MATHEMATICAL FORMULATION OF A FAST-TIME GEOMETRIC
HEADING NAVIGATION MODEL
Gerard T. Fairley and Seamus M. McGovern
U.S. DOT National Transportation Systems Center, Cambridge, Massachusetts

Abstract
This paper introduces a new mathematical method for reckoning paths of constant heading along an oblate ellipsoidal surface (e.g., the Earth) and for determining the distances of those paths. The method is particularly fast and accurate, lending itself to use in computationally intensive computer applications, including fast-time aviation simulations and any navigation-related Monte Carlo simulations, where fast execution times and position accuracy are both desirable. This method performs especially well for paths having a large distance and for headings that include changes in both latitude and longitude. In the paper, the proposed method and a traditional “exact” method are derived and their fundamental governing equations provided. Two other methods are also presented for comparison. All methods can produce a final position given an initial position, heading, and distance to be traveled. These methods can also be used to find the distance to a line of longitude or latitude, given a heading and an initial position. For each of four, an example of total path distance is calculated (two of the methods have known, precise World Geodetic System 84 results which are used). These total path distances are then compared to each other for accuracy and in terms of the required calculation execution times.

Introduction
In this paper, four different numerical methods are used to obtain the distance along Earth’s surface for a constant heading path. They can also be used to reckon a final position given an initial position, a heading angle, and a path distance. One method is new, the geodetic heading navigation algorithm (HEADING). For reckoning a constant heading path of general heading angle or determining path distances, HEADING is preferable to the other models if both speed and accuracy are desired. Two of the other methods are well-known: the geodesic navigation algorithm (GEODESIC) and the ‘exact’ heading navigation algorithm (EXACT). The remaining method, a simple heading navigation algorithm (SIMPLE), was presented by McGovern et al. [1]. The constant heading path starts at one point on Earth’s surface and ends at some other point on Earth’s surface and at each intermediate sampled point the orientation of the path relative to the local longitude is constant and has the same value as the initial heading. The geodesic path, in contrast to the constant heading path, has constant azimuth, and by definition has minimum length.

GEODESIC, HEADING, and EXACT model Earth using the World Geodetic System 1984 (WGS-84) [2]. For these models, Earth’s surface is approximated as an oblate spheroid and will hereafter be referred to as an ellipsoid. GEODESIC has essentially zero error for obtaining geodesics but can have both small and large errors in determining constant heading paths depending on its mode of operation (geodesic or heading). SIMPLE is a model having non-compliance with WGS-84 but whose algorithm has the least complexity of any of the models. However, this algorithm suffers from two sources of error that are discussed below. HEADING and EXACT have only one source of error and thus they may be more accurate than SIMPLE but perhaps less accurate than GEODESIC. GEODESIC and HEADING provide complementary methods of navigation.

The HEADING navigation technique starts from a known position and requires that at each step the orientation of the line of longitude is exactly known and that the heading can be determined exactly relative to this longitude. And then for a desired total distance traveled, predicts the final destination. This HEADING algorithm is an original numerical algorithm for staying on a course of constant heading, for any heading, with an error only slightly greater than that of EXACT. The EXACT algorithm can not be solved in closed form but the exact first order differential equation is known and is solved numerically. However, HEADING is generally faster than EXACT and frequently is an order of magnitude faster than EXACT.
The exact differential line element for the ellipsoid is typically derived using equivalent elliptical geometry defined by an auxiliary circle with parametric latitude. In contrast, this paper uses geocentric latitude to derive this differential line element. The metrics yield equivalent numerical results and the differential longitudes yield equivalent numerical results. These results are expressed in terms of the geodetic latitude.

This paper gives heading navigation results in the limit where practical errors do not exist. There are many applied, or practical, techniques for navigational guidance built with various sensors (see Pratap and Per [3], Beverly [4], Chowdhary [5], Anthony [6], and Bekir [7]). These techniques possess errors inherent in the instrumentation (GPS, gyroscopes, inertial guidance systems, etc.) or in the parameters (e.g., gravitational field) used for determining position and heading. In addition, the environments in which these practical techniques are used can limit the distances over which the approximate constant heading paths can be achieved. The scope of this paper does not consider how to correct these errors and limitations.

Pratap and Per [3] state, “Navigation is the art and science of charting a course from point A to point B and staying on that course.” Staying exactly on a constant heading course can not be mathematically solved in closed form and numerical methods must be used instead. Thus, a course of constant heading is not practically possible since only a differential position change could stay exactly on the desired heading as is shown in the following section. The case of traversing geodesics also requires a numerical solution.

The dead reckoning technique applied to the special case of constant heading is discussed herein. The basis of all positioning and navigation systems includes geodesy, timekeeping, and astronomy [3]. These fields contributed to the technique of dead reckoning which predicts a final destination point given an initial position and requires knowledge of the heading and the distance to be traveled.

SIMPLE, GEODESIC, HEADING, and EXACT Background

The SIMPLE is iterative and the total distance is obtained from the sum of many small (compared to Earth’s average radius) steps. Each step approximates a curved constant heading distance by a constant straight distance. This distance must be straight since it is resolved into north and east components using the heading angle; see Eqs. (28) and (29) for an example of the step length given by \( v \Delta t \), where \( v \) is some velocity and \( \Delta t \) is a constant time step. These two components are each divided by Earth’s average radius to obtain the change in latitude and longitude for the step. The ending latitude and longitude for the step are obtained by adding the change in latitude to the initial latitude and the change in longitude to the initial longitude. There are two errors.

The first error is due to the finite step size and results in trying to place a flat triangle onto a curved Earth. From Eqs. (28) and (29) the resultant velocity \( V \) has components \( V_N \) and \( V_E \) with step distances of \( V \Delta t \), \( V_N \Delta t \), and \( V_E \Delta t \). Since the model requires the relation \( V^2 = V_N^2 + V_E^2 \) which is only true for a planar triangle then the distances must also lie on a planar triangle. A simpler example of this error is to consider trying to place a straight (unbendable) line onto a curved line, or arc. This problem is analogous to that of estimating the perimeter of a circle with a finite (straight) ruler; there will always be an error because the ruler has a finite length and thus \( \pi \) can only be approximated. This triangle is flat but Earth’s extrinsic geometry is curved and thus the triangle can not map Earth’s surface without some error. This is the well known projection problem that requires some mapping approximation.

The second error is that when Earth is considered an ellipsoid of revolution, there is no one appropriate radius and hence the SIMPLE algorithm is not WGS-84 compliant. The single radius could be supplanted by two radii determined from elliptical geometry. However, this would require introducing a formal coordinate system and would result in a much more complicated model having additional computational cost.

These two errors are generally cumulative since the new position computation starts at the last position.

GEODESIC incorporates a well known geometric and numerical algorithm introduced by Vincenty [8]. It reckons a final position along a geodesic given an initial position, a heading, and a distance. A geodesic is the shortest, straightest path...
between two points on the surface of a given manifold (here, the WGS-84 Earth). GEODESIC takes as input the latitude and longitude, an azimuth (angle), and the distance traveled. The output is the desired latitude and longitude. They are separated from the input position by a geodesic path. This algorithm is extremely accurate. When it was used to obtain distances of ¼ Earth’s perimeter, the result was within 2 millimeters of many closed-form series solutions. The error here is not cumulative. However, if the distance traveled is small and the azimuth is taken as the heading then a path of constant heading will be approximated. For a path composed of a series of linked small steps, each with constant heading, the error will be cumulative.

HEADING takes as input the latitude and longitude angles, the local heading angle, and the desired distance to be traveled. It outputs the final latitude and longitude obtained by traveling along this constant heading for the specified distance. HEADING has only one source of error and this is also one of the errors SIMPLE has: essentially trying to paste a flat triangle onto a curved Earth. The error here is cumulative since the path is composed of small steps and the error exists at each step.

EXACT gives exact (disregarding numerical integration error) distances, for paths of constant heading, between any two points on the ellipsoid. The result is well-known but the derivation given herein may be useful. EXACT is constructed with a differential geometry technique typically used in General Relativity. Differences between the models will be most apparent for large separations. And because only a few large distances of constant heading are known in closed form for the ellipsoid surface, it was of interest to include an exact solution for validation of the other models. Again, the error here is only due to numerical integration.

Coordinate System

WGS-84 is used to describe positions and paths on Earth [2]. Thus, the two-dimensional surface of an ellipsoid is parameterized by the geodetic latitude, φ, and longitude, λ, angles and equivalently by a three dimensional Cartesian coordinate system. Height, or altitude, is described by the usual geodetic convention perpendicular to Earth’s surface. These coordinates were mainly chosen for clarity in applications involving long distances and to conform to the majority of Earth-based applications.

HEADING Model

The equation of the Earth’s surface, with center at the origin, is defined as an ellipsoid with semi-major axis \( a_{\text{earth}} \) and semi-minor axis \( b_{\text{earth}} \)

\[
\frac{x^2}{a_{\text{earth}}^2} + \frac{y^2}{b_{\text{earth}}^2} + \frac{z^2}{c^2} = 1.
\]  

(1)

A vector tangent to Earth’s surface at a given point on that surface and having a given heading orientation, \( h \), is defined as

\[
\vec{t} = A'(\phi, \lambda, h)\hat{x} + B'(\phi, \lambda, h)\hat{y} + C'(\phi, \lambda, h)\hat{z}.
\]  

(2)

A vector normal to the Earth’s surface at a given point on that surface is defined as

\[
\vec{N} = A(\phi, \lambda)\hat{x} + B(\phi, \lambda)\hat{y} + C(\phi, \lambda)\hat{z}.
\]  

(3)

Since \( \vec{N} \) and \( \vec{t} \) are perpendicular their scalar product is zero

\[
\vec{N} \cdot \vec{t} = 0.
\]  

(4)

The vector tangent and parallel to lines of longitude at a given point is defined as

\[
\vec{L} = A''(\phi, \lambda)\hat{x} + B''(\phi, \lambda)\hat{y} + C''(\phi, \lambda)\hat{z}.
\]  

(5)

Since \( \vec{N} \) and \( \vec{L} \) are also perpendicular

\[
\vec{N} \cdot \vec{L} = 0.
\]  

(6)

Thus, these three vectors at each point on the ellipsoid surface form a local orthogonal coordinate system or a tangent space

\[
\vec{t} \times \vec{L} = \vec{N}.
\]  

(7)

This tangent space at a point \( P \) (fig. 3.A.3, p. 118 [3]), where \( \vec{t} \) corresponds to E, \( \vec{N} \) corresponds to U, and \( \vec{L} \) corresponds to N. \( \vec{t} \) is said to lie in the tangent plane at \( P \). In addition, this paper uses the same latitude, longitude, and Cartesian axes shown in the previously mentioned Pratap and Per [3] figure. Thus, the following relation also holds

\[
\vec{t} \cdot \vec{L} = 0.
\]  

(8)
The well-known geodetic-to-Cartesian transformations are

\[
\begin{align*}
    x &= (N + \text{Alt}) \cos(\phi) \cos(\lambda) \\
    y &= (N + \text{Alt}) \cos(\phi) \sin(\lambda) \\
    z &= (N(1 - e_\text{earth}^2) + \text{Alt}) \sin(\phi),
\end{align*}
\]

(9)

The geometry for these equations is shown in Pratap and Per [3]. Since \( \vec{N} \) is normal to the Earth’s surface, its vector form is easily determined (consider the altitude \( \text{Alt} \) to be zero). The unit vector is then

\[
\vec{N} = \cos(\phi) \cos(\lambda) \hat{x} + \cos(\phi) \sin(\lambda) \hat{y} + \sin(\phi) \hat{z}
\]

(10)

and the following constants are defined from \( \vec{N} \)

\[
\begin{align*}
    A &= \cos(\phi) \cos(\lambda) \\
    B &= \cos(\phi) \sin(\lambda) \\
    C &= \sin(\phi)
\end{align*}
\]

(11)

Since \( \vec{L} \) is normal to \( \vec{N} \), its unit vector form is now also easily determined by inspection using Eq. (6)

\[
\vec{L} = -\sin(\phi) \cos(\lambda) \hat{x} - \sin(\phi) \sin(\lambda) \hat{y} + \cos(\phi) \hat{z}
\]

(12)

Also, additional constants are defined from \( \vec{L} \)

\[
\begin{align*}
    A'' &= -\sin(\phi) \cos(\lambda) \\
    B'' &= -\sin(\phi) \sin(\lambda) \\
    C'' &= \cos(\phi)
\end{align*}
\]

(13)

By using Eqs. (4) and (8), \( \vec{t} \) can be algebraically determined up to a multiplicative factor

\[
\vec{t} = B' \left( \varepsilon \hat{x} + \hat{y} + \left( \frac{Ae + B}{C} \right) \hat{z} \right).
\]

(14)

The variable \( \varepsilon \) is given by

\[
\varepsilon = \frac{-b' \pm \sqrt{b'^2 - a'c'}}{a'}
\]

(15)

with

\[
\begin{align*}
    a' &= Ar^2 - \left( 1 + \frac{A^2}{C^2} \right) \cos^2(h) \\
    b' &= Br^2 - \left( 1 + \frac{B^2}{C^2} \right) \cos^2(h)
\end{align*}
\]

(17)

and

\[
\begin{align*}
    b' &= ArBr - \frac{AB}{C^2} \cos^2(h).
\end{align*}
\]

(18)

In Eqs. (15), (16), (17), and (18) \( h \) is the heading angle; \( Ar \) and \( Br \) are given by

\[
\begin{align*}
    Ar &= A'' - \frac{AC''}{C} \\
    Br &= B'' - \frac{BC''}{C}.
\end{align*}
\]

(19)

(20)

Since the unprimed and double primed variables are previously defined in Eqs. (11) and (13), \( Ar \) and \( Br \) are completely defined. Hence \( \varepsilon \) is defined. And if a position, of latitude, \( \phi \), longitude, \( \lambda \), and heading are specified, then \( \vec{t} \) is completely specified and will point from that position towards the heading direction exactly.

Once an initial latitude and longitude are known, it is possible to use Eq. (9) with altitude of zero, to determine the initial Cartesian coordinates, \((x_0, y_0, z_0)\), of the ellipsoid surface. In order to create a constant heading path it is necessary to step from this initial position along a path to a nearby position where \( \vec{t} \) has the same heading and such that for all intermediate points along this path \( \vec{t} \) has this same heading. Thus creating a constant heading path would require sequential steps each of infinitesimal length. This is explored in the EXACT model given below but there is currently no existing closed form solution for obtaining finite length constant heading paths. If an infinitesimal step could be taken, then HEADING would be exactly correct. However, a finite step must be taken. In this case, the smallest step yielding acceptable computational performance (accuracy and speed) is desired. A small step in the heading direction is accomplished by first generating the Cartesian coordinates of the final position on the tangent plane.

A line on the tangent plane, with initial position \((x_0, y_0, z_0)\) and final position \((x_1, y_1, z_1)\) defines a vector
\[ \vec{I} = (x_1 - x_0)\hat{x} + (y_1 - y_0)\hat{y} + (z_1 - z_0)\hat{z}. \]  

(21)

It is desired that \( \vec{I} \) and \( \vec{t} \) be parallel and thus that

\[ \vec{I} \times \vec{t} = 0. \]  

(22)

Then the point \((x_1, y_1, z_1)\) is completely defined by the three independent equations generated with the cross product. After some algebra the following two equations result

\[ y_1 = y_0 - \frac{C(z_1 - z_0)}{A\varepsilon + B} \]  

(23)

and

\[ x_1 = x_0 + \varepsilon(y_1 - y_0). \]  

(24)

Thus it is possible to step from \((x_0, y_0, z_0)\) to \((x_1, y_1, z_1)\) by choosing some \(z_1\) near \(z_0\). The values for \(y_1\) and then \(x_1\) are computed and after this step, an inverse program computes the latitude, \(\phi_1\), and longitude, \(\lambda_1\), corresponding to \((x_1, y_1, z_1)\). This latitude and longitude also update the variables of Eqs. (11) and (13). Then the new orientation of the heading vector \(\vec{I}\) is such that it maintains the heading with respect to the line of longitude at point \((x_1, y_1, z_1)\). The next initial point \((x_0, y_0, z_0)\) is obtained by setting \(x_0 - x_1, y_0 - y_1, z_0 - z_1\). The cycle repeats with a new step from point \((x_0, y_0, z_0)\) to a point \((x_1, y_1, z_1)\) by incrementing \(z_1\). The total distance for the path is obtained by adding the straight line separations of each step.

This methodology does not work for stepping around the lines of latitude because the \(z\)-coordinate is constant. Instead, the equation for the ellipse, see Eq. (1), is solved for \(y_1\)

\[ y_1 = \pm \sqrt{\frac{R^2 - x_1^2}{b^2}} \]  

(25)

with

\[ R = a_{earth}\left(1 - \frac{z_1^2}{b_{earth}^2}\right). \]  

(26)

Then \(x_1\) is incremented with \(z_1\) constant.

**GEODESIC Model**

The heart of the GEODESIC model is the direct formula determined by Vincenty [8]. This outputs a final position given an initial point by computing the geodesic path connecting the two positions. This formula is not quite in closed form. It has an iterative component and can be coded fairly easily. A function called reckon was constructed that takes an initial latitude and longitude, an azimuth angle, and a distance, as input and outputs a latitude and longitude point separated from the initial point by a geodesic path.

If the final position is known then the distance between the two points can be obtained from an inverse formula provided by Vincenty. Inverse also outputs the azimuth given two points on the ellipsoid. This azimuth is generally different from the heading variable used in the other three models and thus the geodesic line does not generally maintain a constant heading along its length. The geodesic line is defined by being the shortest distance and being the 'straightest' line between two points.

However, if the input distance is small then the reckon function approximates a constant heading course for an azimuth equal to the heading. Reckon can then be stepped, or iterated, to produce an approximation of a constant heading course of large distance. In this case, the total distance of the path can be obtained from the sum of the distances for each step.

**SIMPLE Model**

The heading algorithm in SIMPLE is given by Eqs. (27), (28), and (29). The latitude and longitude are determined in an iterative process

\[ \text{LAT} = \text{LAT} + \frac{dN}{R} \]  

(27)

\[ \text{LONG} = \text{LONG} + \frac{dE}{R} \]

\[ dN = V_N \cdot dt \]

\[ dE = V_E \cdot dt \]  

(28)

\[ V_N = V \cos(h) \]

\[ V_E = V \sin(h) . \]  

(29)
In Eq. (29) the velocity $V$ is decomposed into north, $N$, and east, $E$, velocity components using the definition of the current heading of the given iteration. In Eq. (28) a north distance, $dN$, and an east distance, $dE$, is computed for some time step $dt$. Thus, $dN$ lies parallel to a line of longitude and $dE$ lies parallel to a line of latitude. In Eq. (27) these distances $dN$ and $dE$ are divided by a single radius $R$ to give small angles of latitude and longitude, respectively, that are added to the current latitude and longitude. However, this scheme only works exactly for a sphere where there is a single radius. Two different radii could be obtained for the ellipsoid by adding substantial additional structure to the model. In that case the algorithm could have the following modification of Eq. (27)

\begin{align*}
LAT &= LAT + \frac{dN}{R_1} \\
LONG &= LONG + \frac{dE}{R_2}
\end{align*}

where $R_1$ is the radius for the circle of latitude given by Eq. (26) and $R_2 = N$ given in Eq. (9).

**EXACT Model**

Here is given an overview of the derivation for the line element on the surface of the ellipsoid as a function of the latitude. Also, the differential longitudinal element is given. A Simpson’s rule numerical integration then sums the differential distances along Earth’s ellipsoidal surface to yield finite distances of constant heading.

While it is possible to construct a line element directly from the geodetic position vector, the following transformation simplifies the elements of the metric tensor (note: the same elliptic variables from the HEADING section apply here)

\begin{align*}
r \sin(\alpha) &= z = N(\phi)(1 - e_{\text{earth}}^2 \sin(\phi)) \\
r \cos(\alpha) &= N(\phi) \cos(\phi)
\end{align*}

This yields

\[ \alpha = \tan^{-1} \left( (1 - e_{\text{earth}}^2) \tan(\phi) \right) \]

and

\[ r = \frac{N(\phi) \cos(\phi)}{\cos(\alpha)} \].

The variable $r$ is thus

\[ r(\phi) = a_{\text{earth}} \sqrt{1 + \left(1 - e_{\text{earth}}^2 \right)^2 \tan^2(\phi)} \] .

The following position vector defines these coordinates

\[ \vec{r} = r \cos(\alpha) \cos(\lambda) \hat{x} + r \cos(\alpha) \sin(\lambda) \hat{y} + r \sin(\alpha) \hat{z} . \]

Basis vectors form the orthogonal tangent space of a manifold and can be obtained from partial differentiation of the position vector with respect to the coordinates. Thus, the basis vectors are by inspection

\begin{align*}
\vec{e}_\alpha &= \left( \frac{\partial r}{\partial \alpha} \cos(\alpha) - r \sin(\alpha) \right) \cos(\lambda) \hat{x} \\
&+ \left( \frac{\partial r}{\partial \alpha} \cos(\alpha) - r \sin(\alpha) \right) \sin(\lambda) \hat{y} \\
&+ \left( \frac{\partial r}{\partial \alpha} \sin(\alpha) - r \cos(\alpha) \right) \hat{z} \\
\vec{e}_\lambda &= -r \cos(\alpha) \sin(\lambda) \hat{x} + r \cos(\alpha) \cos(\lambda) \hat{y}
\end{align*}

The covariant metric tensor components are defined from

\[ g_{ij} = \vec{e}_i \cdot \vec{e}_j \]

with the following results

\begin{align*}
g_{rr} &= 1 \\
g_{\alpha\alpha} &= r^2 + \left( \frac{dr}{d\alpha} \right)^2 \\
g_{\lambda\lambda} &= r^2 \cos^2(\alpha) \\
g_{r\alpha} &= \left( \frac{dr}{d\alpha} \right) \\
g_{r\lambda} &= 0 \\
g_{\alpha\lambda} &= 0
\end{align*}

Then the line element is (using the Einstein summation convention)

\[ \frac{ds}{dt} = \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} \]
\[ ds^2 = g_{\mu\nu} dx^\mu dx^\nu \]
\[ ds = \sqrt{g_{\mu\nu} dr^2 + g_{\alpha\alpha} d\alpha^2 + g_{\beta\beta} d\beta^2 + g_{\gamma\gamma} d\gamma^2} \]

This simplifies to
\[ ds = \sqrt{3 \left( \frac{dr}{d\alpha} \right)^2 + r^2 \left( 1 + \cos^2(\alpha) \left( \frac{d\lambda}{d\phi} \right)^2 \right)} d\alpha \]

(39)

From Diston [9], the following result is given for \( \frac{d\lambda}{d\phi} \) in terms of the heading, \( h \)
\[ \tan(h) = \cos(\phi) \frac{1 - e_\text{earth}^2 \sin^2(\phi)}{1 - e_\text{earth}^2} \left( \frac{d\lambda}{d\phi} \right) \]

(40)

and \( \frac{d\phi}{d\alpha} \) is determined by differentiating Eq. (32)
\[ \frac{d\alpha}{d\phi} = \frac{1 - e_\text{earth}^2}{\cos^2(\phi) + (1 - e_\text{earth}^2)^2 \sin^2(\phi)} \cdot \frac{1}{r(\phi)} \]

(41)

It is also necessary to find
\[ \frac{dr}{d\alpha} = \frac{dr}{d\phi} \frac{d\phi}{d\alpha} \]

(42)

Differentiating Eq. (34) gives
\[ \frac{dr}{d\phi} = a_\text{earth} \left( 1 - e_\text{earth}^2 \right) \tan(\phi) - \frac{e_\text{earth}^2}{\cos^2(\phi) \left( 1 + (1 - e_\text{earth}^2)^2 \tan^2(\phi) \right)^2} r(\phi) \]

(43)

So the line element, Eq. (40), is a function of only \( \phi \) and \( h \) since \( \cos(\alpha) \) follows from Eq. (32). To determine the finite line element now requires only that some numerical integration scheme (such as Simpson’s Rule) be used to iterate the differential line element over the extent of \( \phi \) desired. In addition the longitude corresponding to the integration can be determined from Eq. (41).

**Speed and Accuracy Tests**

The four models were tested for computational speed and accuracy for a number of different geometrical paths along the surface of the ellipsoid. For all tests and models the step size was approximately 4 m. In the first test (see Table 1), the four models were compared for speed of execution and distance accuracy for a path along the equator. The heading models were required to start distance computations on the equator at 180 degrees longitude, advance in position with a heading of 90 degrees while remaining on the equator, and end at the same latitude but at -90 degrees longitude. Thus the computations were carried out for ¼ of Earth’s equatorial perimeter.

<table>
<thead>
<tr>
<th>Model</th>
<th>Name</th>
<th>Time (s)</th>
<th>Distance (nmi)</th>
<th>Ending Longitude (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE</td>
<td></td>
<td>2.844</td>
<td>5401.1222</td>
<td>-90.00004</td>
</tr>
<tr>
<td>HEADING</td>
<td></td>
<td>11.109</td>
<td>5409.6948</td>
<td>-90.00000</td>
</tr>
<tr>
<td>GEODESIC</td>
<td></td>
<td>11.062</td>
<td>5409.6952</td>
<td>-90.00001</td>
</tr>
<tr>
<td>EXACT</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

It should be noted, however, that the GEODESIC, for this test, can be applied using only a single step. That is, since the beginning and ending positions are known and the equatorial path is a geodesic, the distance can be computed without taking multiple steps and this should be considered the most accurate result: 5409.6945 nmi with a time of 0.063 s. The EXACT numerical algorithm is unstable for a heading of 90 degrees and for headings close to 90 degrees and thus has no data. This is one drawback of EXACT.

One quarter of the equatorial perimeter has a WGS-84 distance of \( 2\pi a_\text{earth}/4 = 5409.6945 \) nmi. Thus, from Table 1 all of the model’s distance performances are well within 0.1 percent of this value. The model’s speed performances were mostly within an order of magnitude from each other. This test and the path of constant longitude test are the only tests that can be compared to a known closed form (series) solution.

The three models were also tested for speed of execution and distance accuracy for a path along a line of longitude. The heading models were required to start distance computations at the equator on -90 degrees longitude and advance northward in position

4.E.2-7
to the North Pole. Thus the computations were carried out for \( \frac{1}{4} \) of Earth’s circumference as seen in Table 2.

Table 2. Circumferential Speed and Accuracy

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Time (s)</th>
<th>Distance (nmi)</th>
<th>Ending Latitude (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE</td>
<td>2.867</td>
<td>5401.1221</td>
<td>90.00</td>
</tr>
<tr>
<td>HEADING</td>
<td>1.125</td>
<td>5397.5321</td>
<td>89.99</td>
</tr>
<tr>
<td>GEODESIC</td>
<td>13.377</td>
<td>5400.0301</td>
<td>90.00</td>
</tr>
<tr>
<td>EXACT</td>
<td>22.578</td>
<td>5400.6599</td>
<td>90.00</td>
</tr>
</tbody>
</table>

GEODESIC can be applied also in this case using only a single step since again the beginning and ending positions are known and the circumferences are also geodesics. This yields a distance of 5400.6235 nmi with a time of 0.03 s. It is interesting to see that this GEODESIC single step gives a result quite close to EXACT.

The circumference does not have a closed form solution but can be approximated with the following infinite series obtained from the complete elliptic integral of the second kind

\[
C = 2\pi a_{\text{earth}} \left( 1 - \frac{1}{2} \frac{e_{\text{earth}}^2}{1} - \frac{1 \cdot 3}{2 \cdot 4} \frac{e_{\text{earth}}^4}{3} - \ldots \right)
\]

(45)

One quarter of the circumference thus has a WGS-84 distance of \( C/4 = 5400.629437 \) nmi. Thus, from Table 2 all of the model’s distance performances were well within 0.1 percent of the actual value. The model’s speed performances were mostly within an order of magnitude from each other but EXACT was much slower.

EXACT yields a closed form solution for the longitude over a finite distance in terms of the latitude, but there is not a closed form solution for a finite distance traveled due to the complexity of Eq. (40). However, this equation does give an exact differential distance along the ellipsoid having a given heading. The following test (see Table 3) shows the time, arc-length distance, and ending longitude for the four models starting at the equator

Table 3. Speed and Accuracy at 20 Degree Heading For One-Degree Latitude Change

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Time (s)</th>
<th>Distance (nmi)</th>
<th>Ending Longitude (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE</td>
<td>0.079</td>
<td>63.864560</td>
<td>-89.63602</td>
</tr>
<tr>
<td>HEADING</td>
<td>0.109</td>
<td>63.532109</td>
<td>-89.63848</td>
</tr>
<tr>
<td>GEODESIC</td>
<td>0.281</td>
<td>63.538208</td>
<td>-89.63844</td>
</tr>
<tr>
<td>EXACT</td>
<td>0.375</td>
<td>63.537153</td>
<td>-89.63844</td>
</tr>
</tbody>
</table>

The GEODESIC applied in this case using a single step gave an expectedly fast time of 0.093 s and ending longitude of -89.63844 degrees when a distance of 63.532110 nmi and heading of 20 degrees was applied. Thus, in this case the single step and multiple-step GEODESIC give similar results.

For this test, SIMPLE and HEADING were about three times faster in execution than either of GEODESIC and EXACT. Since HEADING, GEODESIC, and EXACT are closer in distance to each other than SIMPLE is to any of them, it is likely that SIMPLE is the least accurate. It should be noted that the single step GEODESIC hardly deviated from the heading models for this reasonably large distance.

The next test (see Table 4), shows the time, arc-length distance, and ending longitude for the four models starting at the equator (and -90.0 longitude) and traveling at a 60 degree constant heading until 40 degrees latitude is reached.

Table 4. Speed and Accuracy at 60 Degree Heading For 40-Degree Latitude Change

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Time (s)</th>
<th>Distance (nmi)</th>
<th>Ending Longitude (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE</td>
<td>2.750</td>
<td>4800.9986</td>
<td>-20.71794</td>
</tr>
<tr>
<td>HEADING</td>
<td>0.750</td>
<td>4783.5755</td>
<td>-14.71592</td>
</tr>
<tr>
<td>GEODESIC</td>
<td>11.671</td>
<td>4783.5116</td>
<td>-14.71683</td>
</tr>
<tr>
<td>EXACT</td>
<td>11.703</td>
<td>4783.5145</td>
<td>-14.71688</td>
</tr>
</tbody>
</table>

GEODESIC applied using a single step again gave an expectedly fast time of 0.094 s but an ending
longitude of -11.94507 degrees when a distance of 4783.5151 nmi and 'heading' of 60 degrees was applied. The single-step and many-step GEODESIC gave very different results. For very large paths with general headings it is expected that the heading algorithms give different results from the geodesic algorithms.

The execution speed of HEADING was almost four times faster than SIMPLE and well over an order of magnitude faster than GEODESIC and EXACT. EXACT was the slowest. The ending single step GEODESIC longitude is much different from the other heading models as expected for the long distance with changing latitude and longitude. Also notable is how close the HEADING, GEODESIC, and EXACT results are. This is also as expected since GEODESIC takes small incremental steps with constant heading (or azimuth). The closeness of these distances also helps to verify the accuracy of HEADING since the EXACT errors are known to be only caused by the numerical integration scheme.

**Conclusion**

Four different heading models were compared. One new heading model was introduced: the HEADING. For reckoning a constant heading path of general heading angle or determining path distances, HEADING is preferable to the other models if both speed and accuracy are desired. In addition, HEADING performs especially well for paths of large distance and for headings having changes in both latitude and longitude.

SIMPLE and HEADING are particularly fast; in some cases over one order of magnitude faster than EXACT and GEODESIC. But HEADING was generally more accurate than SIMPLE: especially for paths where both the latitude and longitude change. This is also consistent with SIMPLE having the two previously mentioned error sources and HEADING having one error source. GEODESIC was about as slow as EXACT when used to produce heading paths, but is extremely accurate for producing geodesic paths. For applications requiring geodesics, GEODESIC is preferable. GEODESIC was also quite useful in providing very accurate constant heading paths for use in comparison with the heading models. EXACT arguably gives the most accurate results but was particularly slow. HEADING was almost as accurate as EXACT but generally executes much faster.

All the models can be configured to produce a final position given an initial position, heading, and distance to be traveled. The models can also be used to find the distance to a line of longitude or latitude given a heading from an initial position.

**References**


**Acknowledgments**

This paper is based on work supported under RITA/VNTSC #FA7GB2 HA021.
Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors.

Appendix I—Nomenclature

\( a_{\text{earth}} \) – major axis of Earth
\( a' \) – variable
\( A \) – normal vector variable
\( A' \) – tangent vector variable
\( A'' \) – longitudinal vector variable
\( Ar \) – variable
\( Alt \) – altitude above ellipse
\( b_{\text{earth}} \) – minor axis of Earth
\( b' \) – variable
\( B \) – normal vector variable
\( B' \) – tangent vector variable
\( B'' \) – longitudinal vector variable
\( Br \) – variable
\( C \) – normal vector variable
\( C' \) – tangent vector variable
\( C''' \) – longitudinal vector variable
\( c' \) – variable
\( C \) – circumference
\( ds \) – surface line element
\( dE \) – differential distance
\( dN \) – differential distance
\( e_r \) – basis vector
\( e_{\alpha} \) – basis vector
\( e_t \) – basis vector
\( e_{\text{earth}} \) – eccentricity of Earth
\( g_{ij} \) – metric tensor
\( h \) – heading
\( l \) – line
\( L \) – longitudinal tangent to Earth surface
\( LAT \) – latitude
\( LONG \) – longitude
\( N \) – normal to Earth surface
\( R \) – average Earth radius
\( R_1 \) – an Earth radius
\( R_2 \) – an Earth radius
\( r \) – radius
\( t \) – heading tangent to Earth surface
\( V_N \) – north component of velocity
\( V_E \) – east component of velocity
\( x \) – ellipse coordinate
\( x_1 \) – variable
\( x_0 \) – variable
\( y \) – ellipse coordinate
\( y_1 \) – variable
\( y_0 \) – variable
\( z \) – ellipse coordinate
\( z_1 \) – variable
\( z_0 \) – variable
\( \alpha \) – angle
\( \epsilon \) – variable
\( \lambda \) – longitude of ellipse
\( \varphi \) – latitude of ellipse

29th Digital Avionics Systems Conference
October 3-7, 2010