**Setup of the stand-alone version of OnSSET**

**OnSSET Software Requirements**

**Python** - **Anaconda package**

OnSSET is written in python, an open source programming language used widely in many applications. Python[[1]](#footnote-2) is a necessary requirement for the OnSSET tool to work.

Programming in python usually relies on the usage of pre-defined functions that can be found in the so called modules. In order to work with OnSSET, certain modules need to be installed/updated. The easiest way to do so is by installing Anaconda, a package that contains various useful modules. Anaconda includes all the modules required to run OnSSET.

**Python Interfaces**

1. Integrated Development Environment (IDEs) - PyCharm

Integrated Development Environment programs are used in order to ease the programming process when multiple or long scripts are required. There are plenty IDEs developed for Python (you can find a few here <http://noeticforce.com/best-python-ide-for-programmers-windows-and-mac>). KTH dESA has been using PyCharm as the standard IDE to run OnSSET.

1. Jupyter Notebook (via Anaconda)

Jupyter notebook is a console-based, interactive computing approach providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. Jupyter Notebook is used for the online OnSSET interface, recommended for small analyses and exploring code and results.

**PyCharm configuration and running the code**

Once you have downloaded all the files and saved them in a folder, open PyCharm.

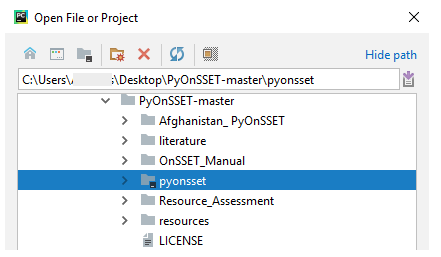
1. ****Select *File* > and then select the folder as your working directory as shown below (Fig. 1).

Fig. 1

1. Go to *File* > *Default Settings* > *Project Interpreter* and from the drop-down list select the Anaconda directory which we installed in the previous steps. Once all the packages load, click Apply and OK (Fig. 2). If Anaconda does not show up in the drop down list, click *Add local* to the right (Fig. 3) and browse to find the correct path. Make sure to select the *Inherit global site-packages* before clicking OK.

Fig. 2

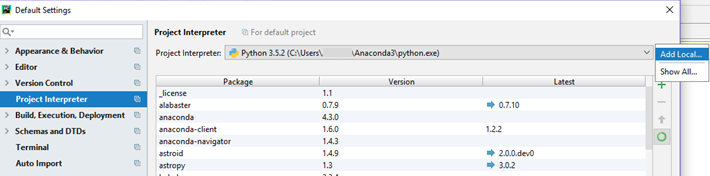
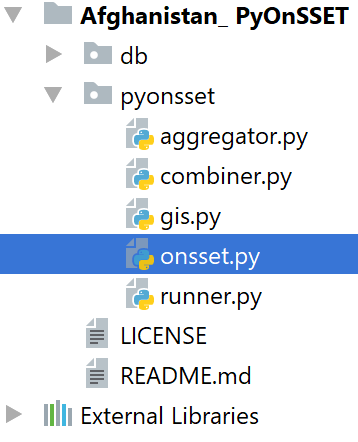


Fig. 3

1. Once you have set your working directory, from the drop-down menu in the *1: Project* tab to the left, double-click on a) onsset.py and b) runner.py in that order, in order to open them (Fig. 4).

Fig. 4



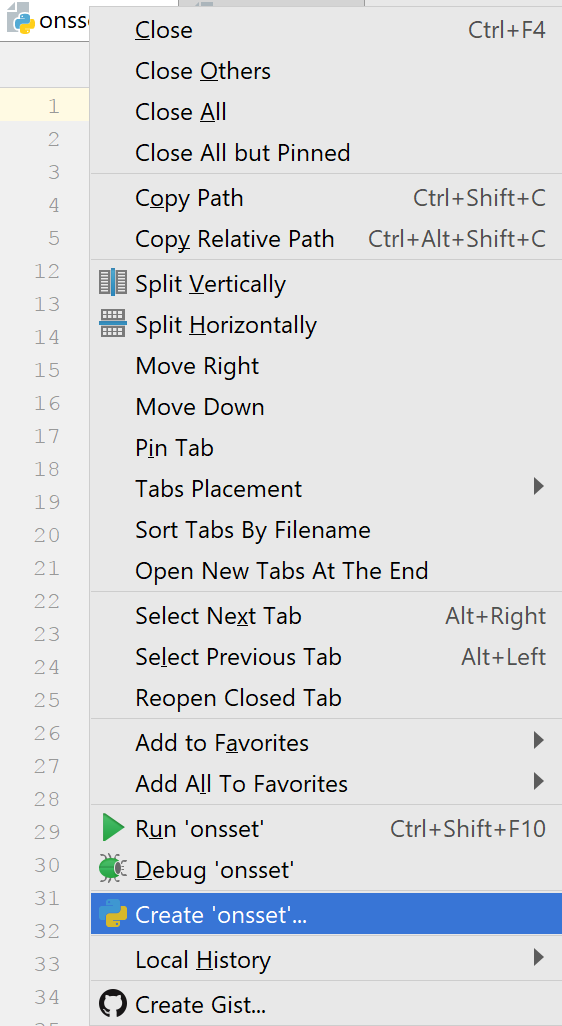
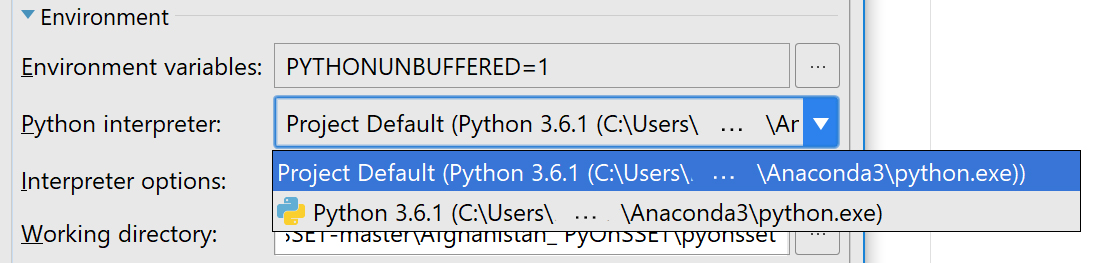
1. Now that you have opened “onsset.py”, right-click on its tab and click on “Create ‘onsset’…” (Fig. 5).
   1. In the configuration window, make sure that the Anaconda interpreter is selected in the “Python interpreter” option before you click OK (Fig. 6).
   2. You are ready to run “onsset.py” by clicking the Run button at the upper-right corner of your window (Fig. 7)
   3. Repeat the same steps explained above (4.1 and 4.2) for “runner.py” **only** if you get a “Process finished with exit code 0” message in the built-in console. Otherwise, move to step 5 before you run “runner.py”.

Fig. 6



Fig. 7

Fig. 5

5. After running a python script, the output is shown in the console built in PyCharm.

* + If you get a “Process finished with exit code 0” message, this means that everything worked OK.
  + Otherwise, the console shows messages with errors or warnings. In case you get a “ModuleNotFoundError” message, this means that a module or package is missing.

**Installing/updating modules and packages**

1) Using Anaconda

Search for “**Anaconda Prompt**” in your Windows Search. In the command line, type:

**conda install <package name>**

and the installation/update will go through automatically. (Note: In the Proceed ([y]/[n])? prompt, type “**y**”).

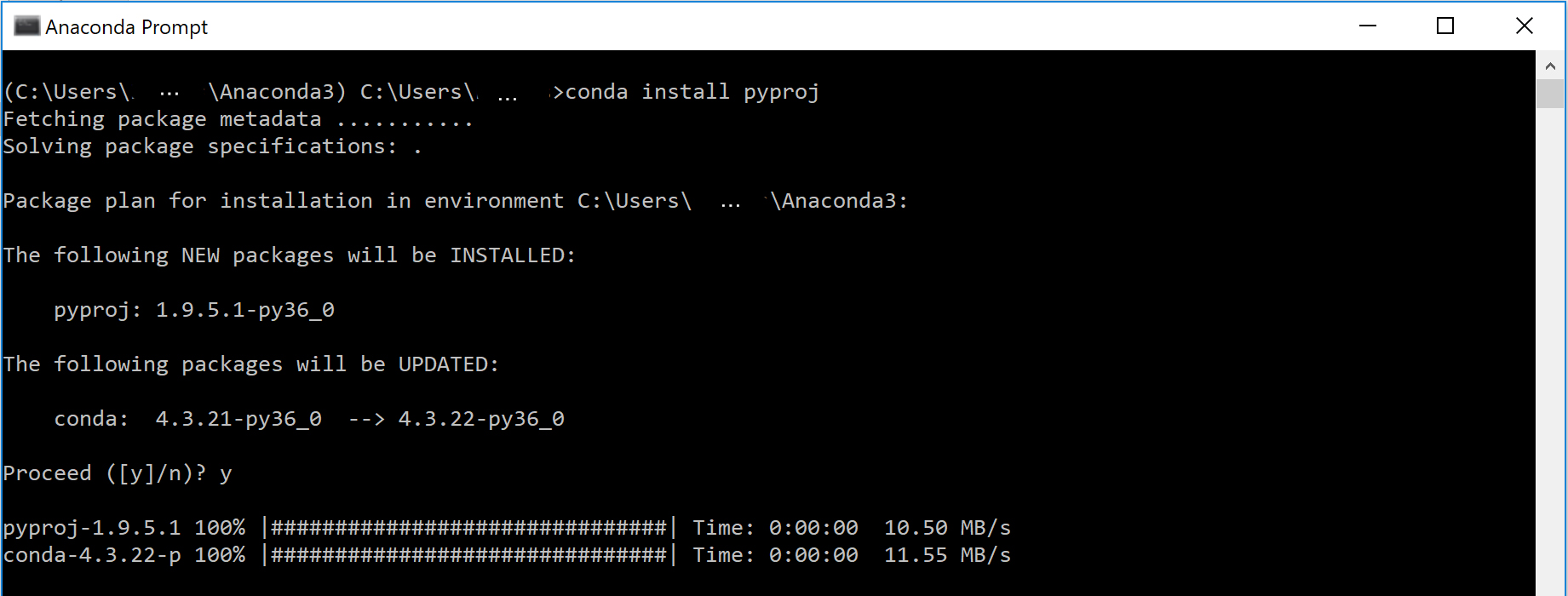
e.g. installation of module “**pyproj**” (Fig. 8)

Fig. 8

2) Using pip

**pip** is a package management system used to install and manage software packages written in Python. Search for “**Command Prompt**” in your Windows Search.Using the command line, navigate to your Python directory and then type “**pip**” and enter. This will show you all the possible choices you have within pip. In order to update pip itself you can just type (Fig. 9):

pip install --upgrade pip

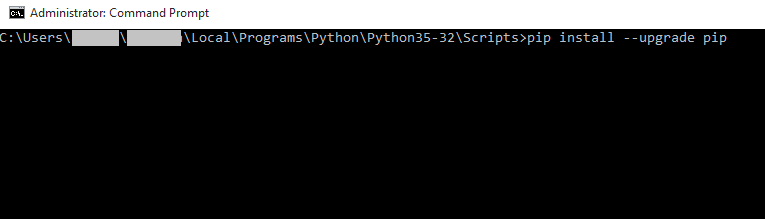
****

Fig. 9

In order to install any additional module required you can type:

**pip install <package name>**

e.g. installation of module “**matplotlib**” (Fig. 10)

**pip install matplotlib**

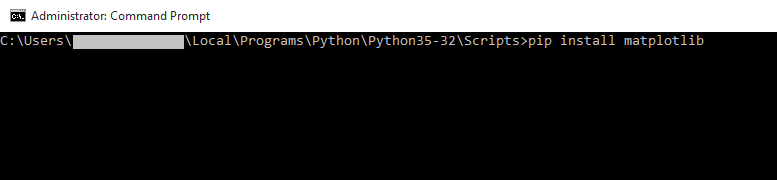


Fig. 10

6. Once you have successfully finished with the debugging process (module installation etc.), and while “**runner.py**” is running, you will be asked to type in the appropriate input, then press enter, so that you get your final results (Fig. 11).

Fig. 11

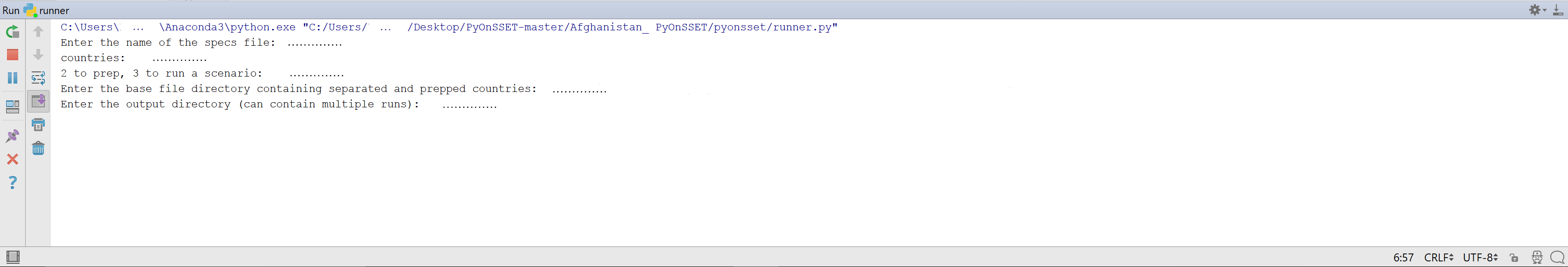


Fig. 10

1. When prompted to “Enter the directory of the specs file”, type the directory of the folder with all your files, e.g.” **~/Desktop/OnSSET**”.
2. When prompted to “**Enter the name of the specs file**”, type the name of the specs file, e.g. “**specs.xlsx**”
3. When prompted to enter “**Countries:**” type the name of the country, capitalized as in the country column of the csv-file, e.g. “**Togo**”
4. When prompted to “**Enter 1 to split, 2 to prep, 3 to run a scenario:**”, first enter “**2**” to prepare and calibrate the file.
5. When prompted to “**Enter the base file directory containing separated countries**”enter the directory of the pyonsset2018 folder, i.e. **~/Desktop/OnSSET**”, where the csv-file containing the extracted GIS data is located. This will prepare and calibrate the csv-file.
6. When prompted to “Enter the output directory for the calibrated file” enter the directory where you want to save the calibrated file, e.g. the new Calibrated folder at “**~/Desktop/OnSSET/Calibrated**”.
7. After successful preparation and calibration of the input data, the scenario runs can begin. Start by initializing *runner.py* again, by pressing the run button at the upper right corner of your window.
8. Repeat steps a) and b) with the same inputs.
9. When prompted to “**Enter 1 to split, 2 to prep, 3 to run a scenario:**”, this time enter “**3**” to run a scenario.
10. When prompted to “**Enter the base file directory containing separated and prepped countries:**” enter the directory of the pyonsset2018 folder with the calibrated file, i.e. **~/Desktop/OnSSET/Calibrated**”, where the csv-file containing prepared and calibrated GIS data is located.
11. When prompted to “**Enter the output directory (can contain multiple runs):**” enter the directory where the results should be saved, e.g. the *Results* folder as in “**~/Desktop/OnSSET/Calibrated /Results**”.
12. When prompted to “**Enter the tier number for urban:**” enter the tier number (a numerical from 1 to 5) that should be simulated in urban areas, e.g. “**4**”.
13. When prompted to “**Enter the tier number for rural:**” enter the tier number (a numerical from 1 to 5) that should be simulated in urban areas, e.g. “**2**”.
14. When prompted to “**Use high diesel value? <y/n>**” enter “**y**” to use the high diesel value in the specs-file, or enter “**n**” to use the low diesel value in the specs-file.
15. When prompted to “**Combine countries into a single file? <y/n>**” enter “**n**” (“y” is only used if running multiple countries/regions at once).

Steps c)-e) need only be run once unless a new csv-file with extracted GIS-data is provided, the initial urbanization or electrification rate is changed in the specs-file, or the code for the methods run in this step is modified. Steps e)-m) can be repeated multiple times to run scenarios.

**NOTES:**

Please make sure to:

* **Not** edit or modify in any way your input/output .csv files unless required.
* Close any open .csv or .xlsx files before running the programs.

**Additional Info**

* Basic navigating commands for DOS (cmd) can be found [here](https://community.sophos.com/kb/en-us/13195).
* [Modules](https://docs.python.org/3/installing/index.html) and [packages](https://packaging.python.org/tutorials/installing-packages/) installation documentation from [python.org](http://python.org).

1. Python itself can be downloaded and installed for free using the official website:

   <https://www.python.org/downloads/>. [↑](#footnote-ref-2)