University of Copenhagen

MSc program in Geology–Geoscience

Geodynamics – shaping Earth's surface

A short guide to Jupyter Notebooks

1 Introduction

Jupyter Notebooks serve as an excellent tool for Python coding, offering interactivity, user–friendly interfaces, and the capability to seamlessly integrate code with text, images, and equations. Throughout this course, we will extensively utilize Jupyter Notebooks in both weekly exercises and the final research project. Proficiency in Python programming is not a prerequisite, as most, if not all, of the code is pre–written for you. The goal is not to burden you with learning how to code but rather to concentrate on interpreting results and evaluating the sensibility of our parameter choices. Consider this an opportunity to passively acquire some programming knowledge, recognizing the benefits of automating repetitive tasks and embracing the computational power that surpasses traditional pen-and-paper methods.

I emphasize, no prior knowledge of Python is required for this course; by design you can successfully complete the final research project by simply copy-pasting from the weekly exercises. However, for those keen on acquiring some programming skills, introductory materials are available. These can be found in *Module 0: Preliminary Material* under *Introduction to Python*, and include Jupyter Notebooks that cover fundamental topics like variable declaration, module and function imports, as well as tasks such as loading and saving text files. These resources should provide you with just enough knowledge to navigate and comprehend the codes provided throughout the course.

2 Installing Jupyter Notebooks

Weekly files will be uploaded to the course's Absalon page. To run these files, you will have two options: A) have a local installation in your computer of Python (the code interpreter) and JupyterLab/VisualStudioCode (the code editor), or B) use an online service to edit and run Jupyter Notebook, such as Google Colab.

A **Local installation**: Go with a local installation if you wish for a lasting and scalable Python–programming solution. I recommend installing an Anaconda distribution (`https://www.anaconda.com/download/success`), which comes with Python (v3.8 suggested), a lot of useful pre–installed packages, and dedicated text editors (e.g., Spider and JupyterLab). Once installed, you can check the installed your editors at Anaconda Navigator (Figure 1).

B **Online service**: Go with an online service if you wish for a quick and easy solution. I recommend using Google Colab, where you can run notebooks stored in your Google Drive (Figure 2). Because of dependency calls, you will have to upload to Google Drive the whole SCRIPTS folder (with DEPENDENCIES and DATA).

For either case, I recommend having the following folder structure:

```
2024-Geodynamics
└── SCRIPTS
    ├── DATA/
    ├── DEPENDENCIES/
    ├── EX1_ISOSTACY.ipynb
    ├── EX2_GRAVITY.ipynb
    ├── EX3_PLATE_MOTIONS.ipynb
    └── ...
```

Week 0 material will contain all the necessary material of DATA and DEPENDENCIES. Each subsequent week you will be provided with corresponding exercises, which you can add to the SCRIPTS folder before running.

3   Running Jupyter Notebooks

A **Local installation**: If you installed JupyterLab from the Anaconda Navigator, you can launch the app from the Navigator itself. Anaconda can take a while to open, so you might want to launch a Command Prompt instead, and run the command "jupyter lab". JupyterLab will open an editor window in your internet explorer (Fig 2). Navigate from here to where your notebooks are located and double–clicking to open them.

B **Online service**: If you uploaded all the Week 0 files on your Google Drive, simply double–click on the notebook.

4   Running cells

To run a cell click on it and find the "play" button. It might be on the left side of the cell, or on the top part of the window. You can also find buttons for running all cells, all the cells above the current one, and all the cells below the current one. If you change something in a cells, you need to run it for the changes to take effect.

5   Printing output

In Jupyter Notebooks (not Python), you can print the content of a variable by typing it at the bottom of the cell before running it. This will only print one variable, the one at the very bottom. To print more than one variable, you can use the "print()" command (e.g. print(myvariable)).

6   Getting help with Python functions

You can print help on any function by typing at the bottom of a cell: ?myfunction()
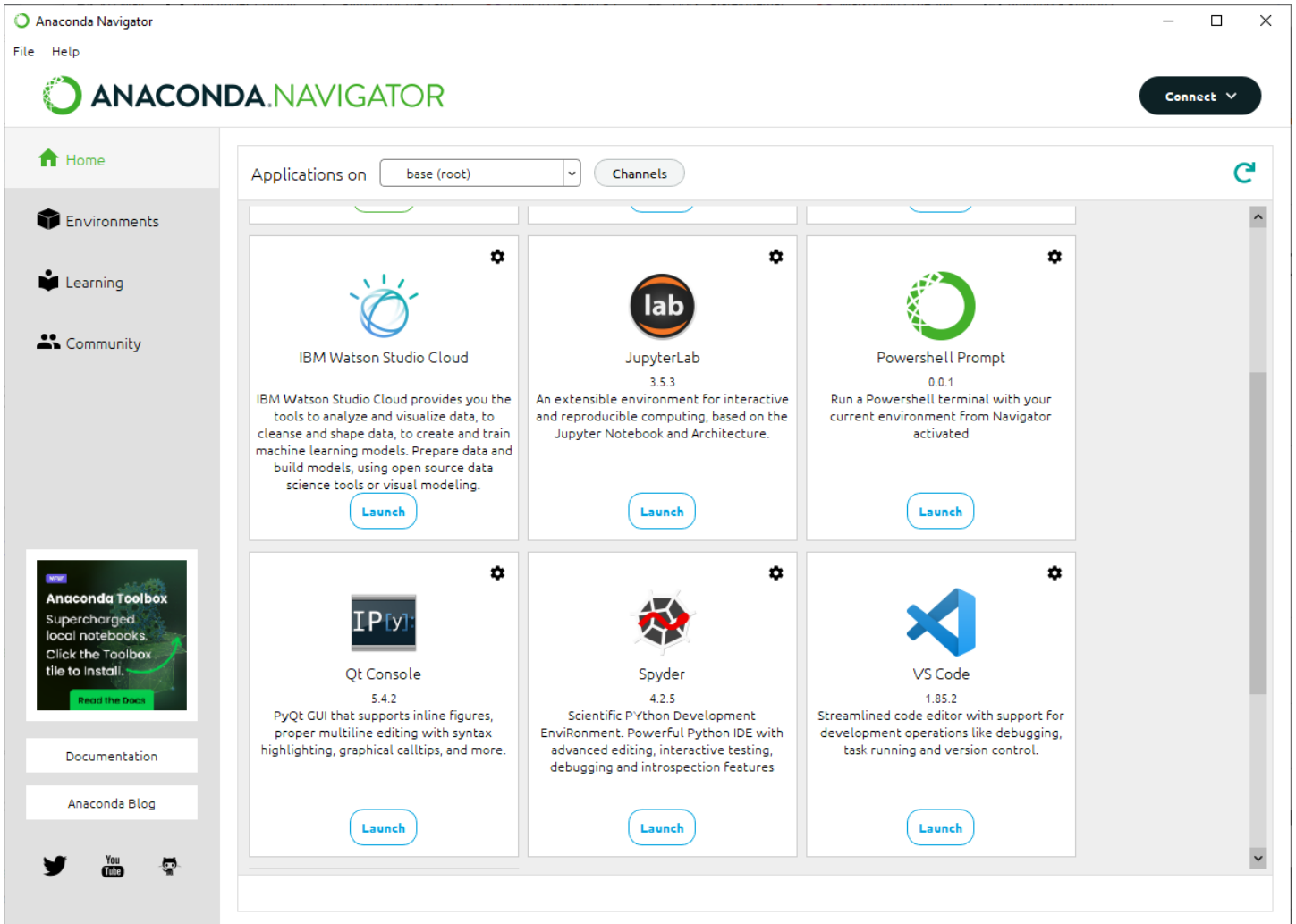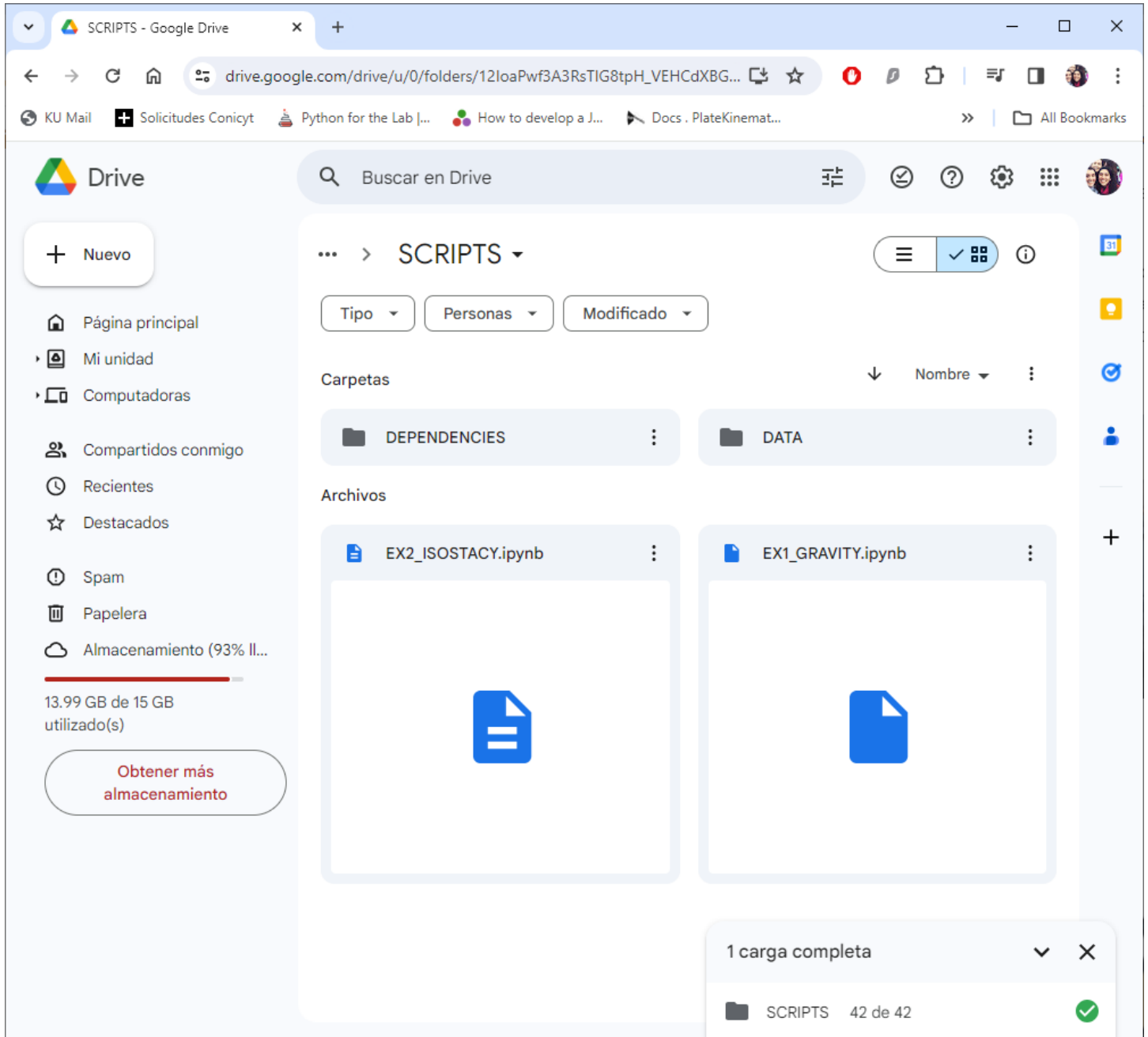
Figure 1: Anaconda Navigator window.

Figure 2: Google colab start window (Sorry for the Spanish).