Python installation guide for IUCAF SMS2025

B.Winkel, F. Giovanardi, M.Bautista

29 Sept – 3 Oct 2025

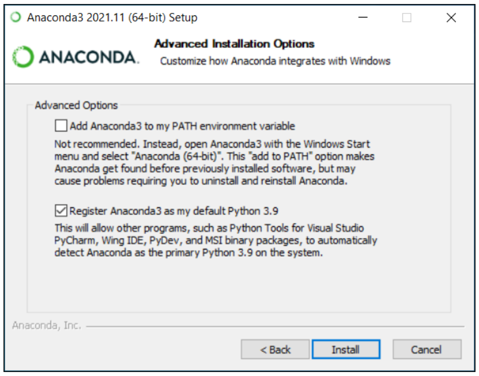
Here's a beginner-friendly guide to managing and installing Python on your computer. It provides step-by-step instructions to help you get started with Python.
pycraf documentation: https://bwinkel.github.io/pycraf/

1 Download Anaconda

Go to web page and download the Anaconda https://www.anaconda.com/download. Anaconda is a Python distribution for scientific computing that aims to simplify package management and deployment.

2 Install Anaconda

Installation instructions can be found at https://docs.anaconda.com/free/anaconda/install/.
Select "Register Anaconda as my default Python 3.x"



Anaconda3 2021.11 (64-bit) Setup

Advanced Installation Options
Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda3 to my PATH environment variable

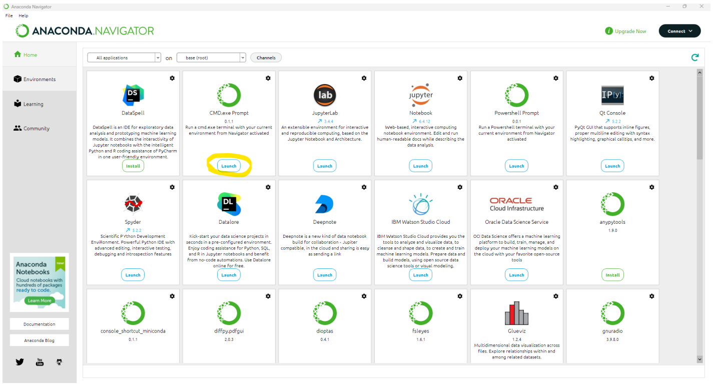
Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda3 as my default Python 3.9

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.

3 Open Anaconda prompt (Anaconda PowerShell)

1. To open Anaconda Prompt (2 methods):
   - Windows: Click Start, search for Anaconda Prompt, and click to open.
   - macOS: Use Cmd+Space to open Spotlight Search and type "Navigator" to open the program. Or simply open a terminal.
   - Linux–CentOS: Open Applications > System Tools > terminal.
   - Linux–Ubuntu: Open the Dash by clicking the Ubuntu icon, then type "terminal".
2. If you didn't find the anaconda prompt, you can open Anaconda Navigator and click on "CMD.exe Prompt" as screenshot below.



DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

CMD.exe Prompt 0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

JupyterLab 3.4.4
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook 6.4.12
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt 0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console 5.2.2
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder 5.2.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Datalore
Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.

Deepnote
Deepnote is a new kind of data notebook build for collaboration - Jupiter compatible, in the cloud and sharing is easy as sending a link

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

ORACLE Cloud Infrastructure
Oracle Data Science Service
OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools

anypytools 1.9.0

console_shortcut_miniconda 0.1.1

diffpy.pdfgui 2.0.3

dioptas 0.4.1

fsleyes 1.6.1

Glueviz 1.2.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.

gnuradio 3.9.8.0

4 The conda package manager

Anaconda comes with a really nice package manager, called “conda” (which is the main reason why using Anaconda is really easy). With “conda” one can install the Python packages which one desires. But it can do even more. To keep the installation clean and avoid too complicated setups, it is recommended to organize the work into so-call “environments”. *A conda environment is a standalone workspace that contains a specific set of packages and their dependencies, allowing for isolated and reproducible software environments.* For further details: https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html Hence, each environment can have its own Python version and different packages (or different versions of packages).
Create a new environment (for example “python-env”) setting the Python version to 3.10 (type in the prompt):

```
conda create -n python-env python=3.10
```

Before you can work with the new environment, you have to activate it:

```
conda activate python-env
```

You can always install (or update) individual packages in your environment (assuming it is currently activated)

```
conda install -c conda-forge PackageName
```

It is also possible to have packages installed while creating an environment:

```
conda create -n python-env python=3.10 matplotlib numpy scipy
```

Deleting an environment is also easy:

```
conda remove -n python-env --all
```

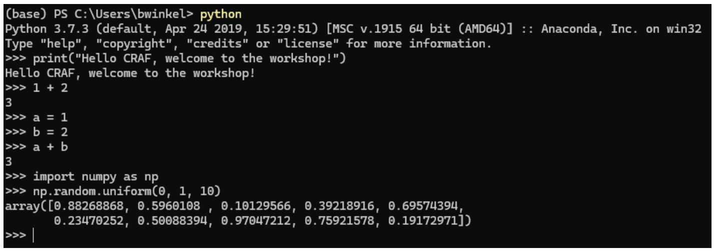
You can now simply start Python and do stuff:

```
python
```

```
>>> print("Hello CRAF")
```

If you want to go back into the base environment, simply type

```
conda deactivate
```



```
(base) PS C:\Users\bwinkel> python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello CRAF, welcome to the workshop!")
Hello CRAF, welcome to the workshop!
>>> 1 + 2
3
>>> a = 1
>>> b = 2
>>> a + b
3
>>> import numpy as np
>>> np.random.uniform(0, 1, 10)
array([0.88268868, 0.5960108 , 0.10129566, 0.39218916, 0.69574394,
       0.23470252, 0.50088394, 0.97047212, 0.75921578, 0.19172971])
>>> |
```

Figure 1: Python interactive shell

5 Installing pycraf

The following command will install pycraf along with necessary packages in one go[1]:

```
conda create --yes -c conda-forge -n pycraf-env python=3.10 "astropy>=5" \
    cysgp4 cython h5py ipdb ipykernel ipython ipywidgets "matplotlib>=3.3" \
    numpy pycraf "pyproj>=2.2" pytest pytest-astropy pytest-remotedata \
    "scipy<1.10" "sgp4>2"
```

To test pycraf, do the following:

```
conda activate pycraf-env
python
```

```
>>> import pycraf
>>> pycraf.test(remote_data='all')
```

6 Jupyter

During the pycraf workshop, we will be working with Jupyter, which allows to create "Notebooks" containing text (in Markdown), images, Python code and results in integrated form. It is recommended to install Jupyter into its own conda environment:

```
conda create --yes -c conda-forge -n jupyter-env python=3.10 ipdb ipywidgets \
    ipython jupyterlab matplotlib nb_conda_kernels nodejs numpy pandas scipy
```

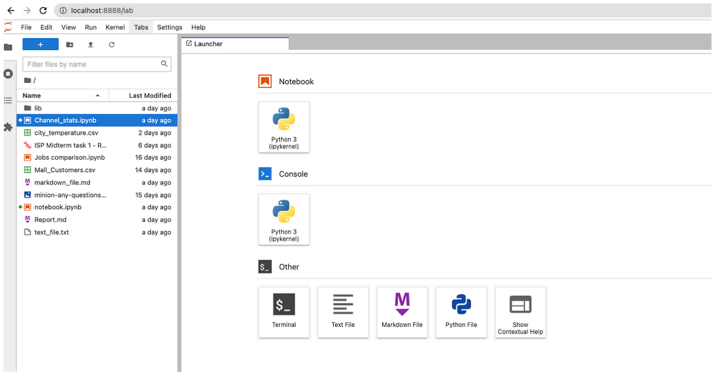
To launch Jupyter; simply activate the environment and start it via

```
conda activate jupyter-env
jupyter lab
```

Your browser will open the Jupyter interface. In each (new) Notebook, you can specify the "kernel" (aka environment) that you want to use. In our case, we will be choosing the "pycraf-env" environment. You can also open the terminal through the Jupyter interface (File > New > Terminal). Before installing any packages, make sure the right environment is activated.
In rare cases, Jupyter is not automatically showing the installed conda environments (i.e., "pycraf-env"). In this case, one can register it for use with Jupyter:

```
conda activate pycraf-env
python -m ipykernel install --user --name pycraf-env \
    --display-name "Python 3.10 (env: pycraf-env)"
```

[1] On Windows, you will have to remove the "\" from the line endings and put everything on one line!



Name	Last Modified
lib	a day ago
Channel_stats.ipynb	a day ago
city_temperature.csv	2 days ago
ISP Midterm task 1 - R...	6 days ago
Jobs comparison.ipynb	16 days ago
Mall_Customers.csv	14 days ago
markdown_file.md	a day ago
minion-any-questions...	15 days ago
notebook.ipynb	a day ago
Report.md	a day ago
text_file.txt	a day ago

Figure 2: Jupyter in the web browser

7 Working with terrain height data

To make full use of the path attenuation calculations provided by pycraf one needs terrain height data. The easiest way to obtain this is to download data produce by NASA's Shuttle Radar Topography Mission (SRTM). A good place to get them is `http://viewfinderpanoramas.org/Coverage%20map%20viewfinderpanoramas_org3.htm`
Don't download everything, as this would need a lot of disk space! The tiles around your observatory suffice (perhaps Europe, if disk space is not an issue). Store the downloaded files in a dedicated folder on your machine for convenience.
The best way to let pycraf know where the tiles are located is to set an appropriate environment variable on your system.
On Windows use

```
set SRTMDATA=C:\\[path-to-srtm]\\
```

On Linux:

```
export SRTMDATA=[path-to-srtm]/
```

This will only apply for the current shell in which you typed the command. It can be made permanent, of course. On Linux, you could add it to your shell config file (e.g., ".bashrc"). On Windows 7+, you could click on the start menu, type "env" and open the dialog that allows to edit your user environment settings.

8 Pycraf documentation

If you want to take a look in more detail about pycraf, please visit the following links:

- **GitHub repository:** `https://github.com/bwinkel/pycraf` . The code and some tutorial/examples are provided in this GitHub repository. You can download everything on your computer.
- **Pycraf documentation:** `https://bwinkel.github.io/pycraf/` . In this webpage, you can find all the information related with pycraf, for example, how to install pycraf or different sections for each sub-packages (e.g. pathprof, antenna pattern, etc) with some code examples and explanation. Furthermore, some notebooks are provided as an example of how to cover the different parts of pycraf (`https://nbviewer.org/github/bwinkel/pycraf/tree/master/notebooks/`).

9 Tips for advanced users

9.1 Complete installation

If you want everything that is remotely relevant for compatibility studies, do install the following:

```
conda create --yes -c conda-forge -n pycraf-env python=3.10 "astropy>=5" \
    cartopy cysgp4 cython fiona geopandas h5py ipdb ipykernel ipython \
    ipywidgets "matplotlib>=3.3" numpy openpyxl osmnx pandas pycraf \
    "pyproj>=2.2" pytest pytest-astropy pytest-remotedata rasterio \
    reproject "scipy<1.10" "sgp4>2" shapely
```

9.2 Pycraf Graphical User Interface

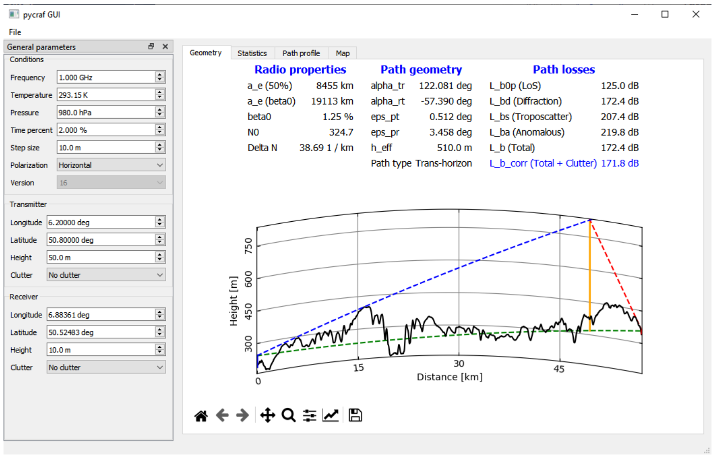
There is also a graphical user interface for “pycraf” (with some very limited functionality). If you want to try it, use the following installation command instead:

```
conda create --yes -c conda-forge -n pycraf-env python=3.10 "astropy>=5" \
    cysgp4 cython h5py ipdb ipykernel ipython ipywidgets "matplotlib>=3.3" \
    numpy pycraf pycraf-gui "pyproj>=2.2" "pyqt>=5.11" pytest pytest-astropy \
    pytest-qt pytest-remotedata "scipy<1.10" "sgp4>2"
```

Note that on MacOS you might have difficulties with the Qt package and dependencies.
The GUI can be started in the command line:

```
conda activate pycraf-env
pycraf-gui
```

Note that you may need to set an environment variable for terrain height data (e.g. SRTM) for best experience; see below.



Conditions

Frequency 1.000 GHz
Temperature 293.15 K
Pressure 980.0 hPa
Time percent 2.000 %
Step size 10.0 m
Polarization Horizontal
Version 16

Transmitter

Longitude 6.20000 deg
Latitude 50.80000 deg
Height 50.0 m
Clutter No clutter

Receiver

Longitude 6.88361 deg
Latitude 50.52483 deg
Height 10.0 m
Clutter No clutter

Radio properties		Path geometry		Path losses	
a_e (50%)	8455 km	alpha_tr	122.081 deg	L_b0p (LoS)	125.0 dB
a_e (beta0)	19113 km	alpha_rt	-57.390 deg	L_bd (Diffraction)	172.4 dB
beta0	1.25 %	eps_pt	0.512 deg	L_bs (Troposcatter)	207.4 dB
N0	324.7	eps_pr	3.458 deg	L_ba (Anomalous)	219.8 dB
Delta N	38.69 1 / km	h_eff	510.0 m	L_b (Total)	172.4 dB
		Path type	Trans-horizon	L_b_corr (Total + Clutter)	171.8 dB

Figure 3: Pycraf GUI

9.3 Check installed packages and versions

To check all packages previously installed by conda, e.g.

```
conda activate pycraf-env
conda list
```

The packages, which were installed, include:

- numpy: fast array computations
- scipy: for scientific computing and technical computing.
- astropy: astronomical functions; mostly used for physical units support

- matplotlib: for plotting
- pyproj: for the geographical coordinate frames and transformations
- cysgp4 and sgp4: for the satellite orbit calculations
- rasterio: for geotiff reading
- h5py: for caching.

9.4 A faster package manager: mamba

The conda package manager can be slow if the environments are complex (many packages with complicated dependencies). Feel free to try the drop-in replacement "mamba", which can easily be installed with conda. (Needs to be present in each environment, unfortunately.)

9.5 Interactive shell

If you like to work on the command line, try the great interactive python shell, "IPython".

9.6 Text editors and IDEs

For serious programming, you may want to use a good text editor with Python support (e.g., for syntax highlighting and grammar checking), such as Visual Studio Code, Vim, Sublime etc. There are also a few nice integrated development environments (IDEs), e.g., Spyder (for beginners) or PyCharm (for professionals).