

UPCommons

Portal del coneixement obert de la UPC

<http://upcommons.upc.edu/e-prints>

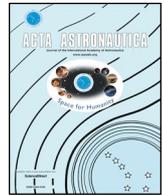
Aquesta és una còpia de la versió *author's final draft* d'un article publicat a la revista *Acta astronautica*.

URL d'aquest document a UPCommons E-prints:

<http://hdl.handle.net/2117/123263>

Article publicat / *Published paper*:

De la Torre, D., Flores, R., Fantino, E. On the solution of Lambert's problem by regularization A: *Acta Astronautica* (2018), vol. 163, p. 26-38. DOI: <[10.1016/j.actaastro.2018.10.010](https://doi.org/10.1016/j.actaastro.2018.10.010)>.



On the solution of Lambert's problem by regularization

D. De La Torre^a, R. Flores^b, E. Fantino^{c,*}

^a Polytechnic University of Catalonia (UPC), Physics Department, ESEIAAT, Colom 11, 08222, Terrassa, Spain

^b International Center for Numerical Methods in Engineering (CIMNE), Polytechnic University of Catalonia (UPC), Building C1, Campus Norte, UPC, Gran Capitán S/n, 08034, Barcelona, Spain

^c Khalifa University of Science and Technology, Department of Aerospace Engineering, P.O. Box 127788, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:

Two-body problem
Lambert's problem
Levi-Civita regularization

ABSTRACT

Lambert's problem is the two-point boundary-value problem resulting from a two-body orbital transfer between two position vectors in a given time. It lies at the very heart of several fundamental astrodynamics and space engineering problems and, as such, it has attracted the interest of scientists over centuries. In this work, we revisit the solution of Lambert's problem based on Levi-Civita regularization developed by Carles Simó in 1973. We offer an exhaustive derivation of the theory, including proofs of all the results and the formulae employed, and we extend the algorithm to deal with multi-revolution transfers. Then, after investigating a range of initial guess search techniques and testing different numerical methods to approximate the solution, we propose a procedure in which the initial guess is assigned very efficiently by querying a pre-defined interpolating table. Then, in order to achieve both speed and robustness, we combine Newton-Raphson with safety checks to avoid out of boundary deviations to approximate the solution. We validate the method through several tests and applications, and we assess its convergence and performance. The algorithm presents no singularities, converges in all realistic scenarios and its computational cost is comparable with state of the art algorithms.

1. Introduction

Lambert's problem, also known as Gauss' problem or two-body orbital boundary-value problem, is the problem of determining the Keplerian orbit connecting two positions in a specified time (Fig. 1). It arose in the 18th century as a core component in the determination of an orbit from three observations of direction alone, the central observation being used as a source of the missing distance data for the other two observations. Hence, the solution of Lambert's problem was essential for obtaining the elements of the orbits of planets, asteroids and comets. The problem is associated with the name of Johann Heinrich Lambert (1728–1777) because of his fundamental contributions to the topic, including the statement of the theorem that bears his name: *The transfer time Δt of a body moving between two points \mathbf{r}_1 and \mathbf{r}_2 on a conic trajectory is a function only of the sum $r_1 + r_2$ of the distances of the two points from the origin of the force, the linear distance c between the points, and the semi-major axis a of the conic section:*

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

a is the quantity to be determined, i.e., the unknown.

Equation (1) constitutes the general, implicit form of the transfer-time equation, the explicit layout depending on the way in which the

assigned geometry (the set of \mathbf{r}_1 , \mathbf{r}_2 and the transfer angle $\Delta\theta$) is represented and on the choice of the unknown, i.e., a or any other parameter from which the orbital elements or the initial velocity \mathbf{v}_1 can be deduced. The solution of the transfer-time equation cannot be expressed in closed analytical form and, therefore, must be numerically approximated.

Nowadays, Lambert's problem has important applications in trajectory design and navigation, being keystone in computations of rendezvous and interception, missile and spacecraft targeting, interplanetary transfers design and orbit determination. The transfer-time equation has attracted the interest of mathematicians of all times, whereas from the engineering view point its solution calls for accuracy, speed of convergence, robustness and formulations valid for all types of conic sections (universality). As a result, Lambert's problem is certainly one of the most investigated topics in Celestial Mechanics and Astrodynamics. This explains why over the years several solution techniques have been developed, which can be grouped based on the choice of the unknown parameter. The most common representation is based on universal variables, with other major groups of methods solving with respect to the semi-major axis, the semi-latus rectum, the transverse eccentricity vector or the flight path angle. Levi-Civita regularization of the equations of motion [1] allows a representation of the

* Corresponding author.

E-mail address: elena.fantino@ku.ac.ae (E. Fantino).

<https://doi.org/10.1016/j.actaastro.2018.10.010>

Received 26 March 2018; Received in revised form 18 September 2018; Accepted 4 October 2018

0094-5765/© 2018 IAA. Published by Elsevier Ltd. All rights reserved.

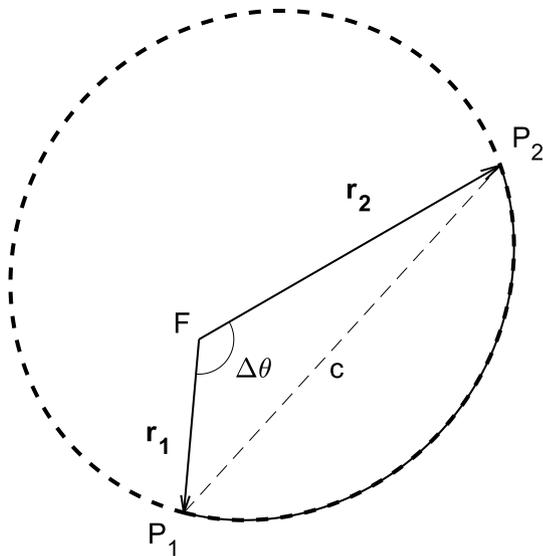


Fig. 1. The geometry of Lambert's problem: the occupied focus F (position of the primary), the start (\mathbf{r}_1, P_1) and end (\mathbf{r}_2, P_2) position vectors, the transfer angle $\Delta\theta$, the chord c and the conic section that solves the problem (in the example shown it is an elliptical orbit). In the multi-revolution case, the secondary completes n revolutions (dashed line) around F starting from P_1 , and eventually describes the arc (solid line) from P_1 to P_2 , so that the total transfer angle is $2\pi n + \Delta\theta$.

transfer-time equation as a function of the eccentric anomaly difference between the two boundary positions on the solution curve.

In the present contribution, we revisit the solution of Lambert's problem by Levi-Civita regularization. The fundamental work of Simó [2] has been our reference throughout. In his original publication, he developed an elegant derivation of the regularized equations of motion, the corresponding expression for the conservation of energy and a formulation of the transfer-time equation in terms of Stumpff functions. The argument of the series, the unknown parameter, is the product of the specific two-body energy and the square of the new (regularized) time coordinate, and such product is equivalent to the square of half the variation in eccentric anomaly on the conic section that solves a given Lambert's problem. Simó's method addresses all the issues that may arise when dealing with the solution of Lambert's problem: singularities, differences among the three types of conic section, degeneracies (most noticeably, the rectilinear orbit), choice of the initial guess, numerical divergence and loss of precision. Finally, he tested the algorithm in a small set of heliocentric and geocentric scenarios. Unfortunately, Simó's paper was written in Spanish, a fact that has undoubtedly limited its international dissemination. Furthermore, the derivation of the Levi-Civita theory was very concise and skipped many nontrivial steps. This made the discussion difficult to follow for readers unfamiliar with the subject.

In a bid to increase the awareness of the scientific community, we offer an exhaustive derivation of the theory including proofs of all the results and the formulae employed. We improve on the original formulation by completely solving the multi-revolution case for the first time in this context. This creates a very powerful tool, suitable even for designing gravity-assisted transfers to the inner planets (see, e.g. [3], for the case of the Messenger probe) or tours of planetary systems. We also investigate a range of initial guess search techniques and test different numerical methods to approximate the solution. We implement the algorithm in modern hardware and perform extensive benchmarking, including determination of the domain of applicability and performance assessment. The need for such rigorous testing lies in the automatic character of most applications, in which many solutions of Lambert's problem are obtained by varying the input parameters. Robustness, accuracy and execution speed are crucial for the purpose. We

perform direct comparisons with the *de facto* standard methods [4], (used by ESA) and [5] (NASA's choice). Simo's scheme offers the same level of robustness and universality as these reference algorithms (i.e., actually treating the three types of conics, solving undefined transfer plane cases and multi-revolution geometries over the entire application domain). It is the firm belief of the authors that Simo's formulation is competitive with the current methods of choice. Unfortunately, as stated above, it has not enjoyed sufficient attention. By providing a detailed description and improvements of the algorithm, including implementation in modern computers and benchmarks against the most popular methods, we hope to encourage further research and development to tap its full potential.

The paper starts with a chronological note on Lambert solvers (Sect. 2). The fundamentals of two-body Levi-Civita regularization and the application to the derivation of the equations of motion and their solution are dealt with in Section 3. The transfer-time equation, including the multi-revolution case, is obtained in Section 4. The method to obtain the orbital elements from the estimate of the unknown parameter is addressed in Sect. 5. The solution procedure and the associated numerical methods are presented in Sect. 6. Section 7 illustrates and discusses the tests carried out to assess the domain of applicability of the formulation and to evaluate the speed of convergence, the computational efficiency and the accuracy of the results. Section 8 concludes the paper and is followed by three appendices, respectively on the Stumpff functions and their properties (App. A), on the computation of the derivative of the transfer time with respect to z (App. B) and on the geometrical meaning of z (App. C).

2. State of the art

The original algorithm by [6] iterated on the ratio of the area of the sector to the area of the triangle for a given transfer geometry (see Fig. 1). This technique was later improved by [7], who also developed a method in which the unknown is half the difference between the eccentric anomalies at P_2 and P_1 on the solution curve. The latter approach was enhanced by [8,9] and [10] by including the multi-revolution case (see Fig. 1) and by proposing an improved initial guess search method.

[11] adopted a formulation based on universal variables, expressing the transfer-time equation via Gauss' f and g functions. [12] reformulated Bate's algorithm by adding a constraint on the flight path angle at P_1 , whereas [13] extended it using $\cos\Delta\theta$ as the unknown and solving with a variable-order Householder algorithm. [14] sets up a Lambert solver based on the Newton-Raphson method applied to universal variables. [5] modified it by means of a Halley iterator and improved the initial guess. [4] also built on [14], devising a Householder iterator and a new initial guess search method.

Most of the techniques that solve Lambert's problem with respect to the semi-major axis rely on the formulation due to [15]. The first investigations in this direction are those of [16] and [17]. More recently, [18] extended Prussing's algorithm by means of a variable-order Householder iterator, whereas [19] linearised the transfer-time equation, obtaining a computationally efficient procedure which is suitable for on-board targeting and orbit determination computations. Other implementations include those of [20], who used a power series, [21,22] and [23].

The main solvers with respect to the semi-latus rectum are due to [11,24] and [25]. [26] employed the transverse eccentricity vector (i.e., the component of the eccentricity vector perpendicular to the chord) and the Newton-Raphson method with numerical derivatives. Analytical derivatives were developed by [27], who addressed the multi-revolution case. The latter investigation was improved by [28] and [29] through simplified formulas for the derivatives.

Solvers with the flight path angle as the unknown were initiated by [30] and extended by [31] to include all orbit types. More recently, [32] treated the case of multiple revolutions, whereas [33] exploited the information of the analytic gradient of the transfer-time equation

with respect to the velocity components to update the iterator. [34] solved the problem via series expansions; [35] iterated on a system of two equations with the energy and eccentric anomaly as unknowns; [36] iterated on the true anomaly; [37] worked with the eccentric anomaly; [38] derived Lambert's problem for rectilinear motion; and [39] was the first to deal with multi-revolution orbits in detail.

The first application of Levi-Civita regularization to the solution of Lambert's problem is due to [2], who also addressed the case of degenerate geometries (most notably, rectilinear motion), and discussed the choice of the initial guess, the derivative of the transfer-time equation as part of the Newton-Raphson iterator, and the issue of numerical degradation. [40] developed an equivalent formulation using Kustaanheimo-Stiefel regularization [41] and gave a recipe for the case of multiple revolutions. [42] solved the problem in the regularized space via optimal control theory.

Several authors have reviewed and compared the algorithms, for example [43–45] and [46]. Some recent comparisons, e.g. [47,48] and [49] include implementations of the algorithm in GPU architectures, analyse the resulting performance and offer comparisons with CPU architectures.

3. Levi-Civita regularization of two-body motion

Regularization is the elimination of singularities occurring in the equations of motion by adopting properly selected variables. Levi-Civita regularization [1,50–52] is a transformation of space and time, and the problem is considered in the plane. The method proceeds through three steps:

1. Introduction of a fictitious time s (note that this is a Sundman transformation [53]) such that:

$$dt = \frac{r}{\tau} ds, \quad (2)$$

being r the distance between the two bodies and τ a constant (in the following its value will be set equal to 1).

2. Definition of a transformation (conformal squaring) between the physical \mathbf{r} -plane and the complex \mathbf{u} -plane (Fig. 2):

$$\mathbf{u}^2 = \mathbf{r}. \quad (3)$$

3. Introduction of the specific mechanical energy -2ρ :

$$-2\rho = \frac{v^2}{2} - \frac{\mu}{r}, \quad (4)$$

μ being the gravitational parameter of the central body and v the relative speed.

The differential equations of motion

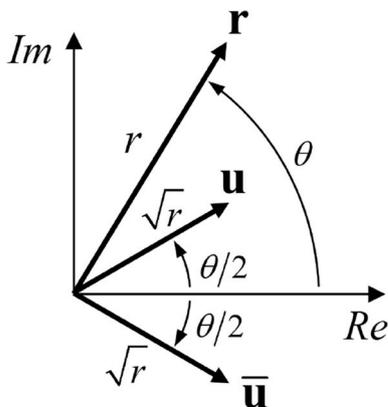


Fig. 2. Conformal squaring in the complex plane.

$$\ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} = 0 \quad (5)$$

can be written in the new space and time variables by applying the rules

$$\frac{d}{dt} = \frac{1}{r} \frac{d}{ds}, \quad (6)$$

$$\frac{d^2}{dt^2} = \frac{1}{r^3} \left(r \frac{d^2}{ds^2} - r' \frac{d}{ds} \right), \quad (7)$$

in which the apostrophe (') denotes differentiation with respect to s . Now¹,

$$r = |\mathbf{u}|^2 = (\mathbf{u}, \mathbf{u}) = \mathbf{u}\bar{\mathbf{u}}, \quad (8)$$

with $\bar{\mathbf{u}}$ being the complex conjugate of \mathbf{u} . Differentiating Eq. (3) with respect to s yields

$$\mathbf{r}' = 2\mathbf{u}\mathbf{u}', \quad (9)$$

$$\mathbf{r}'' = 2(\mathbf{u}\mathbf{u}'' + \mathbf{u}'\mathbf{u}'). \quad (10)$$

Differentiating Eq. (8) provides²

$$\mathbf{r}' = \mathbf{u}'\bar{\mathbf{u}} + \mathbf{u}\bar{\mathbf{u}}' = (\mathbf{u}', \mathbf{u}) + (\mathbf{u}, \mathbf{u}') = 2(\mathbf{u}, \mathbf{u}'). \quad (11)$$

Substitution of Eq. (7) into Eq. (5) gives

$$r\mathbf{r}'' - r'\mathbf{r}' + \mu\mathbf{r} = 0, \quad (12)$$

which, by means of Eqs. (3) and (8)–(11), becomes

$$2(\mathbf{u}, \mathbf{u})(\mathbf{u}\mathbf{u}'' + \mathbf{u}'\mathbf{u}') - 4(\mathbf{u}, \mathbf{u}')\mathbf{u}\mathbf{u}' + \mu\mathbf{u}\mathbf{u} = 0. \quad (13)$$

Expanding the scalar products yields

$$2\mathbf{u}\bar{\mathbf{u}}(\mathbf{u}\mathbf{u}'' + \mathbf{u}'\mathbf{u}') - 2(\mathbf{u}'\bar{\mathbf{u}} + \mathbf{u}\bar{\mathbf{u}}')\mathbf{u}\mathbf{u}' + \mu\mathbf{u}\mathbf{u} = 0, \quad (14)$$

or

$$2\mathbf{u}\bar{\mathbf{u}}\mathbf{u}\mathbf{u}'' - 2\mathbf{u}\bar{\mathbf{u}}'\mathbf{u}\mathbf{u}' + \mu\mathbf{u}\mathbf{u} = 0. \quad (15)$$

The second term can be rewritten as $-2(\mathbf{u}', \mathbf{u}')\mathbf{u}\mathbf{u}$. Dividing both sides of Eq. (15) by \mathbf{u} yields

$$2(\mathbf{u}, \mathbf{u})\mathbf{u}'' - 2(\mathbf{u}', \mathbf{u}')\mathbf{u} + \mu\mathbf{u} = 0. \quad (16)$$

On the other hand, using Eq. (6) in Eq. (4) provides

$$\frac{1}{2r}|\mathbf{r}'|^2 = \mu - 2\rho r, \quad (17)$$

which, thanks to Eqs. (9) and (8), becomes

$$2(\mathbf{u}', \mathbf{u}') = \mu - 2\rho(\mathbf{u}, \mathbf{u}). \quad (18)$$

Finally, substitution of Eq. (18) into Eq. (16) yields

$$\mathbf{u}'' + \rho\mathbf{u} = 0. \quad (19)$$

The transformed differential equations of motion are linear. Given the initial conditions \mathbf{u}_0 and \mathbf{u}'_0 at $s = s_0$, the solution of Eq. (19) is

$$\mathbf{u} = \mathbf{u}_0 \cos[\sqrt{\rho}(s - s_0)] + \mathbf{u}'_0 \frac{1}{\sqrt{\rho}} \sin[\sqrt{\rho}(s - s_0)] \quad \text{if } \rho > 0 \quad (20)$$

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{u}'_0(s - s_0) \quad \text{if } \rho = 0 \quad (21)$$

$$\mathbf{u} = \mathbf{u}_0 \cosh[\sqrt{-\rho}(s - s_0)] + \mathbf{u}'_0 \frac{1}{\sqrt{-\rho}} \sinh[\sqrt{-\rho}(s - s_0)] \quad \text{if } \rho < 0. \quad (22)$$

The three types of conic section (i.e., ellipse, parabola, hyperbola) are all represented and discriminated by the sign of ρ (respectively, positive, null and negative). The solution of Eq. (19) can be written in a more suitable form for the present study by means of a Taylor series

¹ Given the vectors \mathbf{a} and \mathbf{b} , we denote their scalar product by (\mathbf{a}, \mathbf{b}) , whereas $\mathbf{a}\mathbf{b}$ is the product of their complex representations.

² For any pair of complex numbers $\mathbf{a}\bar{\mathbf{b}} + \bar{\mathbf{a}}\mathbf{b} = 2\text{Re}(\mathbf{a}\bar{\mathbf{b}}) = 2|\mathbf{a}||\mathbf{b}|\cos(\arg(\mathbf{a}) - \arg(\mathbf{b})) = 2(\mathbf{a}, \mathbf{b})$

expansion around s_0 , which holds for all three curves (universality):

$$\mathbf{u}(s) = \sum_{k=0}^{\infty} \frac{\mathbf{u}^{(k)}(s_0)}{k!} (s - s_0)^k = \mathbf{u}_0 \sum_{j=0}^{\infty} \frac{(-1)^j \rho^j (s - s_0)^{(2j)}}{(2j)!} + \mathbf{u}'_0 (s - s_0) \sum_{j=0}^{\infty} \frac{(-1)^j \rho^j (s - s_0)^{(2j)}}{(2j + 1)!}. \quad (23)$$

In the second line of Eq. (23), the terms of odd and even order have been separated, and Eq. (19) has been used to rewrite the $(2j)^{\text{th}}$ and $(2j + 1)^{\text{th}}$ derivatives of \mathbf{u} :

$$\mathbf{u}^{(2j)}(s_0) = (-\rho)^j \mathbf{u}_0, \quad (24)$$

$$\mathbf{u}^{(2j+1)}(s_0) = (-\rho)^j \mathbf{u}'_0. \quad (25)$$

The series that appear in Eq. (23) are the Stumpff functions c_0 and c_1 of argument $\rho(s - s_0)^2$ (see Appendix A):

$$\mathbf{u}(s) = \mathbf{u}_0 c_0(\rho(s - s_0)^2) + \mathbf{u}'_0 (s - s_0) c_1(\rho(s - s_0)^2). \quad (26)$$

By a similar reasoning, the first derivative $\mathbf{u}'(s)$ that appears in Eq. (18) can be expressed as

$$\mathbf{u}'(s) = \mathbf{u}'_0 c_0(\rho(s - s_0)^2) - \mathbf{u}_0 \rho (s - s_0) c_1(\rho(s - s_0)^2). \quad (27)$$

4. The transfer-time equation

Given \mathbf{r}_1 and \mathbf{r}_2 and the time $\Delta t = t_2 - t_1$ ($t_1 < t_2$) to transfer from one to the other, two vectors \mathbf{u}_1 and \mathbf{u}_2 are introduced in the complex plane, such that

$$\mathbf{u}_1^2 = \mathbf{r}_1, \quad (28)$$

$$\mathbf{u}_2^2 = \mathbf{r}_2. \quad (29)$$

According to the properties of the square root, the angle between \mathbf{u}_1 and \mathbf{u}_2 is equal to half the angle between the original vectors (Fig. 2), i.e.:

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = \frac{1}{2} \langle \mathbf{r}_1, \mathbf{r}_2 \rangle = \frac{\Delta\theta}{2}. \quad (30)$$

Without loss of generality, the orthogonal reference frame in the complex plane can be oriented with the x -axis parallel to \mathbf{u}_1 . In other words,

$$\mathbf{u}_1 = \begin{pmatrix} A \\ 0 \end{pmatrix}, \quad (31)$$

$$\mathbf{u}_2 = \begin{pmatrix} B \\ C \end{pmatrix}, \quad (32)$$

in which

$$A = \sqrt{r_1}, \quad (33)$$

$$B = \sqrt{r_2} \cos\left(\frac{\Delta\theta}{2}\right), \quad (34)$$

$$C = \sqrt{r_2} \sin\left(\frac{\Delta\theta}{2}\right). \quad (35)$$

If the fictitious time s is zero when the body is at \mathbf{r}_1 , then the total (fictitious) transfer time equals the value of s at \mathbf{r}_2 . Naming the latter quantity s_2 and using Eq. (26) with $s_0 = 0$ and $s = 0$ or $s = s_2$ yields:

$$\mathbf{u}_1 = \mathbf{u}_0 c_0(0), \quad (36)$$

$$\mathbf{u}_2 = \mathbf{u}_0 c_0(\rho s_2^2) + \mathbf{u}'_0 s_2 c_1(\rho s_2^2). \quad (37)$$

Using Eq. (26) to evaluate the scalar product $(\mathbf{u}, \mathbf{u}) = r$ gives

$$r = (\mathbf{u}, \mathbf{u}) = (\mathbf{u}_0, \mathbf{u}_0) c_0^2(\rho s_2^2) + (\mathbf{u}'_0, \mathbf{u}'_0) s_2^2 c_1^2(\rho s_2^2) + 2(\mathbf{u}_0, \mathbf{u}'_0) c_0(\rho s_2^2) s_2 c_1(\rho s_2^2). \quad (38)$$

Defining

$$\mathbf{u}_0 = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}, \quad (39)$$

$$\mathbf{u}'_0 = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}, \quad (40)$$

with

$$\alpha_1 = A, \quad (41)$$

$$\alpha_2 = 0, \quad (42)$$

$$\beta_1 = \frac{B - A c_0(\rho s_2^2)}{s_2 c_1(\rho s_2^2)}, \quad (43)$$

$$\beta_2 = \frac{C}{s_2 c_1(\rho s_2^2)}, \quad (44)$$

the scalar products that appear in Eq. (38) are given by

$$(\mathbf{u}_0, \mathbf{u}_0) = \alpha_1^2 + \alpha_2^2, \quad (45)$$

$$(\mathbf{u}'_0, \mathbf{u}'_0) = \beta_1^2 + \beta_2^2, \quad (46)$$

$$(\mathbf{u}_0, \mathbf{u}'_0) = \alpha_1 \beta_1 + \alpha_2 \beta_2. \quad (47)$$

Integrating Eq. (2) between t_1 and t_2 yields

$$t_2 - t_1 \equiv \Delta t = \int_0^{s_2} r ds = (\alpha_1^2 + \alpha_2^2) \int_0^{s_2} c_0^2(\rho s^2) ds + (\beta_1^2 + \beta_2^2) \int_0^{s_2} s^2 c_1^2(\rho s^2) ds + 2(\alpha_1 \beta_1 + \alpha_2 \beta_2) \int_0^{s_2} c_0(\rho s^2) s c_1(\rho s^2) ds. \quad (48)$$

The relations proved in Appendix A can be employed to manipulate the integrands of Eq. (48) as follows:

$$\Delta t = (\alpha_1^2 + \alpha_2^2) \int_0^{s_2} [1 - 2\rho s^2 c_2(4\rho s^2)] ds + (\beta_1^2 + \beta_2^2) \int_0^{s_2} 2s^2 c_2(4\rho s^2) ds + 2(\alpha_1 \beta_1 + \alpha_2 \beta_2) \int_0^{s_2} s c_1(4\rho s^2) ds. \quad (49)$$

Then, using relation ν) of Appendix A,

$$\Delta t = (\alpha_1^2 + \alpha_2^2) s_2 - (\alpha_1^2 + \alpha_2^2) 2\rho s_2^3 c_3(4\rho s_2^2) + (\beta_1^2 + \beta_2^2) 2s_2^3 c_3(4\rho s_2^2) + 2(\alpha_1 \beta_1 + \alpha_2 \beta_2) s_2^2 c_2(4\rho s_2^2), \quad (50)$$

in which c_2 is the second-order Stumpff function. Substituting Eqs. (33)–(35) gives

$$\Delta t = A^2 s_2 - 2A^2 \rho s_2^3 c_3(4\rho s_2^2) + \frac{[B^2 + C^2 + A^2 c_0^2(\rho s_2^2) - 2ABc_0(\rho s_2^2)] 2s_2 c_3(4\rho s_2^2)}{c_1^2(\rho s_2^2)} + \frac{2A [B - A c_0(\rho s_2^2)] s_2 c_2(4\rho s_2^2)}{c_1(\rho s_2^2)}. \quad (51)$$

Denoting

$$\rho s_2^2 = z \quad (52)$$

and multiplying both sides of Eq. (51) by $c_1^2(z)$, yields the expression of the transfer time for a given geometry (i.e., A, B, C), energy ρ and fictitious time of arrival s_2 :

$$\frac{c_1^2(z)}{s_2} \Delta t = A^2 c_1^2(z) - 2A^2 z c_3(4z) c_1^2(z) + 2B^2 c_3(4z) + 2C^2 c_3(4z) + 2A^2 c_0^2(z) c_3(4z) - 4ABc_0(z) c_3(4z) + 2ABc_1(z) c_2(4z) - 2A^2 c_0(z) c_2(4z) c_1(z). \quad (53)$$

The four terms in A^2 can be further manipulated using the relations $i)$, $ii)$, $iii)$ and Eq. (74) of Appendix A:

$$c_1^2(z) = 2c_2(4z), \quad (54)$$

$$z c_3(4z) c_1^2(z) = 2z c_3(4z) c_2(4z), \quad (55)$$

$$c_0^2(z)c_3(4z) = c_3(4z)[1 - 2zc_2(4z)], \quad (56)$$

$$c_0(z)c_2(4z)c_1(z) = c_2(4z)c_1(4z) = c_2(4z)[1 - 4zc_3(4z)]. \quad (57)$$

Eventually,

$$\frac{c_1^2(z)}{s_2} \Delta t = 2Pc_3(4z) + 2AB [c_1(z)c_2(4z) - 2c_0(z)c_3(4z)], \quad (58)$$

where

$$P = A^2 + B^2 + C^2 \equiv r_1 + r_2. \quad (59)$$

The explicit dependence from s_2 can be eliminated by introducing the following expression for μ

$$\mu = 2\rho \left[\frac{P - 2ABC_0(z)}{zc_1^2(z)} \right], \quad (60)$$

obtained from Eq. (18) with Eqs. (26), (27) and (38) and using the identity

$$c_0^2(z) + zc_1^2(z) = 1, \quad (61)$$

which follows from the formulas *i*) and *ii*) of Appendix A. The substitution

$$Q = 2AB \equiv 2\sqrt{r_1 r_2} \cos\left(\frac{\Delta\theta}{2}\right) \quad (62)$$

provides

$$\Delta t = \frac{2Pc_3(4z) + Q[c_1(z)c_2(4z) - 2c_0(z)c_3(4z)]}{c_1^3(z)} \sqrt{\frac{2P - Qc_0(z)}{\mu}}, \quad (63)$$

in which Δt is determined by the geometry (i.e., P , Q) and by the value of the parameter (i.e., z). Appendix C proves that \sqrt{z} (respectively, $\sqrt{-z}$ in case of hyperbolic motion) is equal to half the variation in eccentric anomaly between P_1 and P_2 on the solution orbit. Note that the parameter z is equal to 1/4 times the homonym free parameter of [11] and [7], even though the underlying formulation is very distinct.

Based on the relationship between the two-body mechanical energy -2ρ and the semimajor axis a of the conic section,

$$a = \frac{\mu}{4\rho}, \quad (64)$$

Eq. (60) allows to express the latter as

$$a = \frac{P - Qc_0(z)}{2zc_1^2(z)}. \quad (65)$$

Note that the sign of the semimajor axis is determined by the sign of z (positive for ellipses, negative for hyperbolas). As a matter of fact,

$$P - Qc_0(z) = r_1 + r_2 - 2\sqrt{r_1 r_2} \cos\left(\frac{\Delta\theta}{2}\right) \begin{cases} \cos\sqrt{z} & \text{if } z > 0 \\ 1 & \text{if } z = 0 \\ \cosh\sqrt{-z} & \text{if } z < 0 \end{cases} \quad (66)$$

It is easy to prove that $P - Qc_0(z)$ is always larger than $(\sqrt{r_1} - \sqrt{r_2})^2$ for $z \geq 0$ and thus positive. When the motion is hyperbolic it might be possible, depending on the geometry, to find a value of z that makes Eq. (66) vanish. Note however that, according to Eq. (63), the corresponding transfer time is zero (motion with infinite velocity). As this limiting case has no physical significance, lower values of z need not be considered. It can therefore be safely assumed that $P - Qc_0(z) > 0$.

Eq. (63) is singularity-free and universal (i.e., applicable to all three types of conic section). It gives the time of flight over arc lengths of up to 2π . Fig. 3 illustrates the dependence of Δt on z for four values of $\Delta\theta$, $r_1 = 1$ NDU and $r_2 = 2$ NDU, with NDU meaning normalized distance units. Based on Kepler's third law, we define the period of a circular orbit of unit radius (1 NDU) around a primary of unit gravitational parameter as being equal to 2π normalized units of time (NTU). The two short-way (i.e., $\Delta\theta < \pi$) transfers are characterized by a time of flight that vanishes for some finite, negative value of z , for which

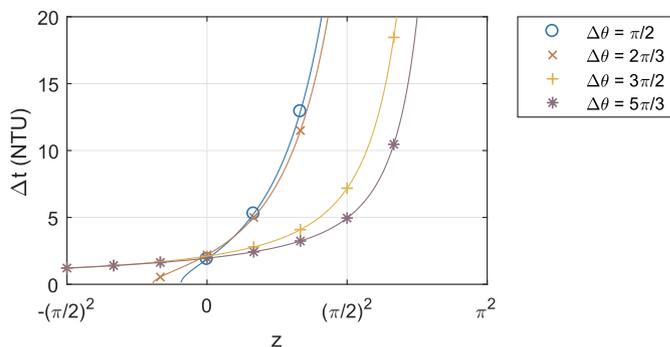


Fig. 3. Transfer time Δt as a function of z for four values of $\Delta\theta$ (two short-way and two long-way trajectories), $r_1 = 1$ NDU and $r_2 = 2$ NDU. For the meaning of the units, the reader is referred to the text.

$P - Qc_0(z)$ is zero. The trajectory is a straight line (the eccentricity of the hyperbola tends to infinity) and the speed is infinite. The two long-way (i.e., $\pi < \Delta\theta < 2\pi$) geometries exhibit a horizontal $t = 0$ asymptote for negative z , instead: the secondary must pass through the pericenter of the solution hyperbola and, even if the eccentricity grows as z decreases, the transfer time cannot become zero (in particular, the expression $P - Qc_0(z)$ is always strictly positive). All transfer-time curves exhibit a vertical asymptote at $z = \pi^2$: as z increases, the transfer arc grows larger, the semimajor axis increases and the eccentricity tends to 1. Hence, at the limit the solution is a parabolic orbit whose $\Delta t = \infty$ because $c_1^3(\pi^2) = 0$ in the denominator of the transfer-time equation.

4.1. Multi-revolution case

The case of elliptical orbits can be extended to account for transfer arcs larger than 2π , i.e., in which the secondary completes an integer number n of revolutions around the primary before arriving at P_2 . The period T of an elliptical orbit of semi-major axis a ,

$$T = 2\pi \sqrt{\frac{a^3}{\mu}}, \quad (67)$$

can be written as a function of the unknown z , as stated in Eq. (65). Thus, the expression of the transfer time is modified by adding n orbital periods. Then, letting n start from 0 yields the most general form of Eq. (63):

$$\Delta t = \frac{2Pc_3(4z) + Q [c_1(z)c_2(4z) - 2c_0(z)c_3(4z)]}{c_1^3(z)} \sqrt{\frac{2P - Qc_0(z)}{\mu}} + \frac{n\pi}{c_1^3(z)} \sqrt{\frac{1}{2\mu} \left[\frac{P - Qc_0(z)}{z} \right]^3}. \quad (68)$$

Fig. 4 illustrates the dependence of Δt on z for a sample of

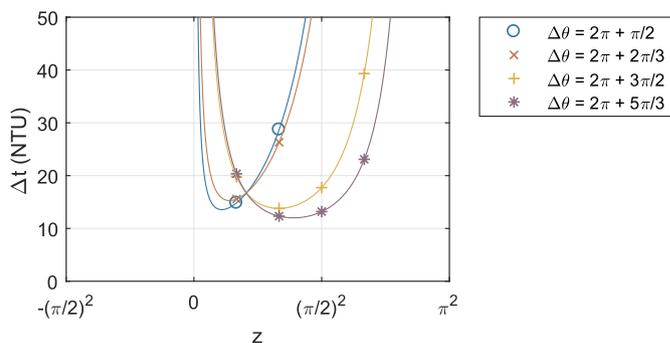


Fig. 4. Transfer time Δt as a function of z for four values of $\Delta\theta$, $r_1 = 1$ NDU, $r_2 = 2$ NDU and $n = 1$. For the meaning of the units, the reader is referred to the text.

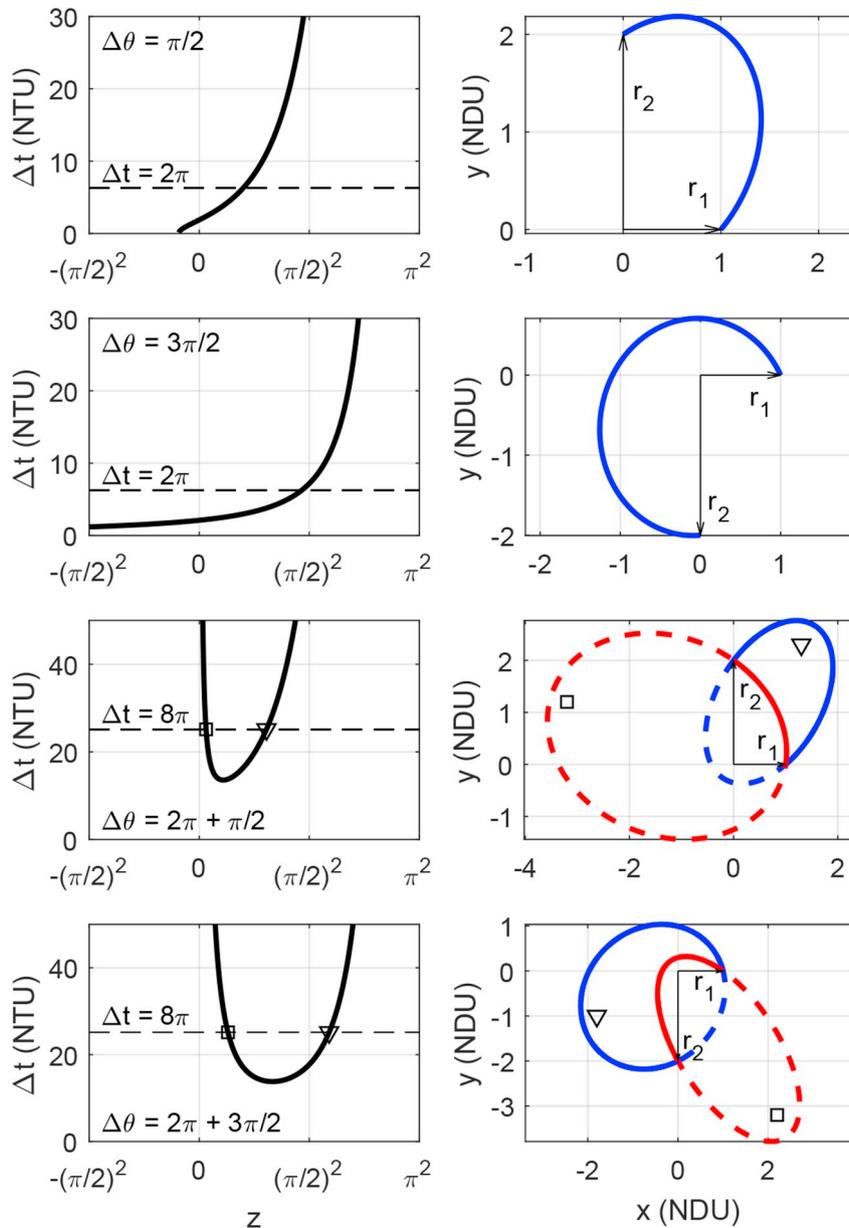


Fig. 5. Examples of transfer-time curves (left column) and specific Lambert's problem solutions (right column) for $n = 0$ (upper two rows) and $n = 1$ (lower two rows). The geometry is represented by the transfer angle (indicated in the top left corner of each row). Lambert's problem is solved for one value of the transfer time (as indicated by the y -coordinate of the horizontal line), giving rise to one ellipse when $n = 0$ and two ellipses when $n = 1$. The correspondence between the value of z in the left subplot and the transfer orbit in the right subplot of each row is indicated by a graphic marker. For the meaning of the units, the reader is referred to the text.

geometries with $n = 1$. In general, for high enough values of Δt , the problem has two solutions, corresponding to two elliptical orbits with short and long period, respectively. The lower half of Fig. 5 shows two such geometries, respectively corresponding to $\Delta\theta = 2\pi + \pi/2$ and $\Delta\theta = 2\pi + 3\pi/2$. The left subplots give the transfer-time curve, whereas the right subplots illustrate the solution orbits. The selected value for the transfer time is 8π , allowing for two trajectories in each case (the correspondence between the value of z and the transfer orbit is indicated by a graphic marker). The transfer-time curve exhibits an absolute minimum because a minimum time is required in order for the secondary to revolve around the primary between the two given positions. At either side of the minimum (respectively, for z tending to 0 and to π^2), the transfer time tends to infinity. In both cases, the eccentricity tends to 1 and the trajectory becomes a parabola: $z = 0$ makes the denominator of the second term of Eq. (68) vanish, whereas when

$z = \pi^2$ the denominators of both terms are zero.

5. From z to the initial value problem

Once z is known, Simó computes the semimajor axis of the orbit with Eq. (65). Next, the eccentricity is obtained with

$$e^2 = 1 - \frac{A^2 C^2}{a^2 z c_1^2(z)}. \quad (69)$$

The remaining orbital elements are computed using a standard procedure (see for example [54]). The velocity \mathbf{v}_1 at \mathbf{r}_1 is finally obtained from the orbital elements and the true anomaly. We followed Simó's procedure for determining the velocity in order to fully document his original work. However, the reader must be aware that there are more efficient ways of obtaining the velocity vector from the z

parameter (see Ref. [11], for example).

6. A regularized Lambert solver

Since the transfer-time equation (Eq. (68)) does not admit a closed-form analytical solution, z must be approximated by some numerical method. Given μ , Δt , r_1 , r_2 , $\Delta\theta$, n and a choice between long- and short-period if $n > 0$, the procedure to solve Lambert's problem by the method presented in the previous sections, called RLSolver hereinafter, is the following:

1. Computation of the geometrical constants P and Q (Eqs. (59) and (62));
2. If $n > 0$, determination of the minimum transfer time Δt^* and the corresponding value z^* of the unknown: if $\Delta t < \Delta t^*$, the problem does not admit solution;
3. Choice of an initial guess z_0 for z ;
4. Approximation of z by application of a numerical solution method to Eq. (68);
5. Computation of \mathbf{v}_1 from the orbital elements and the true anomaly θ_1 .

The value of Δt^* for multi-revolution orbits is approximated by means of the Regula-Falsi iteration scheme, which in the present case means finding the zero of the derivative of Eq. (68) with respect to z . The development of the formula for the derivative is presented in App. B. Higher-order methods were not considered due to the computational cost of the corresponding derivatives of the transfer time equation. The Regula-Falsi method is applied to the initial interval $[0.2\pi^2, 0.4\pi^2]$, which has proved to be a suitable choice for most problems. If the solution is outside the interval, the bounds are iteratively moved towards it. The iteration tolerance is set to 10^{-13} to ensure the accuracy of the minimum value of the transfer time equation (Δt^*), which is critical to determine whether the problem admits solution and to discriminate between the two solution branches. The number of iterations required to confine the solution is typically less than six, but as high as twelve near $r_2/r_1 \sim 1$ or $\Delta\theta \sim 0$. The total number of computations of the derivative is eleven on average, but twice as much near the above limits. An improved Regula-Falsi method has been implemented based on the Illinois method [55], which speeds up the convergence with almost no increase in computing time.

6.1. The initial guess

Five methods to choose the initial guess z_0 have been explored.

- Half-interval point. The initial guess z_0 is the midpoint in the z -domain of each conic in the transfer-time plots. For example, $z_0 = \pi^2$ for single-revolution elliptical orbits. However, in the case of long-way hyperbolic orbits, whose z -domain is unbounded, a lower limit must be provided. To this end, any of the boundary-finding methods discussed below is adequate. The half-interval point approach is computationally very fast, but the initial guess is not very accurate, the errors approach $0.5\pi^2$ near the asymptotes.
- Half-angle point. Following [2], $z_0 = (\Delta\theta/2)^2$ for elliptical orbits and $z_0 = 0$ for hyperbolic orbits. For elliptical orbits, this choice is more accurate than the half-interval point.
- Piecewise linear interpolation. A linear function is used to approximate the transfer time for a given geometry. Hence, the initial guess is the value of z corresponding to Δt on such function. Since the linear approximation is very poor near the asymptote, the z -domain is divided into subintervals, yielding a piecewise linear approximation. Numerical experiments show that the interval $[0, 0.3\pi^2]$ works well for most elliptical orbits, $[-\pi^2, 0]$ is a good choice for

long-way hyperbolic transfers, and $[0.9z_f, 0]$, with $P - Qc_0(z_f) = 0$, is suitable in most short-way hyperbolas. If the solution is outside the z -domain of interpolation, the boundaries are iteratively moved towards the solution. The number of iterations required to confine the solution is lower than four in most cases, and as high as nine near the asymptotes.

- Piecewise nonlinear interpolation. A nonlinear (e.g., quadratic, cubic, exponential) function is used to approximate the transfer time. This option must take into account the computational cost of inverting the function. Compared to the piecewise linear interpolation, the method yields a more accurate initial guess and requires fewer interpolating intervals. In general, five functions are sufficient to cover the z -domain: one for elliptical orbits with $n = 0$, two for hyperbolic transfers (respectively, for long and short ways) and two for the multi-revolution case (respectively, for the short- and long-period solutions), and no iterations are required to confine the solution. Unfortunately, the initial guess can degrade very quickly close to the asymptotes, at times resulting in no convergence at all. This issue can be addressed by dividing the z -domain into smaller subintervals, at the cost of slightly reduced performance.
- Interpolating table. A pre-computed four-dimensional grid of values of z , $\Delta\theta$, r_2/r_1 and Δt is stored and queried during execution. This method is extremely fast within the domain covered by the table, and can be reasonably accurate. An optional trilinear interpolation can be applied to further increase the accuracy of the initial guess at the expense of a slight increase in computing time. However, two tables are required if $n > 0$ (respectively for short- and long-period solutions) and, depending on the resolution of the tables, the size of the arrays may become a limiting factor. Besides, any out-of-range combination of the input parameters can yield very poor initial guesses. Experiments have shown that this approach suffers from poor convergence near the asymptotes. This difficulty can be solved by introducing specific controls and/or resorting to either piecewise interpolation method, at the expense of code simplicity. Since the interpolating table is not an iterative method, it is always faster than the two previous methods.

Usually, an improvement in the initial guess is beneficial, lowering the number of iterations required for convergence. However, an overly complex initial guess search algorithm may increase the total execution time, even when the number of iterations is lower. An appropriate balance among code complexity, number of iterations, convergence domain and total computing time should be sought. Furthermore, the choice of the initial guess search method must take into account the iteration scheme, with some methods (e.g., bisection) working on a set of solution-bracketing bounds whilst others (e.g., high-order Householder methods) requiring a very accurate initial guess. In the present work, each pass of the iterative scheme requires evaluation of the derivative of the transfer time equation (see Sect. 6.2). Since this is a computationally expensive operation, improvements in the initial guess have a very beneficial effect on the algorithm efficiency. The piecewise linear interpolation and the interpolating table are the only strategies that ensure an appropriate initial guess estimation for all geometries. Other types of interpolation may yield incorrect estimations if evaluated out of their interpolation bounds, and static values provide poor initial guesses. The interpolating table, once precomputed, does not involve any complex operations at runtime. Meanwhile, the piecewise linear interpolation scheme is more complex, as it requires searching for the solution bounds at runtime. Therefore, the interpolating table method has been adopted.

6.2. Numerical solution methods

Several numerical methods can be used to approximate the solution

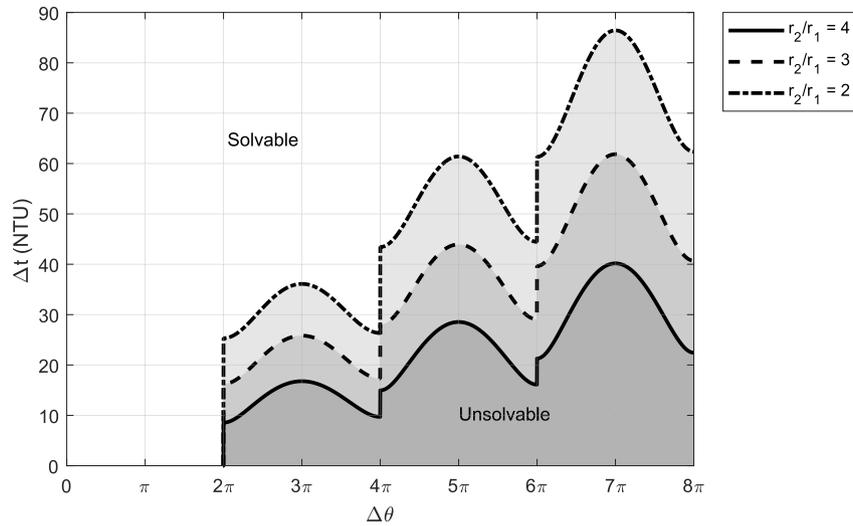


Fig. 6. Space and time domain of applicability of the RLSolver. The single-revolution geometries ($\Delta\theta$ between 0 and 2π) can always be solved, whereas the multi-revolution transfers (here $n = 1, 2, 3$) are solvable only above the minimum-transfer time curve, of which three examples are given, respectively representing ratios r_2/r_1 of 2, 3 and 4. The unsolvable geometries correspond to the shaded areas below each curve.

of Eq. (68). In this work, the following have been considered: bisection, Regula-Falsi (RF) and Newton-Raphson (NR), whereas higher-order Householder methods have been disregarded because of the cost of computing high-order derivatives of the transfer time. The original implementations of RF and NR present convergence problems, so modified versions have been investigated. Eventually, NR with a set of safety checks to avoid out-of-boundary deviations offers the best combination of computational speed and robustness. The resulting MNR (M standing for modified) requires an initial guess and two solution-bracketing boundaries. The method executes NR, which requires the derivative of the transfer time with respect to z (see App. B). At each iteration, MNR checks whether the current z falls between the boundaries, and, if necessary, it brings it back to the closest boundary. Due to the monotonic nature of the transfer-time equation, NR converges from such point. In the case of multi-revolution transfers, z^* and Δt^* are estimated at this stage to distinguish between long- and short-period solutions.

7. Performance tests

RLSolver has been tested to assess:

- the space and time domain of applicability, i.e., the ranges of r_1 , r_2 , $\Delta\theta$ and Δt for which it solves the corresponding Lambert's problems (Sect. 7.2);
- the accuracy of the results (Sect. 7.3);
- the execution time (Sect. 7.4).

The solver performance has been tested against the algorithms of [4] and [5] (hereinafter denoted I15 and G90, respectively).

The output of the computation is the velocity vector \mathbf{v}_1 at the beginning of each transfer. Then, three benchmarks have been designed for testing. The basic benchmark (BB) solves for transfers between $r_1 = 1$ NDU and $r_2 = 2$ NDU ($\mu = 1$), $\Delta\theta$ is varied between 0 and 2π (i.e., $n = 0$) and, for each value of $\Delta\theta$, Δt is varied logarithmically between $2\pi \cdot 10^{-3}$ and $2\pi \cdot 10^3$ NTU. The domains of $\Delta\theta$ and Δt are subdivided into 1000 intervals, yielding one million test points. We also present specific benchmarks for the multi-revolution transfer with $n = 1$ (angles between 2π and 4π). In this case there is no solution for transfer times below the minimum (shaded areas of Fig. 6). The range of Δt explored is from $\Delta t^* + 10^{-9}$ to $\Delta t^* + 10^3$. The benchmark named BS computes the short transfer, while test BL corresponds to the long trajectory. Prior to the execution of the tests, RLSolver has undergone an assessment of its

convergence properties (Sect. 7.1).

7.1. Convergence tolerance

The machine precision level sets the theoretical upper limit to the numerical accuracy of the output of a computation, 15 significant figures for double floating-point precision. In general, the accuracy of the solution (i.e., z) depends on the number of iterations performed, and this, in turn, depends on the tolerance chosen for convergence. In RLSolver, like in the majority of Lambert's solvers, this tolerance is applied to the transfer time. Below a certain threshold, increasing the number of iterations does not bring a better accuracy because the machine precision has been reached³. This threshold can be determined by repeatedly solving the same problem for decreasing values of the convergence tolerance: the accuracy of the results does not increase indefinitely, rather, it reaches an upper limit. The corresponding tolerance is the threshold sought. The study has been carried out on all the points of BB, BL and BS using convergence tolerances from 1 to 10^{-16} at steps of one order of magnitude. In the vast majority of cases, the threshold convergence tolerance is reached in four to six iterations. For the points of BL and BS close to the minimum transfer time, 20 Newton-Raphson iterations or more are required due to the small slope of the function. Some hyperbolic orbits with low values of time of flight and near-parabolic trajectories require nine to ten iterations. Such situations, however, are very unlikely in realistic mission scenarios. Setting the convergence tolerance at its threshold value allows to obtain the best possible accuracy with the minimum number of iterations. Note that the accuracy level chosen (machine precision basically) is not required for many practical applications. However, this allows us to make an unbiased comparison of the accuracy of the different solvers.

7.2. Space and time domain of applicability

We successfully tested RLSolver over a wide range of times of flight, number of revolutions (n) and r_2/r_1 ratios. It can handle, for example, the cases $\Delta t = 0$ (infinitely fast rectilinear trajectory), $r_1 = r_2$ and $r_2 = 0$ ($P_2 \equiv F$). Table 1 reports results for other potentially conflictive cases.

³ or because the solution lies in the highly inclined part of the transfer-time equation (close to the vertical asymptote), where two different Δt s may correspond to extremely close values of z , hence iterating further does not cause any appreciable variation in the solution.

Table 1

Limit and degenerate cases: orbit type (first column), transfer angle (second column), value of z (third column), number of iterations (fourth column), x - and y -components of the velocity vectors at the beginning (v_1) and at the end (v_2) of the transfer (fifth and sixth columns, respectively).

Orbit type	$\Delta\theta$	z	Iterations	v_{1x}, v_{1y}	v_{2x}, v_{2y}
Elliptic	0	1.828	5	1.096, 0.000	0.449, 0.000
Elliptic	π	2.674	4	0.053, 1.155	0.053, -0.577
Elliptic	2π	7.114	5	1.067, 0.000	0.371, 0.000
Parabolic	0	0.000	1	1.414, 0.000	1.000, 0.000
Parabolic	π	0.000	1	-0.816, 1.155	-0.816, -0.577
Hyperbolic	0	-0.103	5	3.279, 0.000	3.123, 0.000
Hyperbolic	π	-9.822	3	-9.393, 1.155	-9.393, -0.577

Table 2

Average execution times of RLSolver, G90 and I15 on the BB (first column), BS (second column) and BL (third column), in microseconds.

Algorithm	BB	BS	BL
RLSolver	1.16	5.14	5.30
G90	0.51	1.77	1.76
I15	0.38	1.41	1.30

These include parabolas, highly-eccentric hyperbolic orbits and undefined transfer planes ($\Delta\theta = 0, \pi, 2\pi$).

7.3. Accuracy

The convergence tolerance study has been carried out also on I15.⁴ At the respective threshold tolerance, the results provided by I15 and RLSolver have been compared on all the Lambert's problems of BB, BL and BS by evaluating the relative difference RD of the velocity vector:

$$RD = \frac{\sqrt{(\mathbf{v}_1^{RLSolver} - \mathbf{v}_1^{I15}) \cdot (\mathbf{v}_1^{RLSolver} - \mathbf{v}_1^{I15})}}{v_1^{I15}}. \quad (70)$$

The maps of RD for the three benchmarks are provided in Fig. 7: everywhere RD is close to the double floating-point precision, except for some hyperbolic orbits with very low Δt and multi-revolution transfers close to Δt^* for which RD increases to 10^{-11} . To explore a wider parameter space, the tests have been repeated for $r_1/r_2 = \{0.5, 2.0, 4.0\}$ and $n = \{2,4\}$. No significant changes in accuracy have been observed.

7.4. Execution speed

On a typical five-iteration execution, the time is distributed as follows: for $n = 0$ problems, 45% is spent on the evaluation of the transfer-time equation and its derivative in the time-convergence loop, 25% is required to compute v_1 from z , 15% is employed to determine the initial guess (including the query to the interpolating table and the associated safety checks) and the rest is used by auxiliary tasks (essentially constants assignments, memory allocation, problem geometry definition, out-of-bounds checks and convergence criteria, and the computation of Eqs. (65) and (69) and the rest of orbital parameters). For $n > 0$, 60% of the execution time is consumed in finding Δt^* (with an average of eleven evaluations of the transfer time equation derivative), 24% in the evaluation of the transfer-time equation and its derivative on the MNR iterator, 6% is spent on v_1 , 3% on the initial guess, and the rest on the auxiliary tasks. In the transfer-time computation routine, roughly one third of the time goes into the calculation of the Stumpff coefficients.

⁴The accuracy characteristics of I15 and G90 are similar, therefore the comparison has been made against I15 only.

The routine for the transfer-time derivative has been optimized to include both the evaluation of the transfer time and its derivative in the same call, so that its execution is only 1.7 times longer than the basic transfer-time routine. The above time distribution can be easily scaled to account for a different number of iterations.

A simulation has been performed on the three scenarios (BB, BS and BL) to estimate the average execution times. The algorithms are run with a tolerance of 10^{-9} which corresponds to an average of five iterations for RLSolver. The simulations have been repeated 100 times in order to obtain meaningful values for the average execution time, yielding a total of 10^8 evaluations on each test scenario. The three methods (RLSolver, G90 and I15) have been programmed in Fortran language. The executables have been built with Intel Fortran 19.0.0.117 with optimization level -O3, under Ubuntu 18.04.1 LTS. The test system sports an Intel Xeon E5-2630 v3 processor at 2.40 GHz with 24 GB of RAM. Table 2 compares the performance of the three methods. I15 (the clear winner) is between three and four times faster than RLSolver. While considerably slower than I15, RLSolver performance remains quite acceptable for many applications. For example, solving a 1000 by 1000 porkchop plot takes only 1.2 s.

8. Discussion and conclusions

This paper revisits the solution of Lambert's problem via Levi-Civita regularization, a method developed by Ref. [2] in which the transfer time for a given geometry is expressed through a quantity, called z , proportional to the square of the eccentric anomaly difference between the end-points of the transfer on the solution curve. The Stumpff functions of argument z that appear in the transfer-time expression arise from the solution of the regularized two-body equation of motion. This is the equation of the simple harmonic oscillator in which the constant coefficient is the mechanical energy, a quantity that discriminates the three types of conic section and, thus, makes the method universal.

Here, the theory of Levi-Civita regularization applied to Lambert's problem is demonstrated with full details including the explanation of the geometrical meaning of z , the Proof of the relevant properties of the Stumpff functions and the transcription of the problem statement in the regularized space. We improve on the original formulation by completely solving the multi-revolution ($n > 0$) case for the first time in this context, an upgrade that enables the application of the algorithm to modern problems, such as the design of multi-revolution trajectories among the moons of the giant planets or to the inner solar system. The features of the transfer-time equation are illustrated, the geometry of the solutions (e.g., short-, long-way) is discussed, the multiple conic sections (short-, long-period) arising when $n > 0$ are presented and the degeneracies that the method can accommodate are analysed. After investigating a range of initial guess search techniques and testing different numerical methods to approximate the solution, we propose a solution procedure in which the initial guess is assigned very efficiently by querying a pre-defined interpolation table. Then, in order to achieve both speed and robustness, we combine Newton-Raphson with safety checks to avoid out of boundary deviations to approximate z . To facilitate the reproduction of our results, we provide the complete algorithmic recipe for the computation of the derivative of the transfer time with respect to z .

Nowadays, the solution of the Lambert's problem is key in Astrodynamics. Its most common applications are the computation of porkchop plots (direct transfers between planets) and the optimization of, e.g., multi-gravity assist interplanetary trajectories. In all these cases, the solution of the two-body orbital boundary value problem must be calculated repeatedly in an automatic search. Accuracy and execution speed are certainly important performance features, but the capability to deal with limit cases and degenerate geometries without failure is also crucial. The method here discussed has been coded in Fortran and tested extensively in order to quantify, in the first place, its domain of applicability, i.e., the geometries and transfer times for

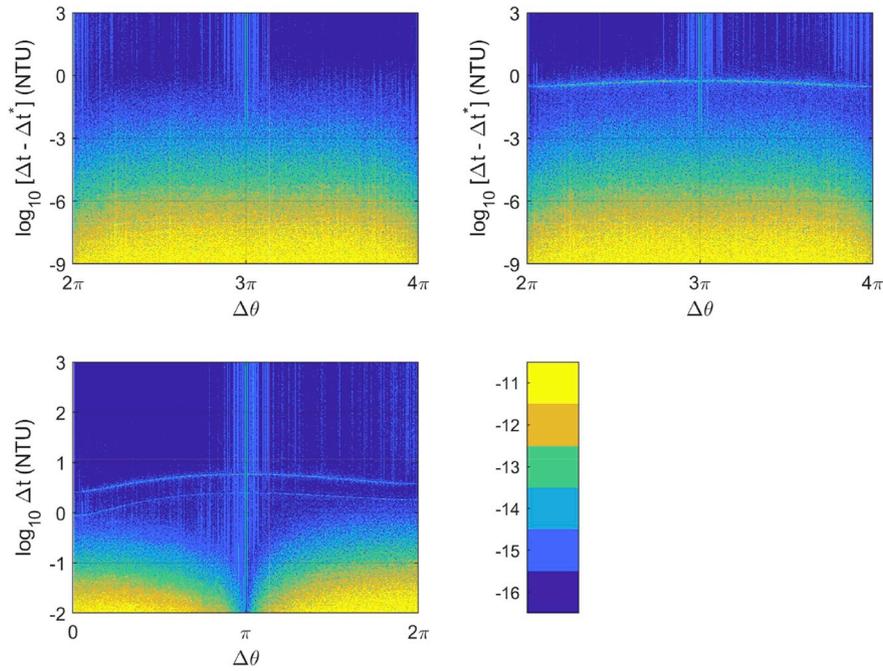


Fig. 7. Maps of $\log_{10}(RD)$, the relative difference between the results of RLSolver and those of I15 obtained at their respective convergence tolerance thresholds. The three maps correspond to Lambert's problems of BB (bottom), BS (top-left) and BL (top-right), respectively.

which it can approximate z and solve the corresponding initial value problem. Then, once its convergence properties have been understood and appropriately tuned, the accuracy of the results has been compared with that of Izzo's algorithm [4], thus obtaining an external validation of the present method. Execution speed assessments yielded acceptable results, especially after extrapolation to the computation of a porkchop plot of medium size. The higher algebraic complexity of the regularized transfer-time equation over the equivalent formulations by [4] and [5] results in longer execution times (a factor of four higher in the worst case). Note, however, that Simo's method offers room for improvement (e.g., in the computation of velocities, as indicated in Sect. 5), so

the difference can be reduced.

In conclusion, the method meets all the requirements of modern applications and is an interesting alternative to state-of-the-art techniques such as those developed by Izzo and Gooding that are world-wide rated as the standard Lambert solvers.

Acknowledgements

The authors wish to thank Prof. Carles Simó for his inestimable suggestions.

Appendix A. Stumpff functions

Karl Stumpff (1895–1970) introduced a special family of functions that appear frequently in Astrodynamics. They were later called the Stumpff functions, and they are defined in terms of the series (see also [7], Chapt. 4)

$$c_k(z) = \frac{1}{k!} - \frac{z}{(k+2)!} + \frac{z^2}{(k+4)!} + \dots = \sum_{i=0}^{\infty} \frac{(-1)^i z^i}{(k+2i)!}, \quad k = 0, 1, 2, 3, \dots \quad (71)$$

The series is absolutely convergent for all real values of z . The degree k of the series is a positive integer number. The first two Stumpff functions, namely $c_0(z)$ and $c_1(z)$, admit simple closed-form expressions:

$$c_0(z) = \begin{cases} \cos\sqrt{z} & \text{if } z > 0 \\ 1 & \text{if } z = 0 \\ \cosh\sqrt{-z} & \text{if } z < 0 \end{cases} \quad (72)$$

$$c_1(z) = \begin{cases} \frac{\sin\sqrt{z}}{\sqrt{z}} & \text{if } z > 0 \\ 1 & \text{if } z = 0 \\ \frac{\sinh\sqrt{-z}}{\sqrt{-z}} & \text{if } z < 0 \end{cases} \quad (73)$$

The Stumpff functions of higher degree can be obtained by means of the recursive formula:

$$z c_{k+2}(z) = \frac{1}{k!} - c_k(z). \quad (74)$$

The Proof of which follows from the straightforward application of the definition (Eq. (71)). Four properties of the Stumpff functions are relevant for the present study. Note that the first three are proved here for $z > 0$ (the case $z < 0$ exploits equivalent properties of hyperbolic functions, whereas the case $z = 0$ is trivial).

i) $c_0^2(z) = 1 - 2zc_2(4z)$

Proof Defining $z = y^2$ allows to write

$$c_0(4z) = c_0(4y^2) = \cos\sqrt{4y^2} = 2\cos^2\sqrt{y^2} - 1 = 2c_0^2(y^2) - 1 = 2c_0^2(z) - 1, \quad (75)$$

where use has been made of the well-known trigonometric identity $\cos 2y = 2\cos^2y - 1$. Then, by applying Eq. (74), one can write

$$zc_2(z) = 1 - c_0(z), \quad (76)$$

$$4zc_2(4z) = 1 - c_0(4z). \quad (77)$$

Hence,

$$4zc_2(4z) = 1 - 2c_0^2(z) + 1. \quad (78)$$

ii) $c_1^2(z) = 2c_2(4z)$

Proof Application of Eq. (74) to $k = 0$ provides $c_2(z) = \frac{1 - \cos\sqrt{z}}{z}$. Then, by recalling that $2\sin^2z = 1 - \cos 2z$,

$$c_1^2(z) = \frac{\sin^2\sqrt{z}}{\sqrt{z}^2} = \frac{1 - \cos 2\sqrt{z}}{2z} = 2\frac{1 - \cos\sqrt{4z}}{4z} = 2c_2(4z). \quad (79)$$

iii) $c_0(z)c_1(z) = c_1(4z)$

Proof The trigonometric identity $2\sin u \cos v = \sin(u + v) + \sin(u - v)$ can be applied to the product $c_0(z)c_1(z)$:

$$c_0(z)c_1(z) = \frac{\sin\sqrt{z}\cos\sqrt{z}}{\sqrt{z}} = \frac{\sin 2\sqrt{z}}{2\sqrt{z}} = \frac{\sin\sqrt{4z}}{\sqrt{4z}} = c_1(4z). \quad (80)$$

iv) $\frac{dc_k(z)}{dz} = \frac{kc_{k+2}(z) - c_{k+1}(z)}{2}$ [40]

Proof Differentiation of Eq. (71) with respect to the argument provides

$$\frac{dc_k(z)}{dz} = \sum_{j=1}^{\infty} \frac{(-1)^j j z^{j-1}}{(k+2j)!}. \quad (81)$$

On the other hand,

$$kc_{k+2}(z) - c_{k+1}(z) = \sum_{i=0}^{\infty} \frac{k(-1)^i z^i}{(k+2+2i)!} - \sum_{i=0}^{\infty} \frac{(-1)^i z^i}{(k+1+2i)!} = \sum_{i=0}^{\infty} \frac{[k - (k+2+2i)](-1)^i z^i}{(k+2+2i)!} = 2 \sum_{i=0}^{\infty} \frac{(i+1)(-1)^{i+1} z^i}{(k+2+2i)!} \equiv 2 \frac{dc_k(z)}{dz}. \quad (82)$$

v) $\int s^k c_k(\rho s^2) ds = s^{k+1} c_{k+1}(\rho s^2)$

Proof:

$$\int s^k c_k(\rho s^2) ds = \int \sum_{i=0}^{\infty} \frac{(-1)^i \rho^i s^{k+2i}}{(k+2i)!} ds = \sum_{i=0}^{\infty} \frac{(-1)^i \rho^i s^{k+2i+1}}{(k+1+2i)(k+2i)!} = s^{k+1} \sum_{i=0}^{\infty} \frac{(-1)^i \rho^i s^{2i}}{(k+1+2i)!} \equiv s^{k+1} c_{k+1}(\rho s^2). \quad (83)$$

Appendix B. Derivative of the transfer time with respect to z

In order to apply the chain rule of differentiation to Eq. (68), it is convenient to group terms in its right-hand side in the following way:

$$\Delta t = \frac{RT}{S} + U, \quad (84)$$

where

$$R = 2Pc_3(4z) + Q [c_1(z)c_2(4z) - 2c_0(z)c_3(4z)], \quad (85)$$

$$S = c_1^3(z), \quad (86)$$

$$T = \sqrt{2 \frac{P - Qc_0(z)}{\mu}}, \quad (87)$$

$$U = 2\pi n \sqrt{\frac{1}{\mu} \left[\frac{P - Qc_0(z)}{2zc_1^2(z)} \right]^3}. \quad (88)$$

The first derivative of Eq. (84) is:

$$\frac{d\Delta t}{dz} = \frac{T}{S} \frac{dR}{dz} - \frac{RT}{S^2} \frac{dS}{dz} + \frac{R}{S} \frac{dT}{dz} + \frac{dU}{dz}, \quad (89)$$

where

$$\frac{dR}{dz} = 2P \frac{dc_3(4z)}{dz} + Q \left[c_2(4z) \frac{dc_1(z)}{dz} + c_1(z) \frac{dc_2(4z)}{dz} - 2c_3(4z) \frac{dc_0(z)}{dz} - 2c_0(z) \frac{dc_3(4z)}{dz} \right], \quad (90)$$

$$\frac{dS}{dz} = 3c_1^2(z) \frac{dc_1(z)}{dz}, \quad (91)$$

$$\frac{dT}{dz} = -\frac{Q}{\sqrt{2\mu V}} \frac{dc_0(z)}{dz}, \quad (92)$$

$$\frac{dU}{dz} = -\frac{3n\pi}{2z^2 c_1^4(z)} \sqrt{\frac{V}{2z\mu}} \left[zc_1(z)Q \frac{dc_0(z)}{dz} + Vc_1(z) + 2zV \frac{dc_1(z)}{dz} \right]. \quad (93)$$

In Eqs. (92) and (93) $V = P - Qc_0(z)$. The derivatives of the Stumpff functions with respect to the argument can be computed by means of the recursive formula iv) proved in Appendix A. In particular,

$$\frac{dc_0(z)}{dz} = -\frac{c_1(z)}{2}, \quad (94)$$

$$\frac{dc_1(z)}{dz} = \frac{c_3(z) - c_2(z)}{2}, \quad (95)$$

$$\frac{dc_2(4z)}{dz} = 2[2c_4(4z) - c_3(4z)], \quad (96)$$

$$\frac{dc_3(4z)}{dz} = 2[3c_5(4z) - c_4(4z)]. \quad (97)$$

Appendix C. Geometrical meaning of z

Elliptical orbits

If M denotes the mean anomaly, E the eccentric anomaly and e the eccentricity, differentiation of Kepler's equation $M = E - e \sin E$ with respect to time t yields

$$\dot{M} \equiv \sqrt{\frac{\mu}{a^3}} = \dot{E}(1 - e \cos E). \quad (98)$$

Hence,

$$\frac{dt}{dE} = (1 - e \cos E) \sqrt{\frac{a^3}{\mu}} = r \sqrt{\frac{a}{\mu}} \quad (99)$$

because $r = a(1 - e \cos E)$. Eventually, after recalling that in the ellipse $2\rho = \mu/(2a)$,

$$dt = \frac{r}{2\sqrt{\rho}} dE. \quad (100)$$

The substitution of Eq. (2) with $\tau = 1$ yields

$$dE = 2\sqrt{\rho} ds. \quad (101)$$

Integration between $s_0 = 0$ and s_2 gives

$$\frac{E_2 - E_0}{2} \equiv \frac{\Delta E}{2} = \sqrt{\rho} s_2 \equiv \sqrt{z}. \quad (102)$$

Hyperbolic orbits

A similar reasoning applied to the equivalent form of Kepler's equation, i.e., $M = e \sinh H - H$ with H the eccentric anomaly, leads to

$$\frac{H_2 - H_0}{2} \equiv \frac{\Delta H}{2} = \sqrt{-\rho} s_2 \equiv \sqrt{-z}. \quad (103)$$

Parabolic orbits

This is a limit case: $z = 0$ because $\rho = 0$.

References

- [1] T. Levi-Civita, Sur la résolution qualitative du problème restreint des trois corps, *Acta Math.* 30 (1906) 305–327.
- [2] C. Simó, Solución del problema de Lambert mediante regularización, *Collect. Math.* 24/3 (1973) 231–247.
- [3] J.V. McAdams, R.W. Farquhar, A.H. Taylor, B.G. Williams, MESSENGER mission design and navigation, *Space Sci. Rev.* 131 (1–4) (2007) 219–246.
- [4] D. Izzo, Revisiting Lambert's problem, *Celest. Mech. Dyn. Astron.* 121 (2015) 1–15.
- [5] R.H. Gooding, A procedure for the solution of Lambert's orbital boundary-value problem, *Celest. Mech. Dyn. Astron.* 48 (1990) 145–165.
- [6] C.F. Gauss, *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*, (1809).
- [7] R.H. Battin, *An introduction to the mathematics and methods of astrodynamics*, AIAA Education Series, Reston, Virginia, 2001 ISBN: 1-56347-342-9.
- [8] L.A. Loechler, *An Elegant Lambert Algorithm for Multiple Revolution Orbits*, M.Sc. Thesis Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988.

- [9] H. Shen, P. Tsiotras, Using battin's method to obtain multiple-revolution Lambert's solutions, *Adv. Astronaut. Sci.* 757 (2004) 1–18.
- [10] J. Ahn, J. Bang, S.I. Lee, Acceleration of zero-revolution Lambert's algorithms using table-based initialization, *J. Guid. Contr. Dynam.* 38 (2015) 335–342.
- [11] R.R. Bate, D.D. Mueller, J.E. White, *Fundamentals of Astrodynamics*, Dover Publications, New York, 1971 ISBN: 0-486-60061-0.
- [12] Q. Luo, Z. Meng, C. Han, Solution algorithm of a quasi-Lambert's problem with fixed flight-direction angle constraint, *Celest. Mech. Dyn. Astron.* 109 (2011) 409–427.
- [13] N. Arora, R.P. Russell, A fast and robust multiple revolution Lambert algorithm using a cosine transformation, *AAS/AIAA Astrodynamics Specialist Conference*, 11–15 August, vol. 13, Hilton Head Island, South Carolina, 2013, p. 728.
- [14] R.C. Blanchard, E.R. Lancaster, A Unified Form of Lambert's Theorem. NASA TN D-5368, (1969).
- [15] J.L. Lagrange, *Mécanique Analytique* vol. II, (1788).
- [16] S.I. Ochoa, J.E. Prussing, Multiple revolution solutions to Lambert's problem, *Adv. Astronaut. Sci.* 79 (1992) 1205–1216.
- [17] J.E. Prussing, A class of optimal two-impulse rendezvous using multiple-revolution Lambert solutions, *J. Astronaut. Sci.* 48 (2000) 131–148.
- [18] S.E. Wailliez, On Lambert's problem and the elliptic time of flight equation: a simple semi-analytical inversion method, *Adv. Space Res.* 53 (2014) 890–898.
- [19] J.W. McMahon, D.J. Scheeres, Linearized Lambert's problem solution, *J. Guid. Contr. Dynam.* 39 (2016) 2205–2218.
- [20] J.D. Thorne, Convergence behavior of series solutions of the Lambert problem, *J. Guid. Contr. Dynam.* 8 (2014) 1–6.
- [21] V.E. Usachov, A.S. Ezhov, A.V. Simonov, Optimization of interplanetary trajectories for flights to neighboring circumsolar space, *Sol. Syst. Res.* 47 (2013) 528–534.
- [22] L.I. Jiali, C. Tao, W. Songyan, M.A. Ping, Graphic semi-major axis iteration algorithm for Lambert problem, *Proceedings of the 34th Chinese Control Conference*, Hangzhou, China, 28–30 July, 2015, pp. 5277–5281.
- [23] R. Jiang, T. Chao, S. Wang, M. Yang, Improved semi-major Axis iterated method for Lambert's problem, *Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference*, 12–14 August, Nanjing, China, 2016, pp. 1423–1428.
- [24] S. Herrick, A. Liu, Two Body Orbital Determination from Two Positions and Time of Flight, (1959) Appendix A, *Aeronutronic Publication No. C-365*.
- [25] F.W. Boltz, Second-order p-iterative solution of the Lambert/Gauss problem, *J. Astronaut. Sci.* 32 (1984) 475–485.
- [26] G. Avanzini, A simple Lambert algorithm, *J. Guid. Contr. Dynam.* 31 (2008) 1587–1594.
- [27] Q. He, J. Li, C. Han, Multiple-revolution solutions of the transverse-eccentricity-based Lambert problem, *J. Guid. Contr. Dynam.* 33 (2010) 265–269.
- [28] G. Zhang, D. Zhou, D. Mortari, Optimal two-impulse rendezvous using constrained multiple-revolution Lambert solutions, *Celest. Mech. Dyn. Astron.* 110 (2011) 305–317.
- [29] C. Wen, Y. Zhao, P. Shi, Derivative analysis and algorithm modification of transverse-eccentricity-based Lambert problem, *J. Guid. Contr. Dynam.* 37 (2014) 1195–1201.
- [30] A.D. Wheelon, Free flight of a ballistic missile, *Am. Rocket Soc. J.* 29 (1959) 915–926.
- [31] S.L. Nelson, P. Zarchan, Alternative approach to the solution of Lambert's problem, *J. Guid. Contr. Dynam.* 15 (1992) 1003–1009.
- [32] P.V. Arlulkar, S.D. Naik, Solution based on dynamical approach for multiple-revolution Lambert problem, *J. Guid. Contr. Dynam.* 34 (2011) 920–923.
- [33] J. Ahn, S.I. Lee, Lambert algorithm using analytic gradients, *J. Guid. Contr. Dynam.* 36 (2013) 1751–1761.
- [34] G.S. Gedeon, A practical note on the use of Lambert's equation, *AIAA J.* 3 (1) (1965) 149–150.
- [35] E.T. Pitkin, A general solution of the Lambert problem, *J. Astronaut. Sci.* 15 (1968) 270.
- [36] L.G. Taff, M.S. Randall, Two locations, two times, and the element set, *Celestial Mech.* 37 (1985) 149–159.
- [37] F.T. Sun, N.X. Vinh, T. Chen, Analytic study of the solution families of the extended Godal's time equation for Lambert's problem, *Astrodynamics* 1985 (1986) 815.
- [38] A.J. Sarnecki, Minimal orbital dynamics, *Acta Astronaut.* 17 (1988) 881–891.
- [39] F.T. Sun, On the minimum time trajectory and multiple solutions of Lambert's problem, *American Institute of Aeronautics and Astronautics Conference*, Paper 79-164, 1979, p. 26.
- [40] J. Kriz, A uniform solution of the Lambert problem, *Celestial Mech.* 4 (1976) 509–513.
- [41] P. Kustaanheimo, E. Stiefel, Perturbation theory of Kepler motion based on spinor regularization, *Journal für die reine und angewandte Mathematik* 218 (1965) 204–219.
- [42] D.J. Jezewski, K/S two-point-boundary-value problems, *Celestial Mech.* 14 (1976) 105–111.
- [43] A.T. Monuki, Derivation of Z-Iterator and Comparison of Lambert Routines, TRW Systems Group, 7121.3–173, 1973, pp. 1–22.
- [44] A.R. Klumpp, Universal Lambert and Kepler algorithms for autonomous rendezvous, *Astrodynamics* 1989 (1990) 1–24.
- [45] G.E. Peterson, E.T. Campbell, A.M. Hawkins, et al., Relative performance of Lambert solvers 1: zero revolution methods, *Adv. Astronaut. Sci.* 136 (1991) 1495–1510.
- [46] S.I. Lee, J. Ahn, J. Bang, Dynamic selection of zero-revolution Lambert algorithms using performance comparison map, *Aero. Sci. Technol.* 51 (2016) 96–105.
- [47] N. Arora, R.P. Russell, A GPU accelerated multiple revolution Lambert solver for fast mission design, *Adv. Astronaut. Sci.* 136 (2010) 1477–1494.
- [48] N. Parrish, Accelerating Lambert's Problem on the GPU in MATLAB, B.Sc. Thesis California Polytechnic State University, San Luis Obispo, California, 2012.
- [49] S. Wagner, B. Wie, B. Kaplinger, Computational solutions to Lambert's problem on modern graphics processing units, *J. Guid. Contr. Dynam.* 38 (2015) 1–6.
- [50] V. Szebehely, *Theory of Orbits*, Academic Press, New York and London, 1967.
- [51] A. Celletti, B.A. Steves, A.J. Maciejewski, M. Hendry (Eds.), *Basics Of Regularization Theory. Chaotic Worlds: from Order to Disorder in Gravitational N-body Dynamical Systems*, 2006, pp. 203–225.
- [52] J. Waldvogel, *Fundamentals of regularization in celestial mechanics and linear perturbation theories*, Scottish Universities Summer School in Physics, May 28 - June 8, Sabhal Mor Ostaig, Scotland, 2007.
- [53] K.F. Sundman, Mémoire sur le Problème des trois corps, *Acta Math.* 36 (1912) 105–179.
- [54] A.E. Roy, *Orbital Motion*, fourth ed., Institute of Physics Publishing, Bristol, UK, 1967 ISBN: 0-7503-1015-6.
- [55] J.F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.