

MOHID Notebooks

Guilherme Franz, Eng. Dr.

guilherme.franz@aquafLOWconsultoria.com



Jupyter Notebooks

- **Interactive and Flexible Environment**

- Combines live code, narrative text, and visualizations in one document.
- Supports multiple programming languages (e.g., Python, R, Julia).

- **Cell-Based Structure**

- Divides content into independent cells for code or rich text (Markdown).
- Allows immediate execution and visualisation of outputs.

- **Reproducible Research and Collaboration**

- Embeds explanations alongside analysis, enhancing clarity and reproducibility.
- Enables the sharing of fully annotated documents easily.

Available on the MOHID GitHub repository

https://github.com/Mohid-Water-Modelling-System/MOHID_Jupyter-Notebooks

The screenshot shows the GitHub repository page for **MOHID_Jupyter-Notebooks** under the **Mohid-Water-Modelling-System** organization. The page is viewed on the **master** branch. The repository is public and has 6 watchers, 0 forks, and 0 stars. The commit history shows a recent update to **README.md** by **guifranz** 4 minutes ago. The file list includes **MOHID_Lagrangian**, **MOHID_Preprocessing**, **MOHID_Water**, and **README.md**. The right sidebar shows the repository description: "Jupyter Notebooks for the MOHID Water Modelling System" and a list of tags: **preprocessing**, **jupyter-notebooks**, and **mohid**.

Mohid-Water-Modelling-System / MOHID_Jupyter-Notebooks

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

MOHID_Jupyter-Notebooks Public

master 1 Branch 0 Tags

Go to file

Code

guifranz Update README.md 20cc5e7 · 4 minutes ago 32 Commits

MOHID_Lagrangian	Download wind from ERA5 Reanalysis	3 weeks ago
MOHID_Preprocessing	Update MOHID_Preprocessing.ipynb	30 minutes ago
MOHID_Water	Get river data	yesterday
README.md	Update README.md	4 minutes ago

README

About

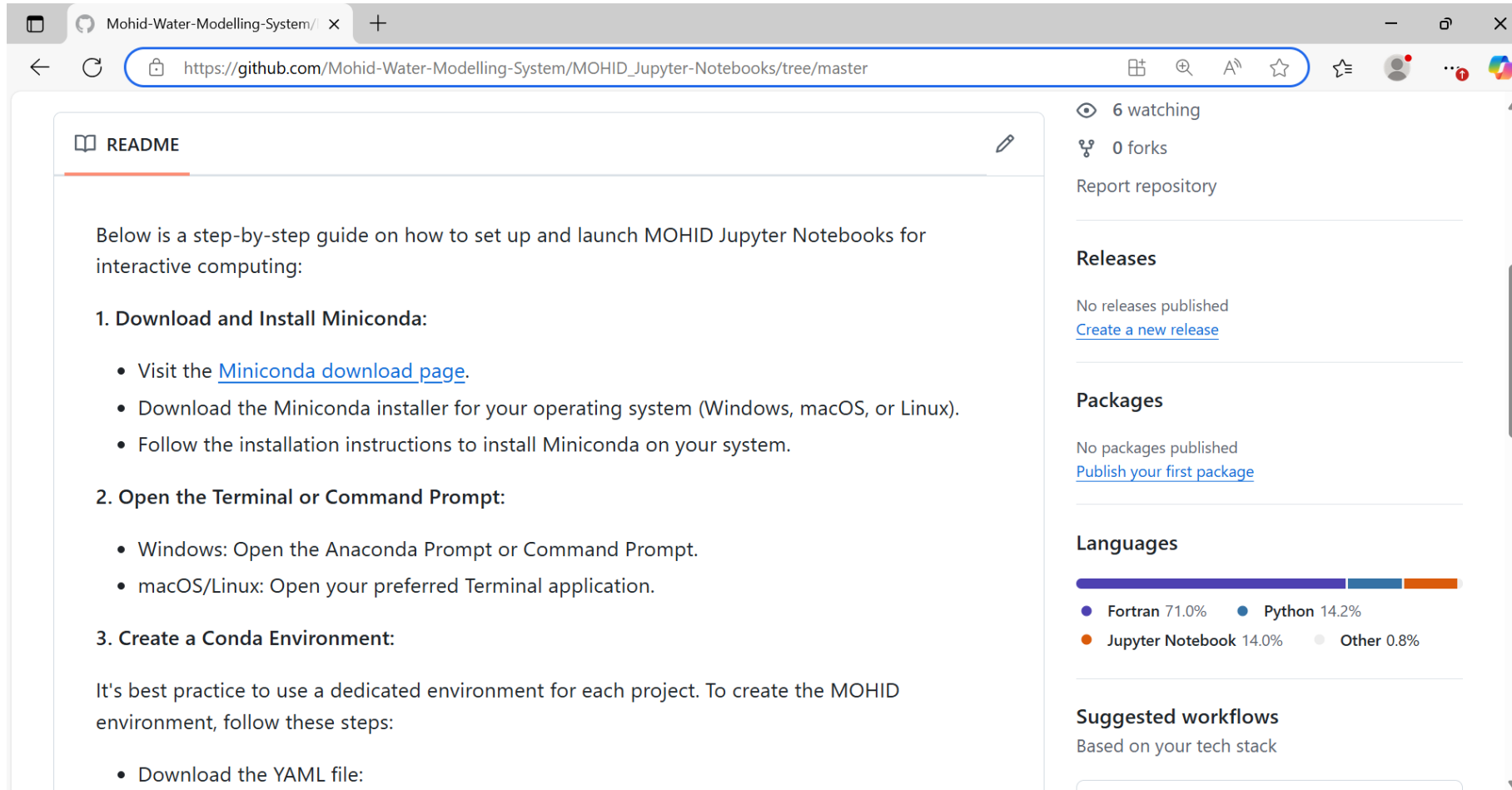
Jupyter Notebooks for the MOHID Water Modelling System

preprocessing jupyter-notebooks mohid

Readme Activity Custom properties 0 stars 6 watching 0 forks Report repository

Available on the MOHID GitHub repository

https://github.com/Mohid-Water-Modelling-System/MOHID_Jupyter-Notebooks



The screenshot shows the GitHub repository page for MOHID_Jupyter-Notebooks. The browser address bar displays the URL: https://github.com/Mohid-Water-Modelling-System/MOHID_Jupyter-Notebooks/tree/master. The repository name is "Mohid-Water-Modelling-System/".

README

Below is a step-by-step guide on how to set up and launch MOHID Jupyter Notebooks for interactive computing:

- 1. Download and Install Miniconda:**
 - Visit the [Miniconda download page](#).
 - Download the Miniconda installer for your operating system (Windows, macOS, or Linux).
 - Follow the installation instructions to install Miniconda on your system.
- 2. Open the Terminal or Command Prompt:**
 - Windows: Open the Anaconda Prompt or Command Prompt.
 - macOS/Linux: Open your preferred Terminal application.
- 3. Create a Conda Environment:**

It's best practice to use a dedicated environment for each project. To create the MOHID environment, follow these steps:

 - Download the YAML file:

Repository Statistics:

- 6 watching
- 0 forks
- Report repository

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

Horizontal bar chart showing language distribution:

Language	Percentage
Fortran	71.0%
Python	14.2%
Jupyter Notebook	14.0%
Other	0.8%

Suggested workflows

Based on your tech stack

Available on the MOHID GitHub repository

https://github.com/Mohid-Water-Modelling-System/MOHID_Jupyter-Notebooks

The screenshot shows the GitHub repository page for `MOHID_Jupyter-Notebooks`. The left sidebar displays the file tree with the `UserGuides` directory selected. The main content area shows the commit history for the `UserGuides` directory, listing three PDF files: `MOHID_Lagrangian_UserGuide.pdf`, `MOHID_Preprocessing_UserGuide.pdf`, and `MOHID_Water_UserGuide.pdf`.

Name	Last commit message	Last commit date
..		
MOHID_Lagrangian_UserGuide.pdf	Create MOHID_Lagrangian_UserGuide	5 days ago
MOHID_Preprocessing_UserGuide.pdf	Create MOHID_Preprocessing_UserGuide.pdf	2 weeks ago
MOHID_Water_UserGuide.pdf	Create MOHID_Water_UserGuide	last week

MOHID Lagrangian Notebook

- Lagrangian models use hydrodynamic and wind fields to calculate the movement of material systems, such as marine debris.
- The execution of the Lagrangian model is independent of the source of the hydrodynamic and wind fields.
- A Jupyter Notebook was developed to help users of the MOHID Lagrangian model explore available hydrodynamic and wind fields online, where Copernicus products play a key role, or use their own solutions, e.g., based on MOHID Water.

MOHID Lagrangian Notebook

The screenshot shows a web browser window displaying the MOHID Lagrangian Notebook. The browser's address bar shows the URL `localhost:8888/lab/tree/MOHID_Lagrangian.ipynb`. The notebook interface includes a left-hand sidebar with a file explorer showing the following files:

Name	Modified	Size
run_cases	7d ago	
MOHID_Lagrangian.ipynb	7d ago	38.7 KB
MOHID_Lagrangian_envir...	12m ago	341 B
update_xml_case.py	7d ago	15.7 KB

The main notebook area displays the title "MOHID Lagrangian" and a description: "This Jupyter Notebook aims to help implement and run the MOHID Lagrangian model." Below this, there are two notes:

Note 1: Execute each cell through the button from the top MENU (or keyboard shortcut `Shift + Enter`).

Note 2: Use the Kernel and Cell menus to restart the kernel and clear outputs.

The notebook also features a "Table of contents" section with the following items:

- 1. Import required libraries
- 2. General options
 - 2.1 Set run case
 - 2.2 Set dates
 - 2.3 Draw a polygon to select the area of interest
- 3. Download currents for the area of interest
 - 3.1 Create Copernicus Marine credentials file
 - 3.2 Set CMEMS product and depths
 - 3.3 Download CMEMS
- 4. Download wind field for the area of interest

The bottom status bar indicates the notebook is running on "Python 3 (ipykernel)" in "Idle" mode, with the cursor at "Ln 1, Col 1". The Windows taskbar at the bottom shows the time as 2:18 PM on 6/4/2025.

Download current and wind fields

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

File Edit View Run Kernel Tabs Settings Help

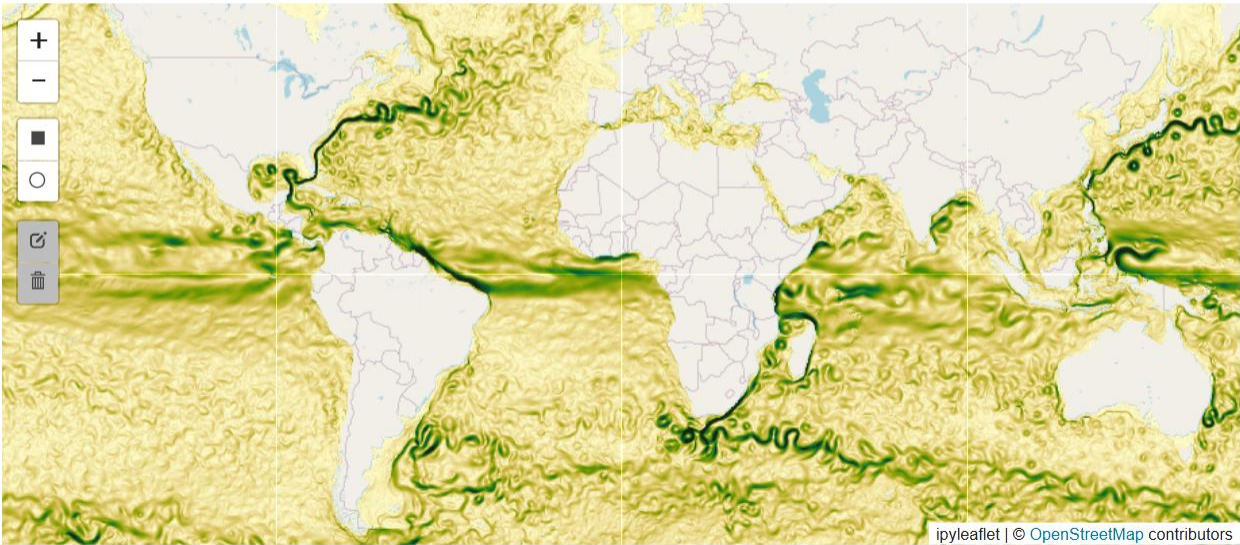
Launcher MOHID_Lagrangian.ipynb

Notebook Python 3 (ipykernel)

```
# Add DrawControl to the map
m.add_control(draw_control)

# Display the interactive map
m
```

[4]:



ipyleaflet | © OpenStreetMap contributors

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 28, Col 43 MOHID_Lagrangian.ipynb 0

Download current and wind fields

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

File Edit View Run Kernel Tabs Settings Help

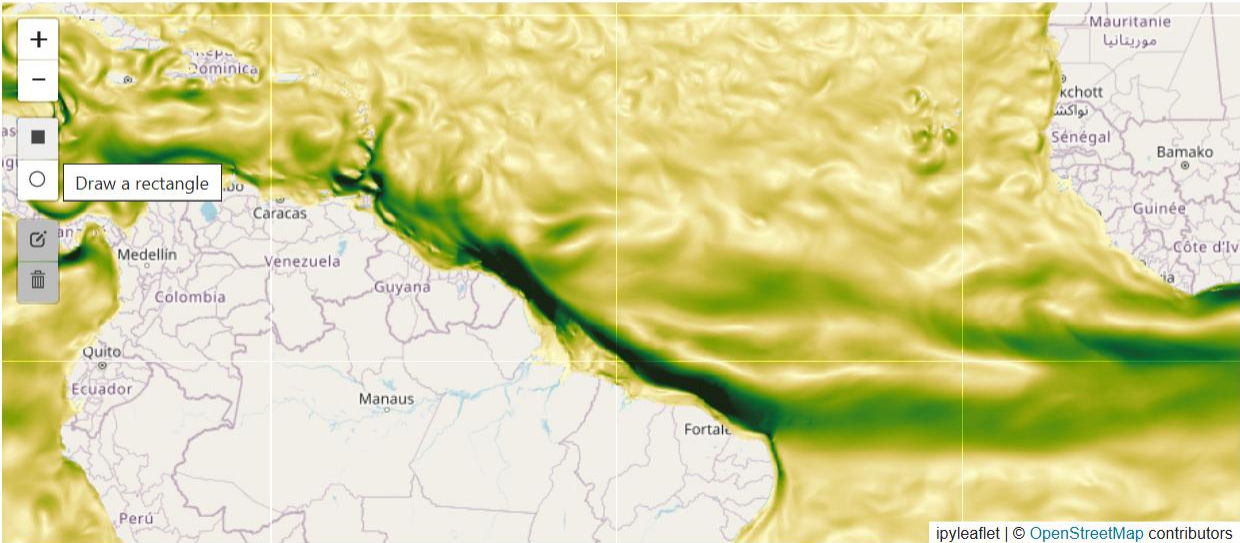
Launcher x MOHID_Lagrangian.ipynb x +

Notebook Python 3 (ipykernel)

```
# Add DrawControl to the map
m.add_control(draw_control)

# Display the interactive map
m
```

[4]:



ipyleaflet | © OpenStreetMap contributors

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb# Idle Mode: Command Ln 28, Col 43 MOHID_Lagrangian.ipynb 0

Download current and wind fields

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

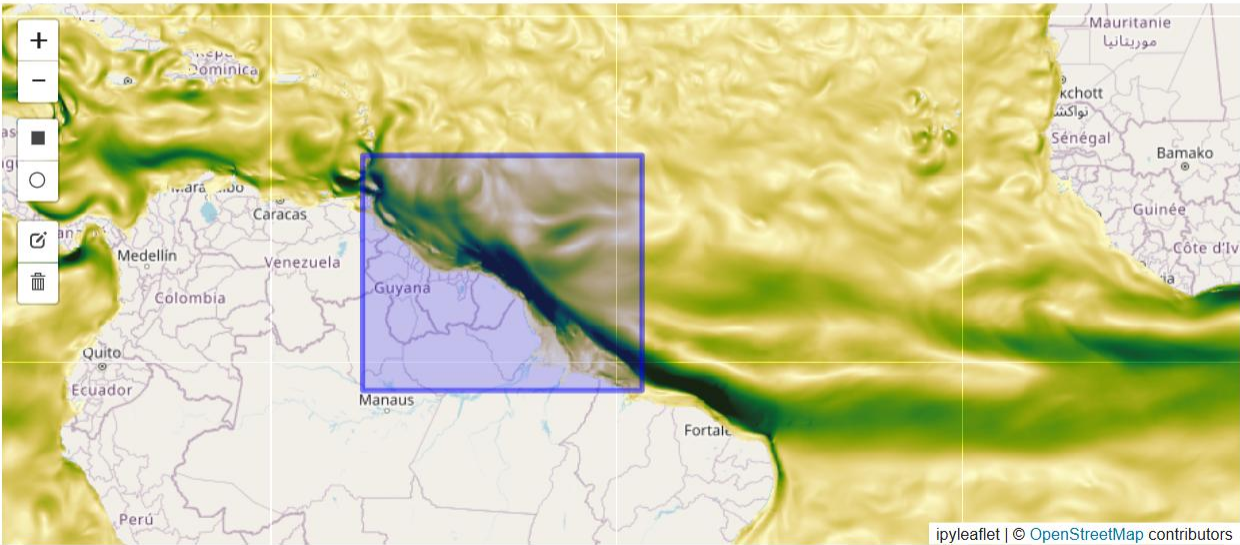
File Edit View Run Kernel Tabs Settings Help

Launcher MOHID_Lagrangian.ipynb

```
# Add DrawControl to the map
m.add_control(draw_control)

# Display the interactive map
m
```

[4]:



ipyleaflet | © OpenStreetMap contributors

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 28, Col 43 MOHID_Lagrangian.ipynb 0

Download current and wind fields

The screenshot displays a JupyterLab environment with a browser window at the top showing the URL `localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries`. The JupyterLab interface includes a file browser on the left, a central code editor, and a bottom status bar.

File Browser:

Name	Modi...	Size
run_cases	7d ago	
MOHID_Lagra...	now	40.7 KB
MOHID_Lagra...	12m ago	341 B
update_xml_c...	7d ago	15.7 KB

Code Editor:

```
start_datetime = str(start_date.strftime('%Y-%m-%d')) + ' 00:00:00',
end_datetime = str(end_date.strftime('%Y-%m-%d')) + ' 00:00:00',
variables = variable,
output_directory = output_dir_cmems,
output_filename = output_file_cmems,
netcdf3_compatible = True)

#####

output_file_cmems = "cmems_" + str(start_date.strftime("%Y%m%d")) + "_" + str(end_date.strftime("%Y%m%d")) + ".nc"

if not os.path.exists(output_dir_cmems):
    os.makedirs(output_dir_cmems)

nc_files = glob.iglob(os.path.join(output_dir_cmems, "*.nc"))

for filename in nc_files:
    os.remove(filename)

download_file()
```

Output:

```
INFO - 2025-06-04T17:32:57Z - Selected dataset version: "202406"
INFO - 2025-06-04T17:32:57Z - Selected dataset part: "default"
INFO - 2025-06-04T17:33:08Z - Starting download. Please wait...
100% [Progress Bar] 50/50 [00:23<00:00, 1.72it/s]
INFO - 2025-06-04T17:33:33Z - Successfully downloaded to C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-N
otebooks-master\MOHID_Lagrangian\run_cases\Plastic_Case\nc_fields\currents\cmems_20250101_20250105.nc
```

Status Bar: Simple | 1 | Python 3 (ipykernel) | Idle | Mode: Command | Ln 1, Col 1 | MOHID_Lagrangian.ipynb | 0

Define sources

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Lagrangian.ipynb

Notebook Python 3 (ipykernel)

```
# -----  
# Display the map  
# -----  
m
```

[10]:

ipyleaflet | © OpenStreetMap contributors

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 MOHID_Lagrangian.ipynb 0

Define sources

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

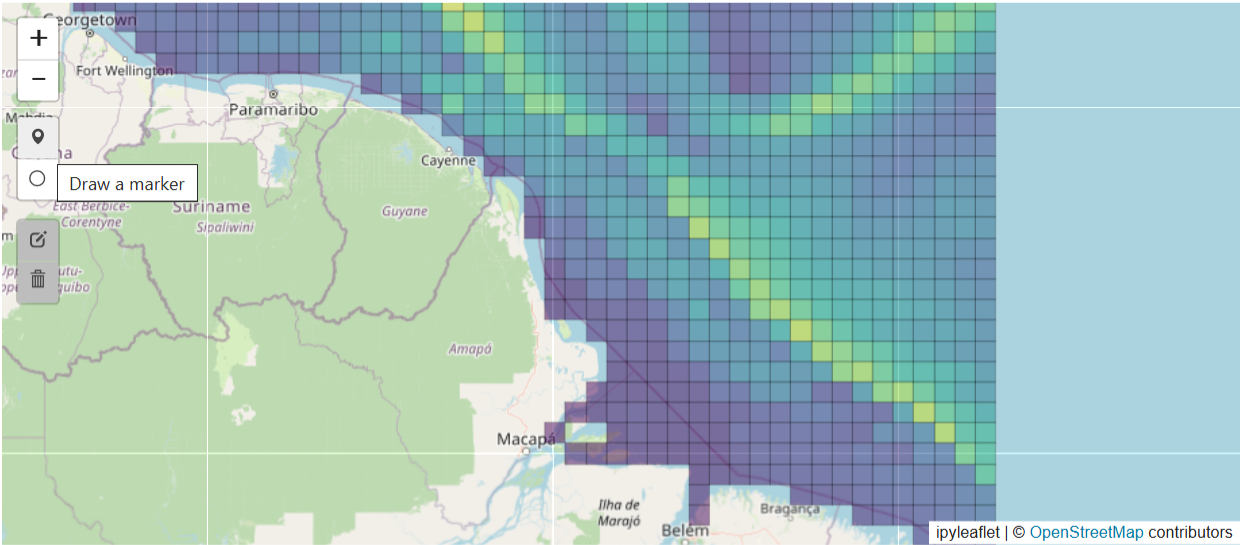
File Edit View Run Kernel Tabs Settings Help

Launcher MOHID_Lagrangian.ipynb

Notebook Python 3 (ipykernel)

```
# -----  
# Display the map  
# -----  
m
```

[10]:



ipyleaflet | © OpenStreetMap contributors

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb# | Idle

Mode: Command Ln 11, Col 1 MOHID_Lagrangian.ipynb 0

Define sources

The screenshot displays a JupyterLab environment with a web browser at the top showing the URL `localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries`. The JupyterLab interface includes a file browser on the left, a central code editor, and a bottom status bar.

File Browser:

Name	Modi...	Size
run_cases	7d ago	
MOHID_Lagra...	now	74.7 KB
MOHID_Lagra...	12m ago	341 B
update_xml_c...	7d ago	15.7 KB

Code Editor:

```
# -----  
# Display the map  
# -----  
m
```

Below the code editor, a map visualization is shown. The map displays a grid of colored cells (blue, green, yellow) over a geographical area including Suriname, Guyane, and Amapá. A blue location pin is placed on the map. A text input field labeled "Source name:" contains the text "S1". A green "Confirm" button is visible above the map.

Status Bar:

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 11, Col 1 MOHID_Lagrangian.ipynb 0

Define sources

The screenshot displays a JupyterLab environment with a web browser at the top showing the URL `localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries`. The interface includes a file explorer on the left, a central code editor, and a bottom status bar.

File Explorer: Shows a directory structure with files like `run_cases`, `MOHID_Lagra...` (74.7 KB), and `update_xml_c...` (15.7 KB).

Code Editor: Contains the following code snippet:

```
# -----  
# Display the map  
# -----  
m
```

Below the code, a message indicates that `ask_marker_name` was invoked for marker 0 and marker 1. A form labeled "Source name:" has the text "S2" entered.

Map Visualization: A map of South America, specifically showing Suriname, Guyana, and Amapá. The map is overlaid with a grid of colored squares (blue, green, yellow, orange, red) representing data points. Two blue location pins are visible on the map. A "Confirm" button is present above the map.

Status Bar: Shows "Simple" mode, "Python 3 (ipykernel) | Idle", and "Mode: Command". The notebook is titled "MOHID_Lagrangian.ipynb" and is at line 11, column 1.

Run and visualise results

The screenshot displays the JupyterLab web interface in a browser window. The address bar shows the URL `localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries`. The interface includes a left-hand sidebar with a file explorer showing a directory structure with files like `run_cases`, `MOHID_Lagra...`, and `update_xml_c...`. The main area contains a notebook with the following content:

6. Setup MOHID Lagrangian xml input files

6.1 Parameter definitions

```
[11]: # Get time limits (min and max time values in the dataset's 'time' variable)
start, end = min(dataset['time'].values), max(dataset['time'].values)

# Convert the numpy.datetime64 to a Python datetime object using pandas
Start = pd.to_datetime(start) #Date of initial instant based on nc file
End = pd.to_datetime(end) #Date of final instant based on nc file
#Start = datetime.datetime(2024, 1, 1, 0, 0, 0)
#End = datetime.datetime(2024, 1, 2, 0, 0, 0)

Integrator = 2 #Integration Algorithm 1:Euler, 2:Multi-Step Euler, 3:RK4 (default=1)
Threads = "auto" #Computation threads for shared memory computation (default=auto)
OutputWriteTime = 86400 #Time out data (seconds)
BufferSize = 86400 #control the amount of hydrodynamic data to store in RAM memory (seconds)

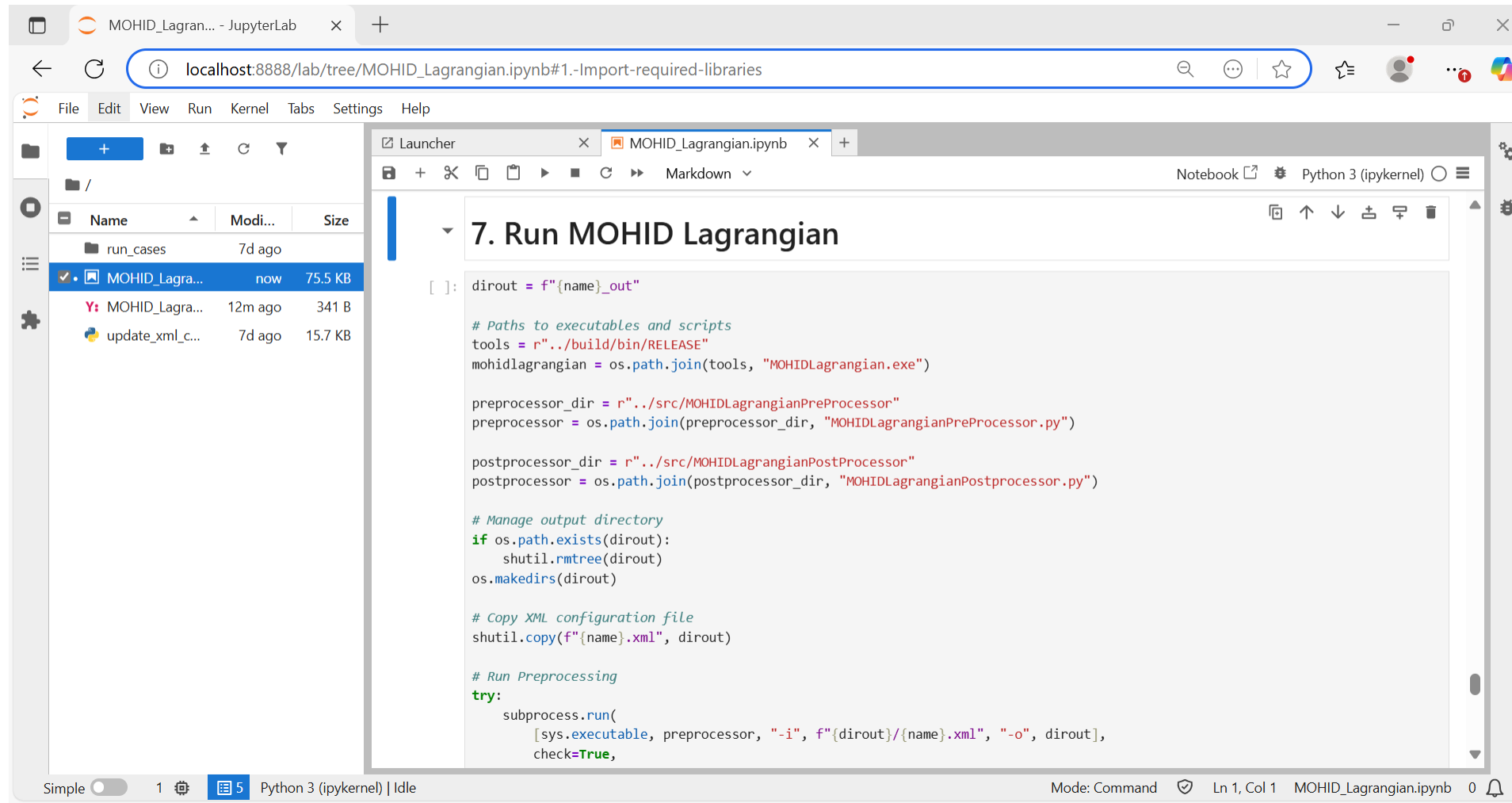
# Run the update function
update_parameter_definitions(xml_file_path,Start,End,Integrator,Threads,OutputWriteTime,BufferSize)

Updated <parameters> block in 'Plastic_Case.xml' successfully.
```

6.2 Simulation definitions

The bottom status bar indicates the current mode is 'Command', the cursor is at 'Ln 1, Col 1', and the active kernel is 'Python 3 (ipykernel)'.

Run and visualise results



MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

File Edit View Run Kernel Tabs Settings Help

Launcher MOHID_Lagrangian.ipynb

7. Run MOHID Lagrangian

```
[ ]: dirout = f"{name}_out"

# Paths to executables and scripts
tools = r"../build/bin/RELEASE"
mohidlagrangian = os.path.join(tools, "MOHIDLagrangian.exe")

preprocessor_dir = r"../src/MOHIDLagrangianPreProcessor"
preprocessor = os.path.join(preprocessor_dir, "MOHIDLagrangianPreProcessor.py")

postprocessor_dir = r"../src/MOHIDLagrangianPostProcessor"
postprocessor = os.path.join(postprocessor_dir, "MOHIDLagrangianPostprocessor.py")

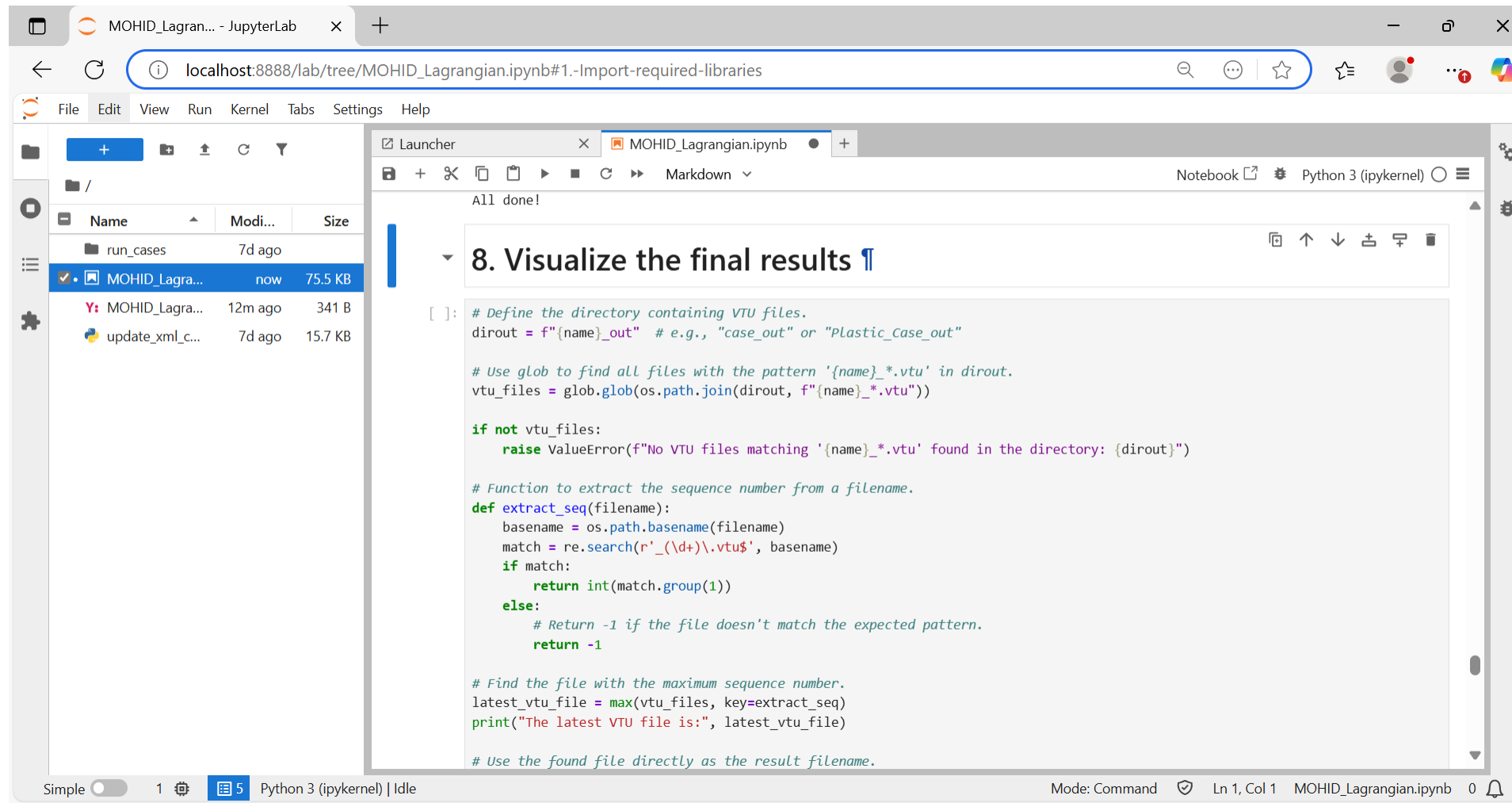
# Manage output directory
if os.path.exists(dirout):
    shutil.rmtree(dirout)
os.makedirs(dirout)

# Copy XML configuration file
shutil.copy(f"{name}.xml", dirout)

# Run Preprocessing
try:
    subprocess.run(
        [sys.executable, preprocessor, "-i", f"{dirout}/{name}.xml", "-o", dirout],
        check=True,
```

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 MOHID_Lagrangian.ipynb 0

Run and visualise results



The screenshot shows a JupyterLab interface with a file browser on the left and a code editor on the right. The file browser displays a directory structure with files like `run_cases`, `MOHID_Lagra...`, and `update_xml_c...`. The code editor shows a Python script for visualizing results, including file globbing and sequence extraction.

8. Visualize the final results ¶

```
[ ]: # Define the directory containing VTU files.
dirout = f'{name}_out' # e.g., "case_out" or "Plastic_Case_out"

# Use glob to find all files with the pattern '{name}_*.vtu' in dirout.
vtu_files = glob.glob(os.path.join(dirout, f'{name}_*.vtu'))

if not vtu_files:
    raise ValueError(f"No VTU files matching '{name}_*.vtu' found in the directory: {dirout}")

# Function to extract the sequence number from a filename.
def extract_seq(filename):
    basename = os.path.basename(filename)
    match = re.search(r'_(\d+)\.vtu$', basename)
    if match:
        return int(match.group(1))
    else:
        # Return -1 if the file doesn't match the expected pattern.
        return -1

# Find the file with the maximum sequence number.
latest_vtu_file = max(vtu_files, key=extract_seq)
print("The latest VTU file is:", latest_vtu_file)

# Use the found file directly as the result filename.
```

Run and visualise results

MOHID_Lagran... - JupyterLab

localhost:8888/lab/tree/MOHID_Lagrangian.ipynb#1.-Import-required-libraries

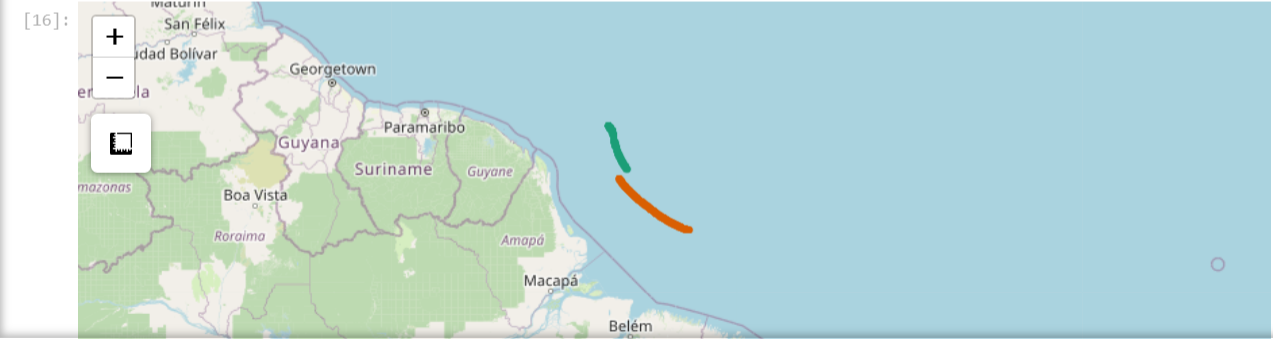
File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Lagrangian.ipynb

```
fill_opacity=0.7,  
popup=f"Source: {src}"  
) .add_to(map_vtu)  
  
map_vtu.save(os.path.join(dirout, "map_vtu.html"))  
print("Map saved as map_vtu.html")  
  
# Display the map inline in a Jupyter Notebook (if applicable)  
map_vtu
```

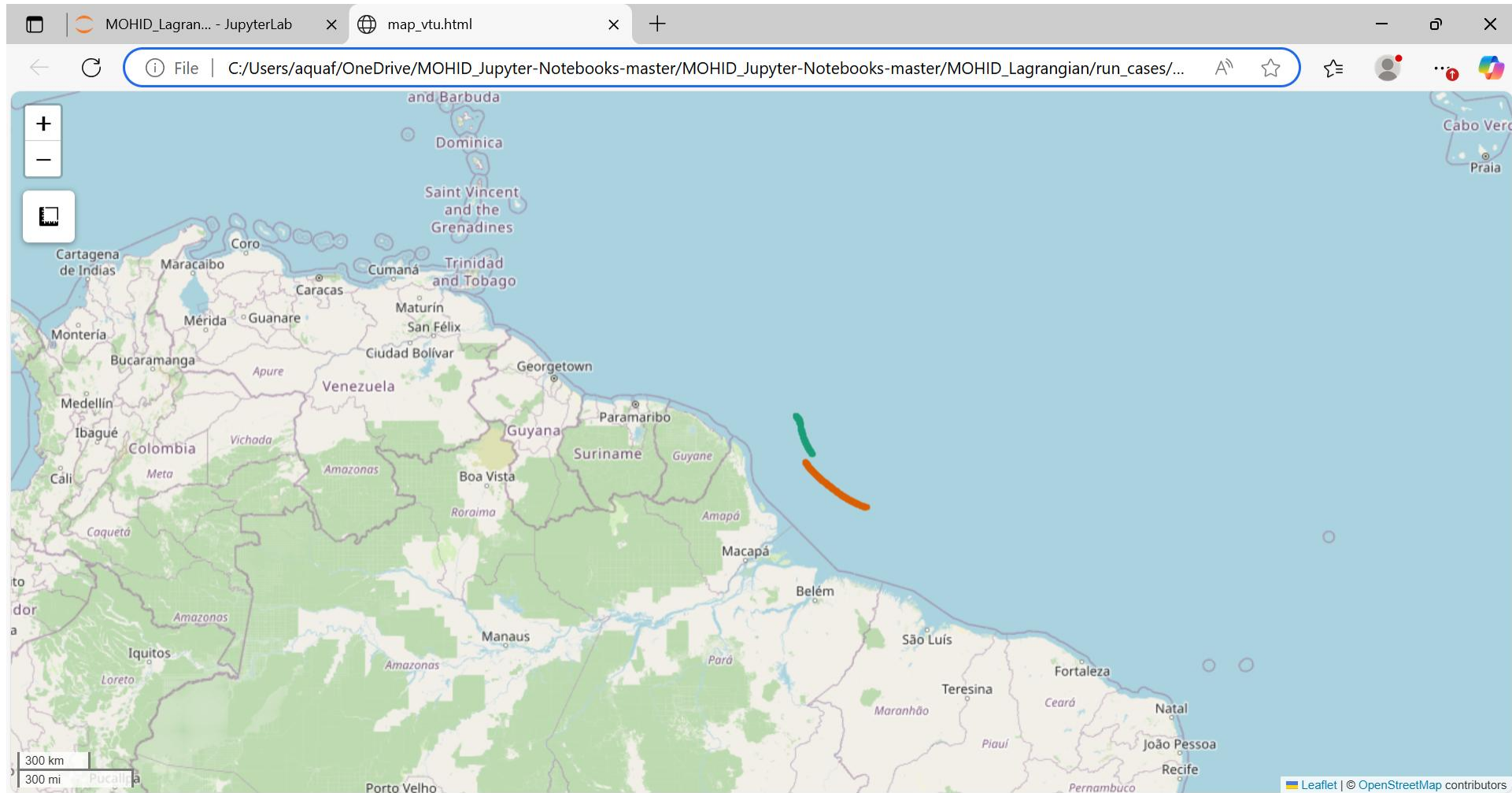
The latest VTU file is: Plastic_Case_out\Plastic_Case_00004.vtu
Unique sources: [0, 1]
Map saved as map_vtu.html

[16]:



Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 MOHID_Lagrangian.ipynb 0

Run and visualise results



MOHID Preprocessing Notebook

The screenshot shows the JupyterLab interface for the MOHID Preprocessing Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Preprocessing.ipynb`. The left sidebar contains a file tree with the following items:

Name	Mo...	Size
Coastlines	43m ago	
Data	43m ago	
MOHID_Preprocessing...	43m ago	44.9 KB
MOHID_Preprocessing...	43m ago	252 B

The main notebook area displays the following content:

MOHID Preprocessing

- Create regular grids
- Download, load and filter coastlines for grid region
- Perform interpolation on bathymetric data
- Update griddata depth values
- Convert the griddata to a shapefile
- Plot bathymetry

Note 1: Execute each cell through the button from the top MENU (or keyboard shortcut `Shift + Enter`).

Note 2: Use the Kernel and Cell menus to restart the kernel and clear outputs.

Table of contents

- [1. Import required libraries](#)
- [2. Load the XYZ data](#)
- [3. Get grid dimensions and spacing](#)
- [4. Grid generation](#)
- [5. Save the grid to a MOHID-compatible file](#)

The bottom status bar shows: `Simple` (toggle), `1`, `Python 3 (ipykernel) | Idle`, `Mode: Command`, `Ln 1, Col 1`, `MOHID_Preprocessing.ipynb`, and `0` notifications.

MOHID Preprocessing Notebook

The screenshot shows a JupyterLab environment with the following components:

- Browser Tabs:** 'Mohid-Water-Modelling-System/' and 'MOHID_Prepro... - JupyterLab'.
- Address Bar:** 'localhost:8888/lab/tree/MOHID_Preprocessing.ipynb'.
- File Explorer (Left Sidebar):**

Name	Mo...	Size
Coastlines	43m ago	
Data	43m ago	
MOHID_Preprocessing...	43m ago	44.9 KB
MOHID_Preprocessing...	43m ago	252 B
- Notebook Interface:**
 - Table of contents:**
 - 1. Import required libraries
 - 2. Load the XYZ data
 - 3. Get grid dimensions and spacing
 - 4. Grid generation
 - 5. Save the grid to a MOHID-compatible file
 - 6. Download or load GSHHG Coastline Data
 - 7. Load and Filter Coastlines for Grid Region
 - 8. Interpolate bathymetric data
 - 9. Load a previously generated Mohid griddata file
 - 10. Visualize and update depth values by clicking on the map
 - 11. Save the griddata to a MOHID-compatible file
 - 12. Convert the griddata to a shapefile
 - 13. Save shapefile to MOHID griddata
 - 14. Plot MOHID griddata
 - 1. Import required libraries:**

```
[ ]: import numpy as np
import pandas as pd
```

At the bottom, the status bar shows: 'localhost:8888/lab/tree/MOHID_Preprocessing.ipynb#1.-Import-required-libraries', 'Mode: Command', 'Ln 1, Col 1', 'MOHID_Preprocessing.ipynb', and '0'.

MOHID Preprocessing Notebook

The screenshot displays the JupyterLab interface for the 'MOHID Preprocessing Notebook'. The browser address bar shows 'localhost:8888/lab/tree/MOHID_Preprocessing.ipynb'. The left sidebar contains a file explorer with a table of files:

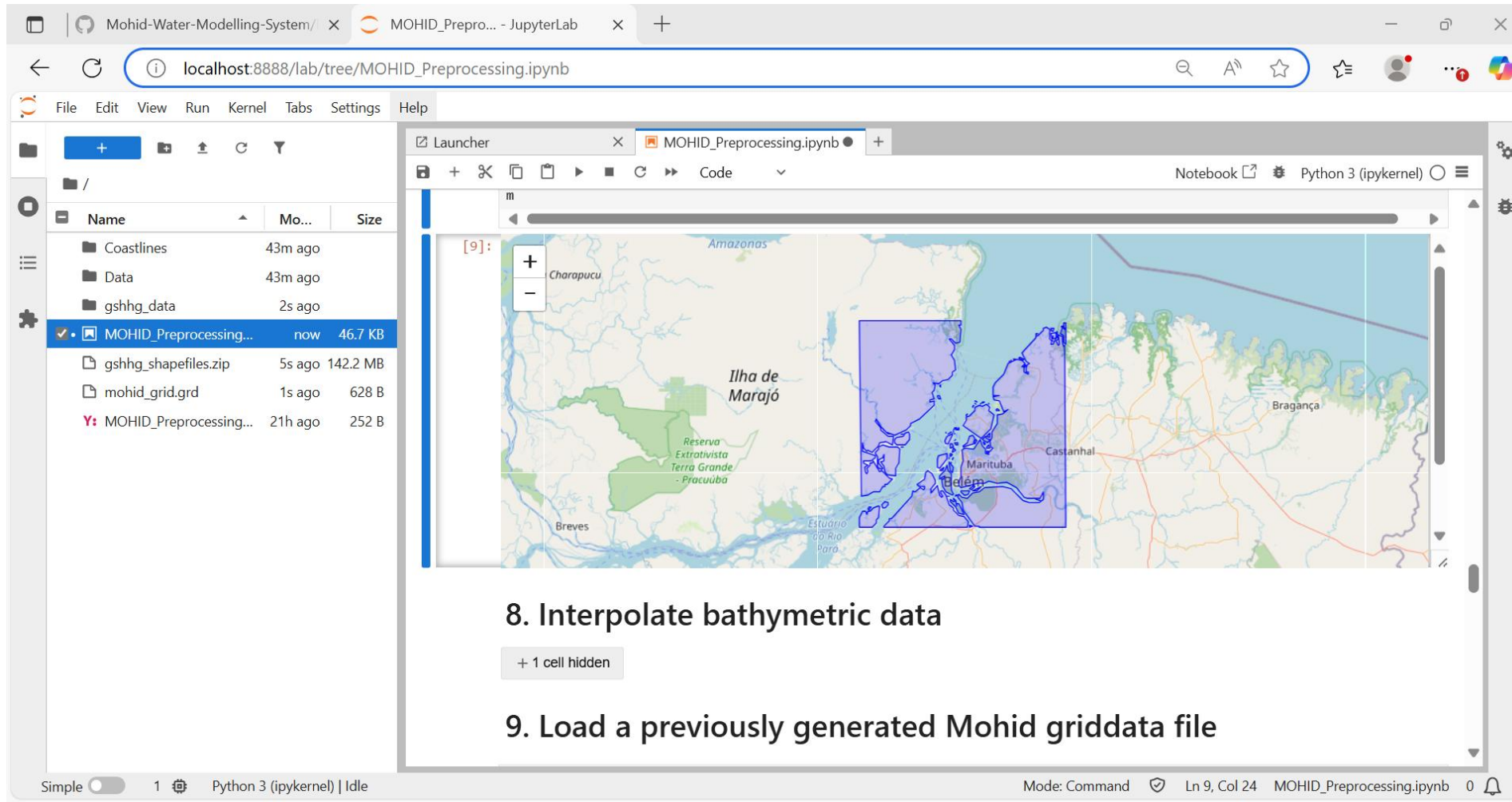
Name	Mo...	Size
Coastlines	43m ago	
Data	43m ago	
MOHID_Preprocessing...	now	46.5 KB
MOHID_Preprocessing...	43m ago	252 B

The main area shows the notebook 'MOHID_Preprocessing.ipynb' in 'Code' mode. It features an interactive map of a coastal region with a blue grid overlay. The map includes labels for 'Ilha de Marajó', 'Muana', 'Castanhal', and 'Inhangapi'. Below the map, the text 'Interactive Map for Grid Generation' is followed by two instructions:

1. Click anywhere on the map to find the origin coordinates (latitude and longitude).
2. The grid will be visualized as a mesh directly on the map.

The bottom status bar indicates 'Simple' mode, 'Python 3 (ipykernel)', and 'Idle'. The right side shows 'Mode: Command', 'Ln 7, Col 1', and the file name 'MOHID_Preprocessing.ipynb'.

MOHID Preprocessing Notebook



The screenshot displays a JupyterLab environment with a browser window at the top showing the URL `localhost:8888/lab/tree/MOHID_Preprocessing.ipynb`. The interface includes a file browser on the left, a central code editor, and a bottom status bar.

File Browser:

Name	Mo...	Size
Coastlines	43m ago	
Data	43m ago	
gshhg_data	2s ago	
MOHID_Preprocessing...	now	46.7 KB
gshhg_shapefiles.zip	5s ago	142.2 MB
mohid_grid.grd	1s ago	628 B
MOHID_Preprocessing...	21h ago	252 B

Code Editor:

The code editor shows a map of the Amazon region, specifically the area around Belém, Brazil. The map includes labels for *Ilha de Marajó*, *Reserva Extrativista Terra Grande - Pracuuba*, *Charapucu*, *Marituba*, *Castanhal*, *Bragança*, *Breves*, and *Estuário do Rio Para*. A blue rectangular region is highlighted on the map, indicating the area of interest for the preprocessing steps.

8. Interpolate bathymetric data

+ 1 cell hidden

9. Load a previously generated Mohid griddata file

Status Bar:

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 9, Col 24 MOHID_Preprocessing.ipynb 0

MOHID Preprocessing Notebook


localhost:8888/lab/tree/MOHID_Preprocessing.ipynb#8.-Interpolate-bathymetric-data

File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Preprocessing.ipynb +

Python 3 (ipykernel)

Total time: 0.80 seconds



ipyleaflet | © OpenStreetMap contributors

11. Save the griddata to a MOHID-compatible file

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 MOHID_Preprocessing.ipynb 0

MOHID Preprocessing Notebook

MOHID-Water-Modelling-System/ x MOHID_Prepro... - JupyterLab x +

localhost:8888/lab/tree/MOHID_Preprocessing.ipynb#8.-Interpolate-bathymetric-data

File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Preprocessing.ipynb +

Code Python 3 (ipykernel)

Cell (47, 47) Depth -99.0:

Submit

Total time: 0.80 seconds

Submit

ipyleaflet | © OpenStreetMap contributors

Simple 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 17, Col 17 MOHID_Preprocessing.ipynb 0

MOHID Preprocessing Notebook

localhost:8888/lab/tree/MOHID_Preprocessing.ipynb#8.-Interpolate-bathymetric-data

File Edit View Run Kernel Tabs Settings Help

Launcher

MOHID_Preprocessing.ipynb

Code

Notebook Python 3 (ipykernel)

Cell (47, 47) Depth -99.0: 5

Submit

Total time: 0.80 seconds

ipyleaflet | © OpenStreetMap contributors

Simple 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 17, Col 17 MOHID_Preprocessing.ipynb 0

MOHID Preprocessing Notebook

localhost:8888/lab/tree/MOHID_Preprocessing.ipynb#8.-Interpolate-bathymetric-data

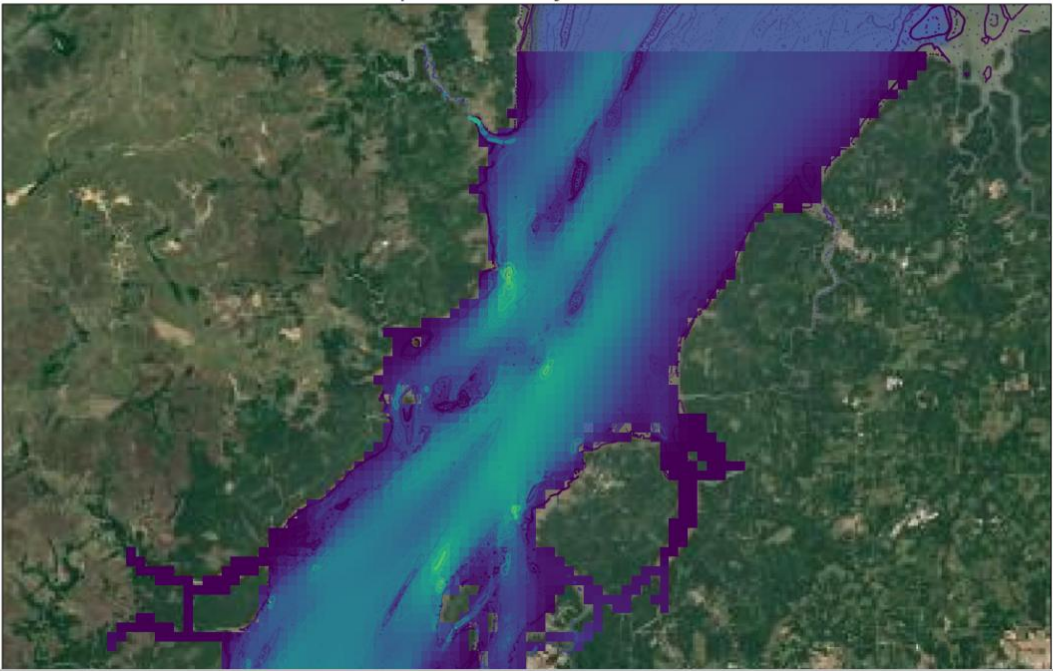
File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Preprocessing.ipynb +

Code Python 3 (ipykernel)

Automatically calculated zoom level: 9

Interpolated Bathymetric Data



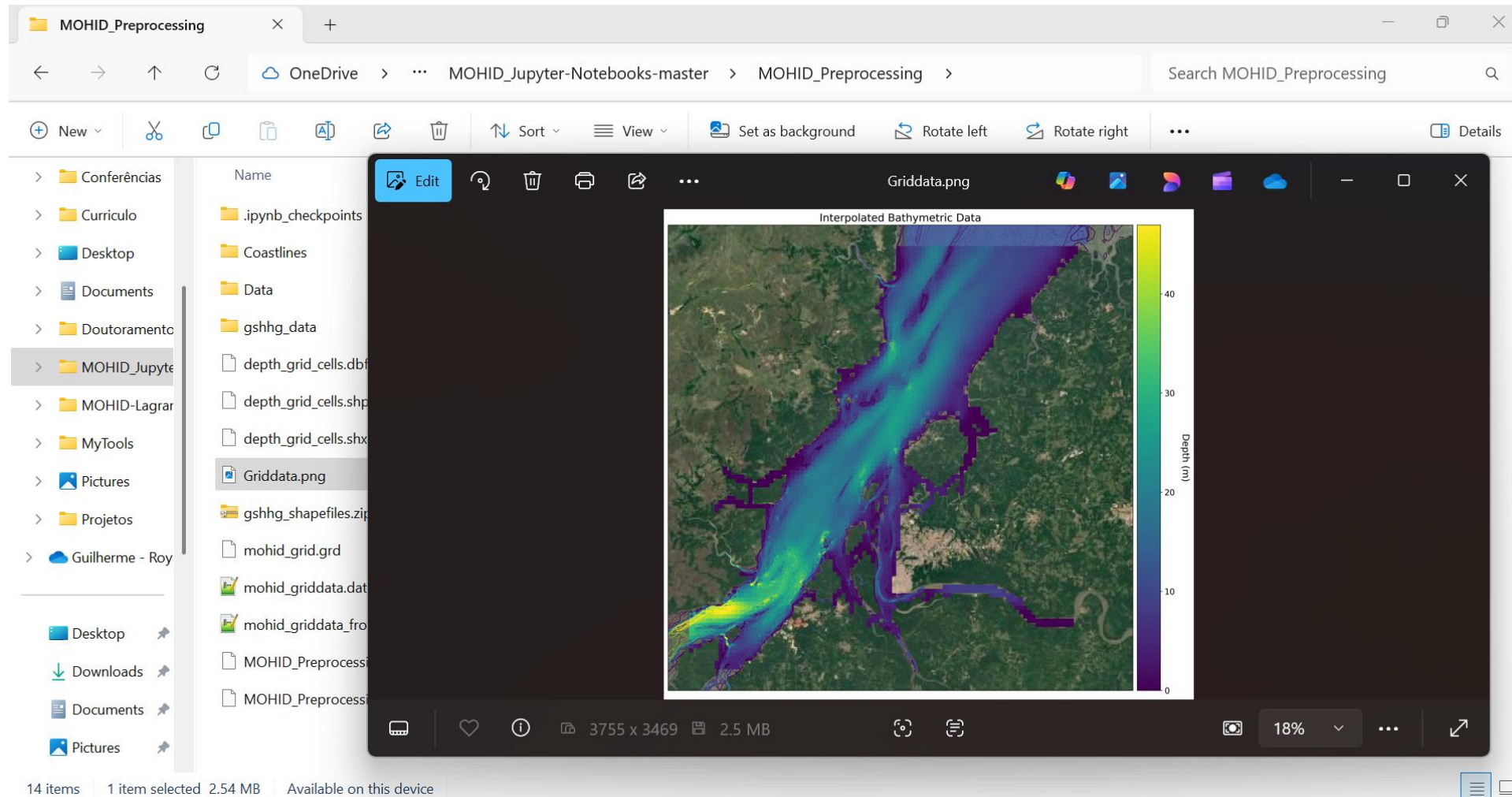
Depth (m)

Name	Mo...	Size
Coastlines	43m ago	
Data	43m ago	
gshhg_data	2s ago	
MOHID_Preprocessin...	now	49.1 KB
depth_grid_cells.dbf	now	1.4 MB
depth_grid_cells.shp	now	1.3 MB
depth_grid_cells.shx	now	78.2 KB
Griddata.png	2s ago	2.5 MB
gshhg_shapefiles.zip	3h ago	142.2 MB
mohid_grid.grd	2h ago	628 B
mohid_griddata_from...	28m ago	64.9 KB
mohid_griddata.dat	55m ago	64.9 KB
MOHID_Preprocessin...	yesterday	252 B

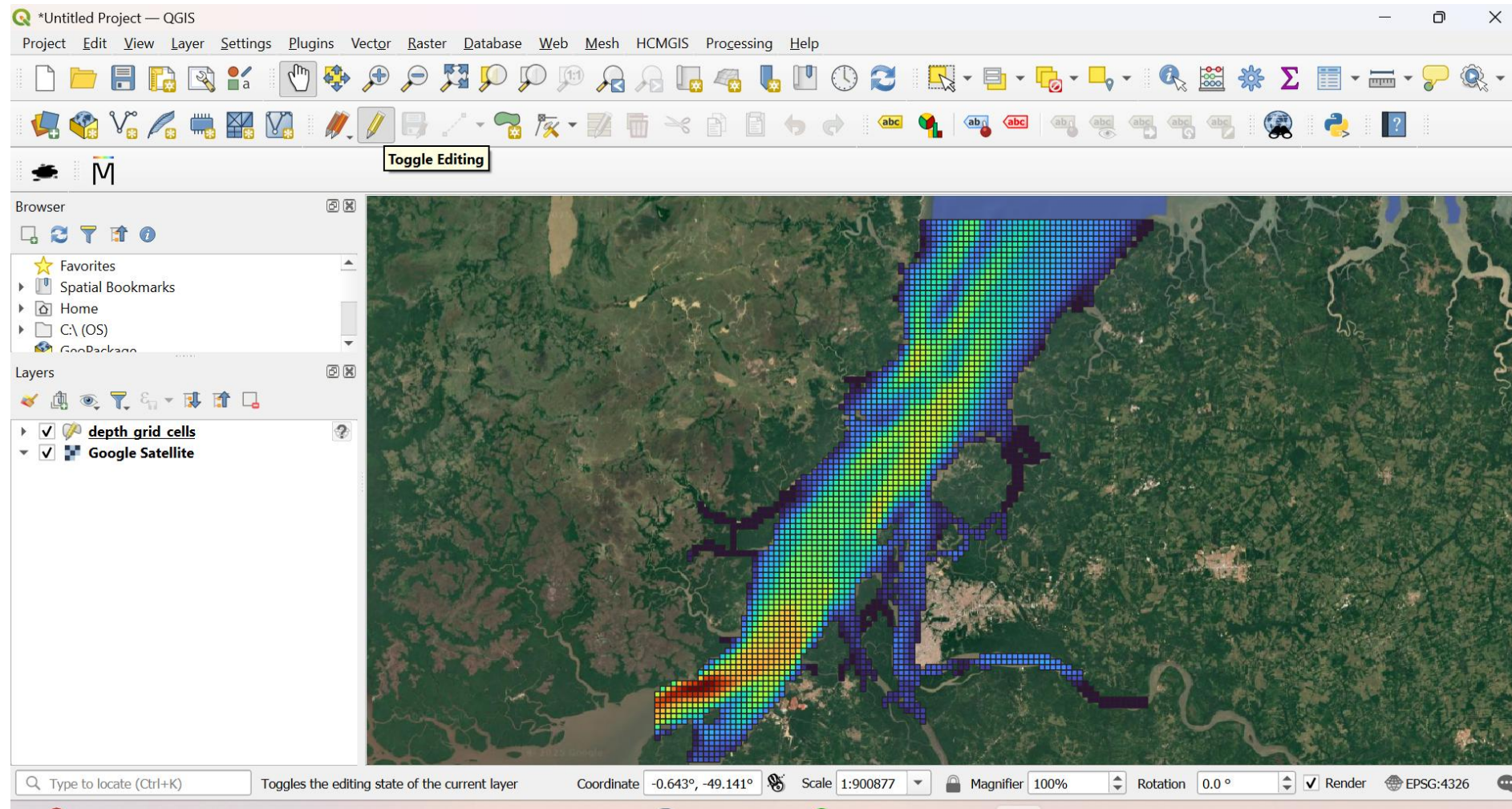
Simple 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 1, Col 1 MOHID_Preprocessing.ipynb 0

MOHID Preprocessing Notebook



MOHID Preprocessing Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab environment with the MOHID Water Notebook open. The browser address bar shows `localhost:8888/lab/tree/MOHID_Water.ipynb`. The notebook's title bar indicates it is a Python 3 (ipykernel) notebook. The left sidebar shows a file tree with folders `releases`, `run_cases`, and `work`, and files `MOHID_...` (76.2 KB) and `MOHID_...` (337 B). The main notebook area has a title `MOHID Water` and a description: "This Jupyter Notebook aims to help implement and run the MOHID Water model." Below this, two notes are provided: **Note 1:** Execute each cell through the `▶` button from the top MENU (or keyboard shortcut `Shift + Enter`). **Note 2:** Use the Kernel and Cell menus to restart the kernel and clear outputs. A **Table of contents** section lists the notebook's structure:

- 1. [Import required libraries](#)
- 2. [General options](#)
 - 2.1 [Set run case](#)
 - 2.2 [Load MOHID griddata](#)
 - 2.3 [Plot MOHID griddata](#)
 - 2.4 [Define a bounding box](#)
 - 2.5 [Set dates](#)
- 3. [Boundary Conditions](#)
 - 3.1 [Oceanic](#)
 - 3.1.1 [Create Copernicus Marine credentials file](#)

The status bar at the bottom shows the kernel is `Python 3 (ipykernel)` and is in `Idle` mode. The current cell is `Ln 1, Col 1` in the `MOHID_Water.ipynb` file.

MOHID Water Notebook

The screenshot shows the JupyterLab interface for the MOHID Water Notebook. The browser address bar displays `localhost:8888/lab/tree/MOHID_Water.ipynb`. The JupyterLab interface includes a file explorer on the left, a central notebook editor, and a bottom status bar.

File Explorer (Left Sidebar):

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	2d ago	76.2 KB
MOHID_...	2d ago	337 B

Notebook Editor (Main Area):

Table of contents

- 1. Import required libraries
- 2. General options
 - 2.1 Set run case
 - 2.2 Load MOHID griddata
 - 2.3 Plot MOHID griddata
 - 2.4 Define a bounding box
 - 2.5 Set dates
- 3. Boundary Conditions
 - 3.1 Oceanic
 - 3.1.1 Create Copernicus Marine credentials file
 - 3.1.2 Set CMEMS product
 - 3.1.3 Download CMEMS
 - 3.1.4 Plot CMEMS
 - 3.2 Meteorological
 - 3.2.1 Setup the CDS API personal access token
 - 3.2.2 Download ERA5 Reanalysis
 - 3.2.3 Plot ERA5
 - 3.3 Tide
 - 3.3.1 Download FES2014.zip
 - 3.3.2 Crop FES2014.hdf5 to your grid area
 - 3.3.3 Plot a specific dataset (e.g., M2 amplitude)
 - 3.4 Rivers

Status Bar (Bottom): Simple ☐ 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 MOHID_Water.ipynb 0

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the local path: `localhost:8888/lab/tree/MOHID_Water.ipynb`.

File Browser (Left Panel):

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	2d ago	76.2 KB
MOHID_...	2d ago	337 B

Notebook Editor (Center):

The notebook is titled "MOHID_Water.ipynb" and is in "Python 3 (ipykernel)" mode. The table of contents on the left lists the following sections:

- 3.3.3 Plot a specific dataset (e.g., M2 amplitude)
- 3.4 Rivers
 - 3.4.1 Download river data
 - 3.4.2 Draw markers on the map to define the river coordinates
 - 3.4.3 Create river data file in MOHID format
- 4. Setup MOHID Water input files
 - 4.1 Model
 - 4.2 Discharges
- 5. Run MOHID Water
 - 5.1 Install MSMPI (Windows)
 - 5.2 Start Simulation
- 6. Visualize results

The main content area shows the following sections:

- 1. Import required libraries**
+ 1 cell hidden
- 2. General options**
 - 2.1 Set run case**
+ 1 cell hidden

Status Bar (Bottom): Simple | 1 | Python 3 (ipykernel) | Idle | Mode: Command | Ln 1, Col 1 | MOHID_Water.ipynb | 0

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the local path: `localhost:8888/lab/tree/MOHID_Water.ipynb`. The left sidebar contains a file explorer with a table of files and folders:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	622.6 KB
MOHID_...	2d ago	337 B

The main notebook area shows the following structure:

- 3.1 Oceanic**
 - 3.1.1 Create Copernicus Marine credentials file**

#It has to be done only once!

```
[*]: #The login command will check your Copernicus Marine credentials and create the configuration file.
copernicusmarine.login()
```

INFO - 2025-05-30T12:03:53Z - Downloading Copernicus Marine data requires a Copernicus Marine username and password, sign up for free at: <https://data.marine.copernicus.eu/register>

Copernicus Marine username:
 - 3.1.2 Set CMEMS product**

+ 1 cell hidden
 - 3.1.3 Download CMEMS**
 - 3.1.4 Plot CMEMS**
- 3.2 Meteorological**

The bottom status bar indicates the notebook is running on Python 3 (ipykernel) in Command mode, with the cursor at Line 1, Column 51.

MOHID Water Notebook

The screenshot shows the JupyterLab interface for the MOHID Water Notebook. The browser address bar indicates the notebook is running on localhost:8888. The left sidebar displays a file explorer with a tree view showing folders like 'releases', 'run_cases', and 'work', and a table listing files including 'MOHID_...' (623.9 KB) and 'MOHID_...' (337 B). The main notebook area shows a code cell with the following content:

```
[ ]: #daily mean
product_id = ["cmems_mod_glo_phy-cur_anfc_0.083deg_P1D-m", "cmems_mod_glo_phy-so_anfc_0.083deg_P1D-m", "cmems_mod_glo_phy-thetao_anfc_0.083deg_P1D-m"]

start_depth = 0.49402499198913574
end_depth = 5727.9
```

Below the code cell, the notebook displays a table of contents with the following sections:

- 3.1.2 Set CMEMS product
- 3.1.3 Download CMEMS
- 3.1.4 Plot CMEMS
- 3.2 Meteorological
- 3.3 Tide
- 3.4 Rivers
- 4. Setup MOHID Water input files

The bottom status bar shows the notebook is running on Python 3 (ipykernel) in Idle mode, with the cursor at Line 5, Column 1.

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The left sidebar contains a file explorer with a table of files and folders:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	623.9 KB
Y: MOHID_...	2d ago	337 B

The main notebook area shows a code cell titled "3.1.3 Download CMEMS" with the following Python code:

```
[ ]: #This file can later be used as input to CMEMS2HDF5.py for operational purposes
input_file = os.path.join(os.getcwd(),"work","CMEMS","Input_CMEMS2HDF5.py")

with open(input_file, 'w') as file:
    file.write(f"backup_path=r'{backup_path_ocean}'\n")
    file.write(f"daily={daily}\n")
    file.write(f"forecast={forecast}\n")
    file.write(f"number_of_runs={number_of_runs}\n")
    file.write(f"refday_to_start={refday_to_start}\n")
    file.write(f"product_id={product_id}\n")
    file.write(f"start_depth={start_depth}\n")
    file.write(f"end_depth={end_depth}\n")
    file.write(f"min_lon={min_lon}\n")
    file.write(f"max_lon={max_lon}\n")
    file.write(f"min_lat={min_lat}\n")
    file.write(f"max_lat={max_lat}\n")
    file.write(f"start_date_str='{start_date_str}'\n")
    file.write(f"end_date_str='{end_date_str}'\n")

%cd work/CMEMS/
%run CMEMS2HDF5.py

# Return to the original directory
%cd -
```

The bottom status bar indicates the notebook is running on Python 3 (ipykernel) in Idle mode, with the cursor at Line 1, Column 1.

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The left sidebar contains a file explorer with a table of files:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	5h ago	628.7 KB
Y: MOHID_...	2d ago	337 B

The main notebook area shows the following code cells:

```
%cd work/CMEMS/  
%run CMEMS2HDF5.py  
  
# Return to the original directory  
%cd -
```

The output of the code execution is displayed in a red-shaded area:

```
ooks-master\MOHID_Water\work\CMEMS\CMEMS_thetao.nc  
Download successful: CMEMS_thetao.nc  
Downloading: CMEMS_zos.nc for 2025-01-01 to 2025-01-06  
  
INFO - 2025-05-30T12:10:36Z - Selected dataset version: "202406"  
INFO - 2025-05-30T12:10:36Z - Selected dataset part: "default"  
INFO - 2025-05-30T12:10:45Z - Starting download. Please wait...  
  
100% ██████████ 3/3 [00:02<00:00, 1.28s/it]  
  
INFO - 2025-05-30T12:10:48Z - Successfully downloaded to C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\work\CMEMS\CMEMS_zos.nc  
Download successful: CMEMS_zos.nc  
Executing ConvertToHDF5Action_zos.dat...  
Executing ConvertToHDF5Action_cur.dat...  
Executing ConvertToHDF5Action_so.dat...  
Executing ConvertToHDF5Action_thetao.dat...  
C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water
```

Below the code cells, the section header **3.1.4 Plot CMEMS** is visible.

The bottom status bar indicates the notebook is running on Python 3 (ipykernel) in Idle mode, with the cursor at Line 6, Column 35.

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The interface includes a left-hand sidebar with a file explorer showing a directory structure with folders like `releases`, `run_cases`, and `work`, and a file named `MOHID_...` (628.7 KB). The main workspace contains a code editor for the notebook `MOHID_Water.ipynb`, which is currently in 'Command' mode. The code editor shows the following Python code:

```
3.1.4 Plot CMEMS

[*]: variable = "velocity modulus" # Change as needed

# =====
# DEFINE VARIABLE-LABEL DICTIONARY
# =====
variable_label_dict = {
    "velocity modulus": "Velocity Modulus(m/s)",
    "salinity": "Salinity(psu)",
    "temperature": "Temperature(°C)",
    "water level": "Water Level(m)"
}

variable_vector = ["velocity U", "velocity V"]

label = variable_label_dict.get(variable, "Unknown Variable") # Fetch label from dictionary

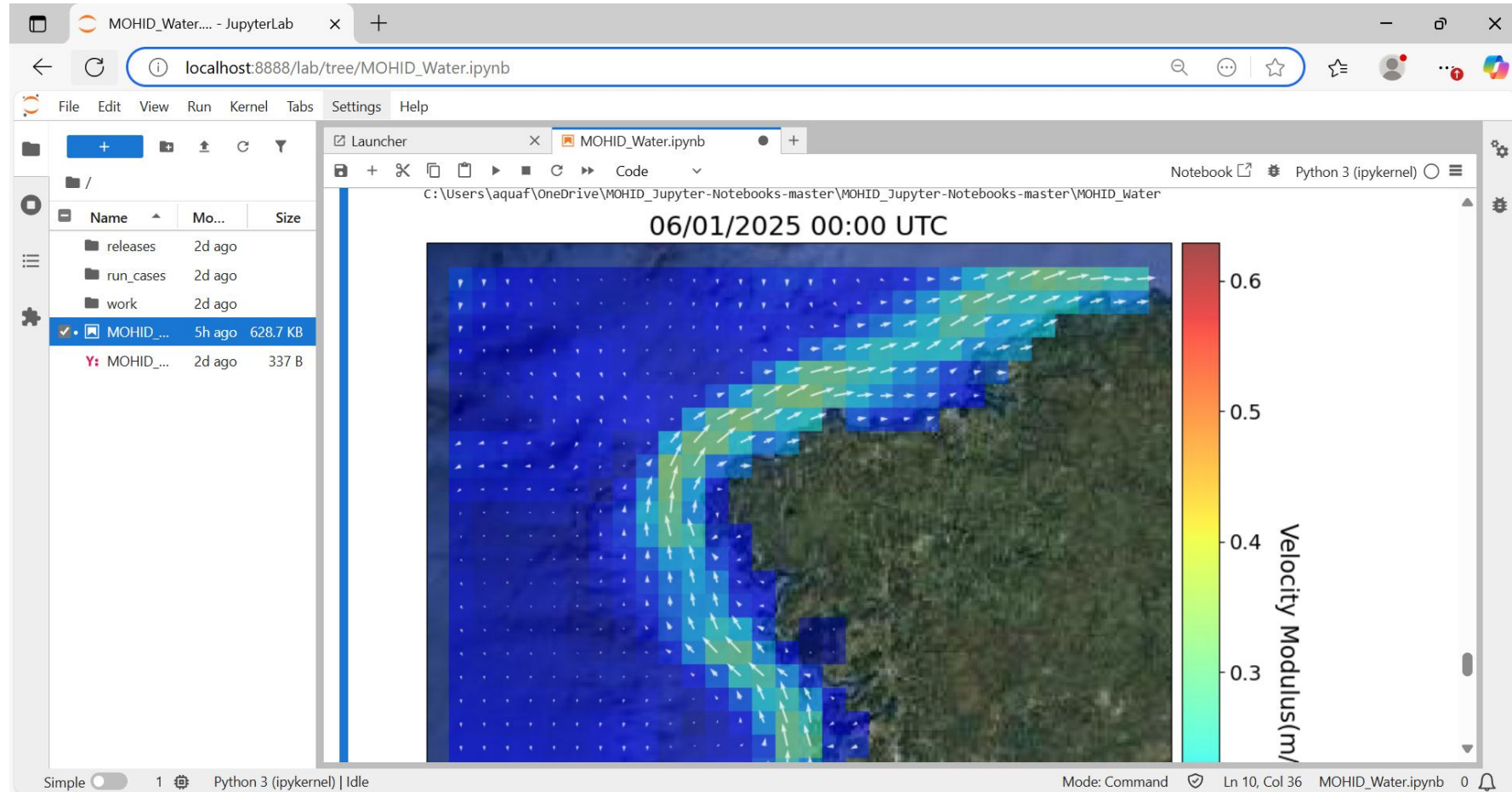
# =====
# SET-UP: Define paths and file names
# =====

start_date = datetime.datetime.strptime(start_date_str, "%Y-%m-%d").date()
end_date = datetime.datetime.strptime(end_date_str, "%Y-%m-%d").date()

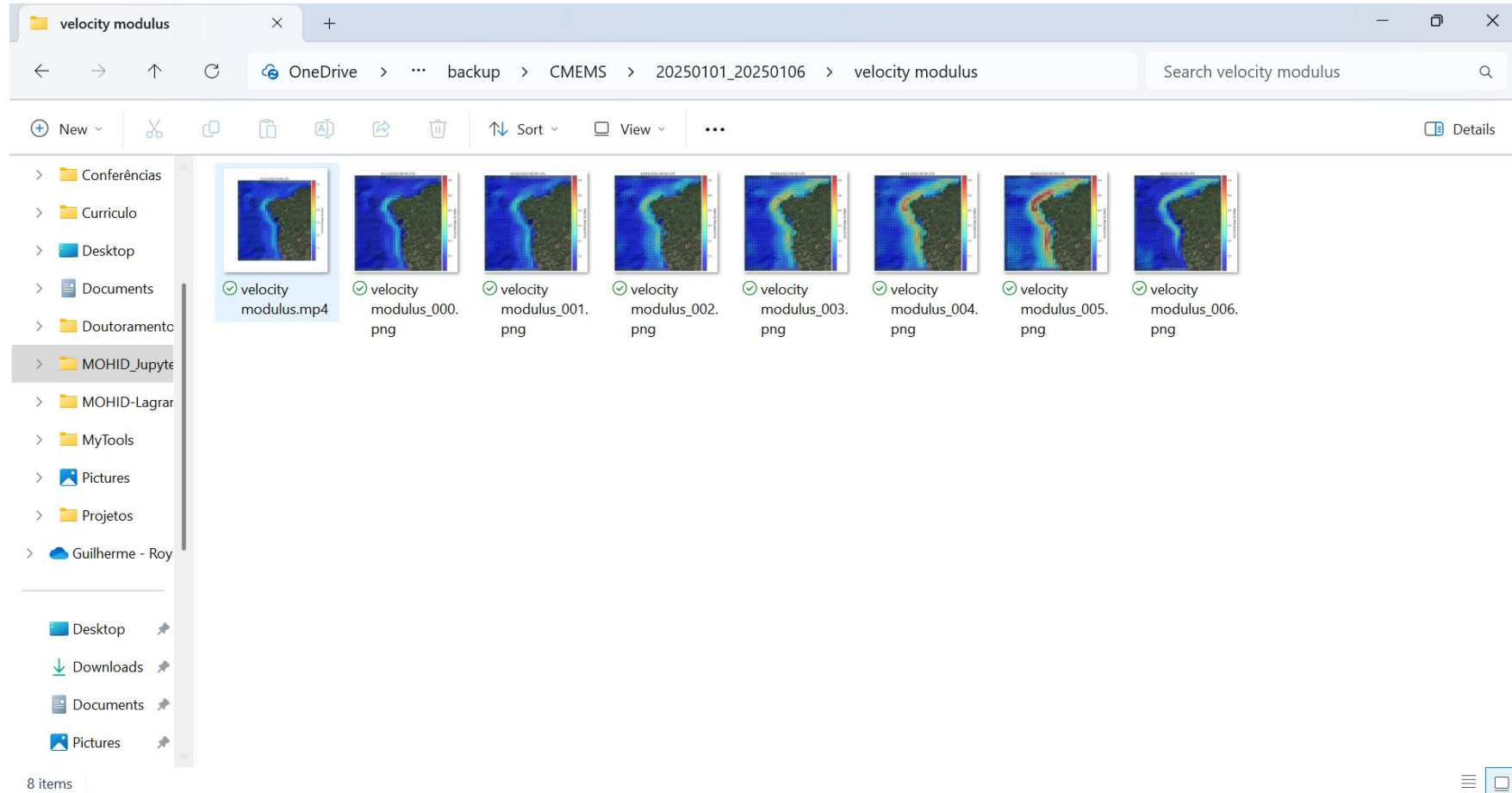
if daily == 1:
    #take the first day
    start_date = start_date + datetime.timedelta(days = 0)
```

The status bar at the bottom indicates the kernel is 'Python 3 (ipykernel)' and is 'Busy'. The mode is 'Command', and the cursor is at 'Ln 1, Col 1'.

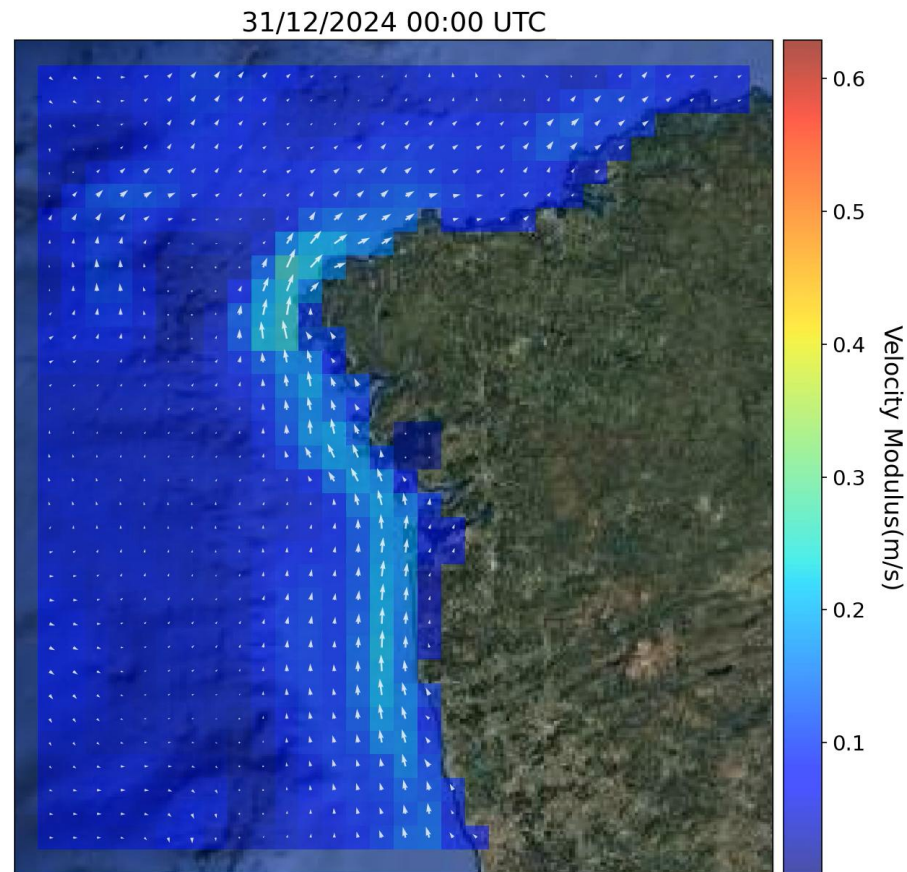
MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab environment with the MOHID Water Notebook open. The browser address bar shows `localhost:8888/lab/tree/MOHID_Water.ipynb`. The notebook is titled "MOHID_Water.ipynb" and is running on a Python 3 (ipykernel) environment.

The left sidebar shows a file browser with the following contents:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	1.2 MB
Y: MOHID_...	2d ago	337 B

The main notebook area shows the following content:

3.2 Meteorological

3.2.1 Setup the CDS API personal access token

It has to be done only once!

If you do not have an account yet, please register (<https://cds.climate.copernicus.eu/>). If you are not logged in, please login. Once logged in, copy the URL and key.

Create a file named `.cdsapirc` in your home directory.

`$HOME/.cdsapirc` (in your Unix/Linux environment)

`%USERPROFILE%.cdsapirc` file (in your windows environment, `%USERPROFILE%` is usually located at `C:\Users\Username` folder).

Paste the URL and key into `.cdsapirc` file.

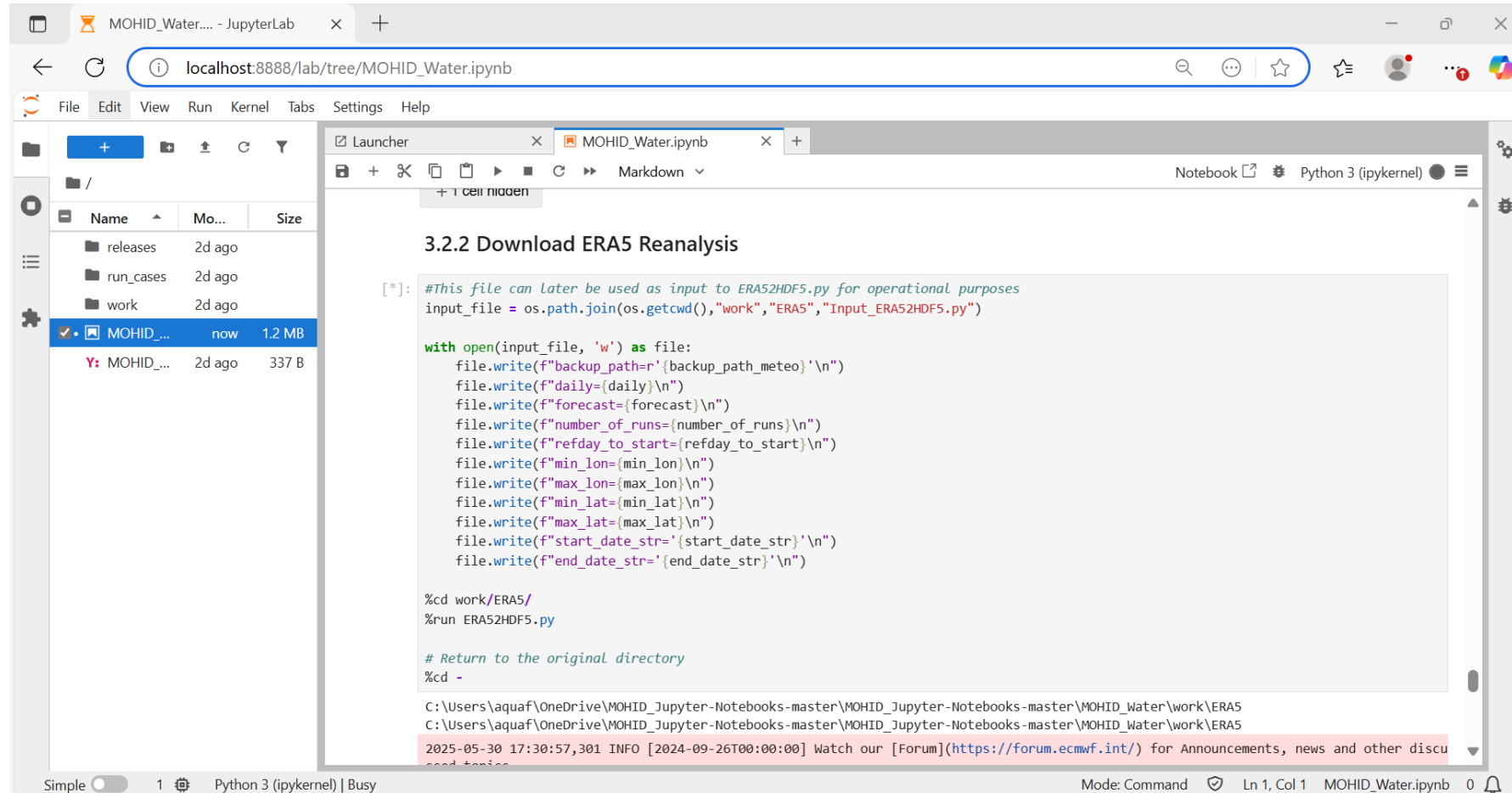
The CDS API expects to find the `.cdsapirc` file in your home directory.

3.2.2 Download ERA5 Reanalysis

3.2.3 Plot ERA5

The bottom status bar indicates the notebook is in "Simple" mode, running on "Python 3 (ipykernel)", and shows the current position as "Ln 1, Col 1".

MOHID Water Notebook



The screenshot displays a JupyterLab environment. The left sidebar features a file explorer showing a directory structure with folders 'releases', 'run_cases', and 'work', and files 'MOHID_...' (1.2 MB) and 'MOHID_...' (337 B). The main area shows a notebook titled 'MOHID_Water.ipynb' with a code cell containing the following text:

```
[*]: #This file can later be used as input to ERA52HDF5.py for operational purposes
input_file = os.path.join(os.getcwd(), "work", "ERA5", "Input_ERA52HDF5.py")

with open(input_file, 'w') as file:
    file.write(f"backup_path=r'{backup_path_meteo}'\n")
    file.write(f"daily={daily}\n")
    file.write(f"forecast={forecast}\n")
    file.write(f"number_of_runs={number_of_runs}\n")
    file.write(f"refday_to_start={refday_to_start}\n")
    file.write(f"min_lon={min_lon}\n")
    file.write(f"max_lon={max_lon}\n")
    file.write(f"min_lat={min_lat}\n")
    file.write(f"max_lat={max_lat}\n")
    file.write(f"start_date_str='{start_date_str}'\n")
    file.write(f"end_date_str='{end_date_str}'\n")

%cd work/ERA5/
%run ERA52HDF5.py

# Return to the original directory
%cd -
```

The output of the code cell shows the current directory path: C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\work\ERA5. The status bar at the bottom indicates the notebook is running on Python 3 (ipykernel) and is in Command mode.

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The JupyterLab interface includes a top menu bar with options like File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a file browser sidebar shows a directory structure with folders `releases`, `run_cases`, and `work`, and files `MOHID_...` (26m ago, 1.2 MB) and `Y: MOHID_...` (2d ago, 337 B). The main workspace contains a code editor for `MOHID_Water.ipynb` with a Python 3 (ipykernel) environment. The notebook title is `3.2.3 Plot ERA5`. The code in the cell is as follows:

```
[*]: variable = "wind velocity" # Change as needed

# =====
# VARIABLE-LABEL DICTIONARY
# =====
variable_label_dict = {
    "wind velocity": "Wind Velocity(m/s)",
    "air temperature": "Temperature(°C)",
    "solar radiation": "Solar Radiation(W/m²)",
    "albedo": "Albedo(-)",
    "atmospheric pressure": "Atmospheric Pressure(Pa)",
    "cloud cover": "Cloud Cover(-)",
    "dew point": "dew point(°C)",
    "downward long wave radiation": "Downward Long Wave Radiation(W/m²)",
    "pbl height": "Pbl Height(m)",
    "precipitation": "Precipitation(mm/h)",
    "relative humidity": "Relative humidity(-)"
}

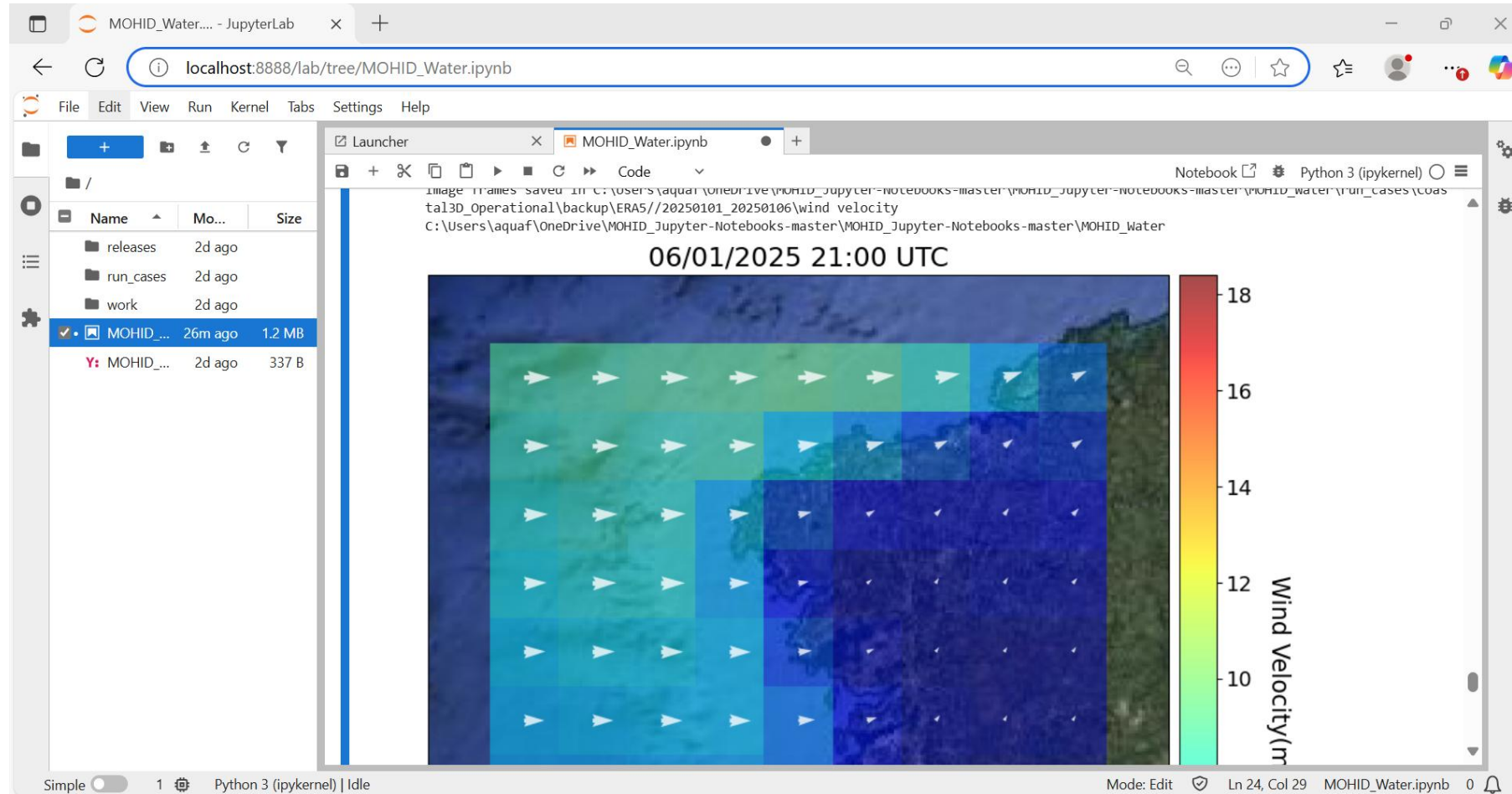
variable_vector = ["wind velocity X", "wind velocity Y"]

label = variable_label_dict.get(variable, "Unknown Variable") # Fetch Label from dictionary
# =====
# SET-UP: Define paths and file names
# =====

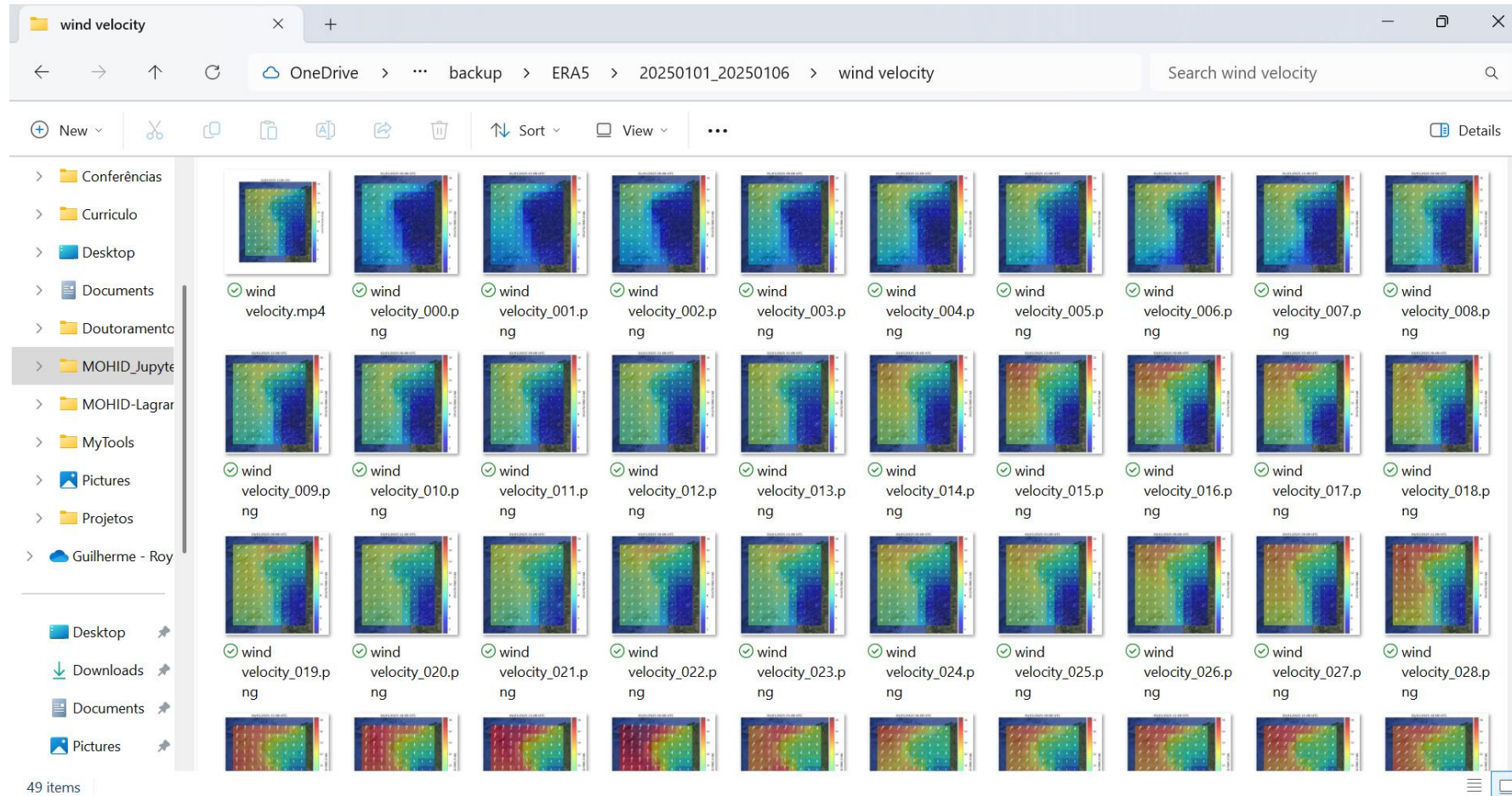
start_date = datetime.datetime.strptime(start_date_str, "%Y-%m-%d").date()
```

The status bar at the bottom indicates the notebook is in 'Simple' mode, running on 'Python 3 (ipykernel)' with a 'Busy' status. The current mode is 'Command', and the cursor is at 'Ln 1, Col 1' in the file `MOHID_Water.ipynb`.

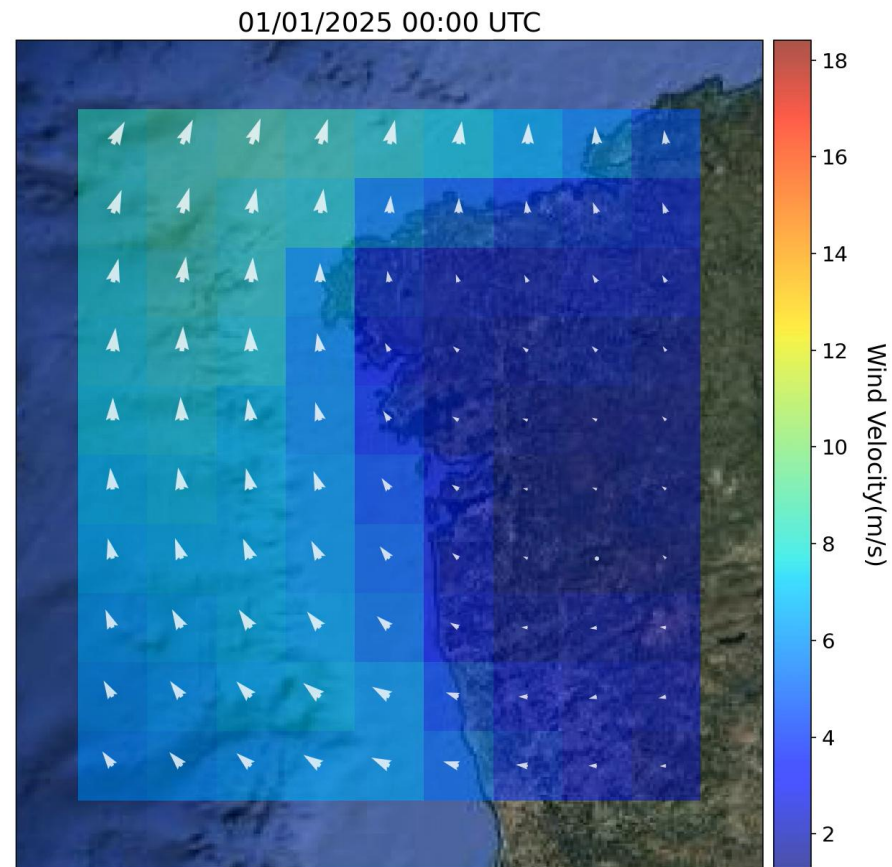
MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The left sidebar contains a file explorer with a table of files:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	1.7 MB
Y: MOHID_...	2d ago	337 B

The main notebook area shows the following content:

3.3 Tide

3.3.1 Download FES2014.zip

It has to be done only once!

```
[ ]: # URL of the ZIP file.
url = "http://www.mohid.com/PublicData/Products/Software/FES2014.zip"

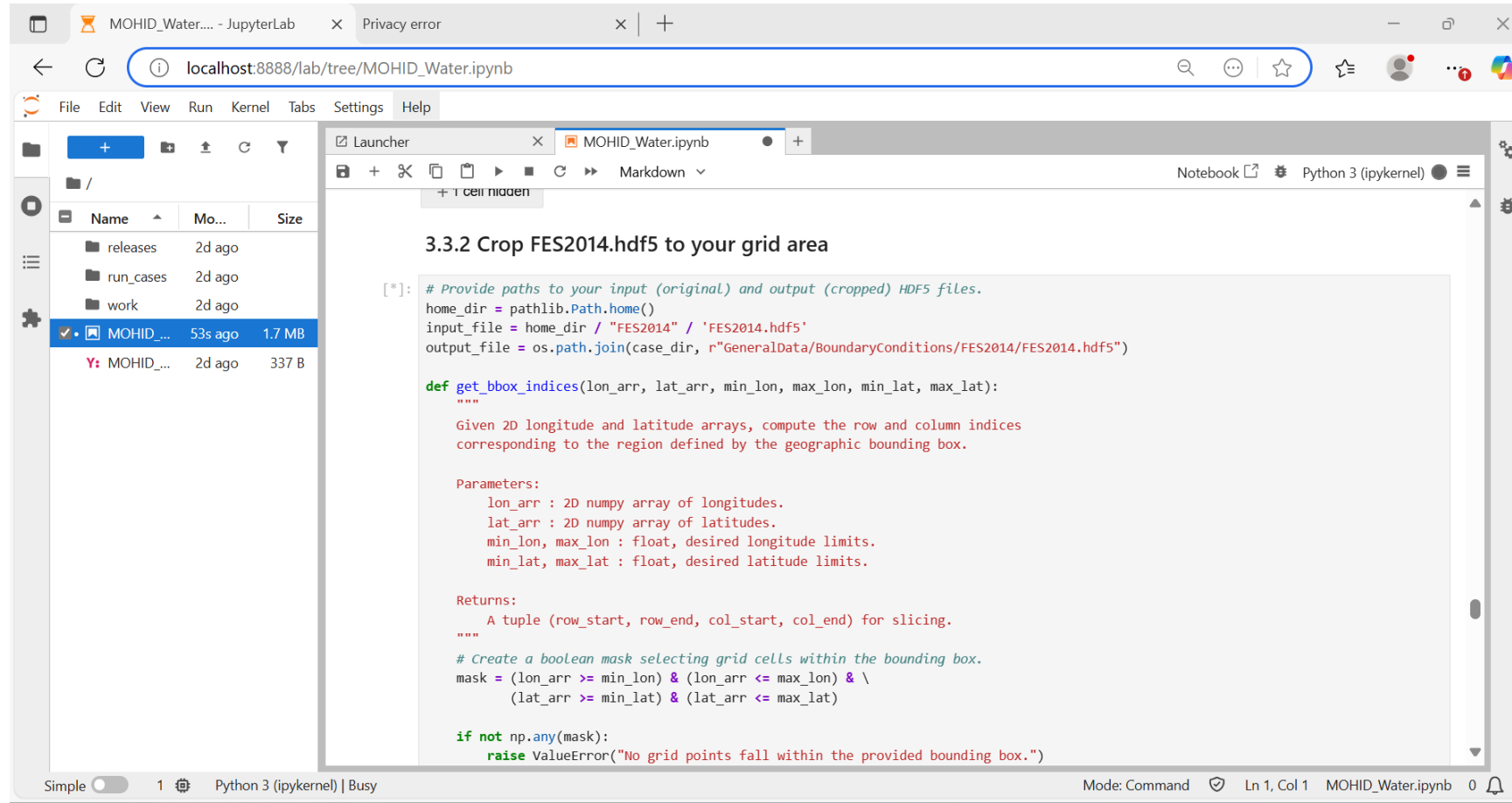
# Define paths using pathlib for cross-platform compatibility.
home_dir = pathlib.Path.home()
local_zip_path = home_dir / "FES2014.zip"
extract_dir = home_dir / "FES2014"

def download_file(url, save_path):
    try:
        response = requests.get(url, stream=True)
        response.raise_for_status() # Raise exception for HTTP errors
        total_size = int(response.headers.get('content-length', 0))

        with open(save_path, 'wb') as f, tqdm(
            desc="Downloading",
            total=total_size,
            unit='B',
            unit_scale=True,
            unit_divisor=1024
        ) as bar:
            for chunk in response.iter_content(chunk_size=8192):
```

The status bar at the bottom indicates the notebook is running on Python 3 (ipykernel) in Idle mode, with the cursor at Line 1, Column 1.

MOHID Water Notebook



The screenshot displays the JupyterLab interface in a web browser. The address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The interface includes a top menu bar with options like File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a file explorer sidebar shows a directory structure with folders like `releases`, `run_cases`, and `work`, and files like `MOHID_...`. The main area contains a notebook titled `MOHID_Water.ipynb` with a code cell. The code cell is titled `3.3.2 Crop FES2014.hdf5 to your grid area` and contains Python code for handling HDF5 files and geographic bounding boxes.

```
[*]: # Provide paths to your input (original) and output (cropped) HDF5 files.
home_dir = pathlib.Path.home()
input_file = home_dir / "FES2014" / "FES2014.hdf5"
output_file = os.path.join(case_dir, r"GeneralData/BoundaryConditions/FES2014/FES2014.hdf5")

def get_bbox_indices(lon_arr, lat_arr, min_lon, max_lon, min_lat, max_lat):
    """
    Given 2D longitude and latitude arrays, compute the row and column indices
    corresponding to the region defined by the geographic bounding box.

    Parameters:
        lon_arr : 2D numpy array of longitudes.
        lat_arr : 2D numpy array of latitudes.
        min_lon, max_lon : float, desired longitude limits.
        min_lat, max_lat : float, desired latitude limits.

    Returns:
        A tuple (row_start, row_end, col_start, col_end) for slicing.
    """
    # Create a boolean mask selecting grid cells within the bounding box.
    mask = (lon_arr >= min_lon) & (lon_arr <= max_lon) & \
           (lat_arr >= min_lat) & (lat_arr <= max_lat)

    if not np.any(mask):
        raise ValueError("No grid points fall within the provided bounding box.")
```

The bottom status bar indicates the current mode is 'Command', the kernel is 'Python 3 (ipykernel)', and the file is `MOHID_Water.ipynb`.

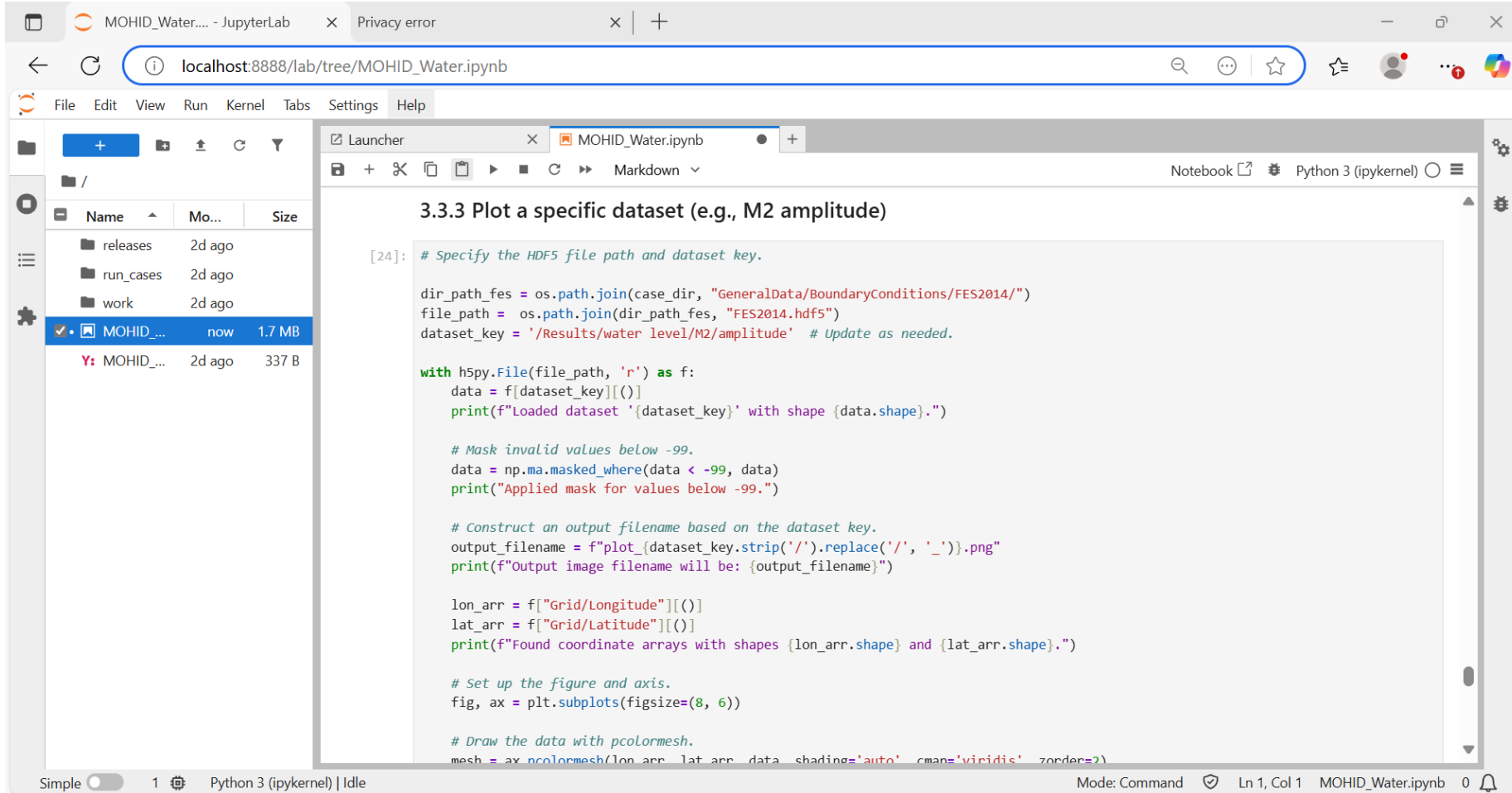
MOHID Water Notebook

The screenshot displays a Windows File Explorer window titled 'FES2014'. The address bar shows the path: OneDrive > Coastal3D_Operational > GeneralData > BoundaryConditions > FES2014. The search bar contains 'Search FES2014'. The ribbon includes 'New', 'Cut', 'Copy', 'Paste', 'Share', 'Delete', 'Sort', 'View', and 'Details'. The left sidebar shows a list of folders: Conferências, Currículo, Desktop, Documents, Doutoramento, MOHID_Jupyter (selected), MOHID-Lagrar, MyTools, Pictures, Projetos, and Guilherme - Roy. Below this is a section with Desktop, Downloads, Documents, and Pictures, each with a star icon. The main pane shows a table with one item:

Name	Status	Date modified	Type	Size
FES2014.hdf5		5/30/2025 4:10 PM	HDF5 Data File	348 KB

The bottom status bar indicates '1 item'.

MOHID Water Notebook



The screenshot displays the JupyterLab interface for a notebook named "MOHID_Water.ipynb". The browser address bar shows "localhost:8888/lab/tree/MOHID_Water.ipynb". The left sidebar features a file browser with a table of files:

Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	1.7 MB
Y: MOHID_...	2d ago	337 B

The main area shows the notebook content, which includes a section titled "3.3.3 Plot a specific dataset (e.g., M2 amplitude)". The code in the cell is as follows:

```
[24]: # Specify the HDF5 file path and dataset key.

dir_path_fes = os.path.join(case_dir, "GeneralData/BoundaryConditions/FES2014/")
file_path = os.path.join(dir_path_fes, "FES2014.hdf5")
dataset_key = '/Results/water_level/M2/amplitude' # Update as needed.

with h5py.File(file_path, 'r') as f:
    data = f[dataset_key][()]
    print(f"Loaded dataset '{dataset_key}' with shape {data.shape}.")

# Mask invalid values below -99.
data = np.ma.masked_where(data < -99, data)
print("Applied mask for values below -99.")

# Construct an output filename based on the dataset key.
output_filename = f"plot_{dataset_key.strip('/').replace('/', '_')}.png"
print(f"Output image filename will be: {output_filename}")

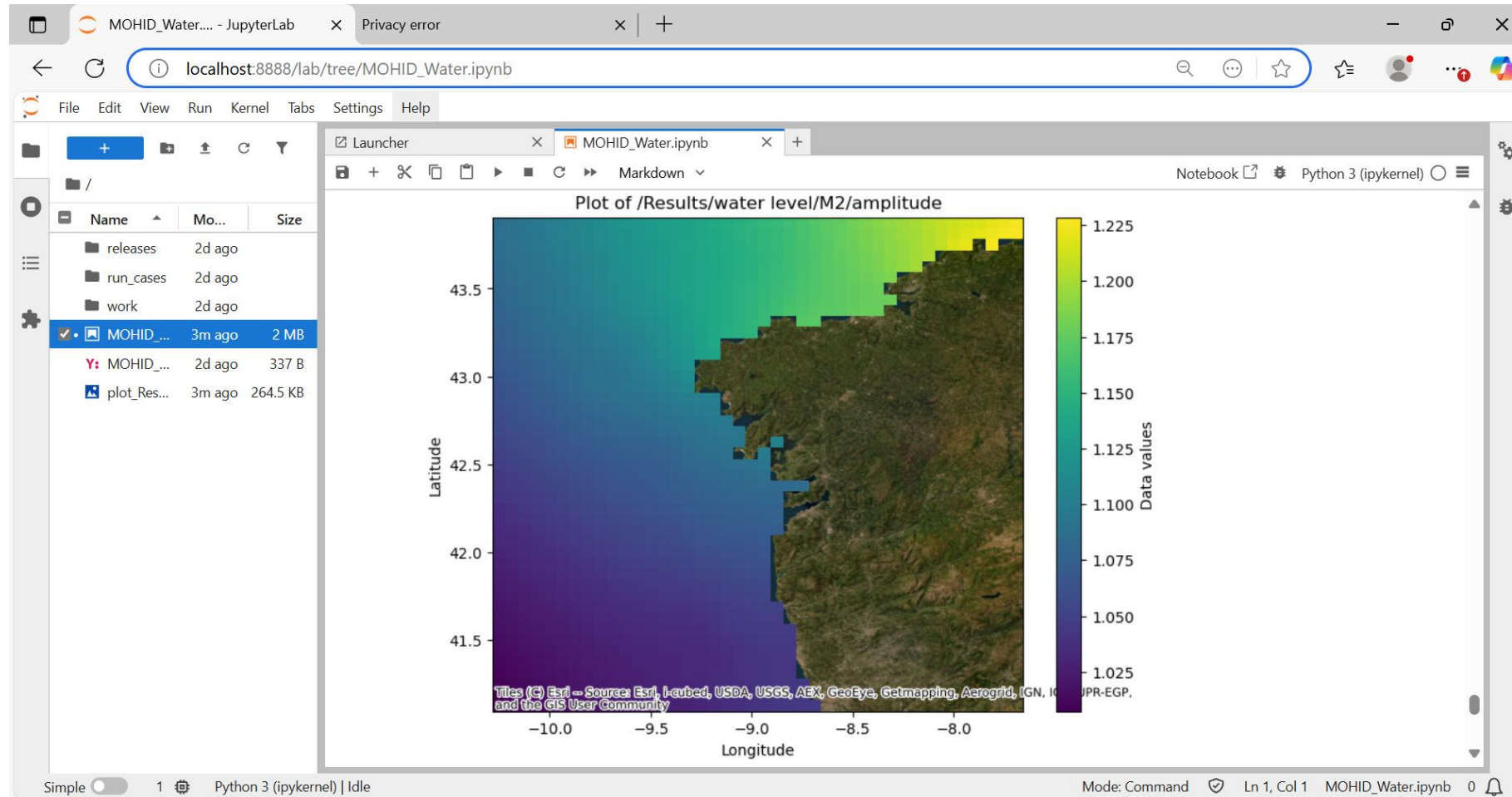
lon_arr = f["Grid/Longitude"][()]
lat_arr = f["Grid/Latitude"][()]
print(f"Found coordinate arrays with shapes {lon_arr.shape} and {lat_arr.shape}.")

# Set up the figure and axis.
fig, ax = plt.subplots(figsize=(8, 6))

# Draw the data with pcolormesh.
mesh = ax.pcolormesh(lon_arr, lat_arr, data, shading='auto', cmap='viridis', zorder=2)
```

The status bar at the bottom indicates "Simple" mode, "Python 3 (ipykernel) | Idle", and "Mode: Command". The notebook title "MOHID_Water.ipynb" and a bell icon are also visible.

MOHID Water Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab interface for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The left sidebar contains a file explorer with a table of files and folders:

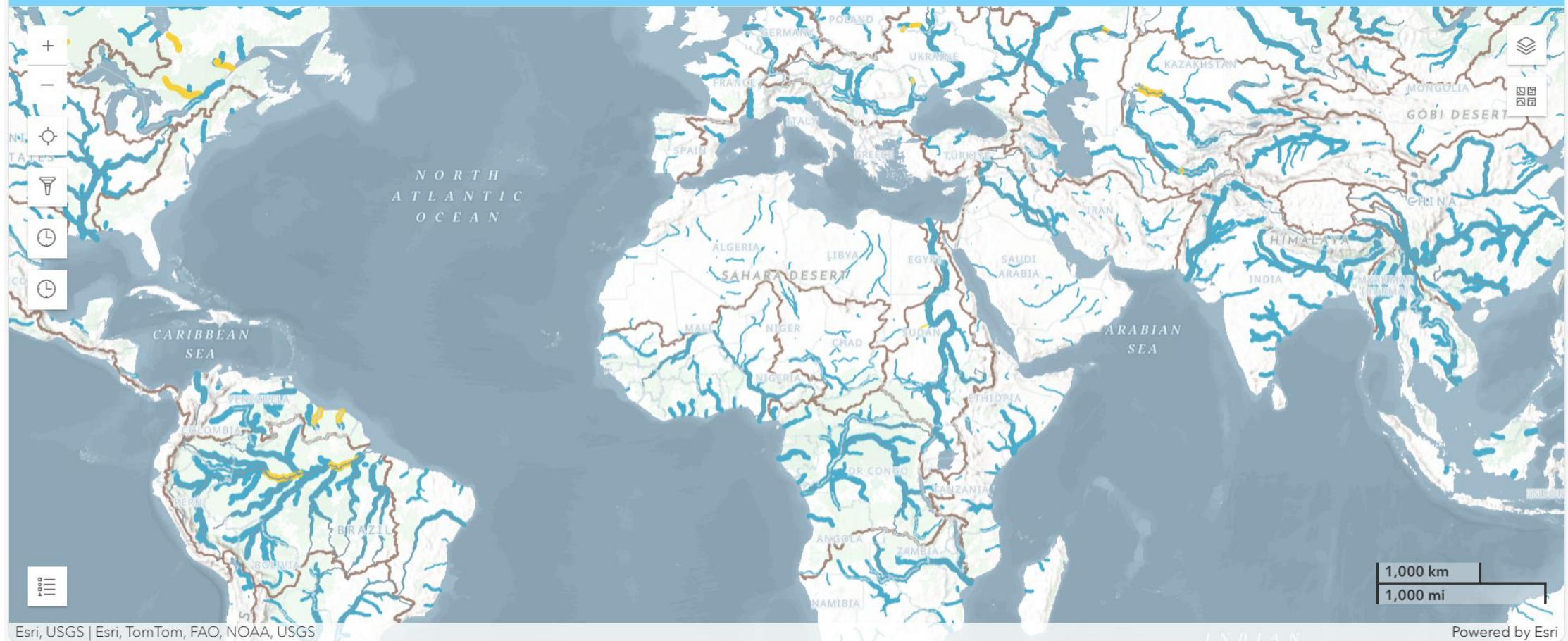
Name	Mo...	Size
releases	2d ago	
run_cases	2d ago	
work	2d ago	
MOHID_...	now	2 MB
Y: MOHID_...	2d ago	337 B

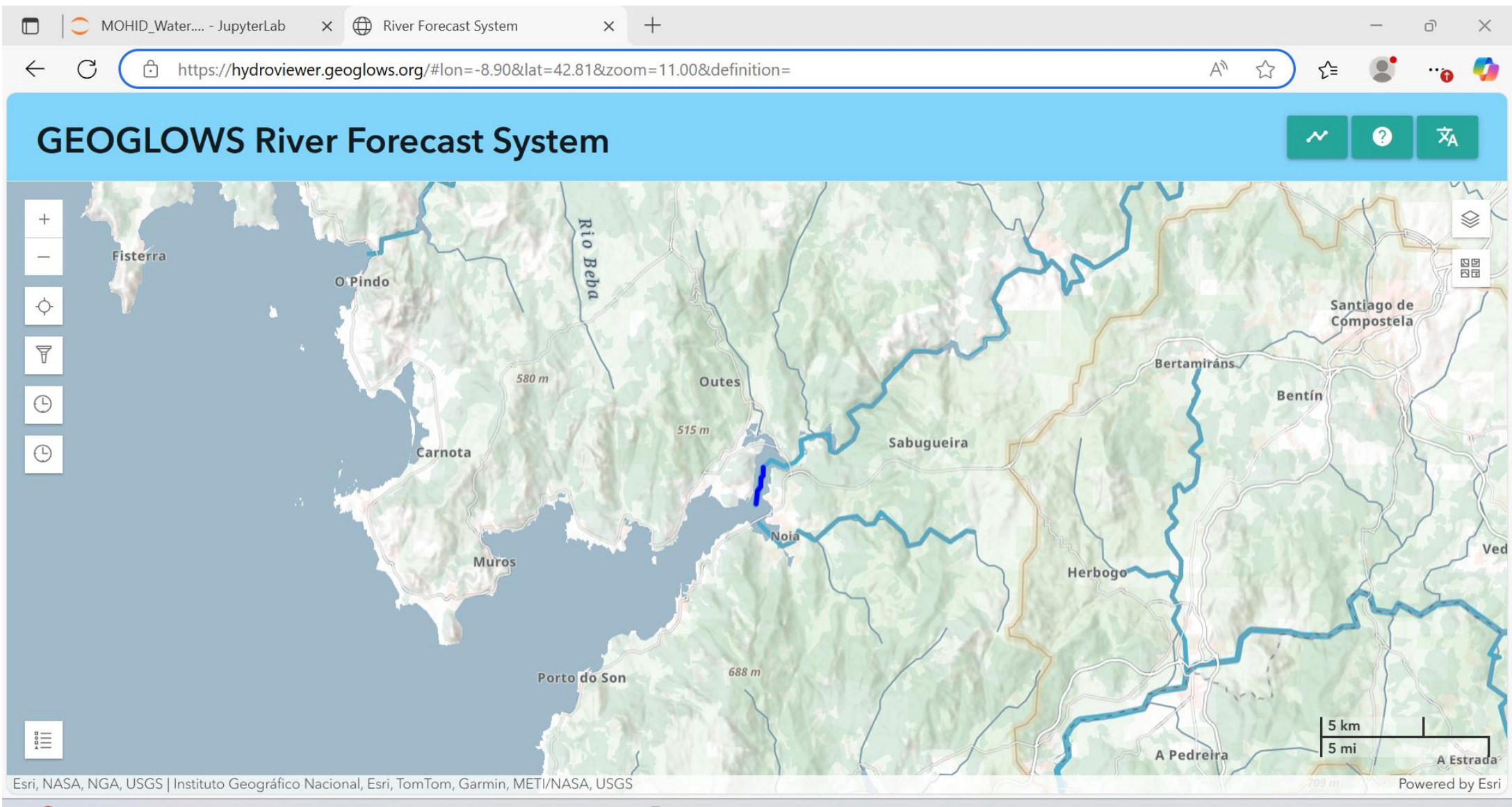
The main notebook area shows the following content:

- 3.4 Rivers**
 - Go to the link below to get river's ID (comid) from GEOGLOWS River Forecast System:
<https://hydroviewer.geogloss.org/#lon=10.00&lat=18.00&zoom=3.00&definition=>
 - 3.4.1 Download river data**
+ 1 cell hidden
 - 3.4.2 Draw markers on the map to define the river coordinates**
+ 2 cells hidden
 - 3.4.3 Create river data file in MOHID format**
+ 1 cell hidden
- 4. Setup MOHID Water input files**
+ 5 cells hidden

The bottom status bar indicates the notebook is running on Python 3 (ipykernel) in Simple mode, with the cursor at Line 1, Column 1.

GEOGLOWS River Forecast System





GEOGLOWS River Forecast System



230182503

Forecast✓

Retrospective✓

Enter River ID

05/30/2025



Retrospective Simulation for River: 230182503

1 Year 5 Years 10 Years 30 Years All

Daily Average Monthly Average



☐ Use Bias Corrected?

SAVE FORECAST DATA (CSV)

SAVE RETROSPECTIVE DATA (CSV)

CLOSE

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb`. The JupyterLab interface includes a left sidebar with a file explorer showing a directory structure with folders like `releases`, `run_cases`, and `work`, and files like `MOHID_...`. The main area shows the notebook content, which is titled "3.4.1 Download river data". The code in the notebook cell is as follows:

```
[ ]: #Define a unique identifier. This ID will be used to retrieve the corresponding hydrological data.
comid = 230182503 #change as needed

discharge_path= (os.path.join(case_dir, "GeneralData", "Discharges"))

# Define the initial date as timezone-aware (UTC)
initial_river_date = pd.Timestamp("2015-01-01", tz="UTC")

simulated_df = geoglows.data.retrospective(comid)
simulated_df.index = pd.to_datetime(simulated_df.index)

# Replace negative values with zero
simulated_df[simulated_df < 0] = 0

# Filter the DataFrame to include only data from initial_date onward
simulated_df = simulated_df[simulated_df.index >= initial_river_date]

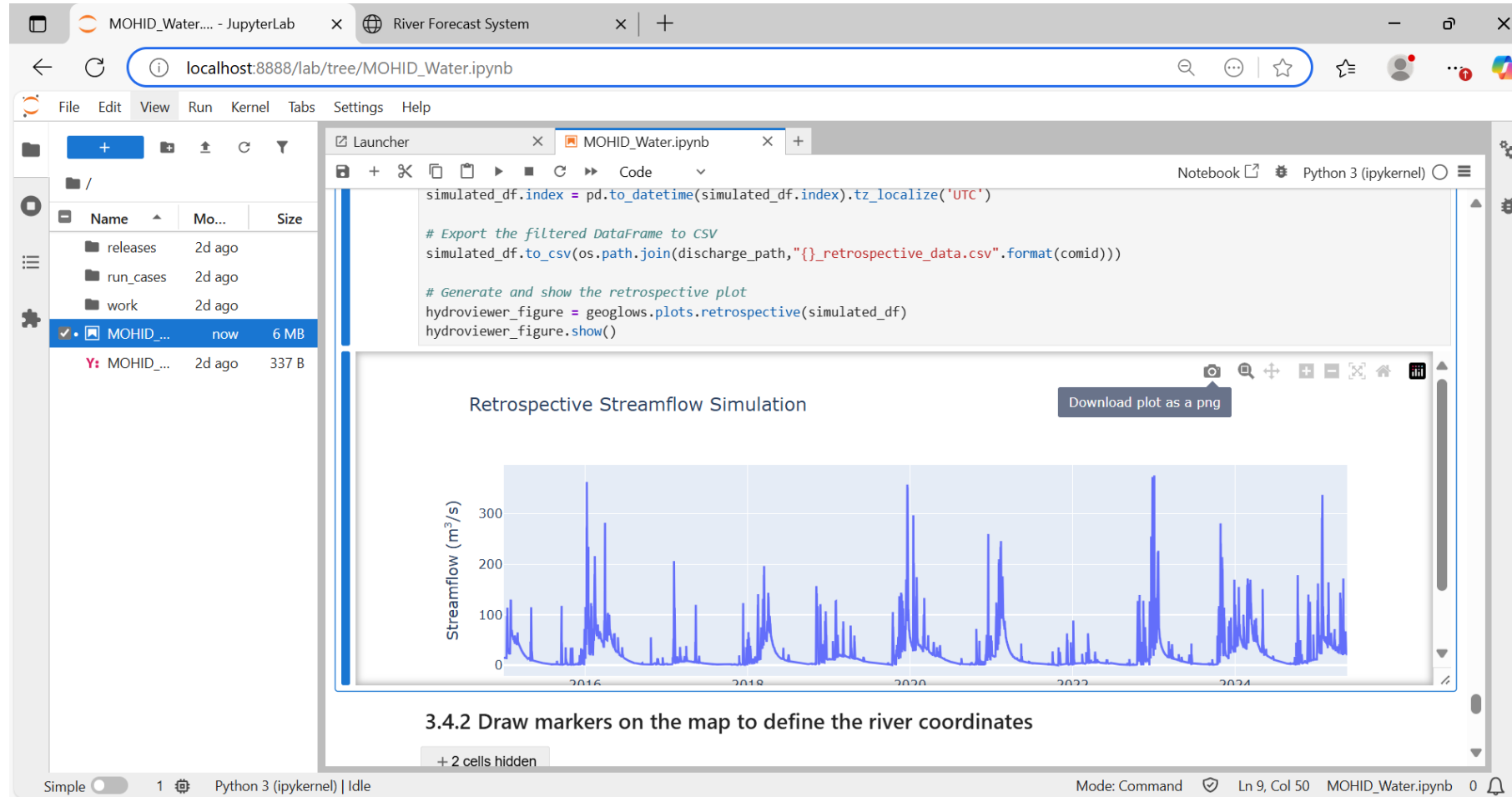
# Reformat the index to remove time-of-day components while keeping timezone info
simulated_df.index = simulated_df.index.to_series().dt.strftime("%Y-%m-%d")
simulated_df.index = pd.to_datetime(simulated_df.index).tz_localize('UTC')

# Export the filtered DataFrame to CSV
simulated_df.to_csv(os.path.join(discharge_path, "{}_retrospective_data.csv".format(comid)))

# Generate and show the retrospective plot
hydroviewer_figure = geoglows.plots.retrospective(simulated_df)
```

The status bar at the bottom indicates the notebook is in "Simple" mode, using "Python 3 (ipykernel)", and the current cell is at "Ln 2, Col 9".

MOHID Water Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the local path: `localhost:8888/lab/tree/MOHID_Water.ipynb`. The interface includes a top menu bar with options like File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left, a file browser sidebar shows the directory structure with folders like 'releases', 'run_cases', and 'work', and files like 'MOHID_...' and 'MOHID_...'. The main area is a code editor for the notebook 'MOHID_Water.ipynb', currently showing a code cell with the title '3.4.2 Draw markers on the map to define the river coordinates'. The code cell contains the following Python code:

```
[27]: import matplotlib.colors as mcolors

# Start timing
start_time = time.time()

LonGrid = np.array(x_grid)
LatGrid = np.array(y_grid)
min_lon, max_lon = LonGrid.min(), LonGrid.max()
min_lat, max_lat = LatGrid.min(), LatGrid.max()

# Create an Output widget to capture the callback prints
output = widgets.Output()
display(output)

# -----
# Create the map.
m = Map(center=(LatGrid.mean(), LonGrid.mean()), zoom=8)
marker = None # For interactive marker

# Store the block (batch) layers in a dictionary, keyed by (block_row, block_col).
block_layers = {}

# Set a block (batch) size. (This controls the spatial grouping.)
block_size = 10 # Adjust as needed.

# Precompute grid cell corners (for all cells).
# These arrays are of shape (M-1, N-1) if LonGrid and LatGrid are shape (M, N).
```

The bottom status bar indicates the current mode is 'Command', the kernel is 'Python 3 (ipykernel)', and the file is 'MOHID_Water.ipynb'.

MOHID Water Notebook

MOHID_Water.... - JupyterLab x River Forecast System x | +

localhost:8888/lab/tree/MOHID_Water.ipynb

File Edit View Run Kernel Tabs Settings Help


Launcher x MOHID_Water.ipynb x +

Notebook Python 3 (ipykernel)

m

Total time: 10.69 seconds

[27]:



[]: `print(markers_dict)`

Simple 1 Python 3 (ipykernel) | Idle Mode: Command Ln 18, Col 40 MOHID_Water.ipynb 0

MOHID Water Notebook

MOHID_Water.... - JupyterLab x River Forecast System x +

localhost:8888/lab/tree/MOHID_Water.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Water.ipynb x +

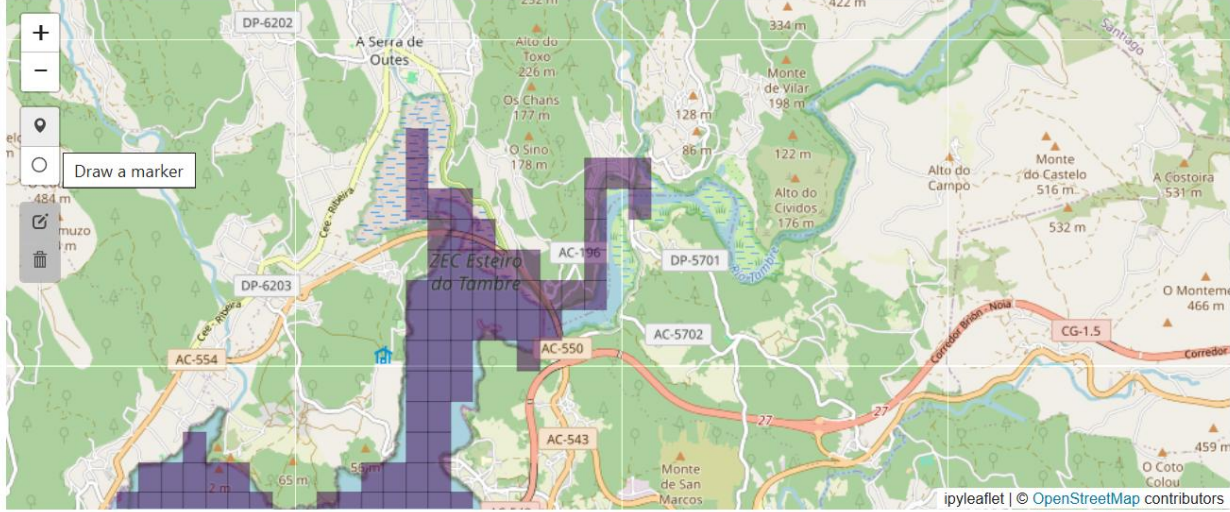
Notebook Python 3 (ipykernel)

20s ago 6 MB

2d ago 337 B

Total time: 10.69 seconds

[27]:



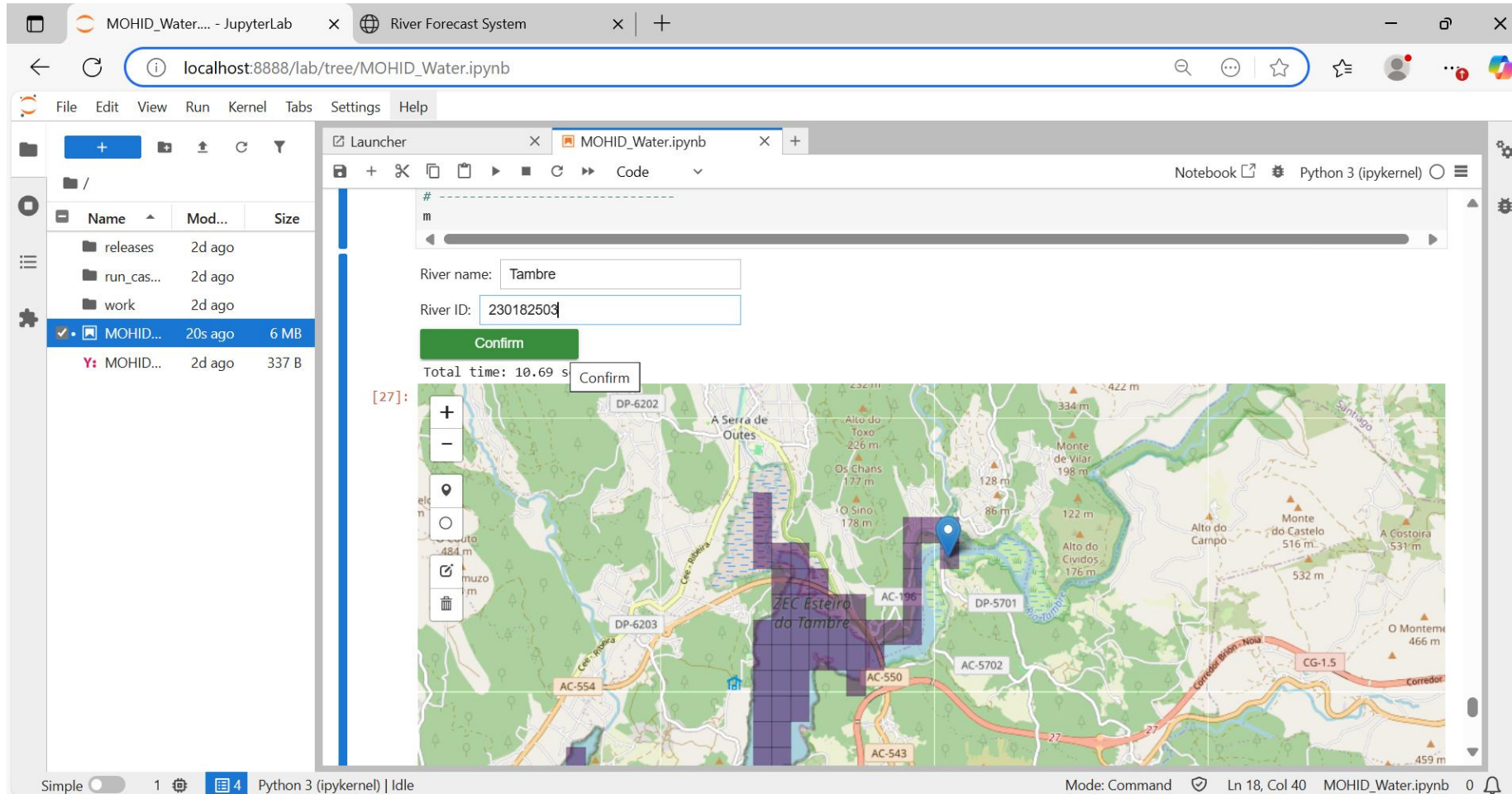
ipyleaflet | © OpenStreetMap contributors

[]: print(markers_dict)

localhost:8888/lab/tree/MOHID_Water.ipynb# | Idle

Mode: Command Ln 18, Col 40 MOHID_Water.ipynb 0

MOHID Water Notebook



MOHID Water Notebook

MOHID_Water.... - JupyterLab x River Forecast System x | +

localhost:8888/lab/tree/MOHID_Water.ipynb

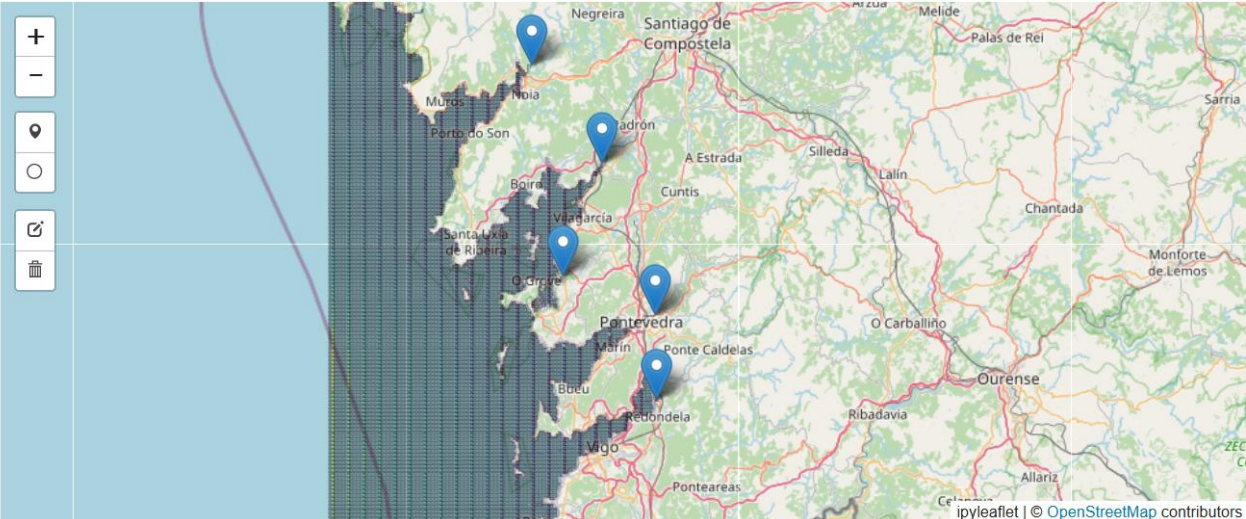
File Edit View Run Kernel Tabs Settings Help

Launcher x MOHID_Water.ipynb

Markdown

Notebook Python 3 (ipykernel)

[27]:



[32]: `print(markers_dict)`

```
{0: {'location': [42.827131, -8.874738], 'name': 'Tambre', 'river_id': '230182503'}, 1: {'location': [42.67707, -8.724964], 'name': 'Ulla', 'river_id': '230161838'}, 2: {'location': [42.500172, -8.80892], 'name': 'Umia', 'river_id': '230109620'}, 3: {'location': [42.43999, -8.614011], 'name': 'Lerez', 'river_id': '230107447'}, 4: {'location': [42.307586, -8.610746], 'name': 'Verdugo', 'river_id': '230112893'}}
```

Simple 1 30 Python 3 (ipykernel) | Idle Saving started Mode: Command Ln 1, Col 1 MOHID_Water.ipynb 0

MOHID Water Notebook

The screenshot displays a JupyterLab environment running in a web browser at `localhost:8888/lab/tree/MOHID_Water.ipynb`. The interface includes a file browser on the left, a central code editor, and a bottom status bar.

File Browser: Shows a directory structure with folders like `releases`, `run_cas...`, and `work`. A file named `MOHID...` is highlighted, showing a size of 6 MB.

Code Editor: The notebook is titled "3.4.3 Create river data file in MOHID format". The code is as follows:

```
[ ]: import pandas as pd

for marker_id, marker_data in markers_dict.items():
    # Extract river_id with a default value if missing.
    river_id = marker_data.get("river_id", "unknown")

    # Generate filenames using f-strings.
    csv_filename = os.path.join(discharge_path, "{}_retrospective_data.csv".format(river_id))
    dat_filename = os.path.join(discharge_path, "{}_river_data.dat".format(river_id))

    print(csv_filename, dat_filename)

    # Additional variables
    time_units = "HOURS"

    serie_initial_data = initial_river_date.strftime("%Y %m %d") + " 0 0 0"

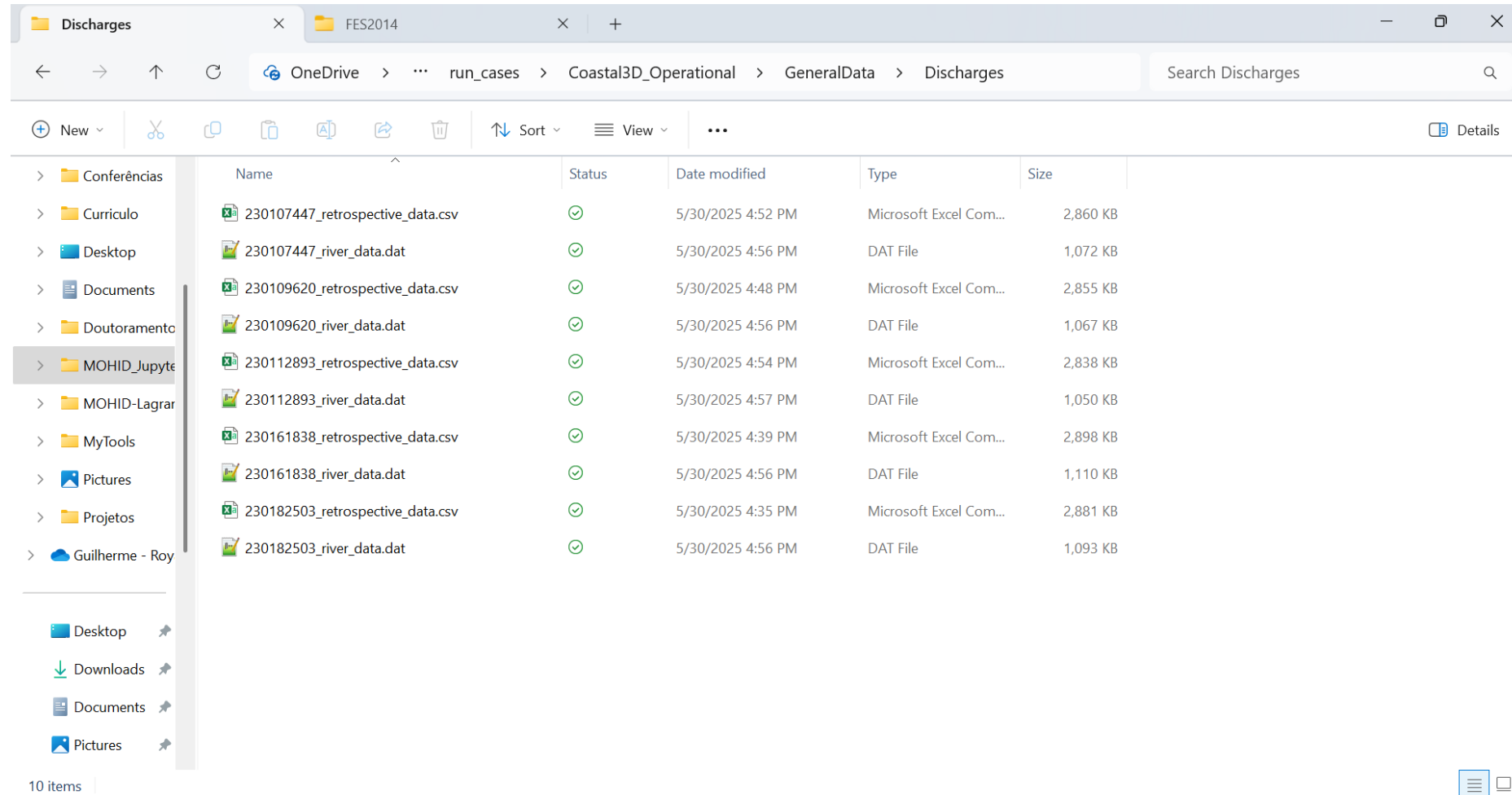
    # Get the river name from the marker data, defaulting if missing.
    river_name = marker_data.get("name", "Unnamed River")

    # markers_dict[marker_id]['location'] is assumed to be [lat, lon] so that:
    coord_x = marker_data['location'][1] # Longitude
    coord_y = marker_data['location'][0] # Latitude

    # For debugging: print out the variables:
    print(f"River Name: {river_name}")
```

Status Bar: Indicates the current mode is "Command", the cursor is at "Ln 1, Col 1", and the kernel is "Python 3 (ipykernel)".

MOHID Water Notebook



The screenshot shows a OneDrive file explorer window with the address bar indicating the path: OneDrive > ... > run_cases > Coastal3D_Operational > GeneralData > Discharges. The search bar contains the text "Search Discharges". The left sidebar shows a list of folders, with "MOHID_Jupyter" selected. The main area displays a table of files with columns: Name, Status, Date modified, Type, and Size. The table contains 10 items, each with a corresponding icon and a green checkmark in the Status column.

Name	Status	Date modified	Type	Size
230107447_retrospective_data.csv	✓	5/30/2025 4:52 PM	Microsoft Excel Com...	2,860 KB
230107447_river_data.dat	✓	5/30/2025 4:56 PM	DAT File	1,072 KB
230109620_retrospective_data.csv	✓	5/30/2025 4:48 PM	Microsoft Excel Com...	2,855 KB
230109620_river_data.dat	✓	5/30/2025 4:56 PM	DAT File	1,067 KB
230112893_retrospective_data.csv	✓	5/30/2025 4:54 PM	Microsoft Excel Com...	2,838 KB
230112893_river_data.dat	✓	5/30/2025 4:57 PM	DAT File	1,050 KB
230161838_retrospective_data.csv	✓	5/30/2025 4:39 PM	Microsoft Excel Com...	2,898 KB
230161838_river_data.dat	✓	5/30/2025 4:56 PM	DAT File	1,110 KB
230182503_retrospective_data.csv	✓	5/30/2025 4:35 PM	Microsoft Excel Com...	2,881 KB
230182503_river_data.dat	✓	5/30/2025 4:56 PM	DAT File	1,093 KB

10 items

MOHID Water Notebook

The image shows a Windows file explorer window with the path `OneDrive > run_cases > Coastal3D_Operational > GeneralData > Discharges`. The file list includes several CSV and DAT files, with `230107447_river_data.dat` selected. An overlaying text editor window displays the content of this file.

File Explorer Path: `OneDrive > run_cases > Coastal3D_Operational > GeneralData > Discharges`

File List:

- `230107447_retrospective_data.csv`
- `230107447_river_data.dat` (selected)
- `230109620_retrospective_data.csv`
- `230109620_river_data.dat`
- `230112893_retrospective_data.csv`
- `230112893_river_data.dat`
- `230161838_retrospective_data.csv`
- `230161838_river_data.dat`
- `230182503_retrospective_data.csv`
- `230182503_river_data.dat`

Text Editor Content (230107447_river_data.dat):

```
1  TIME_UNITS: HOURS
2  SERIE_INITIAL_DATA: 2015 01 01 0 0 0
3  NAME : Lerez
4  COORD_X : -8.614011
5  COORD_Y : 42.43999
6
7  Date Flow_Modulus_[m3/s]
8  <BeginTimeSerie>
9  0 5.84
10 1 5.84
11 2 5.84
12 3 5.84
13 4 5.84
14 5 5.84
15 6 5.84
16 7 5.84
17 8 5.85
18 9 5.89
19 10 5.96
20 11 6.0
21 12 6.02
22 13 6.05
23 14 6.09
```

Editor Status Bar: `length : 1,096,898 ltr Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 IN`

MOHID Water Notebook

The screenshot displays a JupyterLab interface with a browser window at the top showing the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#2.5-Set-dates`. The notebook has a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. On the left, a file browser shows a directory structure with folders 'releases', 'run_cas...', and 'work', and files 'MOHID...' (10s ago, 945.1 KB) and 'MOHID...' (2d ago, 337 B). The main area shows a code cell titled '4. Setup MOHID Water input files' with the following Python code:

```
[22]: continuous = 0 # if initial run continuous=0, else continuous=1

data_dir = os.path.join(case_dir, "Level_1", "data")
exe_dir = os.path.join(case_dir, "Level_1", "exe")
results_dir = os.path.join(case_dir, "Level_1", "res")

if not os.path.exists(results_dir):
    os.makedirs(results_dir)

if continuous == 0:
    nomfich = os.path.join(data_dir, "Nomfich_1.dat")
    run_dir = os.path.join(results_dir, "run1")
else:
    nomfich = os.path.join(data_dir, "Nomfich_2.dat")
    run_dir = os.path.join(results_dir, "run2")

if not os.path.exists(run_dir):
    os.makedirs(run_dir)

shutil.copy(nomfich, os.path.join(exe_dir, "Nomfich.dat"))
```

Below the code cell, the output shows the file path: `'C:\\Users\\aquaf\\OneDrive\\MOHID_Jupyter-Notebooks-master\\MOHID_Jupyter-Notebooks-master\\MOHID_Water\\run_cases\\Coastal3D_Operati onal\\Level_1\\exe\\Nomfich.dat'`. The bottom status bar indicates 'Simple', '1', 'Python 3 (ipykernel) | Idle', 'Mode: Edit', 'Ln 1, Col 1', 'MOHID_Water.ipynb', and '0'.

MOHID Water Notebook

The screenshot displays a JupyterLab environment with the following components:

- Browser Tab:** The address bar shows `localhost:8888/lab/tree/MOHID_Water.ipynb#4.2-Discharges`.
- File Browser (Left Panel):** A table listing files in the current directory:

Name	Mod...	Size
releases	2d ago	
run_cas...	2d ago	
work	2d ago	
MOHID...	now	6 MB
Y: MOHID...	2d ago	337 B
- Notebook Editor (Main Area):** The notebook is titled "MOHID_Water.ipynb" and is in "Python 3 (ipykernel)" mode. The current cell contains the following Python code:

```
[42]: courant = 5

# Calculate the cell spacing
dx = np.abs(np_x[0][1] - np_x[0][0])
dy = np.abs(np_y[1][0] - np_y[0][0])

velocity = (z_max*9.81)**0.5
dy_meters = dy * 111320 # Approximate meters per degree latitude
dt = courant*dy_meters/velocity

def find_nearest_divisor(n, dt):
    divisors = [i for i in range(1, n+1) if n % i == 0]
    return min(divisors, key=lambda x: abs(x - dt))

n = 86400
dt = find_nearest_divisor(n, dt)

print(f"Computed velocity: {velocity} m/s")
print(f"Computed dy: {dy_meters} m")
print(f"Computed dt: {dt} s")

def write_dt(file_name, dt):
    # Read all lines from the file
    with open(file_name, 'r') as file:
        file_lines = file.readlines()
```
- Status Bar (Bottom):** Shows "Simple" mode, "Python 3 (ipykernel) | Idle", and "Mode: Command". The cursor is at "Ln 1, Col 1" in the file "MOHID_Water.ipynb".

MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#4.2-Discharges`. The interface includes a left-hand file explorer, a central code editor, and a bottom status bar.

File Explorer: The left sidebar shows a file tree with the following items:

Name	Mod...	Size
releases	2d ago	
run_cas...	2d ago	
work	2d ago	
MOHID...	now	6 MB
MOHID...	2d ago	337 B

Code Editor: The main area displays the notebook content, specifically section 4.2 Discharges. The code defines a function `write_discharge` that writes discharge data to a file. The code is as follows:

```
[ ]: def write_discharge(input_file_discharge):
    with open(input_file_discharge, 'w') as file:
        # Write header lines
        file.write("IGNORE_ON          : 1\n")
        if continuous == 0:
            file.write("SLOW_START          : 432000\n")

        for marker_id, marker_data in markers_dict.items():
            # Extract river_id with a default value if missing.
            river_id = marker_data.get("river_id", "unknown")

            # Get the river name from the marker data, defaulting if missing.
            river_name = marker_data.get("name", "Unnamed River")

            # markers_dict[marker_id]['location'] is assumed to be [lat, lon]:
            coord_x = marker_data['location'][1] # Longitude
            coord_y = marker_data['location'][0] # Latitude

            file.write("<begindischarge>\n")

            file.write(f"NAME          : {river_name}\n")
            file.write(f"K_CELL        : 1\n")
            file.write(f"COORD_X       : {coord_x}\n")
            file.write(f"COORD_Y       : {coord_y}\n")
            file.write(f"VERTICAL_DISCHARGE : 5\n")
```

Status Bar: The bottom status bar indicates the current mode is "Command", the cursor is at "Ln 1, Col 1", and the file is "MOHID_Water.ipynb".

MOHID Water Notebook

The image shows a Windows file explorer window and a text editor window. The file explorer is displaying a directory structure for 'data' under 'MOHID_Water > run_cases > Coastal3D_Operational > Level_1'. The text editor is open to 'Discharges_1.dat' and shows the following content:

```
1 IGNORE_ON : 1
2 SLOW_START : 432000
3 <begindischarge>
4 NAME : Tambre
5 K_CELL : 1
6 COORD_X : -8.874738
7 COORD_Y : 42.827131
8 VERTICAL_DISCHARGE : 5
9 DATA_BASE_FILE : ../../GeneralData/Discharges/230182503_river_data.d
10 FLOW_COLUMN : 2
11
12 <<beginproperty>>
13 NAME : salinity
14 UNITS : psu
15 DEFAULTVALUE : 0.01
16 <<endproperty>>
17
18 <<beginproperty>>
19 NAME : temperature
20 UNITS : °C
21 DEFAULTVALUE : 10
22 !TIME_SERIE_COLUMN : 2
23 !FILENAME : ../../GeneralData/Discharges/River1Temp.dat
24 <<endproperty>>
25
26 <enddischarge>
27
```

The status bar at the bottom of the text editor shows: length : 3,363 lines : 128 Ln : 1 Col : 1 Pos : 1. The file explorer shows 26 items, with 1 item selected (3.28 KB) and available on this device.

MOHID Water Notebook

The screenshot shows a web browser window displaying the MOHID Water Notebook. The browser's address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#4.2-Discharges`. The notebook interface includes a left sidebar with a file explorer showing a directory structure with folders like `releases`, `run_cas...`, and `work`, and files like `MOHID...` and `MSMpi...`. The main area displays the notebook content, which is currently on a cell titled `5.1 Install MSMPI (Windows)`. This cell contains a code block with the following text:

```
[47]: # Define the URL for MSMPI
msmpi_url = "https://www.microsoft.com/en-us/download/details.aspx?id=57467"
download_command = f"powershell Invoke-WebRequest -Uri {msmpi_url} -OutFile MSMpiSetup.exe"

# Run the command
subprocess.run(download_command, shell=True)

install_command = "MSMpiSetup.exe /quiet /norestart"
subprocess.run(install_command, shell=True)

check_command = "mpiexec -help"
subprocess.run(check_command, shell=True)

[47]: CompletedProcess(args='mpiexec -help', returncode=0)
```

Below the code block, the notebook shows the next section, `5.2 Start Simulation`, followed by a button labeled `+ 1 cell hidden`. The bottom of the interface shows the status bar with `Python 3 (ipykernel) | Idle` and `Mode: Command`.

MOHID Water Notebook

The screenshot displays a JupyterLab environment. The top browser bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#4.2-Discharges`. The left sidebar contains a file explorer with a table of files:

Name	Mod...	Size
releases	2d ago	
run_cas...	2d ago	
work	2d ago	
MOHID...	now	6 MB
MOHID...	2d ago	337 B
MSMpi...	2s ago	122.6 KB

The main area shows a notebook with the title "5.2 Start Simulation". The code cell contains the following Python code:

```
[ ]: #print("Number of CPU cores:", os.cpu_count())
#np = os.cpu_count() - 2 #number of parallel processes you want to launch
np = 6

release = (os.path.join(os.getcwd(), "releases", "MOHIDWater_v24.10_x64_MPI", "MOHIDWater_v24.10_x64_MPI.exe"))
DomainConsolidation = (os.path.join(os.getcwd(), "releases", "MOHIDWater_v24.10_x64_MPI", "DomainConsolidation.exe"))

backup_path = (os.path.join(case_dir, "backup"))
boundary_conditions_dir = (os.path.join(case_dir, "GeneralData", "BoundaryConditions"))

file_name_meteo = "Meteo.hdf5"
file_hydro = "CMEMS.hdf5"

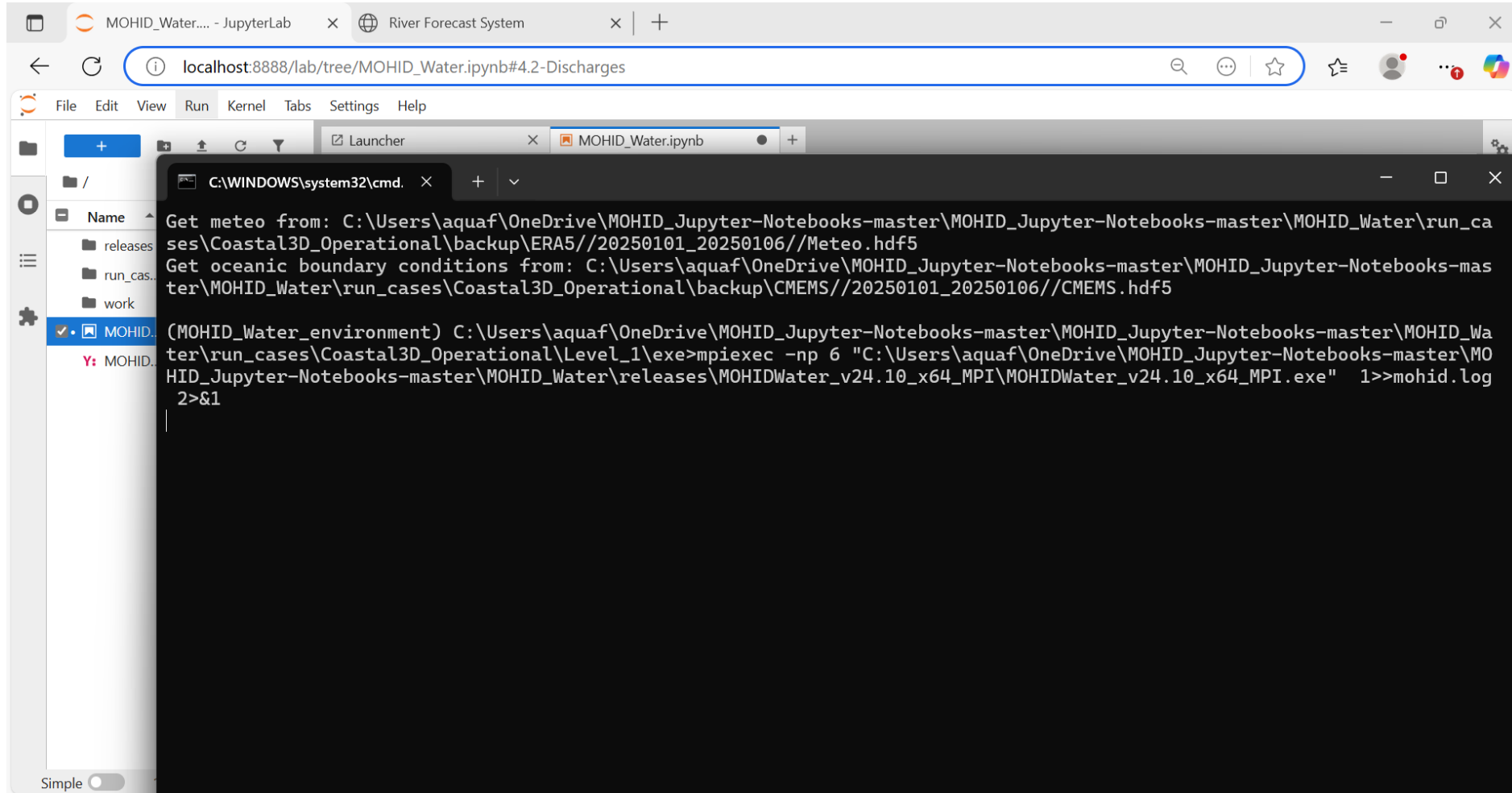
xmart_path = os.path.join(os.getcwd(), "work", "XMART")

#This file can later be used as input to XMART.py for operational purposes
input_file = os.path.join(xmart_path, "Input_XMART.py")

with open(input_file, 'w') as file:
    file.write(f"release=r'{release}'\n")
    file.write(f"DomainConsolidation=r'{DomainConsolidation}'\n")
    file.write(f"np={np}\n")
    file.write(f"backup_dir=r'{backup_path}'\n")
    file.write(f"dir_meteo=r'{backup_path_meteo}'\n")
    file.write(f"dir_hydro=r'{backup_path_hydro}'\n")
```

The bottom status bar indicates the current mode is "Command", the cursor is at line 9, column 88, and the notebook is named "MOHID_Water.ipynb".

MOHID Water Notebook



The screenshot displays a JupyterLab environment with a terminal window open. The terminal shows the execution of a MOHID Water notebook, which includes commands to retrieve meteorological and oceanic boundary conditions and to run the MOHID Water model using MPI.

```
C:\WINDOWS\system32\cmd. x + v
Get meteo from: C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\run_cases\Coastal3D_Operational\backup\ERA5//20250101_20250106//Meteo.hdf5
Get oceanic boundary conditions from: C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\run_cases\Coastal3D_Operational\backup\CMEMS//20250101_20250106//CMEMS.hdf5
(MOHID_Water_environment) C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\run_cases\Coastal3D_Operational\Level_1\exe>mpiexec -np 6 "C:\Users\aquaf\OneDrive\MOHID_Jupyter-Notebooks-master\MOHID_Jupyter-Notebooks-master\MOHID_Water\releases\MOHIDWater_v24.10_x64_MPI\MOHIDWater_v24.10_x64_MPI.exe" 1>>mohid.log 2>&1
```

MOHID Water Notebook

The image shows a Windows File Explorer window in the background, displaying the directory structure of a folder named 'exe'. The path is 'Start OneDrive > ... > run_cases > Coastal3D_Operational > Level_1 > exe'. The file list includes several 'Error_and_Messages_*.log' files, 'mohid.log', 'Nomfich.dat', 'run.bat', 'Tree.dat', and several 'UsedKeyWords_*.dat' files. The 'mohid.log' file is selected.

In the foreground, a Notepad++ window is open, displaying the contents of the 'mohid.log' file. The text in the log file is as follows:

```
23361 ----- MOHID -----
23362 ----- MOHID -----
23363
23364
23365 Program Mohid Water successfully terminated
23366
23367
23368 Total Elapsed Time      :      16655.04    4h 37min 35s
23369
23370 Total CPU time          :      6800.09
23371
23372 ----- MOHID -----
23373 CPU usage (%)           :      40.83
23374
23375 Program Mohid Water successfully terminated
23376
23377 Workcycle Elapsed Time :      16507.36
23378
23379
23380 Total Elapsed Time      :      16655.05    4h 37min 35s
23381
23382 ----- MOHID -----
23383
23384 Total CPU time          :      12115.69
23385
23386
23387 Program Mohid Water successfully terminated
23388 Program Mohid Water successfully terminated
23389 Workcycle CPU time      :      6683.56
```

The status bar at the bottom of the Notepad++ window shows: 'length: 722,362 lines: 23,475 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 IN'.

MOHID Water Notebook

The screenshot shows a web browser window displaying the MOHID Water Notebook. The browser's address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#2.5-Set-dates`. The notebook interface includes a left sidebar with a file explorer showing a directory structure with folders like `releases`, `run_cas...`, and `work`, and files like `MOHID...` and `MOHID...`. The main area displays the notebook content, which is titled "6. Visualize results". The code in the notebook cell is as follows:

```
[8]: variable = "velocity modulus" # Change as needed

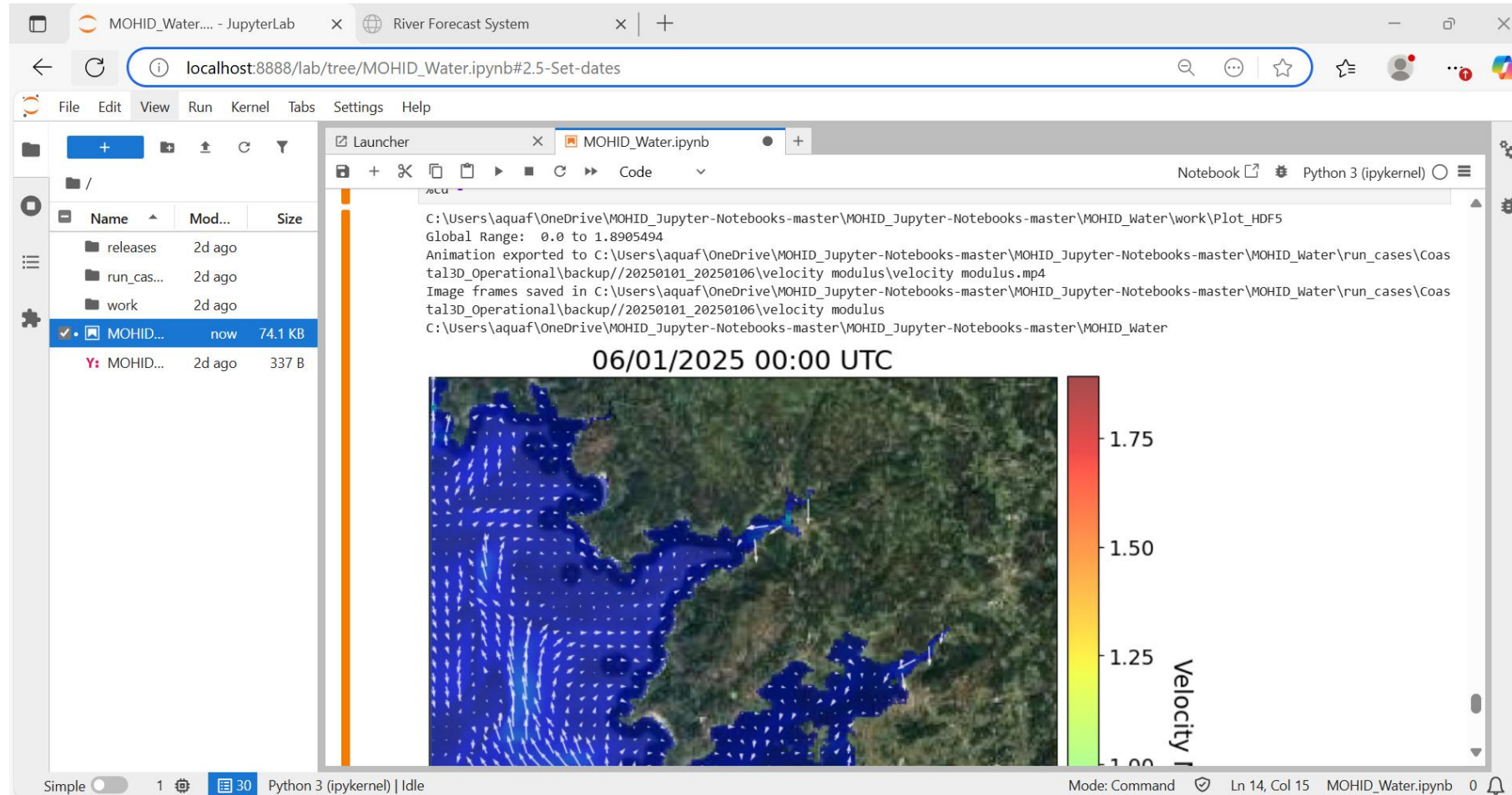
# =====
# DEFINE VARIABLE-LABEL DICTIONARY
# =====
variable_label_dict = {
    "velocity modulus": "Velocity Modulus(m/s)",
    "salinity": "Salinity(psu)",
    "temperature": "Temperature(°C)",
    "water level": "Water Level(m)"
}

# User-specified parameters for skipping time steps, adjusting extent, vectors, etc.
skip_time = 1 # Sample every nth time step
extent_cells = 1 # Number of extra cells added to the plot extent
increase_zoom_level = 1 # Increase computed zoom level by this amount to improve background image resolution
skip_vector = 5 # Skip factor when plotting vectors (to reduce clutter)
vector_scale = 10 # Scale for the current vector arrows
vector_color = 'white' # Color for the wind vectors
transparency_factor = 0.5
dpi = 150 #specify the DPI

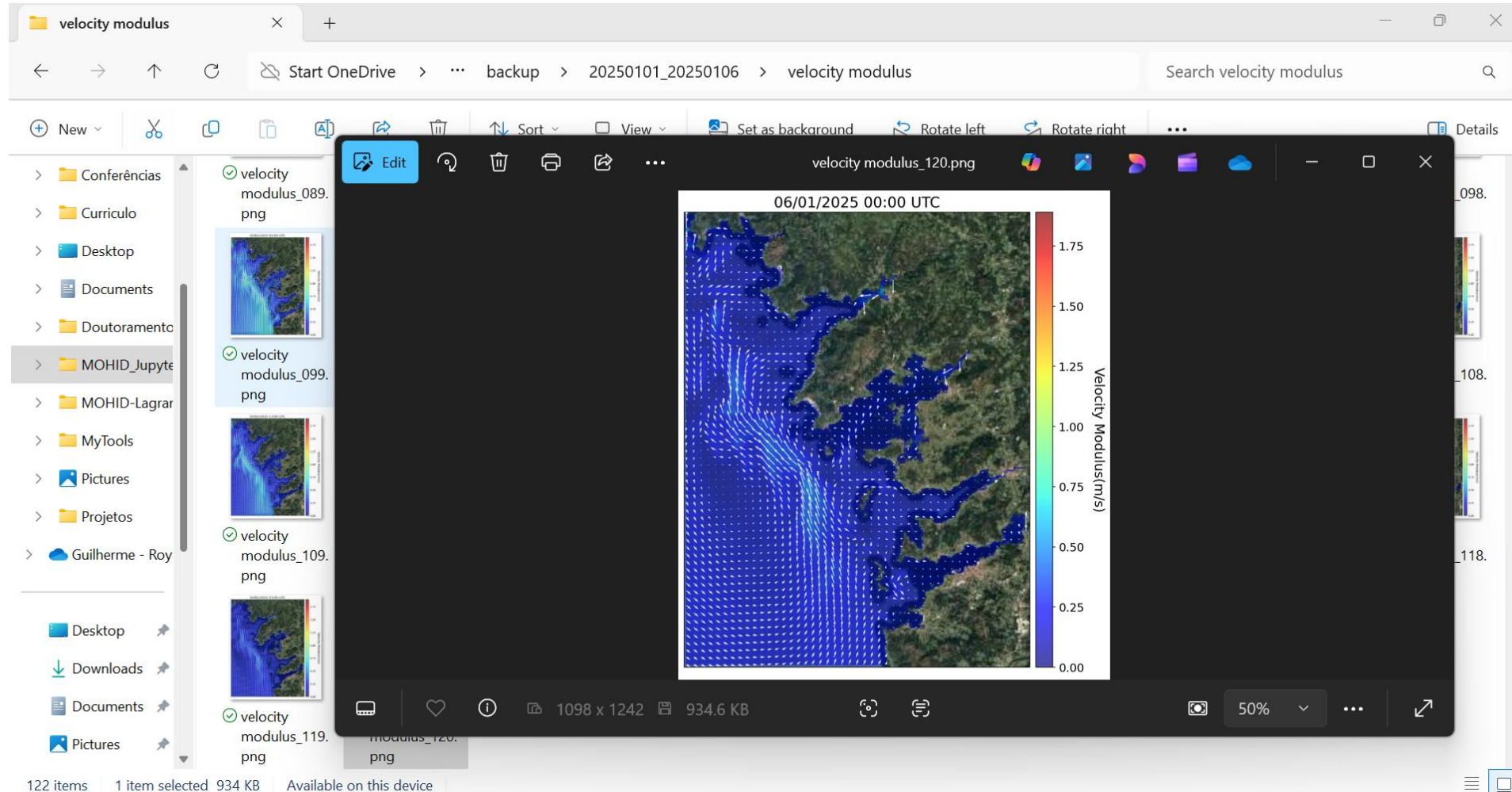
# Option to enable or disable vector overlay and image frame saving
show_vectors = True # Set to False to disable wind vectors in the animation
save_frames = True # Set to False to disable saving individual image frames
```

The notebook interface also shows a status bar at the bottom indicating the current mode is "Edit", the line and column number is "Ln 25, Col 82", and the file name is "MOHID_Water.ipynb".

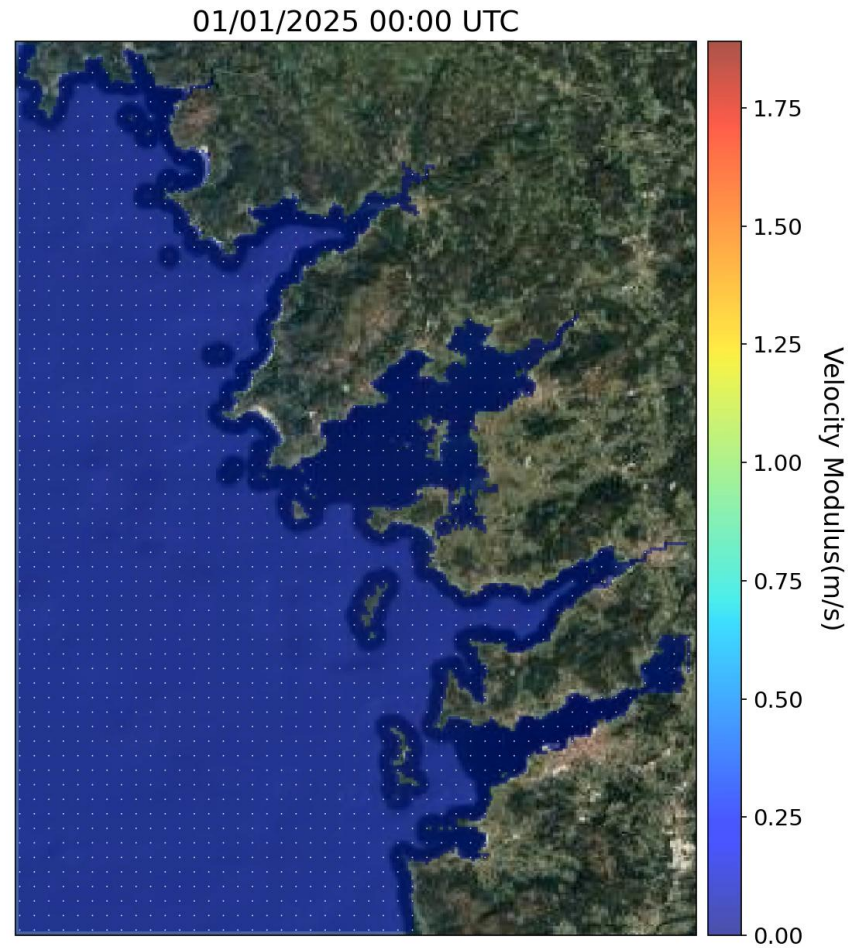
MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook

The screenshot displays the JupyterLab environment for the MOHID Water Notebook. The browser address bar shows the URL `localhost:8888/lab/tree/MOHID_Water.ipynb#2.5-Set-dates`. The interface includes a left-hand sidebar with a file explorer showing a directory structure with folders like `releases`, `run_cas...`, and `work`, and files like `MOHID...` and `MOHID...`. The main area is a code editor for the notebook `MOHID_Water.ipynb`, which is currently in 'Code' view. The code is titled '6. Visualize results' and contains a cell with the following content:

```
[8]: variable = "salinity" # Change as needed

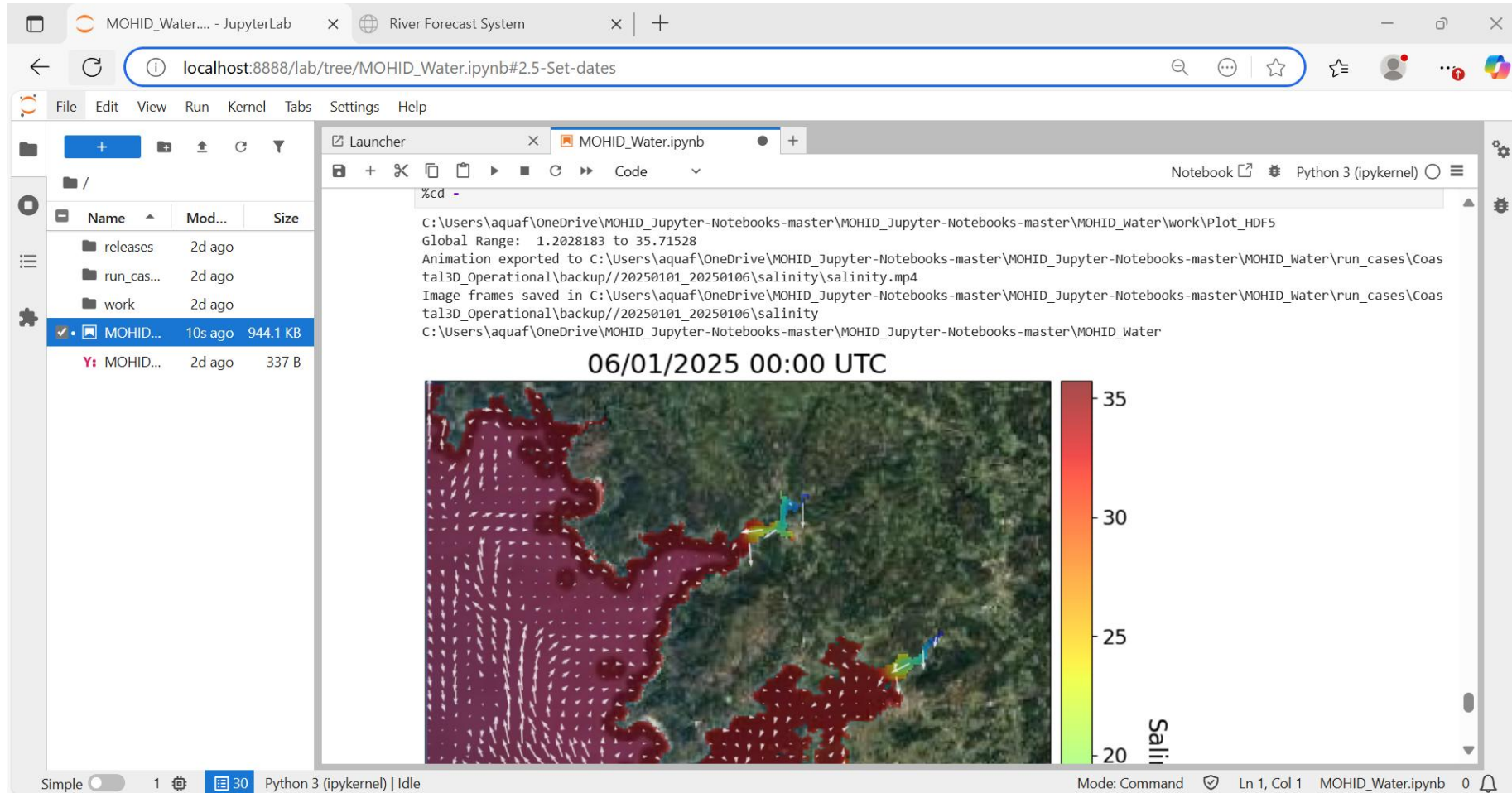
# =====
# DEFINE VARIABLE-LABEL DICTIONARY
# =====
variable_label_dict = {
    "velocity modulus": "Velocity Modulus(m/s)",
    "salinity": "Salinity(psu)",
    "temperature": "Temperature(°C)",
    "water level": "Water Level(m)"
}

# User-specified parameters for skipping time steps, adjusting extent, vectors, etc.
skip_time = 3 # Sample every nth time step
extent_cells = 1 # Number of extra cells added to the plot extent
increase_zoom_level = 1 # Increase computed zoom level by this amount to improve background image resolution
skip_vector = 5 # Skip factor when plotting vectors (to reduce clutter)
vector_scale = 10 # Scale for the current vector arrows
vector_color = 'white' # Color for the wind vectors
transparency_factor = 0.5
dpi = 150 #specify the DPI

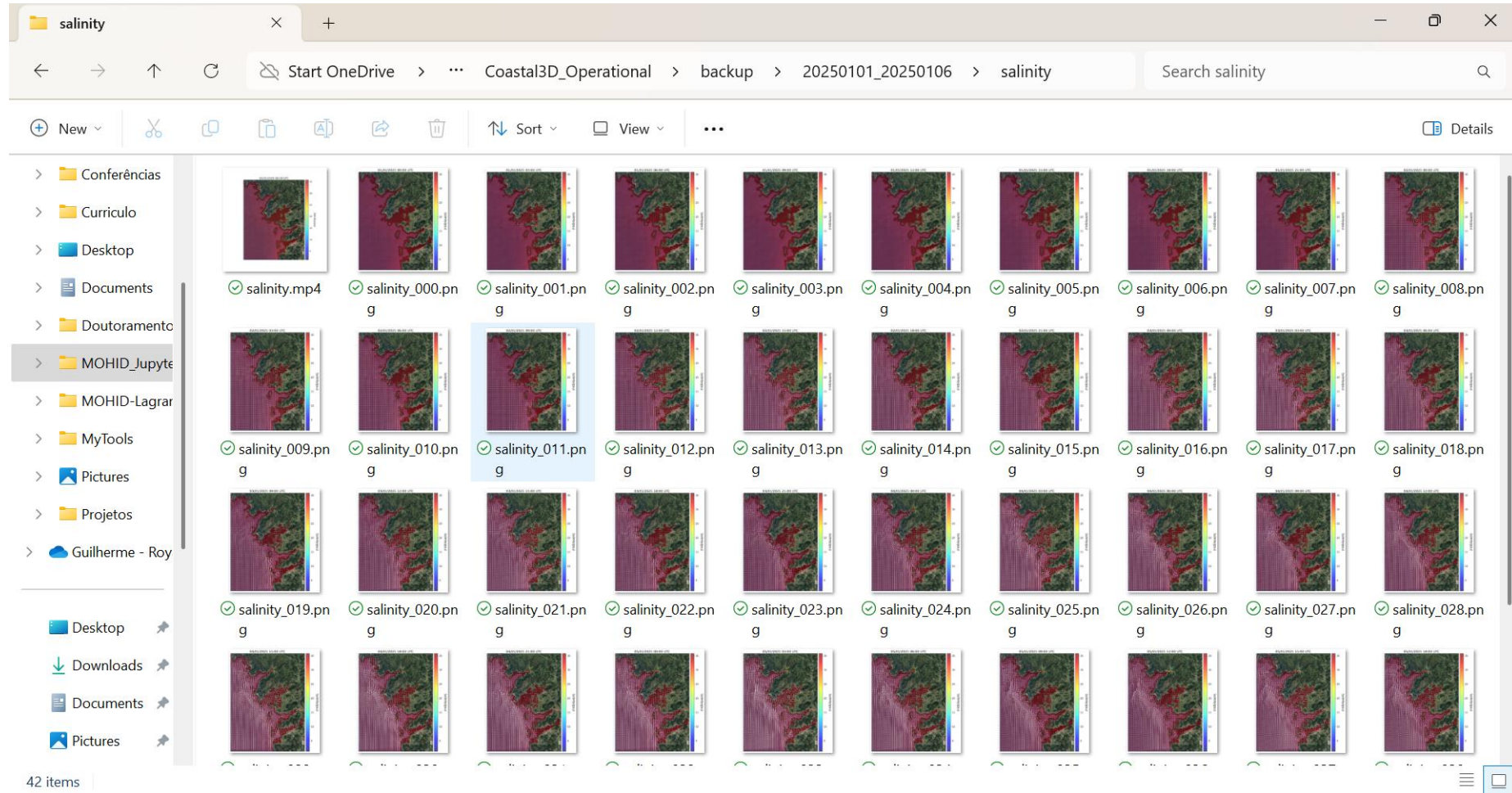
# Option to enable or disable vector overlay and image frame saving
show_vectors = True # Set to False to disable wind vectors in the animation
save_frames = True # Set to False to disable saving individual image frames
```

The bottom status bar indicates the notebook is running on 'Python 3 (ipykernel)' in 'Idle' mode, with the cursor at 'Ln 1, Col 13'.

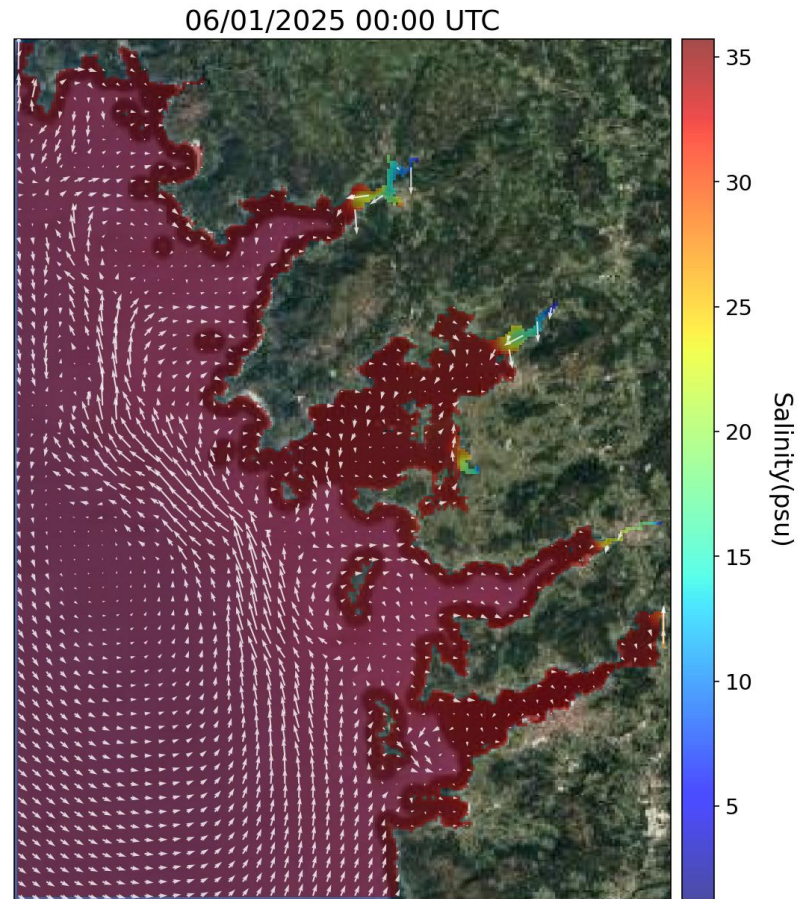
MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook



MOHID Water Notebook

The screenshot shows the JupyterLab interface for the MOHID Water Notebook. The browser address bar indicates the notebook is running on localhost:8888. The left sidebar displays a file explorer with a table of files:

Name	Mod...	Size
releases	2d ago	
run_cas...	2d ago	
work	2d ago	
MOHID...	now	916.3 KB
Y: MOHID...	2d ago	337 B

The main notebook area shows the following sections:

2.5 Set dates

```
[5]: #Set dates for boundary conditions download

#Define a 5-day interval if it's the initial run for model warm-up
start_date_str = "2025-1-6" #"%Y-%m-%d"
end_date_str = "2025-1-10" #"%Y-%m-%d"

#if daily = 1, one day per file, else just one file for the interval end_date - start_date.
#set daily = 0 for warm-up
daily = 1

forecast = 0

#The keywords below are only used if forecast = 1
refday_to_start = 0 #0 is today, -1 yesterday, 1 tomorrow
number_of_runs = 1 #
```

3. Boundary Conditions

+ 33 cells hidden

4. Setup MOHID Water input files

+ 5 cells hidden

The bottom status bar shows the notebook is running on Python 3 (ipykernel) in Idle mode, with the cursor at Line 1, Column 1.

MOHID Water Notebook

backup

Start OneDrive > ... MOHID_Water > run_cases > Coastal3D_Operational > backup >

Search backup

New

Sort

View

Details

Name	Status	Date modified	Type	Size
20250101_20250106	✓	6/1/2025 10:07 AM	File folder	
20250106_20250107	↻	6/1/2025 1:14 PM	File folder	
20250107_20250108	↻	6/1/2025 2:14 PM	File folder	
20250108_20250109	↻	6/1/2025 3:13 PM	File folder	
20250109_20250110	↻	6/1/2025 4:13 PM	File folder	
CMEMS	✓	6/1/2025 10:28 AM	File folder	
ERAS	✓	6/1/2025 11:53 AM	File folder	

7 items

Next step

- MOHID_Postprocessing Notebook
 - Time series
 - Extract time series
 - Plot time series
 - Compare with measurements
 - Statistics
 - Harmonic Analysis
 - 2D Maps
 - Plot 2D Maps
 - Statistics
 - Plot vertical cuts
 - Export to Geo-tiffs
 - Animations

Future possibilities of Notebooks

- MOHID_Oil
- MOHID_HNS
- MOHID_Sediment
- MOHID_WaterQuality
- MOHID_Outfall
- ...

Thank you!

For additional information, please contact:
guilherme.franz@aquafLOWconsultoria.com

