Converting between Earth-Centered, Earth Fixed and Geodetic Coordinates

D. Rose - November, 2014

Abstract

This article describes how to convert from Geodetic coordinates (latitude, longitude and height above ellipsoid) to Earth-Centered, Earth-Fixed coordinates, and back again. The algorithm is closed form (i.e., not iterative), fast, and highly accurate. Example code is provided in Java.

Definition of Geodetic Coordinates:

The Geodetic system uses polar coordinates defined as follows:

Latitude: Angle north and south of the equator. Positive angles are in the northern hemisphere, and negative angles are in the southern hemisphere. The range of angles is -90 degrees (-π/2 radian) to +90 degrees (+π/2 radians). Points on the equator have a latitude of zero.

Longitude: Angle east and west of the Prime Meridian. The Prime Meridian is a north-south line that passes through Greenwich, United Kingdom. Positive longitudes are to the east of the Prime Meridian, and negative angles are to the west. The range of angles is -180 degrees (-π radians) to +180 degrees (+π radians).

Height: Also called altitude or elevation, this represents the height above the Earth ellipsoid, measured in meters. The Earth ellipsoid is a mathematical surface defined by a semi-major axis and a semi-minor axis. The most common values for these two parameters are defined by the World Geodetic Standard 1984 (WGS-84). The WGS-84 ellipsoid is intended to correspond to mean sea level, although in practice the actual mean sea level varies around the world due to ocean currents, Coriolis effects, and local variations in Earth's gravitational field. A Geodetic height of zero therefore roughly corresponds to sea level, with positive values increasing away from the Earth's center. The theoretical range of height values is from the center of the Earth (about -6,371km) to positive infinity. In practice, however, height values deeper that a few kilometers, or higher than geosynchronous orbit (about 36,000km) are seldom used.

Definition of Earth-Centered, Earth-Fixed (ECEF) Coordinates:

ECEF is a right-handed Cartesian coordinate system with the origin at the Earth's center, and that is fixed with respect to the Earth (i.e., rotates along with the Earth). Units are in meters. The three axis are defined as follows:

X:	Passes through the equator at the Prime Meridian (latitude = 0, longitude = 0).
Y:	Passes through the equator 90 degrees east of the Prime Meridian (latitude = 0, longitude = 90 degrees).
Z:	Passes through the North Pole (latitude = 90 degrees, longitude = any value).

Both of these coordinate systems have advantages and disadvantages. The Geodetic system is used for navigation, mapping and GPS applications, and its three components can be intuitively interpreted as representing north/south, east/west, and up/down movements respectively. The ECEF system, on the other hand, is more convenient for calculations involving Euclidean geometry and rotation matrices. It is therefore frequently necessary to convert back and forth between the two systems.

As it happens, the conversion from Geodetic to ECEF is relatively straight forward, involving only a few calculations. The inverse operation, however, is more difficult. In years past, converting from ECEF to Geodetic required an iterative algorithm. Then in 1994 Jijie Zhu from the University of Beijing published the first closed form solution. Since then, several alternative closed form solutions have been developed, each with its strengths and weaknesses. The solution given below was developed by Donald K. Olson, then at the U. S. Naval Air Warfare Center (Olson, D. K. (1996). "Converting earth-Centered, Earth-Fixed Coordinates to Geodetic Coordinates," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. 32, No. 1, January 1996, pp. 473-476).

Sample Code:

A Java port of Olson's ECEF-to-Geodetic algorithm is shown below, along with the standard Geodetic-to-ECEF algorithm. The first two class variables (***a*** and ***e2***) define the WGS-84 ellipsoid, where ***a*** is the semi-major axis (the Earth's radius at the equator) and ***e2*** is the eccentricity squared. The remaining constants can be derived from the first two as documented in the comments, and doing so does not degrade accuracy or increase the computation time.

```
public class Coord {
    private static final double  a = 6378137.0;              //WGS-84 semi-major axis
    private static final double e2 = 6.6943799901377997e-3;  //WGS-84 first eccentricity square
    private static final double a1 = 4.2697672707157535e+4;  //a1 = a*e2
    private static final double a2 = 1.8230912546075455e+9;  //a2 = a1*a1
    private static final double a3 = 1.4291722289812413e+2;  //a3 = a1*e2/2
    private static final double a4 = 4.5577281365188637e+9;  //a4 = 2.5*a2
    private static final double a5 = 4.2840589930055659e+4;  //a5 = a1+a3
    private static final double a6 = 9.9330562000986220e-1;  //a6 = 1-e2
    private static double zp,w2,w,r2,r,s2,c2,s,c,ss;
    private static double g,rg,rf,u,v,m,f,p,x,y,z;
    private static double n,lat,lon,alt;

    //Convert Earth-Centered-Earth-Fixed (ECEF) to lat, Lon, Altitude
    //Input is a three element array containing x, y, z in meters
    //Returned array contains lat and lon in radians, and altitude in meters
     public static double[] ecef_to_geo( double[] ecef ){
        double[] geo = new double[3];   //Results go here (Lat, Lon, Altitude)
        x = ecef[0];
        y = ecef[1];
```

```
        z = ecef[2];
        zp = Math.abs( z );
        w2 = x*x + y*y;
        w = Math.sqrt( w2 );
        r2 = w2 + z*z;
        r = Math.sqrt( r2 );
        geo[1] = Math.atan2( y, x );       //Lon (final)
        s2 = z*z/r2;
        c2 = w2/r2;
        u = a2/r;
        v = a3 - a4/r;
        if( c2 > 0.3 ){
            s = ( zp/r )*( 1.0 + c2*( a1 + u + s2*v )/r );
            geo[0] = Math.asin( s );      //Lat
            ss = s*s;
            c = Math.sqrt( 1.0 - ss );
        }
        else{
            c = ( w/r )*( 1.0 - s2*( a5 - u - c2*v )/r );
            geo[0] = Math.acos( c );      //Lat
            ss = 1.0 - c*c;
            s = Math.sqrt( ss );
        }
        g = 1.0 - e2*ss;
        rg = a/Math.sqrt( g );
        rf = a6*rg;
        u = w - rg*c;
        v = zp - rf*s;
        f = c*u + s*v;
        m = c*v - s*u;
        p = m/( rf/g + f );
        geo[0] = geo[0] + p;      //Lat
        geo[2] = f + m*p/2.0;     //Altitude
        if( z < 0.0 ){
            geo[0] *= -1.0;     //Lat
        }
        return( geo );    //Return Lat, Lon, Altitude in that order
    }

    //Convert Lat, Lon, Altitude to Earth-Centered-Earth-Fixed (ECEF)
    //Input is a three element array containing lat, lon (rads) and alt (m)
    //Returned array contains x, y, z in meters
    public static double[] geo_to_ecef( double[] geo ) {
        double[] ecef = new double[3];  //Results go here (x, y, z)
        lat = geo[0];
        lon = geo[1];
        alt = geo[2];
        n = a/Math.sqrt( 1 - e2*Math.sin( lat )*Math.sin( lat ) );
        ecef[0] = ( n + alt )*Math.cos( lat )*Math.cos( lon );    //ECEF x
        ecef[1] = ( n + alt )*Math.cos( lat )*Math.sin( lon );    //ECEF y
        ecef[2] = ( n*(1 - e2 ) + alt )*Math.sin( lat );          //ECEF z
        return( ecef );     //Return x, y, z in ECEF
    }
}
```

Accuracy:

Olson's algorithm is computationally cheap, and unlike some solutions requires no special treatment at the poles or equator. It is also extremely accurate. In his original paper Olson ran a large number of points, regularly sampled in latitude and longitude, and computed the 3D error in meters for each. At altitudes between -10km and +100km he found an average error of 0.7×10^{-9} meters and a maximum error of 2.7×10^{-9} meters. In a second test, with the range of elevations increased to between -3000km (about half the Earth's radius) and +30,000km, the average error was 2.1×10^{-9} meters, and the maximum error was 1.4×10^{-8} meters.

In our own testing using the code shown above, we generated 100 million points from a random uniform distribution in latitude, longitude and elevation, with elevation restricted to the range -1000km to +100,000km. We generated the points in Geodetic coordinates (lat, lon, alt), converted them to ECEF, then back to Geodetic. We then compared the original Geodetic coordinates with the results of the double conversion.

For the latitude and longitude measurements, the maximum error for any point was 4.44×10^{-16} radians. At the Earth's surface, this corresponds to a maximum position error of 2.8×10^{-9} meters. The maximum altitude error for any point was 4.47×10^{-8} meters. These errors are smaller than the wavelength of visible light, which is accurate indeed.

Comments and error reports may be sent to the following address. We may post comments of general interest. Be sure to identify the page you are commenting on.

EngineeringNotes@DancesWithCode.net.
