PyGTide A Python module and wrapper for ETERNA PREDICT to compute gravitational tides on Earth

Gabriel C. Rau[a,b,*]

[a] *Institute of Applied Geosciences, Karlsruhe Institute of Technology, Karlsruhe, Germany*

[b] *Connected Waters Initiative Research Centre, School of Civil and Environmental Engineering, UNSW Sydney, Australia*

Abstract

This user guide gives a brief overview of **PyGTide**, a Python module to compute gravitational tides on Earth based on ETERNA PREDICT by *Wenzel* [1996]. **PyGTide** is freely available on GitHub and released under the Mozilla Public License 2.0. Installation and use are outlined and the wrapper class is explained in detail. Further, results calculated using **PyGTide** and *TSoft* [*Van Camp and Vauterin*, 2005] are briefly compared using an example.

1. Background

Earth tides are variations in gravity on Earth induced by the relative movement of celestial bodies such as the sun or moon [*Agnew*, 2010] and cause the strongest variation in gravity [*Xu et al.*, 2004]. Gravity on Earth has an average value of 9.8 m/s^2 and the unit for gravity is Gal (Galileo) where 1 Gal equals 1 cm/s^2 or 0.01 m/s^2. Gravity can be measured with a precision of 0.1 nm/s^2 (or 10^{-11} g) [*Van Camp et al.*, 2017].

Measurements of parameters in the geosciences or earth sciences contain influences of Earth tides. For example, Earth tide signatures are found in measurements of atmospheric pressure [*Chapman*, 1951] and groundwater level measurements [e.g. *Meinzer*, 1939]. Such signatures can be used to understand and quantify Earth processes and properties [e.g. *Bredehoeft*, 1967; *Acworth et al.*, 2016]. This requires knowledge of the gravity variations caused by Earth tides. While Earth tides can be measured through gravity, it is sufficient and more convenient to calculate the gravity variations. In fact, Earth tides can be predicted with such accuracy that calculations are often used to reveal Earth process revealed as residuals when subtracted from gravity measurements [e.g. *Longuevergne et al.*, 2009].

Earth tides can be calculated using scientific software such as *TSoft* [*Van Camp and Vauterin*, 2005] or *ETERNA* [*Wenzel*, 1996]. The latter has been incorporated as an executable in *PyGrav*, a Python-based package for handling gravity measurements [*Hector and Hinderer*, 2016]. However, there is no true Python package available to calculate the gravitational variations caused by Earth tides. **PyGTide** aims to address this shortcoming by providing an easy to use module that can be incorporated into scientific computations with Python.

*Department of Engineering Geology, Institute of Applied Geosciences
Email address: `gabriel@hydrogeo.science` (Gabriel C. Rau)

2. A brief overview of ETERNA PREDICT

ETERNA was originally written in *Fortran 77* by Prof. Dr.-Ing. Hans-Georg Wenzel [*Wenzel*, 1996] from the Black Forest Observatory in Schiltach which was part of the Universität Karlsruhe, predecessor of the Karlsruhe Institute of Technology (KIT) in Germany. *ETERNA PREDICT* was released as version 3.3 in October 1996 [*Wenzel*, 1996]. *ETERNA* 3.3 allows the recording, preprocessing and analysis of earth tide observations under operating system MS-DOS on an IBM-AT compatible personal computer 80386/387 upwards. *ETERNA* contains a subroutine PREDICT which remains the most sophisticated and accurate mathematical routine for computing synthetic model tides. *Kudryavtsev* [2004] updated *ETERNA* to include the latest tidal catalogue in version 3.4.

Predictions use the provided geo-location (latitude, longitude, height) and start time as well as sampling rate to calculate a time series of the tidal potential. This relies on sophisticated mathematical relationships which describe the movement of celestial bodies relative to the Earth's rotation. *ETERNA PREDICT* contains generic calculation routines which rely on external text files containing the many constants that inform the calculation. These files are located in a subdirectory named *commdat* and are briefly described in the following subsections.

2.1. Time scale conversion

The movement of celestial bodies requires the connection of two different time scales: (1) the time that is relevant to applications on Earth as measured by clocks referred to as Coordinated Universal Time (UTC), (2) the absolute time of the solar system referred to as terrestrial time (TT). A comprehensive overview of the different time scales can be found here by Steve Allen from the University of California Observatories.

A text based overview of the different time references is as follows:

```
                                                        ET  1960-1983
                                                       TDT  1984-2000
 UTC  1972-   GPS  1980-    TAI  1958-                  TT  2001-
----+---------+-------------+-------------------------+-----
    |         |             |                         |
    |<-- TAI-UTC  (leaps)  -->|<-----    TT-TAI    ----->|
    |         |             |      32.184s  fixed     |
    |<GPS-UTC>|<-  TAI-GPS  ->|                       |
    |         |   19s  fixed  |                       |
    |                                                 |
    |<----- DDT  =  TT-UTC  =  delta-T  +  delta-UT  ------->|
    |                                                 |
   <->|  delta-UT  =  UT1-UTC  (maximum  +/-0.9  sec)       |
-------+-------------------------------------------------+-----
       |<------------  delta-T  =  TT-UT1  -------------->|
      UT1  (UT)                                      TT/TDT/ET
```

ETERNA PREDICT contains a routine that converts between UTC and TT so that the gravity values can be calculated for times that are relevant to Earth's time reference. This routine relies on an external text file called *etddt.dat* which contains the difference between both time scales. For continued accuracy of future predictions, this file must be updated to account for the accumulated leap seconds. This is available as data product from the *International Earth*

Rotation Service (IERS). A convenient method for automatically updating this file is described in Section 4.2.6.

2.2. Pole wobble and length of day (LOD) corrections

Pole tides are caused by a variation in the geocentric position of the Earth's rotation axis. As a result, the rotation axis can shift within a square of $20\,m$ [*ESA*, 2013]. These shifts cause the Earth to rotate unevenly with periods of 12 months (annual wobble) and 14 months (Chandler wobble) and can affect the gravity tides by up to $13\,\mu Gal$ which must be considered [*Wahr*, 1985].

The Earth's rotation rate, also called length of day (LOD), changes over time. This is due to a multitude of different processes acting at variable frequencies [*Eubanks*, 1993] such as a complex interplay between angular momentum and mass movement (i.e., tidal ocean water redistribution) [*Ray et al.*, 1994]. The origin of some of these influences have not yet been properly attributed [*Shen and Peng*, 2016]. However, this length of day affects gravity tides and must therefore be accounted for in calculations.

ETERNA PREDICT considers the pole and LOD corrections in its calculations, but this relies on knowledge of the pole coordinates as well as the LOD values. Daily values for both are contained in an external text file called *etpolut1.dat*. This file must be updated regularly in order to enable continuously accurate Earth tide predictions. Fortunately, the pole coordinates and LOD values are measured and daily updates are available from the *International Earth Rotation Service (IERS)* starting in 1962. Furthermore, the *United States Naval Observatory (USNO)* offers daily forecasts for up to one year ahead in time. A convenient method for automatic updating of **PyGTide**'s pole coordinate and LOD database is described in Section 4.2.7.

2.3. Tidal catalogues

Calculating the tide-generating potential relies on many astronomical constants. These are archived in so called tidal catalogues which have undergone an evolution over time in order to increase the accuracy of the prediction. Table 1 contains the details of the tidal catalogues that are available for use with ETERNA.

3. Installation and use of PyGTide

3.1. How to install

PyGTide is a Python class that relies on a DLL that was compiled from *Fortran* code using *F2PY* in the Windows 10 operating system. Note that **PyGTide** is currently only available and has only been tested for Windows 7 and 10. The following steps are required to make **PyGTide** work:

1. Download and install *Anaconda 5.2+* (for Python 3.6) for Windows 7/10 (64bit).
2. Open the Anaconda Navigator and ensure that the packages libpython (as a minimum v2.1) and mingw (as a minimum v4.7) are installed. The following standard libraries are also required but usually installed by default: numpy, pandas and datetime.
3. Download **PyGTide** and extract files in a local directory.

PyGTide, doi:10.5281/zenodo.1346260

Authors of catalog	**Name**	**Waves***	**RMSE [nGal]**	tidalpoten#
			(time domain)	
Doodson [1921]	-	378	102[1]	1
Cartwright and Edden [1973]	-	505	37.4[1]	2
Büllesfeld [1985]	-	656	24[1]	3
Tamura [1987]	T87	1,200	6.7[1]	4
Xi and Hou [1987]	XI1989	2,934	7.9[1]	5
Tamura [1993]	T93	2,114	3[1]	
Roosbeek [1996]	RATGP95	6,499	2[1]	6
Hartmann and Wenzel [1995]	HW95	12,935	0.14[2]	7
Kudryavtsev [2004]	KSM03	28,806	0.025[3]	8

Table 1: Overview of tidal catalogs, the number of waves used to calculate the tide generating potential and root-mean-square (RMS) accuracy in time and frequency domains. *All catalogues were transformed into the HW95 normalization and format by *Wenzel* [1996] enabling a comparison of the number of waves. [1]Using a benchmark series in the range between 1970-2029 [*Hartmann and Wenzel*, 1995]. [2]Using DE200 ephemerides in a timespan of 300 years [*Hartmann and Wenzel*, 1995]. [3]Using DE/LE405 ephemerides in the timespan 1600-2200 [*Kudryavtsev*, 2004]. #Keyword described in Section 4.2.2.

3.2. A quick start guide

The code contained in the file *test.py* illustrates how **PyGTide** is used to calculate the Earth tide potential for a defined geo-location and over a specified time period:

```
# PyGTide: A Python module and wrapper for ETERNA PREDICT
# to compute gravitational tides on Earth
import pygtide
import datetime as dt

# create a PyGTide object
pt = pygtide.pygtide()

# define a start date
start = dt.datetime(2017,1,1)

# calculate the gravitational tides
latitude = 49.00937
longitude = 8.40444
height = 120
start = dt.datetime(2017,1,1)
duration = 31*24
samplerate = 3600
pt.predict(latitude, longitude, height, start, duration, samplerate)

# retrieve the results as dataframe
data = pt.results()

# output
print(data)
```

Figure 1 shows a plot of the data that was produced by the test file.

PyGTide, doi:10.5281/zenodo.1346260

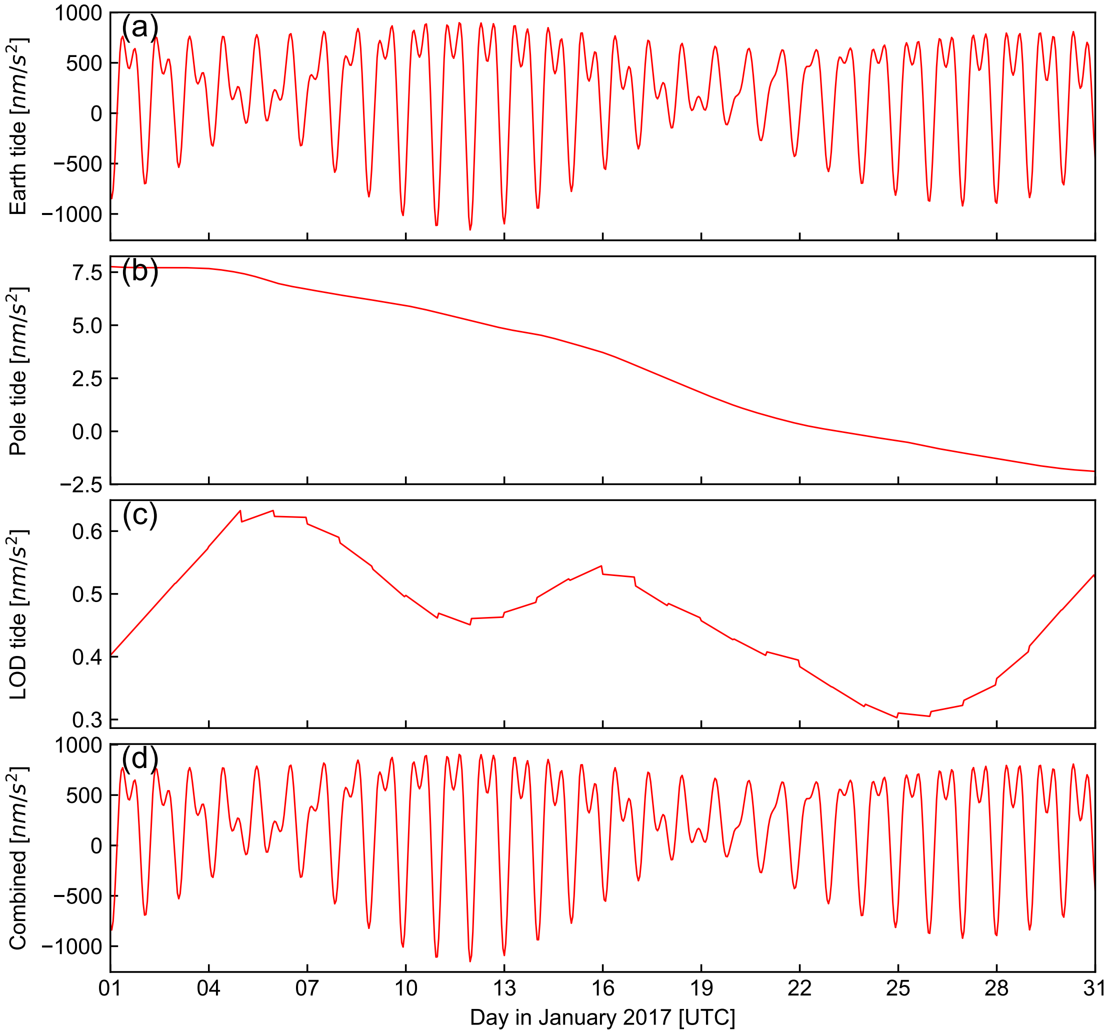

Figure 1: Earth tides calculated for Karlsruhe (Germany) in January 2017 using **PyGTide**: (a) Simple Earth tides. The gravity variations induced by the pole wobble (b) and length of day changes (c). (d) Combined Earth tides considering all effects.

3.3. Comparison between TSoft and PyGTide

It is useful to verify the calculations of **PyGTide** with those of existing software. Here, we used *TSoft* which is a peer-reviewed software package for the analysis of time series and Earth tides [*Van Camp and Vauterin*, 2005]. According to the *TSoft* user guide, the program uses the tidal catalogue T87 developed by *Tamura* [1987]. *Hartmann and Wenzel* [1995] found that this is ≈50 times less accurate than their later development. However, their catalogue is ≈7 times less accurate than KSM03 developed by *Kudryavtsev* [2004]. Further, *TSoft* does not account for pole wobble or length of day influences. Theoretically, this should make **PyGTide** significantly more accurate than *TSoft*.

The output of **PyGTide** (without pole and LOD tide) was compared with that from *TSoft* using the same wave groups. Earth tides were calculated for Karlsruhe (Latitude: 49.00937°, Longitude: 8.40444°, Height 120 m) in the arbitrary timespan January 2017. Calculations were done using an hourly sampling rate, twice with **PyGTide** using the T87 and KSM03 tidal catalogues and once with *TSoft*. Figure 2 shows the results of this comparison including gravity residuals. It is clear that **PyGTide** produces Earth tide predictions that are equal to *TSoft* when using the same tidal catalogue and errors are thought to be caused by rounding (blue). The larger residuals when comparing *TSoft*'s predictions with **PyGTide** (red) reflect the higher precision achieved using the KSM03 catalogue (Table 1).

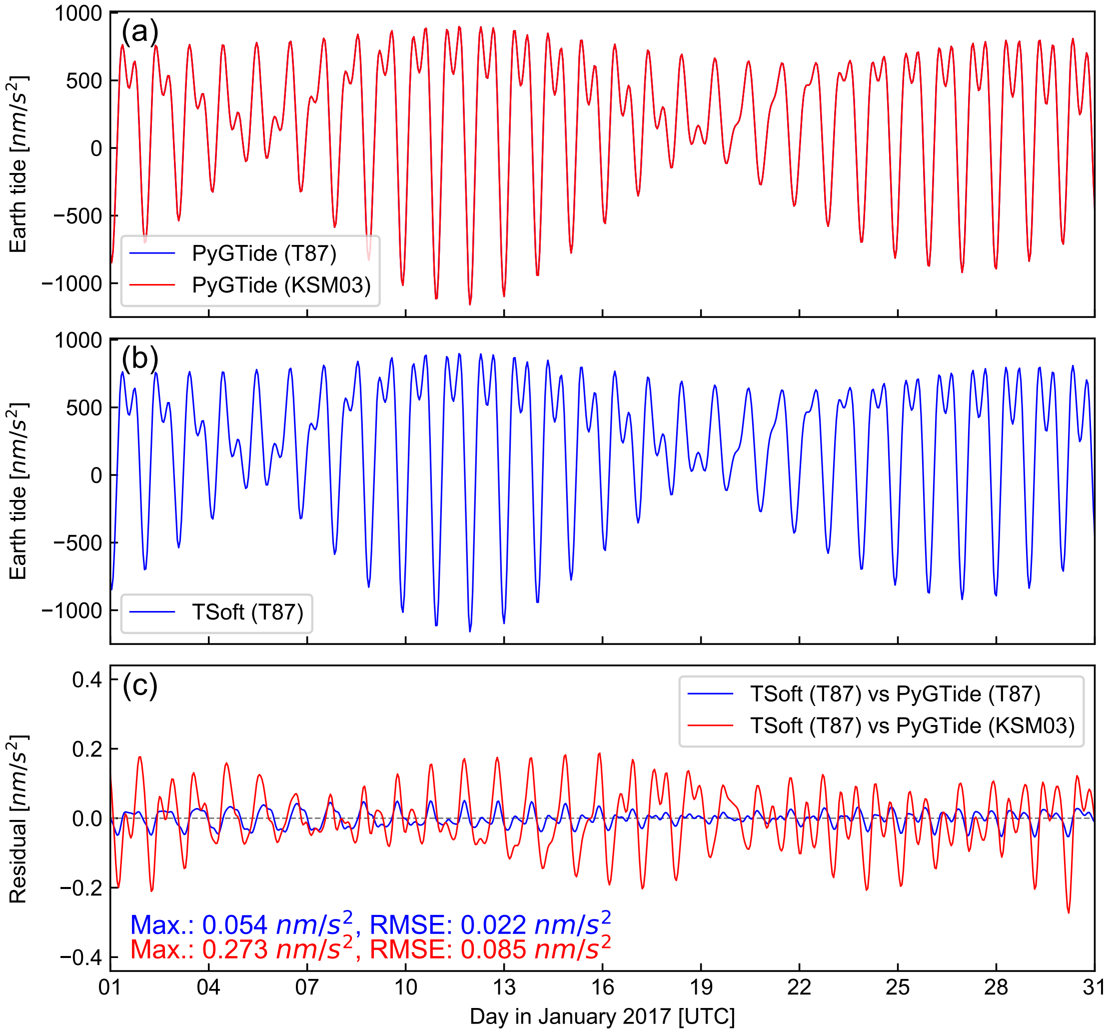

Figure 2: Earth tide time series calculated for Karlsruhe (Germany) in January 2017 using (a) **PyGTide** with the tidal catalogues T87 [*Tamura*, 1987]) and KSM03 [*Kudryavtsev*, 2004], (b) *TSoft* [*Van Camp and Vauterin*, 2005] with T87. (c) Residuals when **PyGTide** results are compared with *TSoft* for the different tidal catalogues.

4. Description of the **PyGTide** class

4.1. **PyGTide**: A module and wrapper for etpred

While Fortran is still popular because it provides computational speed, scientific computation has rapidly evolved and provides many more convenient options. However, re-writing the code in a modern programming language would cost significant effort and time apart from the risk of sacrificing the computational integrity of the original code. The best way forward was therefore to make the code available as a Python package by using and preserving the original code. This was made possible through *F2PY*, a *Fortran* to *Python* interface generator dedicated to provide a connection between *Python* and *Fortran* languages. However, this was not straight forward and required significant work.

To facilitate a compilation of the original module with *F2PY* and to facilitate handover of input variables as well as the calculated output, the code was significantly streamlined and modernised. The following contains a list of changes to the original *Fortran 77* code:

- *Fortran* COMMON blocks were transformed into modules
- modules were streamlined and divided into logical parts
- DATA blocks were changed to variables
- continuous lines of code (line breaks) were updated for *Fortran 95* compatibility
- the main program was changed into a subroutine (for *F2PY* compliance)
- repeated constants were defined once only in modules
- a new module (*inout*) was created to facilitate variable exchange between *Python* and *Fortran*
- a new array (*args*) was created to hand over the desired input arguments from *Python*
- a new allocatable array (*etpdata*) was created to hand over the calculated data

While working on the code, the following bugs were encountered and fixed:

- The original date and time data contained a rounding bug when the sampling rate was lower than 60 seconds. This was successfully fixed within the original *Fortran* code. Dates and times are now correctly calculated.

A new subroutine was created as an entry point for exchanging variables with Python. The code was then compiled into a module called *etpred* (abbreviation for *ETERNA PREDICT*) using *F2PY* (file name: *etpred.cp36-win_amd64.pyd*). The **PyGTide** class was written as a convenience wrapper for the *etpred* module. **PyGTide** facilitates a class-like access to calculations by means of variable exchange and error checking in order to avoid crashes of the compiled machine code. The following subsections describe the functions of the **PyGTide** class.

4.2. The PyGTide class

4.2.1. Function update():

This function refreshes the internal variables of the class **PyGTide** using the module *etpred*. Always returns *True*.

PyGTide, doi:10.5281/zenodo.1346260

4.2.2. Function predict(latitude, longitude, height, startdate, duration, samprate, **control):

This function takes the user input variables, provides error checking and, if everything is error free, calls the *etpred* module to calculate model tides. Returns *True* upon success. Results can be obtained using the function results().

Mandatory keywords are as follows:

- Keyword latitude: A decimal latitude value (WGS84 coordinate system).
- Keyword longitude: A decimal longitude value (WGS84 coordinate system).
- Keyword height: A decimal height value (WGS84 coordinate system).
- Keyword startdate: A start date for the calculated time series (requires *datetime* format).
- Keyword duration: A decimal duration of the time series (in hours).
- Keyword samprate: A decimal value determining the sampling rate of the calculated time series (in seconds).

Optional **control keywords are as follows:

- Keyword statgravit: Gravity of the station in m/s^2. This is necessary for tidal tilt only. If the gravity is unknown, use a value of less than 1.0 and the program will compute and subsequently use the normal gravity value referring to GRS80 reference system.
- Keyword statazimut: Azimuth of the instrument in degree decimal, reckoned clockwise from north. This parameter is used for tidal tilt, horizontal displacement and horizontal strain only.
- Keyword tidalpoten: Parameter for the tidal potential catalogue to be used. This is specified in Table 1. Default value is 8 for using the latest KSM03 tidal catalog.
- Keyword tidalcompo: Determines the calculated Earth tide component. Defaults to 0. Available components are:

  -1 for tidal potential in m^2/s^2

  0 (default) for tidal gravity in nm/s^2

  1 for tidal tilt in mas, at azimuth statazimut.

  2 for tidal vertical displacement in mm

  3 for tidal horizontal displacement in mm, at azimuth statazimut.

  4 for tidal vertical strain in $nstr$ (10^{-9})

  5 for tidal horizontal strain in $nstr$ (10^{-9}), at azimuth statazimut.

  6 for tidal areal strain in $nstr$ (10^{-9})

  7 for tidal shear strain in $nstr$ (10^{-9})

  8 for tidal volume strain in $nstr$ (10^{-9})

PyGTide, doi:10.5281/zenodo.1346260

- Keyword amtruncate: Amplitude threshold for the tidal potential catalogue m^2/s^2. Defaults to $1 \cdot 10^{-10}$. Only tidal waves with amplitudes exceeding the amplitude threshold are computed. This reduces the execution time, but also the accuracy of the computed tidal signals.

- Keyword poltidecor: Amplitude factor for gravity pole tide correction. If the amplitude factor is greater zero, gravity pole tides will be computed using the *International Earth Rotation Service (IERS)* measurements or *United States Naval Observatory (USNO)* forecasts of daily pole coordinates. Default value is 1.16 [*Boy and Hinderer*, 2006].

- Keyword lodtidecor: Amplitude factor for gravity length of day (LOD) tide correction. If the amplitude factor is greater zero, gravity LOD tides will be computed using the *International Earth Rotation Service (IERS)* measurements or *United States Naval Observatory (USNO)* forecasts of daily pole coordinates. Default value is 1.16 REF??.

- Keyword fileout: Legacy support: Value determines whether or not the output is written to the text files that were used by *ETERNA PREDICT*. A value of 0 suppresses the file output. If set to 1, the routine writes two text files called *pygtide.out.prd* and *pygtide.out.prn* in the original format into the directory of the module. Defaults value is 0 (disabled).

- Keyword screenout: Legacy support: Value determines if the original *Fortran* screen output is enabled or disabled. If set to 1, the routine writes output to the screen (but not the Python terminal!). Defaults value is 0 (output is redirected to NULLFILE).

4.2.3. Function results(round=6):

If *etpret* was successfully executed, then returns the results from the prediction in *pandas* dataframe format. Else returns *False*. The keyword *round* determines the number of decimal digits to which the results are rounded (default is 6).

4.2.4. Function data(round=6):

If *etpret* was successfully executed, then returns the results from the prediction in raw format. Else returns *False*. The keyword *round* determines the number of decimal digits to which the results are rounded (default is 6).

4.2.5. Function datetime():

If *etpret* was successfully executed, then returns the date and time of the predicted time series in string format. Else returns *False*.

4.2.6. Function update_etddt():

This function updates the database which contains the time correction between UTC and Terrestrial Time. The data is automatically pulled from the *International Earth Rotation Service (IERS)* website via FTP. The data is merged with existing data preceding this database.

Note: This function relies on the external file *pygtide_update_data.py* which must be located in the same directory as the **PyGTide** class.

PyGTide, doi:10.5281/zenodo.1346260

4.2.7. Function **update_etpolut1()**:

This function updates the database which contains the pole coordinates. The data is automatically pulled from the *International Earth Rotation Service (IERS)* (FTP) and *United States Naval Observatory (USNO)* (FTP) online services. The data is automatically merged and saved as *ETERNA PREDICT* compliant database.

Note: This function relies on the external file *pygtide_update_data.py* which must be located in the same directory as the **PyGTide** class.

References

Acworth, R. I., L. J. S. Halloran, G. C. Rau, M. O. Cuthbert, and T. L. Bernardi (2016), An objective frequency domain method for quantifying confined aquifer compressible storage using Earth and atmospheric tides, *Geophysical Research Letters*, *43*(22), 611–671, doi:10.1002/2016GL071328.

Agnew, D. C. (2010), Earth Tides, *Geodesy: Treatise on Geophysics*, p. 163.

Boy, J. P., and J. Hinderer (2006), Study of the seasonal gravity signal in superconducting gravimeter data, *Journal of Geodynamics*, *41*(1-3), 227–233, doi:10.1016/j.jog.2005.08.035.

Bredehoeft, J. D. (1967), Response of well-aquifer systems to Earth tides, *Journal of Geophysical Research*, *72*(12), 3075–3087, doi:10.1029/JZ072i012p03075.

Büllesfeld, F. J. (1985), *Ein Beitrag zur harmonischen Darstellung des gezeitenerzeugenden Potentials*, C: Deutsche Geodätische Kommission bei der Bayerischen, Beck.

Cartwright, D. E., and A. C. Edden (1973), Corrected Tables of Tidal Harmonics, *Geophysical Journal International*, *33*(3), 253–264, doi:10.1111/j.1365-246X.1973.tb03420.x.

Chapman, S. (1951), Atmospheric Tides and Oscillations, in *Compendium of Meteorology*, pp. 510–530, American Meteorological Society, Boston, MA, doi:10.1007/978-1-940033-70-9_43.

Doodson, A. T. (1921), The Harmonic Development of the Tide-Generating Potential, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *100*(704), 305–329, doi:10.1098/rspa.1921.0088.

ESA (2013), Pole Tide: https://gssc.esa.int/navipedia/index.php/Pole_Tide.

Eubanks, T. M. (1993), Variations in the orientation of the Earth, in *Contributions of Space Geodesy to Geodynamics: Earth Dynamics*, vol. 24, pp. 1–54, doi:10.1029/GD024p0001.

Hartmann, T., and H. . Wenzel (1995), The HW95 tidal potential catalogue, *Geophysical Research Letters*, *22*(24), 3553–3556, doi:10.1029/95GL03324.

Hector, B., and J. Hinderer (2016), PyGrav, a Python-based program for handling and processing relative gravity data, *Computers and Geosciences*, *91*, 90–97, doi:10.1016/j.cageo.2016.03.010.

Kudryavtsev, S. M. (2004), Improved harmonic development of the Earth tide-generating potential, *Journal of Geodesy*, *77*(12), 829–838, doi:10.1007/s00190-003-0361-2.

PyGTide, doi:10.5281/zenodo.1346260

Longuevergne, L., J. Boy, N. Florsch, D. Viville, G. Ferhat, P. Ulrich, B. Luck, and J. Hinderer (2009), Local and global hydrological contributions to gravity variations observed in Strasbourg, *Journal of Geodynamics*, *48*(3-5), 189–194, doi:10.1016/j.jog.2009.09.008.

Meinzer, O. E. (1939), Ground water in the United States, a summary of ground-water conditions and resources, utilization of water from wells and springs, methods of scientific investigation, and literature relating to the subject, *Tech. rep.*, U.S. G.P.O.

Ray, R. D., D. J. Steinberg, B. F. Chao, and D. E. Cartwright (1994), Diurnal and semidiurnal variations in the Earth's rotation rate induced by oceanic tides, *Science*, *264*(5160), 830–832, doi:10.1126/science.264.5160.830.

Roosbeek, F. (1996), RATGP95: a harmonic development of the tide-generating potential using an analytical method, *Geophysical Journal International*, *126*(1), 197–204, doi:10.1111/j.1365-246X.1996.tb05278.x.

Shen, W., and C. Peng (2016), Detection of different-time-scale signals in the length of day variation based on EEMD analysis technique, *Geodesy and Geodynamics*, *7*(3), 180–186, doi:10.1016/j.geog.2016.05.002.

Tamura, Y. (1987), A harmonic development of the tide-generating potential, *Bulletin d'Informations des Marées Terrestres*, *99*, 68136855.

Tamura, Y. (1993), Additional terms to the tidal harmonic tables, in *Proceedings 12th International Symposium on Earth Tides*, pp. 345–350, Science Press, Beijing/New York, Beijing.

Van Camp, M., and P. Vauterin (2005), Tsoft: Graphical and interactive software for the analysis of time series and Earth tides, *Computers and Geosciences*, *31*(5), 631–640, doi:10.1016/j.cageo.2004.11.015.

Van Camp, M., O. de Viron, A. Watlet, B. Meurers, O. Francis, and C. Caudron (2017), Geophysics From Terrestrial Time-Variable Gravity Measurements, *Reviews of Geophysics*, doi:10.1002/2017RG000566.

Wahr, J. M. (1985), Deformation induced by polar motion, *Journal of Geophysical Research*, *90*(B11), 9363–9368, doi:10.1029/JB090iB11p09363.

Wenzel, H.-G. (1996), The nanoGal software: Earth tide data processing package: Eterna 3.3, *Bulletin d'Informations des Marées Terrestres*, *124*, 9425–9439.

Xi, Q. W., and T. H. Hou (1987), A new complete development of the tide-generating potential for the epoch J2000. 0, *Bulletin d'Informations des Marées Terrestres*, *99*, 67666812.

Xu, J., H. Sun, and B. Ducarme (2004), A global experimental model for gravity tides of the Earth, *Journal of Geodynamics*, *38*(3-5 SPEC.ISS.), 293–306, doi:10.1016/j.jog.2004.07.003.

PyGTide, doi:10.5281/zenodo.1346260