Solving Lambert's Problem

By: Eliot Aretskin-Hariton
Date: 6/5/2012

Solving Lambert's Problem requires finding the orbit of a spacecraft based on two observations of position and the time between those observations. It has its roots in the determination of the orbit of Ceres which Gauss solved many years past. Today it is used by Air Force Space Command which monitors the satellites of foreign countries. If a ballistic missile is launched, solving Lambert's problem will allows them to calculate the trajectory and extrapolate where it will hit. Many solution techniques exist and they all have their caveats. For the purpose of this study, we combine a number of different Lambert solvers into one program with the goal of facilitating fast and accurate selection of the appropriate Lambert solver for small and large angles, mutli-revolution, and prograde or retrograde motion.

Submitted as part of AE557 final project at Cal Poly, SLO, with Dr. Kira Abercromby

Lambert's problem has been tackled by many of the greatest geometers in the last two hundred and fifty years. Thorn, Battin, Gooding, Gauss, Sun, to name a few, have developed creative ways to solve this problem. The problem statement is generally straightforward:

> *Given two observed position vectors and a time between the observations and knowing the Sphere of Influence (SOI) the body is in, characterize the orbit profile by determining the velocity at the first and second observation.*

Solving this problem is, however, considerably more complex. The solution space typically has many singularities which can stump the best solving routines. We will review some of the solutions that currently exist and their strengths and weaknesses when it comes to resolving single-rev, multi-rev, prograde, retrograde, small-angle, and large-angle problems. We will show how these solutions can be combined into a larger solving program which can select the appropriate solver to use and validate the answer to ensure it does not pass through the central body. Finally we will review one of the newer techniques proposed by Dr. Gim Der in his quest to find the ultimate Lambert solver (that has his name on it rather than Gooding's).

Solvers

Lambert did formulate a minimum energy solution which is not complex enough for the analysis we are looking to perform. A more advanced solution to Lambert's problem was formulated by Gauss, over two hundred years ago, in his quest to determine the orbit of Ceres. The Gauss method and implementation is described in detail in Vallado [1]. Key for us is to understand the circumstances in which the solution will be unable to come to resolve an accurate answer. Gauss's formulation, as implemented for this project, is only applicable to single-rev solutions. Hyperbolic, parabolic and elliptical solutions are all possible. The formulation works best for a small angle ($\Delta v<10^o$) between the two observations. It can fail for very small angles $\Delta v<2^o$ as well (see test 3 in *Test_Cases.m*). Small angles can cause the 'L' value in this formulation to approach zero which causes the iteration on y to be unable to converge. In some cases, Gauss will be able to solve problems for larger angles ($\Delta v<42^o$). However, the results from such analysis will be less accurate than the equivalent Universal Variables or Battin's solver due to Gauss' difficulty with handling large angles. The implemented method will always fail for $\Delta v>90^o$ due to the truncation of the 'x2' term. Also, the iteration on Gauss is usually a slow process that requires patience so using it repeatedly is not advised.

Unlike Gauss' solution, the Universal Variables (UV) formulation works for all angles except $\Delta v=180^o$. It has a much more robust convergence due to the implementation of a bisection technique to whittle down the solution space with each iteration. While bisection is slightly slower, it is more robust and will steadily progress towards the optimum in cases when Newton's Iteration Method would be flummoxed. A robust iteration technique is desirable as it will allow a solution to be found over the whole range of valid angles. Another benefit of UV is that the solver does not have to understand any information on the conic section type of the resulting orbit. Rather it concentrates its effort on finding the delta V at both observations and a conic section can be deduced from that. We must proceed with caution when using this method as it will give solutions that pass through the central body (physically impossible but mathemagically solvable). Additionally, solutions may not be able to be attained when 'z' cannot be resolved into a positive value after several iterations.

Universal Variables is also useful in that it is easily extended to multi-rev solutions. Multi-rev solutions are always elliptical as hyperbolic and parabolic object would pass out of the SOI. As we will see later, ease of implementation can be the difference between having an algorithm implemented, and leaving it by the wayside.

Battin's Method is useful because it does not have the singularity that UV and Guass solutions have at $\Delta v=180^o$. It has a very good convergence profile which makes it useful in all cases except those that it takes excessive computational cycles to execute. Extending Battin to multi-rev is not covered as part of this presentation; however, it is a logical next step to the presented analysis.

All the methods presented here use the Lagrange coefficients f and g which breakdown for long orbit times.

Implementation

It is expedient to combine all the above constraints and solvers into a computer program which will then select the best solver(s) to use based on the standard input criteria of 2 position vectors (R1,R2), the time between them (dt), the number of revolutions (n_rev), the SOI (mu), and the direction of motion (tm). These were programmed into the *Lamberts_Analysis.m* file.

```
Input Criteria:   Lamberts_Analysis(R1,R2,mu,dt,n_rev,tm)
```

This program uses the above criteria to select which solver to use. First it analyzes the angle between the points which allows it to immediately determine if the Gauss solver is applicable. The program then checks the number of revolutions, if it is more than zero, it will only run only the UV solver because this is the only multi-rev solver implemented herein.

Once the correct solvers are found, the input variables to *Lamberts_Analysis* are passed into the individual solvers. The solver will either be able to come up with a unique solution or it will display an error. Multiple solvers may be run for a single problem (see example 1). One of the most important functions of *Lamberts_Analysis* is that after the calculations are complete, it analyzes each result to determine feasibility. UV and Battin consistently return solutions that pass through the center of the central body (e.g. Earth) so this check is an important element of sanity that ties the results of the Lambert's solvers to reality. This is done by taking the output velocity V1 from each solution and propagating it forward in time through using an unpurturbed Cowell's method (ODE45). This is a valid approach for short time periods. Checks are made at each output of position calculated by Cowell's

Method to ensure that the object does not pass within the radius of Earth + 100km which is considered the Burnup Altitude[1] for this study.

Example Problems

To test *Lamberts_Analysis* we looked at four cases all of which are contained in the Test_Cases.m file. The first and second test case and results are discussed below. Other cases are discussed but the output of the program is not presented.

Test1 Inputs

```
R1=[22592.145603;-1599.915239;-19783.950506] (km)
R2=[1922.067697;4054.157051;-8925.727465] (km)
dt=36000 (sec)
tm=1 (prograde)
n_rev=0 (less than 1 full revolution around central body)
mu=mu of earth (km^3/s^2)
```

Test1 Results:

```
Search Type: Universal Variables
Single Revolution
V1 =
   2.000652695891660
   0.387688615201789
  -2.666947758466661
a =
    2.614876556361197e+004
ecc =
   0.965199479635547
nfeval =
    39
Verifying No Earth Impact
Orbital path given by R1 and V1 does not impact planet within time "dt" given
```

The *Lamberts_Analysis* correctly determines that this is a good problem for UV solver because the angle is not 180 degrees. The resulting velocity vector at time 1 for R1 is displayed and the solver shows that there is no earth impact with this trajectory therefore it is presented as valid. *Lamberts_Analysis* also runs Battin's method:

---

[1] Note: To change the altitude at which the object impacts the planet/burns up in the atmosphere, change the 'r_planet' parameter inside *Lamberts_Analysis.*

```
Search Type: Battin Single Revolution
V1 =
   2.000642972736554
   0.387687076011579
  -2.666935393383449
a =
    2.614858364261956e+004
ecc =
   0.965199531342031
nfeval =
     5
Verifying No Earth Impact
Orbital path given by R1 and V1 does not impact planet within time "dt" given
ans =
Search Complete
```

Battin's method is able to come to the same conclusion as UV which again validates our results. 'nfeval' is used to evaluate the efficiency of each method and indicates how many times it needed to iterate to find the solution.

Test2 Inputs

Same inputs of test 1 except flip prograde to retrograde (tm=-1).

Test2 Results

```
Search Type: Battin Single Revolution
V1 =
   2.966168175553265
  -1.275775724966661
  -0.755458159179246
a =
    2.569524334402737e+004
ecc =
   0.847306704641063
nfeval =
     4
Verifying No Earth Impact
Orbital path given by R1 and V1 vector passes through planet and is invalid
ans =
Search Complete
```

Both UV and Battin's method are run again, only Battin is shown above. There is a resulting velocity, V1. However, our further analysis determines that this orbit is invalid because it passes through the center of the planet. This shows that our sanity check is working.

Other Solvers

The functionality of *Lamberts_Analysis* can easily be extended with the addition of different solvers. One such method that was investigated as a candidate for inclusion with this project is the Der Lambert solver. This solver is based on the paper by Dr. Der titled "The Superior Lambert Algorithm" [2]. The title is perhaps a bit self-aggrandizing and strikingly similar to Battin's "An Elegant Lambert Algorithm" [3] which was published twenty years previous. When it comes to ease of implementation, however, Battin clearly wins.

What appears on the surface as a well presented workflow to executing the algorithm soon turns into a witch-hunt. The main problem that was experienced in implementing the algorithm was solving the Laguerre equation:

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

The individual parts of the equation F(x),F’(x), and F’’(x) are presented in the paper but during implementation it was found that, x values of -0.5 consistently go outside the bounds of -1<x<0. Time constraints precluded testing of positive convergence for x=0.5 values. We did contact Dr. Der but he was not able to proffer any additional help. Further research would involve dissecting his *DerAstro.exe* function which is available online to understand the precise formulation of the Der algorithm.

Conclusion

Solving Lambert’s problem is necessary to keep track of current space assets and activities. Many Lambert solvers exist, some easier to implement than others. A combined solver, *Lamberts_Analysis.m*, was created to take the limitations for each solver and work around them. This combined solver was able to successfully detect and weed out solutions that were physically impossible. The program allows for a good estimate of the orbit to be made without requiring the user to understand the specifics of each individual solver. Additional Solvers can be easily added to increase the utility of the program.

References

[1] Vallado, David A., *“Fundamentals of Astrodynamics and Applications, Third Edition”*, 2007, Chapter 7.6, pages 485 – 494

[2] Der, Gim J., *“The Superior Lambert Algorithm”*, Advanced Maui Optical and Space Surveillance Technologies Conference, September 13, 2011

[3] Battin, R.H., Vaughan, R. M.,*“An Elegant Lambert Algorithm”*, Journal of Guidance Control and Dynamics, Volume 7, Issue 6, Pages 662-670, 1984

All code for this program is written in Matlab and can be downloaded at:

http://ehariton.weebly.com/ae557.html

DerAstro.exe Solver: http://derastrodynamics.com/docs/iOrbit_v1a.zip