

# New Efficient Procedures for Geodetic Coordinate Transformations

Ralph M. Toms  
SRI International  
333 Ravenswood Ave.  
Menlo Park, CA 94025  
(650) 859-2852  
ralph\_toms@sri.com

Keywords:

Coordinate Conversion/System, Correlation, Modeling and Simulation Interoperability

**ABSTRACT:** *The increased need for joint systems simulation, along with a concomitant need for more realistic sensor representation, makes it critical that Earth-related physical phenomenon be accurately modeled. For some assets, accumulated errors in air density, gravitational forces, and other environmental factors can lead to serious inaccuracies. This problem exists for target acquisition, kinetics models, communications processing, and others. Environmental data are often naturally represented with respect to geodetic Earth models. This implies an increased use of geodetic coordinates.*

*Under the Distributed Interactive Simulation (DIS) paradigm, use of Cartesian coordinates (UTM) in many models, coupled with the requirement to use geocentric coordinates in the standardized protocol, induced frequent expensive geodetic coordinate transformations. The geographic coordinate system (GCS) has been developed to avoid some of the distortion problems resulting from using UTM coordinates. However, even with GCS, environmental parameters may still have to be aligned with geodetic coordinates for some applications. If network position data are in geocentric coordinates, a partial transformation to geodetic coordinates is required to determine GCS cell membership.*

*Historically, geodetic coordinate transformations have induced heavy computational loads on system throughput. The development of new, more efficient procedures over the last three years has improved the situation. However, the increasing scope and detail of emerging simulations continue to make this an important problem. Rational function approximations have been developed that provide substantive additional savings in execution time. At the same time, total errors of less than a millimeter are achieved out to the edge of the mesosphere (50 km). A special application of these procedures makes GCS cell determination more efficient.*

## 1. Introduction

The increased need for joint systems simulation, along with a concomitant need for more realistic sensor representation, makes it critical that Earth-related physical phenomenon be accurately modeled. For some assets, accumulated errors in the representation of air density, gravitational forces, and other environmental factors can lead to serious inaccuracies. This problem exists for target acquisition, kinetics models, kinematics models, communications processing, and others. Environmental data are often naturally represented and used with respect to geodetic coordinates. Sometimes environmental data are stored in other coordinate systems, including those formed by projections from an Earth reference ellipse. To request and retrieve such data, the coordinates in which the data are stored must be

transformed into geodetic coordinates. In fact, geodetic coordinate transformations are involved in almost all such transactions, which implies their increased future use.

The issue of efficiency is important because of the central role that coordinate transformation computations play in both stand-alone and distributed applications. The more efficient the transformations, the more entities that can be processed. Reductions in nodal compute times provide more flexibility when trading off bandwidth for processing power. The tradeoff between the representational fidelity of an entity model and the transformations used revolves mainly around the cost of the computations associated with coordinate representations. Use of geodetic coordinates for kinetics models is often avoided because the resulting equations

are perceived to be too computationally intensive. This perception may be open to argument and needs a systematic study to determine its validity. The new procedures may make it feasible to model some dynamics equations directly in geodetic coordinates.

Efficiency is also a concern when extracting environmental data from authoritative databases. In the SEDRIS\* Program, very large databases need to be converted from one coordinate system to another [1]. The reference data are gridded, in a usually non-Euclidean system. When the data are transformed to another system, uniform grids are distorted. Finer sampling along grid boundaries may be needed to ensure lossless and accurate incorporation of the data into another grid in a simulation coordinate frame. While much of this processing may be done prior to an exercise, there is increasing pressure to do it during an exercise. It is contemplated, that as real-world applications involve more and more simulations, battlefield commanders will want the ability to replan while the battle is being prosecuted. As real-world environmental updates become more timely, the need for replanning will become more important and response timelines will shrink.

The problem of transforming from geodetic coordinates to geocentric coordinates has received an inordinate amount of attention for what seems to be a relatively simple problem, but in fact is not. In previously published papers [2,3], the author has both provided a historical review of the problem and developed new methods that retain accuracy, while reducing the computational expense involved. A surprisingly large number of authors [2,3,4] believe that no closed-form solution exists, although it is easily derived. Most authors provide very little in the way of an error analysis. In some cases, only the altitude errors are addressed. For joint applications which some assets operate over long ranges, the angular errors may dominate and cannot be ignored.

Almost none of the papers published prior to 1992 address computational efficiency. Borkowski [4] reports run-time comparisons, but makes no attempt to improve efficiency because it was not an issue for his application. The closed-form solution involves the solution of a quartic equation by the classical method due to Ferrari. Care must be taken to avoid computationally costly, complex arithmetic and the inevitable ill conditioning associated with analytic solutions of this type. This, in

turn, requires use of multiple precision even on machines having extended word length. Borkowski shows how to formulate the quartic equation to avoid both the complex arithmetic and the ill conditioning. However, this reformulation introduces some relatively expensive transcendental function evaluations. In addition, the Ferrari method itself requires several relatively time-consuming multiple square root and cube root operations. As a result, the closed-form solutions are not very efficient. One particular closed-form solution, due to Heikkinen [5],<sup>†</sup> avoids the use of both complex variables and trigonometric functions. In addition, the Heikkinen solution has good numerical stability properties [6]. Consequently, this formulation establishes a good baseline against which alternative formulations can be compared, in terms of both accuracy and computational efficiency.

In a previous paper [7], the classic Bowring iterative method was modified to provide an efficient and accurate solution to the problem. A three-dimensional error metric was defined and the resulting algorithm was tested, for all latitudes, longitudes, and altitudes ranging from well under sea level to ten million kilometers. When using the WGS-84 Earth-reference ellipsoid, the maximum (total) error was less than 1 cm over the test region.

In simulation applications, both accuracy and throughput are issues. The accuracy requirement continues to be misunderstood by many people in the community. A 1-cm accuracy may seem excessive, given that ellipsoidal Earth models represent best fits to the real Earth shape and induce far more than 1-cm measurement error when referenced to an inertial frame. However, when performing simulations, the selected Earth model and associated coordinate system are treated as being ground truth (exact). To verify and validate simulations, particularly distributed simulations, it is essential that the coordinate conversions be accurate. The issue of how such accuracies relate to the real world, and therefore apply to linking virtual and live entities, is addressed in a seminal paper by Lucha [8].

## 2. New Developments

This paper contains several improvements made to the algorithms developed in previous papers published by the author [2,3].

---

\* Synthetic Environment Data Representation and Interchange Specification

---

<sup>†</sup> The Heikkinen solution is included herein as Appendix C for easy reference. All timing estimates given in this paper are compared to an implementation of this closed-form solution.

- The modified Bowring method has been reformulated to achieve further efficiencies.
- A rational function approximation to the tangent of the latitude has been developed that provides increased efficiency in the region lying below the lower boundary of the mesosphere (50-km altitude).
- The observation that the tangent function is monotone with respect to latitude is coupled with the rational function approximation to make determination of tessellation cell membership much more efficient.
- The inverse transformation, from geodetic to geocentric coordinates, is addressed from the point of view of efficiency.

The latter three improvements are discussed in subsequent sections of this paper. The revised formulation of the modified Bowring method is contained in Appendix B.

### 3. Rational Function Approximation

The mathematical formulation of most of the coordinate transformation implementations were developed in the early part of this century or earlier. The majority of cases involve a requirement to solve non-linear algebraic equations in several variables involving many transcendental function evaluations. Exact solutions exist in only a few special cases and, when they do, are often expensive to compute. As a result, power series expansions were used along with iterative methods to solve these problems. This was the appropriate and, in fact, the only practical way to solve such problems at the time. The recent rapid growth in computational capability, coupled with the concomitant development of algorithms for non-linear approximation, makes it feasible to develop much more efficient procedures. Today, some of the more critical transcendental functions are implemented directly in hardware to attain substantial speed improvements. Because of these developments, new computational strategies are appropriate.

It is commonly known that, for approximating functions in one dimension, rational approximations are almost always more efficient than power series expansions or iterative techniques for the same or better accuracy [9]. A rational function approximation to a function  $f$  is a ratio of polynomials of the form,

$$(1) \quad f(x,y) = P_n(x)/Q_m(x)$$

where  $n$  and  $m$  are the orders of the polynomials  $P$  and  $Q$ . If  $F$  is a two-dimensional function, this concept can be

extended so that the rational function approximation is of the form,

$$(2) \quad F(x) = P_n(x,y)/Q_m(x,y).$$

Approximation of functions in this manner requires both a choice of  $m$  and  $n$  and a metric for determining the approximation error. Metrics may include, for example, least squares, minmax, and interpolation on some sample space on which the function is being approximated.

#### 3.1 Error definition and assessment

In geodetic coordinates, two of the components of the system are angular measures and one is a distance measure. To simultaneously evaluate the approximation error in all components, the concept of an "error ball" is introduced. An error ball of radius  $E$  is just a sphere centered at the exact point. To expand on this, suppose that  $(X,Y,Z)$  is the exact location of a point  $P$  in the geocentric coordinate system.\* An approximate transformation of the coordinates of  $P$  results in another point  $P_a$  having approximate geodetic coordinates  $(\phi_a, \lambda_a, h_a)$ . Using the exact relations (A-2), (A-3), and (A-4) of Appendix A, the approximate geodetic coordinates can be transformed into corresponding approximate geocentric coordinates  $(X_a, Y_a, Z_a)$ . The error  $E$  induced by the approximation is defined to be the Euclidean distance between  $P$  and  $P_a$ . That is,

$$(3) \quad E = [(X-X_a)^2 + (Y-Y_a)^2 + (Z-Z_a)^2]^{1/2}.$$

$E$  can be viewed as the radius of a ball (sphere) centered at  $P$  in the geocentric system. The absolute error in any one component of  $(X_a, Y_a, Z_a)$  is less than  $E$ , the radius of the ball.

The space of possible latitudes and longitudes is bounded, but the extent of geodetic height,  $h$ , is not. To select the rational function coefficients and to evaluate the maximum approximation error,  $h$  needs to be limited. For this purpose,  $h$  is taken to be in the closed interval -2 km to 50 km. By a symmetry analysis, it is sufficient to perform the error analysis on the region bounded by latitudes and longitudes in  $[0, \pi/2]$  and  $h$  (in km) in  $[-2.0, 50.0]$ . The procedure used to evaluate the maximum error involves introducing a uniform lattice of points in the three-dimensional region by defining a uniform grid on each of the coordinate axes. The error is computed for each lattice point and the maximum determined. As the lattice is made successively finer,

---

\* All notation is defined in Appendix A.

the maximum error is determined to any reasonable degree of accuracy.

### 3.2 Determining the fitting coefficients

By examining the form of the expression for the tangent of the latitude in the modified Bowring method, and with considerable experimentation, an approximating form was chosen. That is,

$$(4) \quad \tan(\phi) = \frac{Z(a_1 + a_2W^2 + a_3Z^2)}{W(a_4 + a_5W^2 + A_6Z^2)}$$

In the interest of economizing on development time, it was decided to interpolate the coefficients. This will result in a set of linear algebraic equations to determine the coefficients. The process begins by dividing out one of the constants so that,

$$(5) \quad \tan(\phi) = \frac{Z(b_1 + b_2W^2 + b_3Z^2)}{W(b_4 + b_5W^2 + Z^2)}$$

This equation can then be written,

$$(6) \quad \frac{Z b_1}{W} + ZW b_2 + Z^2 b_3 / W - (\tan(\phi)) b_4 - (\tan(\phi))W^2 b_5 = Z^2 \tan(\phi)$$

which is linear in the five unknown constants  $b_n$ .

Five linear equations can be developed by evaluating equation (6) at five points in the region R defined by  $[0, \pi/2] \times [-2.0, 50.0]$ . It is tempting to evaluate these at the four corners of the region and at the center. However, if these evaluations are done along the boundary where  $\phi = 0$ , the coefficient matrix will be singular. To avoid this, take  $\mu = 10^{-14}$  and choose the five points to be  $(\mu, -2.0)$ ,  $(\mu, 50.0)$ ,  $(\pi/2, -2.0)$ ,  $(\pi/2, 50.0)$  and  $(\pi/4, 25.0)$ . These points are nearly at the corners plus the center of the region. The matrix is now non-singular and the five equations can be solved.

Subsequent to obtaining the solution, long division is used to eliminate a multiply operation so that the final equation has the form,

$$(7) \quad \tan(\phi) = \frac{Z(c_1 + (c_2W^2 + c_3Z^2))}{(c_4 + c_5W^2 + Z^2)/W}$$

where

$$(8) \quad \begin{aligned} c_1 &= 0.100225296198830 \times 10^1 \\ c_2 &= -0.392760690070161 \times 10^{-4} \\ c_3 &= 0.241369063936131 \times 10^{12} \\ c_4 &= 0.133902505771241 \times 10^{14} \end{aligned}$$

$$c_5 = 0.984551360760386$$

This rational approximation was used in place of equation (B-6) of Appendix B, and a global error analysis was conducted on the region R. This process resulted in a total error of less than 1 mm on R. Without changing the coefficients, the procedure was tested on the extended region by  $[0, \pi/2] \times [-10.0, 50.0]$ , which also resulted in errors less than a millimeter (less than 0.00098).

Other than computing W, which is always required, no square roots or trigonometric functions are present in the rational approximation, and the computation requires fewer arithmetic operations than equation (B-6) of Appendix B. At the same time, the accuracy is better (1 mm vs 1 cm).

### 3.3 Timing analysis

The only way to really assess run time is to implement the algorithm on a particular machine and test it. Some idea of the relative cost of an algorithm can be obtained by using operation counts if the cost of each operation is known. In the timing analyses for this paper, the closed-form Heikkinen solution was used as a gold standard for comparative purposes. This procedure was implemented and timed by averaging 10,000 cycles of the procedure. All procedures tested were also executed for 10,000 cycles, and the average execution times were scaled to the gold standard by dividing by the average execution time for the Heikkinen solution. This means that the scaled execution time for the standard is one. A scaled time larger than one exceeds the standard. Clearly, the scaled times can also be viewed as percentages.

In Figure 1, the scaled time for the Bowring method was implemented with all of the intermediate trigonometric functions included. The timing for the methods of References [10] and [11] were estimated from operations counts. The methods of Heikkinen, References [2] and [3], and the new procedure of this paper were timed by execution on an Mac IISI. NOTE: Due to variances in implementation of the mathematical service routines, there may be some variances in the results when other machine environments are used. However, the relative placements on the curve should not change much.

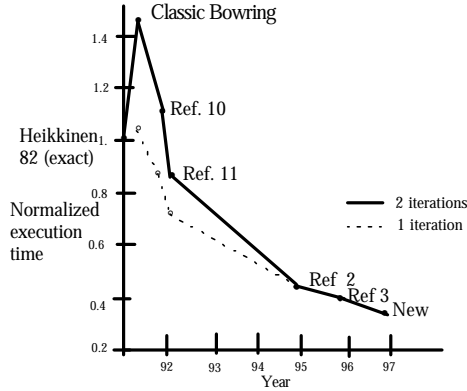


Figure 1. Timing Comparisons

Note that a single iteration of the Bowring method and the exact solution execute in about the same time. Also, note that accuracy has not been addressed in this graph. However, the new procedure is correct to less than a millimeter of error for all angles and for  $h$  between -10 km and 50 km. It is the combination of speed and accuracy and the fact that almost all simulated combat entities lie in the region of applicability that makes the new method attractive.

#### 4. Tessellation Cell Membership

To economize on storage, it is common practice to store terrain and other Earth-related data on sub-regions defined by tessellating the surface of the reference ellipsoid into cells. It is important to associate a point in space with a particular cell. Terrain data are usually provided in terms of latitude and longitude. It is therefore expedient to define the cells in geodetic coordinates. If the point in question is already in geodetic coordinates, the cell membership problem is easy. However, if the point is in geocentric coordinates, the problem is less straightforward. In the GCS coordinate system, the cells are essentially one degree on a side in latitude and longitude [12]. On first examination, it appears that both the latitude and longitude of the point must be calculated by one of the methods given in this paper. However, by leveraging the fact that the tangent function is monotone in its argument, cell membership can be determined without calculating the angles.

The method is illustrated using just the latitude  $\phi$  of the point  $P$ . The problem is to determine  $i$  such that  $\phi$  lies between  $\phi_i$  and  $\phi_{i+1}$  and the  $\phi_i$  are the latitude interval boundaries associated with a cell. Due to the monotonicity of the tangent function, the following are all equivalent:

$$(9) \quad \phi_i < \phi < \phi_{i+1},$$

$$(10) \quad \tan \phi_i < \tan \phi < \tan \phi_{i+1}, \text{ and}$$

$$(11) \quad \tan \phi_i \frac{Z(c_1 + (c_2 W^2 + c_3 Z^2))}{(c_4 + c_5 W^2 + Z^2)} < \tan \phi < \tan \phi_{i+1}.$$

In the first quadrant, this is equivalent to

$$(12) \quad \tan^2 \phi_i \frac{Z^2 (c_1 + (c_2 W^2 + c_3 Z^2))}{(c_4 + c_5 W^2 + Z^2)^2} < W^2 < \tan^2 \phi_{i+1}$$

Since the  $\tan^2 \phi_i$  terms can be stored as constants, equation (11) can be used to determine cell membership without requiring any trigonometric functions or square roots. This concept is readily extended to other quadrants and to tests involving longitude.

#### 5. The Inverse Transformation from Geodetic to Geocentric Coordinates

The inverse transformation is given by equations (A-2), (A-3), (A-4), and (A-5) in Appendix A. While these are exact relationships and appear simple, they are relatively expensive to compute. The question then arises as to whether they can be sufficiently well approximated to save processing time by restricting computations to the one-degree intervals associated with tessellation cells. Before doing any curve fitting, however, the problem should be bounded in terms of both the accuracy and number of operations required.

The answer to the question just posed is probably very dependent on the machine environment in which the computations are performed. Some advanced machines compute sines and cosines at about the same speed as they compute square root. In this case, it takes about five (long) floating point operations ("floats") for each of these functions. On this basis, it would take about 36 floats (mostly long operations) to perform the transformation. A few machines have dual sine and cosine routines that return a pair of values in one call. Assuming five floats for the pair, this reduces the total operations to about 26 floats. If, however, the trigonometric functions take 12 floats, for example, and the square root is 5, the total number of operations would be about 64 floats. These numbers are an estimate of the operations budget that could be used for approximations. If linear interpolation of the trigonometric terms is used on the one-degree intervals, the total number of floats is about 24, assuming that indexing is free. Of course, there would be more short operations in this approach.

To give the reader an idea of the nature of the error allowance, the following is offered. Suppose each trigonometric function is approximated on a one-degree interval with maximum absolute error  $e$ . Since the term  $e^2 \sin^2(\phi)$  is very small, the binomial theorem can be used to compute  $R_N$  very accurately, even for points a million kilometers from the surface of the ellipsoid. Accordingly,

$$(13) \quad R_N = a(1 - .5e^2 \sin^2(\phi)).$$

Taking the approximation to  $\sin(\phi)$  as  $\sin(\phi) + e$  and ignoring terms in  $e^2$  yields,

$$(14) \quad R_N = a(1 - .5e^2 \sin^2(\phi) + e \sin(\phi)).$$

Since  $|\sin(\phi)| \leq 1$  and using (13),

$$(15) \quad R_N = R_N + ae, \text{ and}$$

$$(16) \quad R_{N+h} = R_N + h + ae.$$

Assuming that  $l = 0$  from equation (A-2) of Appendix A, the approximate value  $X_a$  of  $X$  is given by

$$(17) \quad X_a = (R_N + h + ae)(\cos(\phi) + e)(\cos(\phi) + e)$$

Again, expanding and ignoring terms in  $e^2$  yields,

$$(18) \quad X_a = X + 2(R_N + h)\cos^2(\phi) + ae \cos(\phi).$$

When  $h=0$  and  $\phi = 0$ , this becomes

$$(19) \quad X_a = X + 3ae.$$

Now,  $3ae$  is on the order of  $2.0 \times 10^7 e$ . To have  $2.0 \times 10^7 e$  less than one centimeter (.01),  $e$  must be less than  $0.5 \times 10^{-9}$ . While this is only a point analysis along one axis and ignores some potential cancellation of errors, it indicates the severity of the problem.

On one-degree intervals, approximations of the form  $\sin(\phi) = \phi(a + \phi^2 b)$  are only accurate to about  $10^{-6}$  (linear approximations are much worse). It is likely that much higher order approximations will be needed, or the interpolation interval will have to be reduced to less than a degree. The higher order approximations will make this a losing proposition in terms of operations counts. The decision to compute the reverse transformation in-line will depend on the need and the particular machine environment. Clearly, substantial savings cannot be anticipated unless accuracy requirements are eased.

## 6. Appendices

### 6.1 Appendix A: Coordinate Systems for an Ellipsoidal Reference--Model WGS-84.

Several reference ellipsoids have been used in astrogeodetic work [13,14]. These all have the form

$$(A-1) \quad (X/a)^2 + (Y/a)^2 + (Z/c)^2 = 1.$$

In this paper, the World Geodetic System 1984 (WGS-84) is used for the purpose of exposition. For WGS-84,  $a = 6,378,137.0$  m and  $c = 6,356,752.3142$  m [13]. Figure A-1 depicts the geometry of the geocentric (Cartesian) system and the geodetic system in three dimensions.

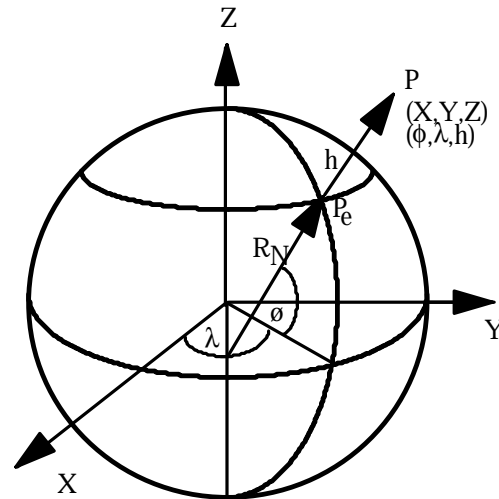


Figure A-1

The geocentric coordinates of a point  $P$  are  $(X, Y, Z)$ , and the corresponding geodetic coordinates of  $P$  are  $(\phi, \lambda, h)$ , where  $\phi$  is latitude,  $\lambda$  is longitude, and  $h$  is the height above the reference ellipsoid. The line connecting the  $Z$  axis to  $P$  is orthogonal to the tangent plane at the point  $P_e$ .

The transformation from geodetic to geocentric coordinates is straightforward [2] and is given by:

$$(A-2) \quad X = (R_N + h) \cos \phi \cos \lambda,$$

$$(A-3) \quad Y = (R_N + h) \cos \phi \sin \lambda, \text{ and}$$

$$(A-4) \quad Z = (R_N c^2 / a^2 + h) \sin \phi,$$

where

$$(A-5) \quad R_N = a / [1 - [\sin^2 \phi] (a^2 - c^2) / a^2]^{1/2}.$$

The inverse transformation is not as easy and is the subject of this Appendix. The longitude  $\lambda$  is given by

$$(A-6) \quad \lambda = \tan^{-1} (Y / X), \text{ and}$$

$$- \lambda .$$

Due to the symmetry of the problem in X and Y, it is sufficient to initially work with a meridional section of the ellipsoid to determine  $\phi$  and  $\lambda$ . This system is depicted in Figure A-2.

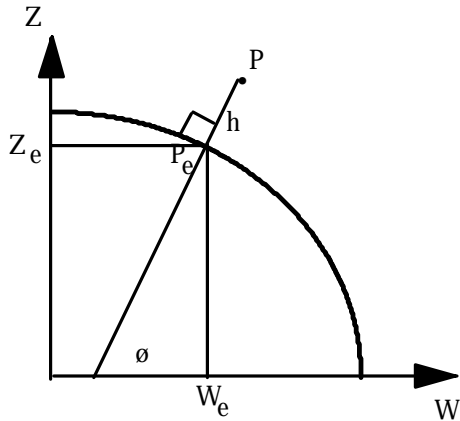


Figure A-2

The meridional ellipse is defined by

$$(A-7) \quad (W / a)^2 + (Z / c)^2 = 1, \text{ and}$$

$$(A-8) \quad W = (X^2 + Y^2)^{1/2}.$$

Some useful relations associated with this coordinate system are given below.

The flattening ratio  $f$  and the eccentricity  $e$  are constants for a particular geode, and are defined by

$$(A-9) \quad f = (a - c) / a, \text{ and}$$

$$(A-10) \quad e^2 = (a^2 - c^2) / a^2.$$

It is convenient to define an additional constant  $e'$  by

$$(A-11) \quad e'^2 = (a^2 - c^2) / c^2.$$

Once  $\phi$  has been determined,  $h$  can be computed from

$$(A-12) \quad h = (W / \cos \phi) - R_N$$

for  $\phi$  in non-polar regions. In polar regions, it is preferable to use

$$(A-13) \quad h = Z / \sin \phi + R_N (e^2 - 1),$$

where  $R_N$  is the radius of curvature of the prime vertical and is given by

$$(A-14) \quad R_N = a / (1 - e^2 \sin^2 \phi)^{1/2}.$$

## 6.2 Appendix B: The Bowring Method

### 6.2.1 The improved Bowring method

In 1976, Bowring [7] developed a very rapidly converging iterative procedure based on Newton's method for computing  $\tan \phi$ . The Bowring method is, in fact, the standard procedure used in the Army Corps of Engineers Handbook TEC-SR-7 [13] and in Rapp [6]. In two of the references [10,11], the Bowring method is not recommended for high-speed computation on the basis that the computation of  $\tan \phi$  requires several relatively expensive trigonometric function evaluations per iteration. A simple observation shows, in fact, that no trigonometric calculations are needed during the iteration. This observation, coupled with an improvement in the initial guess, yields an efficient procedure so accurate that only one iteration is required. This means that no termination test is needed in most applications--resulting in further savings in processing time. This method is called the improved Bowring method [2,3]. In this Appendix, further efficiency improvements are made by rearranging the equations used in reference [2]. The result will also be called the improved Bowring method.

The Bowring procedure consists of introducing an auxiliary variable  $\beta$  such that,

$$(B-1) \quad \tan \phi_{i+1} = (Z + c e^{-2} \sin^3 \beta_i) / (W - a e^2 \cos^3 \beta_i), \text{ and}$$

$$(B-2) \quad \tan \beta_{i+1} = (1-f) \tan \phi_{i+1},$$

with the initial value of  $\beta$  given by

$$(B-3) \quad \tan \beta_0 = aZ / cW.$$

The iteration is terminated when  $|\tan \phi_{i+1} - \tan \phi_i|$  is small enough and  $\phi$  is then computed by using the inverse tangent function.

Observe that  $\beta$  does not explicitly appear in equations (B-1), (B-2) or (B-3). Instead,  $\sin \beta$ ,  $\cos \beta$ , and  $\tan \beta$  are required. These terms are readily computed from basic principles. That is, in (B-1) let

$$(B-4) \quad A_i = Z + c e^{-2} \sin^3 \beta_i, \text{ and}$$

$$(B-5) \quad B_i = W - a e^2 \cos^3 \beta_i, \text{ then}$$

$$(B-6) \quad \tan \phi_{i+1} = A_i / B_i.$$

Then, by definition,

$$(B-7) \quad \sin \phi_{i+1} = A_i / (A_i^2 + B_i^2)^{1/2}, \text{ and}$$

$$(B-8) \quad \cos \phi_{i+1} = B_i / (A_i^2 + B_i^2)^{1/2}.$$

The values of  $\sin \beta_{i+1}$  and  $\cos \beta_{i+1}$  to be used in the next iteration are obtained from equation (B-5). The initial condition (B-6) becomes

$$(B-9) \quad \sin \beta_0 = aZ / [(aW)^2 + (cZ)^2]^{1/2}, \text{ and}$$

$$(B-10) \quad \cos \beta_0 = cW / [(aW)^2 + (cZ)^2]^{1/2}.$$

By using (B-4) to (B-10) and equation (B-2), all intervening trigonometric functions are eliminated and replaced with two square roots.

The Bowring method can be further improved by simple modification of the initial value of  $\tan \beta$ . Experimentation with the Bowring procedure has shown that the error in  $\tan f$  is one signed in the first quadrant. Based on this, a multiplicative weighting factor is introduced in equation (B-3) to minimize the error  $E$  after one iteration over all points in the first quadrant and for a suitable interval of  $h$  values.

Introducing the factor  $D$  into (3) yields

$$(B-11) \quad \tan \beta_0 = aDZ / cW.$$

For a specific reference ellipsoid, the value of  $D$  can be selected to minimize the error  $E$  on a set  $S$  that can be determined without knowing  $f$  and  $h$ . The set  $S$  is defined as all points  $P$  in the first quadrant with coordinates  $(W,Z)$  that lie in the region bounded by the ellipses

$$(B-12) \quad [W / (a+h\text{Min})]^2 + [Z / (c+h\text{Min})]^2 = 1, \text{ and}$$

$$(B-13) \quad [W / (a+h\text{Max})]^2 + [Z / (c+h\text{Max})]^2 = 1.$$

The set  $S$  is depicted in Figure B-1.

A point  $(W,Z)$  is in  $S$  if both of the following conditions hold:

$$(B-14) \quad [W / (a+h\text{Min})]^2 + [Z / (c+h\text{Min})]^2 \leq 1, \text{ and}$$

$$(B-15) \quad [W / (a+h\text{Max})]^2 + [Z / (c+h\text{Max})]^2 \leq 1.$$

For the WGS-84 Earth model, a set of values of  $h\text{Min}$ ,  $h\text{Max}$ , and  $D$  were selected so that the error  $E$  is less than 0.01 m for each region  $S$ , after one iteration of the improved Bowring procedure.

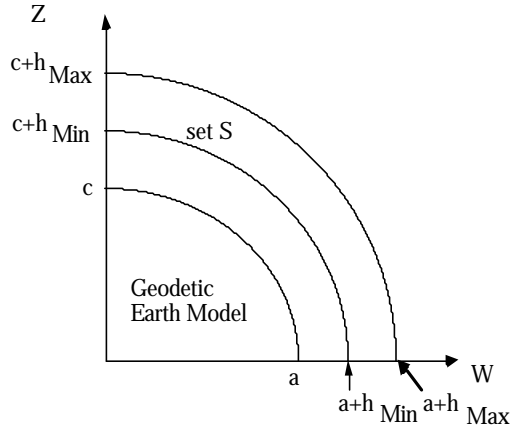


Figure B-1

The resulting values of  $h_{Min}$ ,  $h_{Max}$ , and  $aD/c$  are given in Table B-1.

Table B-1: Optimizing Factors

Region	$h_{Min}$	$h_{Max}$	$aD / c$
1	$-1.*10^5$	$2.*10^6$	1.0026000
2	$2.*10^6$	$6.*10^6$	1.00092592
3	$6.*10^6$	$18.*10^6$	0.999250297
4	$18.*10^6$	$1.*10^9$	0.997523508

For efficiency, the inequalities can be evaluated sequentially in the order given in Table B-1. In this case, only the upper region boundary is evaluated and the inequality can be written in the equivalent form

$$(B-16) \quad \frac{[W]^2 + [Z (a+h_{Max}) / (c+h_{Max})]^2}{(a+h_{Max})^2}$$

This saves a multiply operation. For applications like distributed combat simulation, most of the points will be inside region 1, which extends to 2000 km in altitude. For most applications, it is probably sufficient to ignore the test in (B-16) and just use the region 1 constant 1.0026. Under this policy, the maximum error is less than 42 cm for altitudes less than ten million kilometers.

## 6.2 The improved Bowring method (revisited)

To this point, the equations developed are identical to those of reference [1]. These are now rearranged into a more economical form by defining,

$$(B-17) \quad F = aD/c$$

$$(B-18) \quad Q^2 = (FZ)^2 + W^2.$$

With this definition and the modified first guess (B-11), equation (B-1) can be written

$$(B-19) \quad \tan \phi_{i+1} = \frac{(Z + c e^{-2(FZ/Q)^3})}{(1 - a e^2 W^2/Q^3)}$$

For a given point P with coordinates X,Y,Z, the above equations can be assembled into a step-by-step procedure for a single iteration. It is assumed that fixed constants such as  $e$ ,  $e^2$ , and  $F$ , and terms like  $c$ ,  $e^{-2}$ , and  $\sin^2(67.5 \text{ degrees})$  are all pre-computed.

Step 1. Using (B-8), compute  $W^2$  and  $W=(X^2+Y^2)^{1/2}$ .

Step 2. Use equation (16) along with Table B-1 to determine which region P is in and thereby determine F.

Step 3.  $T=FZ$  and  $T^2$ .

Step 4.  $Q^2=T^2 + W^2$  and Q is computed by taking the square root. Then  $Q^3=Q^2 \cdot Q$ .

Step 5.  $Top = (Z + c e^{-2(TT^2/Q^3)}) / (1 - a e^2 W^2/Q^3)$ .

Step 6.  $\tan \phi = Top/W$ .

Step 7.  $Temp = [Top]^2 + W^2$ .

Step 8.  $Hyp = (Temp)^{1/2}$ .

Step 9.  $S=\sin(\phi) = Top/Hyp$ .

Step 10.  $S2 = S^2$ .

Step 11. Ready to get h. First from (14), get

$$RN = a / (1 - e^2 S2)^{1/2}.$$

Step 12. If  $S2 \sin^2(67.5 \text{ degrees})$ , then

$$h = Z / S + RN (e^2 - 1).$$

or else,

Step 13.  $C=\cos(\phi) = W/Hyp$ , and

$$h = W / C - RN.$$

Step 14. Compute  $\phi$  from  $\tan^{-1}(Top/W)$  and  $\lambda$  from  $\tan^{-1}(Y/X)$ .

Because only a single iteration is used to achieve 1-cm accuracy out to 1,000,000 km, equation (B-3) is not needed.

This algorithm will fail if  $\phi = \pi/2$  or  $-\pi/2$  (or is very close to one of these values). In this case, X is zero (or nearly zero) and both f and h are known immediately. In implementing the algorithm in software, it is important to include tests for these cases and take the appropriate action.

Note that only the squares of X and Y are involved in the procedure and W is positive. The sign of Z then determines the sign of  $\phi$ , so that the procedure yields the proper value of  $\phi$  for all quadrants. Note also that the test on 67.5 degrees latitude defines the region between the Arctic and Antarctic circles. As is addressed in references [2] and [3], the inverse tangents and some of the square roots can be done in-line to further economize on computation time.

### 6.3 Appendix C: The Heikkinen Exact Solution

This formulation was taken directly from Rapp [6] and was adapted to the notation used in this paper.

$$(C-1) \quad W = (X^2 + Y^2)^{1/2}$$

$$(C-2) \quad F = 54c^2Z^2$$

$$(C-3) \quad G = W^2 + (1 - e^2)Z^2 - e^2(a^2 - c^2)$$

$$(C-4) \quad d = e^4FW^2/G^3$$

$$(C-5) \quad s = [1 + d + (d^2 + 2d)^{1/2}]^{1/3}$$

$$(C-6) \quad p = F / (3(s + 1/s + 1)^2 G^2)$$

$$(C-7) \quad q = (1 + 2e^4p)^{1/2}$$

$$(C-8) \quad W_0 = -p e^2 W / (1 + Q) + [.5a^2 (1 + 1/Q) - p(1 - e^2)Z^2 / Q(1 + Q) - .5pW^2]^{1/2}$$

$$(C-9) \quad U = ((W - e^2W_0)^2 + Z^2)^{1/2}$$

$$(C-10) \quad V = ((W - e^2W_0)^2 + (1 - e^2)Z^2)^{1/2}$$

$$(C-11) \quad Z_0 = c^2 Z / aV$$

$$(C-12) \quad h = U(1 - c^2/aV)$$

$$(C-13) \quad \tan \phi = (Z + e'^2 Z_0) / W$$

$$(C-14) \quad \tan \lambda = Y / X$$

## 7. Acknowledgements

This work was conducted by SRI International in support of the SEDRIS Program [1] and was partially sponsored by the Defense Advanced Research Projects Agency (DARPA) under Contract N66001-94-C-6406 with the Naval Command, Control, and Ocean Surveillance Center.

## 8. References

- [1] Synthetic Environment Data Representation and Interchange Specification (SEDRIS), a DoD program sponsored by DMSO, DARPA, and STRICOM.
- [2] Toms, R.M., "An Efficient Algorithm for Geocentric to Geodetic Coordinate Conversion," *Proceedings on Standards for the Interoperability of Distributed Simulations*, Volume I, pp. 635-642, Institute for Simulation and Training, Orlando, FL, September 1995.
- [3] Toms, R.M., "An Improved Algorithm for Geocentric to Geodetic Coordinate Conversion," *Proceedings on Standards for the Interoperability of Distributed Simulations*, Volume 2, pp. 1135-1144, Institute for Simulation and Training, Orlando, FL, February 1996.
- [4] Borkowski, K.M., "Accurate Algorithms to Transform Geocentric to Geodetic Coordinates," *Bulletin Geodesique*, 63, 1989, pp. 50-56.
- [5] Heikkinen, M., "Geschlossene Formeln zur Berechnung Raumlicher Geodatischer Korinaten aus Rechtwinkligen Korrdinaten," *Zeitschrift fur Vermessungswesen*, Vol. 5, 207-211, 1982.
- [6] Rapp, R.H., *Geometric Geodesy, Pt. I*, Department of Geodetic Science and Surveying, The Ohio State University, Columbus, Ohio, 1984.
- [7] Bowring, B.R., "Transformation from Spatial to Geophysical Coordinates," *Survey Rev.*, V 23, 181, 323-327, 1976.
- [8] Lucha, G.V., "On the Consequences of Neglecting Measurement Accuracy Issues in Live and Virtual Interactions," *Proceedings of the 1997 Spring Simulation Interoperability Workshop*, March 1997.
- [9] Hart, J.F., Cheney, E.W., Lawson, C.L., Maehly, H.J., Meesztenyi, C.K., Rice, J.R., Thacher, H.G., and Witzgall, C., *Computer Approximations*, John Wiley & Sons, N. Y., 1978.

- [10] Wise, B., *Geocentric to Geodetic Coordinate Conversions*, BBN Report No. 7756, May 1992.
- [11] Kuo-Chi Lin and Huat Keng Ng, *Interconversions Between Different Coordinate Systems*, White Paper, Institute for Simulation and Training, University of Central Florida, Orlando, FL, 1992.
- [12] Evans, A., and Stanzione, T., "Coordinate Representations for CGF Systems," *Proceedings on Standards for the Interoperability of Distributed Simulations*, Volume I, Institute for Simulation and Training, Orlando, FL, September 1995.
- [13] *Handbook for Transformation of Datums, Projections, Grids and Common Coordinate Systems, Transformation Of*, TEC-SR-7, U.S. Army Corps of Engineers, Topographic Engineering Center, January 1996.
- [14] Birkel, P.A., *SEDRIS Geospatial Reference Model (DRAFT)*, 1998.

### **Author's Biography**

**RALPH M. TOMS** is a Senior Technical Advisor at SRI International who has over 35 years of experience managing and developing real-time embedded systems and simulation products. He specializes in making simulation and embedded system software operate efficiently. He holds undergraduate degrees in Engineering and Mathematics and M.S. and Ph.D. degrees in Applied Mathematics from Oregon State University.