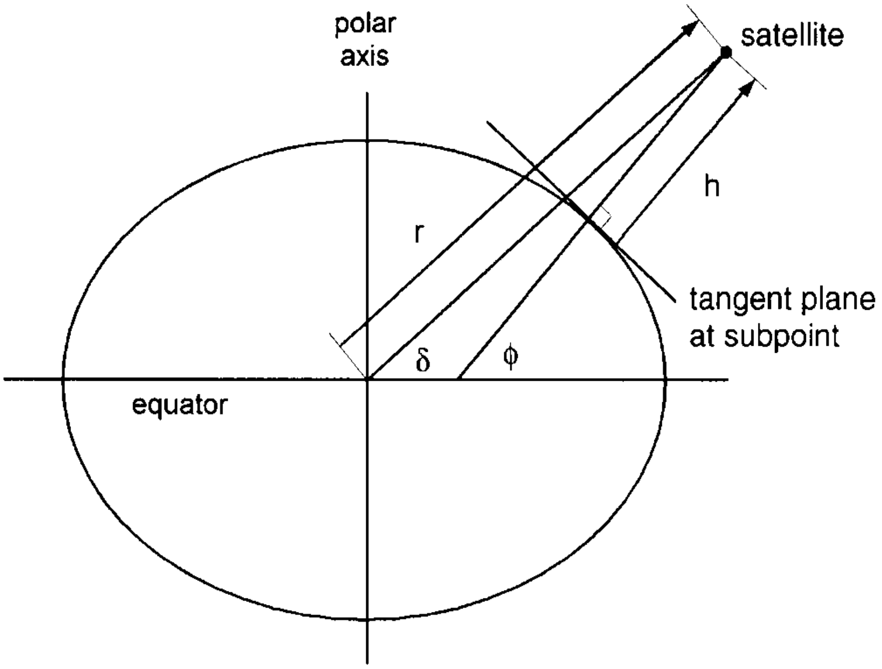
Geodetic and Geocentric Coordinates

This document describes several MATLAB functions that can be used to convert between geocentric and geodetic coordinates. This software suite also includes several scripts that demonstrate how to interact with these functions.

geodet1.m – convert geocentric coordinates to geodetic coordinates - series solution

This MATLAB function uses a series solution to convert geocentric distance and declination to geodetic altitude and latitude. The following diagram illustrates the geometric relationship between geocentric and geodetic coordinates.



In this diagram, δ is the geocentric declination, ϕ is the geodetic latitude, r is the geocentric distance, and h is the geodetic altitude. The exact mathematical relationship between geocentric and geodetic coordinates is given by the following system of two nonlinear equations

$$(c+h)\cos\phi - r\cos\delta = 0$$

$$(s+h)\sin\phi - r\sin\delta = 0$$

where the geodetic constants c and s are given by

$$c = \frac{r_{eq}}{\sqrt{1-\left(2f - f^2\right)\sin^2\phi}} \qquad s = c\left(1-f\right)^2$$

and r_{eq} is the Earth equatorial radius (6378.14 kilometers) and f is the flattening factor for the Earth (1/298.257).

The geodetic latitude is determined using the following expression:

$$\phi = \delta + \left(\frac{\sin 2\delta}{\rho} \right) f + \left[\left(\frac{1}{\rho^2} - \frac{1}{4\rho} \right) \sin 4\delta \right] f^2$$

The geodetic altitude is calculated from

$$\hat{h} = (\hat{r} - 1) + \left\{ \left(\frac{1 - \cos 2\delta}{2} \right) f + \left[\left(\frac{1}{4\rho} - \frac{1}{16} \right) (1 - \cos 4\delta) \right] f^2 \right\}$$

In these equations, ρ is the geocentric distance of the satellite, $\hat{h} = h / r_{eq}$ and $\hat{r} = \rho / r_{eq}$.

The syntax of this MATLAB function is

```
function [alt, lat] = geodet1 (rmag, dec)

% geodetic latitude and altitude

% series solution

% input

%  rmag = geocentric radius (kilometers)
%  dec  = geocentric declination (radians)
%         (+north, -south; -pi/2 <= dec <= +pi/2)

% output

%  alt = geodetic altitude (kilometers)
%  lat = geodetic latitude (radians)
%        (+north, -south; -pi/2 <= lat <= +pi/2)
```

geodet2.m – convert geocentric coordinates to geodetic coordinates – exact solution

This MATLAB function uses the solution of K. M. Borkowski ("Accurate Algorithms to Transform Geocentric to Geodetic Coordinates", *Bullentin Geodesique*, 63 No. 1, 50-56) to convert geocentric distance and declination to geodetic altitude and latitude. The calculation steps for this non-iterative method are as follows:

$$E = \frac{\left[b r_z - \left(a^2 - b^2 \right) \right]}{ar}$$

$$F = \frac{\left[b r_z + \left(a^2 - b^2 \right) \right]}{ar}$$

$$r = \sqrt{r_x^2 + r_y^2}$$

where a is the equatorial radius of the Earth and b is determined from $b = \text{sign}(r_z) \left(a - \frac{a}{f} \right)$.

The calculations continue with

$$P = \frac{4}{3}\left(EF + 1\right)$$

$$Q = 2\left(E^2 - F^2\right)$$

$$D = P^3 + Q^2$$

$$v = \left(D^{1/2} - Q\right)^{1/3}$$

$$G = \frac{1}{2}\left[\sqrt{E^2 + v + E}\right]$$

$$t = \sqrt{\left[G^2 + \frac{F - vG}{2G - E}\right]}$$

The geodetic latitude is determined from the expression

$$\phi = \tan^{-1}\left(a\,\frac{1 - t^2}{2bt}\right)$$

and the geodetic altitude is calculated from

$$h = \left(r - at\right)\cos\phi + \left(r_z - b\right)\sin\phi$$

The syntax of this MATLAB function is

```
function [xlat, alt] = geodet2(decl, rmag)

% geocentric to geodetic coordinates
% exact solution (Borkowski, 1989)

% input

%  decl = geocentric declination (radians)
%  rmag = geocentric distance (kilometers)

% output

%  xlat = geodetic latitude (radians)
%  alt  = geodetic altitude (kilometers)
```

geodet3.m – convert geodetic coordinates to ECF position vector

This function converts geodetic latitude, longitude and altitude to an Earth-centered-fixed (ECF) position vector. The components of this vector are given by

$$\mathbf{r}_{geocentric} = \begin{bmatrix} \left(N + h\right)\cos\phi\cos\lambda_e \\ \left(N + h\right)\cos\phi\sin\lambda_e \\ \left[N\left(1 - e^2\right) + h\right]\sin\phi \end{bmatrix}$$

where

$$\begin{aligned} N &= \frac{r_{eq}}{\sqrt{1-e^2\sin^2\phi}} \\ e^2 &= 2f - f^2 \\ f &= \text{Earth flattening factor} \\ r_{eq} &= \text{Earth equatorial radius} \\ \phi &= \text{geodetic latitude} \\ \lambda_e &= \text{east longitude} \\ h &= \text{geodetic altitude} \end{aligned}$$

The geocentric distance is determined from the components of the geocentric position vector as follows:

$$r = \sqrt{r_x^2 + r_y^2 + r_z^2}$$

The geocentric declination can be computed from the z component and the scalar magnitude of the geocentric position vector with

$$\delta = \sin^{-1}\left(\frac{r_z}{r}\right)$$

The syntax of this MATLAB function is

```
function r = geodet3 (lat, long, alt)

% ecf position vector

% input

%  lat  = geodetic latitude (radians)
%         (+north, -south; -pi/2 <= lat <= +pi/2)
%  long = geodetic longitude (radians)
%  alt  = geodetic altitude (kilometers)

% output

%  r = ecf position vector (kilometers)
```

geodet4.m – convert geodetic coordinates to geocentric coordinates

This MATLAB function converts geodetic latitude and altitude to geocentric position magnitude and geocentric declination.

The equations for this method are as follows:

$$\delta = \phi + \left(\frac{-\sin 2\phi}{\hat{h}+1}\right) f + \left\{ \frac{-\sin 2\phi}{2\left(\hat{h}+1\right)^2} + \left[\frac{1}{4\left(\hat{h}+1\right)^2} + \frac{1}{4\left(\hat{h}+1\right)} \right] \sin 4\phi \right\} f^2$$

and

$$\hat{\rho} = \left(\hat{h}+1\right) + \left(\frac{\cos 2\phi - 1}{2}\right) f + \left\{ \left[\frac{1}{4\left(\hat{h}+1\right)} + \frac{1}{16} \right] \left(1 - \cos 4\phi\right) \right\} f^2$$

where the geocentric distance r and geodetic altitude h have been normalized by $\hat{\rho} = r / r_{eq}$ and $\hat{h} = h / r_{eq}$, respectively, and r_{eq} is the equatorial radius of the Earth.

The syntax of this MATLAB function is

```
function [rmag, dec] = geodet4 (lat, alt)

% geodetic to geocentric coordinates

% input

%  lat  = geodetic latitude (radians)
%         (+north, -south; -pi/2 <= lat <= +pi/2)
%  alt  = geodetic altitude (kilometers)

% output

%  rmag = geocentric position magnitude (kilometers)
%  dec  = geocentric declination (radians)
%         (+north, -south; -pi/2 <= dec <= +pi/2)
```

geodet5.m – convert geocentric coordinates to geodetic coordinates – Sofair solution

This MATLAB function uses the solution of Isaac Sofair, “Improved Method for Calculating Exact Geodetic Latitude and Altitude”, AIAA Journal of Guidance, Control and Dynamics, Vol. 20, No. 4 and Vol. 23, No. 2, to convert geocentric distance and declination to geodetic altitude and latitude.

The syntax of this MATLAB function is

```
function [xalt, xlat] = geodet5(req, rpolar, rsc)

% convert geocentric eci position vector to
% geodetic altitude and latitude

% input

%  req    = equatorial radius (kilometers)
%  rpolar = polar radius (kilometers)
%  rsc    = spacecraft eci position vector (kilometers)

% output

%  xalt = geodetic altitude (kilometers)
%  xlat = geodetic latitude (radians)
%         (+north, -south; -pi/2 <= xlat <= +pi/2)

% reference
% "Improved Method for Calculating Exact Geodetic
% Latitude and Altitude", Isaac Sofair

% AIAA JGCD Vol. 20, No. 4 and Vol. 23, No. 2
```

geodet6.m – convert geocentric coordinates to geodetic coordinates – Jones solution

This MATLAB function uses the solution described in “New solutions for the geodetic coordinate transformation”, G. C. Jones, Journal of Geodesy (2002) 76: 437-446, to convert geocentric distance and declination to geodetic altitude and latitude.

The syntax of this MATLAB function is

```
function [xalt, xlat] = geodet6(req, rpolar, rsc)

% convert geocentric eci position vector to
% geodetic altitude and latitude

% input

%  req    = equatorial radius (kilometers)
%  rpolar = polar radius (kilometers)
%  rsc    = spacecraft eci position vector (kilometers)

% output

%  xalt = geodetic altitude (kilometers)
%  xlat = geodetic latitude (radians)
%         (+north, -south; -pi/2 <= xlat <= +pi/2)

% reference

% "New solutions for the geodetic
% coordinate transformation", G. C. Jones

% Journal of Geodesy (2002) 76: 437-446
```

geodet7.m – convert geocentric coordinates to geodetic coordinates – system of nonlinear equations

This function uses the `fsolve` algorithm of the Optimization Toolbox to solve the system of two nonlinear equations described earlier in this document repeated here for review.

$$(c+h)\cos\phi - r\cos\delta = 0$$

$$(s+h)\sin\phi - r\sin\delta = 0$$

The syntax of this MATLAB function is

```
function [alt, lat] = geodet7(rmag, decl)

% solve system of nonlinear equations that define the
% relationship between geocentric and geodetic coordinates
% input

%  rmag = geocentric radius (kilometers)
%  decl = geocentric declination (radians)

% output

%  altitude = geodetic altitude (kilometers)
%  latitude = geodetic latitude (radians)
```

Here is the MATLAB source code that computes the current values of the nonlinear equations for the `fsolve` numerical method.

```
function f = geodet7_snle(x)

% system of nonlinear equations defining the relationship
% between geocentric and geodetic coordinates

% input

%  x = current argument vector

% output

%  f = nonlinear system evaluated at x

% Orbital Mechanics with MATLAB

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global req flat rmag_wrk decl_wrk

alt = x(1);

lat = x(2);

% calculate "geodetic" constants

c = req / sqrt (1.0 - (2.0 * flat - flat ^ 2) * sin(lat) ^ 2);

s = c * (1.0 - flat) ^ 2;

% system of nonlinear equations

f(1) = (c + alt) * cos(lat) - rmag_wrk * cos(decl_wrk);

f(2) = (s + alt) * sin(lat) - rmag_wrk * sin(decl_wrk);
```

Demonstration script

This software suite contains a MATLAB script named `demo_geodet` that demonstrates how to interact with these useful functions. The following is the output created with this script.

```
geocentric declination      -19.38148629  degrees

geocentric radius          6497.69095120  kilometers
```

```
geodet1 function
================

geodetic latitude           -19.50000000  degrees

geodetic altitude           121.92000000  kilometers

geodet2 function
================

geodetic latitude           -19.50000099  degrees

geodetic altitude           121.92003351  kilometers

geodet5 function
================

geodetic latitude           -19.50000099  degrees

geodetic altitude           121.92003351  kilometers

geodet6 function
================

geodetic latitude           -19.50000099  degrees

geodetic altitude           121.92003351  kilometers

geodet7 function
================

geodetic latitude           -19.50000099  degrees

geodetic altitude           121.92003351  kilometers
```

triaxial.m – geodetic altitude to a triaxial ellipsoid

This MATLAB function calculates the geodetic altitude relative to a triaxial ellipsoidal planet. The algorithm is based on the numerical method described in "Geodetic Altitude to a Triaxial Ellipsoidal Planet", by Charles C. H. Tang, *The Journal of the Astronautical Sciences*, Vol. 36, No. 3, July-September 1988, pp. 279-283.

This function solves for the real root of the following nonlinear equation:

$$f\left(z_p\right)=\left\{1+\frac{c_y}{\left[c_z+\left(b^2-c^2\right)z_p\right]^2}+\frac{c_x}{\left[c_z+\left(a^2-c^2\right)z_p\right]^2}\right\}z_p^2-c^2=0$$

where

$$c_x=\left(acx_s\right)^2 \qquad c_y=\left(bcy_s\right)^2 \qquad c_z=c^2z_s$$

$a,b,c =$ semi-axes $\left(a \ge b > c\right)$

$x_s, y_s, z_s =$ geocentric coordinates of satellite

$z_p =$ z coordinate of subpoint

The bracketing interval used during the root-finding is $z_{p0} - 10 \le z_p \le z_{p0} + 10$ (kilometers) where

$$z_{p0} = \frac{z_s}{\sqrt{\left(x_s / a\right)^2 + \left(y_s / b\right)^2 + \left(z_s / c\right)^2}}$$

The syntax of this MATLAB function is

```
function alt = triaxial(rsc)

% geodetic altitude relative
% to a triaxial ellipsoid

% input

%  rsc = geocentric position vector (km)

% output

%  alt = geodetic altitude (km)
```

The semi-axes in this function are "hardwired" to the values $a = 6378.138$ and $b = 6367$ (kilometers), and the flattening factor is $f = 1/298.257$. These are representative values for the Earth and can be easily changed for other planets or the Moon.

This software suite contains a MATLAB script named `demo_triaxial` that demonstrates how to interact with this function. The following is the output created with this script.

```
program demo_triaxial

altitude relative to a triaxial ellipsoid
-----------------------------------------

altitude =   1519.73117837 kilometers

       sma (km)              eccentricity          inclination (deg)         argper (deg)
 +8.00000000000000e+003  +1.50000000000000e-002  +2.85000000000000e+001  +1.20000000000000e+002

      raan (deg)          true anomaly (deg)          arglat (deg)           period (min)
 +4.50000000000000e+001  +3.00000000000000e+001  +1.50000000000000e+002  +1.18684693004297e+002
```

For this example, the classical orbital elements of the spacecraft are as follows,

semimajor axis = 8000 kilometers

orbital eccentricity = 0.015

orbital inclination = 28.5 degrees

argument of perigee = 120 degrees

RAAN = 45 degrees

true anomaly = 30 degrees