levels of sophistication were considered for the dynamical model to be used in the predictor. However, for simplicity, only gravitational forces were considered, either assuming spherically symmetric bodies or using spherical harmonics to model planetary oblateness. Both of these models ignore longitudinal gravitational asymmetries and non-uniform mass concentrations (mascons) in planetary bodies, atmospheric drag, radiation pressure, etc.

For either case, Newton's Law of Gravitation states that the gravitational force on any body (or any infinitesimal part of that body) from any other body is proportional to the product of their masses and inversely proportional to the square of the distance between them. For a system of N point masses or spherically symmetric bodies, the total gravitational force on any one body, $\mathbf{F}_i$, is equal to the sum of the forces acting on that body by every other body, so[*]

$$\mathbf{F}_i = -Gm_i \sum_{j=1, j\neq i}^{N} \frac{m_j}{\left\|\mathbf{r}_{ji}\right\|^3} \mathbf{r}_{ji} \tag{1}$$

where G is the gravitational constant of proportionality, m_i and m_j are the masses of body i and body j, respectively, and $\mathbf{r}_{ji}$ is the position vector from body j to body i. Ignoring other forces and assuming the mass of each body to be constant, the acceleration can be written using Newton's second law of motion as

$$\mathbf{a}_\mathrm{i} = \frac{d\mathbf{v}_i}{dt} = \frac{d^2\mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i} = -G \sum_{j=1, j\neq i}^{N} \frac{m_j}{\left\|\mathbf{r}_{ji}\right\|^3} \mathbf{r}_{ji} \tag{2}$$

Here $\mathbf{r}_i$ and $\mathbf{v}_i$ are the vectors indicating the position and velocity, respectively, of body i relative to the system's coordinate center, assuming an inertial reference frame. For most purposes, the

[*] See, for example, Ref. 11, pp. 5-9.

International Celestial Reference Frame (ICRF), tied to the solar system barycenter, can be used as such an inertial frame.

For the case of an oblate body modeled by spherical harmonics, the acceleration of an orbiting point mass is most conveniently determined in a local horizon coordinate frame. The radial component, a_r, acts toward the primary body's centers of mass and the transverse component, a_t, acts perpendicular to the radial component and northward or southward toward the equatorial plane. The magnitudes of these components can be derived by integrating the gravitational potential over all infinitesimal masses inside the oblate body, and the first few terms in the series are given by[6]

$$a_r = -\frac{Gm}{r^2}[1 - 3J_2\left(\frac{R}{r}\right)^2 P_2(cos\,\varphi) - 4J_3\left(\frac{R}{r}\right)^3 P_3(cos\,\varphi) - 5J_4\left(\frac{R}{r}\right)^4 P_4(cos\,\varphi) \qquad (3)$$

$$a_t = 3\frac{Gm}{r^2}\left(\frac{R}{r}\right)^2 sin\,\varphi\, cos\,\varphi\left[J_2 + \frac{1}{2}J_3\left(\frac{R}{r}\right) sec\,\varphi\,(5\,cos^2\varphi - 1)\right.$$
$$\left. + \frac{5}{6}J_4\left(\frac{R}{r}\right)^2 (7\,cos^2\varphi - 1)\right] \qquad (4)$$

Here *m* and *R* are the primary body's mass and radius, respectively, *r* is the distance between the point mass and the primary, and φ is the co-latitude of the point mass (the angle between the primary body's polar axis and the line from its center of mass to the point mass). The coefficients J_2, J_3, and J_4 are the Jeffrey's spherical harmonic constants unique to the primary body, and $P_n(cos\varphi)$ indicates the n-th order Legendre polynomial of $cos\varphi$. Note that if the *J* constants are ignored, the radial acceleration in Eq. (3) reduces to the magnitude of the acceleration for a single body in Eq. (2).

Using Eqs. (2) through (4), a comparison was made between the magnitude of maximum relative accelerations experienced by a spacecraft traveling between Earth and the moon from the gravity of various solar system objects. Figure 1 shows these accelerations from Earth, the sun, and the moon, and the maximum possible difference in acceleration caused by Earth's J_2 oblateness term (which occurs at $\varphi \approx 16.8^{o}$). The relative accelerations from other planets and the moon's oblateness were found to be several orders of magnitude smaller, as anticipated, and were not included in the graph or the gravitational model. Also as expected, the sun's gravity plays a significant role, contributing the highest relative acceleration of any object when the spacecraft is between about 0.7 and 0.9 lunar distances. Therefore the gravitational model was made to include the sun, as well as Earth and the moon. Although it decreases rapidly with distance from Earth, the acceleration from Earth's oblateness is significant near departure. This influence could cause considerable changes in the end state of a translunar trajectory. To determine the extent of this influence, the two separate dynamical models were both retained: one without any oblateness effects and one with J_2 through J_4 terms.

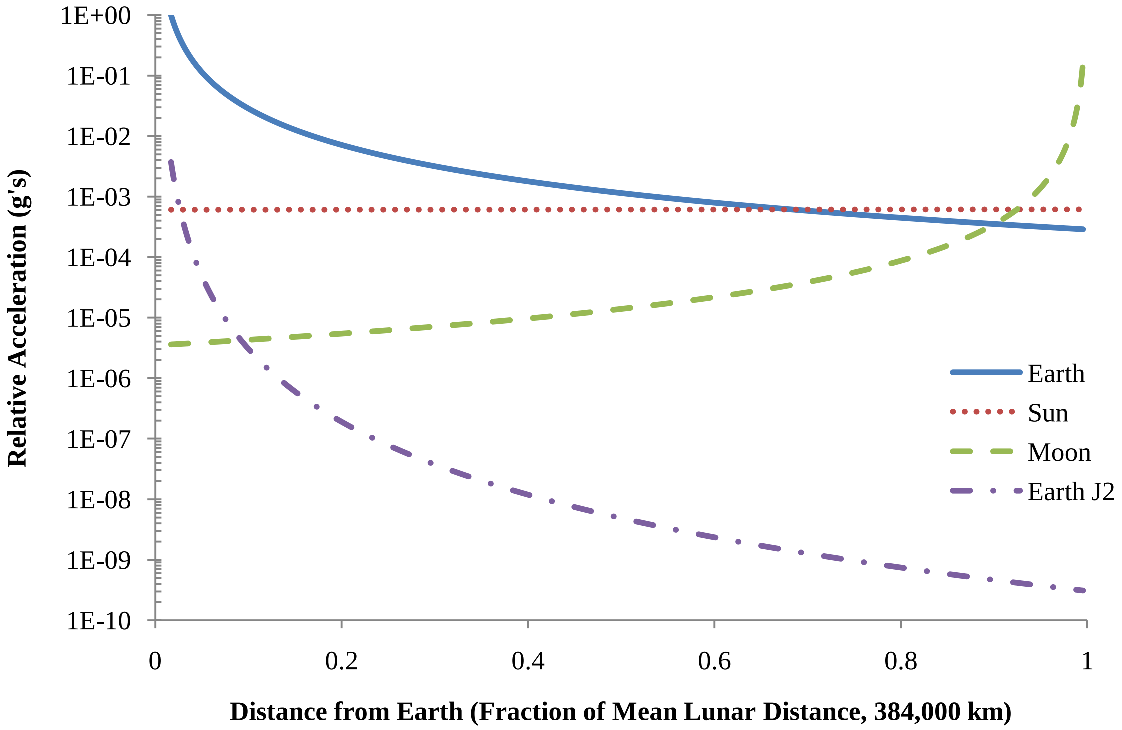

Figure 1. Relative Accelerations on Spacecraft between Earth and the Moon.

EQUATIONS OF MOTION AND REFERENCE FRAMES

For the spherical body assumption, with Earth, the moon, the sun, and the spacecraft denoted by the subscripts ‘E’, ‘M’, ‘S’, and ‘C’, respectively, the following system of equations (assuming an inertial reference frame) results:

$$\frac{d^2\mathbf{r}_E}{dt^2} = -G\frac{m_M}{\|\mathbf{r}_{ME}\|^3}\mathbf{r}_{ME} - G\frac{m_S}{\|\mathbf{r}_{SE}\|^3}\mathbf{r}_{SE} - G\frac{m_C}{\|\vec{r}_{CE}\|^3}\mathbf{r}_{CE} \tag{5}$$

$$\frac{d^2\mathbf{r}_M}{dt^2} = -G\frac{m_E}{\|\mathbf{r}_{EM}\|^3}\mathbf{r}_{EM} - G\frac{m_S}{\|\mathbf{r}_{SM}\|^3}\mathbf{r}_{SM} - G\frac{m_C}{\|\mathbf{r}_{CM}\|^3}\mathbf{r}_{CM} \tag{6}$$

$$\frac{d^2\mathbf{r}_S}{dt^2} = -G\frac{m_E}{\|\mathbf{r}_{ES}\|^3}\mathbf{r}_{ES} - G\frac{m_M}{\|\mathbf{r}_{MS}\|^3}\mathbf{r}_{MS} - G\frac{m_C}{\|\mathbf{r}_{CS}\|^3}\mathbf{r}_{CS} \tag{7}$$

$$\frac{d^2\mathbf{r}_C}{dt^2} = -G\frac{m_E}{\|\mathbf{r}_{EC}\|^3}\mathbf{r}_{EC} - G\frac{m_M}{\|\mathbf{r}_{MC}\|^3}\mathbf{r}_{MC} - G\frac{m_S}{\|\mathbf{r}_{SC}\|^3}\mathbf{r}_{SC} \tag{8}$$

subject to the initial conditions in time, *t*:

$$\mathbf{r}_E(t = 0) = \mathbf{r}_{E0}, \quad \mathbf{v}_E(t = 0) = \mathbf{v}_{E0} \tag{9}$$

$$\mathbf{r}_M(t = 0) = \mathbf{r}_{M0}, \quad \mathbf{v}_M(t = 0) = \mathbf{v}_{M0} \tag{10}$$

$$\mathbf{r}_S(t = 0) = \mathbf{r}_{S0}, \quad \mathbf{v}_S(t = 0) = \mathbf{v}_{S0} \tag{11}$$

$$\mathbf{r}_C(t = 0) = \mathbf{r}_{C0}, \quad \mathbf{v}_C(t = 0) = \mathbf{v}_{C0} \tag{12}$$

Because the mass of the spacecraft, m_C, is much less than the masses of Earth, the sun, and the moon, their equations of motion in Eqs. (5) through (7) can be uncoupled from the spacecraft’s in Eq. (8): Earth, the sun, and the moon comprise a separate three-body system independent of the spacecraft, and determining the trajectory of the spacecraft is now the “restricted” 4-body problem.

If the gravitational influence on the spacecraft was limited to one body (N = 2), the analytical solution would simply be a conic orbit. For N > 2, the global solution has only recently been found (excluding the case of zero angular momentum) as a series expansion.[7] However, this solution converges incredibly slowly, so numerical integration of the equations of motion is often the most efficient means of obtaining results. Approximate methods, such as using two-body solutions with perturbations, are widespread, but not considered here.

Equations (5) through (8) all assume an inertial reference frame. In many cases it is convenient to express position, velocity, and acceleration relative to a moving reference frame. For a generic moving reference frame, with acceleration $\mathbf{a}_0$, angular velocity $\boldsymbol{\omega}$, and angular acceleration $\boldsymbol{\alpha}$ relative to an inertial frame, the acceleration of the spacecraft relative to the moving frame, $\mathbf{a}_c$, is given by

$$\mathbf{a}_\mathrm{c} = \mathbf{a}_\mathrm{g} - \mathbf{a_0} - \boldsymbol{\alpha} \times \mathbf{r}_p - \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \mathbf{r}_p\right) - 2\boldsymbol{\omega} \times \mathbf{v}_p \tag{13}$$

where $\mathbf{r}_p$ and $\mathbf{v}_p$ are the position and velocity, respectively, of the body relative to the moving frame and $\mathbf{a}_g$ is the equivalent gravitational acceleration that would be experienced in an inertial frame. For spherical gravitational forces, $\mathbf{a}_g$ is given by Eq. (2), or for oblateness effects, by Eqs. (3) and (4). Equation (13) would be necessary for any frame geographically fixed to the surface of Earth, such as an Earth-centered, Earth-fixed frame.

Another common moving frame is the Earth-Centered Inertial (ECI) frame, which is fixed to the same directions as the ICRF (i.e., it does not rotate). It is not truly inertial, though, because its center accelerates due to Earth’s movement around the sun (and to a lesser extent around Earth-moon barycenter). For this case, Eqn. (13) reduces to

$$\mathbf{a}_\mathrm{c} = \mathbf{a}_\mathrm{g} - \mathbf{a}_\mathrm{E} \tag{14}$$

and $\mathbf{a}_E$ represents the acceleration of Earth (as the coordinate center, with respect to the solar system barycenter) due to the influence of the sun, moon, and other objects, approximated by

$$\mathbf{a}_\mathrm{E} \approx -G\frac{m_S}{\|\mathbf{r}_{SE}\|^3}\mathbf{r}_{SE} - G\frac{m_M}{\|\mathbf{r}_{ME}\|^3}\mathbf{r}_{ME} \tag{15}$$

Assuming a circular orbit of Earth about the sun with orbital speed, v, and no influence from the moon, the magnitude of this acceleration, a_E, can be approximated roughly as

$$a_E \approx \frac{v^2}{r_{SE}} = \frac{Gm_S}{{r_{SE}}^2} \tag{16}$$

for r_{SE}, the mean distance between the sun and Earth. This acceleration is approximately 0.006 m/s^2, considerably less than the acceleration from Earth's gravity that would be exerted on the spacecraft near Earth. However, at a distance of approximately 260,000 km from Earth (well within the distance traversed during a lunar mission), $\mathbf{a}_E$ has roughly the same magnitude as $\mathbf{a}_g$. Thus for accurate lunar trajectories using Eq. (14) for an ECI frame, the acceleration of Earth cannot be neglected.

EPHEMERIS

There are two possible approaches to solving this problem with numerical integration. The motion of all four bodies can be computed: Earth, the sun, and moon together by Eqs. (5) through (7) and the spacecraft by Eq. (8) from the calculated positions of the those bodies. A more efficient approach, however, is to use either pre-calculated ephemeris tables or curve-fits for the locations of Earth, the moon, and the sun, eliminating the need for integrating Eqs. (5) through (7) at all. These ephemerides can be stored in computer memory and then used at any time to calculate the distance (and thus the acceleration) experienced by the spacecraft from the planetary bodies. If the positions of all the gravitationally influential bodies are known *a priori* from ephemerides, only one equation must be solved numerically, given the initial conditions in Eq. (12). This eliminates the computation of the positions of any planetary body, and the only added computational time is that required to look up positions of planets and perform simple arithmetic. Similarly, for the ECI frame, the acceleration of Earth as the coordinate origin can be obtained from the positions of the planetary bodies, and only one equation must be solved to determine the motion of the spacecraft. These ephemerides also give more accurate positions of planetary bodies, since they are calculated using gravitational influence from additional bodies and are corrected to fit astronomical observations.

To date, however, the only ephemeris implemented in the algorithm has been the DE 421 ephemeris. This ephemeris was selected for its improved lunar accuracy over previous NASA JPL ephemerides.[8] However, it is anticipated that such accuracy is not necessary, and other, faster methods of generating planet positions are being considered for implementation. For instance, curve-fits for lunar and solar positions (if found) could be used to quickly calculate planetary positions at a fraction of the computer memory required for the DE 421 ephemeris. Furthermore, the ephemeris has only been accessed through NASA's CSPICE/MICE MATLAB interface, but accessing it directly by coding in C should be considerably faster.

ASSUMPTIONS

Some simplifying assumptions were made which limited the complexity of the problem and the types of trajectories to be considered. First, the allowable fuel/ΔV budget for any lunar orbit insertion maneuvers was assumed to be inconsequential. Second, Earth parking orbit was assumed to be circular and in the plane of the moon's orbit around Earth. Third, the TLI burn was assumed to be instantaneous and in the plane of the spacecraft's parking orbit. Assuming that the sun's gravity does not contribute a significant out-of-plane perturbation, this means the transfer orbit must also be coplanar with the moon's orbit. This effectively predetermines the inclination of the craft's eventual path around the moon, eliminating lunar inclination as a required target parameter. [*] However, for the gravitational model which includes Earth's oblateness this is not a realistic assumption, because the transverse force serves to alter the inclination. Thus for the Earth oblateness model, either the inclination change must be approximated or the algorithm must also target inclination. Moreover, the inclination can be useful in determining whether the craft achieves a front-side or back-side flyby of the moon, as shown in Figure 2. Future algorithms could be made to include variable TLI burn times as well as non-coplanar trajectories.

For these assumptions, starting from a fixed-altitude circular Earth parking orbit, three initial state variables are sufficient to specify a unique translunar trajectory: the phase angle at departure, γ (equivalent to the time of departure from the specified parking orbit), the flight path angle, φ, and the magnitude of the velocity change, ΔV (Figure 3). Any desired target parameters,

[*] Of course, for a launch directly into Earth parking orbit from Cape Canaveral (at approximately 28° N latitude), these assumptions require that the launch occur when the inclination of the moon's orbit to Earth's equatorial plane is at least 28°. This is near the maximum lunar inclination to the equator, which occurs only once every 18.6 years according to the moon's nodal cycle. See Ref. 11, pp. 344.

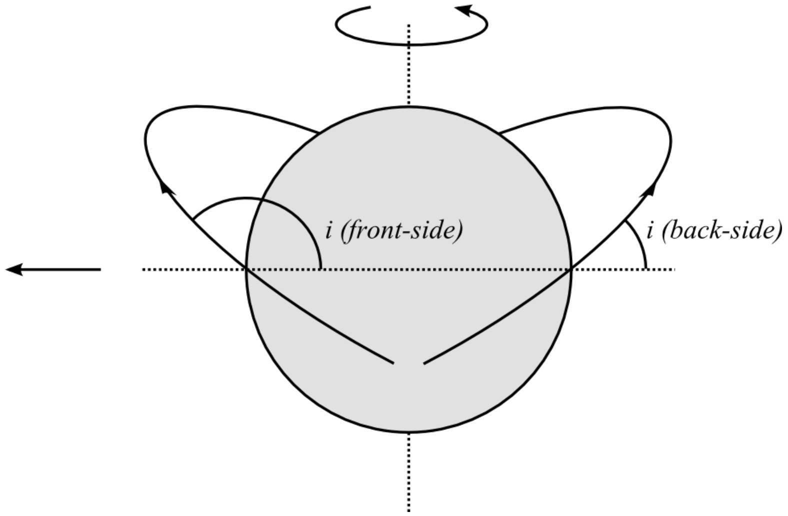

Figure 2. Inclination for Lunar Flybys.

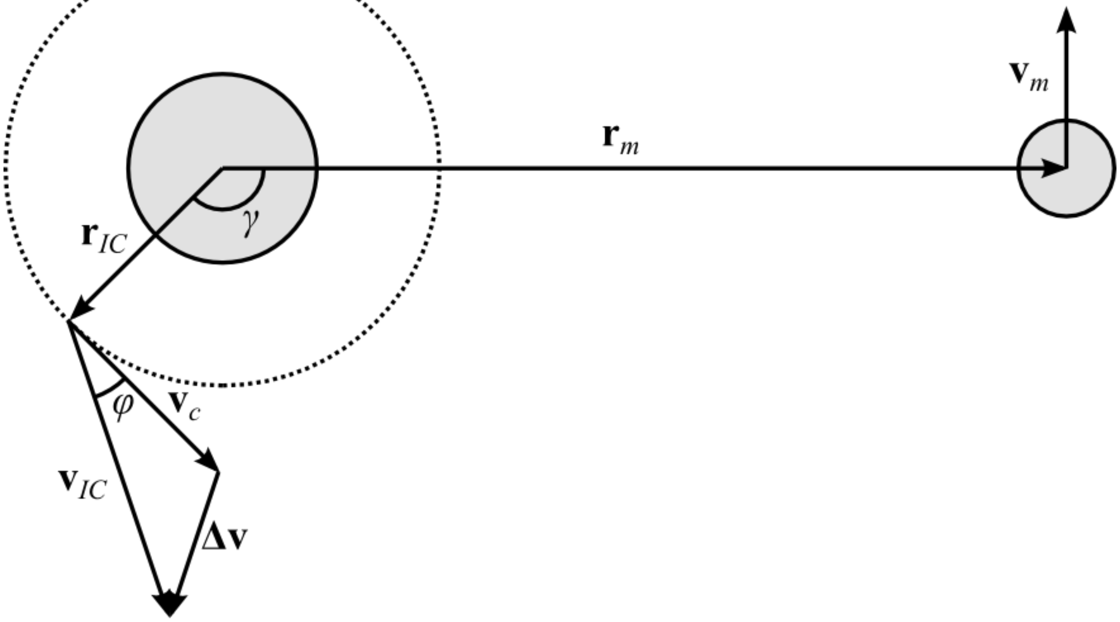

Figure 3. Coplanar Lunar Transfer Departure Geometry.

such as radius and latitude of perilune, are then only a function of these three state variables. It should also be noted that a solution for the initial state variables meeting a given set of target parameters may or may not exist or be unique.

INITIAL CONDITIONS

The following steps must be taken to determine the appropriate initial conditions (in Cartesian coordinates) for a given set of initial state variables. First, the moon's position and velocity vectors relative to Earth ($\mathbf{r}_m$ and $\mathbf{v}_m$, respectively) are determined from the ephemeris at a fixed date. The moon's specific angular momentum, $\mathbf{h}_m$, is simply the cross product of $\mathbf{r}_m$ and $\mathbf{v}_m$. The craft is then assumed to be in line between Earth and the moon, with its position vector given by

$$\mathbf{r}_c = (R_e + A)\frac{\mathbf{r}_m}{\|\mathbf{r}_m\|} \tag{17}$$

where R_e is Earth's (equatorial) radius and A is the altitude of the circular parking orbit. The magnitude of the velocity vector in circular orbit is then

$$v_c = \sqrt{\frac{Gm_E}{R_e + A}} \tag{18}$$

Since the parking orbit is circular, its velocity vector, $\mathbf{v}_c$, must be perpendicular to $\mathbf{r}_c$. Since the parking orbit is restricted to be coplanar with the moon's orbit, $\mathbf{v}_c$ must also be perpendicular to $\mathbf{h}_m$, and by the right-hand rule

$$\mathbf{v}_c \; = -\frac{\mathbf{r}_\mathrm{c} \times \mathbf{h}_m}{\|\mathbf{r}_c\|\|\mathbf{h}_m\|} v_c \tag{19}$$

The craft's initial position and velocity vectors must then be rotated in the orbital plane by the departure phase angle. This requires first transforming the vectors from the ephemeris reference frame to a frame fixed to the orbital plane, then rotating them in the orbital plane, and then transforming them back into the ephemeris frame.

The ephemeris frame is identified by X, Y, and Z axes. This frame must first be rotated about the Z-axis by the orbit's longitude of the ascending node (Ω) to align the X-axis with the nodal vector through Eq. (20). This establishes the intermediate frame $[x_1, y_1, z_1]^\mathrm{T}$. These axes must then be rotated about the x_1-axis by the inclination (i) to align the z_1-axis with the angular momentum vector through Eq. (21). This establishes the orbital frame, $[x_2, y_2, z_2]^T$. Next, the rotation of –γ about the z_2-axis is carried out in the orbital plane to establish the $[x_3, y_3, z_3]^T$ coordinate frame through Eq. (22).

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \boldsymbol{R}_1 \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \qquad \boldsymbol{R}_1 = \begin{bmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \boldsymbol{R}_2 \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \qquad \boldsymbol{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \boldsymbol{R}_3 \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \qquad \boldsymbol{R}_3 = \begin{bmatrix} \cos(-\gamma) & \sin(-\gamma) & 0 \\ -\sin(-\gamma) & \cos(-\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Finally the coordinate frames are transformed back into the frame of the ephemeris by the inverse (equivalent to the transpose) of the $\mathbf{R}_1$ and $\mathbf{R}_2$ rotation matrices. For initial coordinates in the ephemeris frame, $[x, y, z]^T$, the new coordinates $[X, Y, Z]^T$ are given by the rotation

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \boldsymbol{R} = (\boldsymbol{R}_2\boldsymbol{R}_1)^{\boldsymbol{T}} \boldsymbol{R}_3^{\boldsymbol{T}} \boldsymbol{R}_2 \boldsymbol{R}_1 \tag{23}$$

Now $\mathbf{r}_c$ is rotated to form the position vector for the initial conditions, $\mathbf{r}_{IC}$, by

$$\mathbf{r}_{IC} = \mathbf{R}\mathbf{r}_c \tag{24}$$

This rotation also applies to the velocity vector if the flight-path angle is zero (i.e., the TLI burn is conducted tangential to the path in the parking orbit). In this case, the velocity initial condition, $\mathbf{v}_{IC}$, is formed simply by adding the final state variable, ΔV magnitude, to the rotated circular velocity, $\mathbf{R}\mathbf{v}_c$, as

$$\mathbf{v}_{IC} = \mathbf{R}\mathbf{v}_c + \Delta V \frac{\mathbf{R}\mathbf{v}_\mathrm{c}}{v_c}, \qquad \varphi = 0 \tag{25}$$

If the flight-path angle is not zero, the angle of rotation for $\mathbf{R}_3$ in Eq. (22) is simply replaced by –(γ + φ), and the unique velocity rotation matrix is now labeled $\mathbf{R}_v$. Then by trigonometry, $\mathbf{v}_{IC}$ can be shown to be

$$\mathbf{v}_{IC} = \mathbf{R}_v \mathbf{v}_{\mathrm{c}} \left[\cos\varphi + \frac{1}{v_c} \sqrt{\Delta V^2 - {v_c}^2 \sin^2\varphi} \right], \qquad \varphi \neq 0 \tag{26}$$

PREDICTOR

Numerical Integration

The governing differential equations (presented below only for an inertial system) are second order and non-linear in **r**. Thus, the position of the spacecraft must be determined by rearranging its equation into two first-order equations:[9]

$$\frac{d\mathbf{v}_i}{dt} = -G \sum_{j=1, j\neq i}^{N} \frac{m_j}{\left\|\mathbf{r}_{ji}\right\|^3} \mathbf{r}_{ji} \tag{27}$$

and

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \tag{28}$$

The velocity may then be approximated for each time step with a numerical integration scheme such as the Euler or Runge-Kutta method, and in turn, the position can be approximated with a second numerical integration.

For the sake of example, the (explicit) Euler method with uniform Δt time-step can be implemented for this problem for discretized velocity, $\mathbf{V}_{i,n}$, and position, $\mathbf{R}_{i,n}$, for the spacecraft as

$$\mathbf{V}_{i,n+1} = \mathbf{V}_{i,n} + \Delta t \cdot f(t_n, \mathbf{V}_{i,n}) \tag{29}$$

where

$$f\left(t_n, \mathbf{V}_{i,n}\right) = -G \sum_{j=1, j\neq i}^{N} \frac{m_j}{\left\|\mathbf{R}_{ji,n}\right\|^3} \mathbf{R}_{ji,n} \tag{30}$$

and

$$\mathbf{R}_{i,n+1} = \mathbf{R}_{i,n} + \Delta t \cdot g(t_n, \mathbf{R}_{i,n}) \tag{31}$$

where

$$g\left(t_n, \mathbf{R}_{i,n}\right) = \mathbf{V}_{i,n} \tag{32}$$

Runge-Kutta and other methods of various order can be employed in a similar fashion.

The selection of a numerical integration scheme for the predictor is crucial to the efficiency of the targeting algorithm, because the corrector must call upon the integration scheme several times. Thus a variety of numerical integration schemes were compared to determine which ones offer the greatest accuracy for the lowest computational time. In order to evaluate the errors associated with different numerical integration schemes, the Earth-moon system was modeled by the simplified problem of two-body elliptical motion (using Earth and the moon). The results for the computational model were then compared to the analytical solution. It should be noted that this model is by no means an accurate representation of the Earth-moon system. It assumes an inertial frame of reference located at the Earth-moon barycenter. This is certainly not the case, since the Earth-moon system is revolving around the sun. However, the simplified model (albeit using incorrect assumptions) does have an analytical solution, and its time and length scales are on roughly the same order as those of translunar missions. Thus the Earth-moon two-body assumption provides an analytical solution as a baseline for comparison of numerical integration schemes. For this problem, the equation of motion for the position of the moon relative to Earth, $\mathbf{r}$, is given by[*]

$$\frac{d^2\boldsymbol{r}}{dt^2} = -\frac{G(m_E + m_M)}{\|\mathbf{r}\|^3}\boldsymbol{r} \tag{33}$$

Unfortunately, the analytical solution provides the shape of each orbit and the time of flight as a function of orbital parameters. The inverse problem (“Kepler’s problem”) of finding a body’s

[*] The initial conditions for Earth and the moon were those given by the NASA HORIZONS ephemeris generator (using the DE 405 ephemerides) at 00:00:00 CT, 1 January 2010 for the ICRF tied to Earth mean equator (+z direction) and mean dynamical equinox (+x direction) of the J2000 epoch.

position as a function of time (which is precisely how the numerical approximations will operate) itself requires the numerical solution of a transcendental equation*. Thus, it would be beneficial to compare the model to the analytical solution by some means other than by the calculated position at each time. Two alternate methods were used for this purpose, as follows.

First, the total angular momentum of the system must be conserved. On a unit-mass basis, the specific angular momentum, **h**, is given by

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} = \mathbf{r}_E \times \mathbf{v}_E + \mathbf{r}_M \times \mathbf{v}_M \tag{34}$$

and should be the same at any time. The specific angular momentum for the numerical scheme can then be calculated at any time and compared to its value from the initial conditions. The error at any time, $\mathbf{e}_h(t_n)$, is simply the difference between the two, that is

$$\begin{gathered}\mathbf{e}_h(t_n) = \mathbf{H}_n - \mathbf{h}(t = 0) \\ = (\mathbf{R}_{E,n} \times \mathbf{V}_{E,n} + \mathbf{R}_{M,n} \times \mathbf{V}_{M,n}) - (\mathbf{r}_{E0} \times \mathbf{v}_{E0} + \mathbf{r}_{M0} \times \mathbf{v}_{M0})\end{gathered} \tag{35}$$

and the magnitude of the error, $e_h(t_n)$, is simply

$$e_h(t_n) = \|\mathbf{e}_h(t_n)\| \tag{36}$$

Second, the period of the orbit, T, can be determined from the initial conditions by

$$T = 2\pi G(m_E + m_M)\left[-\|\mathbf{v}_0\|^2 + \frac{2G(m_E + m_M)}{\|\mathbf{r}_0\|}\right]^{-\frac{3}{2}} \tag{37}$$

The numerical scheme can thus be set to calculate the position and velocity at the period T. Since the position and velocity for the analytical solution at time t = T must be the same as the initial position and velocity, the errors, $\mathbf{e}_r(t_n = T)$ and $\mathbf{e}_v(t_n = T)$, are simply the differences between the two values:

$$\mathbf{e}_r(t_n = T) = \mathbf{R}_n - \mathbf{r}_0 \tag{38}$$

* See Ref. 11, pp. 177-222.

and

$$\mathbf{e}_v(t_n = T) = \mathbf{V}_n - \mathbf{v}_0 \tag{39}$$

and their magnitudes are given by their norms, as with the error in specific angular momentum. It should be noted that the calculation of T itself involves computer round-off error, but this is incredibly small and causes the same error in input into any numerical integration scheme.

Two types of ordinary differential equation solvers were considered for this problem: fixed time-step and variable time-step solvers. The fixed time-step schemes ranged from first to fifth order (ode1 through ode5). The variable time-step integrators were built-in MATLAB functions (ode23, ode45, etc.) which use at least two different orders of numerical integration schemes and are each best suited for different types of problems.[*] For the fixed time-step solvers, the number of time steps was successively increased to improve accuracy, and for the variable time-step solvers, the maximum relative and absolute tolerances were decreased. For both types of solvers, the required computation times were recorded for each level of accuracy.[†]

Figure 4 shows the relationship between position accuracy and computation time for all of the fixed time-step solvers for one Earth-moon period of revolution (approximately 27.48 days). Note that the errors decrease for every method as the time-step is refined (i.e., the required computation time is increased). As expected, the slope of each curve is roughly the same as the order of the

[*] These codes were obtained from MathWorks, Inc. For more information, see documentation at http://www.mathworks.com/support/tech-notes/1500/1510.html
and http://www.mathworks.com/help/techdoc/ref/ode113.html?BB=1

[†] Two steps were taken to decrease the influence of fluctuations in computer processing. First, no other programs were run while the integration schemes were executed. Second, the computation times for each solver at each level of accuracy were averaged over a total of 25 runs. All integrations were performed on a Dell Precision T3400.

integration scheme. Also note that the higher order solvers level off at an accuracy of approximately 0.015 m. This is the result of round-off error, and is effectively the limit of accuracy for the computer system used for this problem. If the lower-order solvers were allowed to run longer, they would be expected to level out at the same position error of about 0.015 m. Nearly identical results were observed for the specific angular momentum and velocity (Figures 5 and 6, respectively).

The variable time-step solvers provided somewhat different results (Figure 7). These solvers typically have a slope that is much higher than the order of their numerical integration schemes when used with loose tolerances. Thus, unless a high level of accuracy is needed, these solvers provide a much greater improvement in accuracy than their fixed time-step counterparts for the same increase in computation time. After a certain level of accuracy, however, the slopes level off to the order of one of the solver's numerical integration schemes.

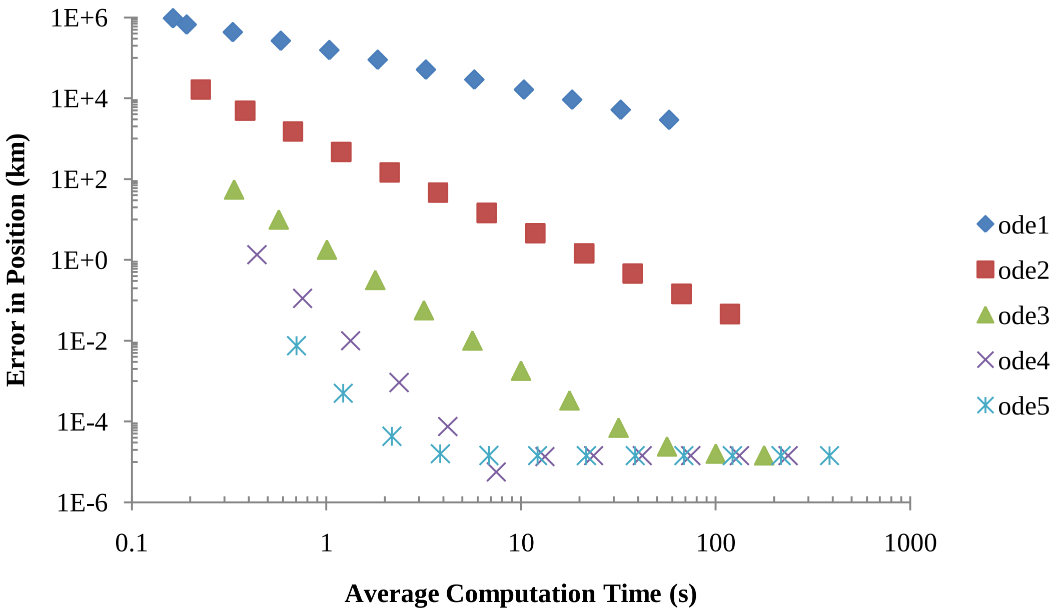

Figure 4. Error in Position after One Period as a Function of Computation Time, Fixed Time-Step Solvers.

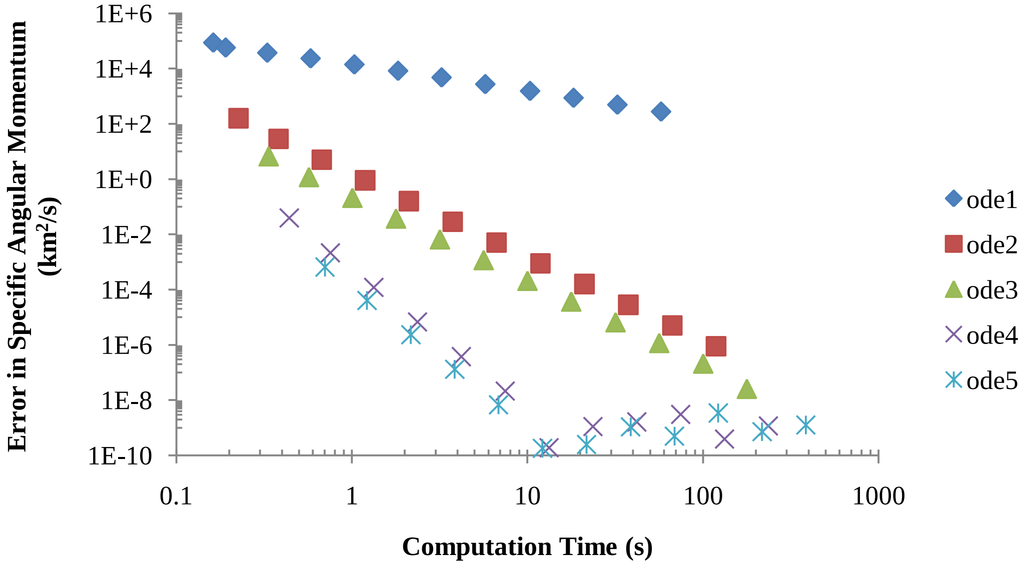

Figure 5. Error in Specific Angular Momentum after One Period as a Function of Computation Time, Fixed Time-Step Solvers.

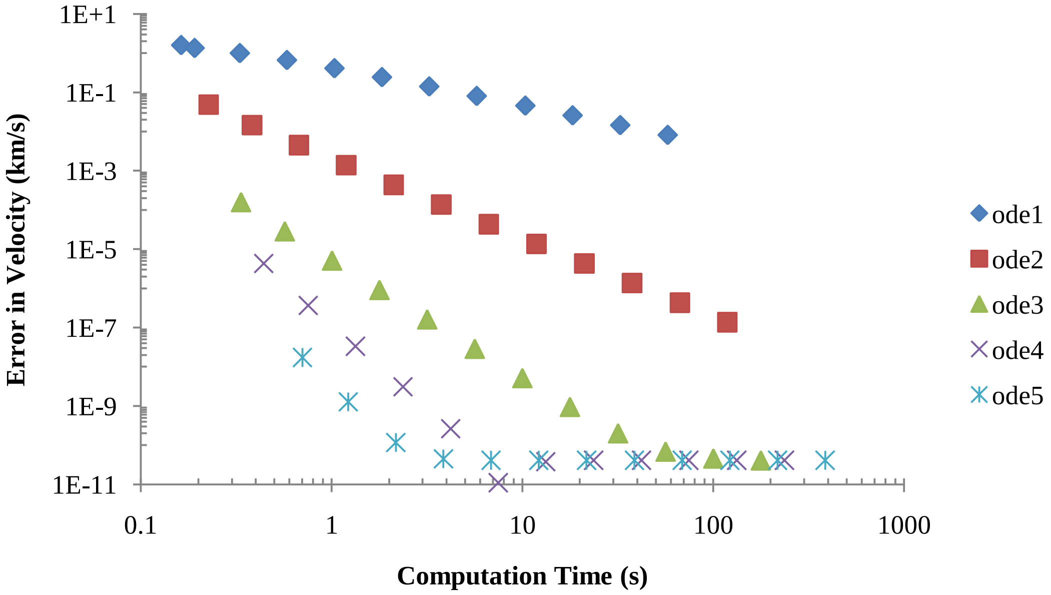

Figure 6. Error in Velocity after One Period as a Function of Computation Time, Fixed Time-Step Solvers.

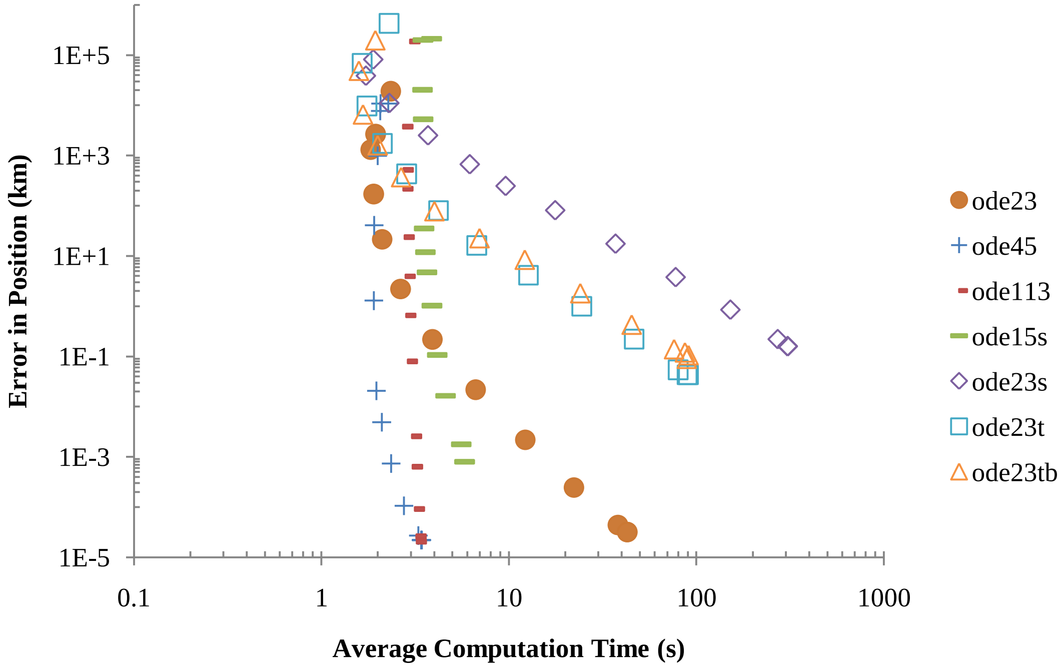

Figure 7. Error in Position after One Period as a Function of Computation Time, Variable Time-Step Solvers.

Because of inconsistencies in the way the two types of solvers operate and are coded, a direct comparison between their computation times is somewhat problematic. For instance, the variable time-step solvers require a start-up time to determine an appropriate step size to maintain the error tolerances. Also, extra features, like error catching and event determination are built into the variable time-step solvers but not the fixed time-step solvers. This ultimately increases the computation time for the variable time-step solvers. In the future, the codes for these solvers could be modified to eliminate this discrepancy, or additional solvers could be written to remove any extraneous operations. Nevertheless, both of these solver types were considered for this analysis, and so there was no option but to make a comparison between them. Simply note that if the extraneous time required by the variable time-step solvers is significant, the solvers themselves would be faster than the results suggest.

Figure 8 shows the superimposed position errors for the fixed time-step solvers and best performing variable time-step solvers. It is clear that the fifth-order fixed time-step method (ode5) outperformed all others, providing the highest accuracy in the shortest amount of time. However, it does not have the built-in error estimation that the variable time-step solvers have. Thus for translunar trajectories, with no analytical solutions, there is no way to estimate how much error is accumulated by using ode5. The ode45 solver was ultimately chosen, because it performed nearly as well as ode5 at high levels of accuracy, and because its intrinsic estimate of the error is a much more intuitive parameter to select than the number of time-steps required for ode5.

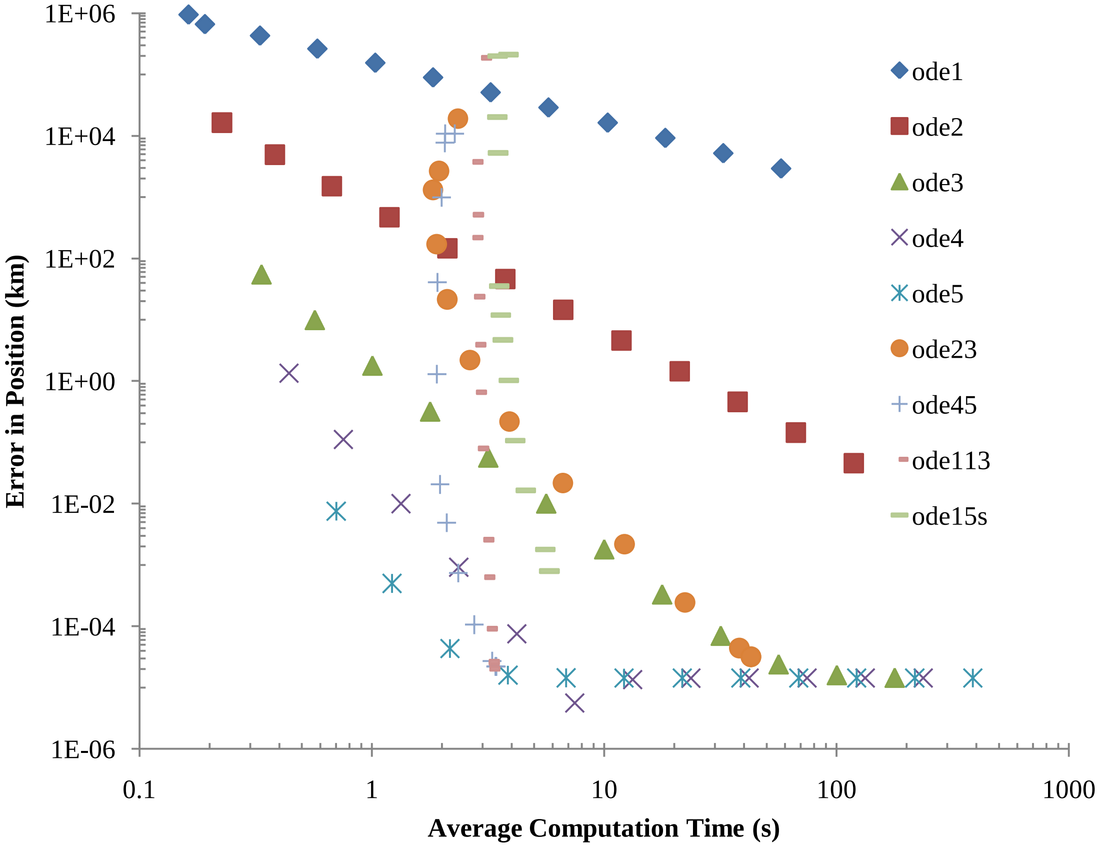

Figure 8. Error in Position after One Period as a Function of Computation Time, Select Solvers.

Perilune

The time required for numerical integration was originally only limited by setting a maximum time-of-flight (thereby setting the end time). After the trajectory was integrated fully, the time of perilune passage was then found by determining the time at which the craft passed closest to the moon. A new approach was then implemented which allowed the integration to be stopped immediately at the time of perilune passage.[*] At perilune passage, by definition, the distance between the spacecraft and the moon, *d*, must be a minimum (and go from decreasing to increasing).[†] The square of this distance (in an *x*, *y*, *z* coordinate system) is simply

$$d^2 = (r_{Cx} - r_{Mx})^2 + \left(r_{Cy} - r_{My}\right)^2 + (r_{Cz} - r_{Mz})^2 \tag{40}$$

and the time rate of change of this quantity is

$$\begin{aligned}\frac{\mathrm{d}(d^2)}{\mathrm{dt}} = 2\big[&(r_{Cx} - r_{Mx})(v_{Cx} - v_{Mx}) + \left(r_{Cy} - r_{My}\right)\left(v_{Cy} - v_{My}\right) \\ &+ (r_{Cz} - r_{Mz})(v_{Cz} - v_{Mz})\big]\end{aligned} \tag{41}$$

The time rate of change in d^2 must be zero at perilune, and go from negative to positive. The ode45 integrator was therefore set to determine this event based on the positions and velocities at the current time-step, narrow in on the precise time and spacecraft location at perilune, and then cease integration. This prevented the excess computation time associated with integrating the trajectory fully and then determining when perilune occurred.

[*] Adapted from MATLAB (7.6.0) Help File: Advanced Event Location – Examples - Differential Equations.

[†] Note that if any calculated radius of perilune is smaller than the moon's radius (approximately 1737 km), the trajectory actually results in a lunar impact.

Sphere of Influence Method

Another method was also considered to try to limit the extent of the required numerical integration. This method assumed regular 2-body Keplerian dynamics at arrival at the moon's sphere of influence. To test the accuracy of this assumption, a few trial trajectories were calculated with ode45. The radius of perilune, r_p, was then calculated analytically at various points along the assumed trajectory relative to the moon (i.e., at various radii for the sphere of influence) as

$$r_p = \frac{h^2}{Gm_M(1 + e)} \tag{42}$$

Here h is the magnitude of the specific angular momentum from Eq. (43) and e is the magnitude of the eccentricity vector, **e**, from Eq. (44) for the current position and velocity vectors, $\mathbf{r}_c$ and $\mathbf{v}_c$, respectively.

$$h = \|\mathbf{r}_c \times \mathbf{v}_c\| \tag{43}$$

$$\mathbf{e} = \frac{\mathbf{v}_c \times (\mathbf{r}_c \times \mathbf{v}_c)}{GM_m} - \frac{\mathbf{r}_c}{\|\mathbf{r}_c\|} \tag{44}$$

These values were compared to the actual radius of perilune determined from the fully integrated trajectory. The results for one particular trajectory, with a calculated radius of perilune of approximately 12,000 km, are shown in Figure 9. The magnitude of the error increases with the square of the distance. This is expected, since the gravitational force of the moon increases with the square of the distance, overshadowing the gravitational influence of Earth and the sun. Note that the standard radius for the lunar sphere of influence (about 66,000 km) gives a nearly 12% underestimate of the true radius of perilune. Modest decreases in the assumed sphere of influence radius lead to significantly improved estimates of the perilune radius. However, it was decided to continue the numerical integration all the way to perilune to assure accuracy. Future

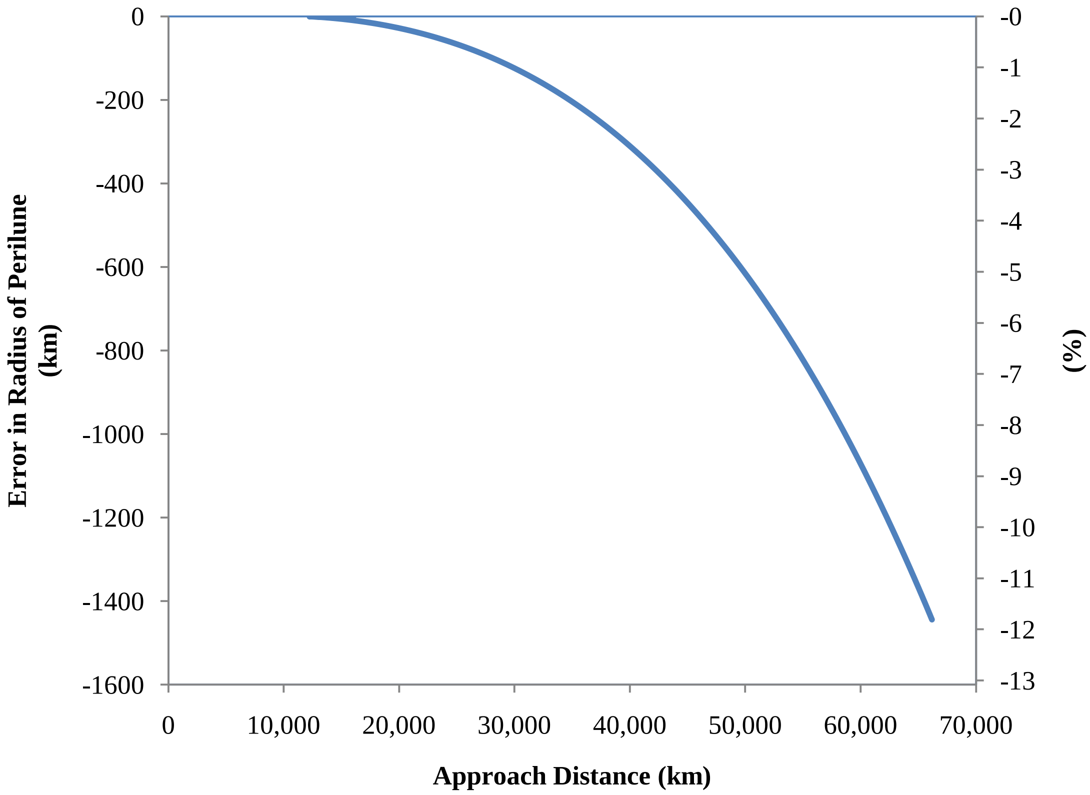

Figure 9. Error in Radius of Perilune for Kepler Assumption as a Function of Approach Distance (Sphere of Influence Radius).

efforts may identify empirical relationships between the perilune radius error and the approach distance for a wide range of trajectories. This would enable the integration to be terminated earlier at the sphere of influence and most of the accuracy to be recovered simply by subtracting the error.

Orbital Parameter Calculations

Given the position and velocity vectors at perilune from numerical integration, the orbital elements at that instant (osculating elements) can be determined from 2-body Keplerian dynamics. It is most useful to express these elements in a frame of reference centered at the moon, since they are targeted primarily based on lunar surface features. To this end, the vectors at perilune were rotated from Earth-Centered Inertial frame for the numerical integration to the Mean Earth/Polar Axis (ME) lunar reference frame (with appropriate rotation matrices supplied by the MICE MATLAB interface).* In this frame, with perilune position, velocity, and specific angular momentum vectors $\mathbf{r}_p$, $\mathbf{v}_p$, and $\mathbf{h}_p$, respectively, the latitude, λ_p, and inclination, i_p, are given by

$$\lambda_p = \arcsin\left(\frac{r_{pz}}{\|\mathbf{r}_\mathrm{p}\|}\right) \tag{45}$$

$$i_p = \arccos\left(\frac{h_{pz}}{\|\mathbf{h}_\mathrm{p}\|}\right) \tag{46}$$

where ‘z’ indicates the vector component in the direction of the moon’s polar axis.

* See “A Standardized Lunar Coordinate System for the Lunar Reconnaissance Orbiter and Lunar Datasets, LRO Project and LGCWG White Paper,” Version 5, NASA Goddard SFC, Greenbelt, MD, Oct. 2008.

INITIAL GUESSES

Any corrector requires an initial guess of the state variables to obtain the desired targeted parameters. If the guesses are too far away from a viable solution, the corrector will typically require too many iterations to converge or not converge at all. Thus, the accuracy of a first guess is crucial to the effectiveness of the corrector. Several ways of providing a good first guess were considered. First, a genetic algorithm can be used to randomly sample and evaluate potential trajectories to select one "good enough" for the corrector. Since the objective function requires lengthy numerical integrations, though, this method was not deemed feasible. Another method, used by Marchand et al., is to optimize many trajectories for a wide range of targeted parameters and tabulate the initial conditions, which can then be used as initial guesses for the corrector.[5] Miele et al. have already investigated optimal free-return trajectories for a model similar to those used in the predictor (only ignoring the sun's gravity and assuming a circular lunar orbit).[10] For a perilune altitude of 100 km, their optimal ΔV for (tangential) TLI was 3.093 km/s at a departure phase angle of 132.5°. For a more accurate model, however, these numbers would necessarily fluctuate with the relative positions of Earth, the moon, and the sun.

Finally, various analytical approximations for the end state parameters can be used to determine a first guess. A patched conic approximation can be used to determine a reference trajectory assuming at any point the craft is only under the gravitational influence of one body, Earth or the moon. The minimum required ΔV occurs for a Hohmann elliptical transfer orbit, tangent to both the circular parking orbit and the moon's orbit around Earth, and can be shown to be approximately

$$\Delta V_{min} = -\frac{Gm_E}{R_E + A} + \sqrt{2Gm_E\left[\frac{1}{(R_E + A)} - \frac{1}{(R_E + A + r_{EM})}\right]} \tag{47}$$

For a 300 km altitude Earth parking orbit, this provides a reasonable first guess of about 3.1 km/s. Bate, Mueller, and White[11] and Battin[12] give iterative methods for determining analytical approximations for other departure conditions. Such analytical methods would likely be preferable to tables, since they would not require preloading any extra data for the targeting algorithm onto the onboard computer. However, these analytical methods have not been implemented yet, and for preliminary analysis it has been beneficial to tabulate trajectories to examine the dependence of the targeted parameters on the initial state variables.

In order to visualize the dependence of the perilune radius, latitude, and inclination on initial state variables, the flight-path angle φ was set equal to zero. The ΔV magnitude was then varied between 3 and 3.5 km/s and the departure phase angle γ between 70º and 200º, and the perilune states were tabulated for various parking orbit altitudes. The tolerance for the ode45 integration scheme was set at 1E-8 (km and km/s).

Figure 10 shows the variation of perilune radius with ΔV and γ for the gravitational model which does not include Earth oblateness. The time of departure (for calculating the moon's position and velocity for establishing a coplanar trajectory) for these trajectories was set at approximately 11pm, Jan.19, 2000 (UTC), corresponding to lunar perigee (approximately 357,000 km). Note that the time of flight increases going from greater ΔV and γ (top right) to lower ΔV and γ (bottom left). Also note the curved blue band between departure phase angles of about 100º and 144º indicating the region in which the closest approach to the moon can be made (less than 50,000 km). A close-up of this region is shown in Figure 11.

The inclination contour (Figure 12) shows the expected constant inclination for either a front-side or back-side lunar flyby: the left hand side (purple) for a front-side flyby at 173.2º and the right-hand side (blue) for a back-side flyby at 6.8º. Very small fluctuations (on the order of 0.02º) were observed, and are the result of the influence of the sun normal to the orbital plane. The

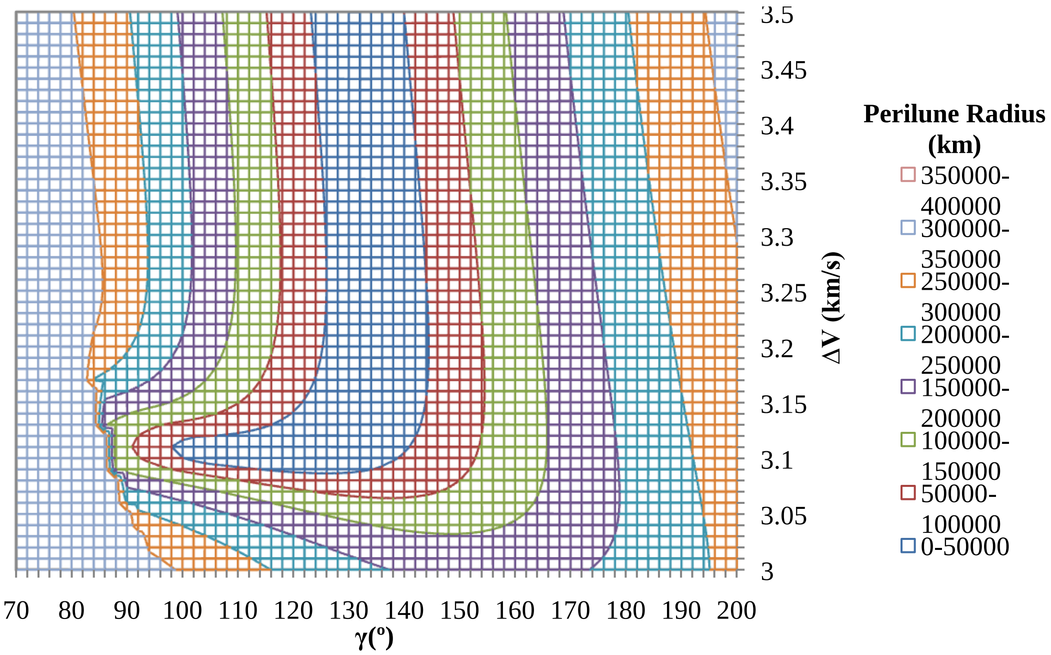

Figure 10. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee.

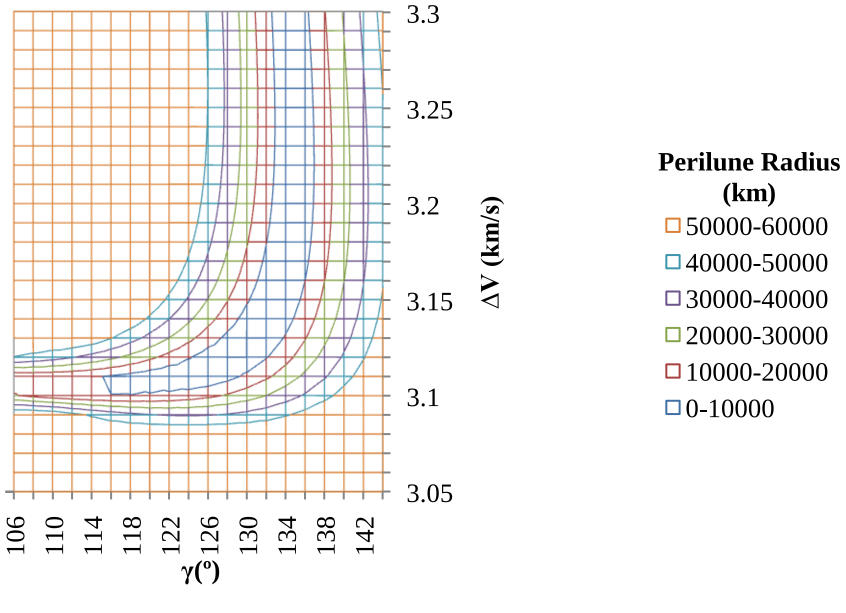

Figure 11. Radius of Perilune for Varying ΔV and γ between 106º and 144º, Point-mass Gravitational Model, Departure at Lunar Perigee.

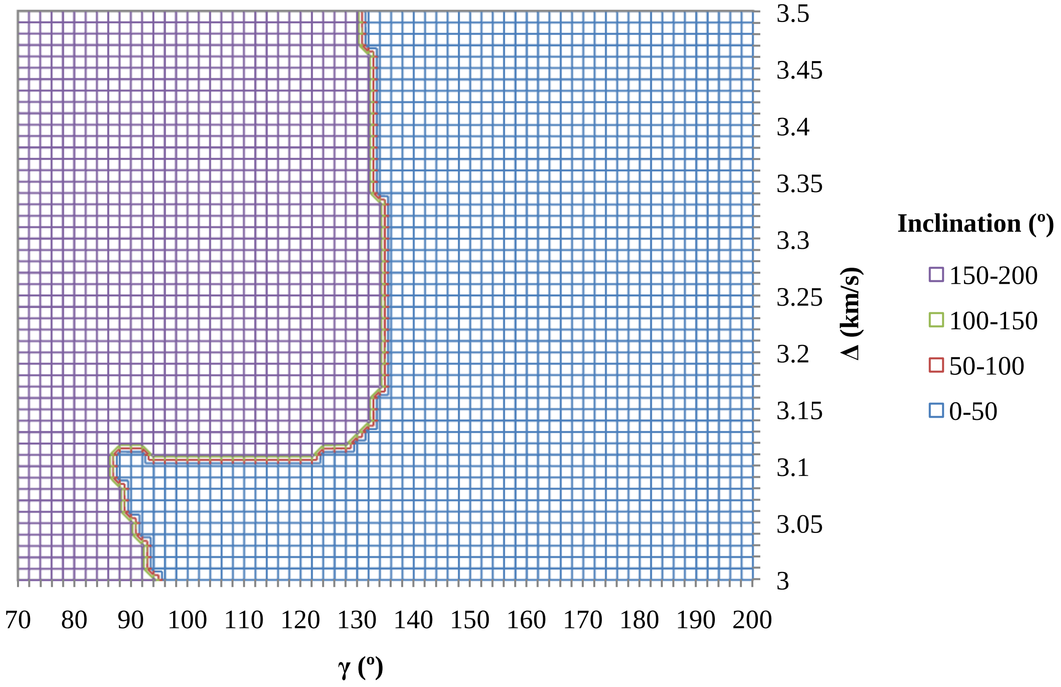

Figure 12. Inclination for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee.

dividing line in the graph indicates the region in which lunar impact would occur and corresponds to the center of the region of lowest perilune radius in Figure 10.

The latitude of perilune (Figure 13) exhibits the same dividing line between front-side and back-side flybys. The maximum perilune latitude is the same as the inclination (for a back-side flyby) either north or south. For the front-side flybys (left of the line), the latitude approaches 6.8° with distance away from the impact line. For back-side flybys, it approaches -6.8° (6.8° S). The contour is incredibly steep around this line, indicating a very narrow band in which the perilune latitude differs by more than about 1° from either 6.8° N or 6.8° S.

Equivalent plots were generated for departure at the time of the moon’s next apogee (approximately 407,000 km) at 1 am, Feb.1, 2000 (UTC). All plots exhibited the same trends, but the impact line occurred at a slightly higher ΔV (approximately 0.01 km/s higher), as expected,

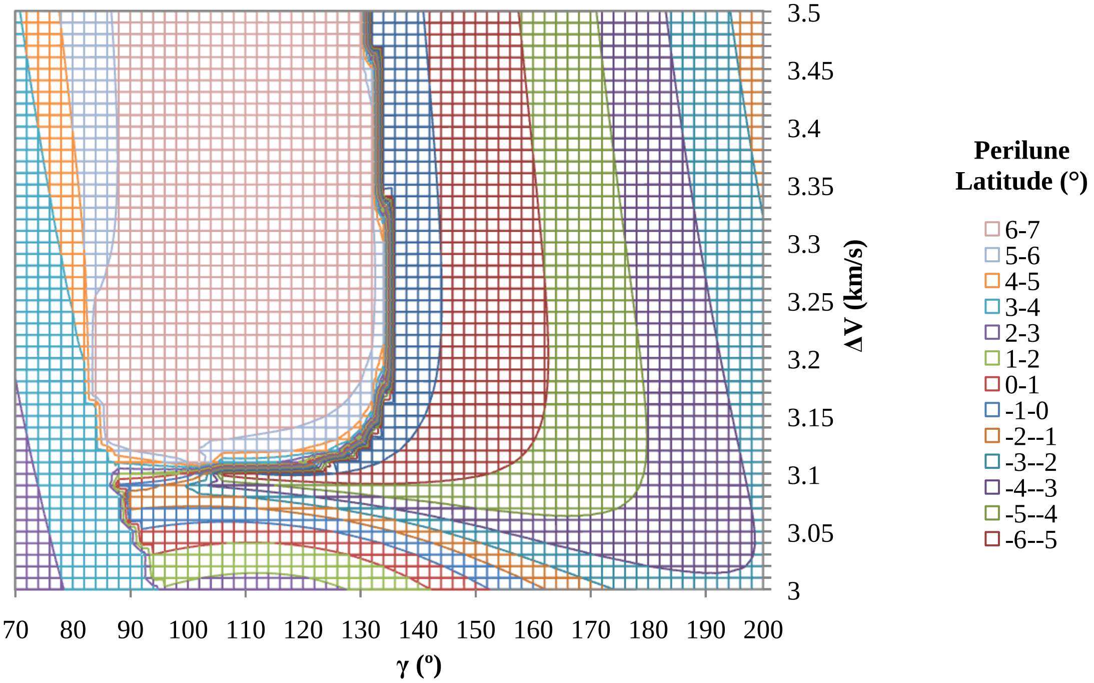

Figure 13. Perilune Latitude for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Perigee.

since the lunar distance was greater. The perilune radius for this case is shown in Figure 14. Note that in addition to the slightly higher ΔV, the range of γ for the 50,000 km range is slightly smaller and shifted toward higher phase angles (about 110° to 146°).

Similar plots were made for initial parking orbit altitudes of 150, 200, 400, and 500 km (Appendix). Increasing altitude had no noticeable effect on the range of phase angles, but decreased the minimum required ΔV from about 3.13 km/s at 150 km altitude to about 3.04 km/s at 500 km altitude. To date, the dependence of the perilune parameters on changing flight-path angle has not been studied, but the flight-path angle could easily be parameterized in much the same way as the parking orbit altitude.

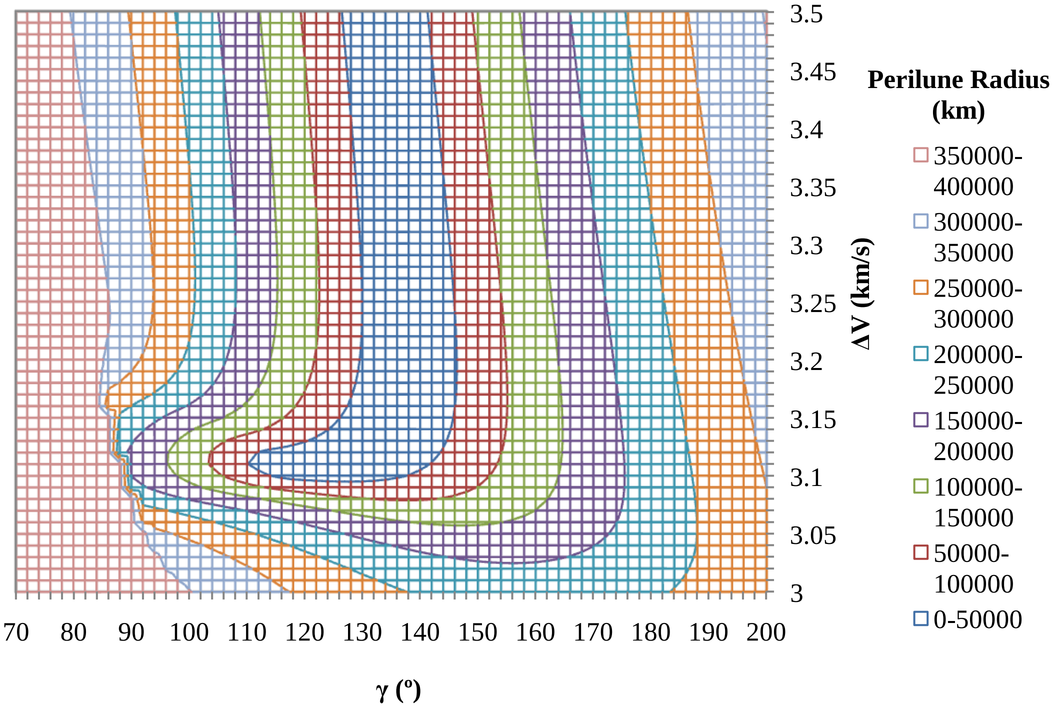

Figure 14. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, Departure at Lunar Apogee.

CORRECTORS

From these plots of perilune radius and latitude, a target trajectory was selected in order to test a variety of corrector methods. For this simple test case with zero flight-path angle (with a fixed parking orbit altitude of 300 km and departure time of noon Jan11, 2000), the perilune radius and latitude are a function of ΔV and γ only, so the system is 2x2 square. A nominal trajectory resulting in a perilune radius of 1.5 lunar radii (2605.5 km) and latitude of 3° was found to occur at a ΔV of (approximately) 3.1034 km/s and a departure phase angle of 124.4°. This trajectory up to the time of perilune is shown in Figure 15, as observed from directly above the ecliptic plane (very nearly the orbital plane) in the Earth-Centered Inertial Frame. The hyperbolic lunar phase of the trajectory (non-rotating ME frame) is shown in Figure 16 for two hours before and after perilune passage. To improve convergence, all the variables were non-dimensionalized so they

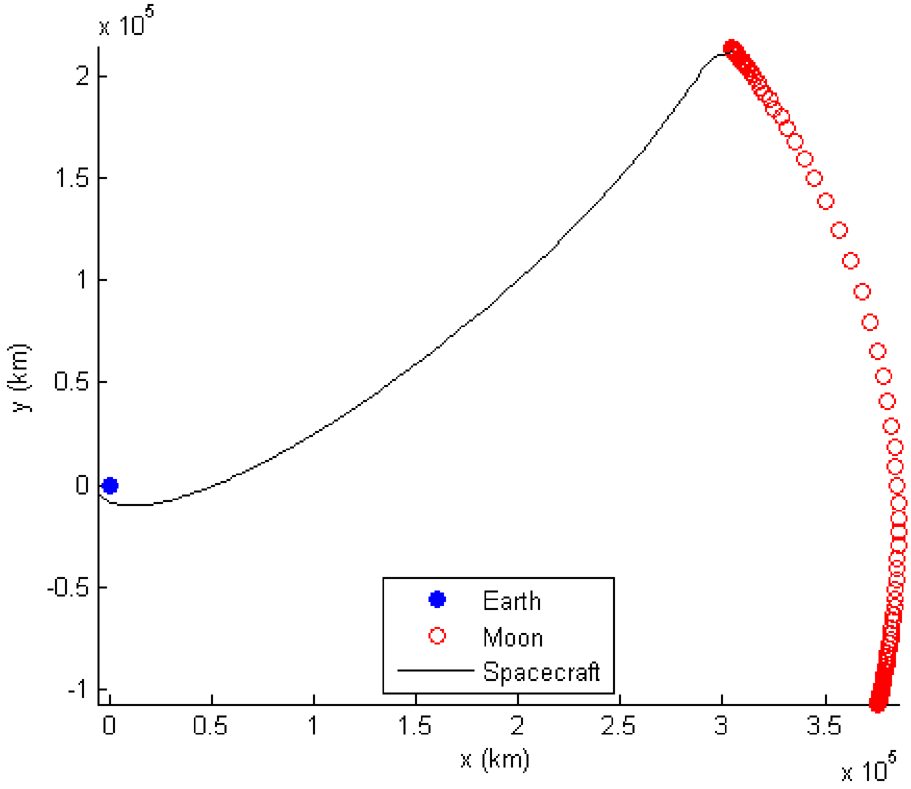

Figure 15. Nominal Translunar Trajectory for Targeted r_p = 2605.5 km, λ_p = 3° (ECI Frame).

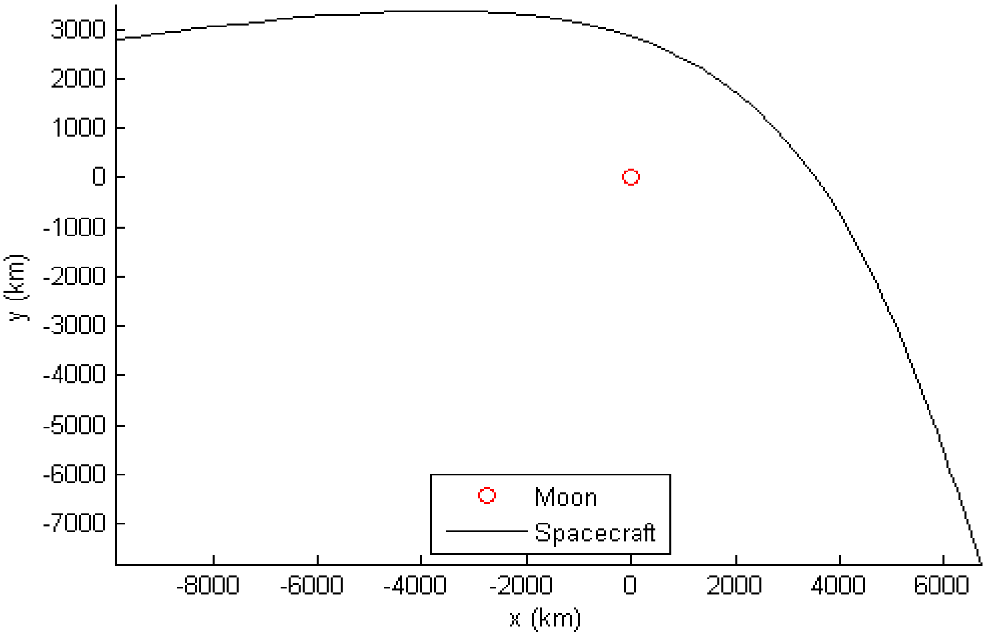

Figure 16: Nominal Translunar Trajectory for Targeted r_p = 2605.5 km, λ_p = 3° (ME Frame).

would have approximately the same scale. The perilune radius was divided by the moon's radius, ΔV and perilune latitude were scaled by 1 km/s and 1°, respectively, and γ was converted to radians.

Four separate corrector methods were tested. First, a simple quasi-Newton Broyden algorithm (using the Sherman-Morrison update to the Jacobian) was coded. With no analytical relationship between the objective function (the error in the targeted parameters) and the initial state variables, a first-order forward finite difference approximation for the initial Jacobian matrix was used. In future efforts, central differences or variational techniques, such as those given by Ocampo and Munoz, could be considered for approximating the Jacobian.[13] The three other correctors analyzed were MATLAB built-in optimization schemes: trust-region dogleg, Levenberg-Marquardt, and Gauss-Newton. For each algorithm, the acceptable tolerances in perilune radius and latitude were set at 1m and 0.001°, respectively.

RESULTS

The initial guess was varied systematically in both ΔV and phase angle to analyze the performance of each targeting algorithm. Also, the step size for the initial finite difference approximation of the Jacobian for the Broyden method was varied. The number of required iterations for the Broyden method (up to 50) is shown in Table 1. Results are shown for a maximum variation of ±0.1% in initial ΔV and departure phase angle guesses (between 3.1003 and 3.1065 km/s and between 124.28° and 124.53°) from the nominal trajectory. Outside of this range, the Broyden method's convergence was much less predictable, although it tended to be more forgiving of higher variations in phase angle than in ΔV. The targeter exhibited the best convergence for a finite-difference step size of 0.01 (km/s of ΔV and radians of departure phase angle). Despite its poor convergence, the Broyden method has the main advantage that it requires only one extra objective function evaluation (i.e., numerical integration of the equations of motion) per iteration.

The other targeters tended to converge much more reliably. The number of iterations and function evaluations required for convergence with the Gauss-Newton algorithm are shown in Table 2, Levenberg-Marquardt in Table 3, and trust-region in Table 4. The Gauss-Newton method required nearly seven function evaluations per iteration and did not converge for ΔV deviations of -0.1% and -0.08%. The Levenberg-Marquardt algorithm converged for every deviation over this range with 25 or fewer iterations, but still required on average about 5.5 function evaluations per iteration. Finally, the trust-region method converged over the entire region with fewer than three function evaluations per iteration. The range for the trust-region method was then expanded ±1% deviation in ΔV and γ, and the method converged for all but 8 out of 625 cases (1.3%). However, the farther away from the nominal case the initial conditions were, the more likely the targeter was to converge toward a second solution which yielded the

Table 1. Number of Broyden Targeter Iterations for Deviations in Initial Guess.

		Departure Phase Angle Deviation										
	(%)	-0.1	-0.08	-0.06	-0.04	-0.02	0	0.02	0.04	0.06	0.08	0.1
		Step Size = 0.01										
ΔV Deviation	-0.1	12	10	10	10	10	10	10	10	10	9	9
	-0.08	12	11	11	11	11	11	11	10	10	10	10
	-0.06	12	12	11	11	11	11	11	10	10	10	6
	-0.04	10	11	11	11	10	9	9	9	7	6	6
	-0.02	11	11	9	9	8	8	8	6	6	9	9
	0	9	7	7	7	6	0	3	7	7	9	7
	0.02	6	6	9	9	9	9	13	10	11	18	16
	0.04	20	36	35	NC	38	NC	20	26	NC	NC	30
	0.06	16	19	27	15	22	22	22	27	24	45	35
	0.08	NC	NC	NC	NC	NC	44	25	33	30	NC	34
	0.1	NC	NC	17	NC	NC	NC	NC	NC	NC	NC	NC
		Step Size = 0.001										
ΔV Deviation	-0.1	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	-0.08	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	-0.06	12	11	11	11	11	9	12	12	13	14	11
	-0.04	11	9	9	9	6	7	9	9	9	11	11
	-0.02	11	11	10	9	9	7	6	8	9	9	9
	0	7	7	7	7	6	0	6	6	7	9	10
	0.02	9	9	9	10	12	14	15	23	17	16	16
	0.04	10	10	10	18	18	18	21	43	25	42	47
	0.06	NC	NC	NC	43	NC	NC	NC	NC	NC	41	25
	0.08	NC	NC	33	NC	38	NC	NC	NC	32	45	37
	0.1	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
		Step Size = 0.0001										
ΔV Deviation	-0.1	NC	NC	NC	NC	NC	NC	NC	NC	39	NC	NC
	-0.08	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	-0.06	48	49	NC	33	NC	NC	NC	NC	NC	NC	NC
	-0.04	14	14	12	12	14	NC	28	24	22	30	NC
	-0.02	NC	NC	27	27	24	NC	NC	27	28	NC	NC
	0	12	12	10	8	7	0	7	11	9	10	11
	0.02	17	16	16	16	16	14	14	13	13	13	12
	0.04	49	39	NC	NC	NC	NC	27	NC	NC	30	40
	0.06	35	NC	38	NC	NC	NC	NC	NC	NC	NC	44
	0.08	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	0.1	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC

Note: "NC" indicates that the algorithm did not converge within 50 iterations.

Table 2. Number of Gauss-Newton Targeter Iterations and Function Evaluations for Deviations in Initial Guess.

		Departure Phase Angle Deviation										
	(%)	-0.1	-0.08	-0.06	-0.04	-0.02	0	0.02	0.04	0.06	0.08	0.1
		Number of Iterations										
ΔV Deviation	-0.1	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	-0.08	81	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
	-0.06	50	66	30	21	21	62	40	25	48	23	21
	-0.04	10	15	15	18	19	19	21	22	23	24	27
	-0.02	4	6	8	7	8	8	9	8	10	10	9
	0	3	3	3	3	1	0	1	3	3	3	3
	0.02	12	12	11	11	10	11	11	11	11	10	9
	0.04	17	18	17	16	16	14	14	14	14	15	14
	0.06	17	25	13	11	14	10	19	10	NC	15	18
	0.08	11	11	49	61	74	10	11	NC	49	17	28
	0.1	9	25	21	8	8	8	11	7	13	8	8
		Number of Function Evaluations										
ΔV Deviation	-0.1	501	501	501	501	501	501	501	501	501	501	501
	-0.08	495	502	502	502	502	502	502	503	503	503	502
	-0.06	314	410	204	142	141	385	263	168	301	153	137
	-0.04	69	100	100	119	124	124	137	143	149	155	173
	-0.02	28	40	54	48	54	54	60	54	67	67	60
	0	21	21	21	21	9	3	9	21	21	21	21
	0.02	79	79	73	72	66	72	72	73	73	66	60
	0.04	109	116	110	104	103	91	91	91	91	97	91
	0.06	116	159	89	75	93	69	125	68	502	102	118
	0.08	76	76	312	382	469	73	78	502	304	117	178
	0.1	59	173	144	54	55	55	75	49	86	55	56

Note: “NC” indicates that the algorithm did not converge within 500 function evaluations.

Table 3. Number of Levenberg-Marquardt Targeter Iterations and Function Evaluations for Deviations in Initial Guess.

		Departure Phase Angle Deviation										
	(%)	-0.1	-0.08	-0.06	-0.04	-0.02	0	0.02	0.04	0.06	0.08	0.1
		Number of Iterations										
ΔV Deviation	-0.1	8	8	8	8	8	8	8	8	8	8	8
	-0.08	6	7	7	7	7	7	7	7	7	7	8
	-0.06	5	6	6	6	6	6	6	6	6	6	6
	-0.04	15	17	17	17	19	19	21	21	23	23	25
	-0.02	4	4	4	4	4	8	8	8	8	8	9
	0	1	1	1	1	1	0	1	1	1	1	1
	0.02	23	17	13	8	8	21	17	9	5	4	5
	0.04	8	8	8	8	8	8	8	8	8	8	8
	0.06	9	9	9	9	9	9	9	9	9	9	9
	0.08	2	4	5	7	40	4	9	4	7	7	7
	0.1	6	6	6	6	6	6	6	4	4	15	13
		Number of Function Evaluations										
ΔV Deviation	-0.1	39	39	39	39	39	39	39	39	39	39	39
	-0.08	32	35	35	35	35	35	35	35	35	35	39
	-0.06	28	32	32	32	32	32	32	32	32	32	32
	-0.04	63	70	70	70	77	77	84	84	91	91	98
	-0.02	21	21	21	21	24	39	39	39	39	39	43
	0	12	12	12	12	12	11	12	12	12	12	12
	0.02	91	70	56	38	38	84	71	43	28	21	29
	0.04	38	38	38	38	38	38	38	38	38	38	38
	0.06	42	42	42	42	42	42	42	42	42	42	42
	0.08	20	25	29	35	151	25	42	24	35	35	35
	0.1	32	32	32	32	32	32	32	25	25	63	56

Table 4. Number of Trust-Region Targeter Iterations and Function Evaluations for Deviations in Initial Guess.

		Departure Phase Angle Deviation										
	(%)	-0.1	-0.08	-0.06	-0.04	-0.02	0	0.02	0.04	0.06	0.08	0.1
		Number of Iterations										
ΔV Deviation	-0.1	12	12	12	12	12	12	12	12	12	12	12
	-0.08	21	21	21	21	21	21	21	21	21	21	21
	-0.06	18	18	19	21	17	13	20	28	23	20	20
	-0.04	7	7	7	7	7	7	7	7	7	8	12
	-0.02	6	6	6	6	10	13	13	13	10	10	10
	0	4	3	3	3	3	0	3	3	3	3	4
	0.02	10	10	11	11	10	10	14	10	13	7	9
	0.04	8	21	21	21	24	23	17	13	22	15	14
	0.06	14	15	15	15	15	15	15	15	15	15	14
	0.08	10	9	13	11	11	31	10	10	11	11	11
	0.1	15	15	15	15	15	15	15	15	15	15	13
		Number of Function Evaluations										
ΔV Deviation	-0.1	33	33	33	33	33	33	33	33	33	33	33
	-0.08	58	58	58	58	58	58	58	58	58	58	58
	-0.06	51	51	54	60	46	38	57	77	66	53	53
	-0.04	22	22	22	22	22	22	22	22	22	25	33
	-0.02	21	21	21	21	29	36	36	36	31	31	31
	0	15	12	12	12	12	3	12	12	12	12	15
	0.02	29	29	30	30	27	27	41	31	36	24	28
	0.04	23	58	58	58	67	62	48	38	61	42	39
	0.06	37	40	40	40	40	40	40	40	40	40	37
	0.08	25	24	34	30	30	86	27	27	30	30	30
	0.1	38	38	38	38	38	38	38	38	38	38	34

same radius and latitude of perilune (at ΔV = 3.1158 km/s and γ = 136.9°). Even over the range of ±10% deviations in ΔV and γ, the trust-region method converged nearly 75% of the time, although for the majority of those cases, it converged toward the second solution.

The accuracy of the targeted solution with the point-mass gravitational model was then checked against the more realistic Earth oblateness model. The perilune conditions for the more accurate trajectory were significantly different, with a perilune radius of 7910 km (a miss of over three lunar radii) and latitude of 5.56° (a miss of 2.56°). Clearly, the target parameters are highly sensitive to the non-spherical Earth gravitational perturbations. This was expected for the inclination and latitude. A better measure of the extent of this error, though, would be to determine the corrective midcourse ΔV required to retarget the desired orbital parameters. For instance, the spacecraft's guidance program could be used to calculate the ΔV necessary to put the spacecraft back on course as a result of the initial error from the simplified dynamic model. With a few modifications, the TLI targeting algorithms themselves could possibly be used for this purpose. Of course, this problem can be eliminated if the oblateness model is used for the predictor, but at the cost of excess computations at each time-step. It is also possible that the target miss errors from assuming a spherical Earth could be recovered by determining empirical relationships between the targets for both dynamical models. However, this would require extra computer overhead to store these relationships, either in tables or curve-fits.

CONCLUSIONS AND RECOMMENDATIONS

Ultimately, the Gauss-Newton and Levenberg-Marquardt correctors were determined to be too computationally costly, since they require so many more integrations of the equations of motion than the trust-region and Broyden methods. The trust-region method proved to be the most versatile corrector, much more likely to converge than the Broyden method. Moreover, poor selection of the initial step-size for the Jacobian for Broyden's method was detrimental to convergence. However, Broyden's method, when it did converge, did so with by far the fewest function evaluations. This is a significant advantage, since each function evaluation requires a lengthy numerical integration by the predictor. If the Broyden method is to be effective, however, future efforts must identify a better way of determining the initial guess, since only with very close initial guesses does the Broyden method converge. Either some feasible starting conditions must be tabulated or better analytical approximations must be incorporated into the algorithm.

Although much work remains to be done, the existing Broyden and trust-region targeters do at least provide a proof of concept for the feasibility of this type of TLI targeting algorithm. Future research will explore several options to significantly improve the targeter's versatility, reduce its required computation time, and verify its accuracy. As suggested, variational methods to approximate the Jacobian and additional methods of storing or calculating good first guesses will be studied to improve convergence rates. The targeter incorporating variable flight-path angle will be studied in detail. It is suspected that the trajectory (and the ΔV required) will be highly sensitive to changes in flight-path angle. The targeter will also be expanded to include a wider range of scenarios, such non-coplanar trajectories, and more targeted parameters, such as a free return to Earth. Less computationally costly ephemerides (if they can be found) will likely be incorporated into the predictor. Also, the algorithm should operate much faster once the associated codes are translated into a lower-level programming language required for implementation onboard a spacecraft. Finally, any completed targeting algorithm should be tested against existing high-fidelity models. This will ensure that the simplifying assumptions and numerical methods in the algorithm are responsible for only small mid-course corrections relative to other factors, such as sub-nominal engine burns.

Ultimately, a modification of these targeting algorithms is expected to provide the means for a spacecraft to conduct a TLI burn autonomously. Such an algorithm could reduce the need for uploading pre-calculated targeting information to the onboard computer and could bypass delays or disruptions in communication between the ground and the spacecraft. This would be incredibly useful in certain scenarios which would otherwise require an abort or alternate mission. In the future an algorithm of this kind could be applied to spacecraft on a wide variety of missions, provided they meet the minimum computational hardware and power consumption requirements.

REFERENCES

[1] Bentley, Earle, “Revisiting Apollo: Translunar Injection (TLI) Guided Burn to the Moon,*” AIAA-Houston Annual Technical Symposium*, NASA/JSC Gilruth Center, 19 May 2006, Presentation for Lunar Missions Session.

[2] Cooper, F. Don, “A Method of Implementing Cutoff Conditions for Saturn V Lunar Missions out of Earth Parking Orbit Assuming a Continuous Ground Launch Window,” NASA TM X-53268, May 25, 1965.

[3] Johnson, Francis, “An Empirical Simulation Equation Expressing the Characteristic Velocity of Optimum TLI Maneuvers for All Missions,” NASA TM X-72257, July 31, 1968.

[4] Yencharis, Jerome D., et al., “Apollo Experience Report – Development of Guidance Targeting Techniques for the Command Module and Launch Vehicle,” NASA TN D-6848, June 12, 1972.

[5] Marchand, Belinda G., et al., “Onboard Autonomous Targeting for the Trans-Earth Phase of Orion,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 943-956.

[6] Tewari, Ashish, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB® and Simulink®*, Birkhäuser, Boston 2007.

[7] Qiu-Dong, Wang, “The Global Solution of the N-Body Problem,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 50, No. 1, 1991, pp. 73-88.

[8] Folkner, William M., Williams, James G., and Boggs, Dale H., “The Planetary and Lunar Ephemeris DE 421,” Jet Propulsion Laboratory IPN Progress Report 42-178, Aug. 15, 2009.

[9] Gerald, Curtis F. and Wheatley, Patrick O., *Applied Numerical Analysis*, 7th ed., Pearson, Boston, 2004, pp. 359-364.

[10] Miele, A., Wang, T., and Mancuso, S., “Optimal Free Return Trajectories for Moon Missions and Mars Missions,” *Journal of the Astronautical Sciences*, Vol. 48, Nos. 2 and 3, 2000, pp. 183-206.

[11] Bate, Roger R., Mueller, Donald D., and White, Jerry E., *Fundamentals of Astrodynamics*, Dover, New York, 1971, pp. 321-352.

[12] Battin, Richard H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA, Virginia, 1999, pp. 437-446.

[13] Ocampo, Cesar and Munoz, Jean-Philippe, “Variational Equations for a Generalized Spacecraft Trajectory Model,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1615-1622.

APPENDIX

Variation of Perilune Radius with Parking Orbit Altitude

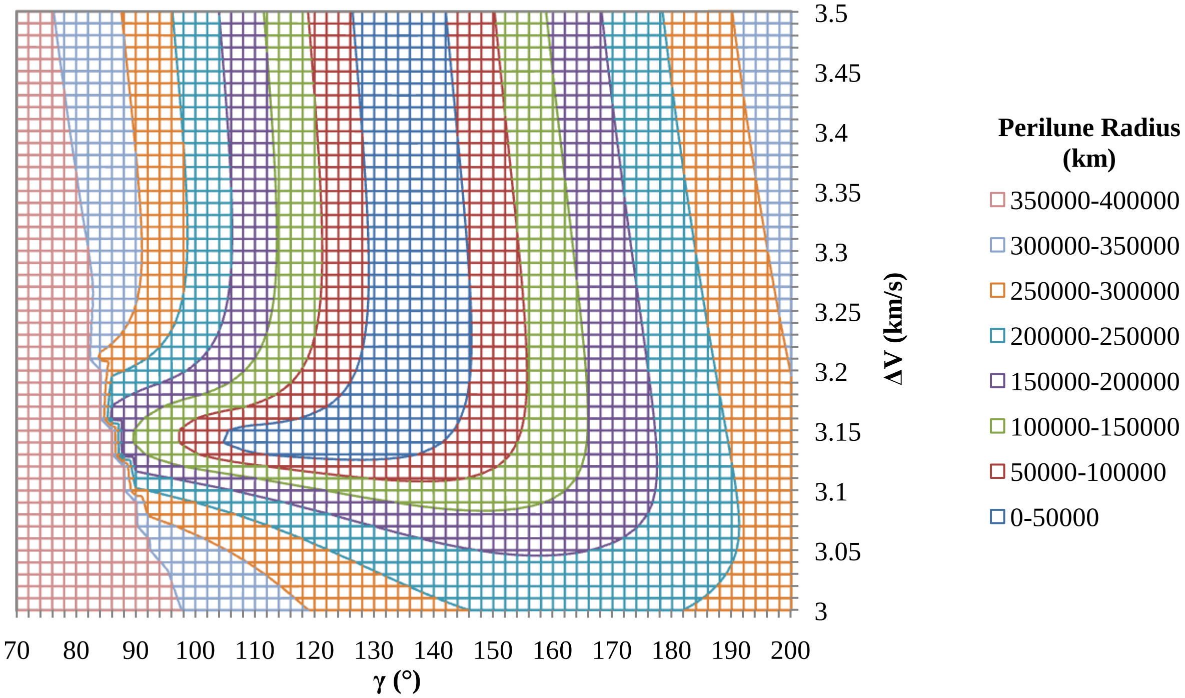

Figure A1. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, A = 150 km.

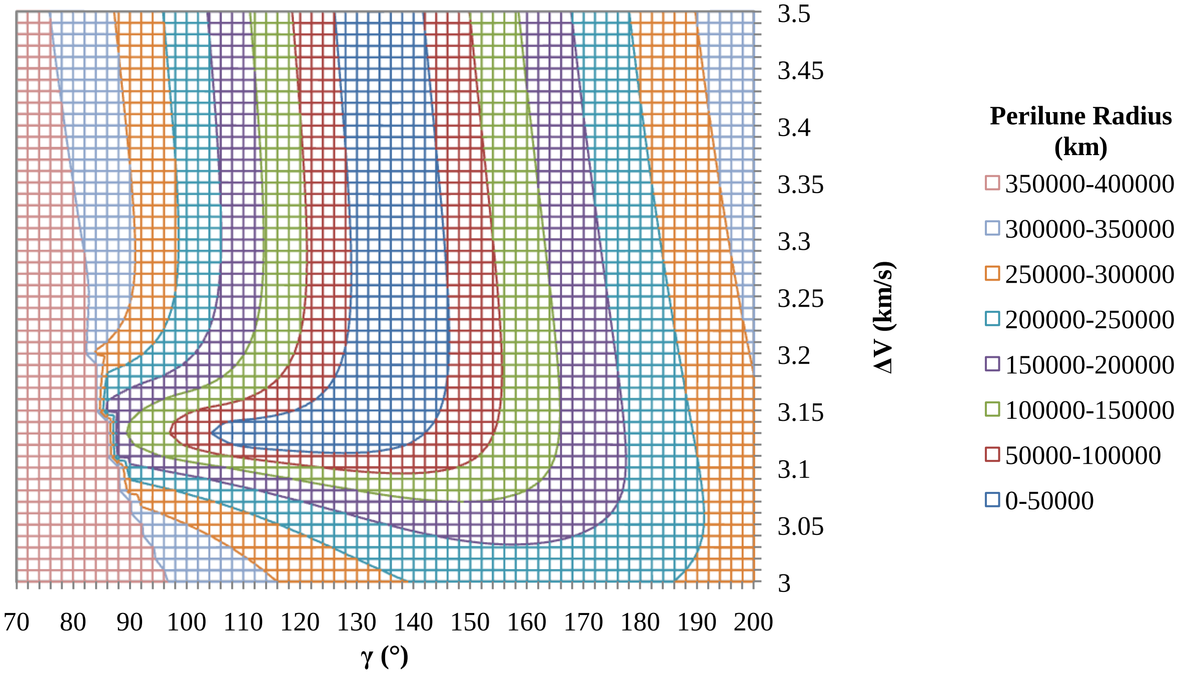

Figure A2. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, A = 200 km.

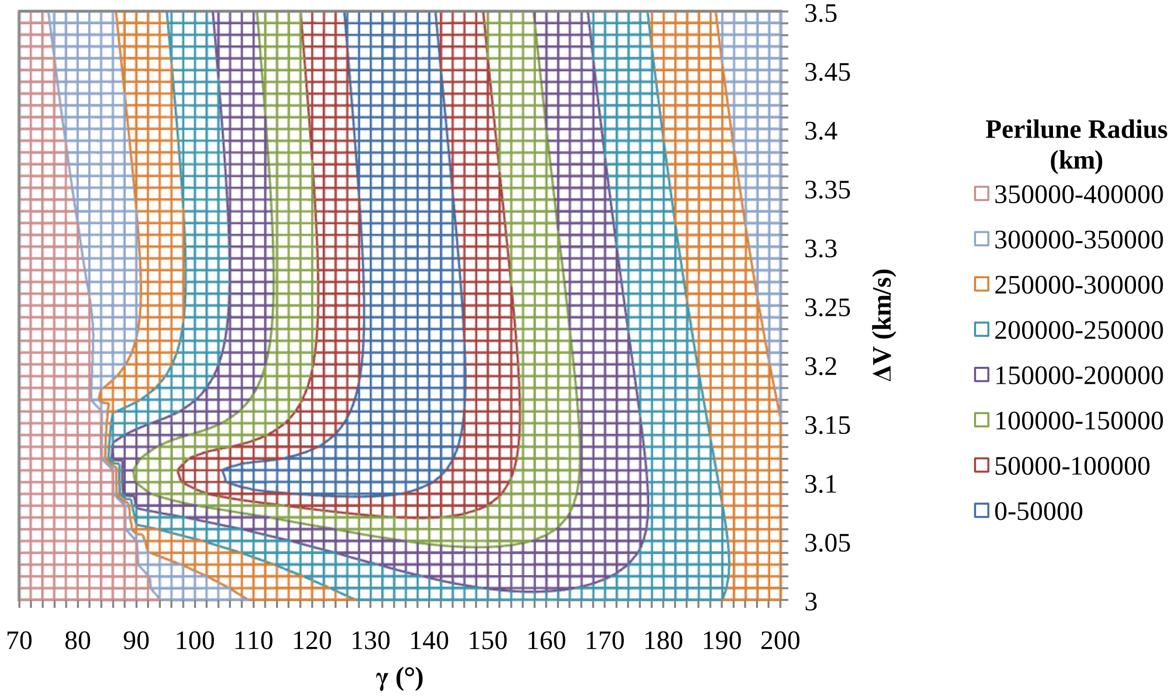

Figure A3. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, A = 300 km.

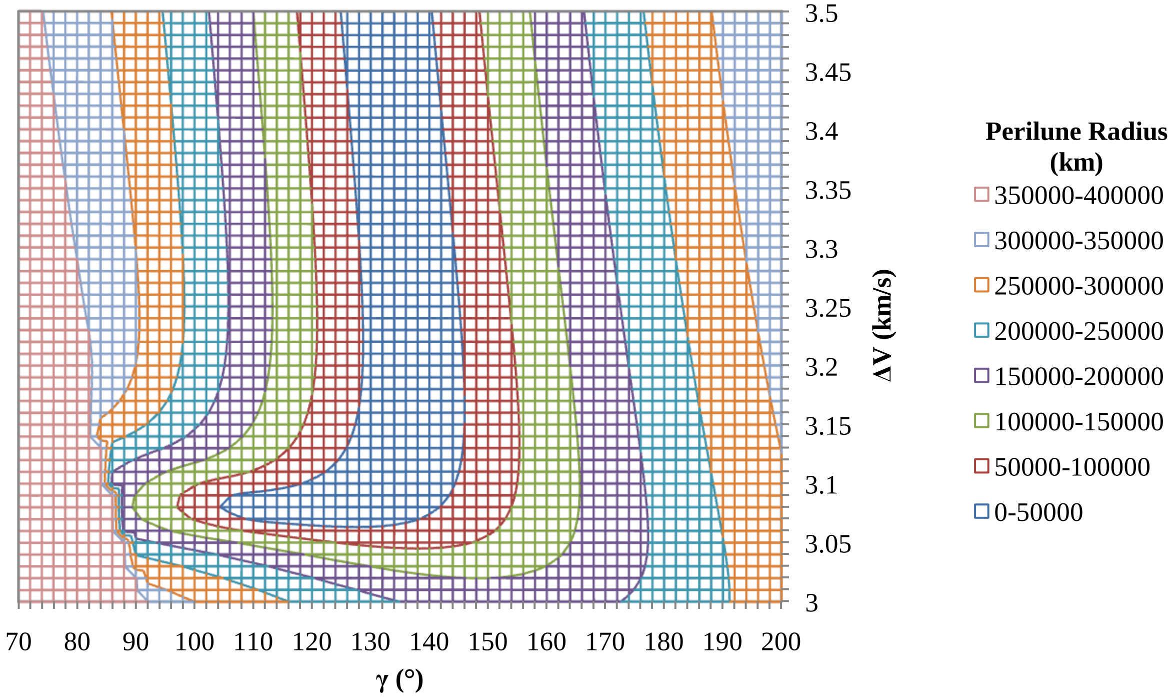

Figure A4. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, A = 400 km.

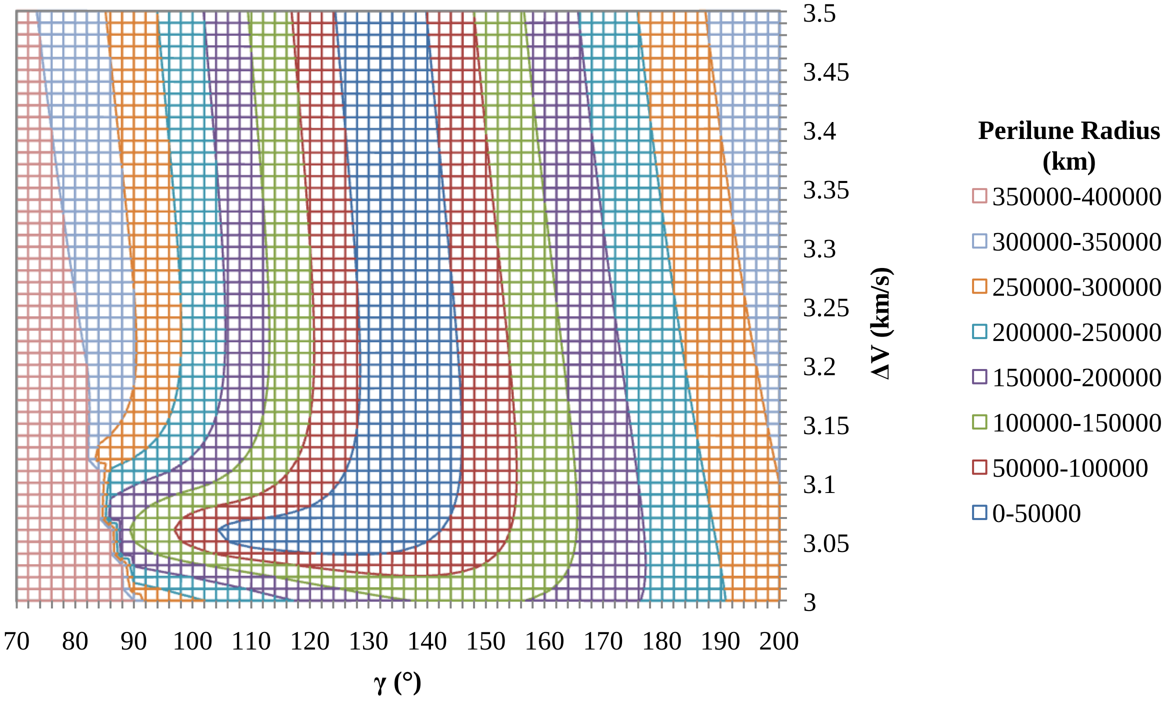

Figure A5. Radius of Perilune for Varying ΔV and γ, Point-mass Gravitational Model, A = 500 km.

VITA

Phillippe Reed was born in Clarksville, TN, to parents Robin and Julia Reed on March 8, 1985 and spent most of his childhood in Clarksville. After graduation from Clarksville High School in May 2003, he attended the University of Tennessee to pursue a degree in aerospace engineering and Latin. While at UT, he also took up coursework in German and Italian, and spent a summer studying Roman history in Rome. He led a senior design team to perform preliminary mission analysis for sending a spacecraft to Enceladus. After graduation in May 2007, he remained at UT another year to continue his studies in German, Italian, and classics. After moving back to Clarksville and writing an associate's degree curriculum in aviation engineering for North Central Institute, he returned to the University of Tennessee to pursue a master's degree in aerospace engineering and focus his thesis research on astrodynamics. He was married to Emily Maddox on May 29, 2010, and they have recently moved to Charleston, South Carolina.