

pymedeas models

This is the official repo for the models of the European MEDEAS project (www.medeas.eu). Please register to the mailing list in order to receive the most recent news about the project.

The available models are the result of translating the original MEDEAS Vensim models (provided as well in the repo) into Python using a fork of version 0.8.2 of [pysd](#) library (included in the code for now).

Currently, pymedeas_w and pymedeas_eu models work fully (in upcoming releases, other regional models might be included).

Prior to running any simulation, in order to choose which of the two models to run (pymedeas_w or pymedeas_eu), you need to open the config.ini file that is found in the main folder using a text editor, and modify the MODEL parameter according to your choice.

Please note that to run pymedeas_eu you will need to import the values of some variables from the results of a simulation run with pymedeas_w. Therefore, before running any simulation with pymedeas_eu, you need to run a similar simulation with the pymedeas_w model. Then, when you launch the pymedeas_eu simulation, you will be asked to load the results file (csv) from the pymedeas_w simulations.

Python 3.7 is required to run the code.

Installation instructions for MS Windows

1. Clone or download the repo in your computer
2. Download and install [Anaconda 3.5 \(Python 3.7\)](#) using the default parameters (Miniconda could be used instead of Anaconda). Write down the installation directory, because you will need it later (we will call this INSTALL_PATH from now on). Read NOTE 1 and NOTE 2 at the end of this section for particular installation cases (e.g. Anaconda already installed)
3. Get the Anaconda installation path (INSTALL_PATH from now on) by running the following command:

```
where anaconda
```

4. Add the INSTALL_PATH to the Windows path. To do so, open the terminal (click on Windows Start Menu, write cmd and type Enter) and run the following command (replacing INSTALL_PATH by your actual installation path):

```
SETX PATH "%PATH%;INSTALL_PATH\Scripts;INSTALL_PATH"
```

5. Log out from your user account and log back in to apply the changes. Verify that the INSTALL_PATH was added to your system PATH, by running:

```
echo %PATH%
```

You should now be able to see the Anaconda installation directory among the other directories in your system PATH.

6. To install pipenv package, open the terminal (cmd) again and run:

```
pip install pipenv
```

7. Still in the terminal, go to the folder where you downloaded the MEDEAS model using cd command (replacing the below path with the actual path to the folder where you downloaded the model):

```
cd C:\Users\UserName\MEDEAS-model
```

8. From there, run the following command to install all Python packages required to run the model in a virtual environment:

```
pipenv install --python=python.exe
```

9. Congratulations, you can move to the next step! Go to Running a simulation from terminal section to verify that everything went well during the previous steps, and to try to run the model.

- NOTE 1: By default Anaconda3 installs in your user directory as C:\Users\UserName\Anaconda3 but if you have spaces in the UserName (e.g. C:\Users\Guido van Rossum\Anaconda3) Anaconda might have issues to install further packages. In that case chose to install for All Users (you will need administrative privileges), which will install Anaconda3 in C:\ProgramData\Anaconda3.
- NOTE 2 : If you have an older version of Anaconda installed (Current version 3.5 was released in October 2018) you should install the new one while keeping/removing the old one, or upgrade to the new version (only possible for > Anaconda 3) using the following command:

```
conda install python=3.7 anaconda=custom
```

Installation Instructions for Linux

1. Clone or download the repo in your computer
2. Download Anaconda 3.5 for linux (Miniconda could be used instead of Anaconda). Open the terminal, go to the folder where the package was saved and run (replacing Anaconda3-5.3.0-Linux-x86_64.sh by the name of downloaded file):

```
bash Anaconda3-5.3.0-Linux-x86_64.sh
```

3. Follow the installation instructions, leaving all parameters by default. This will install anaconda in your home directory (e.g. /home/user/anaconda3). When asked: Do you wish the installer to initialize Anaconda3 in your /home/roger/.bashrc ? [yes|no], Answer yes. You can say no to the installation of Microsoft VSCode.
4. The following command should point to your anaconda installation directory:

```
which python
```

5. Install pipenv from terminal with the following command:

```
pip install pipenv
```

6. Still from the console, go to the folder where the model files are, and execute:

```
pipenv install --python 3.7
```

7. In the output of the previous command you should be able to identify the path where the virtual environment was created.
8. You are done. Go to Running a simulation from terminal section of this document to verify that everything went well during the previous steps, and to try to run the model.

Installation Instructions for MacOS

1. Clone or download the repo in your computer
2. Download and install Anaconda 3.5 for MacOS (Python 3.7) (Miniconda could be used instead of Anaconda).
3. Open a terminal and install Xcode Command line developer tools with the following command:

```
xcode-select --install
```

4. Confirm that pip executable is inside the anaconda3 folder with the command below (you should see something like /Users/your_user/anaconda3/bin/pip):

```
which pip
```

5. Install pipenv with the following command:

```
pip install pipenv
```

6. Still from the console, go to the folder where the model files are, and execute:

```
pipenv install --python 3.7
```

If the previous step did not work, you need to add pipenv path to your system path. To do that, add the following line at the end of your `.bash_profile` (in your home directory, press `Cmd + Shift + .` (dot) to see the `.bash_profile` file) (replace the `ANACONDA_PATH` by the path where you installed anaconda e.g. `/Users/medeas/anaconda3/`):

```
export PATH="ANACONDA_PATH/bin:$PATH"
```

After saving the file, run the following command in the terminal:

```
source ~/.bash_profile
```

Then try again the command of step 4.

7. Congratulations, you can move to the next step! Go to Running a simulation from terminal section of this document to verify that everything went well during the previous steps, and to try to run the model.

Running a simulation from terminal (Windows/Linux/MacOS)

1. Open the `config.ini` file with a text editor and modify it either to run `pymedeas_w` or `pymedeas_eu` model
2. Open a terminal and go to the project folder (using the `cd` command)
3. Activate the project virtual environment running the following command:

```
pipenv shell
```

4. At this point, you should be able to run a default simulation with the following command:

```
python run.py -x bau -f 2050 -t 0.05 -r 1.0 -p
```

NOTE: to see all user options and default parameter values, run:

```
python run.py -h
```

Using the plot GUI to plot previous simulation results from terminal (in csv format)

1. Open a terminal and go to the project folder (using the `cd` command)
2. If it's not active yet, activate the project virtual environment running the following command:

```
pipenv shell
```

3. Run the following command:

```
python plot_tool.py
```

4. Simulation results (csv file) can be found either in the `pymedeas_w` or the `pymedeas_eu` folder. You can load an unlimited number of files, to compare several simulation results.

Model outputs

Simulation results (csv file) can be found either in the `pymedeas_w` or the `pymedeas_eu` folder.

Unless the user provides the desired output file name with the `-n` option when launching the simulation (e.g. `python run.py -n results_my_scenario`), the default results naming convention is the following:

`results_SCENARIO-NAME_INITIAL-DATE_FINAL-DATE_TIME-STEP.csv`

If a results file with the same name already exists, the characters "_old" will be added at the end of the file name. This can happen up to two times. NOTE that if a fourth simulation with the same name is run, the file of the first simulation result will be automatically deleted.

Python IDE of choice (PyCharm Community Edition)

If you would like to use a graphical IDE instead of the command line, we recommend you to download Pycharm Community edition, which is free and open source. You can download it for [Windows](#), [Linux](#) or [macOS](#).

After installation, follow these steps to get the model working:

1. Open the folder where you downloaded the model: go to File\Open and select the model folder
2. Make Pycharm use the virtual environment that you created for the current project (if it doesn't already):
 - Go to File\Settings (PyCharm\Preferences in macOS) and start typing project interpreter.
 - If the environment you created appears in the drop-down list you can skip the next step.
 - If the environment does not show up, you need to tell Pycharm where to find it. Click on the sprocket wheel icon, in the top right corner and click on Add. This will open a new window, where you should check the option Existing environment, and select the path where your environment was stored:
 - You can get the path to your virtualenv by running the following command in a terminal:

```
pipenv --venv
```

- on Windows it should be something like C:\Users\user_name\.virtualenvs\name_of_virtualenv\Scripts\python.exe
 - on macOS: /Users/user_name/.virtualenvs/name_of_virtualenv/bin/python3.7
 - on Linux: /home/user_name/.virtualenvs/name_of_virtualenv/bin/python3.7
 - Click Ok and Apply
3. In the main menu, click on Run/Edit configurations
 - Click on the plus sign on the top left corner and choose Python
 - Fill in the boxes as follows:
 - Name: Run model
 - Script path: select the run.py file from the project folder
 - Parameters : -s -t 0.03125 -r 1.0 -x bau -p
 - Python Interpreter: select the one you added in step 2 from the drop-down menu
 - Click Ok
 4. On the top right corner of the screen you should now see a play icon beside the words Run model, click on it to run the simulation.

If you would like to be able to run the plot GUI to plot results of previous simulations in PyCharm, repeat step 3 changing the parameters to:

- Name: Plot
- Script path: select the plot_tool.py file from the project folder
- Parameters : (leave blank)
- Python Interpreter: select the one you added in step 2 from the drop-down menu