Flood Tool

Oct 20, 2019

CONTENTS

This package implements a flood risk prediction tool.

CHAPTER

ONE

GEODETIC TRANSFORMATIONS

For historical reasons, multiple coordinate systems exist in British mapping. The Ordnance Survey has been mapping the British Isles since the 18th Century and the last major retriangulation from 1936-1962 produced the Ordance Survey National Grid (or **OSGB36**), which defined latitude and longitude across the island of Great Britain[1]. For convenience, a standard Transverse Mercator projection[2] was also defined, producing a notionally flat gridded surface, with gradations called eastings and westings. The scale for these gradations was identified with metres.

The OSGB36 datum is based on the Airy Ellipsoid of 1830, which defines semimajor axes for its model of the earth, a and b, a scaling factor F_0 and ellipsoid height, H.

$$\begin{aligned} a &= 6377563.396, \\ b &= 6356256.910, \\ F_0 &= 0.9996012717, \\ H &= 24.7. \end{aligned}$$

The point of origin for the transverse Mercator projection is defined in the Ordnance Survey longitude-latitude and easting-northing coordinates as

$$\begin{aligned} \phi_0^{OS} &= 49^\circ \text{ north}, \\ \lambda_0^{OS} &= 2^\circ \text{ west}, \\ E_0^{OS} &= 400000m, \\ N_0^{OS} &= -100000m. \end{aligned}$$

More recently, the world has gravitated towards the use of Satellite based GPS equipment, which uses the (globally more appropriate) World Geodetic System 1984 (or **WGS84**). This datum uses a different ellipsoid, which offers a better fit for a global coordinate system. Its key properties are:

$$\begin{aligned} a_{WGS} &= 6378137,, \\ b_{WGS} &= 6356752.314, \\ F_0 &= 0.9996. \end{aligned}$$

For a given point on the WGS84 ellipsoid, an approximate mapping to the OSGB36 datum can be found using a Helmert transformation[3],

$$\mathbf{x}^{OS} = \mathbf{t} + \mathbf{M}\mathbf{x}^{WGS}.$$

[1] A guide to coordinate systems in Great Britain, Ordnance Survey

[2] Map projections - A Working Manual, John P. Snyder, https://doi.org/10.3133/pp1395

[3] Computing Helmert transformations, G Watson, http://www.maths.dundee.ac.uk/gawatson/helmertrev.pdf

Here $\mathbf{x}$ denotes a coordinate in Cartesian space (i.e in 3D) as given by the (invertible) transformation

$$\begin{aligned}
\nu &= \frac{aF_0}{\sqrt{1-e^2\sin^2(\phi^{OS})}}\\
x &= (\nu+H)\sin(\lambda)\cos(\phi)\\
y &= (\nu+H)\cos(\lambda)\cos(\phi)\\
z &= ((1-e^2)\nu+H)\sin(\phi)
\end{aligned}$$

and the transformation parameters are

$$\begin{aligned}
\mathbf{t} &= \begin{pmatrix} -446.448\\ 125.157\\ -542.060 \end{pmatrix},\\
\mathbf{M} &= \begin{bmatrix} 1+s & -r_3 & r_2\\ r_3 & 1+s & -r_1\\ -r_2 & r_1 & 1+s \end{bmatrix},\\
s &= 20.4894\times 10^{-6},\\
\mathbf{r} &= [0.1502'', 0.2470'', 0.8421''].
\end{aligned}$$

Given a latitude, ϕ^{OS} and longitude, λ^{OS} in the OSGB36 datum, easting and northing coordinates, E^{OS} & N^{OS} can then be calculated using the following formulae:

$$\begin{aligned}
\rho &= \frac{aF_0(1-e^2)}{\left(1-e^2\sin^2(\phi^{OS})\right)^{\frac{3}{2}}}\\
\eta &= \sqrt{\frac{\nu}{\rho}-1}\\
M &= bF_0\left[\left(1+n+\frac{5}{4}n^2+\frac{5}{4}n^3\right)(\phi^{OS}-\phi_0^{OS})\right.\\
&\quad -\left(3n+3n^2+\frac{21}{8}n^3\right)\sin(\phi-\phi_0)\cos(\phi^{OS}+\phi_0^{OS})\\
&\quad +\left(\frac{15}{8}n^2+\frac{15}{8}n^3\right)\sin(2(\phi^{OS}-\phi_0^{OS}))\cos(2(\phi^{OS}+\phi_0^{OS}))\\
&\quad \left.-\frac{35}{24}n^3\sin(3(\phi-\phi_0))\cos(3(\phi^{OS}+\phi_0^{OS}))\right]\\
I &= M+N_0^{OS}\\
II &= \frac{\nu}{2}\sin(\phi^{OS})\cos(\phi^{OS})\\
III &= \frac{\nu}{24}\sin(\phi^{OS})cos^3(\phi^{OS})(5-\tan^2(phi^{OS})+9\eta^2)\\
IIIA &= \frac{\nu}{720}\sin(\phi^{OS})cos^5(\phi^{OS})(61-58\tan^2(\phi^{OS})+\tan^4(\phi^{OS}))\\
IV &= \nu\cos(\phi^{OS})\\
V &= \frac{\nu}{6}\cos^3(\phi^{OS})\left(\frac{\nu}{\rho}-\tan^2(\phi^{OS})\right)\\
VI &= \frac{\nu}{120}\cos^5(\phi^{OS})(5-18\tan^2(\phi^{OS})+\tan^4(\phi^{OS})\\
&\quad +14\eta^2-58\tan^2(\phi^{OS})\eta^2)\\
E^{OS} &= E_0^{OS}+IV(\lambda^{OS}-\lambda_0^{OS})+V(\lambda-\lambda_0^{OS})^3+VI(\lambda^{OS}-\lambda_0^{OS})^5\\
N^{OS} &= I+II(\lambda^{OS}-\lambda_0^{OS})^2+III(\lambda-\lambda_0^{OS})^4+IIIA(\lambda^{OS}-\lambda_0^{OS})^6
\end{aligned}$$

CHAPTER

TWO

FUNCTION APIS

class flood_tool.**Tool**(*postcode_file=None*, *risk_file=None*, *values_file=None*)

Class to interact with a postcode database file.

Reads postcode and flood risk files and provides a postcode locator service.

Parameters

- **postcode_file** (*str, optional*) – Filename of a .csv file containing geographic location data for postcodes.
- **risk_file** (*str, optional*) – Filename of a .csv file containing flood risk data.
- **postcode_file** – Filename of a .csv file containing property value data for postcodes.

get_annual_flood_risk(*postcodes*, *probability_bands*)

Get an array of estimated annual flood risk in pounds sterling per year of a flood event from a sequence of postcodes and flood probabilities.

Parameters

- **postcodes** (*sequence of strs*) – Ordered collection of postcodes
- **probability_bands** (*sequence of strs*) – Ordered collection of flood probabilities

Returns array of floats for the annual flood risk in pounds sterling for the input postcodes. Invalid postcodes return *numpy.nan*.

Return type numpy.ndarray

get_easting_northing_flood_probability_band(*easting*, *northing*)

Get an array of flood risk probabilities from arrays of eastings and northings.

Flood risk data is extracted from the Tool flood risk file. Locations not in a risk band circle return *Zero*, otherwise returns the name of the highest band it sits in.

Parameters

- **easting** (*numpy.ndarray of floats*) – OS Eastings of locations of interest
- **northing** (*numpy.ndarray of floats*) – Ordered sequence of postcodes

Returns numpy array of flood probability bands corresponding to input locations.

Return type numpy.ndarray of strs

get_flood_cost(*postcodes*, *probability_bands*)

Get an array of estimated cost of a flood event from a sequence of postcodes. :param postcodes: Ordered collection of postcodes :type postcodes: sequence of strs :param probability_bands: Ordered collection of flood probability bands :type probability_bands: sequence of strs

Returns array of floats for the pound sterling cost for the input postcodes. Invalid postcodes return *numpy.nan*.

Return type numpy.ndarray of floats

`get_lat_long`(*postcodes*)

Get an array of WGS84 (latitude, longitude) pairs from a list of postcodes.

Parameters **`postcodes`** (`sequence of strs`) – Ordered sequence of N postcode strings

Returns Array of Nx2 (latitude, longitdue) pairs for the input postcodes. Invalid postcodes return [*numpy.nan*, *numpy.nan*].

Return type ndarray

`get_sorted_annual_flood_risk`(*postcodes*)

Get a sorted pandas DataFrame of flood risks.

Parameters **`postcodes`** (`sequence of strs`) – Ordered sequence of postcodes

Returns Dataframe of flood risks indexed by (normalized) postcode and ordered by risk, then by lexagraphic (dictionary) order on the postcode. The index is named *Postcode* and the data column *Flood Risk*. Invalid postcodes and duplicates are removed.

Return type pandas.DataFrame

`get_sorted_flood_probability`(*postcodes*)

Get an array of flood risk probabilities from a sequence of postcodes.

Probability is ordered High>Medium>Low>Very low>Zero. Flood risk data is extracted from the *Tool* flood risk file.

Parameters **`postcodes`** (`sequence of strs`) – Ordered sequence of postcodes

Returns Dataframe of flood probabilities indexed by postcode and ordered from *High* to *Zero*, then by lexagraphic (dictionary) order on postcode. The index is named *Postcode*, the data column is named *Probability Band*. Invalid postcodes and duplicates are removed.

Return type pandas.DataFrame

`flood_tool.`**`WGS84toOSGB36`**(*latitude*, *longitude*, *radians=False*)

Wrapper to transform (latitude, longitude) pairs from GPS to OS datum.

`flood_tool.`**`get_easting_northing_from_lat_long`**(*latitude*, *longitude*, *radians=False*)

Convert GPS (latitude, longitude) to OS (easting, northing).

Parameters

- **`latitude`** (`sequence of floats`) – Latitudes to convert.
- **`longitude`** (`sequence of floats`) – Lonitudes to convert.
- **`radians`** (`bool, optional`) – Set to *True* if input is in radians. Otherwise degrees are assumed

Returns

- **easting** (*ndarray of floats*) – OS Eastings of input
- **northing** (*ndarray of floats*) – OS Northings of input

References

A guide to coordinate systems in Great Britain (https://webarchive.nationalarchives.gov.uk/20081023180830/http://www.ordnancesurvey.co.uk/oswebsite/gps/information/coordinatesystemsinfo/guidecontents/index.html)

References

PYTHON MODULE INDEX

INDEX