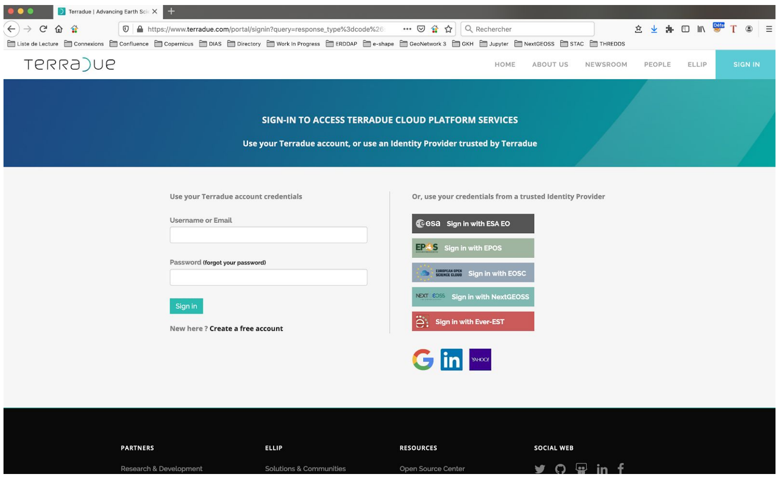
NextGEOSS Solar Energy Pilot

Access to time-series of gridded data from CAMS Radiation (2005-2006 subset)

Authentication

Please follow the next steps in order to access the Jupyter Notebook home page.

1. Go to: https://notebooks.terradue.com (You'll be automatically redirected to the following page):



SIGN-IN TO ACCESS TERRADUE CLOUD PLATFORM SERVICES

Use your Terradue account, or use an Identity Provider trusted by Terradue

Use your Terradue account credentials

Username or Email

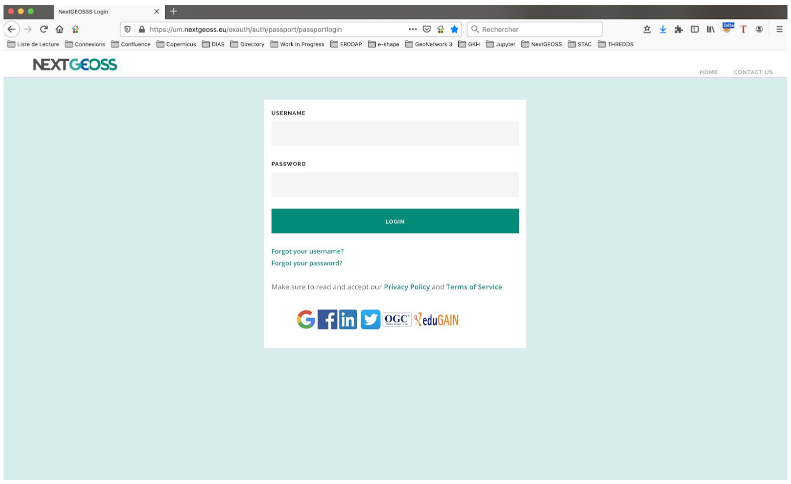
Password (forgot your password)

Sign in

New here ? Create a free account

Or, use your credentials from a trusted Identity Provider

2. Click on the "Sign in with NextGEOSS" button
3. Authenticate in the following page with the login/password credentials you have been provided with:



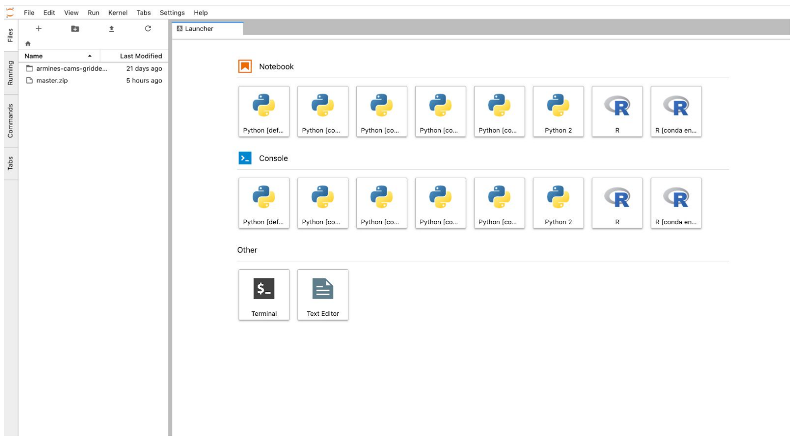
USERNAME

PASSWORD

Forgot your username?
Forgot your password?

Make sure to read and accept our Privacy Policy and Terms of Service

4. Congrat's you are now in your Jupyter Notebook home page !

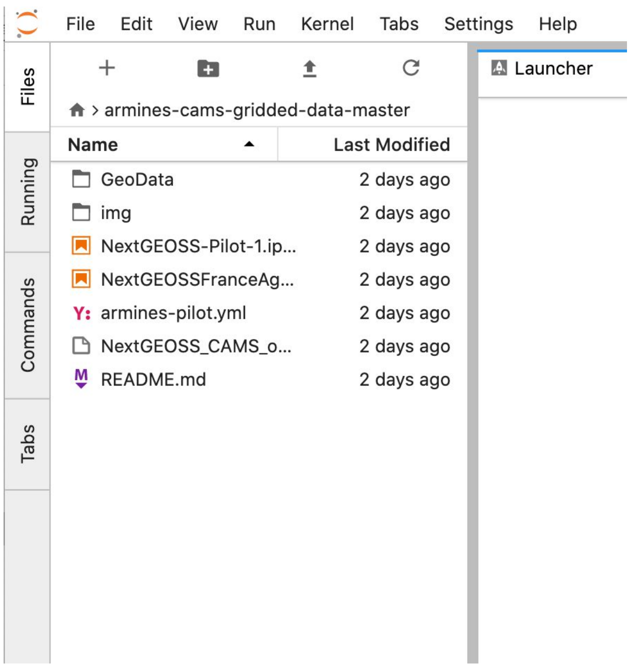


Name	Last Modified
armines-cams-gridde...	21 days ago
master.zip	5 hours ago

Jupyter Notebook

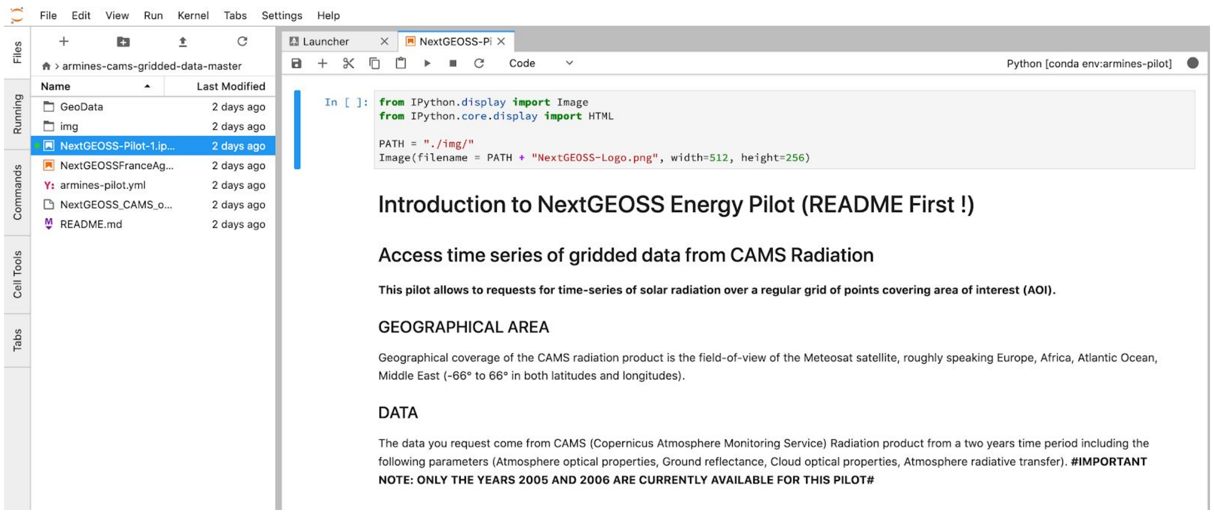
Now, you should be logged-in on your Jupyter Notebook work environment.

- Double click the “armines-cams-gridded-data-master” folder in the left side of the window. You should be able to see the following files:



Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	2 days ago
NextGEOSSFranceAg...	2 days ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago

- Double click on “NextGEOSS-Pilot-1.ipynb”. You should see the Jupyter Notebook as follows.



Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	2 days ago
NextGEOSSFranceAg...	2 days ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago

```
In [ ]: from IPython.display import Image
        from IPython.core.display import HTML

        PATH = "./img/"
        Image(filename = PATH + "NextGEOSS-Logo.png", width=512, height=256)
```

Introduction to NextGEOSS Energy Pilot (README First !)

Access time series of gridded data from CAMS Radiation

This pilot allows to requests for time-series of solar radiation over a regular grid of points covering area of interest (AOI).

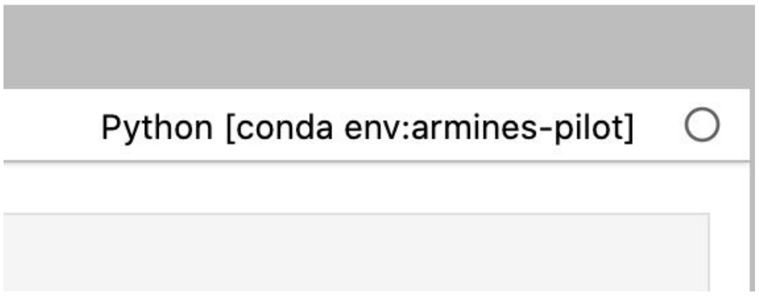
GEOGRAPHICAL AREA

Geographical coverage of the CAMS radiation product is the field-of-view of the Meteosat satellite, roughly speaking Europe, Africa, Atlantic Ocean, Middle East (-66° to 66° in both latitudes and longitudes).

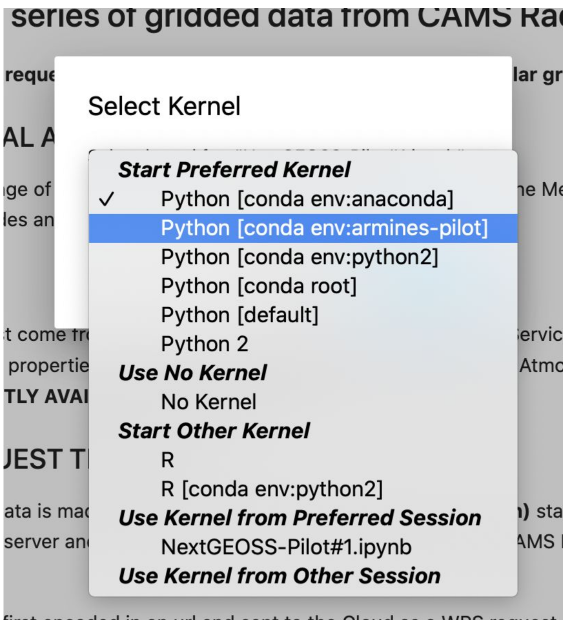
DATA

The data you request come from CAMS (Copernicus Atmosphere Monitoring Service) Radiation product from a two years time period including the following parameters (Atmosphere optical properties, Ground reflectance, Cloud optical properties, Atmosphere radiative transfer). **#IMPORTANT NOTE: ONLY THE YEARS 2005 AND 2006 ARE CURRENTLY AVAILABLE FOR THIS PILOT#**

- You should make sure that the “armines-pilot” kernel is selected on the top right of the main window such as follows.



- If it is not the case, click on “Python [conda env:armines-pilots]” and select it among the available kernel’s list:

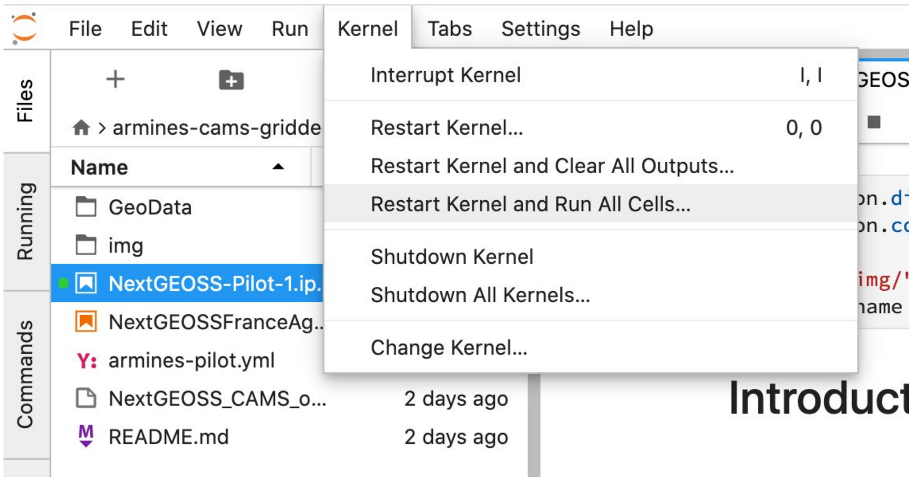


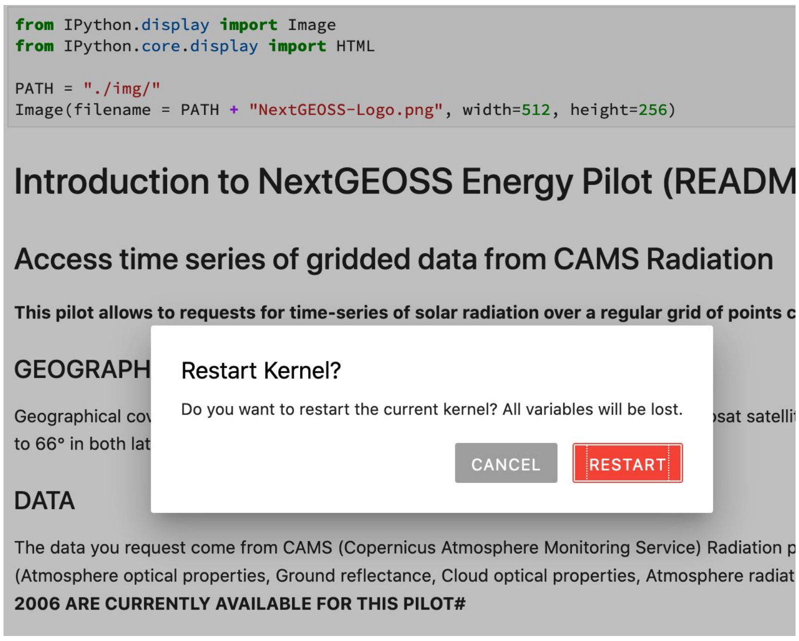
- Please read the “Introduction to NextGEOSS Energy Pilot (README First !)” section for a general understanding of the application.
- You can as well have a look at the code using the scrolling bar on the main window.

Run the application

Forewords: The initial run of the script described below may take some time. A rough estimation is that it will take around 3minutes to complete. Below is the step-by-step process to run the script and additional information of the various features of the python code.

- The application comes with a default set of parameters to launch the access and the retrieval of a time series of solar irradiance over an area of interest (AOI). These default parameters are set to provide you a time series covering France for one day every 15 mn (96 layers).
- As a first start we suggest that you run the application once before doing any change of the above parameters.
- To do so, Go to the “Kernel” menu on the top of the window and select “Restart Kernel and Run All Cells…” and then click on the red “Restart…” button such as follows.





```
from IPython.display import Image
from IPython.core.display import HTML

PATH = "./img/"
Image(filename = PATH + "NextGEOSS-Logo.png", width=512, height=256)
```

Introduction to NextGEOSS Energy Pilot (READM

Access time series of gridded data from CAMS Radiation

This pilot allows to requests for time-series of solar radiation over a regular grid of points c

GEOGRAPH

Geographical cov … sat satelli
to 66° in both lat

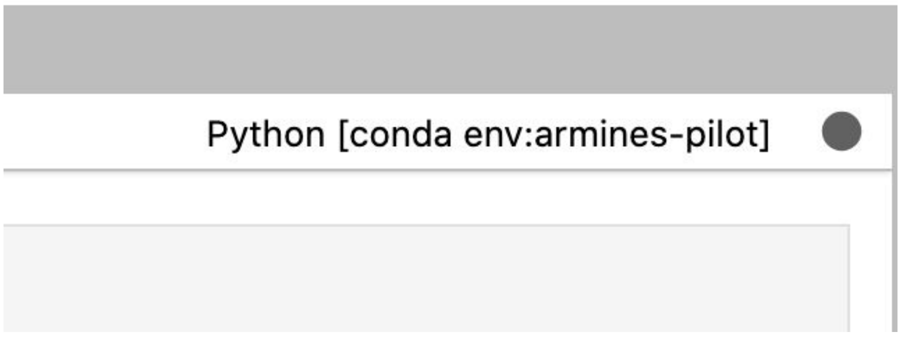
Restart Kernel?

Do you want to restart the current kernel? All variables will be lost.

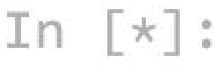
DATA

The data you request come from CAMS (Copernicus Atmosphere Monitoring Service) Radiation p
(Atmosphere optical properties, Ground reflectance, Cloud optical properties, Atmosphere radiat
2006 ARE CURRENTLY AVAILABLE FOR THIS PILOT#

- From there, the process has started. You can see that at the right of “Python [conda env:armines-pilots] the white point has turned from white to to grey. The script is running….



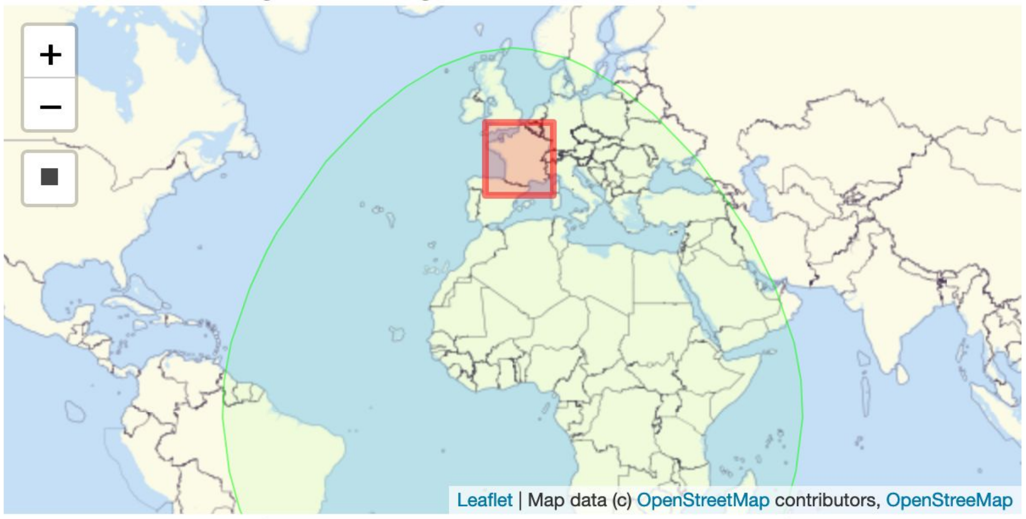
- Another indicator that the script is running is that the cells containing the python script are flagged with an asterix while processing,such as follows.



- You can see that just after the beginning of the script’s run a section of the python code is transformed into a nice graphical user interface (GUI). This GUI will allow you to later select your AOI, width height, time interval and time frame parameters. The green polygon represents the CAMS Radiation data coverage following Meteosat

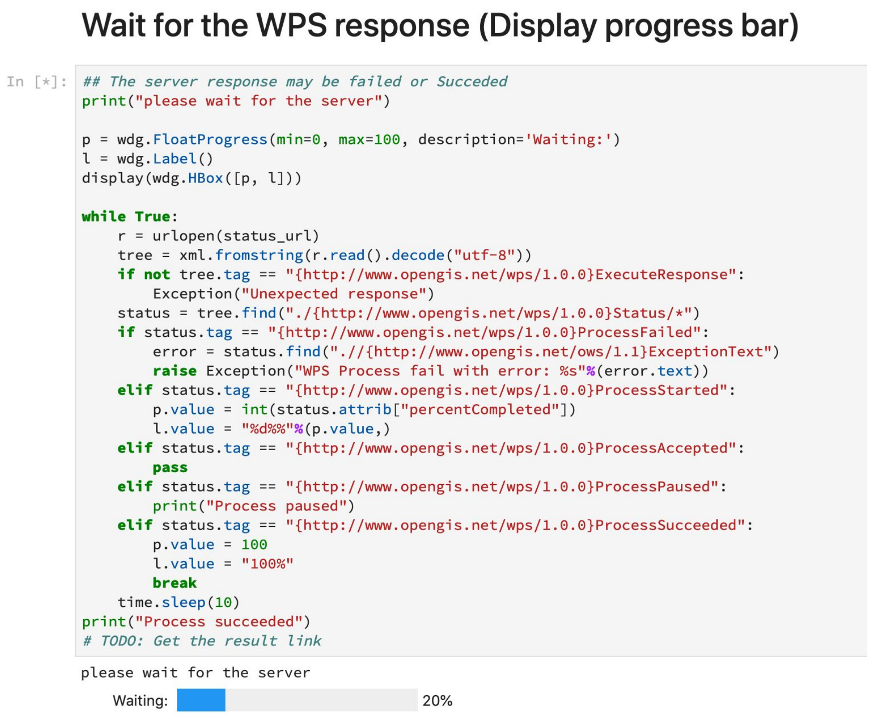
Second Generation field of view. It should look like this.

```
Select an area using the rectangle tool
```


+
−
Leaflet | Map data (c) OpenStreetMap contributors, OpenStreeMap

Output width	300	Longitude at left	-5.5
Output height	300	Longitude at right	8.5
First instant of the tim...	2005-06-22 00:15	Latitude at top	51.3
Interval between insta...	15	Latitude at bottom	41.0
Number of instant	96		

- Scrolling down the notebook application, you'll see that the remote process is launched and monitored such as follows…

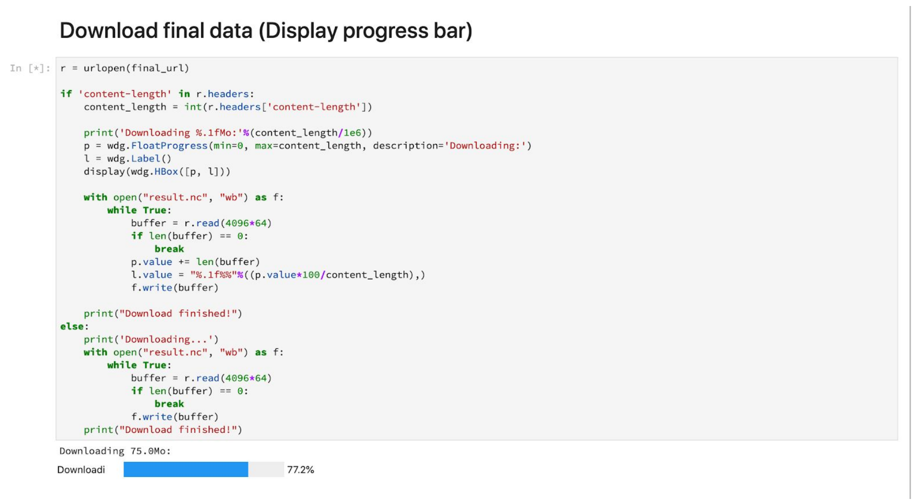


Wait for the WPS response (Display progress bar)

```
In [*]: ## The server response may be failed or Succeded
        print("please wait for the server")

        p = wdg.FloatProgress(min=0, max=100, description='Waiting:')
        l = wdg.Label()
        display(wdg.HBox([p, l]))

        while True:
            r = urlopen(status_url)
            tree = xml.fromstring(r.read().decode("utf-8"))
            if not tree.tag == "{http://www.opengis.net/wps/1.0.0}ExecuteResponse":
                Exception("Unexpected response")
            status = tree.find("./{http://www.opengis.net/wps/1.0.0}Status/*")
            if status.tag == "{http://www.opengis.net/wps/1.0.0}ProcessFailed":
                error = status.find(".//{http://www.opengis.net/ows/1.1}ExceptionText")
                raise Exception("WPS Process fail with error: %s"%(error.text))
            elif status.tag == "{http://www.opengis.net/wps/1.0.0}ProcessStarted":
                p.value = int(status.attrib["percentCompleted"])
                l.value = "%d%%"%(p.value,)
            elif status.tag == "{http://www.opengis.net/wps/1.0.0}ProcessAccepted":
                pass
            elif status.tag == "{http://www.opengis.net/wps/1.0.0}ProcessPaused":
                print("Process paused")
            elif status.tag == "{http://www.opengis.net/wps/1.0.0}ProcessSucceeded":
                p.value = 100
                l.value = "100%"
                break
            time.sleep(10)
        print("Process succeeded")
        # TODO: Get the result link
```

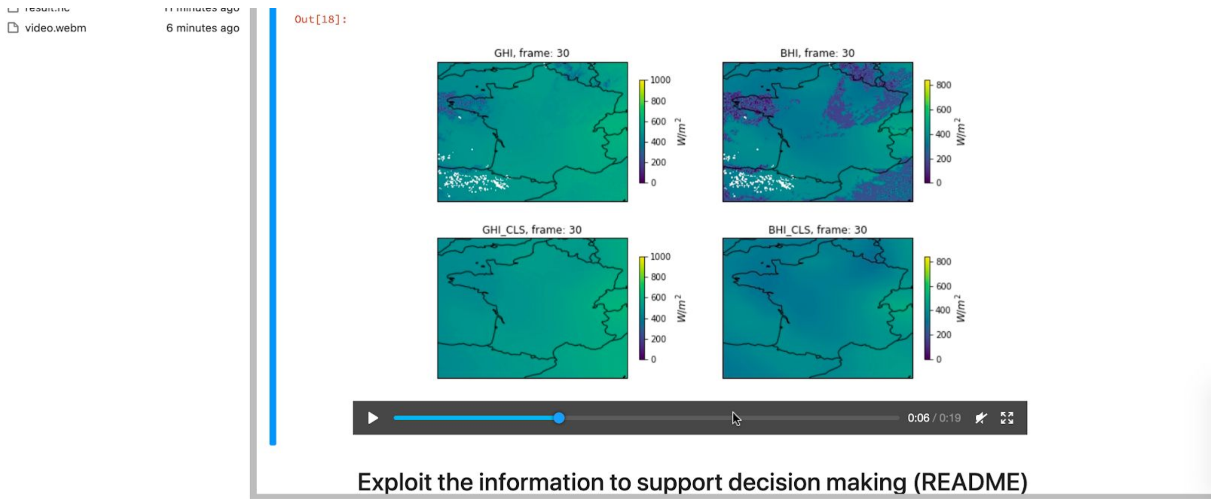
```
please wait for the server
  Waiting:                20%
```

- When 100% is reached the result file download process starts from the remote server to your Jupyter Notebook environment. It is monitored as follows…

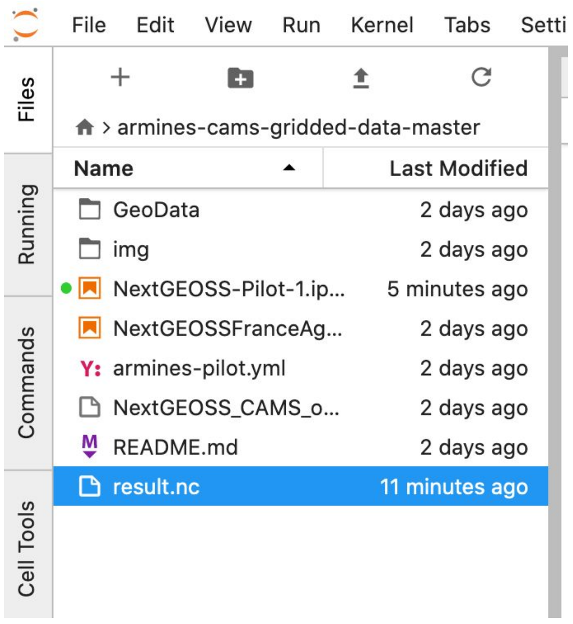

Download final data (Display progress bar)

```
In [*]: r = urlopen(final_url)

        if 'content-length' in r.headers:
            content_length = int(r.headers['content-length'])

            print('Downloading %.1fMo:'%(content_length/1e6))
            p = wdg.FloatProgress(min=0, max=content_length, description='Downloading:')
            l = wdg.Label()
            display(wdg.HBox([p, l]))

            with open("result.nc", "wb") as f:
                while True:
                    buffer = r.read(4096*64)
                    if len(buffer) == 0:
                        break
                    p.value += len(buffer)
                    l.value = "%.1f%%"%((p.value*100/content_length),)
                    f.write(buffer)

            print("Download finished!")
        else:
            print('Downloading...')
            with open("result.nc", "wb") as f:
                while True:
                    buffer = r.read(4096*64)
                    if len(buffer) == 0:
                        break
                    f.write(buffer)
            print("Download finished!")

Downloading 75.0Mo:
Downloadi                     77.2%
```

- When the download is finished, you are provided with a quick view of your result as a form of an animation of the time-series that you've selected. Click on the arrow to start the animation such as follows…

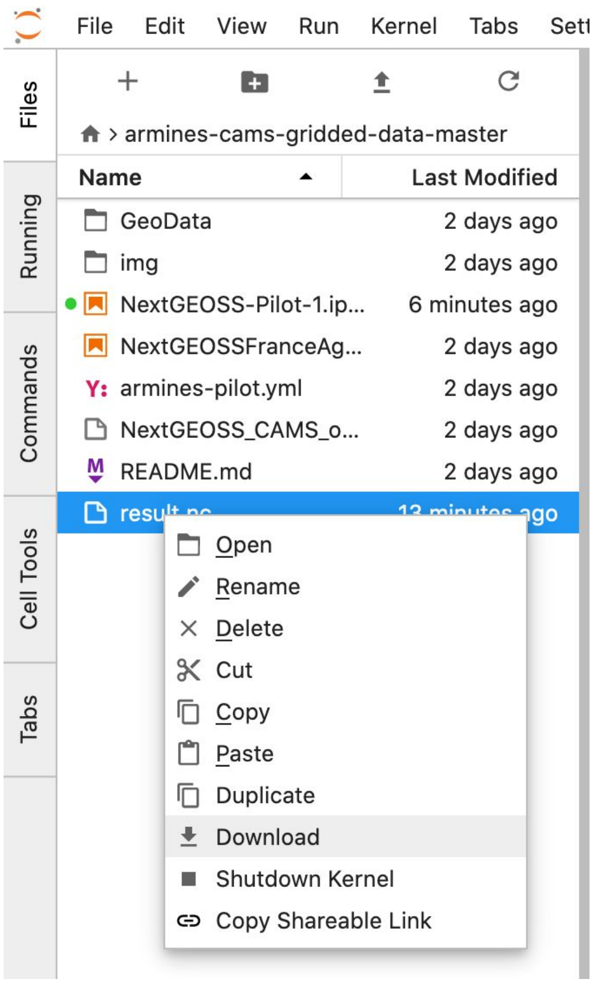


- In our default case this is an animation over 1 day (2005-06-22) every 15 mn. resulting in a file (NetCDF format) containing 96 layers.
- The file called "result.nc" is now visible in your environment folder such as follow…



Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip…	5 minutes ago
NextGEOSSFranceAg…	2 days ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o…	2 days ago
README.md	2 days ago
result.nc	11 minutes ago

- From now on, you can decide to download the file to use it for your own purpose. Simply right click on the "result.nc" file and select the "Download" option…

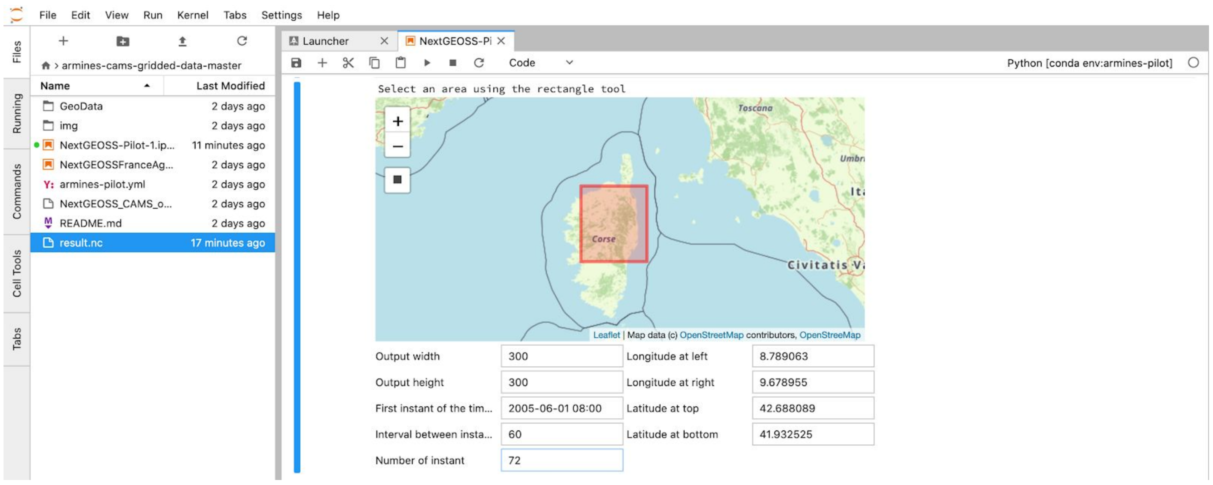


Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	6 minutes ago
NextGEOSSFranceAg...	2 days ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	13 minutes ago

Select you own parameters

Now you can select your own AOI and time parameters.

- Go to the GUI and change the parameters according to your need.
- The inputs parameters are as follow:
  - Output width and height are the size sampling in pixel of the final result file
  - First instant stand for the beginning of the requested time-series
  - Interval is the time rate sampling starting from 15 mn. up to your choice
  - Number of instant is the repeating time of the above interval
- In the example below I select a new AOI with the same output width and eight from June 1st at 8:00 AM with an hourly interval for 3 days.
- Below is an example of such changes…

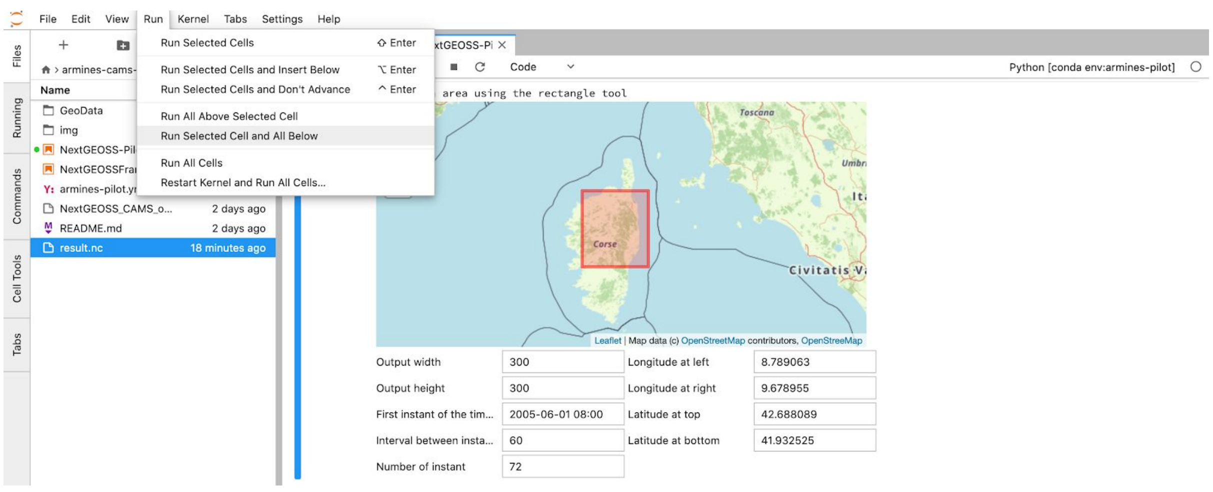

File
Edit
View
Run
Kernel
Tabs
Settings
Help
Files
Running
Commands
Cell Tools
Tabs
armines-cams-gridded-data-master
Launcher
NextGEOSS-Pi
Code
Python [conda env:armines-pilot]
Toscana
Umbri
Corse
Civitatis V
Leaflet | Map data (c) OpenStreetMap contributors, OpenStreeMap

Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	11 minutes ago
NextGEOSSFranceAg...	2 days ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	17 minutes ago

```
Select an area using the rectangle tool
```

Output width	300	Longitude at left	8.789063
Output height	300	Longitude at right	9.678955
First instant of the tim...	2005-06-01 08:00	Latitude at top	42.688089
Interval between insta...	60	Latitude at bottom	41.932525
Number of instant	72		

- When ready, go to the “Run” top menu and select “Run Selected Cell and All Below”. This will take your inputs and send them to the processor for extracting the data based on the AOI and selected parameters of your choice.



```
area using the rectangle tool
```

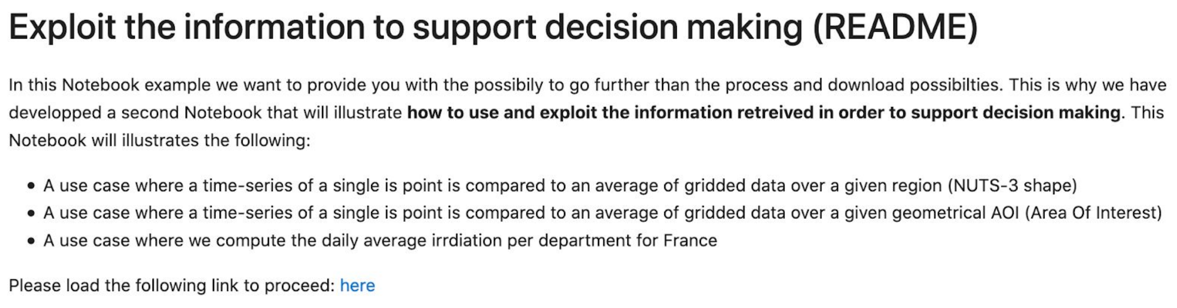
Output width	300	Longitude at left	8.789063
Output height	300	Longitude at right	9.678955
First instant of the tim...	2005-06-01 08:00	Latitude at top	42.688089
Interval between insta...	60	Latitude at bottom	41.932525
Number of instant	72		

- The process time and download bar graph will indicate the status of each process such as for the initial run and when ready a new animation will be generated based on the new “result.nc” file such as follows…

Next step

To illustrate the possibilities of the Jupyter Notebook framework, we have created another Notebook that provides a simple, yet illustrative example of the added value of having access to time series of irradiation over a regular grid. It demonstrates the lack of representativeness of pointwise vs AOI solar resource assessment for an extended region such as the NUTS 3 region and for free polygonal selection.

- You can access this Notebook by clicking on the link at the bottom of the page as shown below or directly double-click on the “NextGEOSSFranceAggregation” link in the left menu.



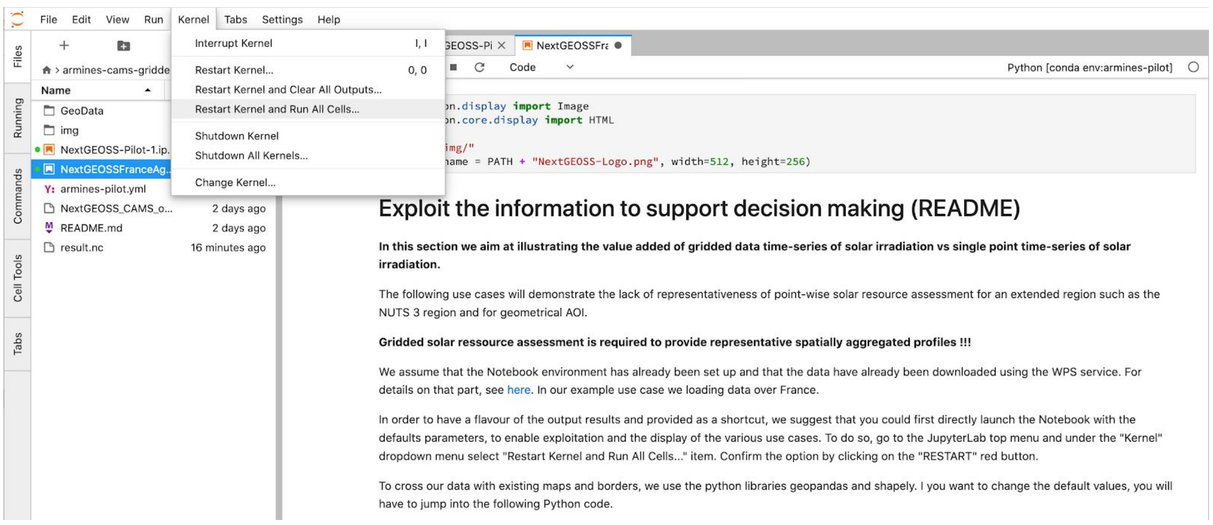
Exploit the information to support decision making (README)

In this Notebook example we want to provide you with the possibily to go further than the process and download possibilties. This is why we have developped a second Notebook that will illustrate **how to use and exploit the information retreived in order to support decision making**. This Notebook will illustrates the following:

- A use case where a time-series of a single is point is compared to an average of gridded data over a given region (NUTS-3 shape)
- A use case where a time-series of a single is point is compared to an average of gridded data over a given geometrical AOI (Area Of Interest)
- A use case where we compute the daily average irrdiation per department for France

Please load the following link to proceed: here

- As for the previous Notebook, run once the script by selecting “Restart the Kernel and Run All Cells” such as follows…



```
on.display import Image
on.core.display import HTML

img/"
name = PATH + "NextGEOSS-Logo.png", width=512, height=256)
```

Exploit the information to support decision making (README)

In this section we aim at illustrating the value added of gridded data time-series of solar irradiation vs single point time-series of solar irradiation.

The following use cases will demonstrate the lack of representativeness of point-wise solar resource assessment for an extended region such as the NUTS 3 region and for geometrical AOI.

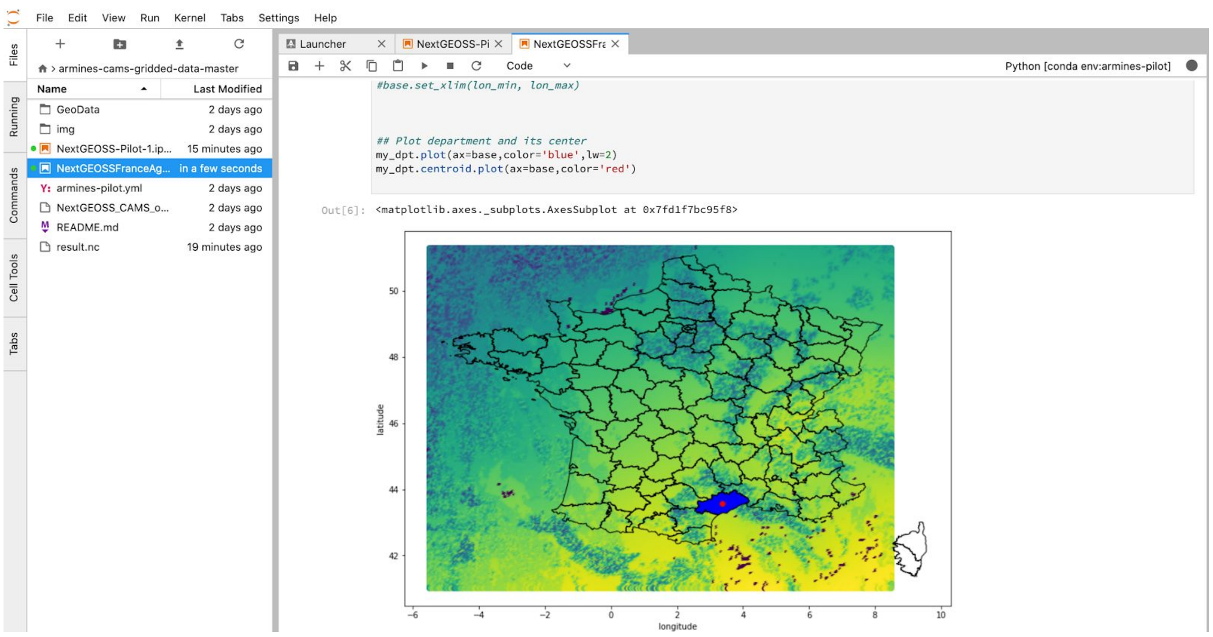
Gridded solar ressource assessment is required to provide representative spatially aggregated profiles !!!

We assume that the Notebook environment has already been set up and that the data have already been downloaded using the WPS service. For details on that part, see here. In our example use case we loading data over France.

In order to have a flavour of the output results and provided as a shortcut, we suggest that you could first directly launch the Notebook with the defaults parameters, to enable exploitation and the display of the various use cases. To do so, go to the JupyterLab top menu and under the "Kernel" dropdown menu select "Restart Kernel and Run All Cells..." item. Confirm the option by clicking on the "RESTART" red button.

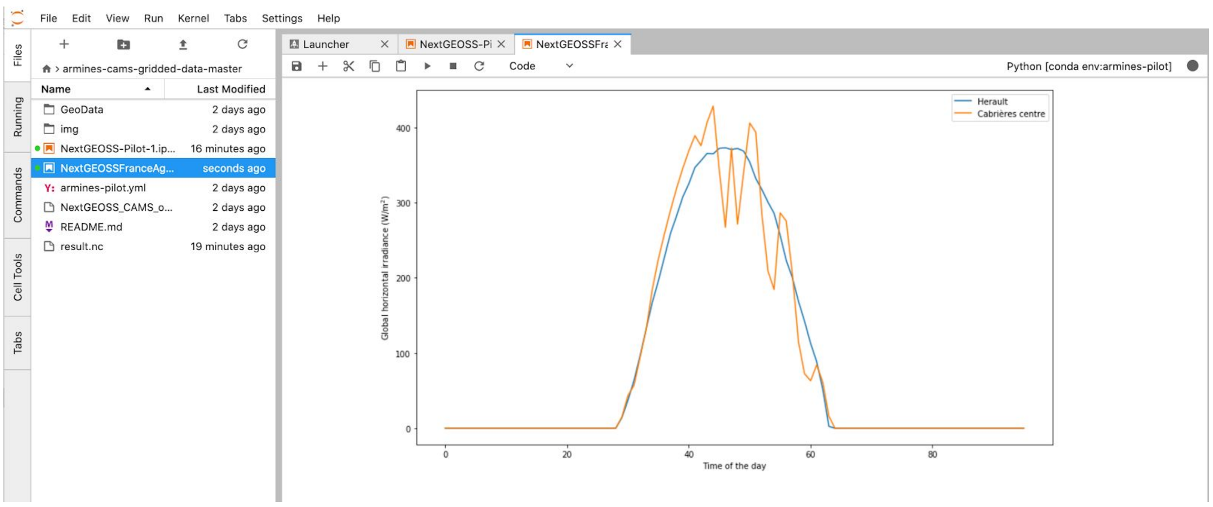
To cross our data with existing maps and borders, we use the python libraries geopandas and shapely. I you want to change the default values, you will have to jump into the following Python code.

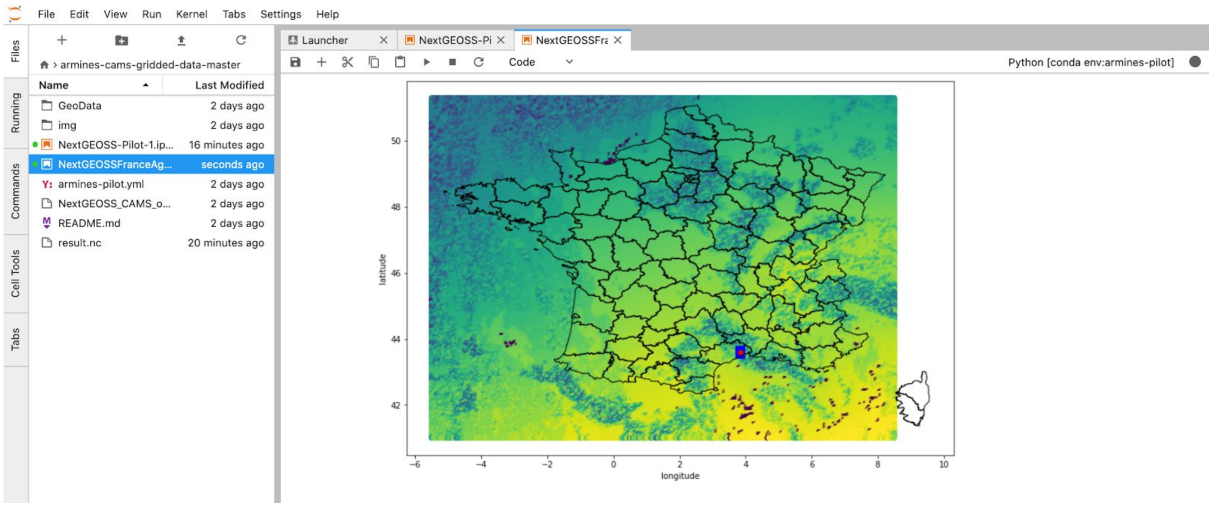
- When finished, you'll get a set of maps and corresponding graphs comparing pointwise vs AOI approach plus aggregated data per departments such as follows.


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
armines-cams-gridded-data-master
Name
Last Modified
GeoData 2 days ago
img 2 days ago
NextGEOSS-Pilot-1.ip… 15 minutes ago
NextGEOSSFranceAg… in a few seconds
armines-pilot.yml 2 days ago
NextGEOSS_CAMS_o… 2 days ago
README.md 2 days ago
result.nc 19 minutes ago
Launcher
NextGEOSS-Pi
NextGEOSSFra
Code
Python [conda env:armines-pilot]
50
48
46
44
42
latitude
−6
−4
−2
0
2
4
6
8
10
longitude

```
#base.set_xlim(lon_min, lon_max)


## Plot department and its center
my_dpt.plot(ax=base,color='blue',lw=2)
my_dpt.centroid.plot(ax=base,color='red')
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd1f7bc95f8>
```

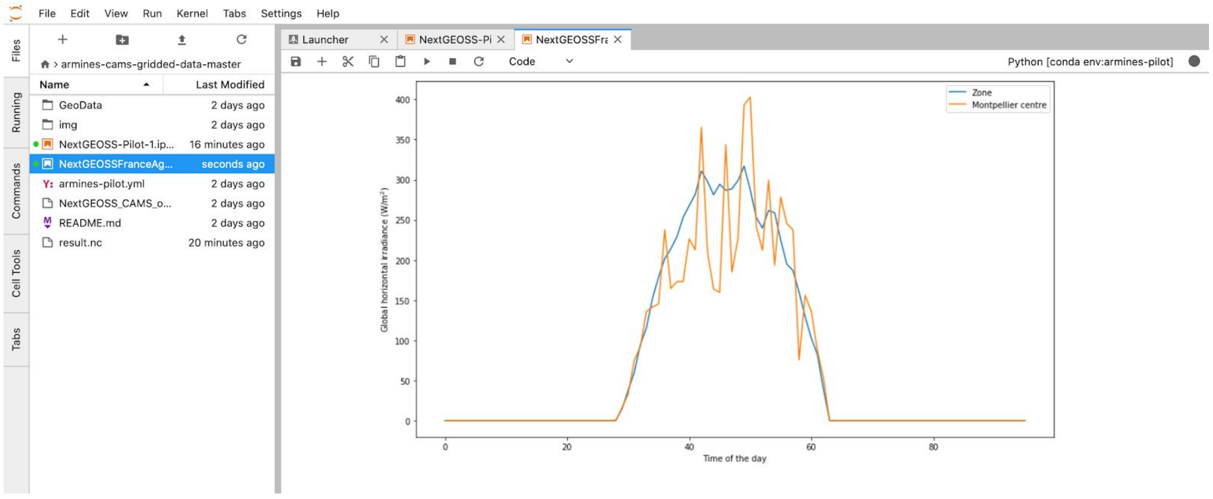

File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
armines-cams-gridded-data-master
Name
Last Modified
GeoData 2 days ago
img 2 days ago
NextGEOSS-Pilot-1.ip… 16 minutes ago
NextGEOSSFranceAg… seconds ago
armines-pilot.yml 2 days ago
NextGEOSS_CAMS_o… 2 days ago
README.md 2 days ago
result.nc 19 minutes ago
Launcher
NextGEOSS-Pi
NextGEOSSFra
Code
Python [conda env:armines-pilot]
Herault
Cabrières centre
400
300
200
100
0
Global horizontal irradiance (W/m²)
0
20
40
60
80
Time of the day



Python [conda env:armines-pilot]

armines-cams-gridded-data-master

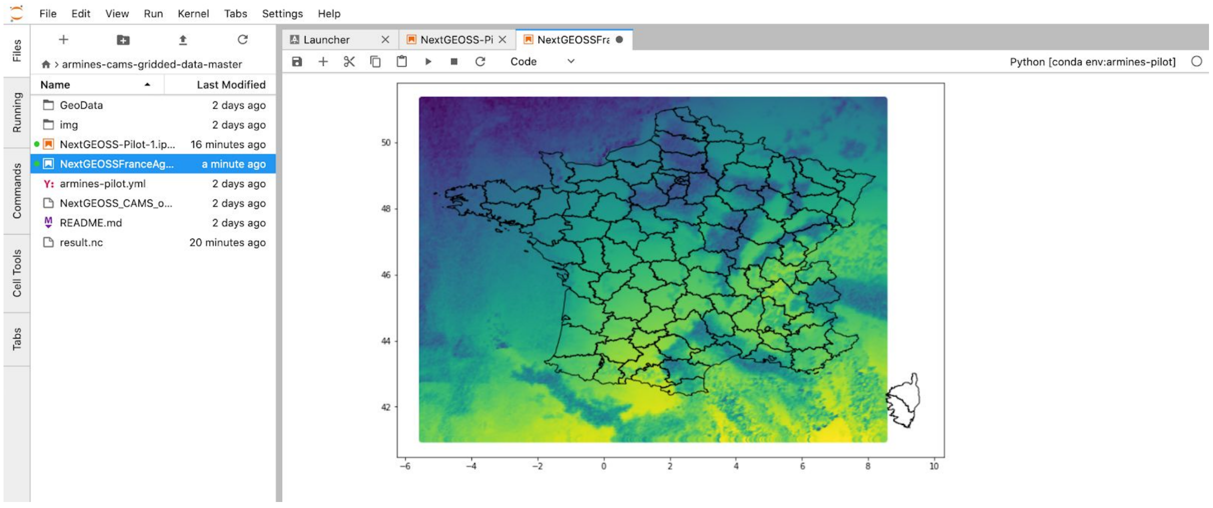
Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	16 minutes ago
NextGEOSSFranceAg...	seconds ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	20 minutes ago



Python [conda env:armines-pilot]

armines-cams-gridded-data-master

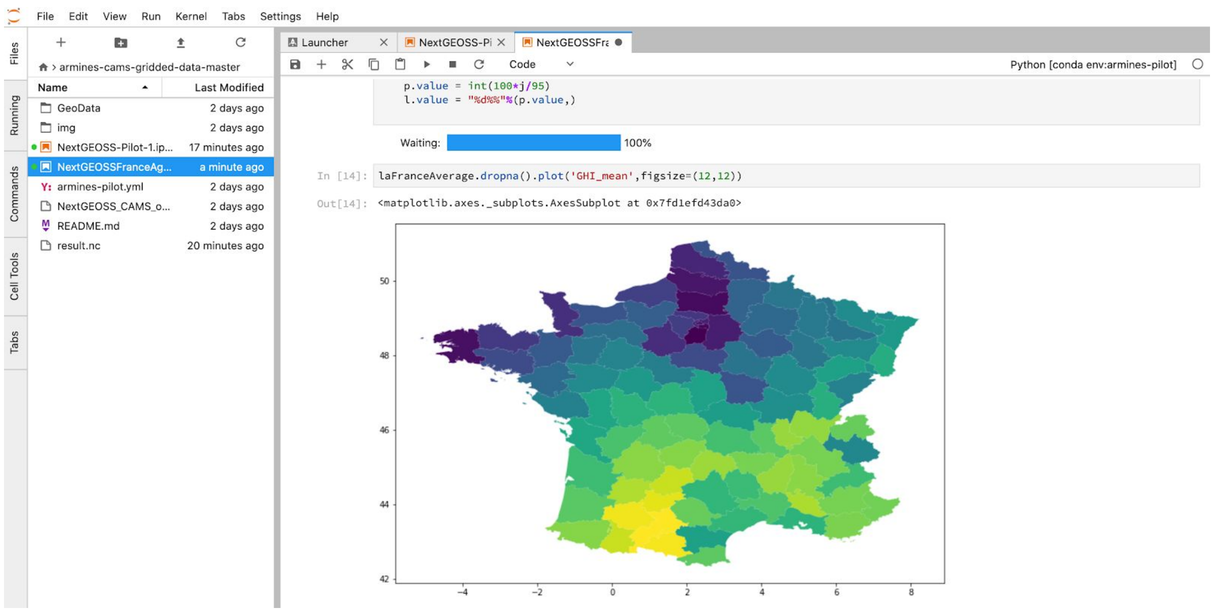
Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	16 minutes ago
NextGEOSSFranceAg...	seconds ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	20 minutes ago



Python [conda env:armines-pilot]

armines-cams-gridded-data-master

Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	16 minutes ago
NextGEOSSFranceAg...	a minute ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	20 minutes ago



armines-cams-gridded-data-master

Name	Last Modified
GeoData	2 days ago
img	2 days ago
NextGEOSS-Pilot-1.ip...	17 minutes ago
NextGEOSSFranceAg...	a minute ago
armines-pilot.yml	2 days ago
NextGEOSS_CAMS_o...	2 days ago
README.md	2 days ago
result.nc	20 minutes ago

```
    p.value = int(100*j/95)
    l.value = "%d%%"%(p.value,)
```

Waiting: 100%

```
In [14]: laFranceAverage.dropna().plot('GHI_mean',figsize=(12,12))

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd1efd43da0>
```

This simple example is intended to demonstrate the easy, yet powerful potential of Jupyter Notebook/Python approach for rendering value added information to support decision making. Feel free to customize this example to suit your own needs.