AIAA/AAS Astrodynamics Specialist Conference and Exhibit, August 16–19, 2004, Providence, Rhode Island

Parameterization Effects on Convergence when Optimizing a Low-Thrust Trajectory with Gravity Assists

T. Troy McConaghy* and James M. Longuski†

Purdue University, West Lafayette, Indiana 47907-2023

Low-thrust trajectories can be modeled by an evenly-spaced sequence of ΔV impulses connected by coasting arcs. Such a model transforms the trajectory optimization problem from an optimal control problem into a finite nonlinear programing problem. Analysis and experimental results are given to help decide which coordinate system to use for the ΔV vectors. In particular, we consider an Earth-Mars flyby mission, an Earth-Tempel 1 rendezvous mission, an Earth-Mars-Ceres rendezvous mission, an Earth-Mercury rendezvous mission, and an Earth-Jupiter flyby mission. If the initial guess is good (i.e. nearly optimal), we find that spherical coordinates lead to the fastest optimization convergence. If the initial guess is bad (i.e. not close to feasible), we find that a feasible solution will be found most quickly if Cartesian coordinates are used. While it would not be prudent to attempt any generalizations, we provide some conjectures for the observed behavior based on the nature of the coordinates we have investigated.

Nomenclature

$\mathbf{d}$	Direction vector
g	Standard acceleration due to gravity, 9.80665 m/s^2
I_{sp}	Engine specific impulse, s
m	Spacecraft mass, kg
N	Number of segments on the trajectory
n_0	Number of segments that have nearly zero ΔV
$\mathcal{S}$	Set of segment numbers having a nonzero ΔV
V_∞	Hyperbolic excess speed, km/s
$\Delta\mathbf{V}$	Impulsive velocity-change vector, km/s
ε	Fudge factor when using Cartesian coordinates, km^2/s^2
η	Constraint function associated with MDV coordinates
θ	Vector clock angle, deg
ϕ	Constraint function associated with MC coordinates
ψ	Vector cone angle, deg

Subscripts

i	Segment index
l	Lower bound
u	Upper bound
x	x-component
y	y-component
z	z-component

*Doctoral candidate, School of Aeronautics and Astronautics, 315 N. Grant St., Student member AIAA, Member AAS.

†Professor, School of Aeronautics and Astronautics, 315 N. Grant St., Associate Fellow AIAA, Member AAS.

I. Introduction

HUMANITY has only just begun to explore the distant places in our solar system, as traveling into space is still very difficult and expensive. For example, it costs about $36,000 per kilogram to launch a spacecraft into geostationary orbit.[1] Once a spacecraft is in interplanetary space, every kilogram is used to its fullest advantage. If a kilogram of propellant can be saved, then that mass can be reallocated to the scientific payload, thereby improving the scientific return of the mission. By varying control inputs like the launch date and the the thrust vector (which is a function of time), one can change the required propellant mass. The trajectory optimization problem is to find control inputs that achieve the goals of the mission, yet use the minimum propellant mass (so the final mass is maximized).

There are many approaches to solving the trajectory optimization problem. The books by Lawden,[2] Marec,[3] Bryson and Ho,[4] and Bryson[5] are excellent references. Betts[6] gives an overview of many common numerical methods; new ones continue to be developed.[7,8] In 1999, Sims and Flanagan described a new technique for preliminary optimization of trajectories.[9] They modeled a low-thrust trajectory as a sequence of impulsive ΔV maneuvers connected by conic arcs. In essence, continuous control is approximated by coasting arcs (long intervals with no control) plus a finite number of impulsive controls. We extend the technique of Sims and Flanagan in this paper.

Once a trajectory model is specified, there are more decisions to be made before the optimization problem is well-defined (i.e. parameterized). For example, one must choose the coordinate system for each vector (e.g. Cartesian or spherical). Since the majority of the optimization variables are associated with the impulsive ΔV vectors, the choice of coordinate system for these vectors is significant. We find that numerical optimization software converges differently depending on the parameterization of the problem. To determine how different parameterizations affect the reliability of the optimization and the speed of convergence, we consider several alternatives.

II. Models

The low-thrust trajectory model of Sims and Flanagan[9] makes a number of assumptions:

1. Each planet-planet leg of the trajectory is divided into a fixed number of equal-duration segments.
2. The effect of the engine thrust on a particular segment is modeled by an instantaneous ΔV maneuver at the midpoint of the segment. (The thrust is modeled as being zero at all other times.)
3. The spacecraft mass changes instantaneously at the midpoints of the segments. The new mass is calculated using the rocket equation.[10]
4. During the intervals of zero thrust, the only force acting on the spacecraft is the gravity of the Sun. Hence the spacecraft follows conic arcs during these intervals.
5. Gravity-assist maneuvers rotate the V_∞ vector instantaneously at the endpoints of the legs.
6. No ΔV maneuver is required to target particular flyby conditions (i.e. flybys can be targeted for free).

An optimizer is used to maximize the final mass of the spacecraft by varying the encounter times, the initial velocity vector, the ΔV vectors, the final velocity vector, the flyby conditions, and the spacecraft mass at each body. These control inputs cannot be chosen arbitrarily. There are a number of constraints:

1. The magnitude of the ΔV on a segment cannot exceed a maximum that depends on the thrust, the mass-flow rate, the segment duration, the engine duty cycle, and the spacecraft mass.
2. Flyby altitudes must be above a minimum height.
3. The first part of each leg is propagated forward to a prespecified matchpoint time and the last part of the leg is propagated backward to the same time. The position, velocity, and mass of the spacecraft must be the same just before and just after the matchpoint time. (The sudden changes in the spacecraft's velocity and mass occur at the segment midpoints, but the matchpoint time occurs at a segment endpoint.)

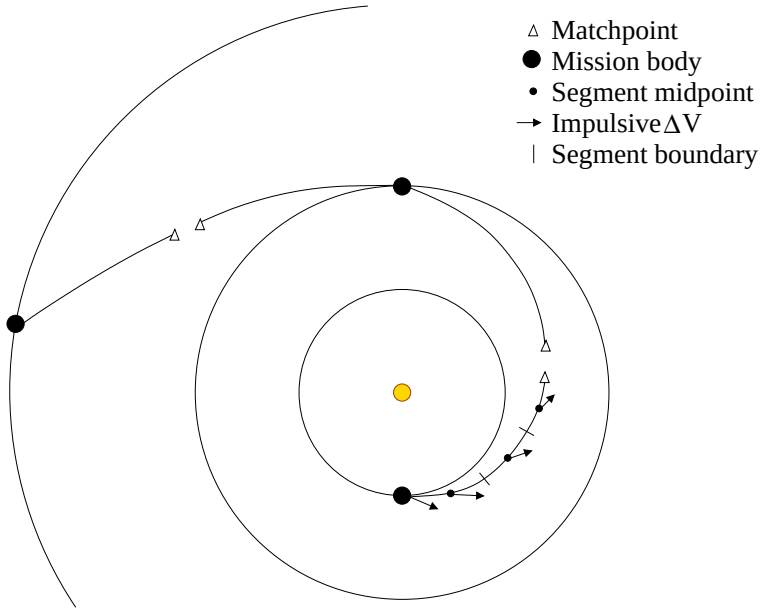

Figure 1. Trajectory model (after Sims and Flanagan[9]).

Figure 1 illustrates the key characteristics of the Sims-Flanagan trajectory model. We note that it has similarities with the models used by Kawaguchi et al.[11] and by Byrnes and Bright.[12]

In this paper, our solar array model is the same as the model Williams and Coverstone-Carroll used for their parametric studies.[13] We assume that the spacecraft uses no power, so all power is available for the engine. The models to calculate engine thrust and mass-flow rate (from the input power) are the same as those of Williams and Coverstone-Carroll.[13] The engine duty cycle is assumed to be 100%. The minimum power needed to operate the engine is assumed to be 0.649 kW and the maximum power usable by the engine is assumed to be 2.6 kW. We use these model parameters throughout, unless otherwise noted.

III. Parameterizing the Optimization Problem

So far, our description of the trajectory model has been geometric. For example, we said that the optimizer can vary the ΔV vectors. Unfortunately, optimization software is not designed to vary geometric vectors (i.e. entities with magnitude and direction). Optimization software is designed to vary an array of scalars. Therefore, each ΔV vector must be represented by a set of scalars (i.e. must be parameterized).

To parameterize a velocity vector, one must choose 1) a reference frame (three basis vectors), 2) an observer (since velocity is relative), and 3) a coordinate system (such as spherical coordinates). In our current software, we assume that the reference frame is the ecliptic and equinox of J2000 (i.e. the "inertial frame" with the x direction toward the first point in Aries and the z direction parallel to Earth's angular momentum vector on January 1, 2000). The observer of the initial spacecraft velocity is chosen to be the Earth. All other velocity vectors have the Sun as their observer. Since each ΔV vector is the difference between two velocity vectors, the motion of the observer cancels out. Therefore, ΔV vectors have no observer associated with them.

There are many coordinate systems for three-dimensional vectors. The most common examples are Cartesian coordinates, spherical coordinates, and cylindrical coordinates. These three are all examples of orthogonal coordinate systems, where surfaces of constant coordinate are orthogonal to each other. Other examples of orthogonal coordinates are conical coordinates, ellipsoidal coordinates and toroidal coordinates.[14] In our software, the initial velocity vector of the spacecraft is parameterized using spherical coordinates and all other velocity vectors are parameterized using Cartesian coordinates.

Since most of the optimization variables are associated with the ΔV vectors, we experiment with parameterizing them in four different coordinate systems. We will now give the definition of those four coordinate systems and then discuss their relative merits.

A. Cartesian Coordinates

The Cartesian coordinate ΔV_x of a ΔV vector is the component of the ΔV vector along the $\hat{\mathbf{x}}$ vector (which is one of the three basis vectors for the frame). The coordinates ΔV_y and ΔV_z have analogous definitions; formally:

$$\Delta V_x \equiv \Delta \mathbf{V} \cdot \hat{\mathbf{x}} \tag{1}$$

$$\Delta V_y \equiv \Delta \mathbf{V} \cdot \hat{\mathbf{y}} \tag{2}$$

$$\Delta V_z \equiv \Delta \mathbf{V} \cdot \hat{\mathbf{z}} \tag{3}$$

B. Spherical Coordinates

We denote the spherical coordinates of a ΔV vector by ΔV, θ, and ψ. The first coordinate, ΔV, is the magnitude of the ΔV vector, i.e.

$$\Delta V = \sqrt{\Delta \mathbf{V} \cdot \Delta \mathbf{V}} \tag{4}$$

The second coordinate, θ, is the "clock angle" from the $\hat{\mathbf{x}}$ vector to the projection of the ΔV vector onto the x-y plane. The third coordinate, ψ, is the "cone angle" from the $\hat{\mathbf{z}}$ vector to the ΔV vector. Figure 2 illustrates both the Cartesian and the spherical coordinates. To represent all possible ΔV vectors, θ can range from 0 deg to 360 deg and ψ can range from 0 deg to 180 deg. (Other choices for the ranges are possible.)

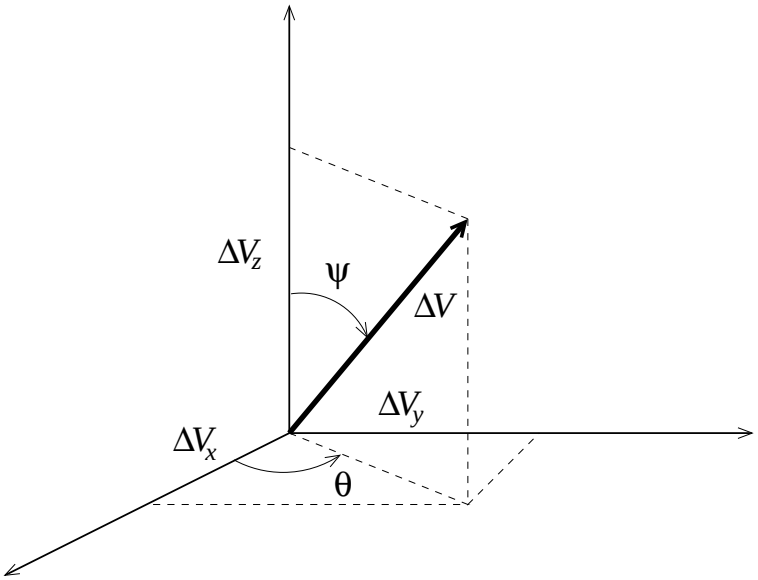

Figure 2. Cartesian and spherical coordinates.

C. Magnitude and Cartesian Coordinates

One can parameterize a ΔV vector by its magnitude and its three Cartesian coordinates: ΔV, ΔV_x, ΔV_y, and ΔV_z. These four coordinates are not independent; they must satisfy

$$\phi(\Delta V, \Delta V_x, \Delta V_y, \Delta V_z) \equiv \Delta V_x^2 + \Delta V_y^2 + \Delta V_z^2 - \Delta V^2 = 0 \tag{5}$$

We call these coordinates "Magnitude and Cartesian Coordinates" (MC coordinates).

D. Magnitude and Direction-Vector Coordinates

One can parameterize a ΔV vector by its magnitude and the Cartesian components of a vector that points in the same direction (which we call the "direction vector" $\mathbf{d}$): ΔV, d_x, d_y and d_z. We call these coordinates "Magnitude and Direction-Vector Coordinates" (MDV coordinates). The Cartesian components of a ΔV vector can be calculated using

$$\Delta V_x = \frac{\Delta V d_x}{d} \tag{6}$$

$$\Delta V_y = \frac{\Delta V d_y}{d} \tag{7}$$

$$\Delta V_z = \frac{\Delta V d_z}{d} \tag{8}$$

where $d \equiv \sqrt{d_x^2 + d_y^2 + d_z^2}$ is the length of the direction vector.

To prevent division by zero, the direction vector must not have zero length, i.e.

$$\eta(\Delta V, d_x, d_y, d_z) \equiv d_x^2 + d_y^2 + d_z^2 > 0 \tag{9}$$

Inequality (9) is the only required constraint associated with MDV coordinates. However, one may choose to impose additional constraints. We considered constraining the direction vector to lie between two spheres of radius d_l and d_u, i.e.

$$d_l^2 \leq \eta(\Delta V, d_x, d_y, d_z) \leq d_u^2 \tag{10}$$

where $d_l > 0$, and $d_u \leq \infty$. The direction vector can be constrained to be a unit vector by setting both d_l and d_u to 1.

When deciding which coordinate system to use for the ΔV vectors, there are many considerations to take into account. For example, since numerical optimization software generally slows down as the number of optimization variables increases, one usually wants to keep the number of optimization variables small. Suppose there are N segments on the trajectory. Then there are N ΔV vectors to parameterize. If Cartesian or spherical coordinates are used, then there will be $3N$ optimization variables associated with the ΔV vectors, but if MC or MDV coordinates are used, there will be $4N$ such variables (33% more).

Furthermore, numerical optimization software generally works faster and more reliably if there are fewer nonlinear constraints. Spherical and Cartesian coordinates have no associated constraints, but MC and MDV coordinates do [Eqs. (5) and (10)]. In fact, when using MC or MDV coordinates, there is a nonlinear constraint associated with *each* ΔV vector!

Cartesian coordinates are not without their problems. For example, the maximum ΔV allowable on segment 2 depends on the mass of the spacecraft as it approaches the midpoint of segment 2. That mass depends on the magnitude of the ΔV that occurred at the midpoint of segment 1. If the ΔV vectors are parameterized using Cartesian coordinates, then one calculates the magnitude of the ΔV on segment 1 using

$$\Delta V_1(\Delta V_{x1},\ \Delta V_{y1},\ \Delta V_{z1}) = \sqrt{\Delta V_{x1}^2 + \Delta V_{y1}^2 + \Delta V_{z1}^2} \tag{11}$$

The right-hand side of Eq. (11) seems fine at first, but closer inspection reveals that the derivative (with respect to ΔV_{x1}, for example) is undefined when $\Delta V_{x1} = 0$, $\Delta V_{y1} = 0$, and $\Delta V_{z1} = 0$. To see why, consider the special case when $\Delta V_{y1} = 0$ and $\Delta V_{z1} = 0$:

$$\Delta V_1(\Delta V_{x1},\ 0,\ 0) = \sqrt{\Delta V_{x1}^2} = |\Delta V_{x1}| \tag{12}$$

The derivative of the absolute value function does not exist at zero (i.e. it is not differentiable at zero). This is a problem because the numerical optimization software that we are using requires all constraint functions to be differentiable everywhere (or at least everywhere in the neighborhood of optimal solutions). The fact that the magnitude function is not differentiable at $(0, 0, 0)$ is pertinent because it is very common for an optimal trajectory to have some ΔV vectors that are zero.

One solution to the nondifferentiability of the magnitude function is to replace it with a differentiable approximation. We use

$$\Delta V_{\text{approx}}(\varepsilon;\ \Delta V_x,\ \Delta V_y,\ \Delta V_z) \equiv \sqrt{\Delta V_x^2 + \Delta V_y^2 + \Delta V_z^2 + \varepsilon} \tag{13}$$

where ε is a small "fudge" factor with the dimensions of velocity squared. Figure 3 shows the difference between the true magnitude and our differentiable approximation when $\varepsilon = 0.001$ km^2/s^2. When the ΔV vector is zero, the error in the approximation is the worst; it is $\sqrt{\varepsilon}$. Since the error is smaller when ε is smaller, one might conclude that ε should be made as small as possible. We have found that when ε is made smaller, our numerical optimization software takes longer to converge, so there is a tradeoff between error and convergence speed. A numerical example of this phenomenon is given in the next section.

Of course, the problem of the nondifferentiable magnitude function can be avoided by using a coordinate system where the magnitude is a differentiable function. Spherical, MC and MDV coordinates all have the magnitude as one of the coordinates, so the function to calculate the magnitude from the coordinates is differentiable. (Of the four coordinate systems we consider, only Cartesian coordinates have the problem of the nondifferentiable magnitude function.)

Another problem with Cartesian coordinates is that it is not as easy to constrain the magnitudes of the ΔV vectors. Constraining the magnitude is easy when using spherical, MC, or MDV coordinates because the magnitude is one of the optimization variables—all one does is put bounds on that optimization variable. (Bounds on optimization variables are easy to satisfy.) When using Cartesian coordinates, however, a nonlinear constraint is required to bound the ΔV magnitude.

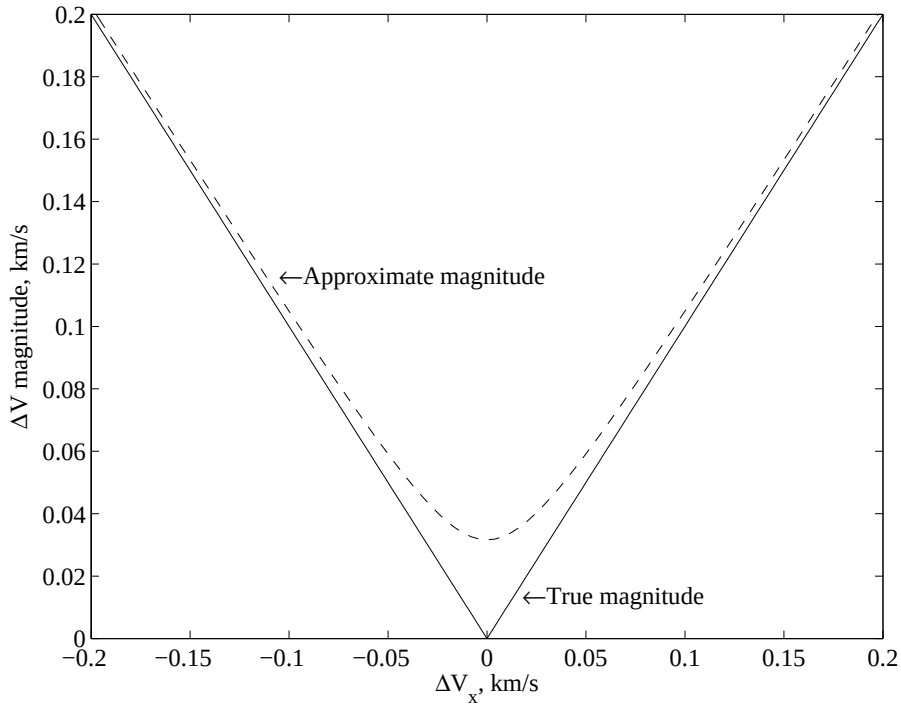

Figure 3. The true magnitude and our differentiable approximation ($\Delta V_y = \Delta V_z = 0, \varepsilon = 0.001$ km^2/s^2).

Spherical coordinates have some problems as well. For example, if the ΔV magnitude is zero (a common occurrence), then the two angles (θ and ψ) have no meaning. A change in θ or ψ has no effect on the objective or constraint functions, so the optimizer can make large changes to the angles with impunity—and it does! Then if the optimizer tries to make the ΔV magnitude nonzero, the ΔV vector may be pointed in a suboptimal direction (or worse). A similar problem happens if the cone angle ψ is 0 or 90 deg—the clock angle θ becomes meaningless (even if the ΔV magnitude is nonzero). MDV coordinates have an analogous problem, only it is the direction vector that becomes meaningless when the ΔV magnitude is zero. Cartesian

and MC coordinates, however, do not have problems with optimization variables that sometimes become meaningless.

Another problem with spherical coordinates arises when one puts bounds on the clock angle θ. For example, one might constrain θ to be between 0 deg and 360 deg. Suppose the optimal value of θ is 355 deg (which is the same as -5 deg) and the initial guess for θ is 2 deg. Then the optimizer will try to move θ from 2 deg to -5 deg, but it will run into the lower bound at 0 deg. (The optimizer will not increase θ from 2 deg to 355 deg.) The θ angle will get stuck at 0 deg (a suboptimal value). A similar problem can happen with bounds on the cone angle ψ. One solution to this problem is to remove the bounds on the angles, but that opens the door for another problem: the optimizer might make some of the angles very large (and large numbers cause numerical problems). We usually bound θ between -360 deg and 360 deg and ψ between 2 deg and 178 deg. We then check the "optimal solution" to make sure there are no angles stuck to their bounds. These "problems with angles" only arise with coordinate systems that have angles (so they do not arise with Cartesian, MC, or MDV coordinates).

There is a subtle yet devastating problem with MC coordinates. Recall that the MC coordinates associated with each ΔV vector must satisfy Eq. (5). In other words, the point $(\Delta V, \Delta V_x, \Delta V_y, \Delta V_z) \in \mathbb{R}^4$ is restricted to the set where Eq. (5) is satisfied (the feasible set). Once the optimizer finds a point in the feasible set, it tries to keep the current solution feasible by ensuring that all steps are tangential to the feasible set, that is, are orthogonal to the gradient of the constraint function:

$$\nabla\phi(\Delta V, \Delta V_x, \Delta V_y, \Delta V_z) = (-2\Delta V, 2\Delta V_x, 2\Delta V_y, 2\Delta V_z) \tag{14}$$

When the ΔV is zero, the gradient of the constraint function is zero, i.e.

$$\nabla\phi(0, 0, 0, 0) = (0, 0, 0, 0) \tag{15}$$

This is a problem because *all* steps are orthogonal to $(0, 0, 0, 0)$. The optimizer will probably take a step in a bad direction and arrive at a new point that violates the constraint. As will be seen in the next section, we had poor results when using MC coordinates and we believe that the problem of the constraint gradient being zero was one of the main culprits. MDV coordinates do not have this problem: their constraint gradient is never zero (on the feasible set).

To calculate the constraint functions, the Cartesian coordinates of the ΔV vectors must be calculated. To calculate Cartesian coordinates from MDV coordinates, one uses Eqs. (6)–(8). That transformation is not differentiable—there is a division by zero that occurs whenever $d = 0$. However, d is prevented from being zero by a constraint, so this is never a problem. The transformations from spherical to Cartesian and from MC to Cartesian coordinates are differentiable.

The transformations from Cartesian to spherical, MC, or MDV coordinates are not differentiable, but those transformations are never used to calculate the objective function or constraint functions, so we do not discuss them any further.

We see that Cartesian, spherical, MC, and MDV coordinates all have problems. To help us decide which coordinate system to use when parameterizing the ΔV vectors, we tried all four in a number of numerical experiments.

IV. Software

We call our trajectory optimization software the Gravity-Assist Low-thrust Local Optimization Program (GALLOP). The objective function (final mass), the constraint functions, and all their first derivatives are (analytically) calculated to double precision. That is, no first derivatives are estimated using finite-difference approximations. (We found that calculating all derivatives improves the speed and robustness of the optimization.) GALLOP has been used to optimize numerous trajectories.[15–19]

GALLOP uses the numerical nonlinear programming optimization software NPOPT, a wrapper for the sparse nonlinear optimizer SNOPT.[20] SNOPT is an implementation of the sequential quadratic programming (SQP) method to find a locally optimal solution to a constrained optimization problem.

When using NPOPT, one can specify some parameters which determine when or how an optimization run will end. One of these parameters is the "major feasibility tolerance." The smaller the major feasibility tolerance, the more accurately the constraints must be satisfied before NPOPT considers a solution feasible. Another of the NPOPT parameters is the "major optimality tolerance." The smaller the major

optimality tolerance, the more accurately the optimality conditions must be satisfied. An optimization run nominally ends once the current solution satisfies both the major feasibility tolerance and the major optimality tolerance. For the experiments described in this paper, we set both tolerances to 10^{-6} (unless otherwise noted).

One can also set a "major iterations limit," so that NPOPT will stop if it exceeds that number of major iterations. (We will not discuss the meaning of a major iteration versus a minor iteration here; the interested reader may consult Gill's paper.[20]) Sometimes, the optimizer cannot find a solution that is feasible or optimal enough. In those cases, other exit conditions may occur. An example is "EXIT – Infeasible problem, nonlinear infeasibilities minimized" (which may occur even if a feasible solution exists).

V. Numerical Results

A. The Effect of Changing ε

We begin our numerical experiments with an example of how ε [the fudge factor in Eq. (13)] affects the outcome when using Cartesian coordinates for the ΔV vectors. We use an Earth-Mars flyby trajectory launching on May 20, 2003 and arriving at Mars on Dec. 6, 2003. The launch V_∞ is 1.66 km/s and the initial spacecraft mass is 585 kg. The launch date, launch V_∞, initial mass and arrival date were all held frozen so the optimizer could not change them. We divided the trajectory into 50 equal-duration segments. To find the "true" maximum final mass, we optimized the trajectory using spherical coordinates for the ΔV vectors. Then we switched over to Cartesian coordinates and reoptimized the trajectory using a number of values for ε. Figures 4 and 5 show how the time to converge and the error in the maximum final mass depend on ε. When ε is increased from 10^{-15} to 10^{-4}, the convergence time decreases by a factor of about 15, whereas the error in the optimal final mass increases by a factor of about 10^5.

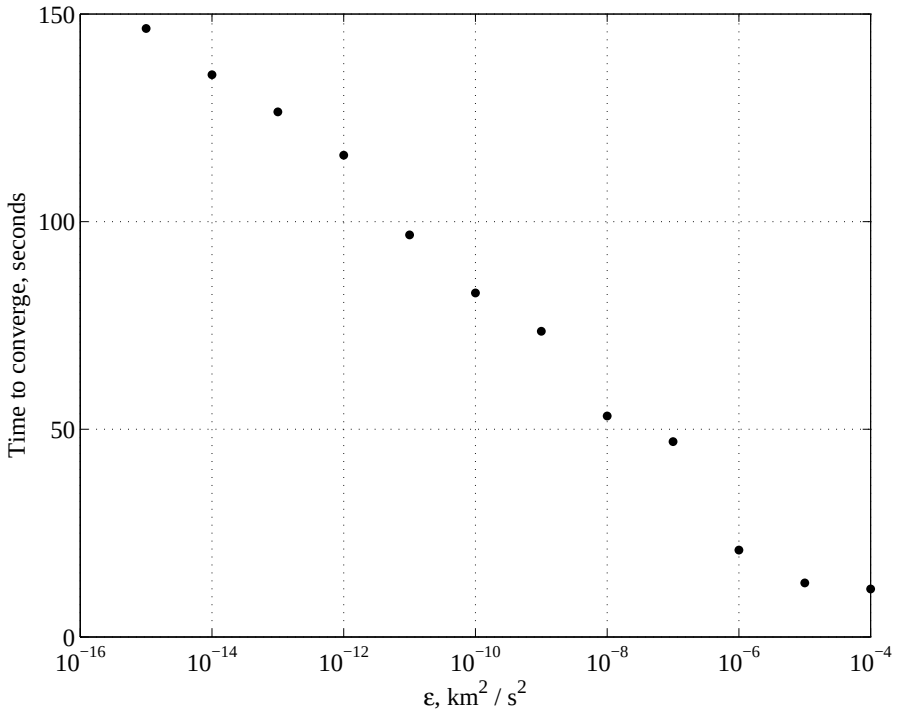

Figure 4. How convergence time depends on ε (for Earth-Mars flyby trajectory).

We can estimate the error in the final mass (when using Cartesian coordinates) for any trajectory. Recall that when using Cartesian coordinates, the magnitude of the i^{th} ΔV vector is calculated using the ΔV_{approx} function defined in Eq. (13).

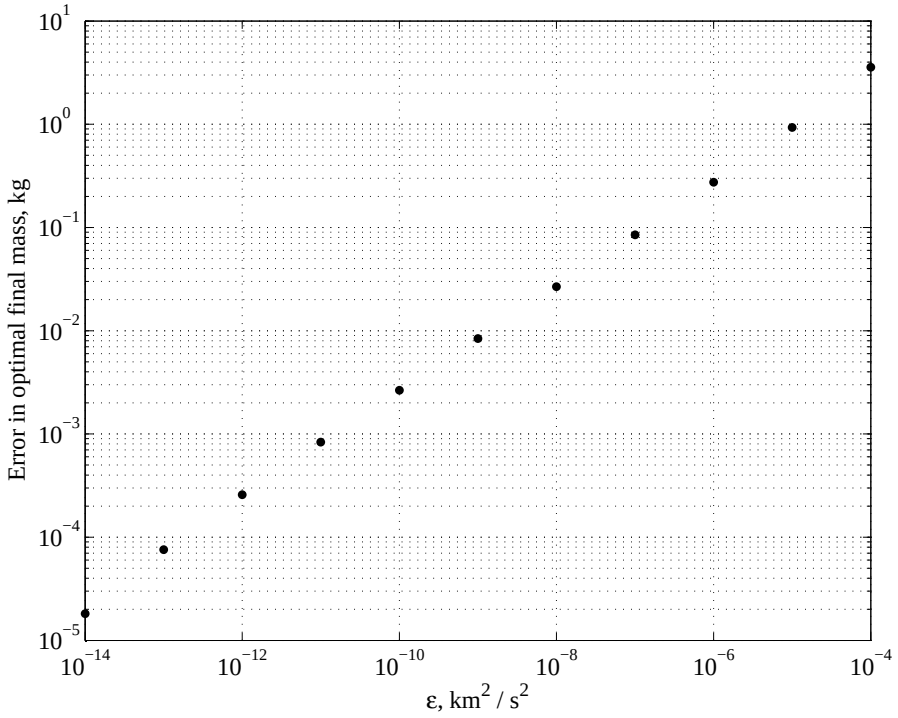

Figure 5. How error in the final mass depends on ε (for Earth-Mars flyby trajectory).

To estimate what the true final mass would be, one just uses $\varepsilon = 0$ when calculating the magnitudes of the ΔV vectors. (The optimizer can't use $\varepsilon = 0$ but there is nothing to prevent us from using $\varepsilon = 0$.) The estimated error is just the difference between the final mass calculated with $\varepsilon = 0$ and the final mass calculated with $\varepsilon > 0$.

$$m_{f\text{error}} \approx m_f^{\varepsilon=0} - m_f^{\varepsilon>0} \tag{16}$$

$$= m_0 \exp\left(-\sum_{i=1}^{N} \frac{\sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2}}{g I_{sp,i}}\right) - m_0 \exp\left(-\sum_{i=1}^{N} \frac{\sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2 + \varepsilon}}{g I_{sp,i}}\right) \tag{17}$$

where m_0 is the initial mass, m_f is the final mass, $g = 9.80665$ m/s^2 and $I_{sp,i}$ is the specific impulse of the engine on segment i. We note that the points on Fig. 5 are nearly collinear. To understand why, we make some simplifying assumptions. We begin by sorting the ΔV vectors into those that are nearly zero and those that are nonzero. Let n_0 be the number of ΔV vectors that are nearly zero. Let $\mathcal{S}$ be the set of segment numbers whose ΔV vectors are nonzero. Assume that

$$\sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2 + \varepsilon} \approx \sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2} \quad \text{for } i \in \mathcal{S} \tag{18}$$

Suppose also that the engine I_{sp} is constant throughout the trajectory. Equation (17) then becomes

$$\begin{aligned} m_{f\text{error}} &\approx m_0 \exp\left(-\sum_{i\in\mathcal{S}} \frac{\sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2}}{gI_{sp}}\right) \\ &\quad -m_0 \exp\left[-\left(\frac{n_0\sqrt{\varepsilon}}{gI_{sp}} + \sum_{i\in\mathcal{S}} \frac{\sqrt{\Delta V_{xi}^2 + \Delta V_{yi}^2 + \Delta V_{zi}^2}}{gI_{sp}}\right)\right] \qquad (19) \\ &= \left(m_f^{\varepsilon=0}\right) \cdot \left[1 - \exp\left(\frac{-n_0\sqrt{\varepsilon}}{gI_{sp}}\right)\right] \qquad (20) \\ &\approx \left(m_f^{\varepsilon=0}\right) \cdot \left[1 - \left(1 - \frac{n_0\sqrt{\varepsilon}}{gI_{sp}}\right)\right] \qquad (21) \\ &= \frac{n_0\sqrt{\varepsilon}\left(m_f^{\varepsilon=0}\right)}{gI_{sp}} \qquad (22) \end{aligned}$$

If we take the logarithm of both sides of Eq. (22), we find that

$$\log(m_{f\text{error}}) \approx \frac{1}{2}(\log\varepsilon) + \left\{\log\left[\frac{n_0\sqrt{\varepsilon}\left(m_f^{\varepsilon=0}\right)}{gI_{sp}}\right]\right\} \qquad (23)$$

Equation (23) has the linear form $y = mx + b$. This explains why the points on Fig. 5 (in which both axes are log-scale) are nearly collinear.

We set ε to 10^{-10} km^2/s^2 for all other experiments described in this paper. This value of ε was chosen to ensure that the optimal final mass would have a very small error. All cited convergence times associated with using Cartesian coordinates should be understood as being contingent on the choice of ε.

B. Convergence Speed when the Initial Guess is Nearly Optimal

When the initial guess is near an optimal solution, one expects the optimizer to converge to the optimal solution fairly quickly. In our experience with optimizing over 30 trajectories,[15–19] we have found that convergence (from a good initial guess) is fastest when using spherical coordinates for the ΔV vectors. We now describe experiments with four different trajectories which help quantify the differences in convergence speed associated with using different coordinate systems.

1. Earth-Mars Flyby Mission

To get a good (but non-optimal) initial guess for an Earth-Mars flyby mission we took an optimal solution (which had fixed launch and arrival dates) and moved the arrival date to a day later. The main characteristics of the mission are given in Table 1. All of the characteristics in Table 1 were held frozen (i.e. they were not optimization variables). Table 2 summarizes the outcomes when using different coordinate systems for the ΔV vectors. We note that when spherical coordinates were used, the optimizer converged more than eight times faster than with any other coordinates.

Table 1. Earth-Mars Flyby Mission

Characteristic	Value
Power available to the spacecraft at 1 AU	6 kW
Launch date	May 20, 2003
Launch V_∞	1.66 km/s
Initial mass, m_0	585 kg
Arrival date	Dec. 7, 2003[a]

[a] The initial guess came from an optimal solution that arrived on Dec. 6, 2003.

Table 2. Dependence of Outcome on Coordinate System for Earth-Mars Flyby Case

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	5.4 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	OSF[a]	1.4 min.
MC	OSF[a]	34 min.
MDV with $d_l = d_u = 1$	OSF[a]	50 sec.
MDV with $d_l = .1$ and $d_u = 10$	OSF[a]	2.0 min.
MDV with $d_l = 1$ and $d_u = \infty$	OSF[a]	44 sec.

[a] Optimal Solution Found ($m_f = 554.6$ kg).

2. Earth-Tempel 1 Rendezvous Mission

We wanted a test trajectory in which the spacecraft does a significant amount of out-of-plane thrusting. That way, all coordinates of the ΔV vectors would be fully exercised. A rendezvous mission to the comet Tempel 1 was chosen. When the spacecraft arrives at Tempel 1 on Dec. 4, 2005, Tempel 1 is 0.37 AU below the ecliptic plane. In order to get so far from the ecliptic, the ΔV vectors must often have a significant z-component. Table 3 summarizes the key characteristics of the Earth-Tempel 1 rendezvous mission. All characteristics in Table 3 were frozen during optimization. Table 4 summarizes the outcomes of the optimization runs when using various coordinate systems.

Table 3. Earth-Tempel 1 Rendezvous Mission

Characteristic	Value
Power available to the spacecraft at 1 AU	10 kW
Launch date	Nov. 28, 2002
Launch V_∞	0.831 km/s
Initial mass, m_0	576 kg
Arrival date	Dec. 4, 2005[a]

[a] The initial guess came from an optimal solution that arrived on Dec. 3, 2005.

Table 4. Dependence of Outcome on Coordinate System for Earth-Tempel 1 Rendezvous Case

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	13 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	OSF[a]	5.0 min.
MC	MILE[b]	2.4 hrs.
MDV with $d_l = d_u = 1$	"OSF"[c]	9.9 min.
MDV with $d_l = .1$ and $d_u = 10$	"OSF"[c]	27 min.
MDV with $d_l = 1$ and $d_u = \infty$	MILE[b,d]	2.8 hrs.

[a] Optimal Solution Found ($m_f = 387.4$ kg).

[b] Major Iteration Limit (2000) Exceeded.

[c] Optimal Solution Found, but the "optimal" final mass is less than 387.4 kg.

[d] Final solution was feasible but not optimal.

As in the Earth-Mars flyby case, the optimizer converged fastest when spherical coordinates were used. Cartesian coordinates came in a distant second, taking over 22 times longer than spherical coordinates. Two of the runs didn't converge to an optimal solution (despite being given over two hours to try). In two of the runs that used MDV coordinates, the optimizer found an "optimal" solution that was not as good as that found using spherical coordinates (e.g. $m_f = 372.2$ kg instead of 387.4 kg).

3. Earth-Mars-Ceres Rendezvous Mission

We wanted a test trajectory with a gravity-assist maneuver, so we chose an Earth-Mars-Ceres rendezvous similar to the one optimized by Sauer.[21] (Ceres is the largest asteroid in the asteroid belt.) Table 5 summarizes the key trajectory characteristics. We note that some of the power model parameters are a bit different from those given in the Model section. All values in Table 5 were held frozen during optimization.

Table 5. Earth-Mars-Ceres Rendezvous Mission

Characteristic	Value
Power available to the spacecraft at 1 AU	5 kW
Power used by the spacecraft	0.125 kW
Minimum power usable by the engine	0.5 kW
Launch date	May 6, 2003
Launch V_∞	1.6003 km/s
Initial mass, m_0	568 kg
Mars flyby date	Feb. 1, 2004
Lower bound on Mars flyby altitude	200 km
Arrival date	June 13, 2006[a]

[a] The initial guess came from an optimal solution that arrived on June 12, 2006.

The outcomes of the tests are summarized in Table 6. Once again, spherical coordinates led to the shortest convergence time, with Cartesian coordinates coming in second. Two of the cases using MDV coordinates found an "optimal" final mass lower than what was found using spherical coordinates (e.g. 413.0 kg vs. 414.0 kg). This seems to be a recurrent problem with MDV coordinates.

Table 6. Dependence of Outcome on Coordinate System for Earth-Mars-Ceres Rendezvous Case

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	3.2 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	OSF[a]	23 sec.
MC	MILE[b]	20 min.
MDV with $d_l = d_u = 1$	MILE[b]	29 min.
MDV with $d_l = .1$ and $d_u = 10$	"OSF"[c]	24 sec.
MDV with $d_l = 1$ and $d_u = \infty$	"OSF"[c]	24 sec.

[a] Optimal Solution Found ($m_f = 414.0$ kg).

[b] Major Iteration Limit (2000) Exceeded, but the final solution is nearly feasible and optimal.

[c] Optimal Solution Found, but the "optimal" final mass is less than 414.0 kg.

4. Earth-Mercury Rendezvous Mission

We also wanted a test case where the spacecraft makes multiple revolutions of the Sun (so that the ΔV vectors also make multiple revolutions). We chose an Earth-Mercury rendezvous mission where the launch V_∞ is zero. Figure 6 shows a plot of the trajectory. The dots are the segment midpoints and the line segments emanating from the dots indicate the direction and magnitude of the ΔV vector on the associated segment. During the 6.3-year flight time, the spacecraft makes almost 12 revolutions of the Sun. Table 7 summarizes the key characteristics of the Earth-Mercury rendezvous mission. Another important characteristic of this test is that we used 244 segments, so the number of optimization variables is significantly larger than in the other tests. We consider this test as being difficult for the optimizer. We note that the initial guess was created by taking an optimal solution and then moving the arrival time 0.1 days later (rather than 1 day later as we did for the other cases). All of the characteristics in Table 7 were held frozen during optimization. The outcomes when using different coordinate systems are summarized in Table 8.

Table 7. Earth-Mercury Rendezvous Mission

Characteristic	Value
Power available to the spacecraft at 1 AU	10 kW
Launch date	March 21, 2007 at 12:00 UT[a]
Launch V_∞	0 km/s
Initial mass, m_0	660 kg
Arrival date	Aug. 22, 2013 at 14:24 UT[a,b]

[a] Universal Time.

[b] The initial guess came from an optimal solution that arrived on Aug. 22, 2013 at 12:00 UT (0.1 days earlier).

The optimizer was only able to converge when spherical coordinates were used, and even then it took 8.3 minutes. All other runs were terminated by the user because they had taken too long and had not been making any progress.

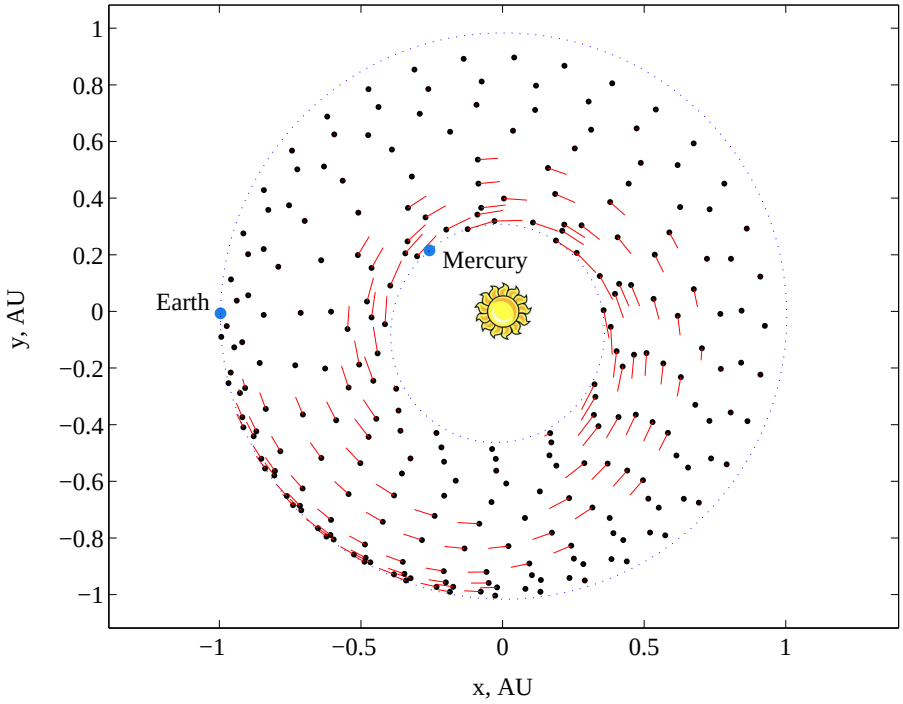

Figure 6. Earth-Mercury rendezvous trajectory.

Table 8. Dependence of Outcome on Coordinate System for Earth-Mercury Rendezvous Case

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	8.3 min.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	RTBU[b]	$>$ 12 hrs.
MC	RTBU[b]	$>$ 12 hrs.
MDV with $d_l = d_u = 1$	RTBU[b]	$>$ 28 hrs.
MDV with $d_l = .1$ and $d_u = 10$	RTBU[b]	$>$ 28 hrs.
MDV with $d_l = 1$ and $d_u = \infty$	RTBU[b]	$>$ 20 hrs.

[a] Optimal Solution Found ($m_f = 371.7$ kg).

[b] Run Terminated by User.

C. Convergence Speed when the Initial Guess is from STOUR-LTGA

Finding an initial guess trajectory for optimization can be a daunting task (especially if there are multiple gravity-assist maneuvers). Petropoulos et al.[22] developed an approach that assumes the low-thrust arcs belong to a low-dimensional family of curves, thereby enabling fast, automated, and comprehensive searches of the design space. Their approach was implemented in a program named STOUR-LTGA (the Satellite Tour design program for Low-Thrust Gravity-Assist trajectories).[23–25]

We often use STOUR-LTGA to find an initial guess trajectory. In this section, we describe convergence tests on two initial guesses found using STOUR-LTGA: An Earth-Mars-Ceres rendezvous and a Earth-Jupiter flyby.

1. Earth-Mars-Ceres Rendezvous

This test is similar to the Earth-Mars-Ceres rendezvous case used earlier, except the arrival date and some power-model parameters are different. Of course, the initial guesses of all the ΔV vectors are different as

well, since they came from STOUR-LTGA rather than a nearly optimal solution. Table 9 summarizes the key mission characteristics, all of which were held frozen during optimization. Table 10 summarizes the outcome when using different coordinate systems.

Table 9. Earth-Mars-Ceres Rendezvous Mission

Characteristic	Value
Power available to the spacecraft at 1 AU	10 kW
Power used by the spacecraft	0 kW
Minimum power usable by the engine	0.649 kW
Launch date	May 6, 2003
Launch V_∞	1.6003 km/s
Initial mass, m_0	568 kg
Mars flyby date	Feb. 1, 2004
Lower bound on Mars flyby altitude	200 km
Arrival date	June 12, 2006

Table 10. Dependence of Outcome on Coordinate System for Earth-Mars-Ceres Rendezvous Mission with STOUR-LTGA Initial Guess

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	6.2 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	OSF[a]	5.4 min.
MC	MILE[b]	1.5 hrs.
MDV with $d_l = d_u = 1$	"OSF"[c]	1.7 min.
MDV with $d_l = .1$ and $d_u = 10$	IPNIM[d]	13 min.
MDV with $d_l = 1$ and $d_u = \infty$	"OSF"[c]	1.0 min.

[a] Optimal Solution Found ($m_f = 435.3$ kg).

[b] Major Iteration Limit (5000) Exceeded.

[c] Optimal Solution Found, but the "optimal" final mass is less than 435.3 kg.

[d] Infeasible Problem, Nonlinear Infeasibilities Minimized.

Once again, using spherical coordinates led to the shortest convergence time. We note that one of the runs using MDV coordinates led to the exit condition "Infeasible problem, nonlinear infeasibilities minimized," which is clearly not true, since we know that a feasible (and optimal) solution exists.

2. Earth-Jupiter Flyby

Our second test using an initial guess from STOUR-LTGA is for an Earth-Jupiter flyby mission. The engine and power models for this test are significantly different from our other tests. We assume that the engine thrust is constant at 2.26 N and the engine I_{sp} is constant at 6000 s. Such parameters could be achieved, for example, by a large electric thruster being powered by a nuclear reactor. Table 11 summarizes the key trajectory characteristics (which were all held frozen). Table 12 shows the outcomes when using the different ΔV coordinates.

Spherical coordinates are clearly the coordinates of choice when the initial guess is nearly optimal or from STOUR-LTGA. We are not sure why. The fact that using spherical coordinates leads to an optimization problem with fewer optimization variables and fewer nonlinear constraints is probably a factor, but Cartesian coordinates tie with spherical coordinates on those counts. We suspect that part of the explanation is that when using spherical coordinates, the optimization variables associated with the ΔV magnitudes are decoupled from the optimization variables associated with the ΔV directions. Since the changes in spacecraft mass depend only on the ΔV magnitudes, the mass calculations are decoupled from the optimization variables

Table 11. Earth-Jupiter Flyby Mission

Characteristic	Value
Engine Thrust	2.26 N
Engine I_{sp}	6000 s
Launch date	April 21, 2022
Launch V_∞	0.4 km/s
Initial mass, m_0	20,000 kg
Arrival date	Dec. 11, 2028

Table 12. Dependence of Outcome on Coordinate System for Earth-Jupiter Flyby Mission with STOUR-LTGA Initial Guess

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical	OSF[a]	3.7 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	MILE[b]	23 min.
MC	MILE[b]	43 min.
MDV with $d_l = d_u = 1$	MILE[b]	18 min.
MDV with $d_l = .1$ and $d_u = 10$	MILE[b]	25 min.
MDV with $d_l = 1$ and $d_u = \infty$	MILE[b]	40 min.

[a] Optimal Solution Found ($m_f = 15,101$ kg).

[b] Major Iteration Limit (5000) Exceeded.

associated with the ΔV directions. Optimization problems with fewer couplings are generally easier to solve. (It is easier to solve a system of equations where some of the variables do not appear in some of the equations, i.e. where the Jacobian matrix of derivatives has some zero elements.)

D. Convergence Speed when the Initial Guess is Very Poor

So far, all of our tests have used an initial guess trajectory with ΔV vectors being nearly optimal or fairly good. We wanted some tests with very poor initial guesses for the ΔV vectors. To construct a very poor initial guess, we took a nearly optimal initial guess (as described in section V. B.) and corrupted its ΔV vectors. To corrupt a ΔV vector, we 1) changed its magnitude to a value sampled uniformly randomly between its lower and upper bound, and 2) changed its direction to a random direction distributed uniformly on the surface of a sphere (accomplished by sampling from a three-dimensional standard normal distribution and then normalizing). All other characteristics of the mission, such as the launch date and launch V_∞ were held frozen during the optimization.

We wanted to determine which coordinate system to use to quickly find a fairly feasible solution. (Once we find a feasible solution, we have already determined that spherical coordinates should be used to get quick convergence to an optimal solution.) To find a "fairly" feasible solution, we set the feasibility tolerance to 10^{-4} (rather than the usual 10^{-6}). We also directed the optimizer (NPOPT) to find a "feasible point only" (and to ignore the objective function). The major optimality tolerance was set quite large (e.g. 10^0) so it would be met easily.

1. Earth-Mars Flyby

Our first test with a very poor initial guess (for the ΔV vectors) is based on the Earth-Mars flyby mission whose key characteristics were given in Table 1. The test outcomes are shown in Table 13. We note that there are two runs with spherical coordinates, one with angles bounded and one with angles unbounded.

In this test, a fairly feasible solution was found fastest when using Cartesian coordinates. MC coordinates came in second and spherical coordinates with angles unbounded came in third. (This is the first time

Table 13. Dependence of Outcome on Coordinate System for Earth-Mars Flyby Mission with Very Poor Initial Guess

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical, angles bounded	FPF[a]	27 sec.
Spherical, angles not bounded	FPF[a]	19 sec.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	FPF[a]	2.7 sec.
MC	FPF[a]	17 sec.
MDV with $d_l = d_u = 1$	IPNIM[b]	0.2 sec.
MDV with $d_l = .1$ and $d_u = 10$	FPF[a]	66 sec.
MDV with $d_l = 1$ and $d_u = \infty$	FPF[a]	19 min.

[a] Feasible Point Found.

[b] Infeasible Problem, Nonlinear Infeasibilities Minimized.

spherical coordinates did not win.) Figure 7 shows how the ΔV magnitudes changed between the (very poor) initial guess and the (fairly feasible) solution found with Cartesian coordinates. As mentioned earlier, we expect that the feasible solution can be quickly optimized using spherical coordinates (and the corresponding optimal ΔV magnitude plot would exhibit a smoother profile).

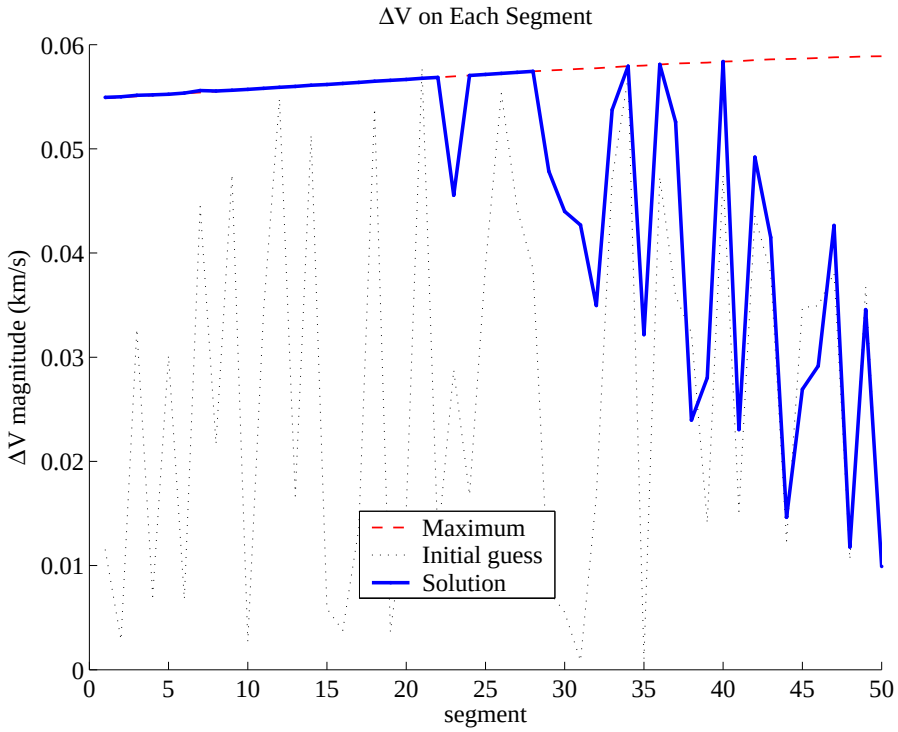

Figure 7. Earth-Mars flyby ΔV magnitudes on the segments (for fairly feasible solution).

2. Earth-Tempel 1 Rendezvous

Our second test with a very poor initial guess (for the ΔV vectors) is based on the Earth-Tempel 1 rendezvous mission whose key characteristics were given in Table 3. The test outcomes are shown in Table 14.

As with the Earth-Mars flyby test case, a fairly feasible solution was found fastest when using Cartesian coordinates. Unbounded spherical coordinates came in second, but took ten times longer than Cartesian.

Table 14. Dependence of Outcome on Coordinate System for Earth-Tempel 1 Rendezvous Mission with Very Poor Initial Guess

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical, angles bounded	FPF[a]	2.8 min.
Spherical, angles not bounded	FPF[a]	2.5 min.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	FPF[a]	15 sec.
MC	FPF[a]	4.1 min.
MDV with $d_l = d_u = 1$	FPF[a]	3.2 min.
MDV with $d_l = .1$ and $d_u = 10$	FPF[a]	4.0 min.
MDV with $d_l = 1$ and $d_u = \infty$	MILE[b]	2.5 hrs.

[a] Feasible Point Found.

[b] Major Iteration Limit Exceeded.

When using spherical coordinates with the angles bounded, the final (fairly feasible) solution had many ΔV vectors with angles on their lower or upper bounds (which is undesirable because angles tend to get stuck to their bounds).

3. Earth-Mars-Ceres Rendezvous

Our final test with a very poor initial guess (for the ΔV vectors) is based on the Earth-Mars-Ceres rendezvous mission whose key characteristics were given in Table 5. The test outcomes are shown in Table 15.

Table 15. Dependence of Outcome on Coordinate System for Earth-Mars-Ceres Rendezvous Mission with Very Poor Initial Guess

Coordinate System	Exit Condition	Convergence/Exit Time
Spherical, angles bounded	IPNIM[a]	5.9 sec.
Spherical, angles not bounded	FPF[b]	1.0 min.
Cartesian with $\varepsilon = 10^{-10}$ km^2/s^2	FPF[b]	8.0 sec.
MC	FPF[b]	4.4 min.
MDV with $d_l = d_u = 1$	FPF[b]	13 sec.
MDV with $d_l = .1$ and $d_u = 10$	IPNIM[a]	2.5 min.
MDV with $d_l = 1$ and $d_u = \infty$	FPF[b]	46 sec.

[a] Infeasible Problem, Nonlinear Infeasibilities Minimized.

[b] Feasible Point Found.

Once again, convergence to a fairly feasible solution was fastest with Cartesian coordinates. The "solution" found using spherical coordinates with angles bounded had many angles stuck to their bounds.

We note that in the three test cases with a very poor initial guess for the ΔV vectors, ε (the fudge factor associated with Cartesian coordinates) was set to 10^{-10} km^2/s^2. Since we were not too concerned with the accuracy of the solution, we could have set ε much larger, thus reducing the convergence time. Despite our small value for ε, when we used Cartesian coordinates for the ΔV vectors, the optimizer converged from a very poor initial guess to a fairly feasible solution the quickest. The reason for this may be that Cartesian coordinates never have the problem of some coordinates becoming meaningless. (Recall that when using spherical coordinates, the clock angle θ becomes meaningless when the vector is on the z-axis, and both angles become meaningless when the magnitude is zero.)

VI. Future Work

Changing the coordinate system used to parameterize the ΔV vectors is an example of changing the representation of the design space. In the future, we would like to investigate the utility of other representations. Possibilities include:

1. Using a reference frame that rotates with the spacecraft orbit (for the ΔV vectors). Such a feature would make it easy, for example, to move an optimal Earth-Mars transfer one synodic period later (because the ΔV vectors in the spacecraft-rotating frame would be a good initial guess for the optimal trajectory one synodic period later). Also, by using a spacecraft-rotating frame, it is easy to create an initial guess in which all ΔV vectors are along the direction of the spacecraft velocity vector.

2. Using parametric functions of time (e.g. polynomials or Fourier series) to calculate the coordinates of the ΔV vectors (with one function per coordinate and with the function parameters being the optimization variables). The number of function parameters could be smaller than the number of segments, thus reducing the number of optimization variables (but potentially making the problem infeasible).

3. Using different observers, reference frames, or coordinate systems for the spacecraft velocity vectors (such as the incoming velocity at a flyby). For example, by changing the observer of an incoming velocity from the Sun to the flyby planet, one can constrain the flyby V_∞ simply by putting bounds on the optimization variable associated with the V_∞ magnitude.

VII. Conclusions

1. The four coordinate systems we considered (Cartesian, spherical, MC, and MDV) all have problems.

2. When using Cartesian coordinates for the ΔV vectors, the value of the fudge factor ε strongly affects convergence time and the accuracy of the optimal final mass. (However, one can calculate a good estimate of the "true" final mass.)

3. When a good initial guess is available, spherical coordinates should be used to get fast convergence to an optimal solution.

4. When using MDV coordinates, the optimizer often finds an "optimal" solution with a final mass that is not as good as was found using spherical or Cartesian coordinates.

5. When the initial guess is not feasible and not very good, Cartesian coordinates should be used first to find a fairly feasible solution. Once a fairly feasible solution is found, spherical coordinates may be used to finish the optimization process.

6. When using spherical coordinates, bounds on the angles are not required.

Acknowledgments

This research has been supported in part by an award from the Purdue Research Foundation and in part by the Jet Propulsion Laboratory (JPL), California Institute of Technology, under Contract 1250863 (Jon A. Sims, Technical Manager). We are very grateful to Jon A. Sims, Greg J. Whiffen, Philip E. Gill, Paul A. Finlayson, Anastassios E. Petropoulos, Ed A. Rinderle, and Theresa D. Kowalkowski for their helpful advice and suggestions.

References

[1]Koelle, D. E., "Specific Transportation Costs to GEO—Past, Present and Future," *Acta Astronautica*, Vol. 53, No. 4–10, Aug. 2003, pp. 797–803.

[2]Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963.

[3]Marec, J.-P., *Optimal Space Trajectories, Studies in Astronautics, Vol. 1*, Elsevier Scientific, Amsterdam, 1979.

[4]Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control*, John Wiley & Sons, New York, 1979.

[5]Bryson, A. E., *Dynamic Optimization*, Prentice Hall, New Jersey, 1998.

[6]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, Mar.–Apr. 1998, pp. 193–207.

[7]Whiffen, G. J., and Sims, J. A., "Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization," 11th Annual AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 01-209, Santa Barbara, CA, Feb. 2001.

[8]Scheeres, D. J., Park, C., and Guibout, V. M., "Solving Optimal Control Problems with Generating Functions," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 03-575, Big Sky, MO, Aug. 2003.

[9]Sims, J. A., and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 99-338, Girdwood, AK, Aug. 1999.

[10]Tsiolkovsky, K. E., "Exploration of the Universe with Reaction Machines," (in Russian), *The Science Review*, #5, St. Petersburg, Russia, 1903.

[11]Kawaguchi, J., Takiura, K., and Matsuo, H., "On the Optimization and Application of Electric Propulsion to Mars and Sample and Return Mission," 4th Annual AAS/AIAA Spaceflight Mechanics Meeting, AAS Paper 94-183, Cocoa Beach, FL, Feb. 1994.

[12]Byrnes, D. V., and Bright, L. E., "Design of High-Accuracy Multiple Flyby Trajectories Using Constrained Optimization," AAS/AIAA Astrodynamics Conference, AAS Paper 95-307, Halifax, NS, Canada, Feb. 1995.

[13]Williams, S. N., and Coverstone-Carroll, V., "Benefits of Solar Electric Propulsion for the Next Generation of Planetary Exploration Missions," *Journal of the Astronautical Sciences*, Vol. 45, No. 2, 1997, pp. 143–159.

[14]Morse, P. M., and Feshbach, H., *Methods of Theoretical Physics, Part I*, McGraw-Hill, New York, 1953, pp. 659–666.

[15]McConaghy, T. T., Debban, T. J., Petropoulos, A. E., and Longuski, J. M., "Design and Optimization of Low-Thrust Trajectories with Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp. 380–387.

[16]Debban, T. J., McConaghy, T. T., and Longuski, J. M., "Design and Optimization of Low-Thrust Gravity-Assist Trajectories to Selected Planets," AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2002-4729, Monterey, CA, Aug. 2002.

[17]Chen, J. K., McConaghy, T. T., Okutsu, M., and Longuski, J. M., "A Low-Thrust Version of the Aldrin Cycler," AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2002-4421, Monterey, CA, Aug. 2002.

[18]Chen, J. K., McConaghy, T. T., Landau, D. F., and Longuski, J. M., "A Powered Earth-Mars Cycler with Three Synodic-Period Repeat Time," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 03-510, Big Sky, MO, Aug. 2003.

[19]Yam, C. H., McConaghy, T. T., Chen, K. J., and Longuski, J. M., "Preliminary Design of Nuclear Electric Propulsion Missions to the Outer Planets," AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2004-5393, Providence, RI, Aug. 2004.

[20]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[21]Sauer, C. G., "Solar Electric Performance for Medlite and Delta Class Planetary Missions," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 97-726, Sun Valley, ID, Aug. 1997. Also in *Advances in the Astronautical Sciences*, Univelt Inc., San Diego, CA, Vol. 97, Part II, 1997, pp. 1951–1968.

[22]Petropoulos, A. E., Longuski, J. M., and Vinh, N. X., "Shape-Based Analytic Representations of Low-Thrust Trajectories for Gravity-Assist Applications," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 99-337, Girdwood, AK, Aug. 1999. Also in *Advances in the Astronautical Sciences*, Univelt Inc., San Diego, CA, Vol. 103, Part I, 2000, pp. 563–581.

[23]Petropoulos, A. E., and Longuski, J. M., "Automated Design of Low-Thrust Gravity-Assist Trajectories," AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2000-4033, Denver, CO, Aug. 2000. Also in *A Collection of Technical Papers*, American Institute of Aeronautics and Astronautics, Reston, VA, pp. 157–166.

[24]Petropoulos, A. E., and Longuski, J. M., "A Shape-Based Algorithm for the Automated Design of Low-Thrust, Gravity-Assist Trajectories," AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 01-467, Quebec City, QC, Canada, July–Aug. 2001.

[25]Petropoulos, A. E., "A Shape-Based Approach to Automated, Low-Thrust, Gravity-Assist Trajectory Design," Ph.D. Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, May 2001.