

NAME

geod – Geodesic computations

SYNOPSIS

geod *+ellps=<ellipse>* [**-a****f****l****p****t****w****W** [args]] [*+args*] file[s]

invgeod *+ellps=<ellipse>* [**-a****f****l****p****t****w****W** [args]] [*+args*] file[s]

DESCRIPTION

geod (direct) and **invgeod** (inverse) perform geodesic (Great Circle) computations for determining latitude, longitude and back azimuth of a terminus point given a initial point latitude, longitude, azimuth and distance (direct) or the forward and back azimuths and distance between an initial and terminus point latitudes and longitudes (inverse). The results are accurate to round off for $|f| < 1/50$, where f is flattening.

invgeod may not be available on all platforms; in this case use *geod -I* instead.

The following command-line options can appear in any order:

- I** Specifies that the inverse geodesic computation is to be performed. May be used with execution of **geod** as an alternative to **invgeod** execution.
- a** Latitude and longitudes of the initial and terminal points, forward and back azimuths and distance are output.
- t<a>** Where *a* specifies a character employed as the first character to denote a control line to be passed through without processing.
- le** Gives a listing of all the ellipsoids that may be selected with the *+ellps=* option.
- lu** Gives a listing of all the units that may be selected with the *+units=* option.
- f <format>**
Where *format* is a printf format string to control the output form of the geographic coordinate values. The default mode is DMS for geographic coordinates and "**%.3f**" for distance.
- F <format>**
Where *format* is a printf format string to control the output form of the distance value (**-F**). The default mode is DMS for geographic coordinates and "**%.3f**" for distance.
- w<n>** Where *n* is the number of significant fractional digits to employ for seconds output (when the option is not specified, **-w3** is assumed).
- W<n>**
Where *n* is the number of significant fractional digits to employ for seconds output. When **-W** is employed the fields will be constant width with leading zeroes.
- p** This option causes the azimuthal values to be output as unsigned DMS numbers between 0 and 360 degrees. Also note *-f*.

The *+args* command-line options are associated with geodetic parameters for specifying the ellipsoidal or sphere to use. controls. The options are processed in left to right order from the command line. Reentry of an option is ignored with the first occurrence assumed to be the desired value.

See the PROJ documentation for a full list of these parameters and controls.

One or more files (processed in left to right order) specify the source of data to be transformed. A **-** will specify the location of processing standard input. If no files are specified, the input is assumed to be from stdin.

For direct determinations input data must be in latitude, longitude, azimuth and distance order and output will be latitude, longitude and back azimuth of the terminus point. Latitude, longitude of the initial and terminus point are input for the inverse mode and respective forward and back azimuth from the initial and

terminus points are output along with the distance between the points.

Input geographic coordinates (latitude and longitude) and azimuthal data must be in decimal degrees or DMS format and input distance data must be in units consistent with the ellipsoid major axis or sphere radius units. The latitude must lie in the range $[-90^{\circ}, 90^{\circ}]$. Output geographic coordinates will be in DMS (if the `-f` switch is not employed) to 0.001" with trailing, zero-valued minute-second fields deleted. Output distance data will be in the same units as the ellipsoid or sphere radius.

The Earth's ellipsoidal figure may be selected in the same manner as program **proj** by using `+ellps=`, `+a=`, `+es=`, etc.

geod may also be used to determine intermediate points along either a geodesic line between two points or along an arc of specified distance from a geographic point. In both cases an initial point must be specified with `+lat_1=lat` and `+lon_1=lon` parameters and either a terminus point `+lat_2=lat` and `+lon_2=lon` or a distance and azimuth from the initial point with `+S=distance` and `+A=azimuth` must be specified.

If points along a geodesic are to be determined then either `+n_S=integer` specifying the number of intermediate points and/or `+del_S=distance` specifying the incremental distance between points must be specified.

To determine points along an arc equidistant from the initial point both `+del_A=angle` and `+n_A=integer` must be specified which determine the respective angular increments and number of points to be determined.

EXAMPLES

The following script determines the geodesic azimuths and distance in U.S. statute miles from Boston, MA, to Portland, OR:

```
geod +ellps=clrk66 <<EOF -I +units=us-mi
42d15'N 71d07'W 45d31'N 123d41'W
EOF
```

which gives the results:

```
-66d31'50.141" 75d39'13.083" 2587.504
```

where the first two values are the azimuth from Boston to Portland, the back azimuth from Portland to Boston followed by the distance.

An example of forward geodesic use is to use the Boston location and determine Portland's location by azimuth and distance:

```
geod +ellps=clrk66 <<EOF +units=us-mi
42d15'N 71d07'W -66d31'50.141" 2587.504
EOF
```

which gives:

```
45d31'0.003"N 123d40'59.985"W 75d39'13.094"
```

NOTE:

Lack of precision in the distance value compromises the precision of the Portland location.

FURTHER READING

1. *GeographicLib*.

2. C. F. F. Karney, *Algorithms for Geodesics*, J. Geodesy **87**(1), 43–55 (2013); *addenda*.
3. *A geodesic bibliography*.

SEE ALSO

proj(1), **cs2cs(1)**, **cct(1)**, **geod(1)**, **gie(1)**

BUGS

A list of know bugs can be found at <https://github.com/OSGeo/proj.4/issues> where new bug reports can be submitted to.

HOME PAGE

<https://proj4.org/>

AUTHOR

Charles Karney

COPYRIGHT

1983-2018