GEO LIBRARY FOR MATLAB

1. Introduction

This library contains a set of functions aimed to deal with different coordinate systems and includes several utilities to compute rhumbs and different kind of lines connecting points. Using it with the KML library is a useful tool in the context of the subject. Below is brief description of each function together with the parameters used.

2. Conversion between coordinate systems

FUNCION NAME	LLA2ECEF
DESCRIPTION	Converts lat/lon/height (LLA) coordinates to Earth-Centered Earth-Fixed (ECEF) coordinates, using WGS84 ellipsoid
MATLAB SINTAXIS	`ecef = lla2ecef(llh);`
INPUT PARAMETERS	`llh  =` 3x1 array containing lat, long, height location (rad, rad, user units)
OUTPUT PARAMETERS	`ecef =` 3x1 array with x,y,z (user units)

FUNCION NAME	ECEF2LLA
DESCRIPTION	Converts cartesian (x,y,z) coordinates of a reference point (ECEF) to lat, lon, height (LLA) coordinates in the WGS 84 system
MATLAB SINTAXIS	`llh  = ecef2lla(ecef);`
INPUT PARAMETERS	`ecef =` 3x1 arrar x,y,z (user units)
OUTPUT PARAMETERS	`llh  =` 3x1 array lat, long, height location (rad, rad, user units)

3. Great circle and rhumb lines computation

3.1. Direct problem solution

FUNCION NAME	ORTO_RECKON
DESCRIPTION	Travels a given distance along a given azimuth starting at a given initial point, using the WGS-84 Earth ellipsoid. Returns the endpoint and final azimut.
MATLAB SINTAXIS	`[lat2,lon2,a21] = orto_reckon(lat1,lon1,s,a12);`

INPUT PARAMETERS	`lat1 =` initial latitude (degrees)
	`lon1 =` initial longitude (degrees)
	`s    = distance (meters)`
	`a12  = initial azimuth (degrees)`

OUTPUT PARAMETERS	`lat2, lon2 =` endpoint (degrees) `a21  =` reverse azimuth (degrees), at final point facing back toward initial point

FUNCION NAME	LOXO_RECKON
DESCRIPTION	Travels a given distance following a rhumb line, i.e, with a constant azimuth. Returns the endpoint and final azimut.
MATLAB SINTAXIS	`[lat2,lon2,a21] = loxo_reckon(lat1,lon1,s,a12)`
INPUT PARAMETERS	`lat1` = inital latitude (degrees) `lon1` = initial longitude (degrees) `s` = distance (meters) `a12` = initial azimuth (degrees)
OUTPUT PARAMETERS	`lat2, lon2` = endpoint (degrees) `a21` = reverse azimuth (degrees), at final point facing back toward initial point

FUNCION NAME	VINCENTY_RECKON
DESCRIPTION	Travels a given distance along a given azimuth starting at a given initial point, using the WGS-84 Earth ellipsoid. Returns the endpoint and final azimut `within` a few millimeters of accuracy using Vincenty's algorithm
MATLAB SINTAXIS	`[lat2,lon2,a21] = vincenty_reckon(lat1,lon1,s,a12)`
INPUT PARAMETERS	`lat1` = inital latitude (degrees) `lon1` = initial longitude (degrees) `s` = distance (meters) `a12` = initial azimuth (degrees)
OUTPUT PARAMETERS	`lat2, lon2` = endpoint (degrees) `a21` = reverse azimuth (degrees), at final point facing back toward initial point

3.2. Inverse problem solution

FUNCION NAME	ORTO_DISTAZI
DESCRIPTION	Computes the distance between two points assuming a perfectly spherical earth and Haversine approximation. Computes also forward azimuth, and backward azimuth.

MATLAB SINTAXIS	`s = orto_distazi(lat1,lon1,lat2,lon2);`
	`[s,a12] = orto_distazi(lat1,lon1,lat2,lon2);`
	`[s,a12,a21] = orto_distazi(lat1,lon1,lat2,lon2);`

INPUT PARAMETERS	`lat1` = GEODETIC latitude of first point (degrees) `lon1` = longitude of first point (degrees) `lat2, lon2` = second point (degrees) (inputs may be scalars, vectors, or matrices)
OUTPUT PARAMETERS	`s` = distance in meters `a12` = azimuth in degrees from first point to second point (forward) `a21` = azimuth in degrees from second point to first point (backward) (Azimuths are in degrees clockwise from north.)

FUNCION NAME	LOXO_DISTAZI
DESCRIPTION	Computes the distance between two points through a loxodromic (rhumb) line in a spher. Computes forward azimuth, and backward azimuth. The azimuth remains constant so forward and backward match.

MATLAB SINTAXIS	s = loxo_distazi(lat1,lon1,lat2,lon2);
	[s,a12] = loxo_distazi(lat1,lon1,lat2,lon2);
	[s,a12,a21] = loxo_distazi(lat1,lon1,lat2,lon2);

INPUT PARAMETERS	lat1 = GEODETIC latitude of first point (degrees) lon1 = longitude of first point (degrees) lat2, lon2 = second point (degrees) (inputs may be scalars, vectors, or matrices)
OUTPUT PARAMETERS	s = distance in meters a12 = azimuth in degrees from first point to second point (forward) a21 = azimuth in degrees from second point to first point (backward) (Azimuths are in degrees clockwise from north.)

FUNCION NAME	VINCENTY_DISTAZI
DESCRIPTION	Compute the distance between two points within a few millimeters of accuracy using the WGS-84 Earth ellipsoid. Compute forward azimuth, and compute backward azimuth, all using a vectorized version of Vincenty's algorithm

MATLAB SINTAXIS	s = vincenty_distazi(lat1,lon1,lat2,lon2);
	[s,a12] = vincenty_distazi(lat1,lon1,lat2,lon2);
	[s,a12,a21] = vincenty_distazi(lat1,lon1,lat2,lon2);

INPUT PARAMETERS	lat1 = GEODETIC latitude of first point (degrees) lon1 = longitude of first point (degrees) lat2, lon2 = second point (degrees) (inputs may be scalars, vectors, or matrices)
OUTPUT PARAMETERS	s = distance in meters a12 = azimuth in degrees from first point to second point (forward) a21 = azimuth in degrees from second point to first point (backward) (Azimuths are in degrees clockwise from north.)

4. Utilities

FUNCION NAME	SEX2DEC
DESCRIPTION	Converts from sex degrees to decimal degrees
MATLAB SINTAXIS	`x = sex2dec(rfs);`
INPUT PARAMETERS	TWO FORMATS ADMITTED FORMAT 1 Format 'N039°29''22.00''' Format 'W028°30''54.00''' FORMAT 2 Format 402959N / ddmmss for latitude Format 0005020W / ddmmss for longitude
OUTPUT PARAMETERS	x: result in decimal degrees

FUNCION NAME	DIST_POINT_TO_LINE
DESCRIPTION	Returns the distance in feets between a point and a line defined by two points
MATLAB SINTAXIS	`dist = dist_point_to_line(Lonp1, Latp1, Lonr1, Latr1, Lonr2, Latr2);`
INPUT PARAMETERS	`lonp1, Latp1`: Longitude and latitude of the point `Lonr1, Latr1, Lonr2, Latr2`: Longitude and latitude of the points r1 and r2 that they define the line
OUTPUT PARAMETERS	`dist =` in feets

FUNCION NAME	DIST_FLAT
DESCRIPTION	Returns the distance in kilometers between two points. It assumes 'flat earth' simplification. Aprox: 1 latitude degree = 110 km 1 longitud degrees= 111 km
MATLAB SINTAXIS	`dist = dist_flat(lon1, lat1, lon2, lat2);`
INPUT PARAMETERS	`lon1, lat1 =` first point longitude and latitude `lon2, lat2 =` second point longitude and latitude
OUTPUT PARAMETERS	`dist =` in km