Dynamic Gravity Processor Documentation

Release 0.1

Zachery Brady, Daniel Aliod, Nigel Brady, Chris Bertinato

Jul 05, 2018

Contents:

CHAPTER 1

dgp

1.1 dgp package

1.1.1 Subpackages

dgp.lib package

Submodules

dgp.lib.gravity_ingestor module

gravity_ingestor.py Library for gravity data import functions

dgp.lib.gravity_ingestor.**read_at1a**(*path*, *fill_with_nans=True*, *interp=False*)
Read and parse gravity data file from DGS AT1A (Airborne) meter.

CSV Columns: gravity, long, cross, beam, temp, status, pressure, Etemp, GPSweek, GPSweekseconds

Parameters

- **path** (*str*) – Filesystem path to gravity data file
- **fill_with_nans** (*boolean, default True*) – Fills time gaps with NaNs for all fields
- **interp** (*boolean, default False*) – Interpolate all NaNs for fields of type numpy.number

Returns Gravity data indexed by datetime.

Return type pandas.DataFrame

dgp.lib.gravity_ingestor.**read_zls**(*dirpath*, *begin_time=None*, *end_time=None*, *excludes=['.*']*)
Read and parse gravity data file from ZLS meter.

Files are segmented by hour and data is presented as ASCII in a fixed-width format.

Columns: line name, year, day, hour, minute, second, gravity, spring tension, cross coupling, raw beam, vcc, al, ax, ve2, ax2, xacc2, lacc2, xacc, lacc, par port, platform period

Parameters

- **dirpath** (*str*) – Filesystem path to directory containing files
- **begin_time** (*datetime, optional*) – Data start time if not importing from the first file in the directory
- **end_time** (*datetime, optional*) – Data end time if not importing to the last file in the directory
- **excludes** (*list*) – Files and directories to exclude from directory listing.

Returns Gravity data indexed by datetime.

Return type pandas.DataFrame

dgp.lib.time_utils module

dgp.lib.time_utils.**convert_gps_time**(*gpsweek*, *gpsweekseconds*, *format='unix'*)
convert_gps_time :: (String -> String) -> Float

Converts a GPS time format (weeks + seconds since 6 Jan 1980) to a UNIX timestamp (seconds since 1 Jan 1970) without correcting for UTC leap seconds.

Static values gps_delta and gpsweek_cf are defined by the below functions (optimization) gps_delta is the time difference (in seconds) between UNIX time and GPS time. gps_delta = (dt.datetime(1980, 1, 6) - dt.datetime(1970, 1, 1)).total_seconds()

gpsweek_cf is the coefficient to convert weeks to seconds gpsweek_cf = 7 * 24 * 60 * 60 # 604800

Parameters

- **gpsweek** – Number of weeks since beginning of GPS time (1980-01-06 00:00:00)
- **gpsweekseconds** – Number of seconds since the GPS week parameter

Returns (float) unix timestamp (number of seconds since 1970-01-01 00:00:00)

dgp.lib.time_utils.**datenum_to_datetime**(*timestamp*)

dgp.lib.time_utils.**datetime_to_sow**(*dt*)

dgp.lib.time_utils.**leap_seconds**(***kwargs*)
leapseconds :: Variable type -> Integer

Look-up for the number of leapseconds for a given date.

Parameters

- **week** – Number of weeks since beginning of GPS time (1980-01-06 00:00:00)
- **seconds** – If week is specified, then seconds of week since the beginning of Sunday of that week, otherwise, Unix time in seconds since January 1, 1970 UTC.
- **date** – Date either in the format MM-DD-YYYY or MM/DD/YYYY
- **datetime** – datetime-like

Returns (integer) Number of accumulated leap seconds as of the given date.

dgp.lib.trajectory_ingestor module

trajectory_ingestor.py Library for trajectory data import functions

dgp.lib.trajectory_ingestor.**import_trajectory**(*filepath*, *delim_whitespace=False*, *interval=0*, *interp=False*, *is_utc=False*, *columns=None*, *skiprows=None*, *timeformat='sow'*)

Read and parse ASCII trajectory data in a comma-delimited format.

Parameters

- **path** – str Filesystem path to trajectory data file
- **interval** – float, default 0 Output data rate. Default behavior is to infer the rate.
- **interp** – list of ints or list of strs, default None Gaps in data will be filled with interpolated values. List of column indices (list of ints) or list of column names (list of strs) to interpolate. Default behavior is not to interpolate.
- **is_utc** – boolean, default False Indicates that the timestamps are UTC. The index datetimes will be shifted to remove the GPS-UTC leap second offset.
- **colums** – list of strs, default: None Strings to use as the column names.
- **skiprows** – list-like or integer or callable, default None Line numbers to skip (0-indexed) or number of lines to skip (int) at the start of the file. If callable, the callable function will be evaluated against the row indices, returning True if the row should be skipped and False otherwise. An example of a valid callable argument would be lambda x: x in [0, 2].
- **timeformat** – 'sow' | 'hms' | 'serial', default: 'hms' Indicates the time format to expect. The 'sow' format requires a field named 'week' with the GPS week, and a field named 'sow' with the GPS seconds of week. The 'hms' format requires a field named 'mdy' with the date in the format 'MM/DD/YYYY', and a field named 'hms' with the time in the format 'HH:MM:SS.SSS'. The 'serial' format (not yet implemented) requires a field named 'datenum' with the serial date number.

Returns DataFrame

Module contents

1.1.2 Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index