A CRITICAL REVIEW ON THE STATE-OF-THE-ART OF LAMBERT'S PROBLEM SOLVERS

Jorge Martínez Garrido(1) **and Manuel Sanjurjo Rivo**(2)

(1)*Undergraduate Aerospace Engineer, jorge@on-orbit.dev*
(2)*PhD Aerospace Engineer, msanjurj@ing.uc3m.es*

Universidad Carlos III de Madrid, Av. de la Universidad, 30, 28911 Leganés, Madrid

ABSTRACT

This work presents a deep comparison between some of the latest Lambert's problem solvers from both an analytical and a performance point of view. Each algorithm is analyzed considering its iteration variable, the numerical root finder used, the initial guess and the final velocity vectors construction. The performance results are provided in the form of contour figures relating the required iterations for a particular combination of the non-dimensional time of flight and transfer angle. In addition, the time required per iteration and total one are included to retrieve a full insight of algorithm's overall performance.

1 INTRODUCTION

The boundary value problem (BVP) in the context of the restricted two-body dynamics is known as the Lambert's problem. It states to find for the orbit which connects two known position vectors, $\vec{r_1}$ and $\vec{r_2}$, being known the time of flight between them, Δt, under gravitational field of strength μ.

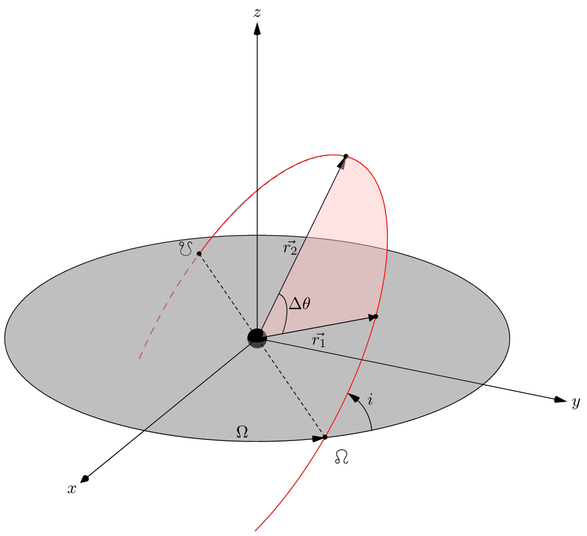

Figure 1: Geometry of the problem as seen from an inertial frame centered in the attractor body.

The problem was originally posed by Johann Heinrich Lambert, in a letter sent to Leonhard Euler, after publishing his book [1] about comets' orbits and their properties. Since its formulation, many popular scientific figures, such us Lagrange or Gauss, have addressed this famous astrodynamics problem, devising a lot of different approaches to solve for it.

Lambert's problem (sometimes referred to as Euler-Lambert or Gauss problem one) gained popularity during the 60's due to its applications in inter-planetary trajectory design, mission analysis and intercepting maneuvers. Since this decade, a plethora of solutions has been devised in the form of computer algorithms. In fact, the problem still remains of interest, as dozens of articles about it have been published over the last 20 years. Therefore, because of the amount of available solvers, a comparison is needed in order to identify which ones perform the best under given conditions.

These kind of comparisons were previously carried out by Klumpp [2] in the 90s and De la Torre

[3] more recently. Therefore, the goal of this report is to expand these analysis by including some of the latest published algorithms, Avanzini 2008 [4] and Arora 2013 [5], and compare them against the most robust and accurate ones: Gooding 1990 [6] and Izzo 2015 [7].

All the solvers presented in this report, together with the performance comparison framework, have been exposed as a Python package under the name of *lamberthub*[1]. This will be useful for future authors when implementing their new solutions and comparing them against already devised ones, making sure no additional performance is introduced because of the selected programming language.

2 STATE OF THE ART

To understand the time evolution of modern Lambert's problem algorithms, it is important to recognize the different approaches developed during the last decades to address the problem. In this section, a brief refresh on such timeline is presented.

2.1 The Lancaster-Gooding-Izzo approach

We might establish the origin of modern Lambert's problem solvers with the publication of Lancaster's article [8] about a universal formulation for solving the problem, which extended also to the multi-revolution case. Lancaster related the non-dimensional transfer time, denoted by T, with an independent variable, named x, for a particular transfer angle parameter, q. The main advantage of his method was the bounding of the solution in different regions, being elliptic for values of $x < 1$, parabolic when $x = 1$ and hyperbolic for $x > 1$.

The original approach formulated by Lancaster setup the basis for Gooding's algorithm [6]. This routine improved previous one by working out an accurate initial guess and taking advantage of its bounded solutions. Gooding carried out a full analysis on his own method convergence, determining that no more than three iterations were required to achieve absolute tolerances near 10^{-13} floating point value. This routine was found to be the most robust one (see Klumpp's analysis [2]) and thus, it was selected for this performance comparison.

Just a few years ago, Izzo developed a new algorithm based on previous authors' solutions. This solver followed again Lancaster's approach but included a last step in which a new independent variable, ξ, was introduced. This change of variable translated into a simpler implementation, since the new time of flight curves presented a smooth evolution for zero revolution transfers. Izzo realized again that the initial guess becomes important for a faster convergence to the solution, and devised an algorithm more simpler (from the implementation point of view) than the one proposed by Gooding. This last fact made it one of the most popular Lambert solvers as of today, being used in several astrodynamics and orbital mechanics software [9] [10]. Because of all these facts, Izzo's solver was included in this analysis too.

2.2 The Bate-Vallado-Arora approach

Another path of solutions was born after Bate introduced a new solver in his book [11] about fundamental astronomy. The notation selected for the independent variable was z, being related with the eccentric and hyperbolic anomalies such that $z = \Delta E$ and $-z = \Delta F$. Although this method also makes use of the universal formulae as the Lancaster-Gooding-Izzo branch, it diverges in the way of bounding the solution.

For the case of Bate's algorithm, the independent variable can grow up to any multiple of 2π, being related thus with the number of revolutions as $z = 2\pi M$, where $M \in [0, \infty)$ is the number of revolutions. As a final comment on this method, Bate did not employed any particular initial guess for its formulation neither imposed a root finder.

[1]lamberthub source code: https://github.com/jorgepiloto/lamberthub

However, years later, Vallado developed a new routine on top Bate's one, which can be found in his popular book [12]. This solver was still based on the universal formulation, although the independent variable was renamed to ψ. In addition, Vallado also pointed out that bisection is a more robust root finder for the algorithm despite requiring more iterations, for particular cases.

The latest proposed algorithm within this branch is the routine devised by Arora [5]. Motivated by the work of Bate, this new formulation introduced a cosine transformation for the eccentric anomaly such that $\cos(\Delta E) = k^2 - 1$, where k is the independent variable in this case. A relation with the hyperbolic anomaly was also established, so a bounded solution was achieved by Arora. He also extended the method to the multi-revolution case, and a similar figure to the one presented originally by Lancaster was obtained. Nevertheless, the main feature of this method, as opposite to other modern routines, is the procedure when solving for the initial guess. This last reason, together with the fact of being the latest of his branch, made us to select it for the performance comparison work.

2.3 The Avanzini-He-Wen approach

Another modern approach was started by Avanzini after he published a very simple Lambert' solver [4]. This modern approach was based on the conservation of the projection of the eccentricity vector, $\vec{e}$, into the chord one, $\vec{c} = \vec{r_2} - \vec{r_1}$. The property seems to be introduced in the literature for the first time by Battin [13]. Only drawback about Avanzini's solver is that he did not extend it to the multi-revolution case. Nevertheless, since this analysis is limited to the zero-revolution (direct arc transfer) analysis, this solver was decided to be included in this work because of being the first one of its class.

Some years later, He [14] expanded Avanzini's algorithm to the multi-revolution case. In addition, He provided the derivative of the Kepler's equation with respect to (w.r.t.) the transverse eccentricity component. Avanzini assumed a step-integration for the numerical root solver as it will be explained in detail in the next section.

The last work on this branch seems to be performed by Wen [15], who gave a full analysis on the available eccentricity-based methods and introduced additional proofs on the analytical behaviour of the time derivative w.r.t. the transverse eccentricity component.

2.4 A note about GPU-accelerated solvers

To conclude the state of the art analysis, it must be pointed out that interest has increased about the acceleration of Lambert's problem solvers via Graphical Processing Units (GPUs). Different articles [16] [17] [18] studying the application of these devices have been published during the last years. Therefore, another performance comparison could be based on the feasibility of a solver to be paralleled in GPUs. Nevertheless, this comparison is out of the scope of this work.

3 ALGORITHMS ANALYSIS

In this section, a review on each one of the selected algorithms is presented, so a comparison from an analytical point of view is achieved. Common parts have been identified, such us the iteration variable (usually called the free-parameter), the initial guess together with the numerical root finder and finally, the way in which velocity vectors are obtained. In addition, curves relating the time of flight as function of the free-parameter are introduced for each one of the algorithms.

Although attached figures might include multi-revolution regions, the analysis focuses is only focused in direct transfer angles, meaning that initial guess study is restricted also to the direct transfer problem. Table 1 is provided as a resume and for quick analytical comparisons.

3.1 Gooding's algorithm

As introduced in previous section, Gooding's algorithm was built on top of Lancaster's one and makes use of the universal formulation. The independent variable was named x and related with the semi-major axis via $x^2 = 1 - s/2a$, being a the orbit's semi-major axis, avoiding the singularity present in other methods when the orbit is parabolic, as $x = 1$ when $a = \infty$. The non-dimensional time of flight for this algorithm was originall defined as $\tau = \sqrt{(8\mu/s^3)}\Delta t$, being the semi-perimeter $s = (r_1 + r_2 + c)/2$.

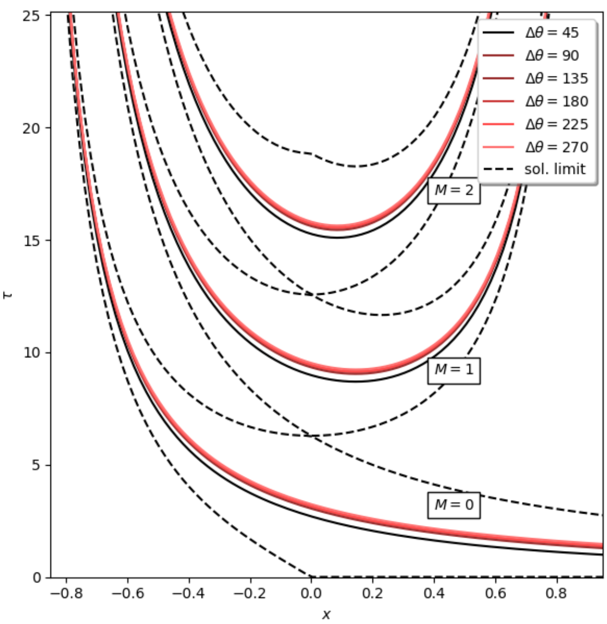

Figure 2: Gooding's time of flight as function of the independent variable, computed for $\rho = 2$. Dashed lines show the limit of the solution regions.

Gooding's solver, as opposite to Lancaster one, introduced the generation of an accurate initial guess, based on a bi-linear approximation for the single-revolution case and a more complex procedure based on the minimum transfer time for the multi-revolution case. In addition, Gooding made use of Halley's method to ensure a fast convergence to the solution.The curves for the time of flight as function of the transfer angle parameter q and the independent variable are provided in figure 2. The lines for $q = -1$ suffer show an slope change when $x = 0$ (minimum energy solution). Lancaster pointed out the necessity of using bisection methods within this region, since derivative based ones fail. However, the initial guess generated by Gooding already takes into account this region, not requiring to change the iterative method at all.

3.2 Avanzini's algorithm

Avanzini's algorithm is the very first one exploiting eccentricity as the independent variable to reach a solution to the problem. Because the orbit equation can be expressed as $\vec{e} \cdot \vec{r} = p - r$, if evaluated at the two known position vectors, $\vec{r_1}$ and $\vec{r_2}$, it is possible to obtain equation 1:

$$\vec{e} \cdot (\vec{r_2} - \vec{r_1}) = r_1 - r_2 \tag{1}$$

where it can be seen that the projection of the eccentricity vector along the chord one remains constant. Hence, by calling the value of the projection the fundamental eccentricity e_F and the transverse one e_T, it is possible to iterate over this last value to find the time of flight which matches the actual one. This process is achieved making use of Kepler's equation, paying attention to elliptic, parabolic or hyperbolic formulae depending on the value of $|\vec{e}|$ found during a particular iteration. A final mathematical transformation is imposed to improve the convergence of the method, being the new independent variable x. Therefore, the final expression to be solved is:

$$\tau = f(\rho, \Delta\theta, e_T) \tag{2}$$

where f is Kepler's equation evaluated at observation vectors' norms ratio $\rho = r_2/r_1$, transfer angle $\Delta\theta$ and the transverse eccentricity component e_T. The condition is that the computed τ matches the current non-dimensional time of flight $\tau^\star$ of the observations. In Avanzini's algorithm, the non-dimensional time of flight is found to be $\tau = \Delta t\sqrt{\mu/r_1^3}$.

Avanzini bounded the values of e_T according to the transfer angle. This is important, as he did not provide the explicit time derivative of Kepler's equation w.r.t. the independent variable, constraining the usage of bounded root finders like bisection or regula-falsi ones. Regarding the initial guess, no particular approach was followed and $x = 0$ value was arbitrarily chosen as the starting value.

The evolution of the non-dimensional time of flight as function of the independent variable is shown in figure 3 for a variety of transfer angles.

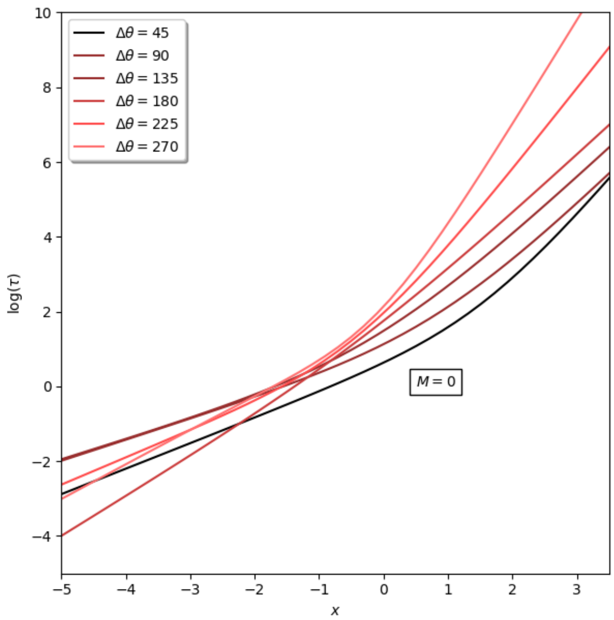

Figure 3: Avanzini's time of flight as function of the independent variable, for $\rho = 2$. Evolution for different non-collinear transfer angles is shown.

3.3 Arora's algorithm

Arora's is one of the most recent published solvers. Its solution originates from Bate's one although it makes use of a cosine transformation to simplify the relation between the independent variable named k and the time of flight. This variable is related with the eccentric anomaly for the elliptic case and also with the hyperbolic anomaly, for the corresponding orbit geometry.

In addition to the mathematical transformation proposed, Arora points that a robust initial guess is crucial if a fast convergence is desired. In fact, most of Arora's work is devoted to the explanations about for the initial guess, being the most extensive up to date. At firts, two major regions are identified: zero-revolution and multi-revolution zones (only the first one is considered in this report.)

The first step, for the zero-revolution initial guess, is to compute the time of flight for a parabolic orbit by making use of available problem geometry. If the current time of flight is shown to be greater or lower than parabolic one, the elliptic or hyperbolic region is selected. After that, a new region classification applies within previous regions, being the hyperbolic divided into two and the elliptic having a total of four zones. Tabulated times of flight are provided within original article, introducing hence a new approach to the initial guess formulation.

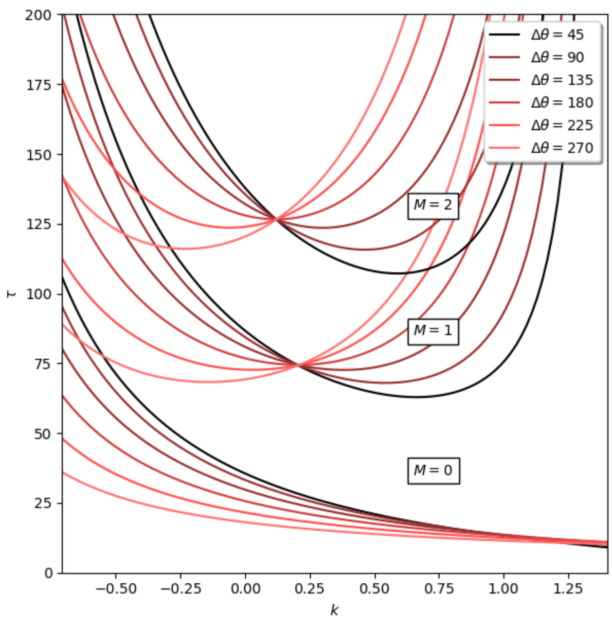

Figure 4: Arora's time of flight as function of the independent variable, for $\rho = 7.098$ (same value from original report). As opposite to Gooding's figure, only shows all possible solutions for a particular problem geometry.

The root solver used is Halley's one, similarly to Gooding, and the construction of velocity vectors is made via f and g functions. Figure 4, as in previous algorithms, shows the time of flight against the independent variable, being adimensionalized in the same way as with Avanzini solver.

3.4 Izzo's algorithm

Izzo's algorithm is the most modern of the solvers analyzed in this work. Most of Izzo's work inherits from Lancaster's one, similarly as Gooding's solver. However, a new Lambert invariant named ξ is introduced depending on the number of revolutions. The mathematical transformation applied makes the time of flight curves to have a new domain going from -∞ up to ∞.

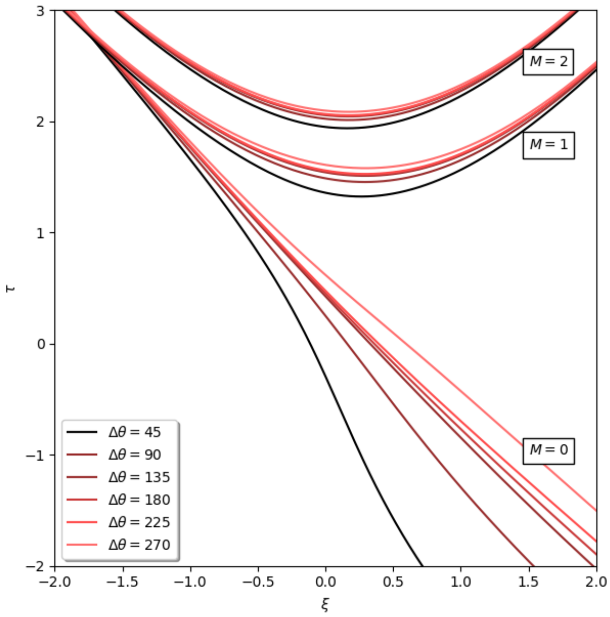

Figure 5: Izzo's time of flight as function of the independent variable, for $\rho = 1$

For these new set of curves, Izzo selected a Householder root finder, as opposite to Gooding and Arora, which made used of Halley's method. Izzo claimed this method to be the most beneficial from the iterative point of view. Regarding the initial guess, a first selection between zero-revolution and multi-revolution zones is applied. For the first one, a simple linear approximation is carried out imposing an asymptotic behaviour.

Finally, the velocity vectors are constructed by following the radial and tangential formulae. For this last solver, figure 5 provides, as in previous cases, the time of flight curves.

3.5 Algorithm's steps comparison table

All previous algorithms were analyzed considering common steps followed by any Lambert's problem solver: iteration free-parameter, the initial guess, the root finder and velocity vectors construction. All the information previously presented about those is collected within table 1.

Notice that, event if the approach followed for finding the solution is different, there are common elements. In particular, it is seen that high-order numerical methods are desired for a fast convergence of the solution. This is usually followed by a robust initial guess which ensures that only a few iterations will be required to achieve accurate solution values.

In addition, radial and tangential velocity construction is seen to be used by two of the four methods analyzed in this work, although Avanzini is the first one from all the cited methods which outputs the classical orbital elements (COE) for complete orbit definition.

Table 1: Algorithms fundamental steps comparison

Step	Method			
	Gooding	Avanzini	Arora	Izzo
Free-parameter	x	e_T	k	ξ
Initial guess	Bi-linear	$x = 0$	Rational formulae	Linear approximation
Root finder	Halley	Regula-falsi	Halley	Householder
Velocity vectors	Radial and tangential	COE to RV	f and g functions	Radial and tangential

4 PERFORMANCE COMPARISON

In the following sections, the performance comparison between Gooding [6], Avanzini [4], Arora [5] and Izzo [7] is presented. At first, the number of iterations required for a particular combination of dimensionless time of flight and transfer time is presented. However, because these contour plots do not provide a full insight of how the algorithm performs from the point of view of time, figures indicating the time per iteration and the total computation time are also provided.

This analysis was performed imposing that the final radius is twice in length as the initial one (i.e $\rho = 2$). Its direction was computed by generating all the values associated with a linear span of transfer angles between 0 and 2π. Finally, the absolute tolerance was selected to be 10^{-5} and the relative one 10^{-7}.

The results regarding benchmarking tests are influenced by the current solvers implementation, the machine specifications (Thinkpad X230 Intel Core i5-3320M CPU at 2.60G EndevourOS, in this case) and additional processes running on it. To avoid spurious values, several iterations were performed and average values were computed.

4.1 Number of required iterations

Figures 6, 7, 8 and 9 show the contour maps were the number of iterations is computed for a particular combination of the transfer angle and the non-dimensional time of flight, computed as $\tau = \sqrt{(8\mu/s^3)}\Delta t$, similarly to Gooding's one.

Arora's algorithm is shown to be the one requiring less iterations, followed by Izzo's. Gooding solver, as noticed by his author, only requires three iterations in many of the cases while Avanzini's procedure is shown to need four of those due to the root-finder used, the regula-falsi one.

4.2 Time required per iteration and total one

The time per iteration does not consider the one required for computing the initial guess neither the velocity construction one. Only the time between the start and end of the iterative process is measured. Results are provided within figures 10, 11, 12 and 13.

The fastest time per iteration is achieved by Gooding's solver, followed by Izzo's, Arora's and Avanzini's solvers in this order. Gooding, Arora and Izzo solver require more time if the region is near $\tau = \pi/2$ or below this value. Regarding Avanzini one, the regula-falsi method is the cause of such a big time per iteration.

Finally, although strongly influenced by the algorithms implementation, the total required time is also considered just to complete the whole performance analysis. This time is measured since the solver function is called till the velocity vectors are returned. Results are collected in figures 14, 15, 16 and 17, being summarized in table 2, were Izzo's solver is found to be the fastest one.

Table 2: Performance comparison average results

Method	Average iterations	Time per iteration	Total time	Iteration workload
Gooding 1990	2.57	23.78 μs/iter	754.83 μs	8.10 %
Avanzini 2008	4.46	153.23 μs/iter	2679.74 μs	25.50 %
Arora 2013	2.06	37.80 μs/iter	558.50 μs	13.94 %
Izzo 2015	2.32	36.37 μs/iter	441.29 μs	19.12 %

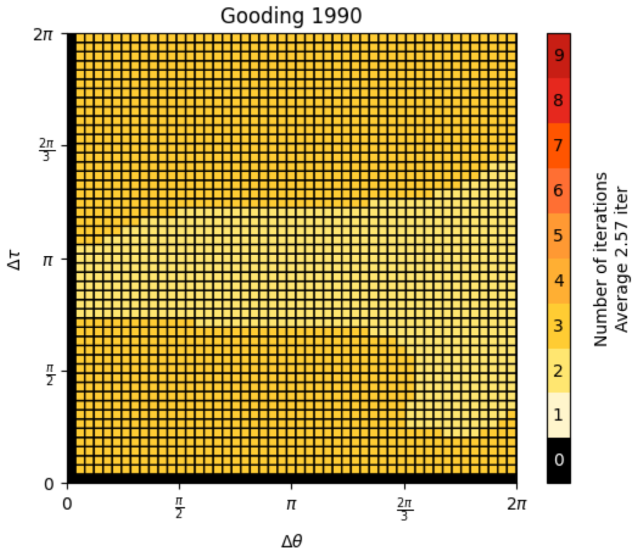

Figure 6: Gooding's algorithm does not exceed more than three iterations. In fact, original routine imposed this as the number of iterations by default, without letting user to select a particular value of them.

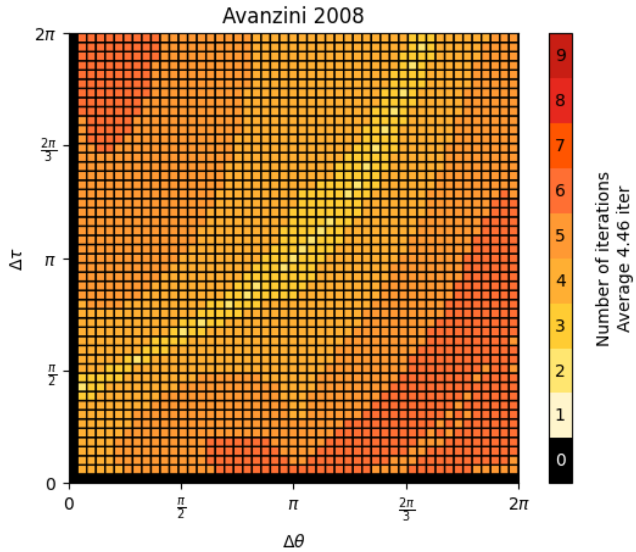

Figure 7: Avanzini's solver requires a greater number of iterations when compared to the rest of solvers. The central region seems to converge quickly while outer regions require more steps to be solved.

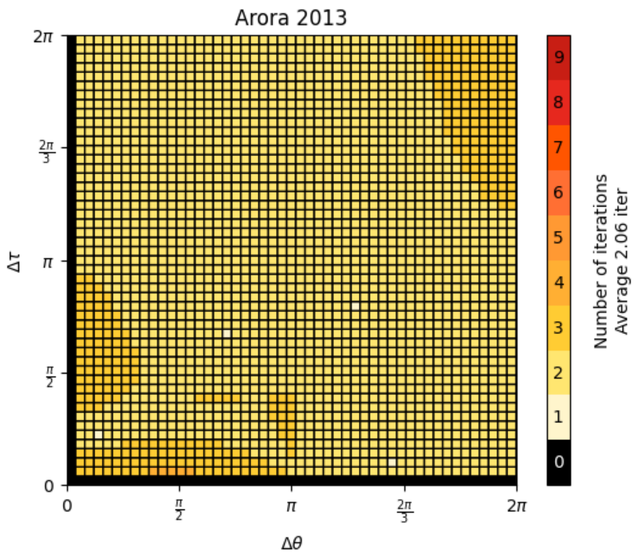

Figure 8: Arora's algorithm shows a uniform number of iterations, around two, while corner regions are seen to require up to three. It requires the lowest number of average iterations of all solvers.

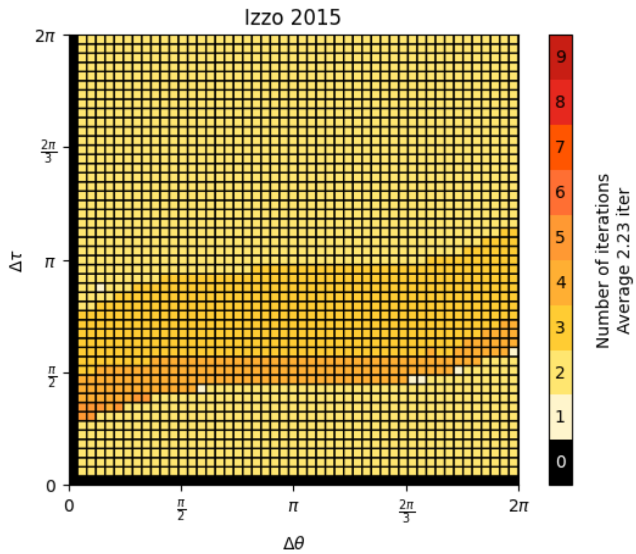

Figure 9: Izzo's solver takes only between two to three iterations in the majority of the cases. However, it shows an increase up to four of them when the non-dimensional time of flight close is to $\tau = \pi/2$.

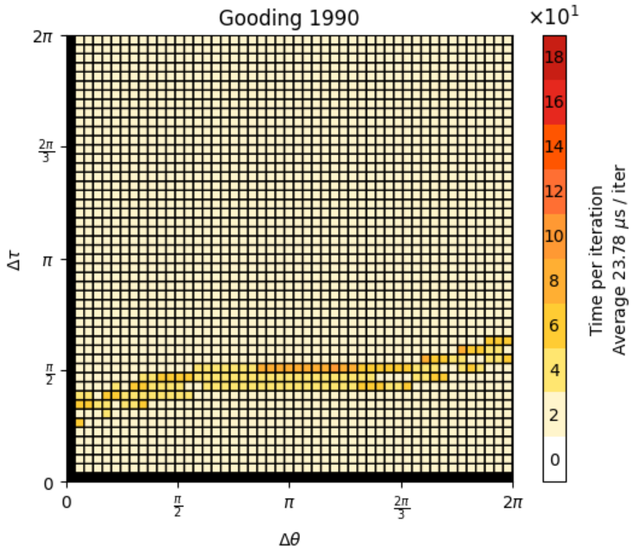

Figure 10: The algorithm developed by Gooding seems to require more time of computation when the solution holds the non-dimensional time close to $\tau = \pi/2$ although it performs very quickly.

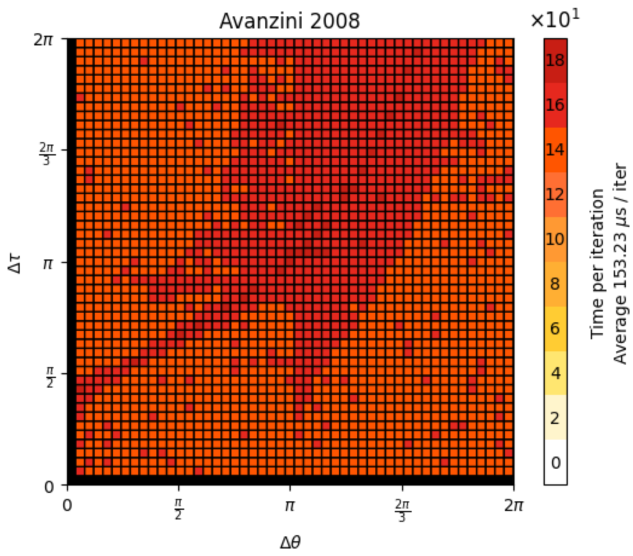

Figure 11: The high time required per iteration in Avanzini's routine is directly related to the root-finder used, that is, the regula-falsi method. In addition, the center region shows a lower performance.

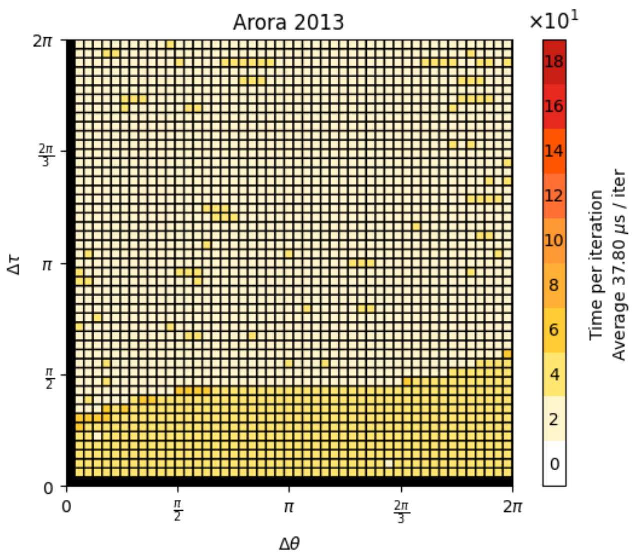

Figure 12: Arora's solver shows an increase in the required iteration time for times of flight below $\tau = \pi/2$ and in the regions located within the corner values.

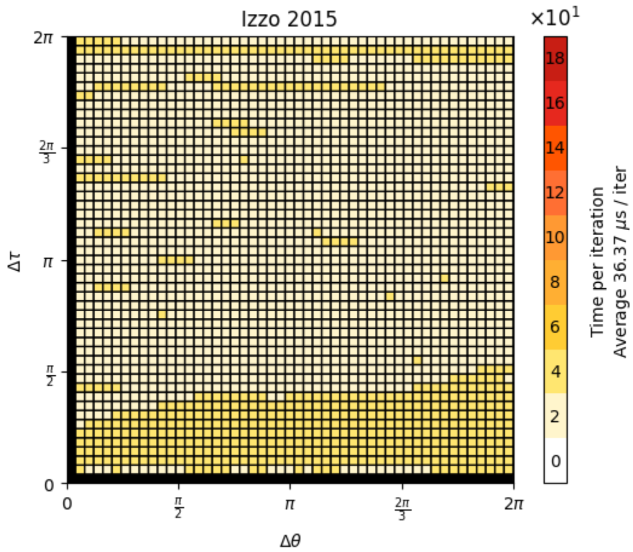

Figure 13: The algorithm by Izzo shows an stable iteration time, although suffers from a time increase in the lower region of the figure, similarly to other solvers.

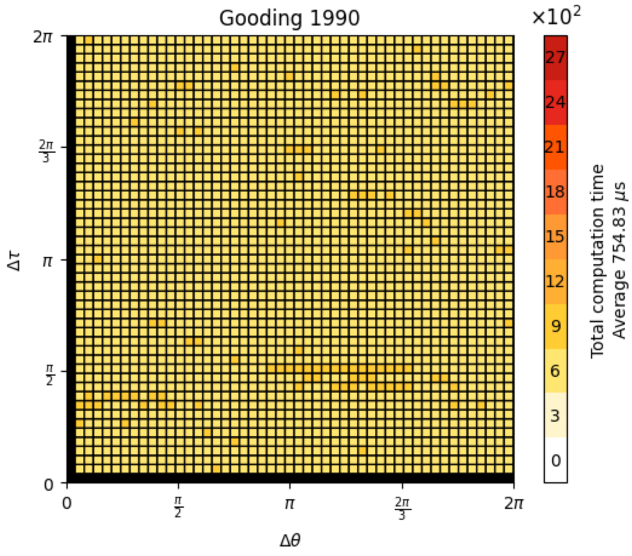

Figure 14: Gooding's total time of computation is seen to be uniform all over the combinations between transfer angle and non-dimensional time. Despite requiring a low number of iterations, its total mean time is relatively high.

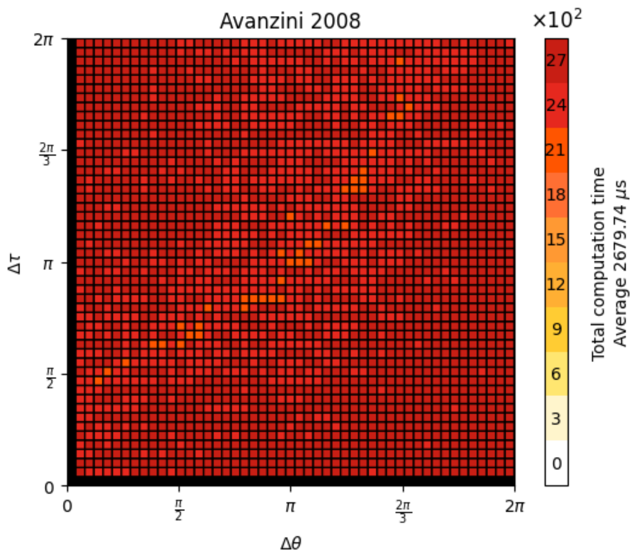

Figure 15: The amount of time required for the solver to retrieve the solution is probably a combination of non-existent initial guess, the numerical method used and the current implementation.

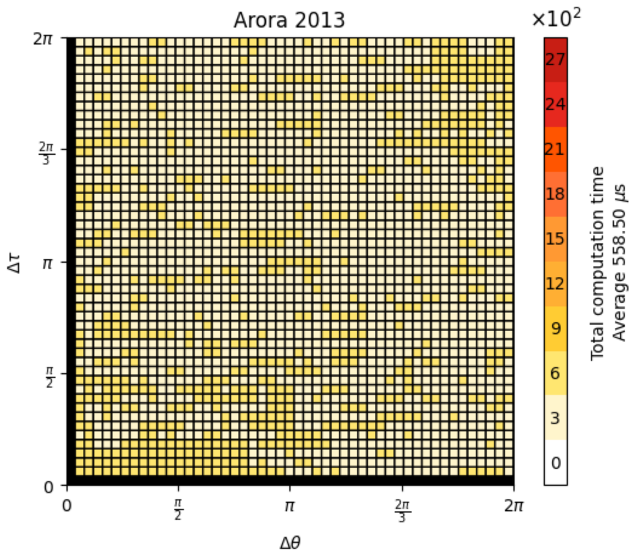

Figure 16: Arora's solver presents a uniform and low computation time for the whole problem. However, when compared to its previous figures, it can be seen that most of the time is consumed during the initial guess computation.

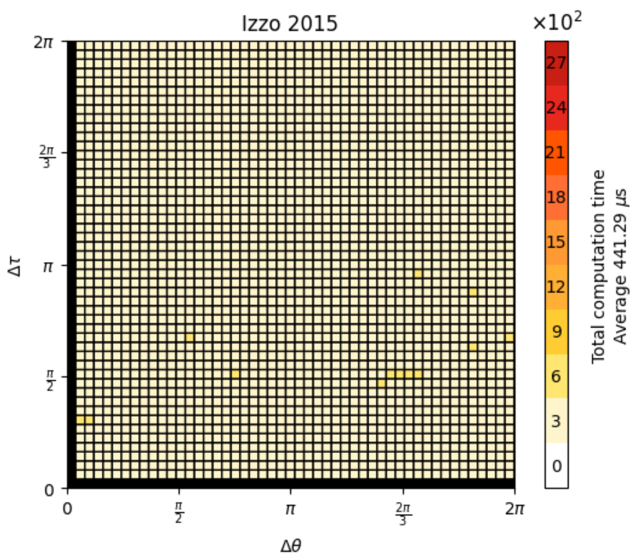

Figure 17: Izzo's shows highest time performance when addressing the whole problem. The total time of computation is seen to be the lowest one when compared to the rest of the solvers. It also shows an stable behaviour.

5 RESULTS AND CONCLUSION

With all the results about the number of required iterations, the time per iteration and the total one, it is possible now to present a variety of conclusions on all these modern Lambert's problem solvers.

5.1 Effects of the initial guess and root-solver

Recalling table 1 and solver's required time per iteration figures, it can be concluded that a robust initial guess followed by a high-order root solver leads to a low time per iteration required, increasing solver's performance.

In fact, the routines developed by Gooding and Arora implemented a 2nd order method (Halley's one) while Izzo makes use of a 3rd order solver (Householder). All these routines are seen to require a very low time per iteration, while the regula-falsi method employed by Avanzini significantly reduces performance.

5.2 About the iteration workload

From the data presented in the iteration workload column (see table 2) and taking into account the total one, it can be stated that Izzo's algorithm performs the best, as it combines a low total computation time, much of which is devoted to the iteration process.

Although Avanzini's solver is seen to have the highest iteration workload, this is due to the low rate of convergence produced by its root finder. Regarding Gooding's and Arora's solvers, the percentage is lower, as the computation of the initial guess requires more time.

5.3 Conclusion

Being collected and analyzed all the results, it is seen that methods based on the universal formulation presented in this work show a better performance than those using the eccentricity solution path.

A robust initial guess in combination with a high order root finder leads to a low time per iteration. However, special attention needs to be payed to the initial guess generation. If the subroutine employed is too complex, the total computation time will increase together with the implementation costs.

The algorithms presented within this work can be sorted in performance as follows: Izzo, Arora, Gooding and Avanzini.

However, previous solvers are just an small set all the available ones. Therefore, this analysis might be extended including more solvers or new solution approaches. In addition, additional scenarios, such us the multi-revolution case could be analyzed, taking advantage of all the different tools provided by this work.

REFERENCES

[1] J. H. Lambert, *Insigniores orbitae cometarum proprietates*, 1761.

[2] A. Klumpp, "Performance comparison of lambert and kepler algorithms," *Interoffice Memorandum, JPL*, 1999.

[3] D. d. l. Torre Sangrà and E. Fantino, "Review of lambert's problem," in *ISSFD 2015: 25th International Symposium on Space Flight Dynamics, 19-23 October, Munich, Germany*, 2015, pp. 1–15.

[4] G. Avanzini, "A simple lambert algorithm," *Journal of guidance, control, and dynamics*, vol. 31, no. 6, pp. 1587–1594, 2008.

[5] N. Arora and R. P. Russell, "A fast and robust multiple revolution lambert algorithm using a cosine transformation," *Paper AAS*, vol. 13, p. 728, 2013.

[6] R. Gooding, "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.

[7] D. Izzo, "Revisiting lambert's problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 121, no. 1, pp. 1–15, 2015.

[8] E. Lancaster and R. Blanchard, *A unified form of Lambert's theorem*. National Aeronautics and Space Administration, 1969, vol. 5368.

[9] D. Izzo, "esa/pykep: Bug fixes and more support on Equinoctial Elements," Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2575462

[10] J. L. C. Rodríguez, J. M. Garrido., A. Hidalgo, S. Bapat, N. Astrakhantsev, C. Eleftheria, M. Eugenia, Dani, A. Chaurasia, Yash-10, A. L. Márquez, D. Sondhi, T. Mrugalski, E. Selwood, O. Ousoultzoglou, P. R. Robles, G. Lindahl, A. Carballo, A. Rode, H. Eichhorn, H. Garg, H. Goyal, I. DesJardin, aOrionis, AntoniiaK, Josvth, P. Rohilla, Angala, O. Streicher, and R. A. Lehmkuhl, "poliastro/poliastro: poliastro 0.15.0 (Earth edition)," May 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4763538

[11] M. Bate and O. Mueller, "White. 1971. fundamentals of astrodynamics," 1978.

[12] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.

[13] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition*. American Institute of Aeronautics and Astronautics, 1999.

[14] Q. He, J. Li, and C. Han, "Multiple-revolution solutions of the transverse-eccentricity-based lambert problem," *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 265–269, 2010.

[15] C. Wen, Y. Zhao, and P. Shi, "Derivative analysis and algorithm modification of transverse-eccentricity-based lambert problem," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1195–1201, 2014.

[16] N. Arora and R. P. Russell, "A gpu accelerated multiple revolution lambert solver for fast mission design," in *AAS/AIAA Space Flight Mechanics Meeting*, vol. 136, 2010, pp. 10–198.

[17] S. Wagner and B. Wie, "Gpu accelerated lambert solution methods for the orbital targeting problem," *Spaceflight Mechanics*, vol. 140, no. 2, 2011.

[18] N. Parrish, "Accelerating lambert's problem on the gpu in matlab," 2012.