1-1-1992

Interconversions Between Different Coordinate Systems

Kuo Chi Lin

Huat Keng Ng

INSTITUTE FOR SIMULATION AND TRAINING

INTERCONVERSIONS BETWEEN DIFFERENT COORDINATE SYSTEMS

Written by
Kuo-Chi Lin, Huat Keng Ng

July 1992
IST-TR-92-24

IST

INSTITUTE FOR SIMULATION AND TRAINING

Interconversions Between Different Coordinate Systems

Written by

Kuo-Chi Lin, Huat Keng Ng

July 1992

IST-TR-92-24

Institute for Simulation and Training • 12424 Research Parkway, Suite 300 • Orlando, Florida 32826

University of Central Florida • Division of Sponsored Research

INTERCONVERSIONS BETWEEN DIFFERENT COORDINATE SYSTEMS

Kuo-Chi Lin, Huat Keng Ng
Institute for Simulation and Training
University of Central Florida
Orlando, Florida 32826

Abstract

There are several different coordinate systems which can be used to describe the position, orientation, and motion of the entities in a simulation exercise. The coordinate systems that are referenced in this paper are the geocentric, geodetic and topocentric coordinate systems. A detailed study made on previously published coordinate conversion algorithms and any encountered problems are presented here. In converting geocentric to geodetic coordinates, four different algorithms yielding the same results are presented. The conclusions drawn from these analyses illustrate that by deriving a set of parametric equations and then utilizing Newton-Raphson's convergence algorithm results in the fastest and most accurate geocentric to geodetic coordinate conversion. In the case of a topocentric to geocentric conversion, it was discovered that the referenced algorithm was inaccurate. The corrected equations are given in this paper.

Introduction

The advent of affordable intercomputer communications networks has made possible the interconnection of simulators so as to allow for real-time interactive training. The precursor to Distributed Interactive Simulation (DIS) was a Defense Advanced Research Projects Agency (DARPA) sponsored program call Simulator Networking (SIMNET). In the SIMNET program, DARPA successfully demonstrated the feasibility of interconnecting multiple distributed simulators, primarily ground based armor vehicles, via a local area network (Ethernet) such that the simulators could interact in real-time. DIS is based upon the foundation of SIMNET and will be enhanced and expanded to provide the standard for future communication of simulators. Due to the expansion of a DIS exercise, simulators will be operating at larger geographic distances. As a result of this requirement, the geocentric coordinate system was chosen to

be the earth-fixed-axis coordinate system vice the flat-earth topocentric coordinate system in SIMNET exercises.

The geocentric coordinate system is defined as the earth-fixed coordinate system with the origin at the centroid of the earth, the x-axis passing through the Prime Meridian at the Equator, the y-axis passing through 90 degrees East longitude at the Equator, and the z-axis passing through the North Pole. The topocentric coordinate system is defined with the origin centered at a selected point on the Earth's surface and aligned at the selected point with East, North, and Up.

In order to establish the coordinate transformation between the two coordinate systems described, a third coordinate system, the geodetic coordinate, will be introduced. The geodetic coordinate is defined using three quantities: latitude, longitude, and the geodetic height. The latter defines the position of a point on the surface of the Earth with respect to the reference ellipsoid. In DIS, the shape of the earth is specified using the World Geodetic System 1984 (WGS84). To define a geodetic coordinate system, the surface of the earth is approximated by a reference ellipsoid which is an ellipsoid of revolution defined by two parameters: the equatorial radius a = 6,378,137 meters (the semimajor axis of the ellipse) and the flattening f = 1/298.257223563. If the polar radius (the semiminor axis of the ellipse) is denoted as b, then b = a * (1-f).

Interconversion From Geodetic to Geocentric

The process of converting between geodetic and geocentric coordinate systems involves transforming a given point in geodetic coordinates with quantities of latitude (ø), longitude (λ), and height (h), into the geocentric cartesian coordinates (X,Y,Z). The approach taken in [1] relies on trigonometry to perform the interconversion. The algorithm in [1] for geodetic to geocentric conversion is accurate and efficient with two minor corrections as described in [3]. The solution presented is an exact solution and the equations are similar to the ones presented in the Military Handbook [2]. The equations are presented below for completeness.

$$X = (R_n + h)\cos\phi\cos\lambda \quad (1)$$

$$Y = (R_n + h)\cos\phi\sin\lambda \quad (2)$$

$$Z = \left(\frac{b^2}{a^2}R_n + h\right)\sin\phi \quad (3)$$

where R_n is defined as the radius of curvature in the prime vertical and is defined by

$$R_n = \frac{a^2}{\sqrt{a^2\cos^2\phi + b^2\sin^2\phi}} \tag{4}$$

Interconversion From Geocentric to Geodetic

The interconversion process involved in converting a geocentric into a geodetic coordinate is more complicated than the previous conversion. The desired solution is to locate the point on the reference ellipsoid that is closest to the original point. The algorithm given in [1] has some errors in the derivation, which are corrected by [3]. However, the results from [1] still do not converge for realistic altitude values in modeling flight simulations. The algorithm given in [2] does converge; however, due to the excessive use of trigonometric functions, it is considerably slower than [3]. The approach taken by [3] does not rely on trigonometry, but instead uses a constrained optimization using Langrange multipliers and the multiplier is then adjusted for convergence. The termination is based on an approximate error measure term.

A different algorithm will be presented here that does not assume the earth to be a sphere in its iteration process. This algorithm will be referred to as algorithm 4. The intention is to compare this approach with the algorithm described in [3], and conclude if the approach is justified. The equation of the reference ellipsoid is as follows,

$$\Phi(x,y,z) = \frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{b^2} - 1 = 0 \tag{5}$$

where a = 6378137m and b = 6356752m, denoting the semimajor and semiminor axis respectively. Let the set of geocentric coordinates (X,Y,Z) be the original point and (x,y,z) be any point in space. Define a vector $\bar{\rho}$,

$$\bar{\rho} = (X - x)\bar{i} + (Y - y)\bar{j} + (Z - z)\bar{k} \tag{6}$$

to be a vector connecting the two given coordinates. Taking the gradient of eq. (5) will result in a vector normal to the tangent at point (x,y,z). This vector is defined as follows:

$$\bar{n} = \nabla\Phi(x,y,z) = \frac{2}{a^2}x\bar{i} + \frac{2}{a^2}y\bar{j} + \frac{2}{b^2}z\bar{k} \tag{7}$$

By defining the relationship between the two vectors, $\bar{\rho}$ and $\bar{n}$, as

$$\bar{\rho} = m\bar{n} \tag{8}$$

where m is a constant, the vector $\bar{\rho}$ is constrained to pass through the point (x,y,z) normal to the ellipsoidal surface. From eqs. (6) and (7), a set of parametric equations of a straight line in space (where m is the parameter) is defined:

$$x = \frac{1}{1 + \left(\frac{2m}{a^2}\right)}X \tag{9}$$

$$y = \frac{1}{1 + \left(\frac{2m}{a^2}\right)}Y \tag{10}$$

$$z = \frac{1}{1 + \left(\frac{2m}{b^2}\right)}Z \tag{11}$$

Substituting eqs. (9), (10), (11) into eq. (5) will constrain (x,y,z) to be on the surface of the ellipsoid. The substitution results are defined by f(m),

$$f(m) = \frac{W^2}{\left[a + \frac{2m}{a}\right]^2} + \frac{Z^2}{\left[b + \frac{2m}{b}\right]^2} - 1 = 0 \tag{12}$$

where $W^2 = X^2 + Y^2$. An iterative numerical approach can be used to determine m, at which time x, y, z, and h can be calculated with the derived equations. Using the Newton-Raphson method for convergence, the value of m can be solved. In order to use Newton-Raphson's algorithm, the derivative of f(m) must be found. This results in the following:

$$f'(m) = \frac{df}{dm} = \frac{-4W^2}{a\left(a + \frac{2m}{a}\right)^3} - \frac{4Z^2}{b\left(b + \frac{2m}{b}\right)^3} \quad (13)$$

The essential algorithm is to first guess a value for m. In the comparison study with algorithm [3], m was set to zero. With this value, substitute into eqs. (9), (10), and (11) to determine the initial set of coordinates. With the first set of x, y, and z points, calculate h with the following equation:

$$h = \sqrt{(X - x)^2 + (Y - y)^2 + (Z - z)^2} \quad (14)$$

Our test for convergence is met if the new calculated h is less than or equal to the previous h by 50 cm, as described in the following equation.

$$|h_i - h_{i-1}| \leq 0.5 \quad \text{where } i = 1, 2, 3, \ldots \quad (15)$$

If eq. (15) is not satisfied, a new value for m must be calculated by following the Newton-Raphson convergence algorithm described below:

$$m_i = m_{i-1} - \frac{f(m_{i-1})}{f'(m_{i-1})} \quad \text{where } i = 1, 2, 3,\ldots \quad (16)$$

With the new m, the convergence process must be reiterated until eq. (15) is satisfied. When convergence is satisfied, x, y, z, and h have been determined by using eqs. (9), (10), (11) and (14). The longitude and latitude is easily computed by using the formulas below:

$$\lambda = \tan^{-1}\left(\frac{y}{x}\right) \quad (17)$$

$$\phi = \tan^{-1}\left(\frac{a^2}{bw}\right)\sqrt{1 - \frac{w^2}{a^2}} \quad (18)$$

where

$$w = \sqrt{x^2 + y^2} \quad (19)$$

The equations derived above are the equations used to perform a geocentric to geodetic coordinate system conversion. The algorithm is best described by a flowchart. This is shown in Figure 1.

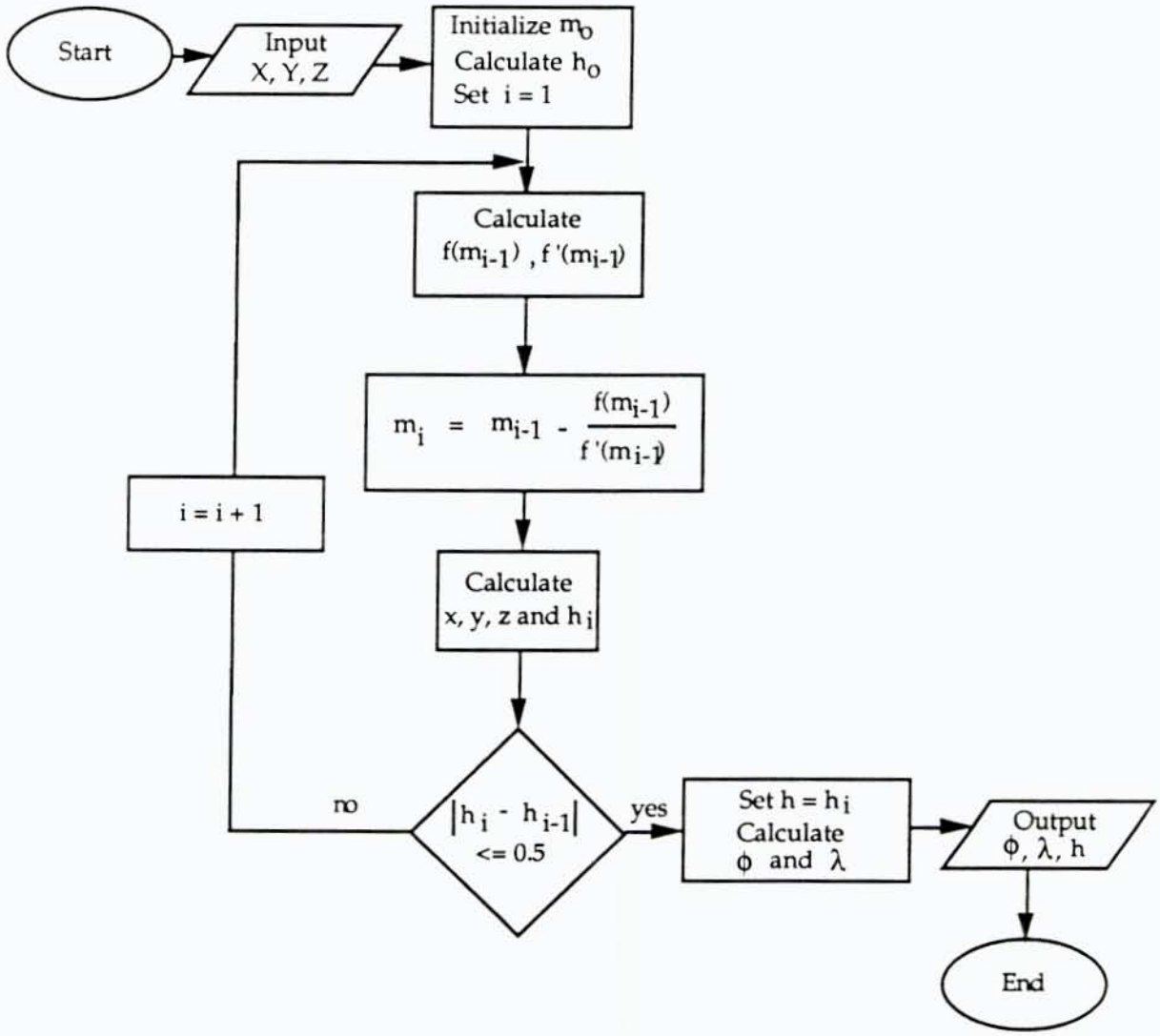

Figure 1: Geocentric to Geodetic Conversion Algorithm.

Comparisons on Different Geocentric to Geodetic Conversion Algorithms

Three algorithms ([1], [2], [3]) were referenced in the previous section, and a new one was presented in detail. This section will attempt to choose one of the four algorithms that best suits a real-time simulation environment. Algorithm [1] has to be eliminated because the iterative step fails to converge for high altitudes, such as those encountered in modeling flight dynamics. In [2], trigonometric functions were used extensively. The computational cost of using a trigonometric function is 12 floats, where one float is defined as the measure of a computational cost in using a single floating point operation. Each iteration in algorithm [2] requires one inverse tangent, one sine and one cosine function. Due to the computational cost involved, algorithm [2] is not recommended for real-time simulation applications, although it has been proven to converge and to be accurate. The algorithm described in [3] and the one described in this paper converge and are both accurate. Both algorithms converge quickly. For example, at a height of 165,000 meters, both converge in two steps. The measured time for each algorithm is illustrated in Table 1 below.

Table 1: Measured Time for the Different Algorithms

Algorithm #	time used for 1 million iterations (sec)	per iteration (sec)
[1]	does not converge	does not converge
[2]	103	1.03E-04
[3]	95	0.95E-04
4	86	0.86E-04

The measured time was taken at one million iterations and the average value was noted. This procedure was performed to overcome any side-effects of running only one iteration. The measurements shown were taken from a Sun SPARCWorkstation. As can be seen from the table above, alorithm [2] takes the longest time to compute one iteration. The algorithm presented in this paper took only 0.86E-04 seconds for one iteration; this is a 19.77% improvement to algorithm [2].

Several runs at different altitudes were taken for the two fastest algorithms, namely, [3] and 4. These values were generated at a latitude of 35N, and a longitude of 40E, and a final tolerance at 50cm. The conversion from geodetic to geocentric coordinates was done using the algorithm in [1] with the corrections noted in [3]. The results are tabulated in the Table 2 for algorithm [3] and Table 3 for algorithm 4.

Table 2: Algorithm [3] Results for Varying Heights.

Given Height (m)	Latitude (deg)	Longitude (deg)	Height (m)
1500	34.999999	40.000000	1499.956301
165000	34.999998	40.000000	164999.882064
3000000	34.999998	40.000000	2999999.811030

Table 3: Algorithm 4 Results for Varying Heights.

Given Height (m)	Latitude (deg)	Longitude (deg)	Height (m)
1500	35.000000	40.000000	1500.000000
165000	35.000000	40.000000	165000.000000
3000000	35.000000	40.000000	3000000.000000

As can be seen from Tables 1, 2 and 3, the results for both algorithms have insignificant differences in either case. Both algorithms are accurate and fast for real-time simulation exercises. If, however, only one algorithm may be chosen, then the algorithm presented in this paper is the most accurate and the fastest.

Interconversion From Topocentric to Geocentric

The geometric relationship between geocentric and topocentric coordinates is shown in Figure 2 below. In order to perform the conversion from topocentric to geocentric, the topocentric coordinate system is rotated about three axes, then translated along its z axis to the origin of the geocentric coordinate system. The algorithm in [1] based their translation using a perfect sphere as their earth model. Due to this assumption, errors were introduced because the translation never passed through the center of the earth. The only time when translation passes through the center of the earth is when latitude is 0, +90 or -90 degrees. When the algorithm in [1] was used, differences as large as 20 km was observed when compared to a translation based on an ellipsoidal earth model.

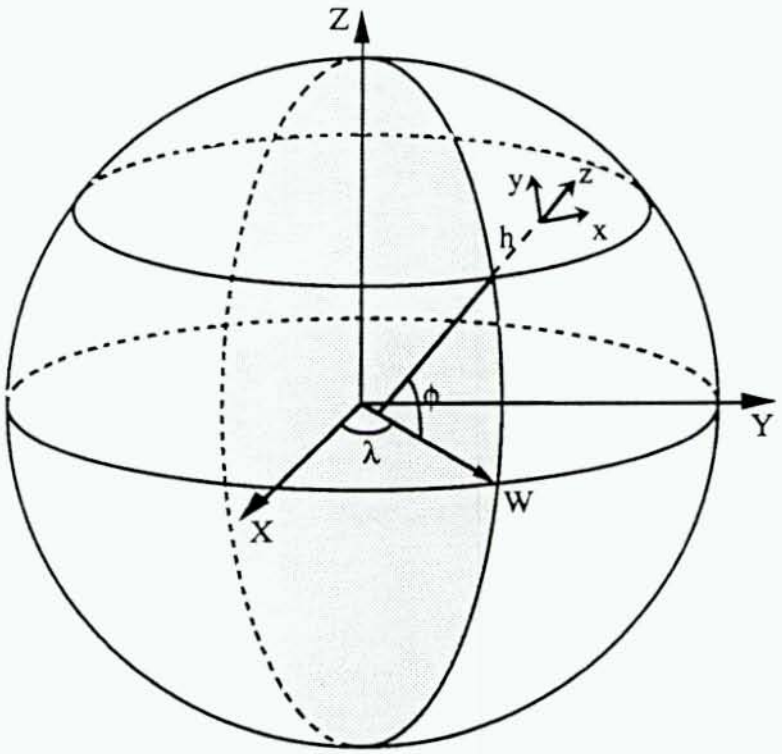

Figure 2: World Coordinate System.

The equation describing this interconversion using an ellipsoidal model can be stated as

$$\begin{Bmatrix} x_g \\ y_g \\ z_g \end{Bmatrix}_{XYZ} = [R]^T \begin{Bmatrix} x_t \\ y_t \\ z_t \end{Bmatrix}_{xyz} + \begin{Bmatrix} x_o \\ y_o \\ z_o \end{Bmatrix}_{XYZ} \tag{20}$$

where the subscripts g, t and o represent geocentric, topocentric and radius of the earth (from the center of the earth to the origin of the topocentric coordinate system) respectively. The XYZ subscript represents a geocentric earth-centered fixed axis, and the xyz subscript represents a topocentric fixed axis. The rotation matrix, R, in terms of latitude (ø), and longitude (λ), is given as

$$\begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \tag{21}$$

The topocentric coordinates are rotated to the geocentric coordinate system before being translated into the center of the earth. The coordinates x_o, y_o, and z_o can be computed by performing the interconversion between geodetic and geocentric coordinate system described in the above section given the latitude, longitude and height. Because the equations derived for the geodetic to geocentric coordinate system interconversion are based on an ellipsoidal model, the radius of curvature of the earth is taken into consideration.

Interconversion From Geocentric to Topocentric

This conversion is similar to the above algorithm. Solving eq. (20) for the topocentric coordinates results in the following:

$$\begin{Bmatrix} x_t \\ y_t \\ z_t \end{Bmatrix}_{xyz} = [R] \left(\begin{Bmatrix} x_g \\ y_g \\ z_g \end{Bmatrix}_{XYZ} - \begin{Bmatrix} x_o \\ y_o \\ z_o \end{Bmatrix}_{XYZ} \right) \tag{22}$$

where the rotation matrix, R, is expressed in eq. (21). The algorithm described in [1] made a translation of the coordinate system based on a perfect sphere and resulted in errors due to the ellipticity of the earth.

Conclusion

In conclusion, this paper has described the various interconversion algorithms for the three coordinate systems, namely, geocentric, geodetic and topocentric. For a geodetic to geocentric conversion, the algorithm presented in [1] had minor errors which have been corrected by [3]. These changes resulted in giving accurate and exact solutions. The results were compared with the algorithm presented in [2] and the results obtained were similar.

In the case of a geocentric to geodetic conversion, an approximation method is required. Algorithm [1] did not converge for realistic heights, therefore, it was eliminated. Algorithm [2] was slow for real-time networking, thus, it was also eliminated. The differences in results obtained from algorithms [3] and 4 were insignificant. However, if one algorithm must be chosen, algorithm 4 should be chosen for the following reasons. First, its per iteration of computation time was faster; and second, it converges to the exact initial values.

In the case of a topocentric to geocentric coordinate conversion, there was an error from paper [1]. The algorithms presented did not take the curvature of the earth into consideration. The equations were derived based on a pure spherical earth. This resulted in approximately a 20 km difference to an ellipsoidal earth at latitude regions of 45 degrees. The topocentric coordinates must be rotated to the geocentric coordinate system before the translation into the center of the earth.

REFERENCES

[1] Burchfiel, Jerry and Stephen Smyth. "Use of Global Coordinates in the SIMNET Protocol", *White Paper ASD-90-10, Second Workshop on Standards for Interoperability of Defense Simulations*, Orlando, FL, January 1990.

[2] "Datums, Projection, Grids and Common Coordinate Systems, Transformation of Department of Defense", *Military Handbook MIL-HDBK-600008*, May 1991.

[3] Wise, Ben. "Geocentric to Geodetic Coordinate Conversions", *BBN Report #7756*, May 1992.