GENeSYS-MOD - Hourly Data Guide

Table of contents

Table of contents **3**
Introduction **4**
List of all Parameters in the hourly data file **5**
Explanation of Parameters **6**
General 6
TS_COOL_LOW 6
TS_HEAT_HIGH 6
TS_HEAT_LOW 6
TS_HP_Airsource 6
TS_HP_GROUNDSOURCE 6
TS_HYDRO_ROR 7
TS_LOAD 7
TS_MOBILITY_PSNG 7
TS_PV_AVG 7
TS_PV_INF 7
TS_PV_OPT 8
TS_PV_TRA 8
TS_WIND_OFFSHORE 8
TS_WIND_OFFSHORE_DEEP 8
TS_WIND_OFFSHORE_SHALLOW 8
TS_WIND_ONSHORE_AVG 8
TS_WIND_ONSHORE_INF 9
TS_WIND_ONSHORE_OPT 9
Timeseries Script 9
Step 1: Requirements 9
Step 2: Set Up 10
Step 3: Change Settings 10
Step 4: Run Code and Use Output 11

Introduction

The following document provides additional information on the different parameters/sheets that need to be filled, in order to adapt the GENeSYS-MOD hourly data files to the respective case-study.
All Sheets are relevant to the case-study and need to be filled.
The parameters marked RED must be filled using original data sources while the Parameters in BLUE can be filled using the ATLITE package which will be explained at the end of the document (see section **Timeseries Script**)

For further information please also use the Technical Manual:
https://drive.google.com/drive/folders/1IPVzJs7gFc7eUe5uhHl6RdXrmP7Ky-bB
Some further information on how to install and use jupyter notebook:
https://docs.jupyter.org/en/latest/install/notebook-classic.html

Contact: joh@wip.tu-berlin.de

List of all Parameters in the hourly data file

Parameter	Explanation	Unit
TS_COOL_LOW	Time Series cooling low	Fraction
TS_HEAT_HIGH	Time Series Heat high	Fraction
TS_HEAT_LOW	Time Series Heat low	Fraction
TS_HP_AIRSOURCE	Time Series Heatpump Air Source	Fraction
TS_HP_GROUNDSOURCE	Time Series Heatpump Groundsource	Fraction
TS_HYDRO_ROR	Time Series Hydro Run of river	Fraction
TS_LOAD	Time Series Load Power	Fraction
TS_MOBILITY_PSNG	Time Series Mobility Passenger	Fraction
TS_PV_AVG	Time Series PV Solar Average	Fraction
TS_PV_INF	Time Series PV Solar Inferior	Fraction
TS_PV_OPT	Time Series PV Solar Optimal	Fraction
TS_PV_TRA	Time Series PV Solar Tracking	Fraction
TS_WIND_OFFSHORE	Time Series Wind Offshore	Fraction
TS_WIND_OFFSHORE_DEEP	Time Series Wind Offshore Deep	Fraction
TS_WIND_OFFSHORE_SHALLOW	Time Series Wind Offshore Shallow	Fraction
TS_WIND_ONSHORE_AVG	Time Series Wind Offshore Average	Fraction
TS_WIND_ONSHORE_INF	Time Series Wind Onshore Inferior	Fraction
TS_WIND_ONSHORE_OPT	Time Series Wind Onshore Optimal	Fraction

Explanation of Parameters

General

- All Parameters are hourly (8760 entries for a full year)

TS_COOL_LOW

Time Series Cooling Low

Sheet can be filled using the Atlite package (see section x for a guide)

TS_HEAT_HIGH

Time Series Heat High

Time Series for IndustrialHeat demand above 1000 °C.
This time series represents heat usage throughout the year, showing the proportion of heat used for each hour compared to other hours. While the data can be provided in a normalized format, it isn't necessary because the time series reduction algorithm automatically normalizes it when the model is run. The key is to use a consistent value for the peak heat usage and adjust all other values proportionally to maintain the correct relationships between them.

TS_HEAT_LOW

Time Series Heat LOW

Time Series for residential heat demand.
Sheet can be filled using the Atlite package (see section x for a guide)

TS_HP_Airsource

Time Series Heat Pump Air Source

Sheet can be filled using the Atlite package (see section x for a guide)

TS_HP_GROUNDSOURCE

Time Series Heatpump Groundsource

Sheet can be filled using the Atlite package (see section x for a guide)

TS_HYDRO_ROR

Time Series Hydro Run of river

This time series represents the capacity factor for Hydro_ROR (run-of-river). The capacity factor for run-of-river hydroelectric plants can vary significantly depending on site-specific conditions, such as river flow consistency and plant design. Typically, it ranges between **40% and 80%**.

To calculate the capacity factor, you can use historical data on installed capacity and actual energy production, applying the following formula:

$$\text{Capacity Factor} = \frac{\text{Actual Annual Energy Production (MWh)}}{\text{Installed Capacity (MW)} \times 8,760}$$

To account for seasonal, weekly, or daily variation (depending on the detail of your sources) you can vary the capacity factor. Below is an example calculation that has monthly variablity.

Month	Energy Produced (MWh)	Installed Capacity (MW)	Hours in Month	Capacity Factor (%)
January	3900	10	744	52.42
February	5200	10	672	77.38
March	4700	10	744	63.17
April	1200	10	720	16.67
May	1600	10	744	21.51
June	1500	10	720	20.83
July	1300	10	744	17.47
August	1100	10	744	14.78
September	1000	10	720	13.89
October	1200	10	744	16.13
November	1800	10	720	25
December	3850	10	744	51.75

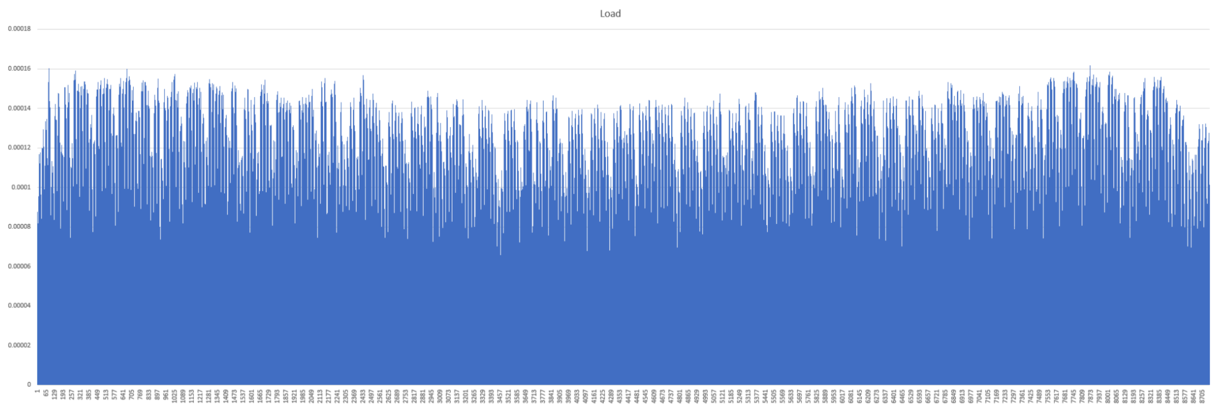
NOTE! In the absence of detailed data, assuming a value between **40% and 50%** can be a reasonable starting point.

TS_LOAD

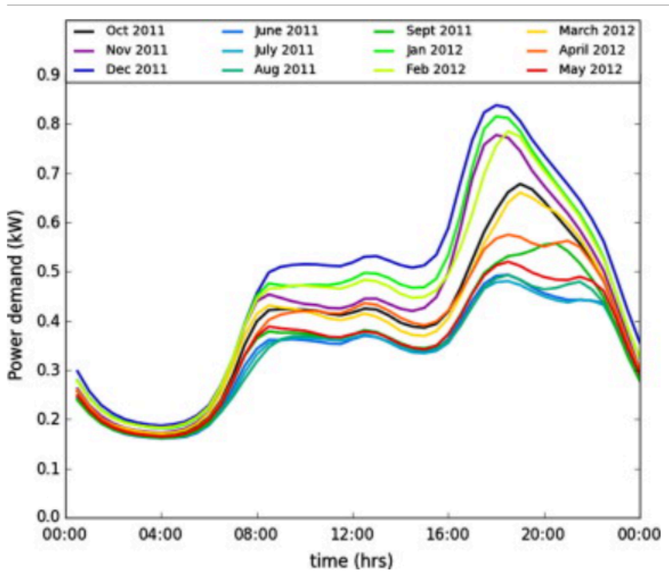
Time Series Load Power

This time series represents power usage (demand profile) throughout the year, showing the proportion of power used for each hour compared to other hours. While the data can be provided in a normalized format, it isn't necessary because the time series reduction algorithm automatically normalizes it when the model is run. The

key is to use a consistent value for the peak power usage and adjust all other values proportionally to maintain the correct relationships between them.
Figure TS_LOAD shows an example load profile visualised over all 8760 hours:



The following image (taken from doi.org/10.1016/j.geoforum.2014.04.01) shows an example of daily load profiles and how they change over the year.



TS_MOBILITY_PSNG

Time Series Mobility Passenger

This time series represents passenger mobility (demand profile) throughout the year, illustrating the relative volume of passengers for each hour compared to others. Although data can be supplied in a normalized format, it isn't required, as the time series reduction algorithm automatically normalizes the values when the model is executed. The key is to establish a consistent value for peak passenger volume and adjust all other values proportionally to preserve the correct relationships.

TS_PV_AVG

Time Series PV Solar Average

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_PV_INF

Time Series PV Solar Inferior

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_PV_OPT

Time Series PV Solar Optimal

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_PV_TRA

Time Series PV Solar Tracking

Sheet can be filled using the Atlite package (see section x for a guide)

TS_WIND_OFFSHORE

Time Series Wind Offshore

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_WIND_OFFSHORE_DEEP

Time Series Wind Offshore Deep

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_WIND_OFFSHORE_SHALLOW

Time Series Wind Offshore Shallow

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_WIND_ONSHORE_AVG

Time Series Wind Offshore Average

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_WIND_ONSHORE_INF

Time Series Wind Onshore Inferior

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

TS_WIND_ONSHORE_OPT

Time Series Wind Onshore Optimal

Sheet can be filled using the Atlite package (see section Timeseries Script for a guide)

Timeseries Script

The timeseries script can be used to create the capacity factors for solar PV, Wind (Onshore and Offshore) and Heatpumps, as well as residential heating and cooling demand. It uses functions from atlite and the ERA5 weather dataset, which is downloaded and transformed locally. This section provides a step by step guide on how to use the script and change settings to fit a specific case study:

Step 1: Requirements

Python

If you don't have Python installed on your system already you can download it here: https://www.python.org/downloads/

Jupyter Notebook

You will need a Jupyter Notebook to run the script. You can install it in one of the following ways:

- Directly install Jupyter Notebook by following the instructions here: https://jupyter.org/instal
- Alternatively, you can install Jupyter via **Anaconda**, which provides a bundled environment manager and Jupyter Notebook setup: https://www.anaconda.com/download

Packages

To use this script, you will also need to have the following packages installed:

- numpy
- matplotlib
- seaborn
- pandas
- geopandas (version v. 0.14.4 or older)
- scikit-learn
- cartopy
- xarray
- atlite

You can install all the necessary packages with the following commands:

pip install numpy matplotlib seaborn pandas scikit-learn cartopy xarray atlite

pip install geopandas==0.14.4

API key

In addition, in order to be able to load the ERA5 cutouts, you need an API key. To activate the API key, follow the instructions at

https://cds.climate.copernicus.eu/how-to-api

Step 2: Set Up

The Timeseries script, along with all necessary data files and functions, can be found here:
https://github.com/GENeSYS-MOD/GENeSYS_MOD.tools/tree/main/GIS_%26_Timeseries_Tool

You can either clone the repository using a version control tool like SourceTree, or download the files directly to your computer. Be sure to note the location where you save the files.
Once the files are on your system, open the file GENeSYS-MOD_RES_Tool.ipynb using **Jupyter Notebook**.

Step 3: Change Settings

Within the file some lines need to be changed to fit your specific location:

Cell 2:

```
timeframe = "2018-01-01"   # timeframe for the timeseries. format example: 2018 OR 2018-01 OR 2018-01-01
filename = "germany_test"   # filename for the downloaded raw data (cutout). this needs to be changed for every new run/year/regi
output_dir = create_output_folder(timeframe)   # creates an output folder, if it does not exist yet

# choose either 0 for country-level (NUTS-0) or 1 for major socio-economic regions (NUTS-1)
admin = 0

# country ISO 3166-1 alpha-2 code. If all regions of cutout are needed, leave list empty
regions = ["DE"]

# cutout cooridnates. Leave empty, if the cutout should be taken from a shapefile (with above specified regions)
cutout_north_west = []
cutout_south_east = []
```

Adjust the time frame according to your case study requirement and choose a filename.Set admin= 0 if you want the data on a national level and admin = 1 for a sub-national level (regions, states, provinces). For regions, write the ISO 3166-1 alpha-2 code of your desired country (list can be found here: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2).

Cell 3:

```
In [ ]: # Here, you need to define the PV slope and azimuth of the panels (defaults to slope of 36.7 and azimuth of 180 if not defined)
        pv_slope = 36.7
        pv_azimuth = 180
```

You can adjust the tilt (pv_slope) and orientation (pv_azimuth) of the solar panel. The optimal tilt angle for your location can be calculated here: https://footprinthero.com/solar-panel-tilt-angle-calculator
For the azimuth it is common to use 180° for the northern hemisphere (panel facing south) and 0° for the southern hemisphere (panel facing north).

Cell 4:

```
In [ ]: cutout = get_cutout(filename,
                            timeframe,
                            regions=regions,
                            dx=0.25,
                            dy=0.25)
```

If needed adjust the spatial resolution of the cutout.

Alternative: Define cutout with coordinates

The steps above define a cutout based on national borders for your selected region. If you want to cover a different area—for example, to include offshore regions for analyzing wind capacity—you can specify the area using geographic coordinates.

```
# cutout cooridnates. Leave empty, if the cutout should be taken from a shapefile (with above specified regions)
cutout_north_west = [17, -20] #latitude,longitude
cutout_south_east = [12, -11]
```

In **Cell 2**, enter two coordinates representing the northwest and southeast corners of your desired rectangular cutout.

```
In [*]: cutout = get_cutout(filename,
                            timeframe,
                            #regions=regions,
                            cutout_north_west = cutout_north_west,
                            cutout_south_east = cutout_south_east,
                            dx=0.25,
                            dy=0.25)
```

Then, in **Cell 3**, when defining your cutout, include these coordinates as parameters and remove the regions parameter by commenting it out.

Step 4: Run Code and Use Output

After adjusting the settings, run the Jupyter Notebook cell by cell from top to bottom. **Note**: Downloading the weather data from Atlite can take a significant amount of time, depending on the size of the region and the timeframe you have selected. It is recommended to use a device with ample system memory (at least 24GB of RAM) and ensure you have a stable internet connection and uninterrupted power supply to avoid issues during the download and processing.

Once the data processing is complete, you will see time series CSV files in your output folder. Open these files and copy the columns into your desired hourly data file for further use.

Additionally, the script will display a map showing the area covered by the cutout and provide the means of all the generated factors, allowing you to check for plausibility.