ANTITACTICAL BALLISTIC MISSILE
GLOBAL EFFECTIVENESS MODEL (AGEM)
INTERCEPT ALGORITHM

BY J. A. LAWTON C. A. BYRUM
STRATEGIC AND SPACE SYSTEMS DEPARTMENT

JULY 1994

Approved for public release; distribution unlimited.

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
DAHLGREN, VIRGINIA 22448-5100

DTIC QUALITY ASSURANCE

94 8 92 1 23 1
ANTITACTICAL BALLISTIC MISSILE
GLOBAL EFFECTIVENESS MODEL (AGEM)
INTERCEPT ALGORITHM

BY J. A. LAWTON  C. A. BYRUM
STRATEGIC AND SPACE SYSTEMS DEPARTMENT

JULY 1994

Approved for public release; distribution is unlimited

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION
Dahlgren, Virginia 22448-5100
FOREWORD

The intercept algorithm presented in this report is central to the Antitactical Ballistic Missile Global Effectiveness Model (AGEM) and is intended to be a robust and operationally flexible solution method. This report discusses the derivation of the exact derivatives required to iteratively solve operationally constrained Lambert problems. Examples of the implemented algorithm's convergence characteristics are also presented.

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 and Space Systems Department
ABSTRACT

This report discusses general solutions to the spaceflight intercept problem between moving source and target vehicles, under the constraint that only one limited thrust burn is permitted. This one burn is approximated as an instantaneous change in velocity, Δv. Given first is a discussion of the solution space, aided by Lambert’s theorem. Generally, two solutions (and sometimes more) to the posed intercept problem exist for a given launch time. Although previous solution techniques in the literature concentrate on finding the minimum time solution, both solutions are significant from an operational and a theoretical standpoint. Next, exact derivatives for two different operational requirements are derived, which facilitate finding all solutions to the intercept problem. Finally, example numerical problems solved by the Antitactical Ballistic Missile Global Effectiveness Model (AGEM) are presented to demonstrate the solution process for both operational requirements.
CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>SOLUTION SPACE</td>
<td>2</td>
</tr>
<tr>
<td>SIMPLE ITERATION SCHEME FOR CASE 1</td>
<td>5</td>
</tr>
<tr>
<td>CASE 1 DERIVATIVE</td>
<td>6</td>
</tr>
<tr>
<td>CASE 2 PROCEDURES</td>
<td>11</td>
</tr>
<tr>
<td>EXAMPLES AND COMPARISONS</td>
<td>13</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>23</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>23</td>
</tr>
<tr>
<td>DISTRIBUTION</td>
<td>(1)</td>
</tr>
</tbody>
</table>

ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Illustration</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TYPICAL $\Delta v$ VERSUS TOF CURVE</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>TYPICAL LAUNCH-TIME/TOF LEVEL CURVE</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>TYPICAL LAUNCH-TIME/INTERCEPT-TIME LEVEL CURVE</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>EXAMPLE SOLUTION SPACE</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>EXAMPLE CONSTRAINTS</td>
<td>15</td>
</tr>
</tbody>
</table>
CONTENTS (Continued)

Figure  Page
6  EXAMPLE 1 CONVERGENCE ........................................... 20
7  EXAMPLE 2 CONVERGENCE ........................................... 21
8  EXAMPLE 7 CONVERGENCE ........................................... 21

TABLES

Table  Page
1  EXAMPLE SUMMARY: FLIGHT TIME LIMITS CONVERGED ........ 15
2  EXAMPLE 1: NEW ALGORITHM—CASE 1 ............................ 16
3  EXAMPLE 2: NEW ALGORITHM—CASE 1 ............................ 16
4  EXAMPLE 3: OLD ALGORITHM—CASE 1 ............................ 17
5  EXAMPLE 4: NEW ALGORITHM—CASE 1 ............................ 18
6  EXAMPLE 5: NEW ALGORITHM—CASE 1 ............................ 18
7  EXAMPLE 6: NEW ALGORITHM—CASE 1 ............................ 19
8  EXAMPLE 7: NEW ALGORITHM—CASE 1 ............................ 19
9  EXAMPLE 8: NEW ALGORITHM—CASE 2 ............................ 20
INTRODUCTION

The spaceflight intercept problem discussed herein is to find the intercept trajectory from a source vehicle in orbit (or on the Earth) to a target vehicle in orbit (or on the Earth), with fixed thrusting capabilities. Many authors have discussed various problems related to this intercept problem. The classical Lambert problem (viz., find the trajectory that flies between two fixed positions in a specified time-of-flight (TOF)) is intimately related, but the problem considered here involves moving source and target vehicles, not fixed positions. The minimum fuel intercept problem does consider moving source and target trajectories, allowing the number of burns and the amount of fuel to be variable. In an operational environment, however, the size and type of engine, amount of fuel, etc., are fixed, so there is a requirement to find intercept solutions with these fixed thrusting capabilities. This is especially important for solid propellant engines, which generally burn only once. This report presents results when the thruster allows only one burn with a fixed characteristic velocity. This burn is approximated as an instantaneous change in velocity, $\Delta v$.

Parks, et al., describe a hybrid technique for solving the one-burn constrained characteristic velocity intercept problem. The technique first uses a simple maximum range check to reduce the search space of launch/intercept time pairs. Then, to find specific solutions, an iterative technique on the interceptor TOF is applied to obtain the $\Delta v$, which achieves intercept when added to the source vehicle velocity. Extensive numerical experience with this algorithm has shown that it always finds the minimum time solution if the initial guess on TOF is sufficiently small (probably due to the small flight path angle assumption), but it never converges to the maximum time solution. The next section shows that both solutions are significant from an operational and a theoretical standpoint. So the algorithm of Parks, et al., is very fast if minimum time solutions are desired, but it is not robust for finding general solutions. Cochran and Davy also developed an iterative algorithm which obtains an initial guess from an approximate solution to find the constrained one-burn minimum time solution to the intercept problem.

This report discusses the fixed impulse intercept problem from a broader viewpoint than that of finding minimum time solutions. Tools are developed to facilitate finding all solutions to the intercept problem, with a view to satisfying a variety of operational requirements. If a specified characteristic velocity must be satisfied, then specific solutions meeting operational requirements can be found with these tools. If a maximum characteristic velocity is given, and the engine can be throttled to obtain lower energy burns, then a range of solutions can be obtained.
SOLUTION SPACE

To understand what type of analysis techniques are required to find solutions to the intercept problem, the shape of the solution space must be understood. Figure 1 contains a typical plot of $|\Delta v|$ versus TOF for orbiting source and target vehicles. It is obtained by solving Lambert's problem for numerous values of the TOF. As the TOF approaches zero, the required $|\Delta v|$ approaches infinity. $\Delta v_g$ is the given characteristic velocity of the source. The points where the horizontal $\Delta v_g$ line crosses the $|\Delta v|$ curve correspond to solutions to the posed problem. The first solution (minimum value of TOF) corresponds to the depressed trajectory solution (low initial flight path angle, low apogee, and short TOF), while the second solution is the lofted trajectory (high initial flight path angle, etc.). This is analogous to the two solutions obtainable for the ballistic missile problem, where trajectories are sought between two fixed positions at the same altitude, with the velocity magnitude at the source point being a fixed value. In that case, it can be shown analytically that for velocities less than escape velocity and ranges less than the maximum range, two solutions (a lofted and a depressed solution) always exist. For velocities of at least escape velocity, only one solution exists. At maximum range and velocities less than escape velocity, again only one solution exists. Extending this result to more general transfer problems, Lambert's theorem states that flight between two fixed positions of different altitudes can be converted into an equivalent problem with equal altitudes, so that the above results are true between any fixed positions. It is therefore not a surprise that this property still generally holds even when considering flight between two moving vehicles.

Any TOF between these two solution times (depicted in Figure 1) is possible if the engine can be throttled or energy dissipation maneuvers are adopted. The subsequent damped oscillations of the $|\Delta v|$ curve correspond to $|\Delta v|$ requirements for second and later revs of the target trajectory. If the horizontal line at $\Delta v_g$ crosses the $|\Delta v|$ curve more than twice, it generally means that long flight times corresponding to later revs of the target are achievable with the given $\Delta v_g$ characteristic velocity. The resulting interceptor trajectories are very lofted, "hanging" until the next pass of the target. These hanging interceptor trajectories approach, but never reach, a parabolic trajectory, so the upper bound for these damped oscillations is the $|\Delta v|$ that yields escape velocity. Usually, however, later rev intercepts are not of practical importance, so the first two solutions are the only ones of interest.

If various launch times are considered, a series of plots like those shown in Figure 1 can be put together to consider the effect of launch time as well as TOF. Figure 2 shows a resultant level curve in launch-time/TOF space for the $\Delta v_g$ level curve. That is, launch-time/TOF pairs on the curve are solutions. Any launch-time/TOF pair on the interior of the region bounded by the curve requires less $|\Delta v|$ than $\Delta v_g$. Since the intercept time is the launch time plus the TOF, Figure 2 is easily converted into a launch-time/intercept-time level curve depicted in Figure 3.
DELTA-V MAGNITUDE (DU/TU)

FIGURE 1. TYPICAL Δv VERSUS TOF CURVE

FIGURE 2. TYPICAL LAUNCH-TIME TOF LEVEL CURVE
The information in Figure 3 can be used to discuss solutions to various operational requirements. First, suppose knowledge of all possible launch times is required to schedule maneuvers, taking into account external operational requirements. For a simplistic example, control facilities may need to coordinate independent maneuvers, so the union and intersection of the launch times for various vehicles may be required. In this case, all times between $t_{L,1}$ and $t_{L,2}$ in Figure 3 yield achievable intercept. Call the need to schedule launch times a Case 1 operational requirement. On the other hand, there may be a need to schedule intercept times. Again, for a simplistic example, monitoring facilities may require intercepts of various vehicles to occur at separate times, so the knowledge of all achievable intercept times between $t_{I,1}$ and $t_{I,2}$ in Figure 3 permits the coordination of intercept times. Call the need to schedule intercept times a Case 2 operational requirement. Notice that different techniques are required to solve Case 1 problems than are required for Case 2 problems, since in general $t_{L,1}$ does not correspond to $t_{I,1}$ (that is, $t_{L,1}$ is not the launch time that achieves intercept at $t_{I,1}$); this is also true for $t_{L,2}$ and $t_{I,2}$. Knowledge of the intervals $[t_{L,1}, t_{L,2}]$ and $[t_{I,1}, t_{I,2}]$ alone are not sufficient; solutions for specific launch times or specific intercept times are also required.
SIMPLE ITERATION SCHEME FOR CASE 1

The double iteration scheme proposed in this report for Case 1 operational requirements is similar to that described by Parks, et al. Since the launch time is fixed in Case 1, the position of the launcher is fixed in the attempt to find the trajectory that intercepts the moving target with the prescribed magnitude of the velocity change vector, $\Delta v$. First, a guess of the TOF of the intercept trajectory, $t$, is made, which fixes the time of intercept, and hence the position of the target at intercept. With the launch and target positions known, and with the given $t$, finding the trajectory between those positions that traverses the path between them in time $t$ is merely a Lambert problem. This Lambert problem can be solved by a variety of iteration schemes; one of the more robust algorithms is given by Battin. The Lambert iteration is the inner loop of the double iteration scheme.

The solution of Lambert's problem results in knowledge of the initial velocity vector of the intercept trajectory; the difference between this velocity and that of the launcher trajectory yields the $\Delta v$ at launch. $|\Delta v|$ is then the characteristic velocity of the interceptor required to perform this maneuver. The specified fixed characteristic velocity of the interceptor vehicle in question ($\Delta v_g$), however, is in general different from $|\Delta v|$. So, the outer loop of this iteration scheme must adjust the TOF, $t$, in order to drive the error $\delta v = |\Delta v| - \Delta v_g$ to zero. Parks, et al., proposed one scheme of updating $t$ based on an approximate derivative. In this report, a better method for updating $t$ is presented, which uses an exact derivative and is more robust over a wider domain of the problem.

Specifically, the exact derivative of $\delta v$ with respect to the TOF, $\frac{d\delta v}{dt}$, for both Case 1 and Case 2 problems, is presented. Knowledge of this derivative allows a classical Newton's method to be performed in order to iterate on $t$

$$t_{n+1} = t_n - \frac{\delta v(t_n)}{\left(\frac{d\delta v}{dt}\right)_{t_n}} \quad (1)$$

The derivation of $\frac{d\delta v}{dt}$ is given in the next section. Furthermore, knowledge of this derivative facilitates finding regions where solutions to $\delta v = 0$ exist and whether they exist for the particular problem at hand.

To illustrate this, refer again to Figure 1, which shows a typical plot of $|\Delta v|$ as a function of $t$ for a case where both the source and target are vehicles in low Earth orbit. The horizontal line corresponds to the hardware prescribed $\Delta v_g$; where the curve crosses this horizontal line corresponds to solutions to the posed intercept problem. If the first possible intercept is desired, then the depressed trajectory is chosen.
The value of $\frac{d\delta v}{dt}$ at the solution is negative for a depressed trajectory and positive for a lofted trajectory. The value of $t$ where $\frac{d\delta v}{dt} = 0$ corresponds to the minimum energy trajectory, which separates the depressed from the lofted trajectory. These observations lead to the following method for finding both depressed and lofted trajectories.

First, solve for $\frac{d\delta v}{dt} = 0$ to find the minimum achievable $|\Delta v|$, denoted $\Delta v_{\text{min}}$, occurring at TOF $t_{\text{min}}$. If $\Delta v_{\text{min}} > \Delta v_g$, then no intercept is possible at this launch time. On the other hand, if intercept is achievable, then the depressed solution occurs at some $t_{\text{dep}} \in (0, \ t_{\text{min}})$, so that the solution is bounded. A Newton iteration, Equation 1, generally finds $t_{\text{dep}}$ swiftly when the initial guess $t_0 \in (0, \ t_{\text{min}})$. It has been found advantageous to augment the Newton iteration with checks to assure that convergence is occurring. A switch to some safe bracketing algorithm such as bisection is recommended in the rare event when the Newton iteration is not converging. To find the lofted trajectory, Newton's method can be used with an initial guess $t_0 > t_{\text{min}}$. Again, checks should be implemented to ensure convergence. A test to assure that $\frac{d\delta v}{dt} = 0$ will also help guarantee that the convergence is occurring in the correct region.

If the minimum possible TOF is the one that is always desired, then starting with a short TOF as the initial guess and iterating with Newton's method until convergence usually works very well. If $\frac{d\delta v}{dt}$ becomes positive in the iteration process, then the minimum energy solution has been passed, and it can be assumed that the minimum $|\Delta v|$ is greater than $\Delta v_g$. That is, no solution is possible. If a solution is possible, the iteration usually finds it rapidly.

**CASE 1 DERIVATIVE**

Since $|\Delta v| = \sqrt{\Delta v \cdot \Delta v}$, the desired derivative of the error in $\Delta v$ with respect to TOF is

$$
\frac{d\delta v}{dt} = \frac{d}{dt}(|\Delta v| - \Delta v_g) = \frac{d}{dt} \sqrt{\Delta v \cdot \Delta v} = \frac{\Delta v \cdot d\Delta v}{|\Delta v|} \cdot \frac{d\Delta v}{dt} \quad (2)
$$
Furthermore, \( \frac{d\Delta v}{dt} = \frac{dv_o}{dt} \) (where \( v_o \) denotes the initial velocity of the intercept trajectory), because the velocity of the source vehicle is fixed with respect to \( t \). So

\[
\frac{d\delta v}{dt} = \frac{\Delta v}{|\Delta v|} \cdot \frac{dv_o}{dt}
\]

This means that the derivative \( \frac{dv_o}{dt} \) is required. For notational convenience, let subscript “1” denote quantities related to the source trajectory, let subscript “2” denote quantities related to the target trajectory, and let subscripts “o” and “f” denote initial and final quantities of the intercept trajectory. Then, as long as \( r_1 \) and \( v_o \) are linearly independent (i.e., the interceptor trajectory is not rectilinear), the final intercept trajectory position can be written as a linear combination of these two vectors

\[
r_2 = f r_1 + g v_o
\]

The scale factors are the so-called “f and g expressions,” which are well-known functions of the intercept geometry. Specifically

\[
f = 1 + \frac{r_2}{p} (\cos \Delta v - 1)
\]

and

\[
g = \frac{r_1 r_2 \sin \Delta v}{\sqrt{\mu p}}
\]

where \( r_1 \) and \( r_2 \) are magnitudes of \( r_1 \) and \( r_2 \), \( \Delta v \) is the angle between \( r_1 \) and \( r_2 \), \( p \) is the semi-latus rectum of the intercept orbit, and \( \mu \) is the gravitational parameter. Solving Equation 4 for \( v_o \) yields

\[
v_o = \frac{r_2 - fr_1}{g}
\]

This gives the derivative

\[
\frac{dv_o}{dt} = \left( v_2 - \frac{df}{dt} \right) g - (r_2 - fr_1) \frac{dg}{dt} \]

In turn, the derivatives \( \frac{df}{dt} \) and
$d^2 \theta$ must be determined. Differentiating Equations 5 and 6 yields

$$\frac{df}{dt} = \left(\frac{\dot{r}^2 + \frac{d}{dt} \frac{dp}{dt}}{p^2}\right) (\cos \Delta v - 1) - \frac{r^2}{p} \sin \Delta v \frac{d\Delta v}{dt}$$

and

$$\frac{dg}{dt} = \left(\frac{r^2 \sin \Delta v + r \cos \Delta v}{\sqrt{\mu p}}\right) \frac{d\Delta v}{dt} - \frac{r_1 r_2 \sin \Delta v}{2 \sqrt{\mu p^3}} \frac{dp}{dt}$$

Since $\cos \Delta v = \frac{r_1 \cdot r_2}{r_1 r_2}$

$$\frac{d\Delta v}{dt} = \frac{\dot{r}_2 (r_1 \cdot r_2) - r_2 (r_1 \cdot \dot{r}_2)}{r_1 r_2 \sin \Delta v}$$

Now the only undetermined derivative is $\frac{dp}{dt}$. To find this derivative, a relationship between the semi-latus rectum, "p," of the intercept orbit and the TOF must be found. The TOF equation as a function of orbital parameters, which in turn are implicit functions of "p," gives the desired relationship. For elliptical trajectories, Bate, et al., give the TOF for the intercept trajectory

$$\tau = \frac{1}{2} \int_{\Delta E}^{\Delta E - \sin \Delta E} \frac{a^3}{\mu} \, d\Delta E$$

where $a$ is the semimajor axis and $\Delta E$ is the change in eccentric anomaly of the intercept trajectory. Each of the three variables on the right side of the equation is a function of "p," so that implicit differentiation of both sides with respect to $\tau$ will cause $\frac{dp}{dt}$ to appear on the right side of the equation. $\frac{dp}{dt}$ can then be solved for algebraically.

To this end, first cast $g$, $a$, and $\Delta E$ in terms of geometric parameters and $p$ (and previously defined functions of these quantities)

$$g = \frac{r_1 r_2 \sin \Delta v}{\sqrt{\mu p}}$$
\[ a = \frac{m k p}{(2m - l^2) p^2 + 2k l p - k^2} \]  

(14)

and

\[ \cos^2\left( \frac{\Delta E}{2} \right) = \frac{(k - l p)^2}{2m p^2} \]  

(15)

Here the auxiliary geometry functions \( k, l, \) and \( m \) are

\[ k = r_1 r_2 (1 - \cos \Delta \nu) \]  

(16)

\[ l = r_1 + r_2 \]  

(17)

and

\[ m = r_1 r_2 (1 + \cos \Delta \nu) \]  

(18)

Hence, Equation 12 can be written functionally as

\[ \tau = F\left( g\left[ r_2(t), \Delta \nu(t), p(t) \right], a\left[ k(t), l(t), m(t), p(t) \right], \Delta E\left[ k(t), l(t), m(t), p(t) \right] \right) \]  

(19)

Implicitly differentiating Equation 19

\[ \frac{dt}{dt} = 1 = \frac{\partial F}{\partial g} \left( \frac{\partial g}{\partial r_2} r_2 + \frac{\partial g}{\partial \Delta \nu} \frac{d \Delta \nu}{dt} + \frac{\partial g}{\partial p} \frac{dp}{dt} \right) \]

\[ + \frac{\partial F}{\partial \alpha} \left( \frac{\partial \alpha}{\partial k} \frac{dk}{dt} + \frac{\partial \alpha}{\partial l} \frac{dl}{dt} + \frac{\partial \alpha}{\partial m} \frac{dm}{dt} + \frac{\partial \alpha}{\partial p} \frac{dp}{dt} \right) \]

\[ + \frac{\partial F}{\partial \Delta E} \left( \frac{\partial \Delta E}{\partial k} \frac{dk}{dt} + \frac{\partial \Delta E}{\partial l} \frac{dl}{dt} + \frac{\partial \Delta E}{\partial m} \frac{dm}{dt} + \frac{\partial \Delta E}{\partial p} \frac{dp}{dt} \right) \]

(20)

Solving Equation 20 for \( \frac{dp}{dt} \)

\[ \frac{dp}{dt} = D \left\{ 1 - \frac{\partial F}{\partial g} \left( \frac{\partial g}{\partial r_2} r_2 + \frac{\partial g}{\partial \Delta \nu} \frac{d \Delta \nu}{dt} \right) \right\} \]

\[ - \frac{\partial F}{\partial \alpha} \left( \frac{\partial \alpha}{\partial k} \frac{dk}{dt} + \frac{\partial \alpha}{\partial l} \frac{dl}{dt} + \frac{\partial \alpha}{\partial m} \frac{dm}{dt} \right) \]

\[ - \frac{\partial F}{\partial \Delta E} \left( \frac{\partial \Delta E}{\partial k} \frac{dk}{dt} + \frac{\partial \Delta E}{\partial l} \frac{dl}{dt} + \frac{\partial \Delta E}{\partial m} \frac{dm}{dt} \right) \}

(21)
where

\[ D = \frac{1}{\frac{\partial F}{\partial p} + \frac{\partial F}{\partial a} + \frac{\partial F}{\partial \Delta E}} \]  (22)

Each of the derivatives in Equations 21 and 22 are easily derived by straightforward differentiation of formulas given above.

Now Equation 21 can be used in Equations 9 and 10, together with Equation 11, to compute Equation 8. Equation 8, in turn, is used in Equation 3 to obtain the desired derivative to be used in Newton’s method, Equation 1.

If the current intercept trajectory is hyperbolic, then \( \frac{d\delta v}{dt} \) must be determined with the TOF equation corresponding to hyperbolic flight

\[ t = g + \sqrt{-\frac{a^3}{\mu}} (\sinh \Delta F - \Delta F) \]  (23)

where \( \Delta F \) is the hyperbolic eccentric anomaly. Let the right side of this equation be called \( G \). Then, analogous to Equation 21, the same procedure is used to find

\[ \frac{dp}{dt} = D_h \left[ 1 - \frac{\partial G}{\partial \dot{r}} \left( \frac{\partial \dot{r}}{\partial a} \frac{d\Delta v}{dt} + \frac{\partial \dot{r}}{\partial \Delta E} \frac{d\Delta v}{dt} \right) - \left( \frac{\partial G}{\partial a} + \frac{\partial G}{\partial \Delta E} \frac{\partial \Delta E}{\partial a} \right) \left( \frac{\partial a}{\partial k} \frac{dk}{dt} + \frac{\partial a}{\partial l} \frac{dl}{dt} + \frac{\partial a}{\partial m} \frac{dm}{dt} \right) \right. \]

\[ \left. \frac{-\partial G}{\partial \Delta F} \frac{\partial \Delta F}{\partial \dot{r}} \left( \frac{\partial \dot{r}}{\partial a} \frac{d\Delta v}{dt} + \frac{\partial \dot{r}}{\partial \Delta E} \frac{d\Delta v}{dt} \right) \right] \]  (24)

where

\[ D_h = \frac{1}{\frac{\partial G}{\partial p} + \frac{\partial G}{\partial a} + \frac{\partial G}{\partial \Delta E} \left( \frac{\partial \Delta E}{\partial a} \frac{\partial a}{\partial p} + \frac{\partial \Delta E}{\partial \Delta E} \frac{\partial \Delta E}{\partial p} \right)} \]  (25)

The only new function here, other than \( G \), is that relating \( \Delta F \) to intercept geometry parameters (and its derivatives)

\[ \cosh \Delta F = 1 - \frac{r_1}{a} (1 - f) \]  (26)
CASE 2 PROCEDURES

For Case 2 operational requirements (find the launch times to intercept at a fixed intercept time), a slightly modified iteration scheme must be employed. In this case, since the intercept time (and hence the intercept position) is fixed, a variable TOF means that the launch location is moving as a function of \( t \). For example,

\[
\frac{dr}{dt} = -\frac{dr}{dt} = -v, \text{ if } r \text{ is the position of the source vehicle, since an increase in } t \text{ means that the time of launch is decreasing.}
\]

The most straightforward method of handling this problem would be to start the above process again, but now allowing \( r_1 \) to vary as a function of \( t \) instead of \( r_2 \).

An alternative method, which makes use of the symmetry of Keplerian orbits, takes advantage of most of the Case 1 derivatives. Consider a trajectory from a starting point to a terminal point. If a trajectory is flown from the terminal point with the negative of the terminal velocity, the same trajectory will be traversed in the opposite direction, reaching the starting point in the same TOF with a velocity equal to the negative of the starting velocity. So, the problem of finding launch times to intercept at a fixed intercept time can be recast into a problem where the launch orbit becomes the target orbit and the target orbit becomes the launch orbit. The only adjustment that must be made is that all velocities involved become the negative of those in the real problem. For the rest of this section, the notation reflects the recast problem (for example, \( r_2 \) in the recast problem is really the launch position in the starting problem, \( v_2 \) is the negative of the starting interceptor velocity, etc.).

The only difference between the recast problem and the Case 1 problem is that the prescribed velocity, \( \Delta v_e \), must now be matched at the terminal point instead of the starting point. Using the notation from Case 1 for the recast problem, the requirement is, given that

\[
\Delta v_f = v_f - v_2
\]

\[
\delta v_f \Delta |\Delta v_f| - \Delta v_e = 0
\]

The requisite derivative for Newton's method of finding the zero of the \( \delta v_f \) function is

\[
\frac{d \delta v_f}{dt} = \frac{d}{dt} \left( |\Delta v_f| - \Delta v_e \right) = \frac{d}{dt} \frac{\sqrt{\Delta v_f \cdot \Delta v_f}}{|\Delta v_f|} = \frac{\Delta v_f}{|\Delta v_f|} \cdot \frac{d \Delta v_f}{dt}
\]

In turn

\[
\frac{d \Delta v_f}{dt} = \frac{d v_f}{dt} - \frac{d v_2}{dt}
\]
But since the change in $v_2$ in TOF is the same as the change in $v_2$ in time, i.e.,
\[
\frac{dv_2}{dt} = \frac{dv_2}{dt},
\]
its derivative is just the gravitational acceleration. So
\[
\frac{d\Delta v_f}{dt} = \frac{dv_f}{dt} + \frac{\mu r_2}{|r_2|^3}
\]  \hspace{1cm} (31)

To find the derivative of $v_f$, write it as
\[
v_f = f r_1 + g v_o
\]  \hspace{1cm} (32)
Then
\[
\frac{dv_f}{dt} = \frac{df}{dt} r_1 + \frac{dg}{dt} v_o + g \frac{dv_o}{dt}
\]  \hspace{1cm} (33)

Note that $\frac{dv_o}{dt}$ has already been determined for Case 1 problems, so the remaining derivatives to be determined are the derivatives of $f$ and $g$. The equation for $\dot{g}$ is
\[
\dot{g} = 1 - \frac{r_1}{p} (1 - \cos \Delta v)
\]  \hspace{1cm} (34)
so that
\[
\frac{dg}{dt} = -\frac{r_1}{p^2} (1 - \cos \Delta v) \frac{dp}{dt} - \frac{r_1}{p} \sin \Delta v \frac{d\Delta v}{dt}
\]  \hspace{1cm} (35)

Everything on the right side of the equation has already been determined in the previous section. As for $f$, the identity
\[
1 = f g - f \dot{g}
\]  \hspace{1cm} (36)
can be differentiated to yield
\[
\frac{df}{dt} = \frac{1}{g} \left( f \frac{dg}{dt} + \dot{g} \frac{df}{dt} - f \frac{dg}{dt} \right)
\]  \hspace{1cm} (37)
Again, all other derivatives have already been determined in the previous section.
EXAMPLES AND COMPARISONS

Various examples of the intercept algorithm used in the Antitactical Ballistic Missile Global Effectiveness Model (AGEM) follow. This model has been used in various configurations to simulate orbiter-to-orbiter, ground-to-orbiter, and ground-to-ground engagements for Strategic Defense Initiative (SDI), antisatellite (ASAT) and antitactical ballistic missile (ATBM) scenarios. The examples that follow are exclusively orbiter-to-orbiter intercepts.

For the orbiting satellite that will launch the interceptors in this series of examples, the initial inertial position and velocity vectors are

\[
\begin{align*}
    r_s &= \begin{bmatrix} 7015.9485 \\ 0.0 \\ 0.0 \\ 0.0 \\ v_s &= \begin{bmatrix} 7.5496259965 \\ 0.0 \\
\end{align*}
\]

The starting inertial position and velocity vectors of the orbiting target are

\[
\begin{align*}
    r_t &= \begin{bmatrix} 5740.3215 \\ 3189.0675 \\ -3189.0675 \\ v_t &= \begin{bmatrix} 2.82181264515 \\ 1.5676736918 \\ 6.64693645322 \\
\end{align*}
\]

Figure 4 shows the \(|\Delta v|\) and \(t\) solution space for interceptors launched at time 0.0 (corresponding to the initial \(r_s\)). The minimum velocity intercept solution is

\[
\begin{align*}
    |\Delta v| &= 0.649912 \text{ km/sec} \\
    v_o &= 7.989828 \text{ km/sec} \\
    t &= 462.3349 \text{ sec}
\end{align*}
\]

In each of the eight examples presented, the required intercept impulse is \(\Delta v_g = 3.95268 \text{ km/sec}\), and the flight time is constrained between 0 and 3000 sec. The 3000-sec limit has been arbitrarily imposed to keep engagement solutions inside a single-orbit revolution of the target. Algorithm iteration convergence tolerances are 0.00001 km/sec for velocity and 0.001 sec for flight time. For these parameters, there are two possible intercept trajectories; one depressed with a flight time of 338.9 sec and one lofted with a flight time of 705.4 sec.
Table 1 summarizes the examples and shows the type of trajectory to which the algorithm converged and the number of iterations required. Figure 5 shows the constraints imposed on each example. For each of the example convergence data tables, n refers to the iteration number and n, refers to the number of iterations required to achieve convergence of the Lambert iteration (inner loop). The flight time and impulse for each step are also shown. The last row in the table presents the last significant iteration trial. The actual final pass through the algorithm yields identical results, to the accuracy presented. In each table, the initial flight time guess (n = 1) is the first row.

Examples 1 and 2 demonstrate the classic use of the algorithm where either the lofted (maximum flight time) or depressed (minimum flight time) trajectory is acceptable. Example 1 uses an initial flight time guess well below the depressed solution, while Example 2 uses an initial guess well above the lofted solution. Convergence data for Example 1 are shown in Table 2. Example 2 data are listed in Table 3.

Example 3, which uses the algorithm specified by Parks, et al., is presented for convergence characteristics comparison and for completeness. Table 4 contains data for this example.
TABLE 1. EXAMPLE SUMMARY. FLIGHT TIME LIMITS AND CONVERGENCE RESULTS

<table>
<thead>
<tr>
<th>Ex.</th>
<th>Eq Type</th>
<th>Case</th>
<th>Min (sec)</th>
<th>Guess (sec)</th>
<th>Max (sec)</th>
<th>Trajectory Type</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New</td>
<td>1</td>
<td>1.0</td>
<td>10.0</td>
<td>3000.0</td>
<td>Depressed</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>New</td>
<td>1</td>
<td>1.0</td>
<td>1500.5</td>
<td>3000.0</td>
<td>Depressed</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Parks</td>
<td>1</td>
<td>1.0</td>
<td>10.0</td>
<td>3000.0</td>
<td>Depressed</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>New</td>
<td>1</td>
<td>1.0</td>
<td>3.0</td>
<td>462.3</td>
<td>Depressed</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>New</td>
<td>1</td>
<td>1.0</td>
<td>231.7</td>
<td>462.3</td>
<td>Depressed</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>New</td>
<td>1</td>
<td>462.3</td>
<td>464.3</td>
<td>3000.0</td>
<td>Lofted</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>New</td>
<td>1</td>
<td>463.3</td>
<td>1731.7</td>
<td>3000.0</td>
<td>Lofted</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>New</td>
<td>2</td>
<td>1.0</td>
<td>10.0</td>
<td>3000.0</td>
<td>Depressed</td>
<td>16</td>
</tr>
</tbody>
</table>

Flight time bounds
- Examples 1, 2, 3 & 8
- Examples 4 & 5
- Examples 6 & 7

FIGURE 5. EXAMPLE CONSTRAINTS
### TABLE 2. EXAMPLE 1: NEW ALGORITHM - CASE 1

(1 ≤ \(t\) ≤ 3000)

| n  | \(n_t\) | \(t\) (sec) | |\(\Delta v\)| (km/sec) |
|----|--------|-------------|-----------|---------------|
| 1  | 12     | 10.0000     |           | 459.318       |
| 2  | 10     | 19.7153     |           | 228.333       |
| 3  | 9      | 38.3218     |           | 112.882       |
| 4  | 8      | 72.4360     |           | 55.2278       |
| 5  | 7      | 129.734     |           | 26.5586       |
| 6  | 6      | 210.451     |           | 12.5754       |
| 7  | 5      | 290.762     |           | 6.30517       |
| 8  | 4      | 332.161     |           | 4.24263       |
| 9  | 5      | 338.804     |           | 3.95835       |
| 10 | 5      | 338.939     |           | 3.95273       |
| 11 | 12     | 338.940     |           | 3.95266       |
| 12 | 8      | 338.940     |           | 3.95268       |

### TABLE 3. EXAMPLE 2: NEW ALGORITHM - CASE 1

(1 ≤ \(t\) ≤ 3000)

| n  | \(n_t\) | \(t\) (sec) | |\(\Delta v\)| (km/sec) |
|----|--------|-------------|-----------|---------------|
| 1  | 5      | 1500.50     |           | 8.71621       |
| 2  | 6      | 196.303     |           | 14.2024       |
| 3  | 5      | 279.510     |           | 6.96989       |
| 4  | 5      | 328.640     |           | 4.39801       |
| 5  | 7      | 338.626     |           | 3.96587       |
| 6  | 9      | 338.940     |           | 3.95271       |
| 7  | 15     | 338.940     |           | 3.95269       |
| 8  | 5      | 338.941     |           | 3.95270       |
| 9  | 5      | 338.941     |           | 3.95265       |
| 10 | 12     | 338.940     |           | 3.95266       |
| 11 | 8      | 338.940     |           | 3.95268       |
TABLE 4. EXAMPLE 3: PARKS, ET AL. - CASE 1
(1 ≤ t ≤ 3000)

| n | n₁ | t (sec) | |Δv| (km/sec) |
|---|----|---------|-------------|
| 1 | 12 | 10.0000 | 459.318 |
| 2 | 10 | 19.8032 | 227.278 |
| 3 | 9  | 38.8237 | 111.301 |
| 4 | 8  | 74.5508 | 53.3897 |
| 5 | 7  | 137.058 | 24.6163 |
| 6 | 6  | 229.529 | 10.6950 |
| 7 | 5  | 318.122 | 4.88246 |
| 8 | 4  | 341.971 | 3.82679 |
| 9 | 7  | 338.225 | 3.98275 |
|10 | 5  | 339.099 | 3.94600 |
|11 | 4  | 338.904 | 3.95416 |
|12 | 8  | 338.948 | 3.95234 |
|13 | 5  | 338.938 | 3.95279 |
|14 | 7  | 338.941 | 3.95266 |
|15 | 7  | 338.940 | 3.95267 |
|16 | 5  | 338.940 | 3.95271 |
|17 | 5  | 338.940 | 3.95262 |
|18 | 5  | 338.939 | 3.95272 |
|19 | 5  | 338.940 | 3.95266 |
|20 | 7  | 338.940 | 3.95268 |

Examples 4 and 5 have been forced to converge on the depressed trajectory by virtue of the flight time limits used. Example 4 has specified an unreasonably small initial flight time guess, while Example 5 has specified an initial guess very near the solution. Example 4 data are listed in Table 5, and Example 5 data are given in Table 6.

Examples 6 and 7 have been forced to find the lofted solution because of the minimum flight time requirements. The initial guess in Example 6 is lower than the expected lofted trajectory solution, while that in Example 7 is above the expected lofted trajectory solution. Data for these examples are listed in Tables 7 and 8, respectively. Notice that in both these examples the algorithm makes use of the implemented safe bracketing method to bring the flight time into the neighborhood of the known solution.
**TABLE 5. EXAMPLE 4: NEW ALGORITHM - CASE 1**
(1 ≤ t ≤ 462.3)

| n  | nᵢ   | t (sec) | |Δv| (km/sec) |
|----|-------|---------|-------------|
| 1  | 14    | 3.00000 | 1552.95     |
| 2  | 12    | 4.97440 | 775.044     |
| 3  | 11    | 11.8480 | 386.204     |
| 4  | 10    | 23.2963 | 191.790     |
| 5  | 9     | 45.0426 | 94.6228     |
| 6  | 8     | 84.2637 | 46.1264     |
| 7  | 7     | 147.999 | 22.0711     |
| 8  | 6     | 232.049 | 10.4694     |
| 9  | 5     | 305.637 | 5.50047     |
| 10 | 4     | 335.90 | 4.09221     |
| 11 | 5     | 338.909 | 3.95398     |
| 12 | 14    | 338.940 | 3.95268     |

**TABLE 6. EXAMPLE 5: NEW ALGORITHM - CASE 1**
(1 ≤ t ≤ 462.3)

| n  | nᵢ | t (sec) | |Δv| (km/sec) |
|----|-----|---------|-------------|
| 1  | 6   | 231.668 | 10.5033     |
| 2  | 5   | 305.398 | 5.51278     |
| 3  | 5   | 335.644 | 4.09221     |
| 4  | 6   | 338.908 | 3.95404     |
| 5  | 5   | 338.940 | 3.95262     |
| 6  | 5   | 338.939 | 3.95272     |
| 7  | 5   | 338.940 | 3.95274     |
| 8  | 6   | 338.941 | 3.95264     |
| 9  | 12  | 338.940 | 3.95266     |
| 10 | 8   | 338.940 | 3.95268     |
### TABLE 7. EXAMPLE 6: NEW ALGORITHM – CASE 1

(462.3 ≤ t ≤ 3000)

| n | n_i | t (sec) | |Δv| (km/sec) |
|---|---|---|---|---|
| 1 | 3 | 464.335 | 0.65162 |
| 2 | 4 | 2415.38 | 11.5062 |
| 3 | 3 | 462.335+ | 0.64991 |
| 4 | 5 | 1731.17* | 9.49904 |
| 5 | 5 | 1308.22* | 7.95633 |
| 6 | 6 | 1096.75* | 6.93482 |
| 7 | 3 | 551.024 | 1.89556 |
| 8 | 5 | 679.003 | 3.65531 |
| 9 | 5 | 704.564 | 3.94370 |
| 10 | 5 | 705.385 | 3.95268 |

**NOTE:** The following indicate that the algorithm’s flight time guess is outside the externally imposed flight time limits:

+ = reset Tff guess to Tff min.
* = using bisection to reset Tff guess

### TABLE 8. EXAMPLE 7: NEW ALGORITHM – CASE 1

(463.3 ≤ t ≤ 3000)

| n | n_i | t (sec) | |Δv| (km/sec) |
|---|---|---|---|---|
| 1 | 5 | 1731.67 | 9.50062 |
| 2 | 3 | 463.335+ | 0.65034 |
| 3 | 5 | 1731.67* | 9.50062 |
| 4 | 5 | 1308.89* | 7.95919 |
| 5 | 6 | 1097.50* | 6.93892 |
| 6 | 3 | 550.488 | 1.88695 |
| 7 | 5 | 678.856 | 3.65359 |
| 8 | 5 | 704.555 | 3.94360 |
| 9 | 5 | 705.385 | 3.95268 |

**NOTE:** The following indicate that the algorithm’s flight time guess is outside the externally imposed flight time limits:

+ = reset Tff guess to Tff min.
* = using bisection to reset Tff guess
Example 8 demonstrates the "backward" (Case 2) solution of the minimum flight time intercept. In this example, the interceptor is "launched" from \( r_l \) propagated forward to 338 sec (the "forward" depressed trajectory impact time). The algorithm is required to find a solution that has an "impact" closing velocity equal to the required velocity. In this "backward" intercept, flight time is effectively negative, so that "impact" at \( r_s \) is before "launch" from \( r_l \). Algorithm convergence data are presented in Table 9.

**TABLE 9. EXAMPLE 8: NEW ALGORITHM - CASE 2**

\( (1 \leq t \leq 3000) \)

| n  | \( n_t \) | \( t \) (sec) | \( |\Delta v| \) (km/sec) |
|----|----------|--------------|------------------------|
| 1  | 9        | 10.0000      | 131.124                |
| 2  | 8        | 19.6991      | 66.5680                |
| 3  | 7        | 38.2312      | 34.3066                |
| 4  | 6        | 72.0749      | 16.2096                |
| 5  | 5        | 128.610      | 10.2259                |
| 6  | 5        | 207.990      | 6.35474                |
| 7  | 5        | 278.969      | 4.62525                |
| 8  | 6        | 331.124      | 4.04192                |
| 9  | 6        | 338.754      | 3.95473                |
| 10 | 5        | 338.937      | 3.95274                |
| 11 | 7        | 338.942      | 3.95266                |
| 12 | 16       | 338.941      | 3.95265                |
| 13 | 5        | 338.937      | 3.95275                |
| 14 | 5        | 338.943      | 3.95264                |
| 15 | 5        | 338.939      | 3.95270                |
| 16 | 16       | 338.941      | 3.95268                |

Though the algorithm presented in this report is substantially more complex than that presented by Parks, et al., this new algorithm converges to a solution more quickly. Examination of the number of iterations required for Examples 1 and 3 in Table 1 demonstrates that the increase in complexity (lines of code) is offset by more rapid convergence characteristics.

Examples 1, 4, and 6 demonstrate how judicious specification of intercept flight time bounds can be used to choose between maximum and minimum intercept flight time solutions. In Example 1, the time limits indicate that there is no real preference of trajectory shape. The bounds set in Examples 4 and 6 force the algorithm to find only the depressed or lofted trajectory shapes, respectively. However, knowledge of the solution space minimum (Figure 4) is required.
Comparison of Examples 1 and 2 shows that the algorithm converged to the depressed trajectory in both examples. This is due to the slope of the curve past the neighborhood of the lofted solution. This value tends to correct the succeeding flight time iteration to a value near or below the depressed solution, which is then converged to. This is a manifestation of the geometry of this example and of the initial flight time guess. Figures 6 and 7 are graphical representations of the convergence behavior for Examples 1 and 2, respectively.

Examples 6 and 7 show the necessity of implementing an auxiliary convergence method, such as a bisection method. In both these examples, the value of the slope at the initial flight time guess causes the next guess by the standard iteration method to fall outside the allowed limits. At this point, as shown in Figure 8 (Example 7), a bisection method can be used to force subsequent flight time iterations into the neighborhood of the solution, at which point the presented equations take over and converge.

In summary, the algorithm presented in this report is robust. In the presence of stressing engagement geometries, Newton's iteration requires augmentation using some safe bracketing method to guarantee convergence within the specified flight time bounds.

FIGURE 6. EXAMPLE 1 CONVERGENCE
FIGURE 7. EXAMPLE 2 CONVERGENCE

FIGURE 8. EXAMPLE 7 CONVERGENCE
CONCLUSIONS

The new algorithm for solving single-impulse intercept problems is a significant improvement over existing methods for three reasons. First, it has the intrinsic ability to find either or both of the solutions that exist in the typical intercept scenario, as opposed to finding only the minimum time solutions as extant methods do. Second, the iteration scheme converges rapidly. Third, the algorithm can solve for either set launch times or set intercept times depending on whether launch times or intercept times need to be scheduled. These attributes combine to produce an algorithm that is both flexible in its ability to handle a wide variety of intercept problems and robust in its capacity to rapidly converge to appropriate solutions. It has proven valuable in diverse space and naval defense analyses.

REFERENCES


DISTRIBUTION

Copies

DOD ACTIVITIES (CONUS)
ATTN SEA 06 1
SEA 06R
COMMANDER
NAVAL SEA SYSTEMS COMMAND
2531 JEFFERSON DAVIS HWY
ARLINGTON VA 22242-5760

ATTN CODE 12
COMMANDER
OFFICE OF CHIEF OF NAVAL RESEARCH
APPLIED RESEARCH AND TECHNOLOGY
DIRECTORATE
800 N QUINCY ST
ARLINGTON VA 22217-5000

ATTN CODE 30
OFFICE OF CHIEF OF NAVAL RESEARCH
OFFICE FOR APPLIED TECHNOLOGY
800 N QUINCY ST
ARLINGTON VA 22217-5000

ATTN CODE 005
COMMANDER
SPACE AND NAVAL WARFARE
SYSTEMS COMMAND
2451 CRYSTAL GATEWAY BLDG 5
WASHINGTON DC 20362-5222

ATTN T SOLER
NGS/NOS/NOAA N/C
6001 EXECUTIVE BLVD
ROCKVILLE MD 20852

ATTN CODE 8000
NAVAL CENTER FOR SPACE
TECHNOLOGY
COMMANDING OFFICER
NAVAL RESEARCH LABORATORY
4555 OVERLOOK DR SW
WASHINGTON DC 20375-5320

ATTN CODE 37
COMMANDER
NAVAL AIR WARFARE CENTER
CHINA LAKE CA 93555-6001

DEFENSE TECHNICAL INFORMATION
CENTER
CAMERON STATION
ALEXANDRIA VA 22304-6145

NON-DOD ACTIVITIES
ATTN F H LUTZE
DEPARTMENT OF AEROSPACE AND
OCEAN ENGINEERING
VIRGINIA TECH
BLACKSBURG VA 24061

ATTN LUIGI PERINI
APPLIED PHYSICS LABORATORY
JOHNS HOPKINS UNIVERSITY
JOHNS HOPKINS RD
LAUREL MD 20723-6099

ATTN GIFT & EXCHANGE DIVISION
LIBRARY OF CONGRESS
WASHINGTON DC 20540

INTERNAL
E231 3
E282 (GRAY)
G205 1
G21 1
G23 1
K13 20
K14 5
K105 1
K106 1
K107 1
K40 1
K44 1
N74 (GIDEP) 1

(1)
This report discusses general solutions to the spaceflight intercept problem between moving source and target vehicles, under the constraint that only one limited thrust burn is permitted. This one burn is approximated as an instantaneous change in velocity, \( \Delta v \). Given first is a discussion of the solution space, aided by Lambert's theorem. Generally, two solutions (and sometimes more) to the posed intercept problem exist for a given launch time. Although previous solution techniques in the literature concentrate on finding the minimum time solution, both solutions are significant from an operational and a theoretical standpoint. Next, exact derivatives for two different operational requirements are derived, which facilitate finding all solutions to the intercept problem. Finally, example numerical problems solved by the Antitactical Ballistic Missile Global Effectiveness Model (AGEM) are presented to demonstrate the solution process for both operational requirements.
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.

<table>
<thead>
<tr>
<th>Block 1. Agency Use Only (Leave blank).</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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).</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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:</td>
</tr>
<tr>
<td>C - Contract</td>
</tr>
<tr>
<td>G - Grant</td>
</tr>
<tr>
<td>PE - Program</td>
</tr>
<tr>
<td>Element</td>
</tr>
</tbody>
</table>

**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.