The Superior Lambert Algorithm

Gim J. Der

DerAstrodynamics

Abstract

Lambert algorithms are used extensively for initial orbit determination, mission planning, space debris correlation, and missile targeting, just to name a few applications. Due to the significance of the Lambert problem in Astrodynamics, Gauss, Battin, Godal, Lancaster, Gooding, Sun and many others (References 1 to 15) have provided numerous formulations leading to various analytic solutions and iterative methods. Most Lambert algorithms and their computer programs can only work within one revolution, break down or converge slowly when the transfer angle is near zero or 180 degrees, and their multi-revolution limitations are either ignored or barely addressed. Despite claims of robustness, many Lambert algorithms fail without notice, and the users seldom have a clue why.

The *DerAstrodynamics* **lambert2** algorithm, which is based on the analytic solution formulated by Sun, works for any number of revolutions and converges rapidly at any transfer angle. It provides significant capability enhancements over every other Lambert algorithm in use today. These include improved speed, accuracy, robustness, and multi-revolution capabilities as well as implementation simplicity. Additionally, the **lambert2** algorithm provides a powerful tool for solving the angles-only problem without artificial singularities (pointed out by Gooding in Reference 16), which involves 3 lines of sight captured by optical sensors, or systems such as the Air Force Space Surveillance System (AFSSS).

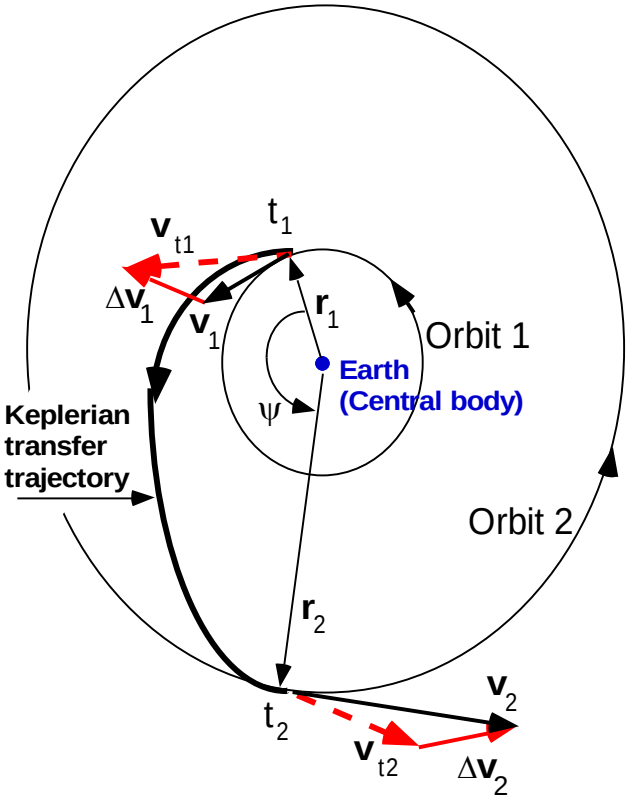
The analytic solution is derived from the extended Godal's time equation by Sun, while the iterative method of solution is that of Laguerre, modified for robustness. The Keplerian solution of a Lambert algorithm can be extended to include the non-Keplerian terms of the Vinti algorithm via a simple targeting technique (References 17 to 19). Accurate analytic non-Keplerian trajectories can be predicted for satellites and ballistic missiles, while performing at least 100 times faster in speed than most numerical integration methods.

Introduction

In 1991, Klumpp of JPL (Reference 10) compared the performance of numerous Lambert algorithms over the last two hundred years and declared the Gooding Lambert algorithm the winner. The Gooding Lambert algorithm (Reference 7), which originates from the Lancaster-Blanchard analytic solution (Reference 6), computes an initial guess of the universal iteration parameter for a high order Halley iterative method. Since Klumpp's limited (115 samples) results of the Gooding Lambert algorithm showed speed, accuracy, robustness, and applicability for multiple revolutions, one might be inclined to ask how *DerAstrodynamics* **lambert2** algorithm can be better. Simply put, it operates on the same theory, but with no starting and convergence problems, proven multi-revolution capability and includes easily more accurate non-Keplerian solutions of Vinti. The outstanding works of the late Professor Sun, regarding the multi-revolution Lambert problem, have been invaluable in developing this analytic solution.

Similar to the analytic solution of a Kepler algorithm, the analytic solution of a Lambert algorithm is not in closed form; an iterative method must be used to deduce a numerical solution. Contrary to the analytic solution of a Kepler algorithm, the analytic solution of a Lambert algorithm should be understood and visualized using any formulation, with or without universal variables. Any multiple-revolution Lambert problem has only elliptic solutions, and the conic solutions for trajectories less than one revolution can be simple, if the independent or unknown iteration parameter is chosen wisely. Since the initial value for the unknown iteration parameter of the **lambert2** algorithm can be visualized and bounded within known limits, there is no need for intelligent starters, averaging, or binary search methods. High-order iterative methods are needed for robustness, but it can be proved that failures still exist as in the examples of the *DerAstrodynamics* **kepler1** algorithm. In Klumpp's report, Conway indicated that all other high-order methods fail. Conway also claimed, without proof, the Laguerre iterative method has never been known to fail. In a space debris study, Der encountered many cases that the Conway-Laguerre iterative method can fail at the rate of one in a thousand. Der modified the implementation of the Laguerre iterative method to ensure Conway's infallibility claim, yet still achieve rapid convergence.

The **lambert2** algorithm is the fastest, most accurate, robust and multi-revolution Lambert algorithm today. If a Lambert algorithm uses the Newton method for the iterative procedure, then it cannot be robust. The principal advantage of the **lambert2** algorithm is that the unknown iteration parameters for any revolution can be visually estimated to within a very small range before the iterations begin. This presents a user with the feel and some physical meaning of the converged numerical solutions. In the unlikely event of failure, the user can easily find out the cause. The **lambert2** algorithm is also compact and contains only two functions. The lines-of-code for the main function is approximately 200, and that of the minimum time subroutine is about 50. It can solve for elliptic orbits of any revolution, and parabolic and hyperbolic orbits of less than one revolution.



The Lambert Problem

Given: $\mathbf{r}_1, \mathbf{r}_2, t_1, t_2$

Find: $\mathbf{v}(t_1) = \mathbf{v}_{t1}$, $\mathbf{v}(t_2) = \mathbf{v}_{t2}$

Then compute:

$\Delta\mathbf{v}_1 = \mathbf{v}_{t1} - \mathbf{v}_1$

$\Delta\mathbf{v}_2 = \mathbf{v}_2 - \mathbf{v}_{t2}$

Figure 1. An example of the input and output of a Lambert algorithm

A Lambert algorithm that computes essentially the Keplerian transfer trajectory as shown in Figure 1 can be extended to include non-Keplerian terms (J_2, J_3, most of J_4) of the Vinti algorithm via a simple targeting technique. Accurate analytic non-Keplerian trajectories can be predicted for satellites and ballistic missiles, while performing at least 100 times faster in speed than most numerical integration methods.

General Formulation of the Lambert Problem

The method of solving the problems of Kepler and Lambert are practically the same. The Kepler problem is an initial value problem of solving the equations of motion

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r} \tag{1}$$

to find the position and velocity vectors $\mathbf{r}(t_2)$ and $\mathbf{v}(t_2)$ at any given time t_2; given the initial position and velocity vectors $\mathbf{r}(t_1) = \mathbf{r}_1$ and $\mathbf{v}(t_1) = \mathbf{v}_1$ at a given initial time t_1, and μ is the gravitational constant for the central body. The classical theory ends with solving for the solution of one unknown, the eccentric anomaly E, in the Kepler Equation:

$$F(E) = E - e\sin E - M = 0 \tag{2}$$

where the mean anomaly M and the eccentricity e can be computed from the given times t_1, t_2 and the initial state vector of $\mathbf{r}_1$ and $\mathbf{v}_1$.

The Lambert problem is a two-point boundary value problem of solving the same equations of motion (1) to find the velocity vectors $\mathbf{v}(t_1)$ at a given initial time t_1 and $\mathbf{v}(t_2)$ at a given time t_2; given the position vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ at the respective times t_1 and t_2 (Figure 1). The classical theory (Gauss or Battin) solves two equations for two unknowns, but can be deduced to solve for one unknown, the semi-major axis, a, in the Lambert Equation:

$$F(a) = \sqrt{\frac{a^3}{\mu}}[(\alpha - \sin\alpha) - (\beta - \sin\beta)] - t = 0 \tag{3}$$

where $t = t_2$ and $t_1 = 0$ without loss of generality, and α and β are functions of the semi-major axis.

Using Sun's notations of Reference 5, the Lambert Equation for multi-revolution elliptic orbits can be expressed as:

$$F(x) = f(x) - f(y) + N\pi - \tau = 0 \tag{4}$$

where x is the only unknown or independent variable to be solved for, y is a function of x, τ is the normalized time computed from the given t_1, t_2, $\mathbf{r}_1$, $\mathbf{r}_2$, μ, and $N \geq 0$ is the orbit revolution number. When N = 0, equation (4) can be reduced to equation (3) with different notations, while x is essentially a function of the semi-major axis. This beautiful equation (4) provides the foundation to develop an algorithm and computational procedure for any revolution of elliptic orbits without difficulty. When N is greater than zero, parabolic and hyperbolic transfer orbits do not exist. The actual expressions of equation (4), including those for parabolic and hyperbolic orbits, are included in Appendix A.

The transcendental Kepler equation (2), and Lambert equations (3) and (4) do not allow any opportunity for closed form solutions, even though there is only one unknown. Numerous iterative methods that are specific to each formulation and independent variable exist, and the use of hyper-geometric functions is one of Battin's tricks. However, simple iterative methods such as those of Newton, Halley and Laguerre are available, if the first and/or the second derivatives of the functions, $F(E)$, $F(a)$, $F(x)$, can be derived.

The Laguerre method as stated in Reference 13 is intended to solve the roots of a polynomial equation of degree n. If the first and second derivatives of $F(x)$ are respectively $F'(x)$ and $F''(x)$, then the Lambert equation (4) can be solved by the iterative formula

$$x_{i+1} = x_i - \frac{n\,F(x_i)}{F'(x_i) \pm \frac{F'(x_i)}{|F'(x_i)|}\sqrt{(n-1)^2\left(F'(x_i)\right)^2 - n(n-1)\,F(x_i)F''(x_i)}} \quad \text{for } i = 1, 2, .. \quad (5)$$

where the degree n = 1, 2, . . . The sign ambiguity in equation (5) is resolved by taking the sign of the numerical value of $F'(x)$. If n = 1, then equation (5) is reduced to the iterative method of Newton. With an open mind and the lightning speed of a modern computer, robustness can be achieved by varying n. When Newton's method is used, a reasonable initial guess of x_1 must be estimated, and some authors have developed sophisticated formulae just for this purpose. Similarly, Gooding of Reference 7 also used complicated starting guesses for the Halley method. Unlike the Newton and Halley methods, the Laguerre method requires that the initial starting value, x_1 for i = 1 in equation (5) can be "just a rough guess", ± 0.5 for any elliptic orbit or revolution. Many Lambert algorithms use the Newton method and robustness is compromised. These Lambert algorithms break down if the Newton method fails and usually the initial guess is poor. If n is allowed to vary in the Laguerre method, then the chance of getting a converged solution increases dramatically. The value of n, which is arbitrary, is initially set to 2 in the **lambert2** algorithm**.**

In addition, the physics of the Lambert problem and the specific formulation of a Lambert algorithm dictate the limits of the independent variable. The Sun Formulation requires the independent variable x be defined as $|x| < 1$ for elliptic orbits for all N, and $x = 1$ for parabolic orbits, and $x > 1$ for hyperbolic orbits when N = 0. When x is bounded within limits, the estimate of the initial guess is simple. For example, if the given position vectors requires that $-1 < x < 0$, then the initial rough guess can be chosen as $x_1 = -0.5$. The Laguerre method, which is very forgiving, has not yet failed to converge to correct solutions. If this "rough guess" for x is satisfactory, then the **lambert2** algorithm can be used to solve multi-revolution Lambert problems with N = 0, 1, 2, ….

The general formulation of the Lambert problem as presented above is simple. Other general formulations of solving two equations and two unknowns, which are complicated, will not be discussed. The Primer Vector approach (Reference 11) and Series Reversion/Inversion method (Reference 12) are also not recommended and discussed later in the Conclusions. For completeness, Sun's expressions for $F(x)$, $F'(x)$ and $F''(x)$ are given in Appendix A.

The Sun Theory for the Multi-Revolution Lambert Problem

This section may be skipped for the readers not interested in the details of the Sun theory. In the following presentation, only elliptic orbits of multi-revolution will be described and Sun's notations of Reference 5 are used unless otherwise specified. Lancaster and Gooding of References 6 and 7 presented almost the same theory with different notations and iterative methods. When two position vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$ are given at the respective times t_1 and t_2, the central angle or transfer angle, ψ, between $\mathbf{r}_1$ and $\mathbf{r}_2$ can be identified by a value of ψ less than, equal to or greater than 180 degrees. Let $\psi_0 = \cos^{-1}[\frac{\mathbf{r}_1 \bullet \mathbf{r}_2}{r_1 r_2}] < \pi$.

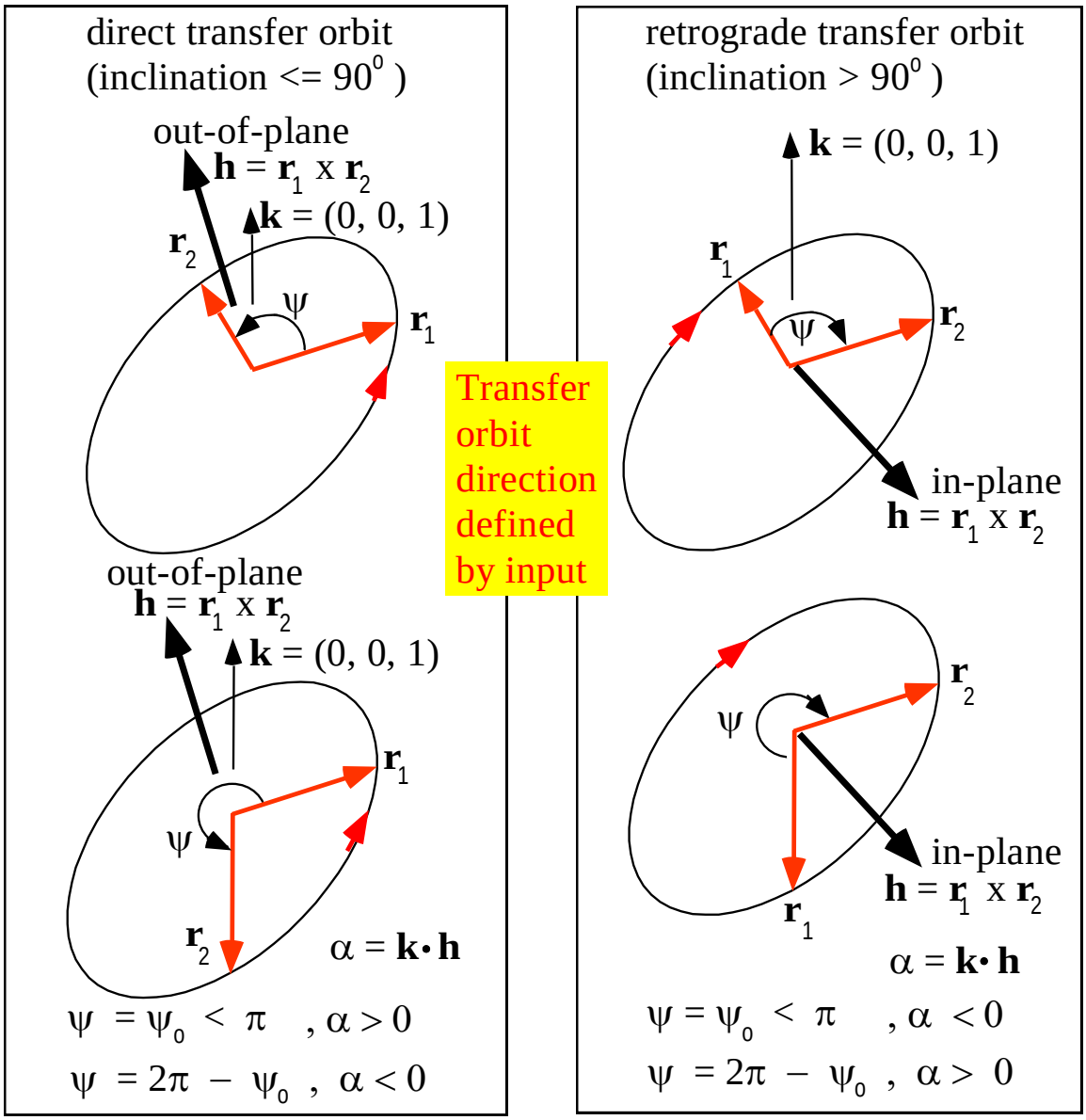


Figure 2. Determination of the transfer orbit direction and angle from the given $\mathbf{r}_1$ and $\mathbf{r}_2$

The special case of the transfer angle, $\psi = k\pi$, and k is zero or an integer will be discussed in the preliminary step of the Computational Procedure. The solutions of the non-degenerate case of $\psi \neq k\pi$ are illustrated in Figure 2. Before a Lambert algorithm is started, the desired transfer orbit to be determined must be specified to be direct (inclination <= 90 degrees) or retrograde (180 > inclination > 90 degrees). In addition to the given $\mathbf{r}_1$, $\mathbf{r}_2$, t_1 and t_2, this input parameter eliminates all the confusions around the output solution of a Lambert problem. In **lambert2**, this user-specified parameter is direct_torbit set to 1.0 or –1.0, which controls the output transfer orbit to be either direct or retrograde.

A second parameter, α, which is computed from the given $\mathbf{r}_1$ and $\mathbf{r}_2$, controls or dictates the transfer angle, ψ, to be less than or greater than 180 degrees. It is determined from the third component of the angular momentum of the transfer orbit as:

$$\alpha = \mathbf{k} \bullet \mathbf{h} = (0,0,1) \bullet (\mathbf{r}_1 \times \mathbf{r}_2)$$

This proper starting procedure allows the transfer angle defined in <u>both</u> $0 < \psi < \pi$ and $\pi < \psi < 2\pi$ for either a direct or retrograde transfer orbit similar to the method of Reference 14. Given $\mathbf{r}_1$, $\mathbf{r}_2$, t_1, t_2 and direct_torbit, one of the four transfer orbit solutions as shown in Figure 2 is a solution of the Lambert problem. The "short and long" transfer orbits as described in many textbooks are misleading or incomplete, and a user may compute a "solution", without noticing that the transfer orbit has an unexpected inclination!

Introducing an Angle parameter, σ, which is related to ψ by

$$\sigma^2 = \frac{4 r_1 r_2}{m^2} \cos^2 \frac{\psi}{2} \tag{6}$$

where r_1 and r_2 are the respective magnitudes of the given position vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$, and $m = r_1 + r_2 + c$, $n = r_1 + r_2 - c$, $c = |\mathbf{r}_1 - \mathbf{r}_2|$. The Angle parameter σ, which varies with the transfer angle ψ, and its sign determined from the given position vectors, α and direct_torbit.

The independent variable or unknown iterative parameter, x, is called the Path parameter. It is related to the semi-major axis, a, and can be expressed as

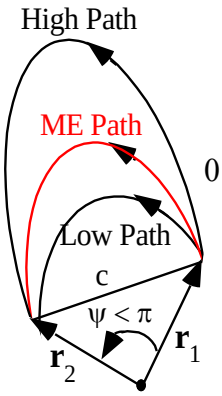
$$x^2 = 1 - \frac{m}{4a}, \qquad y^2 = 1 - \frac{n}{4a} \tag{7}$$

The Angle parameter and Path parameter are related to each other by

$$\sigma^2 = \frac{1 - y^2}{1 - x^2}, \qquad y = \pm\sqrt{[1 - \sigma^2(1 - x^2)]} \tag{8}$$

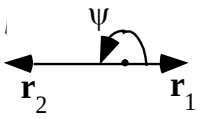
y is a function of σ and x. The sign of y is chosen as that of σ if $\sigma^2 \neq 0$, and | y | = 1 if $\sigma^2 = 0$.

The physical meaning of the Angle parameter and Path parameter and their regions of validity are depicted in Figures 3 and 4. Though the case of $\mathbf{r}_1 = \mathbf{r}_2$, (when ψ = 0 or 360 degrees,) has no practical value, it presents the small-angle difficulties for Initial Orbit Determination where most Lambert algorithms break down or converge slowly. The other situation of difficulty is in Figure 3, Case (b), when ψ = 180 degrees, where some Lambert algorithms break down in this close proximity. Case (a) and Case (c) of Figure 3 are solvable, as long as the transfer angle ψ is not near the singularities of 0, 180 and 360 degrees. The more robust the Lambert algorithm, the closer to the singularities it can approach. For practical long range ballistic missile targeting, ψ is usually far away from any singularities, and therefore almost any Lambert algorithm can work flawlessly. Most Lambert algorithms work only for non-multi-revolution problems. A practical missile-targeting trajectory is usually less than one revolution, so there is only one possible Lambert solution. Starting procedure and singularities will be discussed further in the Computational Procedure.



Case (a)

$0 < \psi < \pi \longleftrightarrow 0 < \sigma < 1$

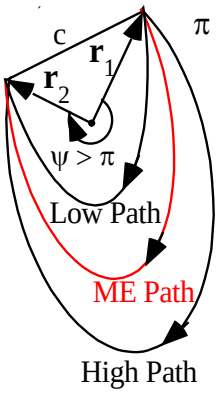


Case (b)

$\psi = \pi \longleftrightarrow \sigma = 0$

Case (c)

$\pi < \psi < 2\pi \longleftrightarrow -1 < \sigma < 0$



Elliptic Orbits
- Low Path, $0 < x < 1$
- ME Path= Minimum Energy Path, $x = 0$
- High Path, $-1 < x < 0$

Parabolic Orbits $x = 1$

Hyperbolic Orbits $x > 1$

Figure 3. Definitions of the transfer angle ψ to the Angle parameter σ and the Path parameter x to orbit types

It should be noted that the initial guess for the unknown Path parameter, $|x_1| < 1$ for elliptic orbits, is required to start the iterative formula of equation (5), and the converged solution of x must satisfy the Lambert equation (4). In other words, we need to find the value of the Path parameter x that makes the Lambert equation (4), $F(x) = 0$. For any number of revolutions, N, Case (a) and Case (c) of Figure 3 also illustrate that the Low Path ($x > 0$) and the High Path ($x < 0$) are separated by the Minimum Energy (ME) Path ($x = 0$) for any ψ. The determination of the Minimum Energy transfer time, t_{ME}, is simple and is shown later in the Computational Procedure section. For a given transfer time, t, between the two given position vectors, it is assumed that $t = t_2$ and $t_1 = 0$, without loss of generality. The value of $(t - t_{ME})$ indicates whether x is positive (low Path) or negative (high Path). That is:

$t < t_{ME}$ for $x > 0$ (Low Path)

$t = t_{ME}$ for $x = 0$ (Minimum Energy Path)

$t > t_{ME}$ for $x < 0$ (High Path)

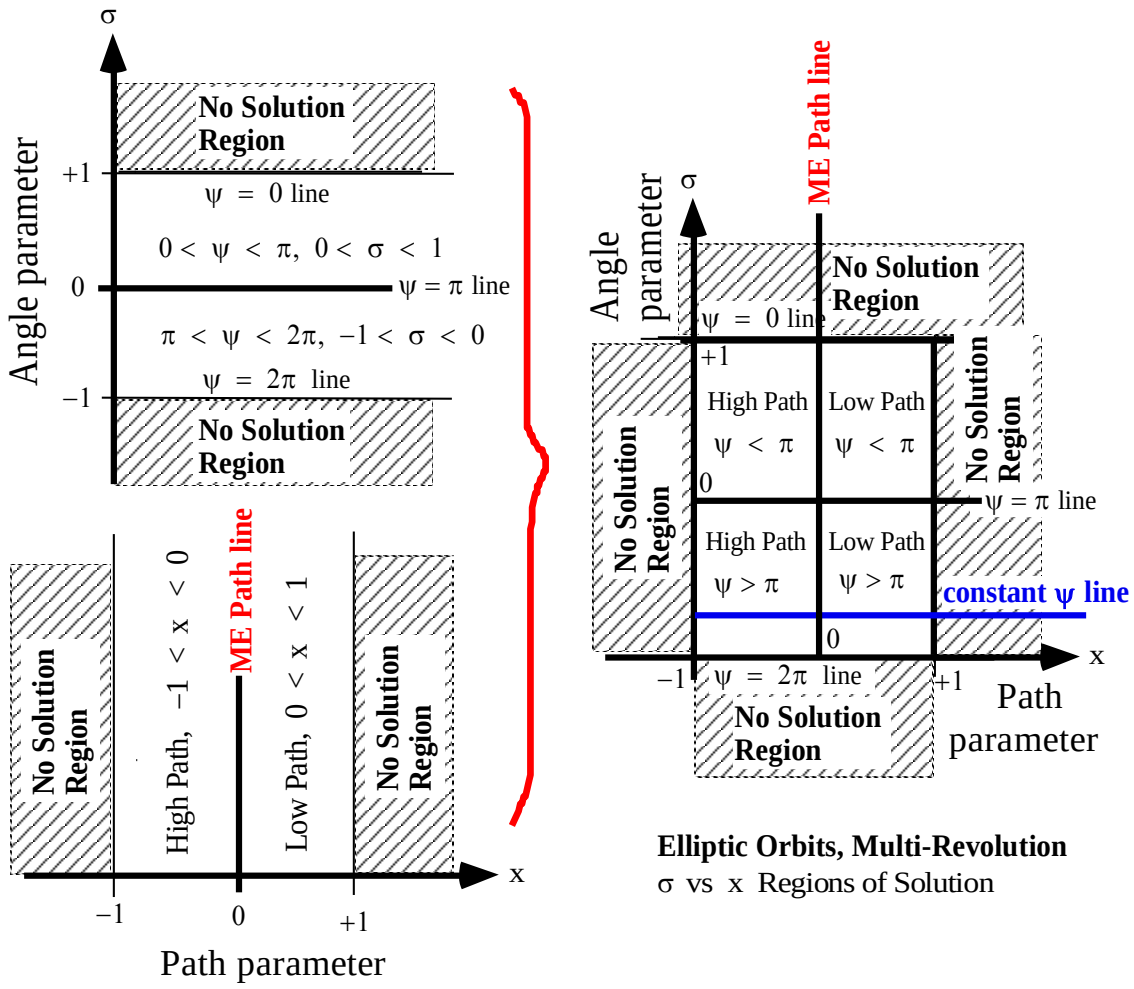


Figure 4. Elliptic orbits, multi-revolution and σ vs x regions of solution.

For elliptic orbits of multi-revolution, $N \geq 0$, the solution regions can be divided into four parts as shown in Figure 4. When the two position vectors are given, σ is unique and can be computed. A constant ψ line, which is straight, can then be drawn as shown in Figure 4 (the blue line, for example). The remaining problem is to determine the value of x in the Low or High Path region, which will intersect the constant ψ line, if a solution exists. Equation (6) suggests that a constant σ line is parabolic while a constant ψ line is straight. The obvious question, then is: why ask for trouble by working with the more difficult parabolic constant σ line, rather than the simple, straight constant ψ line instead?

For N = 0 and σ determined from given initial conditions, the Minimum Energy transfer time, t_{ME}, which indicates the sign of x, is the only parameter needed to initiate the initial guess of x. If x is positive on the Low Path, a simple guess of $x_1 = 0.5$ is good enough to kick start equation (5), and a unique Lambert solution will be computed in a few iterations (normally between 3 to 7) by **lambert2**. If x is negative on the High Path, a simple guess is $x_1 = -0.5$. A unique solution is guaranteed, because the given time t is a single-valued and monotonic function of x for N = 0.

The Minimum Energy time t_{ME} is easily determined with x = 0. When t_{ME} is determined for any N, then x is positive or negative according to t less than or greater than t_{ME}.

For multi-revolution, $N \geq 1$, a single minimum exists for a given σ and t, which is the minimum time of flight t_{MT}. The minimum time of flight t_{MT} can be computed as shown later in the Computational Procedure section by a simple iterative method. When N is specified such that $N \geq 1$, then there will be two distinct solutions, a unique solution, or no solution according to t greater than, equal to, or less than t_{MT}.

In summary, t_{ME}, which indicates the sign of x, is needed for all N. t_{MT}, which indicates the number of possible solutions, is computed only for $N \geq 1$. In any case, both t_{ME} and t_{MT} can be easily determined from the given initial conditions of a Lambert problem.

Figures 5 and 6 depict the solution regions in the τ vs x plots for N = 0 and N = 1 and 2. The normalized time, τ, in the Lambert equation (4), can be computed from

$$\tau = 4t\sqrt{\frac{\mu}{m^3}} \tag{9}$$

where $t = t_2$, $t_1 = 0$, $m = r_1 + r_2 + c$, $c = |\mathbf{r}_1 - \mathbf{r}_2|$, and μ is the gravitational constant. The region of solutions for parabolic and hyperbolic orbits of N = 0 are illustrated in Figure 5. The numerical values of Sun's example of Reference 5, which are reproduced in Example 1 of the Numerical Examples section, are also displayed in Figure 6. The input position vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$, are not given in Reference 5. These position vectors are reconstructed, resulting in a small difference as those in Reference 5. Comparison of the given value of τ, the computed values of t_{ME} and t_{MT} in Figure 6, the possible solutions of multi-revolution Lambert problem for N = 1 can be determined. Similarly, for all N greater than one, the concept of determining the possible number of solutions is the same.

Note that in the **lambert2** computer program the normalized and non-dimensional time τ defined by equation (9) is used throughout for the feel of the magnitudes of time. The input time unit is in seconds with $t = t_2$ and $t_1 = 0$, without loss of generality. However, for illustrations in Figures 5 and 6, it is further divided by π to make the time scale more readable, and with the similar scaling factor with the Path parameter, x.

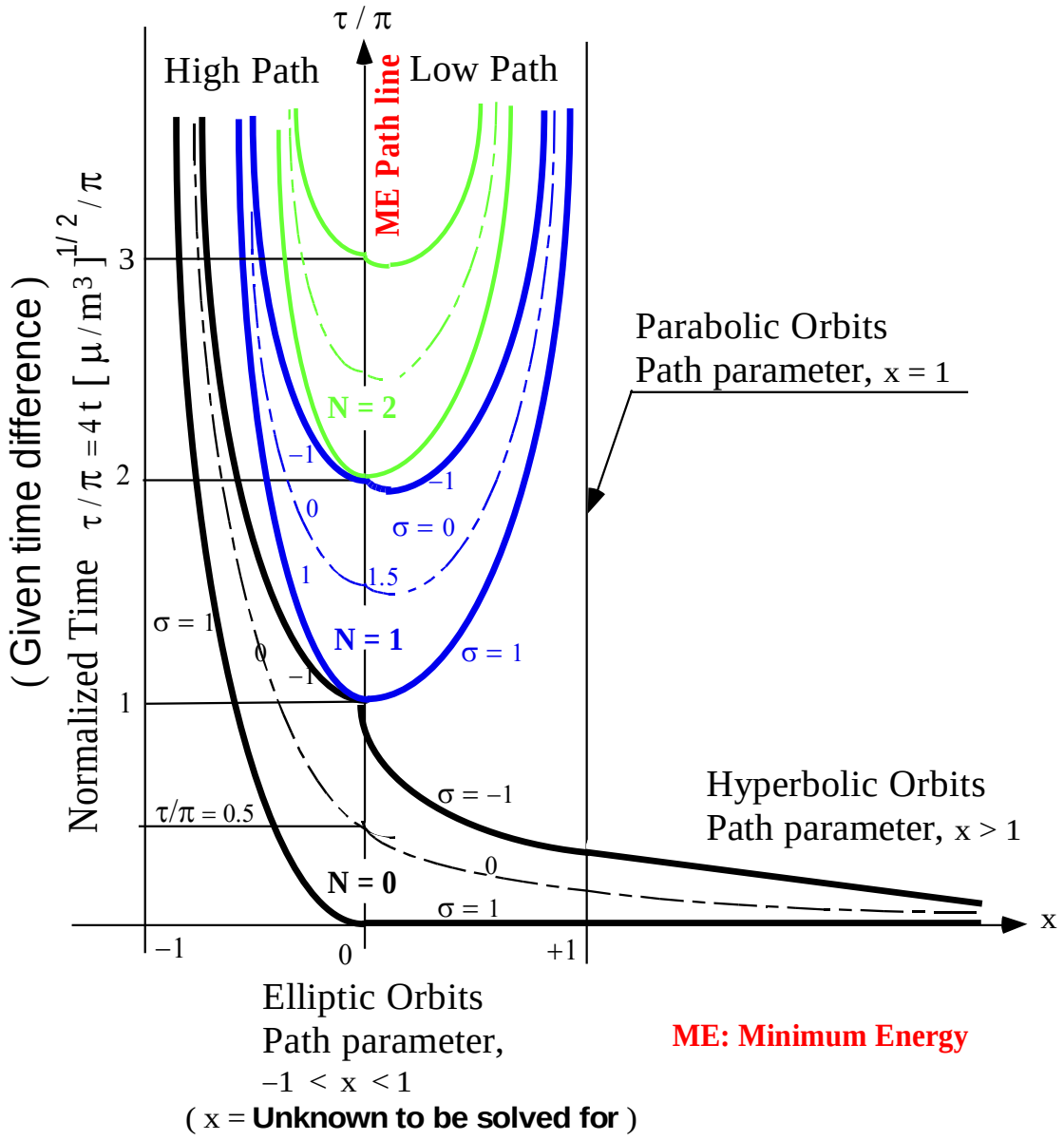


Figure 5. Multi-Revolution and τ vs x Regions of Solution

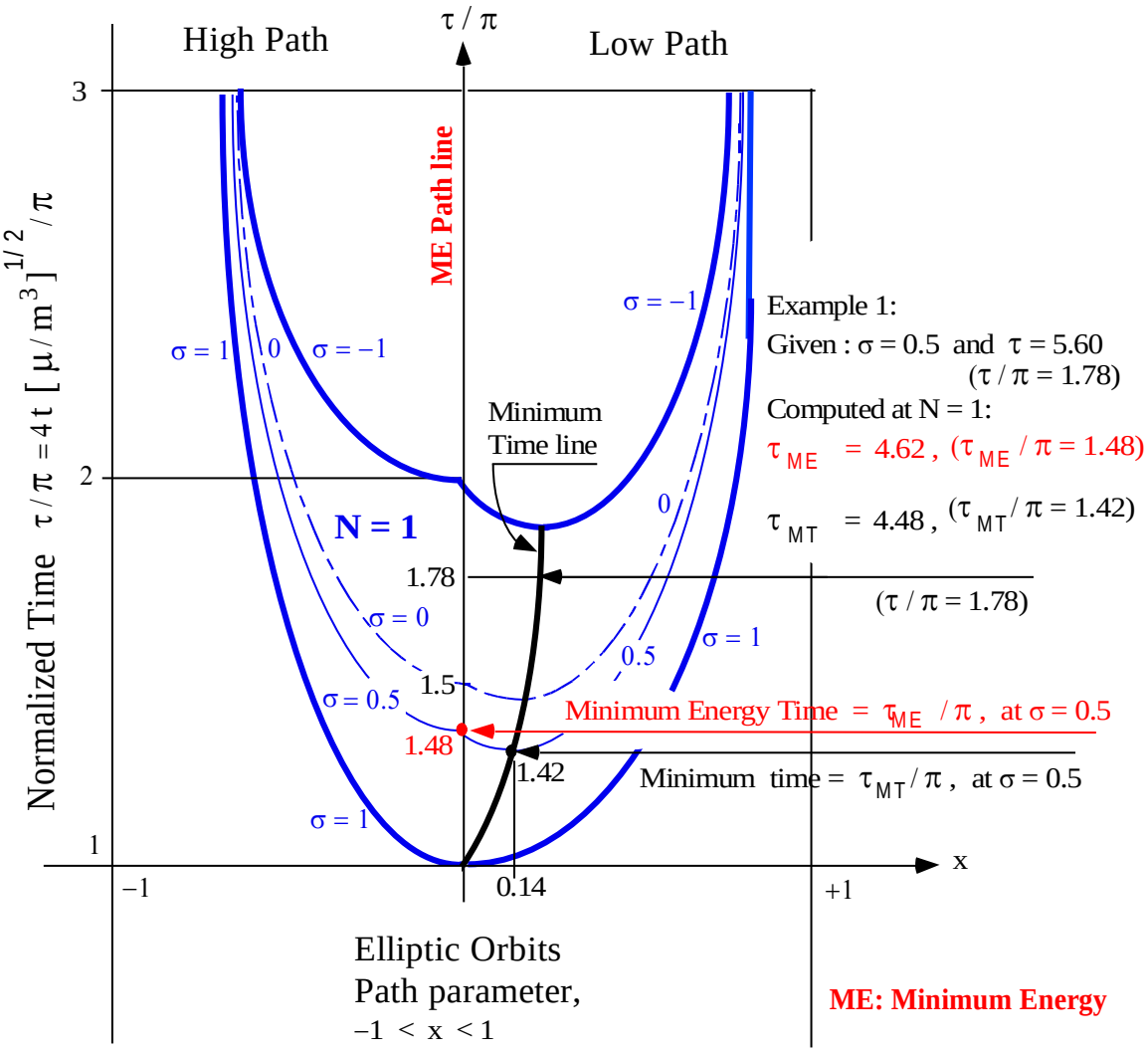


Figure 6. Typical Multi-Revolution and τ vs x Regions of Solution
(Elliptic Orbits Only, N = 1 with Example 1 values)

Computational Procedure

Conceptually, all Lambert formulations are similar to those described in the section General Formulation of the Lambert Problem. Sun's choice of the unknown variable, x, not only allows for simple expressions of the Lambert equation (4) and its derivatives, $F(x)$, $F'(x)$ and $F''(x)$, but computable equations without ambiguity, and is applicable to multi-revolution elliptic orbits that few can match. For elliptic orbits, $|x| < 1$ must be strictly enforced. Parabolic and hyperbolic orbits exist only for less than one revolution (N = 0) and $x \geq 1$. These simple equations are included in Appendix A (reproduced from Reference 5) and implemented in the source code. Since only elliptic orbits exist for multi-revolutions, a physically understandable and sensibly bounded unknown parameter presents great advantage over those formulations using other universal variables. Notice that Gooding of Reference 7 defined this same unknown variable x as the unknown universal variable.

Furthermore the unknown (Path) parameter x is related to or identified by the physical Low and High Paths separated by the Minimum Energy trajectory. Sun's formulation for multi-revolution elliptic transfer orbits can be reduced to the following steps:

0. Preliminary step: Before a Lambert algorithm is started, the transfer orbit to be determined must be specified to be direct (inclination <= 90 degrees) or retrograde (inclination > 90 degrees). This is a hard-coded line in **lambert2** instead of an input parameter in order to keep the normal input parameters of two position vectors and two times. That is:
   direct_torbit = 1.0 for direct transfer orbit with inclination <= 90 degrees
   direct_torbit = -1.0 for retrograde transfer orbit with inclination > 90 degrees

   A second parameter, α, which controls or dictates the transfer angle to be less than or greater than 180 degrees, is determined from the third component of the angular momentum of the transfer orbit. That is:

$$\alpha = (0,0,1) \bullet (\mathbf{r}_1 \times \mathbf{r}_2)$$

   This proper starting step allows the transfer angle defined in <u>both</u> $0 < \psi < \pi$ and $\pi < \psi < 2\pi$ for either a direct or retrograde transfer orbit. Figure 2 shows four transfer orbits, not the two transfer orbits, "short and long", as described in many textbooks. Again, the two solutions of "short and long" transfer orbits are misleading or incomplete, and a user may unintentionally compute a transfer orbit of an unexpected inclination!

   When the transfer angle, $\psi = k\pi$, and k is even or zero, transfer orbits are degenerate or physically meaningless. They will not be treated. If k is odd, the transfer orbits are non-degenerate and real, but there are infinite numbers. For this "singular" case, the inclination of the transfer orbit must be specified first. The special case that the inclination is zero, the transfer orbit is that of the Hohmann. For this reason, **lambert2** only provides multi-revolution Lambert solutions for real transfer orbits for all $\psi \neq k\pi$. The simple Hohmann-type transfers, which are treated in many text books, are left as an exercise for the readers.

1. Compute the transfer angle $\psi_0 = \cos^{-1}[\frac{\mathbf{r}_1 \bullet \mathbf{r}_2}{r_1 r_2}]$ between the two given position vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$, and choose $0 < \psi = \psi_0 < \pi$.

   For direct transfer orbit: $\psi = 2\pi - \psi_0$, if $\alpha < 0$

   For retrograde transfer orbit: $\psi = 2\pi - \psi_0$, if $\alpha > 0$

   Once ψ is determined after the parameters direct_torbit and α, the positive or negative value of the Angle parameter, σ, computed from equation (6) and its sign is determined from ψ as:

   (a) $\sigma > 0$ for $0 < \psi < \pi$ (Case a, Fig. 1)
   (b) $\sigma = 0$ for $\psi = \pi$ (Case b, Fig. 1)
   (c) $\sigma < 0$ for $\pi < \psi < 2\pi$ (Case c, Fig. 1)

2. Compute the normalized time τ using equation (9), with $t = t_2 = (t_2 - t_1)$, $t_1 = 0$, $m = r_1 + r_2 + c$, $c = |\mathbf{r}_1 - \mathbf{r}_2|$.

   $$\tau = 4t\sqrt{\frac{\mu}{m^3}}$$

3. Compute the parabolic normalized time τ_p from the parabolic orbit equation (A3) as:

   $$\tau_p = \frac{2}{3}(1 - \sigma^3) \qquad \text{(A3)}$$

4. Determine orbit type:

   If $(\tau > \tau_p) \quad \Rightarrow$ elliptic, otherwise parabola or hyperbolic

5. In the following, only elliptic orbits are considered (parabolic and hyperbolic orbits are computed in **lambert2**). Compute the maximum number of possible revolutions:

   $N_{max} = \text{Integer}(\tau/\pi)$

   if $N_{max} \geq 1$, then start a do-loop with $\text{nrev} = 0,\ N_{max}$.

6. This step is needed only for multi-revolutions, $N = \text{nrev} \geq 1$. Compute the normalized minimum time of flight τ_{MT}. From Equation (A5), by setting $F'(x) = 0$ gives

   $$\tau_{MT} = \frac{2}{3}(\frac{1}{x_{MT}} - \frac{\sigma^3}{|y_{MT}|}) \qquad (10)$$

   where τ_{MT} is the solution of the minimum time equation and its first derivative:

   $$\Phi = \phi(x) - \phi(y) + N\pi = 0 \qquad (11)$$

   $$\frac{d\Phi}{dx} = \frac{2}{3}\frac{(1 - x^2)^{3/2}}{x^2}(1 - \frac{\sigma^5 x^3}{|y|^3}) \qquad (12)$$

where

$$\phi(u) = \cot^{-1}\left[\frac{u}{\sqrt{(1-u^2)}}\right] - \frac{1}{3u}(2+u^2)\sqrt{(1-u^2)}, \qquad (u = x \text{ or } y) \qquad (13)$$

and the arc-cotangent function is subjected to the restriction of equation (A4). Since the normalized minimum time of flight τ_{MT} is almost known with respect to x as shown in Figure 6 for N = 1 (similar for $N > 1$), only the Newton method is used instead of the Laguerre method. If the Laguerre method is desired, equation (12) can be differentiated with respect to x to obtain the second derivative of Φ. The solution of equation (11) are x_{MT} and y_{MT}, which in turn give the normalized minimum time of flight τ_{MT} of equation (10). Note that $y_{MT} = \pm\sqrt{[1 - \sigma^2(1 - x_{MT}^2)]}$. The sign of y_{MT} is chosen as that of σ if $\sigma^2 \neq 0$, and $|y_{MT}| = 1$ if $\sigma^2 = 0$. If an unbounded independent variable is used as in References 9 and 11, then at certain transfer angle ψ the locus of τ is flat for a wide range of x making x_{MT} and τ_{MT} difficult to obtain.

7. Compute the normalized Minimum Energy time τ_{ME} from the Minimum Energy equation for all $N = nrev \geq 0$:

$$\tau_{ME} = N\pi + \cos^{-1}(\sigma) + \sigma\sqrt{[1 - \sigma^2]} \qquad (14)$$

8. Since τ and τ_{ME} are determined from Steps 2 and 7 respectively, then $(\tau - \tau_{ME})$ indicates whether x is positive (Low Path) or negative (High Path). That is:

   $\tau < \tau_{ME}$ for $x > 0$ (Low Path), giving $x_1 = 0.5$

   $\tau = \tau_{ME}$ for $x = 0$ (Minimum Energy Path)

   $\tau > \tau_{ME}$ for $x < 0$ (High Path), giving $x_1 = -0.5$

   Using this simple initial value of x_1, equation (5) can be started, and a reasonable Lambert solution for $N = 0$ can be obtained. If the initial guess of $|x_1| \geq 1$ for an elliptic orbit, then x_1 should be reset to either 0.5 or -0.5 depending on the Low or High Path. The simple estimate of Reference 15, $x_1 = \frac{1}{2}\tau_{ME}\left(\frac{\tau_{ME}}{\tau} - 1\right)$, which is unnecessary, is included as a test of efficiency and robustness. The complicated starting formulae developed by Gooding of Reference 7 are eliminated. The reader should know the initial values of x for all $N = nrev \geq 1$.

9. Determine the output velocities at t_1 and t_2 from the converged x and y:

$$\begin{aligned} \mathbf{v}_1 &= v_c\,\mathbf{e}_c + v_r\,\mathbf{e}_{r1} \\ \mathbf{v}_2 &= v_c\,\mathbf{e}_c - v_r\,\mathbf{e}_{r2} \end{aligned} \qquad (15)$$

where $v_c = \sqrt{\mu}\left(y/\sqrt{n} + x/\sqrt{m}\right)$, $v_r = \sqrt{\mu}\left(y/\sqrt{n} - x/\sqrt{m}\right)$, $\mathbf{e}_c = (\mathbf{r}_2 - \mathbf{r}_1)/c$, $\mathbf{e}_{r1} = \mathbf{r}_1/r_1$, $\mathbf{e}_{r2} = \mathbf{r}_2/r_2$, $m = r_1 + r_2 + c$, $n = r_1 + r_2 - c$, and $c = |\mathbf{r}_1 - \mathbf{r}_2|$.

Lambert Numerical Examples

Example 1: Sun's Example in Reference 5 with estimated position vectors at times t_1 and t_2, and their respective magnitudes are 30072 km and 9990 km. They are reconstructed from a Molniya Orbit, since $\mathbf{r}_1$ and $\mathbf{r}_2$ are not given in Reference 5. Table 1 shows the summary of key parameters, and Figure 7 depicts the trajectories with $\sigma = \pm 0.5$. These are only the two solutions in Sun paper of Reference 5. Notice that the two trajectories of N = 0 for $\sigma = \pm 0.5$ are lofted high so that more transfer time can be "wasted." For N = 1, all the High Path and Low Path trajectories have to travel more than one orbit before reaching position vector $\mathbf{r}_2$ at t_2, and therefore they are close to the Minimum Energy trajectories.

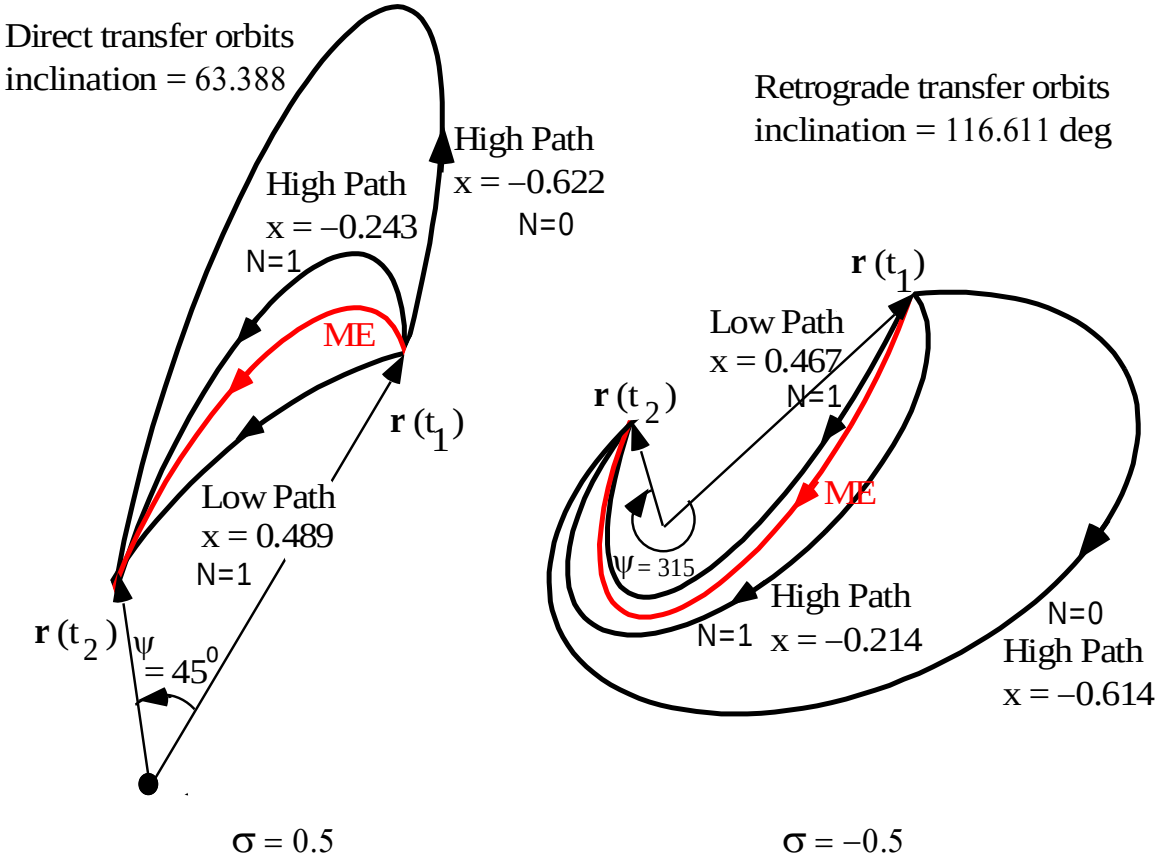


Figure 7. Multi-revolution solutions of Example 1 (Elliptic orbits only, N = 0 and 1)

Table 1 shows the summary of key parameters of Example 1 that are Sun's solutions in Reference 5. Table 1x shows the summary of key parameters of the "other two solutions" that are missing in most textbooks.

Given: $t = t_2 = 36{,}000$ seconds (10 hours), and $t_1 = 0$

ECI position vectors:

$$\mathbf{r}(t_1) = \begin{bmatrix} 22592.1456\ 03 \\ -1599.91523\ 9 \\ -19783.9505\ 06 \end{bmatrix}, \qquad \mathbf{r}(t_2) = \begin{bmatrix} 1922.06769\ 7 \\ 4054.15705\ 1 \\ -8925.72746\ 5 \end{bmatrix}$$

Input: direct_torbit = 1.0 in **lambert2**, to force a direct transfer orbit resulting in the inclination of 63.38801958 degrees

Computed: Transfer Angle: $\psi = 44.7$ (deg), Angle parameter: $\sigma = 0.5$

N = 0 (less than one revolution, one solution)

ECI velocity vectors: (only solution of N = 0, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 2.000652697 \\ 0.387688615 \\ -2.666947760 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -3.79246619 \\ -1.77707641 \\ 6.856814395 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)

ECI velocity vectors: (first solution of N = 1, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 0.50335770 \\ 0.61869408 \\ -1.57176904 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -4.18334626 \\ -1.13262727 \\ 6.13307091 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)

ECI velocity vectors: (second solution of N = 1, Path parameter x > 0, low)

$$\mathbf{v}(t_1) = \begin{bmatrix} -2.45759553 \\ 1.16945801 \\ 0.43161258 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -5.53841370 \\ 0.01822220 \\ 5.49641054 \end{bmatrix}$$

Input: direct_torbit = –1.0 in **lambert2**, to force a retrograde transfer orbit resulting in the inclination of 116.61198041 degrees

Computed: Transfer Angle: $\psi = 315.3$ (deg), Angle parameter: $\sigma = -0.5$

N = 0 (less than one revolution, one solution)

ECI velocity vectors: (only solution of N = 0, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 2.96616042 \\ -1.27577231 \\ -0.75545632 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} 5.84375455 \\ -0.20047673 \\ -5.48615883 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)

ECI velocity vectors: (first solution of N = 1, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 1.33645655 \\ -0.94654565 \\ 0.30211211 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} 4.93628678 \\ 0.39863416 \\ -5.61593092 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)
ECI velocity vectors: (second solution of N = 1, Path parameter x > 0, low)

$$\mathbf{v}(t_1) = \begin{bmatrix} -1.38861608 \\ -0.47836611 \\ 2.21280154 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} 3.92901545 \\ 1.50871943 \\ -6.52926969 \end{bmatrix}$$

	direct_torbit = 1.0; i = 63.4 (deg)			direct_torbit = –1.0; i =116.6 (deg)		
	Transfer Angle: ψ = 44.705 (deg)			Transfer Angle: ψ = 315.295 (deg)		
	Angle parameter: σ = 0.5			Angle parameter: σ = – 0.5		
	N = 0	N = 1	N = 1	N = 0	N = 1	N = 1
Path p.: x_1	– 0.5	– 0.31116	0.44791	– 0.5	– 0.30678	0.44808
Converged x	– 0.62233	– 0.24362	0.48960	– 0.61358	– 0.21437	0.46690
Converged y	0.92014	0.87440	0.89984	– 0.91867	– 0.87248	– 0.89682
ME time: τ_{ME}	1.48005	4.62164	4.62164	1.66154	4.80314	4.80314
Min time: τ_{MT}	---	4.47610	4.47610	---	4.65723	4.65723
v1mag	3.35642	1.76256	2.75567	3.31608	1.66533	2.65586
v2mag	8.03472	7.50985	7.80288	8.01800	7.48762	7.76819
Kind of Path	high	high	low	high	high	low

Table 1. Summary of Sun's solutions of Example 1

	direct_torbit = 1.0; i = 63.4 (deg)			direct_torbit = –1.0; i =116.6 (deg)		
	Transfer Angle: ψ = 315.295 (deg)			Transfer Angle: ψ = 44.705 (deg)		
	Angle parameter: σ = – 0.40815			Angle parameter: σ = 0.40815		
	N = 0	N = 1	N = 1	N = 0	N = 1	N = 1
Path p.: x_1	– 0.5	– 0.31919	0.44802	– 0.5	– 0.32120	0.44796
Converged x	– 0.63837	– 0.29136	0.52668	– 0.64240	– 0.30304	0.53546
Converged y	– 0.94936	– 0.92062	– 0.93788	0.94982	0.92125	0.93870
ME time: τ_{ME}	1.61861	4.76021	4.76021	1.52297	4.66456	4.66456
Min time: τ_{MT}	---	4.61442	4. 61442	---	4.51890	4.51890
v1mag	3.37006	1.76255	2.83363	3.38958	1.81222	2.87541
v2mag	10.56047	10.16233	10.40172	10.56672	10.17106	10.41318
Kind of Path	high	high	low	high	high	low

Table 1x. Summary of the "other two solutions" of Example 1 missing in most textbooks

Example 2: This Example is to test how close the **lambert2** algorithm can get near the singularity point of zero degree transfer angle. The given position vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$ are constructed from a Low Earth Orbit (LEO) with the transfer angle, ψ, at 0.32335 degrees. The Angle parameter is $\sigma = +0.99092$. Since τ/π is 5.53 indicating five revolutions, and therefore there are 11 possible solutions for the constant σ. The first solution is depicted in Figure 8, and is in the solution region of the High Path. The initial guess of x was far from the correct solution, but the robustness of the Laguerre's method ensured convergence. Similar to Example 1, some of the key parameters of the first five trajectories with N = 0, 1 and 2 are shown in Table 2. The High Path ones are, as expected, lofted extremely high (as in Figure 8 for N = 0) to account for the long transfer times and very short distance between the two position vectors.

If the transfer time is short, this small angle problem is reduced to those of the Initial Orbit Determination (IOD) using radar data. The $N \geq 1$ solutions computed by the **lambert2** algorithm will be close to the Minimum Energy one in Figure 8. In general, the IOD solutions are those depicted in Case (a) and Case (c) of Figure 3.

It is interesting to note that the initial estimate of the Path parameter, x_1, from Reference (15) may be less than –1. However, for N = 0, x_1 is set to 0.5 or – 0.5, and the iterative equation (5) of the Laguerre method converged rapidly with almost any reasonable first guess within ± 1. The maximum number of iterations for thousands of test cases was seldom greater than seven, and the solutions are reasonable. For the test case of $\psi = 0.07$ degrees, the number of iterations was 12. With the difficult test of Example 2 and the thousands of test cases, the **lambert2** algorithm is robust. Many other Lambert algorithms tested by the author failed to converge even at N = 0 using this Example.

Given: $t = t_2 = 12300$ seconds and $t_1 = 0$

ECI position vectors:

$$\mathbf{r}(t_1) = \begin{bmatrix} 7231.58074563487 \\ 218.02523761425 \\ 11.79251215952 \end{bmatrix}, \qquad \mathbf{r}(t_2) = \begin{bmatrix} 7357.06485698842 \\ 253.55724281562 \\ 38.81222241557 \end{bmatrix}$$

Input: direct_torbit = 1.0 in **lambert2**, to force a direct transfer orbit resulting in the inclination of 40.19574532 degrees

Computed: Transfer Angle: $\psi = 0.32335$ (deg), Angle parameter: $\sigma = 0.99092$

N = 0 (less than one revolution, one solution)

ECI velocity vectors: (only solution of N = 0, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 8.79257809 \\ 0.27867677 \\ 0.02581527 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -8.68383320 \\ -0.28592643 \\ -0.03453010 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)
ECI velocity vectors: (first solution of N = 1, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 7.63353091 \\ 0.24582764 \\ 0.02569470 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -7.50840227 \\ -0.24335652 \\ -0.02658981 \end{bmatrix}$$

N = 1 (one+ revolution, two solutions)
ECI velocity vectors: (second solution of N = 1, Path parameter x > 0, low)

$$\mathbf{v}(t_1) = \begin{bmatrix} 8.19519089 \\ 2.30595215 \\ 1.75229388 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} 8.07984345 \\ 2.30222567 \\ 1.75189559 \end{bmatrix}$$

N = 2 (two+ revolution, two solutions)
ECI velocity vectors: (first solution of N = 2, Path parameter x < 0, high)

$$\mathbf{v}(t_1) = \begin{bmatrix} 6.51890385 \\ 0.21496104 \\ 0.02618989 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} -6.37230007 \\ -0.20150975 \\ -0.01832295 \end{bmatrix}$$

N = 2 (two+ revolution, two solutions)
ECI velocity vectors: (second solution of N = 2, Path parameter x > 0, low)

$$\mathbf{v}(t_1) = \begin{bmatrix} 7.00660748 \\ 1.96687296 \\ 1.49423471 \end{bmatrix}, \qquad \mathbf{v}(t_2) = \begin{bmatrix} 6.87133644 \\ 1.96250281 \\ 1.49376762 \end{bmatrix}$$

	Transfer Angle: ψ = 0.32335 (deg)			Transfer Angle: ψ = 0.32335 (deg)		
	Angle parameter: σ = 0.99092			Angle parameter: σ = 0.99092		
	N = 0	N = 1	N = 1		N = 2	N = 2
Path p.: x_1	–0.5	–0.41742	0.42047		–0. 27487	0.40561
Converged x	–0.83485	–0.72176	0.82461		–0.61242	0.70139
Converged y	0.83813	0.72773	0.82812		0.62159	0.70790
ME time: τ_{ME}	0.26814	3.40974	3.40974		6.55133	6.55133
Min time: τ_{MT}	---	3.32344	3.32344		6.49126	6.49126
v1mag	8.79703	7.63753	8.69190		6.52250	7.42926
v2mag	8.68860	7.51239	8.58215		6.37551	7.30055
Semi-major axis	12152.14	7686.574	11507.10		5892.482	7247.976
Eccentricity	0.999998	0.999996	0.957687		0.999994	0.950987
Inclination	40.19575	40.19575	40.19575		40.19575	40.19575
Kind of Path	high	high	low		high	low

Table 2. Summary of solutions of Example 2 to challenge any Lambert algorithm

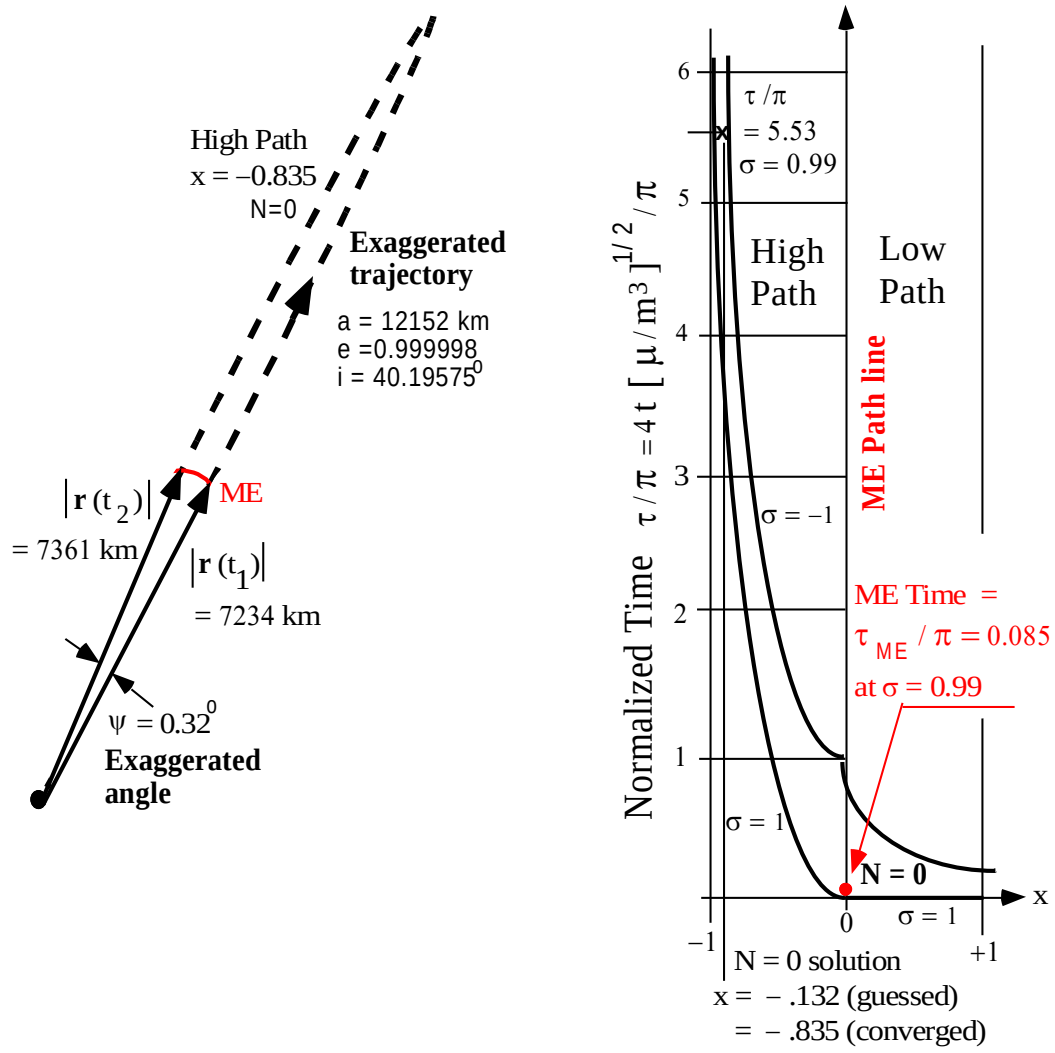


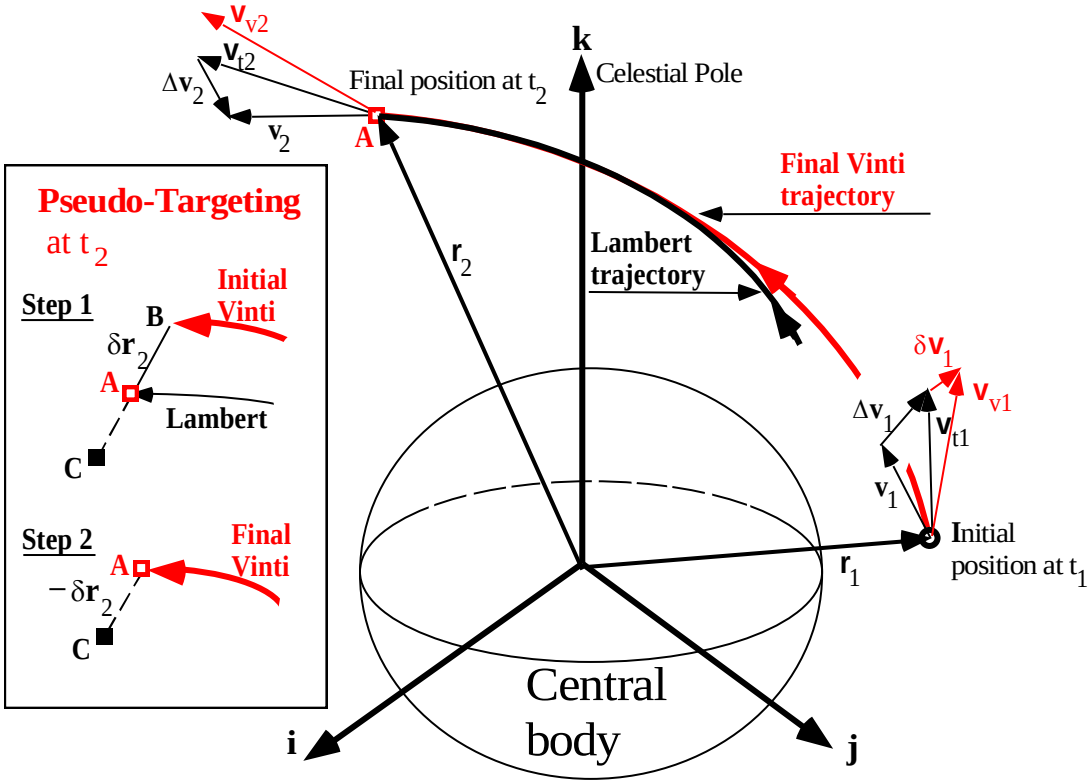
Figure 8. First Solution of Example 2 with very poor initial guess of x to stress the Laguerre's iterative method. (Elliptic Orbit, N = 0)

For $N = nrev \geq 1$, the initial guess of x_1 for the High and Low Paths can be easily estimated with the use of Figures 5 and 6. For the High Path ($x_1 < 0$), $x_1 = -|x|/(nrev+1)$, where x is the last value of x. For the Low Path ($x_1 > 0$), using the value associated with the minimum time x_{MT} (see Figure 6) of nrev gives $x_1 = (x_{MT} + 0.75)/2$.

It is important to understand that the computed multi-revolution Lambert solutions of $N \geq 1$ may not agree with real motion. The user should make sure that the perigee radius is greater than the Earth/central-body radius for real trajectories.

Lambert - Vinti Targeting

The Keplerian solution of the Lambert algorithm can be easily extended to include the non-Keplerian terms (J_2, J_3 and most of J_4) of the Vinti algorithm of Reference 17 via a simple targeting technique. This targeting method, which is commonly used in Applied Optimal Control Theory of Reference 18, is also known as Pseudo-Targeting, Neighboring Trajectory and State Transition methods in other applications.



Lambert / Vinti targeting

Given: $\mathbf{r}_1$, $\mathbf{r}_2$, $\Delta t = t_2 - t_1$, Computed $\mathbf{v}_{t1}$, $\mathbf{v}_{t2}$ **(Lambert)**

Find: $\mathbf{v}_{v1}$, $\mathbf{v}_{v2}$ **(Vinti)**

Figure 9. Theory of Pseudo-Targeting using the Vinti algorithm

Figure 9 illustrates the theory of the Pseudo-Targeting method using the given Lambert inputs, the computed Lambert solution (initial guess for Vinti) and the desired Vinti solution. Conceptually it is desired to hit the target $\mathbf{r}_2$ or point **A** using a Vinti trajectory with transfer orbit velocities $\mathbf{v}_{v1}$ and $\mathbf{v}_{v2}$, instead of a Lambert trajectory that also goes through $\mathbf{r}_1$ and $\mathbf{r}_2$ with transfer orbit velocities $\mathbf{v}_{t1}$ and $\mathbf{v}_{t2}$.

The initial inputs to the Vinti algorithm are the given position vector $\mathbf{r}_1$, the computed Lambert transfer orbit velocity $\mathbf{v}_{t1}$ from the **lambert2** algorithm, and the given times t_1 and t_2. The initial Vinti trajectory, which includes J_2, J_3 and most of J_4, ends at point **B** at t_2 as shown in Step 1 of Figure 9. To correct the position offset, the new target point **C** of Step 2 of Figure 9 is chosen exactly on the opposite side of point **B** with respect to point **A**. Along the way, a 3x3 partial derivative matrix or a state transition matrix is constructed by varying the transfer orbit velocity vector at t_1 and the resulting changes in the position vector at t_2. The small correction due to Vinti targeting results in the change of the transfer orbit velocity by $\delta\mathbf{v}_1$ at t_1, which is the product of the inverse of the state transition matrix and the vector difference of position $\delta\mathbf{r}_2$ at t_2. The desired final Vinti trajectory can be obtained by repeating Steps 1 and 2 of Figure 9. Upon convergence, the Keplerian solution of the Lambert algorithm, $(\mathbf{v}_{t1}, \mathbf{v}_{t2})$, is replaced by the more accurate non-Keplerian solution of the Vinti algorithm, $(\mathbf{v}_{v1}, \mathbf{v}_{v2})$.

The key to convergence and robustness is that the initial guess of the Keplerian transfer orbit velocity $\mathbf{v}_{t1}$ is reasonably close to the non-Keplerian solution of Vinti and the transition matrix is computed by the accurate numerical partials technique of Reference 19. Let the state vector at any time t be denoted as $\mathbf{x} = \begin{bmatrix} \mathbf{r} & \mathbf{v} \end{bmatrix}^T$, where **r** is the position vector and **v** is the velocity vector. The iterative steps are summarized as follows:

1. Compute the Keplerian Lambert solution, $(\mathbf{v}_{t1}, \mathbf{v}_{t2})$ by the **lambert2** algorithm.
2. Compute the nominal Vinti state vector $\mathbf{x}_2^* = \begin{bmatrix} \mathbf{r}_2^* & \mathbf{v}_2^* \end{bmatrix}^T$ at t_2 using the given position vector $\mathbf{r}_1$, the nominal transfer orbit velocity $\mathbf{v}_1^* = \mathbf{v}_{t1}$, and the given times t_1 and t_2 as inputs to the Vinti algorithm.
3. Evaluate the nominal differential correction of the position vector (point **A** – point **B**) at t_2 as
   $$\delta\mathbf{r}_2 = \mathbf{r}_2 - \mathbf{r}_2^*$$
4. Compute the transition matrix, **T**, by accurate numerical partials. Using the Vinti algorithm successively for three times in the neighborhood of the nominal trajectory by perturbing one at a time, each of the three Cartesian components of the nominal transfer orbit velocity vector $\mathbf{v}_1^*$ at t_1 with a step-size of h_i, (i = 1, 2, 3). That is
   $$\mathbf{v}_{1i}^* = \mathbf{v}_1^* + \delta\mathbf{v}_{1i} = \mathbf{v}_1^* + h_i \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$
   The step-size h_i is usually set to 10^{-6} only for the i^{th} Cartesian component. For $j \neq i$, then $h_j = 0$. Using the accurate numerical partials technique, each of the i^{th} neighboring trajectory invokes four more trajectories with the four step-sizes of $h_i/2$, $-h_i/2$, $\rho h_i/2$, $-\rho h_i/2$, where a good choice of ρ is 0.5. Let the corresponding four position

vectors $\mathbf{r}_{2i}$ at t_2 computed by the Vinti algorithm be denoted as $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$ and $\mathbf{y}_4$. The three partial derivatives of the i^{th} column of **T** is given by

$$\frac{\partial \mathbf{r}_{2i}}{\partial \mathbf{v}_{1i}^*} = \frac{(\mathbf{y}_3 - \mathbf{y}_4) - \rho^3(\mathbf{y}_1 - \mathbf{y}_2)}{\rho\, h_i (1-\rho^2)}$$

The 3 x 3 transition matrix can be approximated as

$$\mathbf{T} = \left[\frac{\partial \mathbf{r}_2}{\partial \mathbf{v}_1^*} \right] = = \left[\frac{\partial \mathbf{r}_{21}}{\partial \mathbf{v}_{11}^*} \quad \frac{\partial \mathbf{r}_{22}}{\partial \mathbf{v}_{12}^*} \quad \frac{\partial \mathbf{r}_{23}}{\partial \mathbf{v}_{13}^*} \right]$$

In computing **T**, 12 predictions by the Vinti algorithm are required (four for each i).

5. Update the nominal $\mathbf{v}_1^*$ at t_1 as

$$\mathbf{v}_1^*\big|_{new} = \mathbf{v}_1^*\big|_{old} + \delta \mathbf{v}_1 = \mathbf{v}_1^*\big|_{old} + \mathbf{T}^{-1}\, \delta \mathbf{r}_2$$

Steps 2 to 5 are repeated until the magnitude of $\delta \mathbf{r}_2$ is reduced to an acceptably small value (eg., 10^{-12} km). Note that in Step 2, the new nominal transfer orbit velocity is $\mathbf{v}_1^* = \mathbf{v}_1^*\big|_{new}$ after the first iteration. Normally only three iterations are required. Upon convergence, the desired Vinti velocities of the transfer orbit are $\mathbf{v}_{v1} = \mathbf{v}_1^*$ and $\mathbf{v}_{v2} = \mathbf{v}_2^*$.

The accurate numerical partials technique of computing the state transition matrix **T** requires four times more evaluations than the tradition partial derivative method. However, the rate of convergence is much faster and the choice of the step-size h_i is more forgiving, and therefore improves robustness. The Lambert and Vinti trajectories are both analytic. The non-Keplerian solution of the Lambert-Vinti targeting that includes the gravitational potential terms of J_2, J_3 and most of J_4 can be computed almost instantly. If higher accuracy is desired, then the nominal trajectory and/or the state transition matrix may be computed by numerical integration with all the desired perturbed accelerations. The resulting non-Keplerian velocities of the transfer orbit by Lambert-Numerical targeting will be more accurate than that of the Lambert-Vinti targeting, but at the expense of considerably more CPU times.

Lambert-Vinti Targeting Numerical Examples

Example 1a: Extension of Example 1, N = 1, High Path solution

Given: $t = t_2 = 36{,}000$ seconds (10 hours), and $t_1 = 0$

ECI position vectors:

$$\mathbf{r}(t_1) = \begin{bmatrix} 22592.145603 \\ -1599.915239 \\ -19783.950506 \end{bmatrix}, \qquad \mathbf{r}(t_2) = \begin{bmatrix} 1922.067697 \\ 4054.157051 \\ -8925.727465 \end{bmatrix}$$

Input: direct_torbit = 1.0 in **lambert2**, to force a direct transfer orbit resulting in the inclination of 63.38801958 degrees

Computed: Transfer Angle: $\psi = 44.7$ (deg), Angle parameter: $\sigma = 0.5$

Lambert ECI velocity vectors:

$$\mathbf{v}_{t1} = \mathbf{v}(t_1) = \begin{bmatrix} 0.50335770 \\ 0.61869408 \\ -1.57176904 \end{bmatrix}, \qquad \mathbf{v}_{t2} = \mathbf{v}(t_2) = \begin{bmatrix} -4.18334626 \\ -1.13262727 \\ 6.13307091 \end{bmatrix}$$

Vinti targeting ECI velocity vectors:

$$\mathbf{v}_{v1} = \mathbf{v}(t_1) = \begin{bmatrix} 0.48947268 \\ 0.62699255 \\ -1.57594642 \end{bmatrix}, \qquad \mathbf{v}_{v2} = \mathbf{v}(t_2) = \begin{bmatrix} -4.18527250 \\ -1.05070711 \\ 6.14512956 \end{bmatrix}$$

Example 2a: Extension of Example 2, N = 0, High Path solution

Given: $t = t_2 = 12300$ seconds and $t_1 = 0$

ECI position vectors:

$$\mathbf{r}(t_1) = \begin{bmatrix} 7231.58074563487 \\ 218.02523761425 \\ 11.79251215952 \end{bmatrix}, \qquad \mathbf{r}(t_2) = \begin{bmatrix} 7357.06485698842 \\ 253.55724281562 \\ 38.81222241557 \end{bmatrix}$$

Input: direct_torbit = 1.0 in **lambert2**, to force a direct transfer orbit resulting in the inclination of 40.19574532 degrees

Computed: Transfer Angle: $\psi = 0.32335$ (deg), Angle parameter: $\sigma = 0.99092$

Lambert ECI velocity vectors:

$$\mathbf{v}_{t1} = \mathbf{v}(t_1) = \begin{bmatrix} 8.7925780946 \\ 0.2786767564 \\ 0.0258152736 \end{bmatrix}, \qquad \mathbf{v}_{t2} = \mathbf{v}(t_2) = \begin{bmatrix} -8.6838331963 \\ -0.2859264266 \\ -0.0345301039 \end{bmatrix}$$

Vinti targeting ECI velocity vectors:

$$\mathbf{v}_{v1} = \mathbf{v}(t_1) = \begin{bmatrix} 8.7925788197 \\ 0.2786767791 \\ 0.0258152755 \end{bmatrix}, \qquad \mathbf{v}_{v2} = \mathbf{v}(t_2) = \begin{bmatrix} -8.6838339145 \\ -0.2859264505 \\ -0.0345301070 \end{bmatrix}$$

Conclusions

The *DerAstrodynamics* **lambert2** algorithm can also be used for interplanetary trajectories, if the gravitational constant is replaced by that of the Sun. The four recommendations for further works by Klumpp (Reference 10) in 1991 are accomplished by the elegant formulation of Professor Sun of Reference 5 and the "modified" iterative method of Laguerre. The complicated starting formulae, third derivative evaluation for the Halley method, and the convergence testing developed by Gooding of Reference 7 are eliminated.

In summary:

1. No starter algorithm is needed. The initial guess of the unknown iterative parameter x for N = 0 can be simply set to ± 0.5. For any N, x is positive or negative according to the given time $t = (t_2 - t_1)$, less than or greater than the Minimum Energy time t_{ME}.
2. No averaging and/or lower and upper limits are computed or needed.
3. No Newton method is used; otherwise the algorithm will not be robust.
4. No fixed higher-order/degree iteration equation is designated, but the degree of the polynomial equation of the Laguerre method is allowed to vary.

Despite Conway's claim of robustness by using a "fixed" fifth-degree iterative equation of Laguerre for solving the Kepler equation (Reference 13), it is shown to fail in the *DerAstrodynamics* **kepler1** examples. In fact, the "fixed" fifth-degree iterative equation of Laguerre failed to predict a correct Kepler solution approximately one out of a thousand using the initial state vectors from any day of the NORAD Space Catalog and prediction times of almost any number of days. Similarly Halley's cubic iteration process cannot guarantee robustness, because it is a "fixed" cubic iterative equation.

Since the **lambert2** algorithm has the same independent "universal variable" parameter x as that of Gooding, strictly speaking Sun's formulation is universal.

The Primer Vector approach of Reference 11 has difficulties in the initial guess and convergence due to the unbounded independent parameter, and therefore solutions are not guaranteed. The Series Reversion/Inversion method of Reference 12 presents implementation difficulty of matrix manipulations, and is limited to less than one revolution. Instead of solving a Lambert equation, References 8 and 9 deduce the unknown from a cubic Kepler equation. Since the unknown is unbounded, solutions are not guaranteed. None of these Lambert algorithms and many others can guarantee accuracy and robustness.

The Gooding Lambert algorithm may be the best as suggested by Klumpp of JPL in 1991. The superior **lambert2** algorithm, which has eliminated all the difficulties of the Gooding Lambert algorithm, is the fastest, most accurate and robust, multi-revolution Lambert algorithm in 2011 and beyond. Using the analytic *DerAstrodynamics* **vinti** algorithm and simple targeting, the Keplerian solution of the **lambert2** algorithm can be extended to include the non-Keplerian terms of Vinti to achieve orders of magnitude accuracy improvement.

Appendix A

The Lambert equation from Reference 5 can be expressed as:

$$F(x) = \begin{cases} \dfrac{1}{\sqrt{(1-x^2)^3}}\left\{\cot^{-1}\left[\dfrac{x}{\sqrt{(1-x^2)}}\right] - \cot^{-1}\left[\dfrac{y}{\sqrt{(1-y^2)}}\right] - x\sqrt{(1-x^2)} + y\sqrt{(1-y^2)} + N\pi\right\} - \tau \\ \qquad (\text{elliptic},\ |x| < 1) \qquad \text{(A1)} \\ \dfrac{1}{\sqrt{(x^2-1)^3}}\left\{-\coth^{-1}\left[\dfrac{x}{\sqrt{(x^2-1)}}\right] + \coth^{-1}\left[\dfrac{y}{\sqrt{(y^2-1)}}\right] + x\sqrt{(x^2-1)} - y\sqrt{(y^2-1)}\right\} - \tau \\ \qquad (\text{hyperbolic},\ x > 1) \qquad \text{(A2)} \\ \dfrac{2}{3}(1 - \sigma^3) - \tau \\ \qquad (\text{parabolic},\ x = 1) \qquad \text{(A3)} \end{cases}$$

$$0 \le \cot^{-1}\left[\frac{x}{\sqrt{(1-x^2)}}\right] \le \pi, \qquad -\frac{\pi}{2} \le \cot^{-1}\left[\frac{y}{\sqrt{(1-y^2)}}\right] \le \frac{\pi}{2} \qquad \text{(A4)}$$

The first and second derivatives of the Lambert equation (A1) for multi-revolution elliptic trajectories can be expressed as:

$$F'(x) = \frac{\partial \tau}{\partial x} = \frac{1}{(1-x^2)}\left\{3x\tau - 2(1-\sigma^3)\frac{x}{|y|}\right\} \qquad \text{(A5)}$$

$$F''(x) = \frac{\partial^2 \tau}{\partial x^2} = \frac{1}{x(1-x^2)}\left\{(1+4x^2)F'(x) + 2(1-\sigma^5)\frac{x^3}{|y|^3}\right\} \qquad \text{(A6)}$$

where $y = \pm\sqrt{[1 - \sigma^2(1 - x^2)]}$. The sign of y is chosen as that of σ if $\sigma^2 \neq 0$, and $|y| = 1$ if $\sigma^2 = 0$. When N is specified and τ is given, the multi-revolution elliptic orbits equations (A1), (A5) and (A6) can be substituted into the Laguerre's iterative formula of equation (5). Together with the guessed value of x_1 in Step 8 of the Computational Procedure and any value of n of the polynomial degree (starting at n = 2), the **lambert2** algorithm can be initiated for any revolution $N \geq 0$. When x and y are substituted into equation (A1), equation (A4) must be strictly enforced, otherwise $F(x)$ will be computed incorrectly.

References

1. Gauss, C.F. "Theoria motas corporum coelestium in section-ibus conic solem ambientium", 1809, (English translation by C.H. Davis, Little, Brown & Co., Boston, 1857.
2. Battin, R.H. "An Introduction to The Mathematics and Methods of Astrodynamics", American Institute of Aeronautics and Astronautics, Education Series, 1987.
3. Godal, T. "Method for Determining the Initial Velocity Vector Corresponding to a Given Time of Free Flight Transfer Between Given Points in a Simple Gravitational Field." Astronautik, Vol. 2, 1961, pp 183-186.
4. Vinh, N.X."Invariance in the Lambert's Problem," Lecture Notes, The University of Michigan, 1977.
5. Sun, F.T. "On the Minimum Time Trajectory and Multiple Solutions of Lambert's Problem", AAS/AIAA Astrodynamics Conference, Province Town, Massachusetts, AAS 79-164, June 25-27, 1979.
6. Lancaster, E.R., Blanchard, R.C. "A Unified Form of Lambert's Theorem", NASA TN D 5368, Goddard Space Flight Center, Greenbelt MD, 1969.
7. Gooding, R.H. "A Procedure for the Solution of Lambert's Orbital Boundary-Value Problem", Celestial Mechanics and Dynamic Astronomy 48, Number 2, 1990.
8. Battin, R.H., Vaughan, R.H. "An Elegant Lambert Algorithm", Journal of Guidance and Control, Vol. 7, pp. 662-670, 1984.
9. Loechler, L.A. "An Elegant Lambert Algorithm for Multiple Revolution Orbits", Master of Science Thesis, MIT, 1988.
   http://dspace.mit.edu/bitstream/handle/1721.1/34998/20403052.pdf?sequence=1
10. Klumpp, A., "Performance Comparison of Lambert and Kepler Algorithms", Interoffice Memorandum, JPL, 1999 February.
11. Pressing, J.E., "A Class of Optimal Two-Impulse Rendezvous Using Multiple-Revolution Lambert Solutions", The Journal of the Astronautical Sciences, Vol. 48, Nos 2 and 3, April-September 2000.
12. Thorne, J.D., Bain, R.D., "Series Reversion/Inversion of Lambert's Time Function", The Journal of the Astronautical Sciences, Vol. 43, No. 3, 1995.
13. Pressing, J.E., Conway, B.A., "Orbital Mechanics", Oxford University, 1993.
14. Tewari, A., "Atmospheric and Space Flight Dynamics", Birkhauser, 2007.
15. Monuki, A.T. "Deviation of x-iterator and comparison of Lambert Routines", TRW 7121.3-173, May 7,1973.
16. Gooding, R.H. "A New Procedure for the Solution of the Classical Problem of Minimal Orbit Determination from three Lines of Sight", Celestial Mechanics and Dynamic Astronomy 66, Vol 4, 1997.
17. Vinti, J.P., "Orbital and Celestial Mechanics", AIAA, Volume 177,1998.
18. Bryson, A.E., Ho, Y.C., "Applied Optimal Control", Hemisphere Publishing, New York, 1975.
19. Danchick, R., "Accurate Numerical Partials with Applications to Maximum Likelihood Methods ", Internal Rand Report, 1975.