OCTAVE Mapping Package Documentation

Release 1.1.1

Alfredo Foltran

August 10, 2016

Contents

This package implements simple mapping functions.

Mapping Functions:

CHAPTER 1

Azimuth

Calculates the great circle azimuth from a **point 1** to a **point 2**. The latitude and longitude of these two points can either be given independently or as columns of the matrices **point 1** and **point 2** in the form `[latitude longitude]`.

1.1 Forms

`az = azimuth(lat1, lon1, lat2, lon2)`

`az = azimuth(lat1, lon1, lat2, lon2, units)`

`az = azimuth(pt1, pt2)`

`az = azimuth(pt1, pt2, units)`

Note: The units for the input coordinates and output angles can be *degrees* (the default) or *radians*.

1.2 Examples

Usages:

```
>> azimuth([10, 10], [10, 40])
ans = 87.336
>> azimuth([0, 10], [0, 40])
ans = 90
>> azimuth(pi / 4, 0, pi / 4, -pi / 2, "radians")
ans = 5.3279
```

See also:

Reckon Distance

CHAPTER 2

Reckon

Compute the coordinates of the end-point of a displacement on a sphere. **lat** e **lon** are the coordinates of the starting point, **range** is the covered distance of the displacements along a great circle and **azimuth** is the direction of the displacement relative to the North.

This function can also be used to define a spherical coordinate system with rotated poles.

2.1 Forms

```
[lato, lono] = reckon(lat, lon, range, azimuth)
```

```
[lato, lono] = reckon(lat, lon, range, azimuth, units)
```

Note: The units of all input and output parameters can be either *degrees* (default) or *radians*.

2.2 Examples

Usages:

```
>> [lato, lono] = reckon(0, 10, 30, 90)
lato = 0
lono =  40.000
>> [lato, lono] = reckon(0, 0, pi, pi / 2, "radians")
lato = 0
lono = -3.1416
```

See also:

Azimuth

CHAPTER 3

Distance

Calculates the distance (in degrees) between **point 1** and **point 2**.

3.1 Forms

`dist = distance(pt1, pt2)`

Note: `pt1` and `pt2` are two-column matrices of the form `[latitude longitude]`.

3.2 Examples

Usages:

```
>> distance([37,-76], [37,-9])
ans = 52.309
>> distance([37,-76], [67,-76])
ans = 30.000
```

See also:

Azimuth Vincenty

CHAPTER 4

Departure

Calculates the distance in **degrees** between longitudes `lon1` and `lon2` at latitude defined by `lat`.

4.1 Forms

`dist = departure(lon1, lon2, lat)`

4.2 Examples

Usages:

```
>> departure(0, 10, 0)
ans = 10
>> departure(0, 10, 60)
ans = 5
```

See also:

Distance

CHAPTER 5

Vincenty

Calculates the distance (in kilometers) between **point 1** and **point 2** using the formula devised by Thaddeus Vincenty, with an accurate ellipsoidal model of the earth. The default ellipsoidal model is **WGS-84**, which is the most globally accurate model.

5.1 Forms

```
dist = vincenty(pt1, pt2)
dist = vincenty(pt1, pt2, ellipsoid)
[dist, az] = vincenty(pt1, pt2)
[dist, az] = vincenty(pt1, pt2, ellipsoid)
```

Note: `pt1` and `pt2` are two-column matrices of the form `[latitude longitude]`. The units for the input coordinates angles must be *degrees*. `ellipsoid` defines the reference ellipsoid to use.

Sample values for `ellipsoid` are the following:

- WGS_84 (default) - referenceEllipsoid(7030)
- GRS_80 - referenceEllipsoid(7019)
- Airy - referenceEllipsoid(7001)
- Intl - referenceEllipsoid(7022)
- Clarke - referenceEllipsoid(7012)
- GRS - referenceEllipsoid(7003)

The sample values are the following:

Model	Major (km)	Minor (km)	1 / f
WGS 1984	6378.137	6356.7523142	298.257223563
GRS 1980	6378.137	6356.7523141	298.257222101
G.B. Airy 1830	6377.563396	6356.256909	299.3249646
Internacional 1924	6378.388	6356.911946	297.0
Clarke 1880	6378.249145	6356.51486955	293.465
Australian Nat.	6378.1600	6356.774719	298.25

5.2 Examples

Usages:

```
>> vincenty([37, -76], [37, -9])
ans = 5830.081
>> vincenty([37, -76], [67, -76], referenceEllipsoid(7019))
ans = 3337.843
```

See also:

Distance Reference Ellipsoid

CHAPTER 6

Vincenty (Direct Form)

Compute the coordinates of the end-point of a displacement on a geodesic. The parameters `lat` and `lon` are the coordinates of the starting point, The parameter `range` is the covered distance of the displacements along a specified geodesic and `azi` is the direction of the displacement relative to the North.

6.1 Forms

```
[lato, lono] = vincentyDirect(lat, lon, range, azi)
[lato, lono, azo] = vincentyDirect(lat, lon, range, azi)
[lato, lono] = vincentyDirect(lat, lon, range, azi, dim)
[lato, lono, azo] = vincentyDirect(lat, lon, range, azi, dim)
[lato, lono] = vincentyDirect(lat, lon, range, azi}, dim, ellipsoid)
[lato, lono, azo] = vincentyDirect(lat, lon, range, azi, dim, ellipsoid)
```

Note: The units of all input and output parameters must be 'radians' and/or 'kilometers'. `dim` defines the range dimension to use. `ellipsoid` defines the reference ellipsoid to use.

The possible values for `dim` are 'angle' (default) or 'length'.

Sample values for `ellipsoid` are the following:

- WGS_84 (default) - referenceEllipsoid(7030)
- GRS_80 - referenceEllipsoid(7019)
- Airy - referenceEllipsoid(7001)
- Intl - referenceEllipsoid(7022)
- Clarke - referenceEllipsoid(7012)
- GRS - referenceEllipsoid(7003)

The sample values are the following:

Model	Major (km)	Minor (km)	1 / f
WGS 1984	6378.137	6356.7523142	298.257223563
GRS 1980	6378.137	6356.7523141	298.257222101
G.B. Airy 1830	6377.563396	6356.256909	299.3249646
Internacional 1924	6378.388	6356.911946	297.0
Clarke 1880	6378.249145	6356.51486955	293.465
Australian Nat.	6378.1600	6356.774719	298.25

6.2 Examples

Usages:

```
>> [lat, lon] = vincentyDirect(0, 0, pi, pi / 2)
lat = 0
lon = 3.1311
>> [lat, lon, az] = vincentyDirect(0, 0, pi, pi / 2)
lat = 0
lon = 3.1311
az = 1.5708
```

See also:

Vincenty Reference Ellipsoid

CHAPTER 7

Great-Circle to Small-Circle

Converts a great circle to small circle notation.

7.1 Forms

```
[lato, lono, range] = gc2sc(lat, lon, azi)
[lato, lono] = gc2sc(lat, lon, azi, units)
mat = gc2sc(lat, lon, azi)
mat = gc2sc(lat, lon, azi, units)
```

CHAPTER 8

Degree to Kilometer

Convert a distance along a great circle of the Earth from degrees to kilometer. Great-circle distance uses a spherical model of the earth, using the average great-circle radius of 6372.795 kilometers, resulting in an error of up to about 0.5%.

8.1 Forms

`x = deg2km(a)`

See also:

Kilometer to Degree

CHAPTER 9

Kilometer to Degree

Convert a distance along a great circle of the Earth from kilometer to degrees. A radius Great-circle distance uses a spherical model of the earth, using the average great-circle radius of 6372.795 kilometers, resulting in an error of up to about 0.5%.

9.1 Forms

`a = km2deg(x)`

See also:

Degree to Kilometer

CHAPTER 10

Degree to Radians

Converts angles input in degrees to the equivalent in radians.

10.1 Forms

```
anglout = deg2rad(anglin)
```

See also:

Radians to Degree

CHAPTER 11

Radians to Degree

Converts angles input in radians to the equivalent in degrees.

11.1 Forms

```
anglout = rad2deg(anglin)
```

See also:

Degree to Radians

CHAPTER 12

Geocode

Geolocation service from Google Maps API. Define a geocode method for resolving a location from a string.

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map.

Reverse geocoding is the process of converting geographic coordinates into a human-readable address.

12.1 Forms

[lato, lono] = geocode(addr)

[lato, lono, xml] = geocode(addr)

12.2 Examples

Usages:

```
>> [lat, lon] = geocode('new york,ny')
lat = 40.714
lon = -74.006
>> [lat, lon] = geocode('brasilia,df')
lat = -15.827
lon = -47.922
```

See also:

Reverse

CHAPTER 13

Reverse

Geolocation service from Google Maps API. Define a geocode method for resolving a address from a location.

The term geocoding generally refers to translating a human-readable address into a location on a map. The process of doing the converse, translating a location on the map into a human-readable address, is known as reverse geocoding.

13.1 Forms

addr = reverse([lat lon])

[addr, xml] = reverse([lat lon])

13.2 Examples

Usages:

```
>> reverse([40.714 -74.006])
ans = '58-68 Chambers Street, Nova York, NY 10007, EUA'
>> reverse([-15.827 -47.922])
ans = 'Asa Sul, Brasília, DF, 70383-070, Brasil'
```

See also:

Geocode

CHAPTER 14

nPI to PI

This brings the `angin` into the $[-pipi]$ interval.

14.1 Forms

`angout = npi2pi(angin)`

See also:

ZERO to 2PI

CHAPTER 15

ZERO to 2PI

This brings the `angin` into the $[02 * pi[$ interval.

15.1 Forms

`angout = zero22pi(angin)`

See also:

nPI to PI

CHAPTER 16

Reference Ellipsoid

This function returns a reference ellipsoid object corresponding to the specified `code` (numerical EPSG). The values of the `SemimajorAxis` and `SemiminorAxis` properties are in kilometers. The reference ellipsoid has five properties: `Code`, `Name`, `SemimajorAxis`, `SemiminorAxis` and `Flattening`.

The form code can receive a valid EPSG code. 46 codes are currently implemented between 7001 and 7053 (except for 7017, 7023, 7026 and 7037-7040).

The valid values for name form are as follows: sphere, unitsphere, earth, moon, mercury, venus, mars, jupiter, saturn, uranus, neptune and pluto.

16.1 Forms

`ref = referenceEllipsoid(code)`

`ref = referenceEllipsoid(name)`

CHAPTER 17

Indices and tables

- genindex
- search