SEDRIS Coordinate Transformation Services

Ralph M. Toms
Kevin I. Smith
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
(650) 859-2852, (650) 859-3855
ralph_toms@sri.com, ksmith@sdd.sri.com

Keywords: Coordinate Conversion/System, Correlation, Modeling and Simulation Interoperability

ABSTRACT: *Environmental data are processed and archived as three-dimensional, gridded data in a number of augmented, projection-based coordinate systems. Such systems are employed to simplify the dynamic equations governing the physical phenomenon being studied. Measured data are used to provide initial conditions for interpolation by numerical integration to populate authoritative databases. Construction of a Synthetic Natural Environment (SNE) database in real-world coordinates requires coordinate transformation of the archived data. Since augmented, projection-based systems are not real-world systems, this process distorts the transformed grid structure. Numerous intermediate points may be needed to ensure an accurate, loss-less transformation. These transformations involve many transcendental functions. Legacy software utilizes iterative techniques for several transformations, and each iteration is computationally expensive. Emerging requirements for joint distributed simulation and real-world operational support imply steadily shrinking timelines for developing or modifying SNE databases.*

The SEDRIS Program provides coordinate system services to support the population of SNE databases for those coordinate systems listed in the SEDRIS Geospatial Reference Model. This paper contains revised mathematical formulations of the initial development phase for these transformations that lead to new accurate and efficient algorithms. All iterative procedures have been eliminated for the domain of practical applications. Several intervening transcendental functions have been eliminated either by mathematical reformulation or by using multi-dimensional rational approximations. Comparative timing data are contained in the paper for legacy approaches and the new methods.

1. Introduction

The Defense Modeling and Simulations Office and the DoD Modeling and Simulation (M&S) Executive Agents for Environmental Representation are developing the Synthetic Environment Data Representation and Interchange Specification (SEDRIS). This effort provides an innovative solution to several long-standing problems in M&S that impact environmental data interchange, access to repositories, interoperability, and reuse. Archived environmental data are stored in a variety of coordinate systems, both real-world and projection-based. SEDRIS environmental data file transactions must be accurate and efficient to meet current and future simulation requirements. For a discussion of the need for accuracy see Lucha [1].

The purpose of the SEDRIS Coordinate Transformation Project is to provide enhanced SEDRIS-specific algorithms, software, and documentation to support coordinate transformation requirements. The principal goals of this project are to provide accurate algorithms and implementations with increased efficiency over legacy systems.

Many coordinate system transformation packages already exist. Some are authoritative in the sense that a government agency has developed, improved, and maintained them using subject-matter experts over a substantive period of time. Examples of such government agencies include NIMA, TEC, and USGS.*

Authoritative coordinate transformation codes for mapping, charting, and geodesy (MC&G) applications are designed to provide very accurate results over a specific, but still relatively broad, set of coordinate transformations. SEDRIS only requires a proper subset of the procedures used for MC&G. SEDRIS

* National Imaging and Mapping Agency, U.S. Army Topographical Engineering Center, and U.S. Geological Survey

also requires a number of transformation services that lie outside the field of MC&G—those associated with a variety of space applications, for example. Part of the project involves abstraction and compilation of those transformations that are SEDRIS-specific. These are defined, along with the Earth reference models (ERMs), in the SEDRIS Geospatial Reference Model (GRM) [2]. Several different formulations and software implementations are available for coordinate transformations. As part of this project, the best formulations and implementations were abstracted, modified, and implemented in ANSI C.

Three phases are involved in designing SEDRIS coordinate transformation software: determination of global constants, calculation of relative constants (ERM-specific), and user input specification (initialization loop) and file processing (inner loop). To promote readability in the code, the initialization loop was coded with no attempt to optimize performance. Development of algorithms for the inner loop is the principal topic of this paper.

An error metric is needed to discuss approximation errors or to compare two alternative formulations in terms of accuracy. Both angular and positional components must be computed in each system. To provide an equitable basis for comparison, angular errors are converted to position errors so that a Euclidean error metric can be defined entirely in terms of position errors. The NIMAMUSE™ software DTCC4.1 is used as the "gold standard" against which the accuracy and performance of the formulations in this project are compared.*

Efficiency is somewhat more difficult to evaluate. For a single-machine environment, raw execution times can be used for direct comparison. As part of this program, we have developed a tool, coded in ANSI C, that permits timing of standard operations and mathematical functions on a wide class of computer environments. This tool has been used to design more efficient coordinate transformation algorithms. It permits normalized comparison of procedures and standard operations between one computational environment and another. In addition, it permits the performance assessment of an entire coordinate transformation subprogram. Because the execution times are normalized in terms of equivalent double floating point multiply operations, comparisons can be made between machine environments for alternative coordinate transformation algorithms. This permits comparisons of performance over a class of machine environments. Such comparisons must be done on dedicated machines running under the same operating system environment.

In addition to efficient algorithm design, coding practices can also have a deleterious effect on performance. While some practices may appear to waste processing time, an optimizing compiler may recognize and repair the situation automatically. This makes it difficult to evaluate codes without end-to-end testing and comparison.

In this paper, we reference existing documentation as much as possible when defining the mathematics of each transformation. The intent herein is to focus on those reformulations and new approximations that lead to more compact and efficient procedures.

1.1 Fundamentals and notation

Several reference ellipsoids have been used in astrogeodetic work [2,3]. These all have the form

(1-1) $$\left(\frac{X}{a}\right)^2 + \left(\frac{Y}{a}\right)^2 + \left(\frac{Z}{c}\right)^2 = 1 .$$

Particular values of *a* and *c* are given in the SEDRIS GRM [2] for all 21 ERMs. Figure 1-1 depicts the geometry of the geocentric (Cartesian) system and the geodetic system in three dimensions.

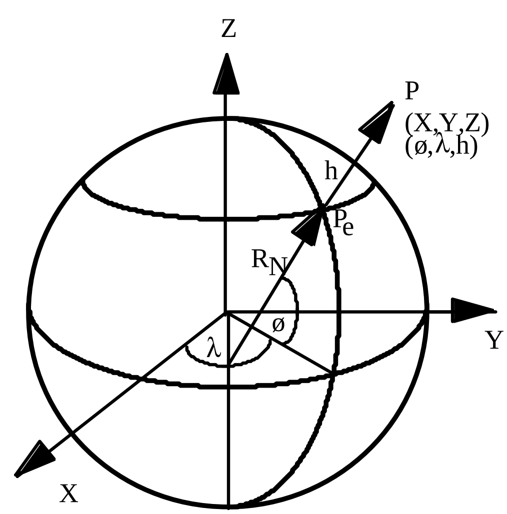

Figure 1-1

* Coordinate transformation software NIMAMUSE™ is available from the NIMA at http://www.nima.mil/geospatial/SW_TOOLS/NIMAMUSE/

The geocentric coordinates of a point P are (X,Y,Z), and the corresponding geodetic coordinates of P are (h, f, l), where f is latitude, l is longitude, and h is the height above the reference ellipsoid. The line between P and P_e. is orthogonal to the tangent plane at P_e.

The transformation from geodetic to geocentric coordinates is given by [3],

(1-2) $$X = (R_N + h)\cos f \cos l \ ,$$

(1-3) $$Y = (R_N + h)\cos f \sin l \ ,$$

(1-4) $$Z = \left(\frac{c^2 R_N}{a^2} + h\right)\sin f \ , \quad \text{where}$$

(1-5) $$R_N = a\left[1 - e^2 \sin^2 f\right]^{-\frac{1}{2}} \qquad \text{and}$$

(1-6) $$e^2 = \left(\frac{a^2 - c^2}{a^2}\right) \ .$$

For the inverse transformation, l is given by

(1-7) $$l = \tan^{-1}\left(\frac{Y}{X}\right), \text{ and } -p \le l \le p \ .$$

While closed-form inverse equations for determining f and h exist, they are not very efficient. More efficient algorithms for determining h and f have recently been developed [4,5,6,7] and are implemented in the SEDRIS software.

Due to the symmetry of the problem in X and Y, it is sufficient to utilize a meridional section of the ellipsoid to determine f and l. This system is depicted in
Figure 1-2.

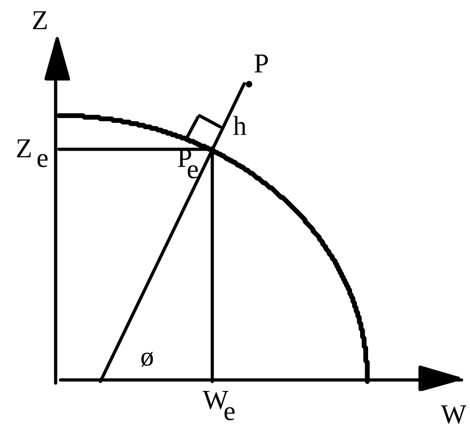

Figure 1-2

The meridional ellipse is defined by

(1-8) $$\left(\frac{W}{a}\right)^2 + \left(\frac{Z}{c}\right)^2 = 1, \text{ and where}$$

(1-9) $$W = \left(X^2 + Y^2\right)^{\frac{1}{2}} .$$

Once ø has been determined, h can be computed from

(1-10) $$h = \left(\frac{W}{\cos f}\right) - R_N \ .$$

for non-polar regions. In polar regions, it is preferable to use

(1-11) $$h = \frac{Z}{\sin f} + R_N\left(e^2 - 1\right).$$

Reference [7] suggests that (1-10) be used for $f \le \left|\frac{p}{4}\right|$ and (1-11) otherwise.

1.2 Error definition and assessment

In this section, we discuss a methodology for assessing the error made in a coordinate transformation due to using one or more approximations in the process. Application to the error definitions given here will implicitly include roundoff error but not coding errors. We assume that double precision is used in all calculations; single precision can introduce errors larger than desired at some points.

Both positional and angular errors need to be addressed. Mixtures of the two types of errors are combined into a single-position error metric. 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 (h_a, f_a, l_a). Using the exact relations (1-2), (1-3), and (1-4), 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,

(1-12) $$E = \left[(X - X_a)^2 + (Y - Y_a)^2 + (Z - Z_a)^2\right]^{\frac{1}{2}}.$$

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 concept of an error ball can be used for transformations from an ERM to a projected coordinate system. In such cases, the three-dimensional surface of the ellipsoidal ERM maps into two dimensions. An exact point on the surface $(0, f, l)$ maps into an exact point (x, y) in the projected coordinate system when an exact mapping is used, or into approximate point (x_a, y_a) when an approximation is used. The error ball in the projected space becomes $R^2 = (x - x_a)^2 + (y - y_a)^2$.

The forward transformations for Lambert conformal and polar stereographic spaces have exact closed-form representations. All of the inverses of these transformations involve either iterative procedures or finite series expansions and are approximate. To evaluate inverses, a point $(0, f, l)$ is projected exactly to create a point (x, y) in the projected space. This point is mapped using the inverse transformation to an approximate point $(0, f_a, l_a)$. The exact and approximate geodetic points can now be compared in terms of the position metric of (1-12). The transverse Mercator and UTM projections present a difficulty since both the forward and inverse transformations do not have closed-form representations.
There are two choices: use an authoratitive source for comparison (DTCC.4.1), or map a point forward and then back to see if approximately the same point is achieved.

The procedure used to evaluate the maximum error involves introducing a uniform lattice of points in either a two- or three-dimensional closed and bounded region by defining a uniform grid on each of the coordinate axes. The error is computed using the appropriate method from above for each lattice point and the maximum determined. As the lattice is made successively finer, the maximum error is determined to any reasonable degree of accuracy. The numerical assessment for finding the maximum errors by sampling a fine grid has additional benefits: first, the effect of roundoff is inherently included; second, the process of sampling on a fine grid may uncover coding or formulations errors that may otherwise be difficult to determine; and finally, care should be taken to sample near known singular points, such as the poles and equator.

2.0 Efficient Evaluation Of Special Functions

In coordinate system computations, certain computationally expensive functions appear repeatedly. Many of these functions are expensive to compute due to the presence of several transcendental functions in the formulation. On occasion, mathematical identities or equivalent formulations can be developed that reduce or eliminate completely the number of transcendental functions required. While it is always best to eliminate superfluous transcendental functions, this is not always possible. However, many of the required functions need only be computed for a very restricted range of arguments. This permits localized rational function approximations to be used that maintain more than sufficient accuracy. This section contains several procedures for reducing computational requirements in such cases.

2.1 Reducing trigonometric evaluations

Evaluation is often required for a finite series of the form

(2-1) $$f(x) = \sum_{n=1}^{4} A_{2n} \sin(2nx),$$

which is costly to evaluate in this form due to the presence of four sine function evaluations. By repeated application of standard multiple angle identities, this can be written in a form more efficient for computation. Letting $S = \sin(f)$ and $C = \cos(f)$, then

(2-2) $$f(x) = SC\left\{B_1 + S^2\left[B_2 + S^2(B_3 + S^2 B_4)\right]\right\},$$ where,

(2-3) $$B_1 = 2A_2 + 4A_4 + 6A_6 + 8A_8,$$

(2-4) $$B_2 = -8A_4 - 32A_6 - 80A_8,$$

(2-5) $$B_3 = 32A_6 + 192A_8, \text{ and}$$

(2-6) $$B_4 = -128A_8.$$

Snyder [8] recommends using a similar sum in terms of $\sin(2f)$ and $\cos(2f)$. However, $\sin(f)$ and $\cos(f)$ are usually needed anyway so that $f(x)$ can be computed without the two additional trigonometric calls [9].

2.2 Rational function approximation

A rational function approximation to another function $F(x)$ on some region R is a ratio of polynomials P_n and Q_m of order *n* and *m* respectively such that

(2-7) $$F \approx f = \frac{P_n}{Q_m},$$

where the region *R*, the function f, and the polynomials may be multidimensional. In this paper, both one- and two-dimensional approximations are used. For a function F of a single variable defined on some interval I, (2-7) becomes

(2-8) $$F(x) \approx f(x) = \frac{P_n(x)}{Q_m(x)}.$$

The $n+m+2$ unknown coefficients of the polynomials are selected to minimize some error criterion associated with I. The error may be absolute or relative. Often the classical p norms for a normed linear space are used on a finite sample of points in I. For $p=2$, this results in a non-linear least-squares approximation. Similarly, when $p=\infty$, the maximum absolute error is minimized on a finite set of points in I. In either case, a set of non-linear algebraic equations needs to be solved. In this paper, the approximating functions are linearized so that a linear system of equations is solved on I. For functions that are known to be even or odd, special forms of the polynomials are used to gain further efficiencies [10]. Good approximations are usually more dependent on the form and order chosen for *P* and *Q* rather than the particular norm used. While the approximations developed in this paper may be improved upon, it is doubtful that any substantive reduction in error or run time can be achieved simply by changing the norm or using non-linear techniques.

The above process is readily generalized to two dimensions so that (2-8) becomes

(2-9) $$F(x,y) \approx f(x,y) = \frac{P_n(x,y)}{Q_m(x,y)}, \text{ and}$$

the interval is replaced by a two-dimensional region *R*.

The unknown coefficients in each polynomial are determined by a linearization procedure. This is illustrated for the one-dimensional case. The equation

(2-10) $$F(x) = \frac{\left(a_1 + a_2x + a_3x^2\right)}{\left(1 + a_4x + a_5x^2\right)}$$

is rewritten as

(2-11) $$F(x)\left(1 + a_4x + a_5x^2\right) - \left(a_1 + a_2x + a_3x^2\right) = 0.$$

After rearrangement,

(2-12) $$a_1 + a_2x + a_3x^2 - a_4\,xF(x) - a_5F(x)x^2 = F(x).$$

Any suitable set of five points in I can be used to construct five linear algebraic equations in five unknowns (the coefficients). These are solved using the direct LU decomposition method [1]. Then

(2-13) $$f(x) = \frac{\left(a_1 + a_2x + a_3x^2\right)}{\left(1 + a_4x + a_5x^2\right)}$$

is the resulting approximating function.

In the interest of saving arithmetic operations, it is useful to rewrite *f(x)* as a continued fraction. In this paper, sufficient accuracy was achieved by using second-order (or less) polynomials in one or two dimensions. The procedure for obtaining a rational fraction form is illustrated below for the one-dimensional case.

From elementary algebra, long division of the two polynomials in (2-13) yields

(2-14) $$f(x) = c_1 + \frac{\left(c_2x + c_3\right)}{\left(c_4 + x\left(c_5 + x\right)\right)}.$$

This form minimizes the number of floating point operations required to compute $f(x)$. Sometimes the error on I can be decreased by optimizing c_1 to minimize the absolute error on the interval. Some care must be taken to make sure that the denominator in (2-14) does not have a zero in the range of interest.

The coefficients in (2-14) are related to those in (2-13) by the set of equations (2-15):

$$(2\text{-}15)\quad \left\{ \begin{array}{lll} c_1 = \dfrac{a_3}{a_5} & c_2 = \dfrac{\left(a_2 + \dfrac{a_4 a_3}{a_5}\right)}{a_5} & \\ c_3 = \dfrac{\left(a_1 + \dfrac{a_3}{a_5}\right)}{a_5} & c_4 = \dfrac{1}{a_5} & c_5 = \dfrac{a_4}{a_5} \end{array} \right\}.$$

2.3 Approximation of R_N

The function $R_N = a\left[1 - e^2 \sin^2 f\right]^{-\frac{1}{2}}$ is common to most of the coordinate transformations addressed in this paper. The argument $A = \left(1 - e^2 \sin(f)\right)$ of the square root is in the closed interval

$$(2\text{-}16)\qquad \left[\left(1 - e^2_{\max} \sin^2(f)\right),\ 1.0\right],$$

where $e^2_{\max}$ is the maximum value of e^2 taken over all 21 ERMs contained in the SEDRIS GRM [3,4]. This value of $e^2_{\max}$ occurs for Clarke's Spheroid of 1860 and is less than 0.082483400, so that

$$(2\text{-}17)\qquad 0.99319649 \quad A \quad 1.0\ .$$

Newton's method for the problem of finding the square root of A in one iteration is given by

$$(2\text{-}18)\qquad \sqrt{A} \sim x_1 = \frac{1}{2}\left(x_0 + \frac{A}{x_0}\right),$$

where the initial estimate is given by some appropriate first guess. Usually x_0 is taken to be of the form $a + bA$ and the a and b are selected to minimize the error on the interval after one iteration. In this case, because the length of the interval involved is relatively small, sufficient accuracy can be obtained by setting $b = \frac{1}{2}$ and just optimizing a for one iteration. In this way, one multiplication is avoided. The initial guess then becomes

$$(2\text{-}19)\qquad x_0 = a + \frac{A}{2},$$

where a is selected to minimize the absolute error on the interval of (2-15) after applying (2-17) and (2-18).

The optimal value of a obtained is 0.4999972172 with the maximum absolute error being less than 0.3992×10^{-11} for all values of A on the interval.

A further economy can be obtained from a rearrangement of terms. Let

$$(2\text{-}20)\qquad z_0 = \frac{(2A+1)}{4} - \frac{e^2 \sin^2(f)}{4}\text{, then}$$

$$(2\text{-}21)\qquad \sqrt{A} \sim x_1 = z_0 + \frac{\left(0.25 - \dfrac{e^2 \sin^2(f)}{4}\right)}{z_0}.$$

Now assuming that $\frac{e^2 \sin^2(f)}{4}$ has already been computed, the rest of the coefficients are constants. Specifically, $\frac{(2A+1)}{4}$ =0.4999986089 and the form (2-21) requires one multiply, one divide, and three additions. This results in a fifty percent or more reduction in execution time when computing $\sqrt{A}$.

2.4 The function G

The function

$$(2\text{-}22)\qquad G(f, e) = \tan\left(\frac{p}{4} + \frac{f}{2}\right)\left[\frac{(1 - e\sin(j))}{(1 + e\sin(j))}\right]^{\frac{e}{2}},$$

where f is latitude, appears in several coordinate transformations [3,8]. It is often coded as written and the result is computationally intensive. For current workstations equipped with a floating point unit (FPU), $G(f,e)$ takes the same time as between 32 and 45 floating point multiplies, depending on the machine. On machines that compute mathematical functions without FPUs, the computation time is even slower, on the order of a factor of four.

There are two ways of speeding up these computations with essentially no loss in accuracy. In the first case, it is noted that $\sin(f)$ and $\cos(f)$ are always needed for other calculations. By using the half-angle identities, the following identity is obtained:

$$(2\text{-}23)\qquad \tan\left(\frac{p}{4} + \frac{f}{2}\right) = \left[\frac{1 + \sin(f)}{1 - \sin(f)}\right]^{\frac{1}{2}}.$$

Using (2-23), the first factor in $G(f,e)$ is computable on most of its range without calling any additional trigonometric subroutines. Near $\frac{p}{2}$, the denominator approaches zero and the form on the left side of (2-23) should be used.

The second factor in $G(f,e)$ is not quite so easy. In this case, the factor

$$F(f,e) = \left[\frac{(1-e\sin(j))}{(1+e\sin(j))}\right]^{\frac{e}{2}} \qquad \text{(2-24)}$$

is approximated with a rational function over all possible values of e and f. It can be shown that when $f \le 0$, then

$$F(f,e) = \frac{1}{F(|f|,e)} \qquad \text{(2-25)}$$

so that only f in the first quadrant need be addressed.

For all 21 ERMs in the SEDRIS GRM, the range of values for e are bounded so that $e_{\min} = .081472980982652$ and $e_{\max} = .082483400044$, which defines a small interval of length 0.0010141.

$F(f,e)$ is approximated by a two-dimensional rational function $f(f,e)$ over the rectangular region R defined by

$$[e_{\min}\ ,\ e_{\max}]\ X\ \left[0.\ ,\ \frac{p}{2}\right]. \qquad \text{(2-26)}$$

Guided by the first few terms in the Taylor series for $F(f,e)$, the form of the approximating function $f(f,e)$ is taken to be

$$f(f,e) = \frac{(a_1 + a_2 e^2 s + a_3 e^2 s^2 + a_4 e^3 s^2 + a_5 e^4 s^2)}{(1 + b_2 e^2 s + b_3 e^2 s^2 + b_4 e^3 s^2 + b_5 e^4 s^2)}. \qquad \text{(2-27)}$$

The linearization procedure outlined above is used to generate nine linear coefficient equations for nine selected points in R that are then solved by LU decomposition. It is natural to select nine evenly spaced points on the rectangle R.

For a particular ERM, e is known and (2-27) reduces to a one-dimensional rational approximation, as in (2-13). That is,

$$f(f,e) = \frac{(a_1 + a_2 e^2 s + (a_3 e^2 + a_4 e^3 + a_5 e^4)s^2)}{(1 + b_2 e^2 s + (b_3 e^2 + b_4 e^3 + b_5 e^4)s^2)}. \qquad \text{(2-28)}$$

Evaluation of (2-28) over R yields maximum errors that are slightly too large. As a consequence, R was subdivided into two rectangles, R_1 and R_2, defined respectively by

$$R_1 \text{ is } [e_{\min}\ ,\ e_{\max}]\ X\ [0.\ ,\ f_c] \text{ and} \qquad \text{(2-29)}$$

$$R_2 \text{ is } [e_{\min}\ ,\ e_{\max}]\ X\ \left[f_c\ ,\ \frac{p}{2}\right]. \qquad \text{(2-30)}$$

With some experimentation, f_c was selected to be 0.6195736111 (35.5 deg), which results in almost equal maximum error on R_1 and R_2. Each of R_1 and R_2 are sampled at nine points on a 3-by-3 uniform grid, resulting in two sets of coefficients that are defined globally. Once the ERM is selected, then the coefficients in (2-27) can be computed as relative constants so that (2-14) and (2-15) can be used to define the continued fraction that minimizes operations.

The coefficients developed for R_1 are

$$\begin{aligned}
a_1 &= \ \ 0.100000000000349 * 10^{1}\\
a_2 &= -0.643155723158021\\
a_3 &= -0.333332134894985\\
a_4 &= -0.241457540671514 * 10^{-04}\\
a_5 &= \ \ 0.143376648162652\\
a_6 &= \ \ 0.356844276587295\\
a_7 &= -0.333332875955149\\
a_8 &= \ \ 0.000000000000000\\
a_9 &= \ \ 0.000000000000000
\end{aligned} \qquad \text{(2-31)}$$

The coefficients for R_2 are

(2-32)

$$
\begin{aligned}
a_1 &= 0.999999999957885\\
a_2 &= -0.115979311942142\times 10^{1}\\
a_3 &= -0.333339671395063\\
a_4 &= 0.276473457331734\times 10^{-3}\\
a_5 &= 0.587786240368508\\
a_6 &= -0.159793128888088\\
a_7 &= -0.333333465982150\\
a_8 &= 0.746505041501704\times 10^{-4}\\
a_9 &= -0.701559218182283\times 10^{-1}.
\end{aligned}
$$

Using these coefficients in (2-28) over R_1 and R_2 for all e in the GRM, the maximum absolute error was determined to be less than $0.7\text{x}10^{-10}$. For the WGS-84 ERM, this error was less than $0.27\text{x}10^{-11}$.

3. Coordinate Transformations

The Phase One SEDRIS coordinate transformation software supports transformations between the following coordinate systems and their inverses: geocentric (GCC) and geodetic (GDC), universal transverse Mercator (UTM) and GDC, polar stereographic (PS) and GDC,and finally Lambert conformal (LCC) and GDC.

3.1 Geocentric and geodetic

Efficient algorithms given in references [6,7] for both transformation directions have been implemented for the SEDRIS coordinate transformation software for all 21 ERMs in the GRM.

3.2 UTM and geodetic

The forward transformation from geodetic to UTM coordinates is mathematically the formulation of [9]. The inverse transformation is based on the inverse power series approach of [3,8]. The effect of a minor error in one of the equations in [9] is eliminated by the use of (2-2) to (2-6). Care was taken in implementation to move all calculations of constants to the outer loop.

3.3 Polar stereographic and geodetic

The equations used in this section are adapted from references [3,8] and are refined for computational efficiency. Figures 3-1 and 3-2 depict the coordinate axes for the northern and southern hemispheres.

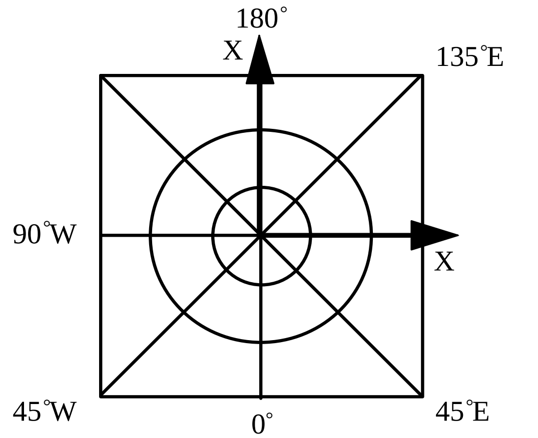

Figure 3-1 Polar Stereographic System, Southern Hemisphere (not to scale)

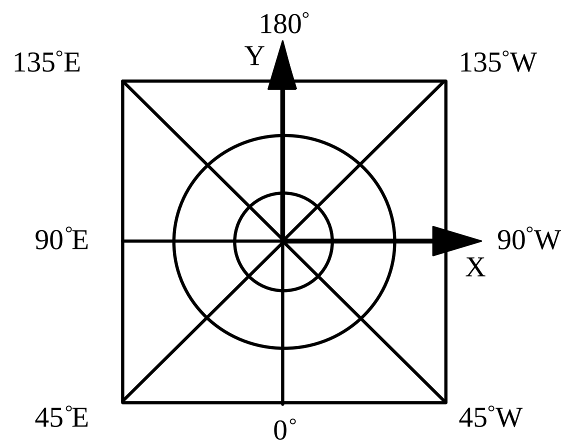

Figure 3-2 Polar Stereographic System, Northern Hemisphere (not to scale)

3.3.1 The forward transformation

Given the point (f, l) on the surface of the ERM, the corresponding polar stereographic coordinates are given by the following equations:

(3-1) $$K = \frac{2a^2}{c}\left[\frac{(1-e)}{1+e}\right]^{\frac{e}{2}}$$ and

(3-2) $$R = KG(|f|, e),$$

where $G(f, e)$ is as defined in Section 2.4; and

(3-3) $$x = R\sin(f),$$

(3-4) $$y = -R\cos(f)$$ (northern hemisphere), and

(3-5) $$y = R\cos(f)$$ (southern hemisphere),

where the sign of f determines the hemisphere.

3.3.2 The inverse transformation

Given the point (x,y) in polar stereographic coordinates and the hemisphere, the point $(f,\ l)$ on the ERM can be determined. The longitude is readily computed:

(3-6) $l = -A\tan\left(\frac{x}{y}\right)$ (in the northern hemisphere),

and

(3-7) $l = A\tan\left(\frac{x}{y}\right)$ (in the southern hemisphere),

where care must be taken to avoid the singular cases. This can be done in C by using the double-argument inverse tangent or by providing in-line tests (which are usually faster).

For determining latitude, the inverse power series is used as in references [3,8]. However, the several intervening trigonometric functions are eliminated and the series of (2-2) is used instead of the one suggested by Snyder [8] because $\sin(c)$ and $\cos(c)$ can be computed without requiring trigonometric evaluations:

(3-8) $$R = \sqrt{x^2 + y^2}\ .$$

In [3,8], an auxiliary angle z is introduced in the definition of the polar stereographic transformation, such that

(3-9) $$\tan\left(\frac{z}{2}\right) = KR, \text{ and}$$

(3-10) $$c = \frac{p}{2} - z\ ,$$

where z is computed by taking the inverse tangent in (3-4).

From (3-10) and elementary trigonometry,

(3-11) $$\sin(c) = \cos(z), \text{ and}$$

(3-12) $$\cos(c) = \sin(z).$$

Now $\sin(z)$ and $\cos(z)$ can be computed directly from half-angle identities and (3-9). That is,

(3-13) $$\sin(z) = \frac{2\tan(\frac{z}{2})}{1+\tan^2(\frac{z}{2})} = \frac{2KR}{1+(KR)^2}, \text{ and}$$

(3-14) $$\cos(z) = \frac{1-\tan^2(\frac{z}{2})}{1+\tan^2(\frac{z}{2})} = \frac{1-(KR)^2}{1+(KR)^2}\ .$$

Following the formulation given in [8], the latitude is given by

(3-15) $$f = c + \sum_{n=1}^{4} A_{2n}\sin(2nc)\ ,$$

and the coefficients A_n are given by

(3-16) $$A_2 = e^2(\frac{1}{2} + e^2(\frac{5}{24} + e^2(\frac{1}{12} + \frac{13}{360}e^2)))\ ,$$

(3-17) $$A_4 = e^4(\frac{7}{48} + e^2(\frac{29}{240} + \frac{811}{11520}e^2))\ ,$$

(3-18) $$A_6 = e^6(\frac{7}{120} + \frac{81}{1120}e^2), \text{ and}$$

(3-19) $$A_8 = \frac{4279}{161280}e^8.$$

To avoid the repeated sine function calls in (3-15), the formulation of (2-2) through (2-6) is used to evaluate the sum in (3-15).

3.4 Lambert conformal and geodetic

The Lambert conformal projected system can have one or two standard parallels associated with it (LCC1 and LCC2) [3,8]. The cases are very similar. Due to space limitations, we address only the case of two standard parallels.

3.4.1 Initialization constants

The equations for the initialization constants are adapted from [3] and uses the fact that for $|j| \geq \frac{}{2}$ so that $\cos^2 f = 1 - \sin^2 j$ The notation R_{N_1} denotes equation (1-5) evaluated at the point j_1. Let L, R_0, R, and Λ be defined by

(3-20) $$L = \frac{\frac{1}{2}\ln\left(\frac{R^2{}_{N_1}\cos^2(f_1)}{R^2{}_{N_2}\cos^2(f_2)}\right)}{\ln\left(\frac{G(f_2,e)}{G(f_1,e)}\right)}\ ,$$

(3-21) $$R_0 = \left[\left(\frac{R^2{}_{N_1}\left(1-\sin^2(f_1)\right)}{L^2}\right)\left(\frac{G(f_1,e)}{G(f_0,e)}\right)\right]^{\frac{L}{2}},$$

(3-22) $R = \left[\left(\frac{R^2{}_{N_1} \left(1 - \sin^2(f_1)\right)}{L^2} \right) \left(\frac{G(f_1, e)}{G(f, e)} \right) \right]^{\frac{L}{2}}$, and

(3-23) $\Lambda = l - l_0$.

This formulation avoids a number of trigonometric and square root evaluations.

3.4.2 Forward transformation

Given a point (f, l), the Lambert conformal conic coordinates for two standard parallels are given by

(3-24) $x = R \sin(L\Lambda)$, and

(3-25) $y = R_0 - R \cos(L\Lambda)$.

3.4.3 The inverse transformation

The inverse transformation of the point (x,y) to geodetic coordinates is given below:

(3-26) $$l = \frac{A\tan\left(\frac{x}{R_0 - y}\right)}{L} - l_0,$$

where care must be taken to apply the right sign to the longitude. The double-argument inverse tangent routine will suffice, but it is generally faster to use the single-argument routine and determine the sign in-line.

Latitude is found by using an inverse power series expansion in a fashion nearly identical to that used in Section 3.3.2 for the inverse of the polar stereographic transformation. The specific formulation is adapted from Snyder [8]. An auxiliary variable z is introduced such that,

(3-27) $$\tan\left(\frac{z}{2}\right) = \frac{1}{G(e, f_1)} \left[\frac{R^2}{\frac{R^2{}_{N_1}\left(1 - \sin^2(f_1)\right)}{L^2}} \right]^{\frac{1}{2L}},$$

where

(3-28) $R^2 = x^2 + (R_0 - y)^2$.

With z defined as in (3-27), the value of j is determined by using equations (3-10) through (3-19).

4.0 Performance and Accuracy

The above algorithms have been implemented in ANSI C and compared to the DTCC 4.1 software (also in C). A Dual Pentium 90 based system was used as a test platform. In all cases, the maximum optimization level and in-line function options were used.

To assess performance, we timed only the inner loop (file processing)—where essentially all the processing time is consumed. To obtain timing data, we cycled a large file of 51,20000 points 16 times to mitigate run-to-run variations. The ratio of the time to process transformations using DTCC 4.1 codes to the corresponding time to process the SEDRIS codes is used as a measure of performance improvement or degradation.

To determine accuracy, we computed the absolute value of the position differences between DTCC 4.1 results and SEDRIS results (two or three dimensions) at each point and then computed the maximum difference. The maximum difference was less than 1mm for all the points tested.

Table 4-1:

Transformation	Time Ratio
GCC--->GDC	1.924
GDC--->GCC	1.066
GDC--->UTM	1.909
UTM--->GDC	1.815
GDC--->SPC	1.358
SPC--->GDC	1.358
GDC--->LCC2	1.406
LCC2--->GD	1.250

The data in Table 4-1 should be considered preliminary, in the sense that the timing tests have only been done for one machine environment. However, results of the timing program tests indicate that the time ratios will not change.

5.0 Acknowledgments

The principal part of this work was done under contract with NAWC Training Systems Division, Contract No. N61339-98-C-0024. The authors wish to thank the management of the SEDRIS program—Farid Mamaghani-IDA/DMSO, Larry Grosberg-STRICOM, Paul Foley-Mitre/DMSO, Karen Williams-NIMA/DMSO/TMPO, and Jerry Lenczowski-NIMA—for their help and support on this project. Thanks also to the Systems Development Division of SRI for supporting the development of this paper.

6.0 References

[1] 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.

[2] Birkel, P.A., *SEDRIS Geospatial Reference Model* (DRAFT), 1998.

[3] *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.

[4] 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.

[5] 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.

[6] Toms, R. M., “New Efficient Procedures for Geodetic Coordinate Transformations,” *Proceedings on Standards for the Interoperability of Distributed Simulations,* Volume 3, pp. 1024-1033, Institute, for Simulation and Training, Orlando, FL, March 1998.

[7] Toms, R. M., “Efficient Procedures for Geodetic Coordinate Transformations,” *Military Applications Society of INFORMS, Proceedings of the First National Meeting*, University of Alabama at Huntsville, Huntsville, AL, 19-21 May 1998.

[8] Snyder, J. P. “*Map Projections-A Working Manual,*” U.S. Geological Survey Professional Paper 1395, United States Government Printing Office, Washington, D.C., 1987.

[9] R. M. Toms, "Efficient Transformations from Geodetic to UTM Coordinate Systems,” *Proceedings on Standards for the Interoperability of Distributed Simulations*, Fifteenth DISWS, Institute for Simulation and Training, Orlando, FL, September 1996.

[10] 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, NY, 1978.

Author Biographies

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.

KEVIN I. SMITH is a Research Engineer at SRI International who has expertise in design and implementation of the interface between computers and embedded systems. He specializes in software packaging of mathematical concepts to provide advanced functionality through a simple GUI. He holds a B.S. degree in Engineering from Harvey Mudd College.