. . .

Processamento do Sinal e do Ruído Sísmicos: Aplicações à Engenharia e ao Meio–Ambiente

ProSeisSN

Notas de Aula e Lista de exercícios

• *Módulo 1* •

Preâmbulo

Esse documento contem as notas de aula referentes ao Módulo acima, incluindo referências relevantes[1], sendo baseado em material disponível livremente na Web, fruto do trabalho dedicado de várias pessoas no Mundo. Como já dito por muitas pessoas, *subi em ombros de gigantes*.

[1] Todo o material do Curso está escrito em Português e Inglês. Como esse é um curso voltado a graduados, supõe-se familiaridade com o Inglês.

Contents

1 Course Environment

The *C*ourse Environment is based on several resources from the Web:

1. **GitHub** to store Course's files in a **Git** repository.
2. **JupyterLab** is a web–based user interface, which works with **Jupyter Notebooks**, containing computer code, and rich text elements.
3. **GitHub Classroom** for storing your assignments and Projects under your course's **GitHub** course organization.
4. **Binder** to open and execute **Jupyter Notebooks** on the cloud.
5. **Anaconda** to create a Python environment for running the data exercises on your computer; *h*ighly recommended.

The Course's repository is in **GitHub**

`https://github.com/jandyr/ProSeisSN.`

Students have the choice of running the exercises and making their projects either web–based, using **Binder**, or installing locally the *C*ourse Environment as described below.

The instructions on the local instalation of the *C*ourse Environment are only valid under **LINUX**. Students who use other operational systems, are referred to the tutorials.

1.1 Git and GitHub

All the Course's files are stored in the **Git** repository `https://github.com/jandyr/ProSeisSN`. To replicate the Course's content on your machine you:

1. Create a *C*ourse directory **ProSeisSN**;
2. Clone using HTTPS on the web; for this you will need to have **Git** installed locally; or
3. Download a **ZIP** file to the directory **ProSeisSN**.

Choose your option clicking on the green button, Figure 1.

Clone ProSeisSN using HTTPS

Replicate the Course's **GitHub** onto your machine. You can

1. Go to `https://github.com/jandyr/ProSeisSN.git`.
2. Clone using the web **URL** clicking on copy, Figure 1.
3. In a terminal enter
   `gitconfig--globaluser.emailyou@example.com\gitconfig--globaluser.name"YourName"`
   changing "you@example.com" and "Your Name" accordingly.
4. Git **clone** the Course's Git to your machine
   `gitclonehttps://github.com/<Course'sGitlink>.`
   where <Course's Git link> is the previously copied link. You'll end up with a directory **ProSeisSN** locally.

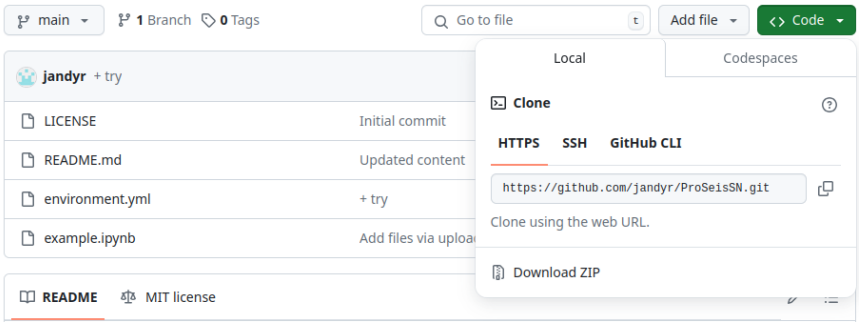


Figure 1: Clone or download to replicate the Course's content on your machine.

Create and manage your own GitHub Account

1. Create one **GitHub** account ⇒ `https://github.com/`.
2. Setup **Git** Locally ⇒ `https://git-scm.com/downloads`.
   Setup your name & email in **Git** by running following commands on a terminal:
   git config –global user.name "your name"
   git config –global user.email "email@provider.com"
3. Connect your local git client with **GitHub** by caching your password. ⇒ `https://git-scm.com/downloads`.
4. Use **GitHub CLI** or **gh** to authenticate with your **GitHub** account by caching your **GitHub** credentials in Git. Install **GitHub CLI** ⇒ `https://github.com/cli/cli#installation`.
   *I*n the command line
   gh auth login
   then:
   - log into **GitHub.com** ;
   - preferred protocol for Git operations, select **HTTPS**;
   - authenticate to your **GitHub** credentials, enter Y;
   - select **Login with a web browser**; and
   - Press Enter to open **GitHub.com** in your browser.

   Installing gh on Linux ⇒ `https://github.com/cli/cli/blob/trunk/docs/install_linux.md`.
   GitHub CLI will automatically store your **GitHub** credentials for you, when you choose HTTPS as your preferred protocol for Git operations and answer "yes" to the prompt asking if you would like to authenticate to Git with your GitHub credentials. You will need to use

gh auth login

   every time you want to want to issue git commands between your

machine and your web **GitHub** `https://cli.github.com/manual/gh_auth_login`.

5. Your new Repository is created:
   ⇒ `https://github.com/yourGit`.

Git communicates between the local repository on our computer and the remote repository on **GitHub**[2]. To back-up any changes in the upstream repository, see Figure 2:

1. Add files from the working directory to the *staging area*,
   git add <notebook you have been working on>
2. Commit to the *Repo*,
   git commit -m 'A commit message'
3. Push to cloud
   git push origin master
4. Assume you add one of your own files to your local **GitHub repo**, *e*.g., **temp.txt**. Add **temp.txt** to your **GitHub repo**
   `gitaddtemp.txt`
   and commit your staged files to the local repository, adding any comment `gitcommit-m'relevantcomment'`.
5. Publish (upload) your *local* commits in the cloud using
   `gitpush`
6. List the untracked files in your directory
   `gitstatus`
7. Track the untracked files with
   `gitadd`

1.2 JupyterLab

Jupyter Notebooks, file extension .ipynb contain computer code, and rich text elements in a **JupyterLab** environment. They are the programming environment of this course, available in a **JupyterLab** setting via cloud computing environments **Binder**, see below.

The basic usage of a **JupyterLab** environment follows:

1. Cell:
   A Cell is a container for text or code to be executed by the notebook's kernel. Type in
   print('Hello World!')
   and click the **Run** Button.
2. Toggle between **edit** and **command** modes with **Esc** and **Enter**. Cell's keyboard shortcuts:
   Scroll up and down with your **Up** and **Down** keys.
   Press **A** or **B** to insert a new cell above or below the active cell.
   M will transform the active cell to a Markdown cell (to format a plain

[2] A "repository", or a "Git project", or a "repo", is a location for storing files.

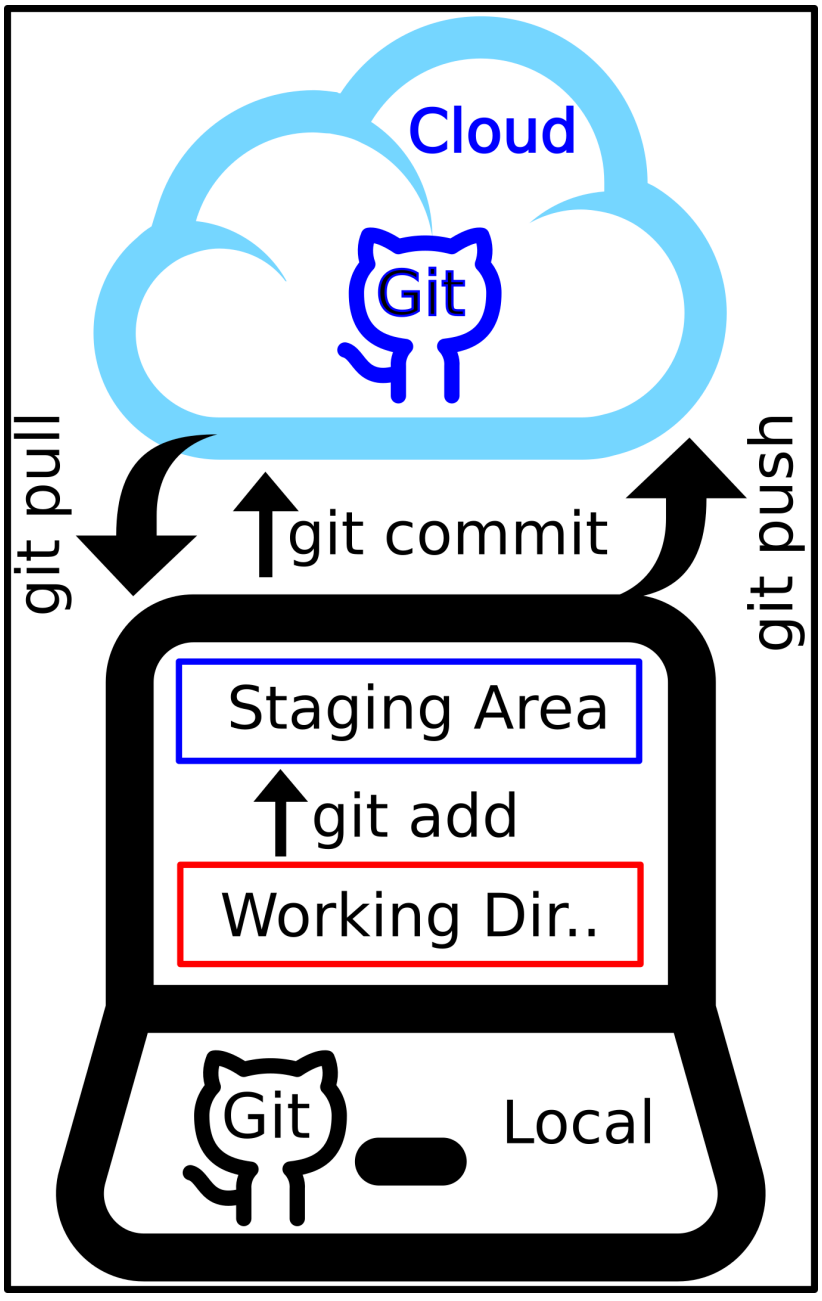


Figure 2: Update Git project using the pull and push commands. Always pull before you push.

text.). **Y** will set the active cell to a code cell.

3. Kernels:
   The kernel executes a code cell. The kernel's state persists over time and between cells — it pertains to the document as a whole and not individual cells.

JupyterLab in local programming

JupyterLab can be also installed as a terminal–launched application, for local programming, in your machine.
⇒ `https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html#usage-with-jupyterhub`.
A tutorial ⇒ `https://www.dataquest.io/blog/jupyter-notebook-tutorial/#`. To install **JupyterLab** using base **conda**,

1. Activate **conda**.
2. Install it with **conda** or **pip**
   conda install -c conda-forge jupyterlab or
   pip install jupyterlab

To run **JupyterLab** locally,

1. Navigate to where your Jupyter Notebooks are (*e.g.*, ... `/ProSeisSN` and open a **terminal**.
   Type in terminal: jupyter notebook &
   and click on **hello.ipynb** and then on **Run** button. Alternatively you can
2. Click the **New** drop-down button in the top-right and select **Python 3(ipykernel)**.
   A **Untitled.ipynb** is created, a text file that describes the contents of your notebook in **JSON**in a Notebook Interface.

GitHub Classroom:

We use **GitHub Classroom** to create, submit, and test codes. Your assignments will be stored as a GitHub repository under your course's GitHub organization. The repository will be private, meaning that your code will not be visible to your classmates or anyone else on the internet. There will be a template repository for each assignment or Project. A tutorial is in `https://hellorob.org/tutorials/github-classroom`.

You should receive a **GitHub Classroom** assignment link. Each assignment will have a template code and eventual rich texts. Click on the link and click

Accept this assignment

A Classroom helps you make private copies of an assignment that you can modify and sAn assignment will have the following,

1. A basic description,
2. A list of problems to solve/answer,
3. A Jupyter notebook or Python code that you need to modify for the assignment, and
4. Data files to use with the code.

1.3 Binder

Binder `https://mybinder.org/` runs **Jupyter Notebooks** in cloud-based computing environment[3]. Just click on the link in the corresponding *Unit* on **GitHub** to start working.

Once the **Binder** instance is ready, you can navigate to the lesson folders and start working with existing notebooks or create a new one. Note that the **Binder** instance is temporary, all your files will be lost after the session; save them before logout!

1.4 Anaconda

Conda create separate independent environments, each containing their own files, packages, and package dependencies. I concentrate here on the command line interface, CLI), which processes text commands. A tutorial is given in `https://conda.io/projects/conda/en/latest/user-guide/getting-started.html`.

1. Install **Anaconda** or **Miniconda** on your machine.
2. Create a Python environment using the file **environment.yml** from the Course's repository in **GitHub** `https://github.com/jandyr/ProSeisSN`, **conda env create -f environment.yml**.
3. Activate the environment by
   conda activate ProSeisSN,
   and deactivate it with
   conda deactivate.

2 Exercises on Wave Propagation

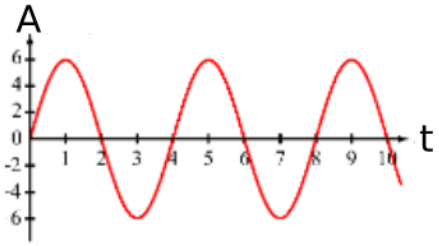
Exercise 1

A onda mostrada na Figura 2 propaga-se a uma velocidade de fase $v_p = 10^{\text{m}}/_{\text{s}}$:

a) Qual é a sua amplitude e comprimento de onda?

b) Qual a sua frequência?

[3] Don't use Internet Explorer



Exercise 2

Seja a onda

$$A\,(x,t) = 10\cos\left(10\pi t + 45\pi x\right)\,\mathrm{m}$$

propagando-se no espaço (x) no tempo (t); todas unidades no SI:

a) Qual a frequência f da onda?
b) Qual o seu comprimento de onda λ?
c) Qual a sua velocidade de fase v_p?

Exercise 3

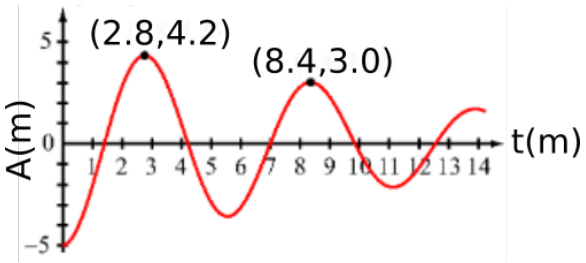
A onda mostrada na Figura 2 varia senoidalmente no espaço (x), com amplitude decrescente. Ela pode ser descrita pela expressão geral,

$$v_p = A_0 \mathrm{e}^{-\alpha x}\cos\left(\frac{2\pi x}{\lambda} + \phi_0\right).$$

a) Qual a sua amplitude em $x = 0$?
b) Use os picos marcados para estimar o seu comprimento de onda λ.
c) Use a amplitude em $x = 0$ para estimar a sua fase inicial ϕ_0 e a relação

$$\cos\left(\alpha \pm \beta\right) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$$

para estimar a sua constante de atenuação α.



Exercise 4

Estime o gradiente de

$$A = x^2 y + x y^2 + x z^2$$

no ponto P=(1, -1, 2).

3 A Crash Course on Python

The *C*ourse Environment is based on **Python**, students that don't the language are strongly advisable to have a brief introduction to it. It is *h*ighly recommended using the **Python** under **Anaconda**; *b*y all means use it!

A good tutorial can be found in
`https://swcarpentry.github.io/python-novice-inflammation/index.html`.

References