(12)

NSWCDD/TR-92/235

LAMBERT ALGORITHM INCLUDING DRAG

BY J. A. LAWTON
STRATEGIC SYSTEMS DEPARTMENT

JANUARY 1992

Approved for public release; distribution is unlimited.

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
Dahlgren, Virginia 22448-5000

NSWCDD/TR-92/235

LAMBERT ALGORITHM INCLUDING DRAG

BY J. A. LAWTON
STRATEGIC SYSTEMS DEPARTMENT

JANUARY 1992

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
U. announced	☐
Justification	
By	
Dist. ibution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

Approved for public release; distribution is unlimited.

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
Dahlgren, Virginia 22448-5000

FOREWORD

The mathematical method presented here for finding solutions to Lambert's problem in the presence of drag is a central element to several of the computer programs used to determine missile performance for the Anti-Tactical Ballistic Missile Program. This report disousses the derivation of the linearized equations and resulting solution method to find approximate intercept solutions between orbiting and/or ground-launched aerospace vehicles. A robust iterative procedure to find exact solutions, based on the approximate solution method, is also presented for cases where the errors of the approximate solution are not sufficiently small. Numerical examples are presented, which serve to illustrate the salient features of the algorithms.

This report has been reviewed by Ted Sims, Head, Space Sciences Branch, and J. L. Sloop, Head, Space and Surface Systems Division.

Approved by:

R. L. SCHMIDT, Head
Strategic Systems Department

ABSTRACT

An algorithm is developed that solves Lambert's problem when a vehicle encounters drag. For an environment of low to moderate atmospheric density, Lambert's solution is obtained by linearizing about a reference Keplerian trajectory. This linearization process requires no numerical ordinary differential equation solution techniques and, as such, is computationally efficient. For flight through a region of higher atmospheric density, or if greater accuracy is required, an iterative technique is used that starts with the linearized solution and does a quasi-Newton iteration process with an analytic initial approximation to the Jacobian. For subsequent iterations, the Jacobian estimate is updated using Broyden's update. The resulting algorithm converges superlinearly.

CONTENTS

INTRODUCTION

Lambert's problem is a classical astrodynamics problem that plays a critical role in diverse space flight applications and current research topics. The definition of Lambert's problem is to find the trajectory of a vehicle that flies between two positions in space in a specified time of flight. The solution of Lambert's problem is central to a large class of rendezvous and intercept problems, including fuel-optimal and time-optimal transfer problems, single impulse intercept problems, etc. Because of its central importance in astrodynamics, many authors have proposed numerical solution techniques that strive for computational speed and robustness with regard to types of orbits that can be handled; of special note among these algorithms is that due to Battin and Vaughan[1,2] because of its tenacity with even ill-conditioned problems and its superlinear convergence properties.

When drag is introduced into the astrodynamics problem, however, these extant methods no longer directly apply. The initial velocity vector from these drag-free solution techniques (which together with the specified initial position vector uniquely defines the trajectory) provides a useful initial guess to iterative methods that incorporate drag in the solution in the resulting trajectory. Such a technique iterates on the three components of the velocity vector to find the three zeros of the differences between the final position of the integrated trajectory and the specified final position. Thus, the problem is that of finding the solutions to three nonlinear equations in three unknowns. A quasi-Newton or secant algorithm designed for n-dimensional nonlinear problems could be directly applied to this problem, but these techniques generally require a significant number of function evaluations (to at least compute an initial numerical Jacobian, and possibly to do line searches if the initial guess is far enough from the solution). Since one function evaluation for the problem at hand means one numerical solution of an ordinary differential equation (using, for example a Runge-Kutta or an adaptive method), which requires significant computational time, the number of function evaluations must be kept as low as possible to maintain computational efficiency.

One way of keeping the number of function evaluations low is to linearize about a Keplerian trajectory (i.e., a trajectory with no drag, which can be analytically integrated). The Keplerian Lambert's problem is solved using Battin and Vaughan's algorithm.[1,2] The linearized change in position, $\delta\mathbf{r}(t)$, is the difference between that trajectory with drag, $\mathbf{r}(t)$, and the Keplerian trajectory, $\mathbf{r}_k(t)$. The linearized initial velocity, $\delta\mathbf{v}_0$, is determined to make the final position change, $\delta\mathbf{r}(t_f)$, zero in the specified time of flight t_f. Then, to first order, the solution to Lambert's problem in the presence of drag is the Keplerian initial velocity plus this linearized initial velocity, $\mathbf{v}_0 = \mathbf{v}_{0k} + \delta\mathbf{v}_0$. The advantage to this technique is that no numerical ordinary differential equation solution technique is required; only one relatively inexpensive Gaussian quadrature is necessary.

For many applications in regions of moderate drag, the true error in the final position at time t_f resulting from this linearized approach is sufficiently small to consider the trajectory to be a solution. For applications requiring high accuracy and/or flight in high drag regions, the linearized solution provides an excellent first iteration in a secant method algorithm. Also, as part of the linearized solution process, an excellent analytical approximation of the Jacobian of the nonlinear problem is obtained, thus negating the need for an initial numerical Jacobian using finite differences to initialize the secant algorithm. The resulting algorithm converges superlinearly, typically to within 1-m accuracy in five or less iterations.

LINEARIZED EQUATIONS IN THE PRESENCE OF DRAG

Let the state of the interceptor vehicle be

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} \tag{1}$$

Furthermore, let σ denote a parameter of the system, which will have one value on the reference trajectory and another on the trajectory of interest. For the problem at hand, $\sigma = \frac{S\,C_D}{m}$, where S is the reference cross-sectional area, C_D is the drag coefficient, and m is the vehicle mass. Thus σ is the inverse ballistic coefficient, and $\sigma = 0$ for the reference Keplerian trajectory since $C_D = 0$.

In general notation, the state differential equation is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\sigma) \tag{2}$$

Or, for inverse-square gravity with drag

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ -\frac{\mu}{r^3}\mathbf{r} - \frac{\sigma}{2}\rho(r)\,|\mathbf{v}_r|\,\mathbf{v}_r \end{bmatrix} \tag{3}$$

where $\mathbf{v}_r$,the velocity of the vehicle relative to the atmosphere, is defined as

$$\mathbf{v}_r \triangleq \mathbf{v} - \boldsymbol{\omega}\times\mathbf{r} \tag{4}$$

and $\boldsymbol{\omega}$ is the Earth angular-rotation vector. Since $\mathbf{v}_r$ is in general not in the plane normal to the angular momentum vector, the motion of the vehicle is no longer planar as it is in drag-free motion. This is why the nonlinear problem cannot be reduced to two equations in two unknowns.

Linearizing about the reference trajectory

$$\delta\dot{\mathbf{x}} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{ref}\delta\mathbf{x} + \left(\frac{\partial \mathbf{f}}{\partial \sigma}\right)_{ref}\sigma \tag{5}$$

Evaluating the partials in Equation (5),

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{ref} = \begin{bmatrix} 0 & I \\ \frac{\mu}{r_k^3}\left(\frac{3\,\mathbf{r}_k\mathbf{r}_k^T}{r_k^2} - I\right) & 0 \end{bmatrix} \tag{6}$$

and

$$\left(\frac{\partial \mathbf{f}}{\partial \sigma}\right)_{ref} = \begin{bmatrix} 0 \\ \frac{1}{2}\rho(\mathbf{r}_k)\,|\,\mathbf{v}_{rk}\,|\,\mathbf{v}_{rk} \end{bmatrix} \tag{7}$$

The subscript "*k*" denotes that these quantities are on the Keplerian trajectory. Substituting Equation (6) and Equation (7) into Equation (5) gives

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} 0 & \mathbf{I} \\ \frac{\mu}{r_k^3}\left(\frac{3\,\mathbf{r}_k\mathbf{r}_k^T}{r_k^2} - \mathbf{I}\right) & 0 \end{bmatrix}\delta\mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{2}\rho(\mathbf{r}_k)\,|\,\mathbf{v}_{rk}\,|\,\mathbf{v}_{rk} \end{bmatrix}\sigma \tag{8}$$

When $\sigma = 0$, Equation (8) has a homogeneous solution

$$\delta\mathbf{x}_h = \Phi(t,t_0)\,\delta\mathbf{x}_0 \tag{9}$$

where $\Phi(t,t_0)$ is the 6×6 state transition matrix. From linear systems theory, $\Phi(t,t_0)$ can be written

$$\Phi(t,t_0) = \begin{bmatrix} \frac{\partial \mathbf{r}_k(t)}{\partial \mathbf{r}_k(t_0)} & \frac{\partial \mathbf{r}_k(t)}{\partial \mathbf{v}_k(t_0)} \\ \frac{\partial \mathbf{v}_k(t)}{\partial \mathbf{r}_k(t_0)} & \frac{\partial \mathbf{v}_k(t)}{\partial \mathbf{v}_k(t_0)} \end{bmatrix} \tag{10}$$

This is true because $\Phi(t,t_0)$ in Equation (9) is a linear transformation from the initial state vector to the final state vector, and for any linear transformation $\mathbf{z}_2 = \mathbf{A}\,\mathbf{z}_1$ straightforward partial differentiation gives $\frac{\partial z_2}{\partial z_1} = A$. The partials in Equation (10) have "*k*" subscripts because the homogeneous solution has $\sigma = 0$, meaning no drag is present, so the motion is Keplerian. In the case of Keplerian motion the final position and velocity are known analytically as functions of the initial position and velocity, and thus each of the partials contained in Equation (10) is determined analytically.

Now let $\sigma \neq 0$, and obtain the total solution through the variation of parameters solution technique:

$$\delta x(t) = \Phi(t,t_0)\,\delta x_0 + \int_{t_0}^{t} \Phi(t,\tau) \begin{bmatrix} 0 \\ d \end{bmatrix} d\tau \tag{11}$$

where

$$d \triangleq -\frac{\sigma}{2}\,\rho(r_k)\,|v_{rk}|\,v_{rk} \tag{12}$$

is the acceleration due to drag.

To determine the change in final position due to drag with the initial velocity equal to the Keplerian velocity, set $\delta x_0 = 0$ in Equation (11), and pick off the first three rows corresponding to the position vector:

$$\delta r_{fe} = \int_{t_0}^{t} \frac{\partial r_k(t_f)}{\partial v_k(\tau)}\, d(\tau)\; d\tau \tag{13}$$

The "*fe*" subscript stands for final error, since this is the error in final position neglecting drag. Determining the linear estimate of the final position error due to drag does not require using a numerical ordinary differential equation solution technique such as a Runge-Kutta but merely a quadrature of an analytic integrand. In practice, a 5-point Gaussian quadrature has been sufficient to accurately compute Equation (13).[3] For Keplerian orbits that extend outside the atmosphere (i.e., above a cutoff altitude of around 80-100 km), only that portion within the atmosphere needs to be integrated.

To solve Lambert's problem, it is desired to find the initial velocity that makes the final position change zero in the specified time of flight. Setting $\delta\mathbf{r}(t_f) = 0$ in Equation (11), and using Equations (10) and (13), gives

$$0 = \left[\frac{\partial \mathbf{r}_k(t_f)}{\partial \mathbf{v}_k(t_0)}\right]\delta\mathbf{v}_0 + \delta\mathbf{r}_{fe} \tag{14}$$

so that

$$\delta\mathbf{v}_0 = -\left[\frac{\partial \mathbf{r}_k(t_f)}{\partial \mathbf{v}_k(t_0)}\right]^{-1}\delta\mathbf{r}_{fe} \tag{15}$$

Then

$$\mathbf{v}_0 = \mathbf{v}_k(t_0) + \delta\mathbf{v}_0 \tag{16}$$

is the initial velocity (to first order) that will cause the vehicle to reach the specified final position in the specified time of flight.

FURTHER ITERATIONS

Often, the accuracy of the linearized solution is sufficient for the application at hand. It may be that in the altitude bands considered, and for the given times of flight for a given analysis, the errors in the trajectories are well within the noise of modeling assumptions and the desired accuracy. For example, experience has shown that for flight above 50-60 km with velocities up to orbital velocity and times of flight in the hundreds of seconds, trajectory errors from the linearized model are within a few kilometers. This level of accuracy is sufficient for system level modeling of low-altitude orbital maneuvers. In contrast, optimization techniques typically require a higher level of accuracy in the knowledge of the state.

If the linearized solution does not have sufficient accuracy, then further iterations are necessary. While the linearized solution method requires no numerical ordinary differential equation methods, thus making it extremely fast, successive iterations by their very nature must involve numerical ordinary differential equation methods. The goal, therefore, is to minimize the total number of times numerical ordinary differential equation solutions must be calculated.

To this end, casting the problem in nonlinear systems parlance, the goal is to find the vector

$$x \triangleq v(t_0) \in \Re^3 \tag{17}$$

such that the map

$$F(x) : \Re^3 \rightarrow \Re^3 \tag{18}$$

$$F(x) \triangleq r(t_f) - r_f \tag{19}$$

is the zero vector. Expanding Equation (19) in a Taylor series, and truncating after the first-order term

$$F(x + \Delta x) = F(x) + J(x)\,\Delta x \tag{20}$$

where

$$J(x) \triangleq \frac{\partial F(x)}{\partial x} \tag{21}$$

is the Jacobian of F(x). Since it is desired that $F(x+\Delta x) = 0$, setting the left side of Equation (20) to zero and solving for Δx results in

$$\Delta x = -J(x)^{-1} F(x) \tag{22}$$

This Δx is a well-known Newton's method step, which successively upgrades x of the next iteration with $x + \Delta x$. This iteration scheme converges quadratically if the initial guess is sufficiently close to the solution and if F(x) is sufficiently well behaved. If the Jacobian cannot be determined analytically, as is the case for the problem at hand, then typically a finite-difference estimate of the Jacobian is calculated at the beginning of the iteration scheme. Give this estimate the symbol A_0. Subsequent iterations then update the estimate of the Jacobian with Broyden's update formula[4]

$$A_{n+1} = A_n + \frac{F(x_n)\,\Delta x_n^T}{\Delta x_n^T \Delta x_n} \tag{23}$$

The resulting iteration scheme using A_n in place of $J(x_n)$ in Equation (22) is commonly called a quasi-Newton algorithm or a secant method.

With the linearized estimate of Lambert's problem from the previous section, the finite difference estimate of the Jacobian need not be calculated. To see this, recast Equation (22) in r and v notation, so that

$$\Delta v(t_0) = -\left[\frac{\partial r(t_f)}{\partial v(t_0)}\right]^{-1} \Delta r(t_f) \tag{24}$$

Comparison of Equations (24) and (15) reveals that the linearized solution method is essentially a first iteration in the quasi-Newton process, using an analytic approximation to the Jacobian on the

Keplerian trajectory, and an estimate for $\Delta\mathbf{r}(t_f)$ from the Gaussian quadrature, $\delta\mathbf{r}_{fe}$ (Equation (13)). Hence, a first iteration is obtained with no function evaluations of $\mathbf{r}(t_f)$. For the second and subsequent iterations, $\mathbf{r}(t_f)$ must be accurately computed with some numerical ordinary differential equation solution method.

As was mentioned, the initial estimate of the Jacobian is the analytically derived partial derivative matrix evaluated for Keplerian flight:

$$A_0 = \left[\frac{\partial \mathbf{r}_k(t_f)}{\partial \mathbf{v}_k(t_0)}\right] \tag{25}$$

This has been used successfully, without update, as an estimate to the Jacobian for the whole iteration process. The convergence, however, is not generally superlinear. Broyden's update can be added to refine the knowledge of the Jacobian at each iteration, at very little extra cost in terms of computational speed. Using Equation (23) with Equations (17) and (19), the update is

$$A_{n+1} = A_n + \frac{(\Delta\mathbf{r}_{f,n+1})(\Delta\mathbf{v}_{0,n+1})^T}{(\Delta\mathbf{v}_{0,n+1})^T(\Delta\mathbf{v}_{0,n+1})} \tag{26}$$

All of these quantities on the right-hand side of Equation (26) are already known from the numerical ordinary differential equation solution of $\mathbf{r}(t_f)$ and from the evaluation of Equation (24), so the cost of updating the Jacobian estimate is minimal. The resulting algorithm does in practice converge superlinearly.

Before leaving this discussion, it is appropriate to mention that more general quasi-Newton methods incorporate line searches,[4] or the equivalent, in order to help secure convergence even when the initial guess is far from the solution. These line searches have proven to be unnecessary in practice for the problem at hand. Experience has shown that in cases where the linear approximation is so far from reaching the desired point as to require line searches, the resulting solution is far from feasible in practical terms, requiring, for example, thousands of kilometers per second of initial velocity. Note that the magnitude of the relative error of the linear approximation can be used as a measure of whether a given set of ballistic coefficient, velocity, and time of flight yields feasible performance for a given altitude band. If the relative errors are large, the vehicle under these conditions behaves like a "feather in the wind" (that is, drag is extremely high for the given mass), thus rendering flight of that vehicle at those altitudes generally infeasible. As for the total iterative algorithm developed here, the quasi-Newton method works very well without line searches for physically meaningful problems.

RECAPITULATION OF THE ALGORITHM

For clarity, a concise recapitulation of the algorithm is given at this point. Given $\mathbf{r}_0$, $\mathbf{r}_f$, and t_f, Lambert's problem for Keplerian flight (no drag) is solved by extant methods for $\mathbf{v}_k(t_0)$. This velocity is the initial guess for the case of Lambert's algorithm with drag. For the first iteration, perform the integration in Equation (13) using 5-point Gaussian quadrature, evaluating the integrand along the Keplerian trajectory, and set

$$\Delta \mathbf{r}_{f,0} = \delta \mathbf{r}_{fe} \tag{27}$$

Also, set A_0 to the numerical approximation of the Jacobian, Equation (25). Then iterate the following five equations until convergence is achieved:

$$\Delta \mathbf{v}_{0,n+1} = -A_n^{-1} \Delta \mathbf{r}_{f,n} \tag{28}$$

$$\mathbf{v}_{0,n+1} = \mathbf{v}_{0,n} + \Delta \mathbf{v}_{0,n+1} \tag{29}$$

$$\mathbf{r}_{f,n+1} = f(\mathbf{v}_{0,n+1}, \mathbf{r}_0, t_f) \tag{30}$$

$$\Delta \mathbf{r}_{f,n+1} = \mathbf{r}_{f,n+1} - \mathbf{r}_f \tag{31}$$

and

$$\mathrm{A}_{n+1} = \mathrm{A}_n + \frac{(\Delta \mathbf{r}_{f,n+1})(\Delta \mathbf{v}_{0,n+1})^T}{(\Delta \mathbf{v}_{0,n+1})^T(\Delta \mathbf{v}_{0,n+1})} \tag{32}$$

EXAMPLES

As a first example of the algorithm derived herein, consider Lambert's problem with initial position $\mathbf{r}_0 = (6418 \quad 0 \quad 0)^T$ km, final position $\mathbf{r}_f = (6412 \quad 563 \quad 0)^T$, and time of flight $t_f = 200$ sec. Three trajectories corresponding to the solution to this problem are drawn in Figure 1. The Keplerian trajectory is the solution if no drag were present. This solution is determined using extant Lambert's algorithms for drag-free flight. The Keplerian solution is $\mathbf{v}_{0k} = (0.927 \quad 2.853 \quad 0)^T$ km/sec. The lowest trajectory in Figure 1 is the trajectory flown with the Keplerian initial velocity in the presence of drag; that is, it is uncompensated for drag. The final position error for this uncompensated trajectory is 8.2 km. The third trajectory is the result of adding the linearized initial velocity to the Keplerian velocity, yielding the total velocity $\mathbf{v}_0 = (0.941 \quad 2.893 \quad 0)^T$ km/sec. The final position error of the linearized trajectory is 0.8 km.

If greater accuracy is required, then iterations using Broyden's update are necessary. If accuracy to within 1 m is desired, then two more iterations are required. The errors of those two iterations are 9.0×10^{-2} km and 1.6×10^{-4} km. Subsequent iterations, if extreme accuracy is required, yield the following sequence of errors: $\{4.0 \times 10^{-6}$ km , 1.4×10^{-9} km$\}$. These iterations demonstrate the superlinear convergence properties of the algorithm.

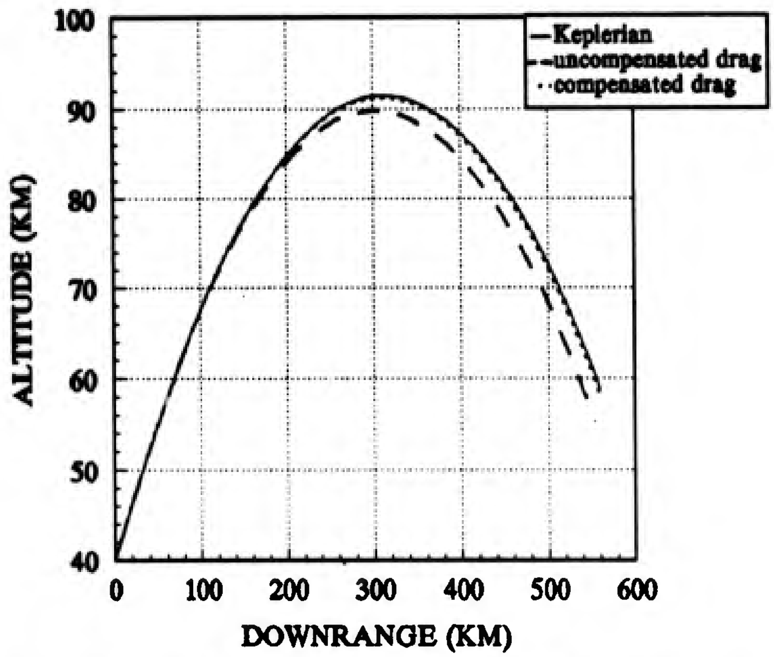

FIGURE 1. EXAMPLE LINEARIZED SOLUTION RESULTS

For a more extreme example, consider Figure 2. Almost the same configuration is examined in this example, except the initial and final altitudes are dropped by 20 km thus significantly increasing the effects of drag on the trajectory. In the figure, the number next to each trajectory is the iteration number. The zeroth iteration is the result of using the Keplerian velocity; it has an error of 174 km. The linearization process, iteration 1, reduces the error to 39 km. Subsequent iterations yield the following sequence of errors: {4.0 km, 2.3 km, 1.5 km, 2.1×10^{-2} km, 1.1×10^{-4} km, 1.5×10^{-6} km, 1.3×10^{-8} km}. Again, the superlinear convergence is evident. The final iteration increased the initial velocity from the Keplerian solution $\mathbf{v}_{0k} = (0.927 \quad 2.853 \quad 0)^T$ km/sec to $\mathbf{v}_0 = (1.232 \quad 3.721 \quad 0)^T$ km/sec.

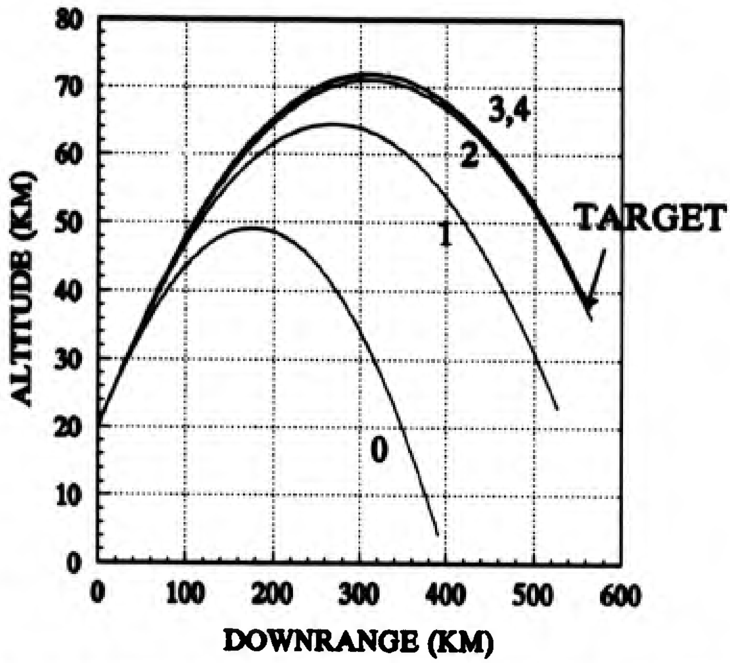

FIGURE 2. SUCCESSIVE TRAJECTORIES IN THE ITERATION PROCESS

CONCLUSIONS

Two algorithms are presented for addressing solutions to Lambert's problem with drag. First, for an environment of low to moderate atmospheric density, and for moderate accuracy requirements, a linearized solution technique is proposed. This technique requires no numerical ordinary differential equation solutions (only a fast 5-point Gaussian quadrature), and has an analytic transition matrix, so it is efficient for those techniques involving drag. It has been used successfully to do large-scale systems-level analysis without requiring inordinate computer time.

For problems requiring greater accuracy, such as trajectory optimization, or for flight at very low altitudes, the linearized solution must be embellished with an iteration process. This process uses the analytic transition matrix from the linearized solution as an initial estimate of the Jacobian, thus negating the need for an expensive finite difference Jacobian. Subsequent iterations improve the Jacobian, using information gained in each iteration, with Broyden's update formula. The resulting algorithm converges superlinearly, requiring few numerical ordinary differential equation solutions in the process. This is important since ordinary differential equation solutions are very expensive in terms of computer run time.

REFERENCES

1. Battin, R. H., and Vaughan, R. M., "An Elegant Lambert Algorithm," *J. Guidance*, Vol. 13, Sep.–Oct. 1990, pp. 662–670.

2. Battin, R. H., "Solving Lambert's Problem," *An Introduction to the Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, New York, 1987, pp. 295–342.

3. Carnahan, B., Luther, H. A. and Wilkes, J. O., "Gaussian Quadrature," *Applied Numerical Methods*, John Wiley & Sons, New York, 1969, pp. 101–105.

4. Dennis, J. E., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, New Jersey, 1983.

DISTRIBUTION

	Copies
ATTN SEA 06	1
SEA 06R	1
COMMANDER NAVAL SEA SYSTEMS COMMAND 2531 NATIONAL CENTER BLDG 3 WASHINGTON DC 20362-5160	
ATTN CODE 12	1
CODE 30	1
OFFICE OF CHIEF OF NAVAL OPERATIONS 800 N QUINCY ST ARLINGTON VA 22217-5000	
ATTN SPAWAR 005	1
COMMANDER SPACE AND NAVAL WARFARE SYSTEMS 2451 CRYSTAL GATEWAY BLDG 5 WASHINGTON DC 20363-5200	
ATTN CODE 37	1
COMMANDING OFFICER NAVAL AIR WARFARE CENTER WEAPONS DIVISION CHINA LAKE CA 93555-6011	
ATTN CODE 8000 (NAVAL CENTER FOR SPACE TECHNOLOGY)	1
NAVAL RESEARCH LABORATORY 4555 OVERLOOK DRIVE SW WASHINGTON DC 20375-5320	
CENTER FOR NAVAL ANALYSES	1
4401 FORD AVE ALEXANDRIA VA 22302-0268	
DEFENSE TECHNICAL INFORMATION CENTER	12
CAMERON STATION ALEXANDRIA VA 22304-6145	
ATTN GIFT AND EXCHANGE DIVISION	4
LIBRARY OF CONGRESS WASHINGTON DC 20540	
ATTN LUIGI PERINI MAIL STOP 6-354	2
JOHNS HOPKINS UNIVERSITY APPLIED PHYSICS LABORATORY JOHN HOPKINS RD LAUREL MD 20723-6099	
ATTN F H LUTZE	1
DEPARTMENT OF AEROSPACE AND OCEAN ENGINEERING VIRGINIA TECH BLACKSBURG VA 24061	

INTERNAL DISTRIBUTION:

	Copies
E231	3
E232	2
G205	1
G21	1
G23	2
G23 (CHADWICK)	1
K10	1
K12	1
K13	10
K14	1
K105	1
K105 (MADDRY)	1
K106	1
K107	1
K40	1
K41	1
K44	1
N24	1
N24 (GODDIN)	1

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1992	Final

4. TITLE AND SUBTITLE
Lambert Algorithm Including Drag

5. FUNDING NUMBERS

6. AUTHOR(S)
J. A. Lawton

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Surface Warfare Center (K13)
Dahlgren Division
Dahlgren, VA 22448-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWCDD/TR-92/235

9. SPONSORING/MONITORING AGENCY NAME(S) AND

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An algorithm is developed that solves Lambert's problem when a vehicle encounters drag. For an environment of low to moderate atmospheric density, Lambert's solution is obtained by linearizing about a reference Keplerian trajectory. This linearization process requires no numerical ordinary differential equation solution techniques and, as such, is computationally efficient. For flight through a region of higher atmospheric density, or if greater accuracy is required, an iterative technique is used that starts with the linearized solution and does a quasi-Newton iteration process with an analytic initial approximation to the Jacobian. For subsequent iterations, the Jacobian estimate is updated using Broyden's update. The resulting algorithm converges superlinearly.

14. SUBJECT TERMS
algorithm   Lambert's problem   drag   trajectory

15. NUMBER OF PAGES
23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and its title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

BLOCK 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If Known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2
NTIS - Leave blank

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief ***(Maximum 200 words)*** factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code ***(NTIS only)***

Block 17.-19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of this page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.