

REVIEW OF LAMBERT'S PROBLEM

David de la Torre Sangrà⁽¹⁾ and Elena Fantino⁽²⁾

⁽¹⁾*Polytechnic University of Catalonia (UPC), E.T.S.E.I.A.T. calle Colom 11, 08222 Terrassa (Spain), david.de.la.torre.sangra@upc.edu*

⁽²⁾*Space Studies Institute of Catalonia (IEEC), Polytechnic University of Catalonia (UPC), E.T.S.E.I.A.T., Colom 11, 08222 Terrassa (Spain), elena.fantino@upc.edu*

Abstract: *Lambert's problem is the orbital boundary-value problem constrained by two points and elapsed time. It is one of the most extensively studied problems in celestial mechanics and astrodynamics, and, as such, it has always attracted the interest of mathematicians and engineers. Its solution lies at the base of algorithms for, e.g., orbit determination, orbit design (mission planning), space rendezvous and interception, space debris correlation, missile and spacecraft targeting. There is abundance of literature discussing various approaches developed over the years to solve Lambert's problem. We have collected more than 70 papers and, of course, the issue is treated in most astrodynamics and celestial mechanics textbooks. From our analysis of the documents, we have been able to identify five or six main solution methods, each associated to a number of revisions and variations, and many, so to say, secondary research lines with little or no posterior development. We have ascertained plenty of literature with proposed solutions, in many cases supplemented by performance comparisons with other methods. We have reviewed and organized the existing bibliography on Lambert's problem and we have performed a quantitative comparison among the existing methods for its solution. The analysis is based on the following issues: choice of the free parameter, number of iterations, generality of the mathematical formulation, limits of applicability (degeneracies, domain of the parameter, special cases and peculiarities), accuracy, and suitability to automatic execution. Eventually we have tested the performance of each code. The solvers that incorporate the best qualities are Bate's algorithm via universal variables with Newton-Raphson and Izzo's Householder algorithm. The former is the fastest, the latter exhibits the best ratio between speed, robustness and accuracy.*

Keywords: *Lambert, Orbits, Two-Body Problem, Transfer Time Equation, Root Finding Algorithms.*

1. Introduction

This contribution deals with a prominent piece of Celestial Mechanics and Astrodynamics, the Two-Body Orbital Boundary-Value Problem (TBOBVP), also known as Gauss' or Lambert's Problem. It is the problem of determining the Keplerian orbit connecting two positions in a given time. Figure 1 illustrates its definition: F is the primary (the Sun in the case of interplanetary orbits, the Earth in the context of geocentric motion), origin of the reference frame, while \mathbf{r}_1 and \mathbf{r}_2 are the positions occupied by the secondary at times t_1 and t_2 , respectively (with $t_2 > t_1$), thus being $\Delta t = t_2 - t_1$ the flight time. The angle θ between \mathbf{r}_1 and \mathbf{r}_2 indicates the direction of motion. Finding the conic section that connects the two positions in the given time is equivalent to determining the velocity \mathbf{v}_1 to be imparted to the secondary at \mathbf{r}_1 to execute the transfer. The case shown as an example in Fig. 1 is an arc of an ellipse with one focus at F . The TBOBVP problem arose towards the end of the 18th century in connection with the determination of the orbits of celestial bodies from observation. Carl Friedrich Gauss (1777-1855) used three observation times instead of two (and

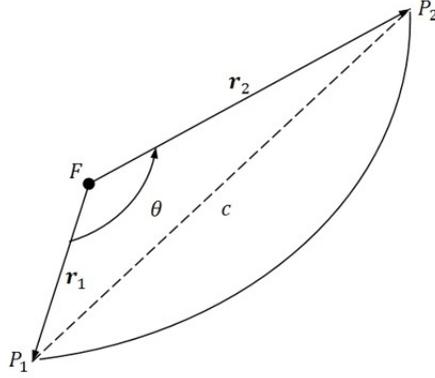


Figure 1. Definition of Lambert's problem.

with this method he was able to correctly determine the orbit of the newly-discovered Ceres). Gauss understood the potential of the TBOBVP and his developments on this subject were remarkable, with the result that the problem now bears his name. The work of Gauss merges with the previous findings of Johann Heinrich Lambert (1728-1777) who deduced the homonymous theorem: *Given the gravitational parameter $\mu = GM$, the time Δt required to accomplish a given transfer is a function of the semimajor axis a of the orbit, the sum $r_1 + r_2$ of the distances from the primary at the beginning and at the end of the transfer and the length c of the chord that connects such positions, i.e.:*

$$\sqrt{\mu}\Delta t = f(a, r_1 + r_2, c). \quad (1)$$

When the geometry of the radius vectors is fixed, there is only one free parameter left that wholly defines the transfer time between \mathbf{r}_1 and \mathbf{r}_2 . In the original formulation, such parameter is the unknown semimajor axis a . In general, once a suitable parametrization of the orbit passing through the given points is obtained, it is possible to rephrase Lambert's problem in terms of the evaluation of the parameter's value, such that the corresponding orbit is characterized by a transfer time that matches exactly the prescribed one. There exist several ways to express f , but no analytical closed-form solution of the transfer time equation (Eq. 1) is possible. Especially valuable and useful are the unified forms, i.e., providing one formulation valid for all three types of conic sections.

In modern Astrodynamics, Lambert's problem has direct application in the solution of intercept and rendezvous, ballistic missiles targeting and interplanetary trajectory design. The relevance of this issue is confirmed by the vast related literature. Several authors since the time of Gauss have studied alternative formulations of the transfer time equation, adopting a variety of independent variables (free parameter) and solution methods. Due to the transcendental character of the transfer time equation, all the available approaches are based on a numerical procedure in which the value of the free parameter is searched iteratively. Then, the orbital elements or the state vector at departure and arrival are obtained by means of orbital mechanics relations.

In this work, we present a review of the algorithms for the solution of Lambert's problem, based on quantitative comparisons. We start by discussing the relationships among the many methods, and we indicate the main representatives of the several research lines, as we identified them from our study of the literature (Sect. 2.). Then, in Sect. 3. we illustrate the tests that we carried out in order to compare the selected methods in terms of performance. We discuss the results and we draw

conclusions in Sect. 4.

2. Solution methods: comparison and discussion

The first formulations of Lambert's problem date back to Lagrange [1] and Gauss [2]. However, studies on the subject turned intensive in the mid 1960s. As of today, more than 60 authors have proposed formulations and solutions to the problem. All these methods can be grouped into a number of major lines of research on the basis of the free parameter adopted. Here we highlight the most productive ones, and for each we indicate the progenitor algorithm and the successive improvements upon it:

- Universal variables:
 - Lancaster & Blanchard [3], Gooding [4, 5], Izzo [6].
 - Bate [7], Vallado [8], Luo [9], Thomson [10], Arora [11].
 - Battin-Vaughan [12], Loechler [13], Shen [14], MacLellan [15].
- Semi-major axis:
 - Lagrange, Thorne [16, 17], Prussing [18], Chen [19], Wailliez [20].
- Semi-latus rectum (p -iteration):
 - Herrick-Liu [21], Boltz [22].
 - Bate [7].
- Eccentricity vector:
 - Avanzini [23], He [24], Zhang [25, 26], Wen [27].
- Kustaanheimo-Stiefel (K-S) regularized coordinates:
 - Simó [28].
 - Kriz [29], Jezewsky [30].

The work of Sun [31] is also worth mentioning, being the first to extensively investigate and analyse the multi-revolution Lambert's problem and derive an expression for the minimum transfer time in this case.

For brevity, in what follows we shall indicate [3] with LB69, [5] with G90, [6] with I15, [7] with B71, [8] with V97, [12] with BV84 and [28] with S73.

We have selected representatives of each line of work and we have studied them. The methods due to Lagrange and Gauss have been retained for historical reasons. From the Universal variables line, the selection includes B71, BV84, and the work by G90 and I15, both superseding the original algorithm of LB69. Finally, the work of S73 has been selected as representative of the K-S branch. In the remainder of this section, we will describe these algorithms. For reasons of space, the reader is referred to the original publications for details and more quantitative aspects.

2.1. Generic Lambert procedure

In general, the procedure to be adopted to solve Lambert's problem consists in:

1. computing the geometric parameters of the transfer;
2. obtaining an initial guess for the free parameter;
3. iterating on the transfer time equation until convergence;

4. computing the velocity vectors.

2.2. Transfer time equation

The algorithm by Lagrange expresses the transfer time as a function of the semi-major axis a . This choice is inconvenient for two reasons: the solution contains a pair of conjugate orbits (i.e., the solution is not unique), and the derivative of the transfer time is singular when a corresponds to the minimum-energy orbit (a_m). To overcome this difficulty, [32] expresses the transfer time equation as a function of the universal variable x ($x^2 = 1 - a_m/a$) which is valid for all types of conic sections, thus obtaining a single-valued and monotonic function for the transfer time.

In his *Theoria Motus*, Gauss develops a method to solve Lambert's problem with a system of two equations and using the sector-to-triangle area ratio y . Then, B71 expands the original Gauss definition to any type of conic section using a power series expansion developed by [33].

In his book, B71 starts from the Gauss f and g functions and expresses the transfer time in terms of the universal variable z using Stumpff functions. z is defined as ΔE^2 for elliptical orbits and as $-\Delta F^2$ for hyperbolic orbits, being ΔE and ΔF the eccentric anomaly in the ellipse and in the hyperbola, respectively. Note that Bate's time equation is singular at $z = (2n\pi)^2$ with $n = 1, 2, 3, \dots$, and $[2n\pi]^2$ and $[2(n+1)\pi]^2$ are the limits of the domain of n -revolution elliptical orbits. Note also that there is a lower limit for z when the transfer angle is less than π . Such limit corresponds to a hyperbolic orbit with transfer time $\Delta t = 0$. Lower values of z imply negative values for the transfer time.

S73 derives the transfer time in regularized space by means of the Levi-Civita coordinate transformation. His free parameter z has a universal definition, valid for the three types of conic section. The transfer time is singular at $z = n\pi^2$ with $n = 1, 2, 3, \dots$, and $[2n\pi]^2$ and $[2(n+1)\pi]^2$ are the limits of the domain of n -revolution elliptical orbits. Also in this case, there exists a lower limit for the value of z for transfers with $\theta < \pi$. The work also provides some insight on how to avoid and correct certain scenarios where numerical precision gets degraded.

BV84 improve the original Gauss formulation in their Elegant Algorithm, extending the convergence region and treating $\theta = \pi$ transfers. The new system of equations is also expressed in terms of the sector area-to-triangle area ratio y and makes use of a continued fraction. The method, however, is singular for 2π transfers.

G90 follows the work of LB69 in terms of the universal variable x ($x^2 = 1 - a_m/a$), implementing a Halley cubic iterator, initial guess value formulas to ensure fast convergence, a power-series expansion in the near-parabolic range to avoid round-off errors, and expanding its multi-revolution capabilities. In his paper, G90 provides a Fortran code with three routines: one computes the time of flight (and several of its derivatives) as a function of x and using the formulation by LB69. The second routine finds x , while implementing all the initial guesses for all orbits and multi-revolution cases. The main code computes the geometric parameters of the transfer, calls the second routine to compute x for each multi-revolution case, and computes the velocity vectors.

I15 also starts from the work by LB69, implements a new initial guess based on empirical results and a House-Holder iteration scheme that converges in about two or three iterations. The algorithm has been implemented in a python code available as part of the ESA PyKEP toolbox. For values of x close to one (i.e., near-parabolic range), I15 makes use of the Lagrange and Battin [32] equations in terms of the universal variable z . As part of the same toolbox, I15 also implements another version of the algorithm using only the Lagrange equation, a set of transformed variables and a Regula-Falsi iteration scheme.

2.3. Solution algorithm

Several methods can be used to numerically solve the equations resulting from the several treatments described in Sect. 2.2.. Each method has its benefits and shortcomings. For example, the Newton-Raphson method needs a single starter and has a high convergence rate but works well only with single-valued monotonic functions. Singularities or steep changes in the derivative slow down the convergence, make the method converge to the wrong solution, or even diverge. The bisection method, requires two starters bracketing the solution and it is usually slower than Newton-Raphson, but it will always converge to the the correct solution when properly bracketed. The iteration methods implemented for the selected Lambert algorithms are presented below.

The Newton-Raphson scheme is a Householder root-finding method of 1st order. The method requires the derivative of the transfer time with respect to the free parameter, which we denote by w . At the i^{th} iteration, w is updated through:

$$w_{i+1} = w_i - \frac{t - TOF}{\frac{dt}{dw}}. \quad (2)$$

TOF is the target value for the time of flight, t is current value of the transfer time and dt/dw is its derivative with respect to w . The Newton-Raphson method is used by Lagrange and B71.

The bisection method does not require any derivatives, since it evaluates the transfer time equation at the arithmetic mean point (e.g., \bar{w}) of a solution-bracketing interval (namely $[w_{low}, w_{up}]$). Then, the resulting value for the time of flight t is compared against the target TOF , and the interval bounds are updated as follows:

$$\begin{aligned} w_{low} &= \bar{w} & t \leq TOF, \\ w_{up} &= \bar{w} & t > TOF. \end{aligned} \quad (3)$$

The bisection method is implemented in the B71 and S73 algorithms.

The Gauss algorithm uses a successive substitution technique called the Gauss Method. The scheme recomputes all parameters with the current value of the free variable and updates it using the recomputed value of the parameters. BV84 also develops a successive substitution solving method, exploiting the property that the largest positive real root of the cubic equation is always the correct solution, and using the resulting equation to directly compute the solution for the cubic.

G90 uses the Halley method, which is a Householder root-finding method of 2nd order. The first and second derivatives of the transfer time equation with respect to the free parameter are then required

and used as follows:

$$w_{n+1} = w_n + \frac{T dt}{dt^2 + T \frac{d^2t}{2}} \quad (4)$$

where $T = TOF - t$, and dt , d^2t are the first and second order derivatives of the time equation, respectively.

I15 uses a 3rd-degree Householder method to iterate on the transfer time equation:

$$w_{n+1} = w_n T \frac{dt^2 - T d^2t/2}{dt(dt^2 - T d^2t) + d^3t d^2t/6} \quad (5)$$

where $T = TOF - t$, dt and higher-order derivatives are computed using LB69's version of the transfer time equation.

In the alternative version of the algorithm at the PyKEP toolbox, I15 uses a Regula-Falsi scheme as iteration method. The Regula-Falsi method requires a solution-bracketing interval as a starter and updates the iteration variable as follows:

$$w_{n+1} = \frac{w_1 T_2 - T_1 w_2}{T_2 - T_1} \quad (6)$$

where w_1 , w_2 are interval limits, while T_1 and T_2 are the values for the difference $t_n - TOF$ at each interval limit. Once the new value of the iterator (w_{n+1}) is computed, the corresponding value of the time of flight (t_{n+1}) is obtained via the transfer time equation. Then, the interval is updated as follows: $w_1 = w_2$, $w_2 = w_{n+1}$, $T_1 = T_2$ and $T_2 = T_{n+1}$. The process is repeated until convergence.

2.4. Choice of the initial guess

The initial guess, or starter, is introduced as input to the solution algorithm. Note that each solution algorithm may require a specific number of initial guesses (e.g., Householder-type methods require a single starter, bisection requires two starters, etc.). The starters proposed by the authors of the selected algorithms are the following:

Lagrange starts from $x = 0$, corresponding to a minimum-energy orbit.

Gauss starts from $y = 1$, which corresponds to a 1:1 triangle-to-sector area ratio.

B71 starts from $z = 0$ for the Newton-Raphson scheme, corresponding to a parabolic orbit. For the bisection method, [8] suggests an upper bound of $z_{up} = 4\pi^2$, corresponding to the $t = 2\pi$ asymptote, and a lower limit of $z_{low} = -4\pi$, which he claims is valid for most orbits except for highly-eccentric ones. In such cases, the lower limit should be extended.

S73 proves that his time equation has a single solution in the interval $z \in [z_f, \pi^2]$, where π^2 is the $t = 2\pi$ asymptote and z_f is the lower limit for the z parameter. The value of z_f corresponds to a specific value for $\theta < \pi$ transfers and $z_f = -\infty$ for $\theta > \pi$ transfers. S73 recommends to

use $z = (\theta/2)^2$ as a starter for the elliptical orbits domain ($z \in [0, \pi^2]$) and $z = 0$ as a starter for hyperbolic orbits ($z \in [z_f, 0]$).

BV84 provides a specific initial guess x for his successive substitution method if the normalized time of flight corresponds to an elliptical orbit, and $x = 0$ otherwise.

G90 derives a set of semi-empirical initial starters for both single-revolution and multi-revolution transfers. For the single-revolution case, he uses a bilinear approximation method to analytically obtain an approximate solution for the $x < 0$ region of Lancaster's transfer time equation, and a weighted combination of approximate solutions for the $x > 0$ region. The multi-revolution starters are far more complex and, for reasons of space, the reader is referred to the corresponding publications.

I15 performs a piece-wise linear approximation of the transfer time equation. Inverting the previous relation provides an approximate solution which is then used as a starter. For the multi-revolution cases, I15 approximates the lower and upper asymptotes of the time equation for the corresponding n -revolution scenario, then he inverts the approximated curves to obtain the appropriate starters. For his Regula-Falsi implementation, I15 uses an empirical set of constant starters for the single-revolution case or multi-revolution case.

2.5. Computation of the velocity vectors

Once the free parameter has been approximated, the velocity vectors at the boundaries of the transfer must be determined. The methods employed by the several authors are highlighted below.

2.5.1. Orbital elements

This method consists in expressing the semi-latus rectum p , the semi-major axis a , the eccentricity e and the true anomaly ν of the selected point as functions of the free parameter. Then, the velocity \mathbf{v}_{pqw} at either \mathbf{r}_1 and \mathbf{r}_2 in the perifocal reference frame is determined as:

$$\mathbf{v}_{pqw} = \sqrt{\frac{\mu}{p}} [-\sin \nu, e + \cos \nu, 0]. \quad (7)$$

Then, with the inclination i , the argument of the periapsis ω and the longitude of the ascending node Ω , the pqw vectors can be rotated into the xyz frame:

$$\mathbf{v}_{ijk} = R_z(\Omega)R_x(i)R_z(\omega)\mathbf{v}_{pqw}, \quad (8)$$

where R_k 's are the elementary rotation matrices following the right-hand sign convention. The orbital elements method is used by S73 and BV84.

2.5.2. Gauss f and g functions

The values for the Gauss functions f , g and \dot{g} are expressed as functions of the free parameter. Then the velocity vectors \mathbf{v}_1 and \mathbf{v}_2 at the beginning and at the end of the transfer, respectively, are found

through:

$$\mathbf{v}_1 = (\mathbf{r}_2 - f\mathbf{r}_1)/g \quad (9)$$

$$\mathbf{v}_2 = (\dot{g}\mathbf{r}_2 - \mathbf{r}_1)/g \quad (10)$$

Note that there exists a singularity in the computation of the velocity when $g = 0$, which corresponds to $\theta = \pi$ transfers. This method is used by Gauss and B71.

2.5.3. Radial and transversal components

The radial v_r and transversal v_t components of the velocity are expressed as functions of the free parameter. Then, either velocity vector \mathbf{v}_i (i.e., \mathbf{v}_1 or \mathbf{v}_2) is calculated as:

$$\mathbf{v}_i = v_{r_i} \frac{\mathbf{r}_i}{r_i} + v_{t_i} \frac{\mathbf{h} \times \mathbf{r}_i}{|\mathbf{h} \times \mathbf{r}_i|}. \quad (11)$$

Here, \mathbf{r}_i/r_i is the unit vector in the radial direction at P_i , and $\mathbf{h} \times \mathbf{r}_i/|\mathbf{h} \times \mathbf{r}_i|$ is the unit vector in the transversal direction. Lagrange, G90 and I15 use this method.

3. Performance tests

The methods discussed in Sect. 2. have been implemented following the description of their authors. Only single-revolution cases have been considered for this contribution. The implementation has been done in Matlab. This programming framework allows for a fast prototyping and includes powerful code analysis and debugging tools, which favours the algorithm-oriented analysis that we aim at in this work. Although a preliminary performance-oriented analysis has also been carried in order to obtain a relative comparison between methods, the correct practice in this case should be to program the algorithms in a high-performance compiled code (e.g., FORTRAN or C). The latter approach will be addressed in a future development of this work.

3.1. Application region

Several simulations have been carried out in order to test the algorithms in a wide range of scenarios. The goal is to stress each method and detect its range of applicability, slow-convergence zones, singularities, etc. The scenario sets a transfer between two radial distances, r_1 and r_2 , such that $r_2/r_1 = 2$. Then the transfer angle θ is varied between 0 and 2π at constant steps, and for each value of the transfer angle, the transfer time is also varied between 0 and 2π . The normalization has been applied which consists in that μ and r_1 are unitary and the circular orbit with radius r_1 has unit mean motion. This setup ensures that the algorithms are tried also in the tough cases, including $\theta = 0, \pi, 2\pi$, parabolic orbits, highly-eccentric hyperbolic orbits. The results of the simulations are discussed below.

Battin's implementation of Lagrange's equation fails for highly-eccentric hyperbolic orbits with $TOF < \pi/4$ (Fig. 5), corresponding to values of the universal variable $x \gg 1$. In this scenario, the Gauss hypergeometric function ${}_2F_1$ does not converge because its convergence radius (unitary) is exceeded. As a result, the equation suffers a steep change in the derivative with which the

Newton-Raphson method can no longer cope (Fig. 2). The method performs 10-20 iterations in the elliptical orbits domain (best case near $\theta \approx \pi$), and executes 40-60 iterations when approaching the non-convergence zone of ${}_2F_1$.

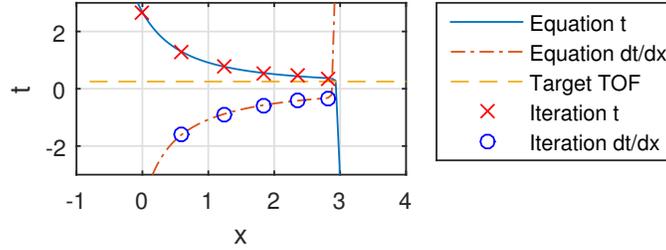


Figure 2. Lagrange Newton-Raphson solver in the low-TOF scenario.

The Gauss Method algorithm expanded by Bate only works for scenarios where $y \approx 1$, which includes both elliptical orbits with a small transfer angle and hyperbolic orbits with a moderate transfer angle. In general, the convergence region is given by the approximate relation $\theta < \pi - TOF$, as shown in Fig. 5. The method, however, fails completely in other cases. It is worth mentioning that for $\theta > \pi$, the value of the triangle-to-sector area ratio y becomes negative, which corresponds to a physically impossible situation. Also, there is a small region for high values of θ where the algorithm converges but the results are obviously incorrect. The precision of the results in the convergence region is also relatively low compared with other algorithms (more than 10^{-2} relative error on the magnitude of the velocity vector). Due to the adoption of the f and g functions for the computation of the velocity vectors, the method also intrinsically fails for the $\theta = \pi$ transfer. The Gauss Method by Bate makes approximately 30 iterations in the elliptical orbits zone and 10-20 in the hyperbolic orbits domain, but the iteration number quickly increases to more than 100 when moving away from the convergence region.

B71 method using universal variables and Newton-Raphson with a constant starter $z = 0$ (parabolic orbit) has two conflictive regions. One corresponds to large values of TOF and high θ in the elliptical orbits range (Fig. 5), where the solution is a point close to the $z = 4\pi^2$ asymptote and the derivative is steep. In this situation, Newton-Raphson has a high chance to “jump” over the asymptote and fall into the multi-revolution region, where the error is propagated further (Fig. 3). The second conflictive zone is associated to $\theta < \pi$ (type I, short-way transfers) and $TOF < \pi/8$ in the hyperbolic orbits region, where the solution falls close to the minimum negative value for z . In this case, Newton-Raphson has a high chance of falling into the imaginary region of B71 equation (Fig. 4), thus yielding imaginary numbers. These two failure regions can be avoided by improving the initial guess, for example with a preliminary search to bracket the solution within bounds and then selecting the geometric mean between bounds as a starter. In this case, the convergence is greatly improved, except for a few cases with extremely low or extremely high values of TOF . The number of iterations in any case ranges from 5-10.

B71 method with bisection and the starter proposed by [8] works for all orbits except those with $\theta > 3/4\pi$ and $TOF < \pi/2$ in the hyperbolic orbits region, where the initial bounds no longer bracket the solution (see Fig. 5). In such cases, he suggests to decrease the value of the lower

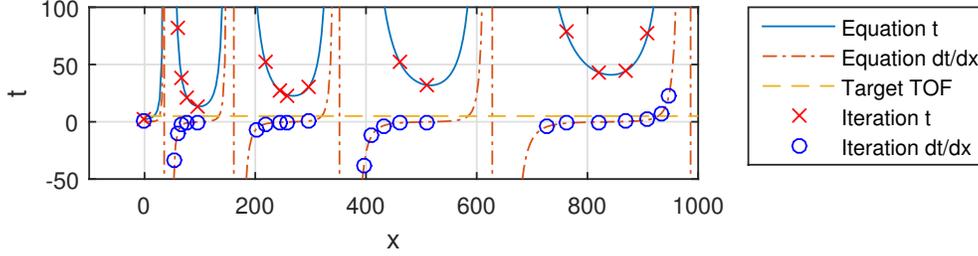


Figure 3. Bate’s Newton-Raphson solver for the scenario with high values of TOF and θ .

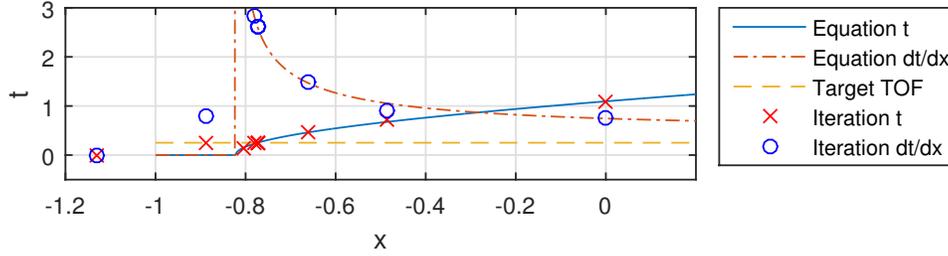


Figure 4. Bate’s Newton-Raphson solver for the scenario with very low TOF and $\theta < \pi$.

starter until the solution is bracketed again. Of course, this new search implies a slight increase in computing time, but convergence is eventually achieved in all regions. The iteration counter is fairly constant and close to 40 iterations. Note that both B71 methods fail to compute the velocity vectors for $\theta = \pi$ due to the singularities in the f and g functions.

S73 algorithm presents the same conflictive regions as B71, as expected. The improvement comes again from a more careful choice of the starter, which in this case has been implemented also with a preliminary search for a solution-bracketing bounds. With the bisection method, the convergence is guaranteed in all regions, as long as the solution is properly bracketed first. S73 computes the velocity vectors with the method of the orbital elements, which gives good results for all orbits (including $\theta = \pi$ transfers). The iteration counter stays fairly constant around 40 iterations in the $0 < TOF < \pi$ region and close to 35 iterations in the $\pi < TOF < 2\pi$ region.

BV84 Elegant Algorithm implemented using Battin’s successive substitution method with direct computation of the cubic root fails for large values of TOF and θ , as illustrated in Fig. 5. In this scenario, one of the parameters in BV84 equations produces a negative square root and therefore the method gives imaginary solutions. The algorithm converges in all other regions, although the precision is relatively low ($10^{-5} - 10^{-3}$ errors relative to the other algorithms). The algorithm converges in about 4-5 iterations for the region $\theta < \pi$ and $TOF < \pi$ (the same region as Gauss), 8-9 iterations for hyperbolic orbits with $\theta > 3/2\pi$, and 6-7 iterations elsewhere.

G90 algorithm has no evident singularities in the entire single-revolution region, except when $TOF = 0$ and $\theta = 0$ which give infinite velocity. The algorithm performs a fixed number of iterations, as Gooding demonstrates that three iterations are sufficient to provide 13 digits precision for all cases in a single-revolution scenario. However, the error slowly increases with the multi-revolution number, which would require the convergence to be re-assessed.

Both versions of I15 algorithm converge in all cases except for $TOF = 0$ and $\theta = 0$. The Regula-Falsi method converges in 5-6 iterations, while the Householder version converges in just 2-3 iterations.

A combined view of the regions of application of the several algorithms is shown in Fig. 5. The optimized versions of both B71 algorithms (marked with an “O” suffix in the figure) implement an improved initial guess estimation.

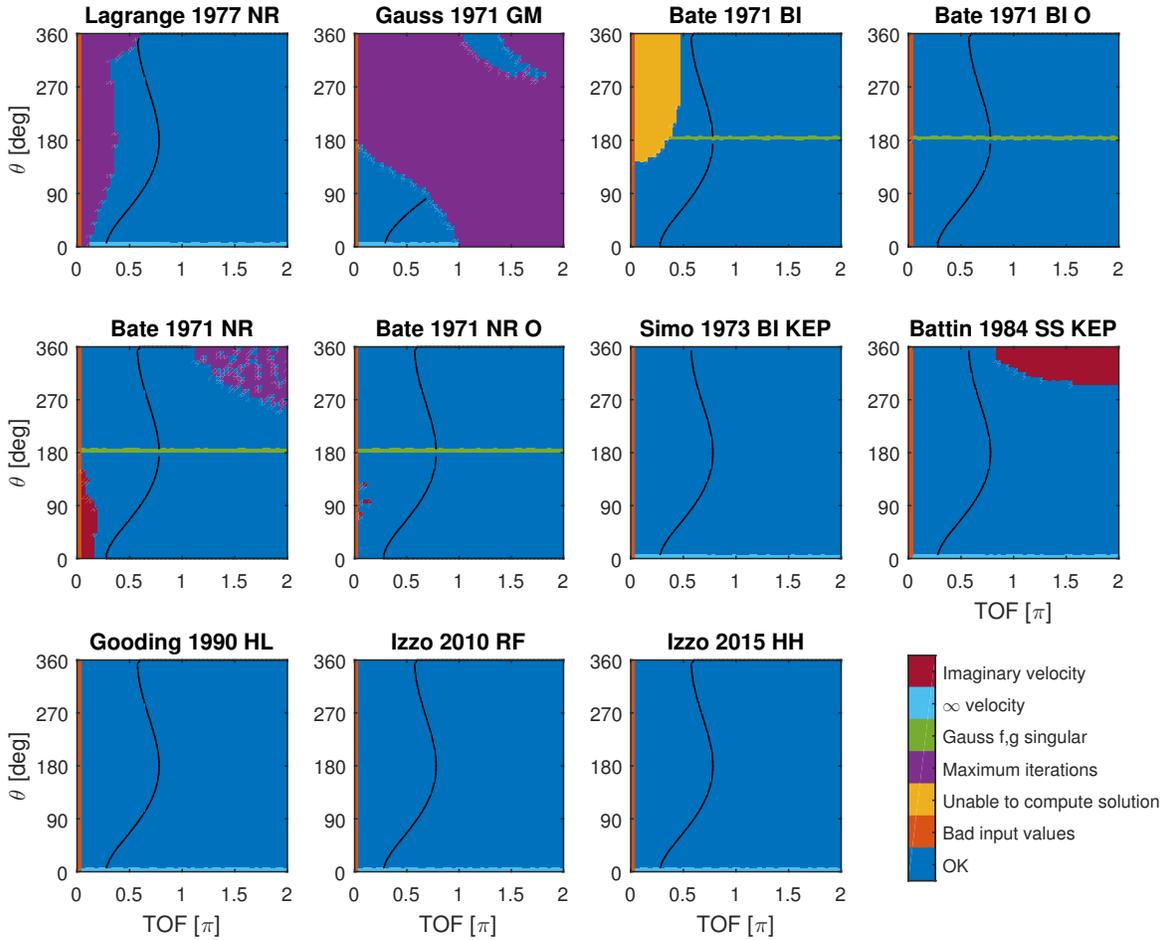


Figure 5. Region of application and degeneracies of Lambert’s algorithms.

3.2. Computational performance

A computational performance test has been carried out in order to estimate the execution time of each algorithm. The algorithms have been tested on a Monte-Carlo simulation of 10^5 random distributed samples within the single-revolution region defined in Sect. 3.1., and only the points for which the algorithm actually converges correctly have been used to compute the resulting average execution time. This procedure aims at simulating a realistic application scenario.

While implementing the algorithms, extremely large improvements in execution time were observed whenever the code was slightly optimised. Certainly, as the algorithms take of the order of

microseconds to execute, the normally small overhead caused by callbacks to external functions (e.g., norm, dot, cross, Stumpff functions, series, etc.) and inefficient computations (e.g., repeating trigonometric calculations) end up having a great impact on the overall execution time. In the end, all the codes were optimized by using only intrinsic functions, pre-computing repeated calculations, minimising the number of calls to external functions and subroutines whenever possible, and keeping a strictly uniform coding style among the implementations. Note that any different implementation will most probably produce slightly different results.

The results of the performance comparison are presented in Fig. 6. The reader should take these results with care: first of all, the computing speed of Matlab’s interpreted code is largely slower than compiled code (around 8 times slower). The Matlab processor may also have automatically used threading and optimization capabilities on specific portions of the code that could eventually alter the results. Finally, executing the code under a UNIX system would provide better results in terms of speed. Hence, these values can be taken as a first-order approximation in a comparison among the algorithms, or if the application is to be run in the Matlab platform.

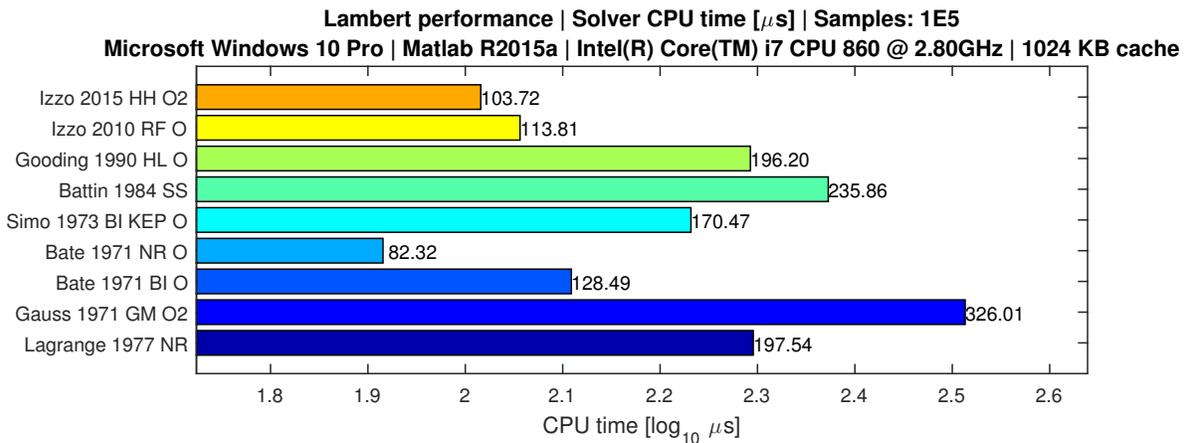


Figure 6. CPU execution time of Lambert algorithms

4. Conclusions

The existing bibliography on Lambert’s problem has been reviewed and organized. The major existing algorithms have been selected, analysed and compared in terms of formulation of the transfer time equation, initial guess estimation, choice of the free parameter, iteration method, and velocity vector computation scheme. A series of tests have been carried out to determine the range of applicability and the computational performance of each algorithm. B71 algorithm via universal variables with Newton-Raphson appears to be the fastest solver, but I15 Householder algorithm exhibits the best ratio between speed, robustness and accuracy.

The foreseen developments include implementing the algorithms in compiled code (FORTRAN), testing the implementation under a GPU computing framework (CUDA), incorporating additional algorithms, and expanding the analysis to include multi-revolution solutions.

Acknowledgements

David de la Torre acknowledges financial support by DLR. The authors are grateful to Prof. Gerard Gómez for suggesting this research and for helpful discussions throughout.

5. References

- [1] Lagrange, J. L. *Mécanique Analytique*, Vol II. Académie des Sciences, Paris, 1788.
- [2] Gauss, C. F. *Theory of the motion of the heavenly bodies moving about the Sun in conic sections: a translation of Carl Frdr. Gauss Theoria motus*. Brown and Co, 1857.
- [3] Lancaster, E. R. “Solution for Lambert problem for short arcs.” Tech. rep., NASA Goddard Space Flight Center, Greenbelt, MD, United States, 1969.
- [4] Gooding, R. H. “On the solution of Lambert’s orbital boundary-value problem.” Tech. rep., Royal Aerospace Establishment, Farnborough, Hante, 1988.
- [5] Gooding, R. H. “A procedure for the solution of Lambert’s orbital boundary-value problem.” *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, No. 2, pp. 145–165, 1990. ISSN 09232958. doi:10.1007/BF00049511.
- [6] Izzo, D. “Revisiting Lambert’s problem.” *Celestial Mechanics and Dynamical Astronomy*, Vol. 121, No. 1, pp. 1–15, Jan. 2015. ISSN 0923-2958. doi:10.1007/s10569-014-9587-y.
- [7] Bate, R., Mueller, D., and White, J. *Fundamentals of Astrodynamics*. Dover Publications, 1 edn., 1971. ISBN 0486600610.
- [8] Vallado, D. A. *Fundamentals of Astrodynamics and Applications*. McGraw-Hill Companies, 1997. ISBN 9780070668348.
- [9] Luo, Q., Meng, Z., and Han, C. “Solution algorithm of a quasi-Lambert’s problem with fixed flight-direction angle constraint.” *Celestial Mechanics and Dynamical Astronomy*, Vol. 109, No. 4, pp. 409–427, 2011. ISSN 09232958. doi:10.1007/s10569-011-9335-5.
- [10] Thompson, B. F. “Enhancing Lambert Targeting Methods to Accommodate 180-Degree Orbital Transfers.” *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, pp. 1925–1929, 2011. ISSN 0731-5090. doi:10.2514/1.53579.
- [11] Arora, N. and Russell, R. P. “A Fast and Robust Multiple Revolution Lambert Algorithm Using a Cosine Transformation.” “Paper AAS 13-728, AAS/AIAA Astrodynamics Specialist Conference,” Hilton Head, SC, 2013.
- [12] Battin, R. H. and Vaughan, R. M. “An elegant Lambert algorithm.” *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 6, pp. 662–670, Nov. 1984. ISSN 0731-5090. doi: 10.2514/3.19910.
- [13] Loechler, L. A. *An Elegant Lambert Algorithm for Multiple Revolution Orbits*. Master thesis, Massachusetts Institute of Technology, 1988.

- [14] Shen, H. and Tsiotras, P. “Using Battin’s Method To Obtain Multiple-Revolution Lambert’s Solutions.” *Advances in the Astronautical Sciences*, , No. 757, pp. 1–18, 2004. ISSN 0065-3438.
- [15] Maclellan, S. J. “Orbital Rendezvous Using an Augmented Lambert Guidance Scheme.” pp. 1–122, 2005.
- [16] Thorne, J. D. and Bain, R. D. “Series Reversion/Inversion of Lambert’s Time Function.” *Journal of the Astronautical Sciences*, Vol. 43, No. 3, pp. 277–287, 1995.
- [17] Thorne, J. D. “Convergence Behavior of Series Solutions of the Lambert Problem.” *Journal of Guidance, Control, and Dynamics*, , No. 8, pp. 1–6, Aug. 2014. ISSN 0731-5090. doi:10.2514/1.G000701.
- [18] Prussing, J. E. “A Class of Optimal Two-Impulse Rendezvous Using Multiple-Revolution Lambert Solutions.” *Journal of the Astronautical Sciences*, Vol. 48, No. 2,3, pp. 131–148, 2000.
- [19] Chen, T., van Kampen, E., Yu, H., and Chu, Q. P. “Optimization of Time-Open Constrained Lambert Rendezvous Using Interval Analysis.” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 1, pp. 175–184, Jan. 2013. ISSN 0731-5090. doi:10.2514/1.56773.
- [20] Wailliez, S. E. “On Lambert’s problem and the elliptic time of flight equation: A simple semi-analytical inversion method.” *Advances in Space Research*, Vol. 53, No. 5, pp. 890–898, Mar. 2014. ISSN 02731177. doi:10.1016/j.asr.2013.12.033.
- [21] Herrick, S. and Liu, A. “Two Body Orbital Determination From Two Positions and Time of Flight.” *Aeronutronic*, Vol. C-365, No. Appendix A, 1959.
- [22] Boltz, F. W. “Second-order p-iterative solution of the Lambert/Gauss problem.” *Journal of the Astronautical Sciences*, Vol. 32, pp. 475–485, 1984.
- [23] Avanzini, G. “A Simple Lambert Algorithm.” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 6, pp. 1587–1594, Nov. 2008. ISSN 0731-5090. doi:10.2514/1.36426.
- [24] He, Q., Li, J., and Han, C. “Multiple-Revolution Solutions of the Transverse-Eccentricity-Based Lambert Problem.” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, pp. 265–269, Jan. 2010. ISSN 0731-5090. doi:10.2514/1.45041.
- [25] Zhang, G., Mortari, D., and Zhou, D. “Constrained Multiple-Revolution Lambert’s Problem.” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, pp. 1779–1786, Nov. 2010. ISSN 0731-5090. doi:10.2514/1.49683.
- [26] Zhang, G., Zhou, D., and Mortari, D. “Optimal two-impulse rendezvous using constrained multiple-revolution Lambert solutions.” *Celestial Mechanics and Dynamical Astronomy*, Vol. 110, No. 4, pp. 305–317, Aug. 2011. ISSN 0923-2958. doi:10.1007/s10569-011-9349-z.
- [27] Wen, C., Zhao, Y., and Shi, P. “Derivative Analysis and Algorithm Modification of Transverse-Eccentricity-Based Lambert Problem.” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, pp. 1195–1201, Jul. 2014. ISSN 0731-5090. doi:10.2514/1.62351.

- [28] Simó, C. “Solución del Problema de Lambert Mediante Regularización.” *Collectanea Mathematica*, Vol. 24, No. 3, pp. 231–247, 1973.
- [29] Kriz, J. “A uniform solution of the Lambert problem.” *Celestial Mechanics*, Vol. 14, No. 4, pp. 509–513, 1976. ISSN 00088714. doi:10.1007/BF01229061.
- [30] Jezewski, D. J. “K/S two-point-boundary-value problems.” *Celestial Mechanics*, Vol. 14, No. 75, pp. 105–111, 1976.
- [31] Sun, F.-T. “On the Minimum Time Trajectory and Multiple Solutions of Lambert’s Problem.” “AAS/AIAA, Astrodynamics Specialist Conference,” Provincetown, Mass., June 25-27, 1979.
- [32] Battin, R. H. *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Edition. American Institute of Aeronautics and Astronautics, Reston, VA, Jan. 1999. ISBN 978-1-56347-342-5. doi:10.2514/4.861543.
- [33] Moulton, F. R. *An Introduction to Celestial Mechanics*. Dover Publications, 2 edn., 1984. ISBN 978-0486646879.