Asteroseismology
Quaternionic Transform Workflow
Technical part

Manu Parra-Royón - mparra@iaa.es
IAA-CSIC, Spain

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Table of contents

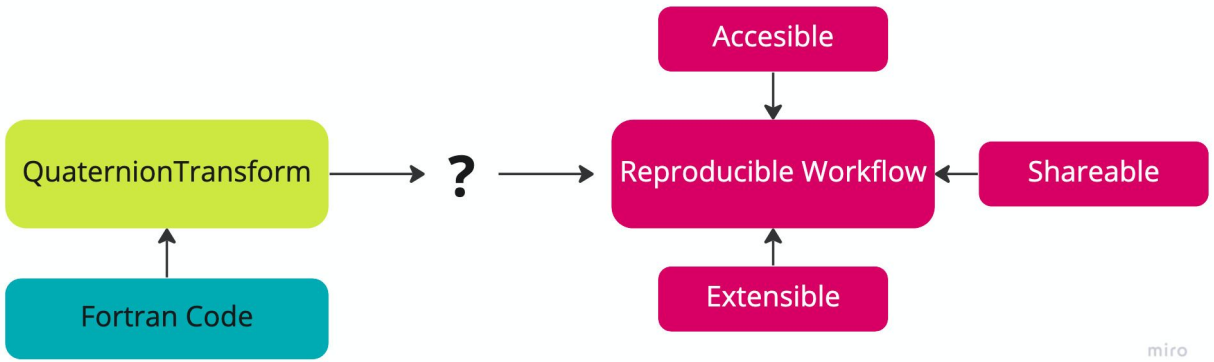
Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

>> Problem



>> We can use the ESAP platform to provide this workflow

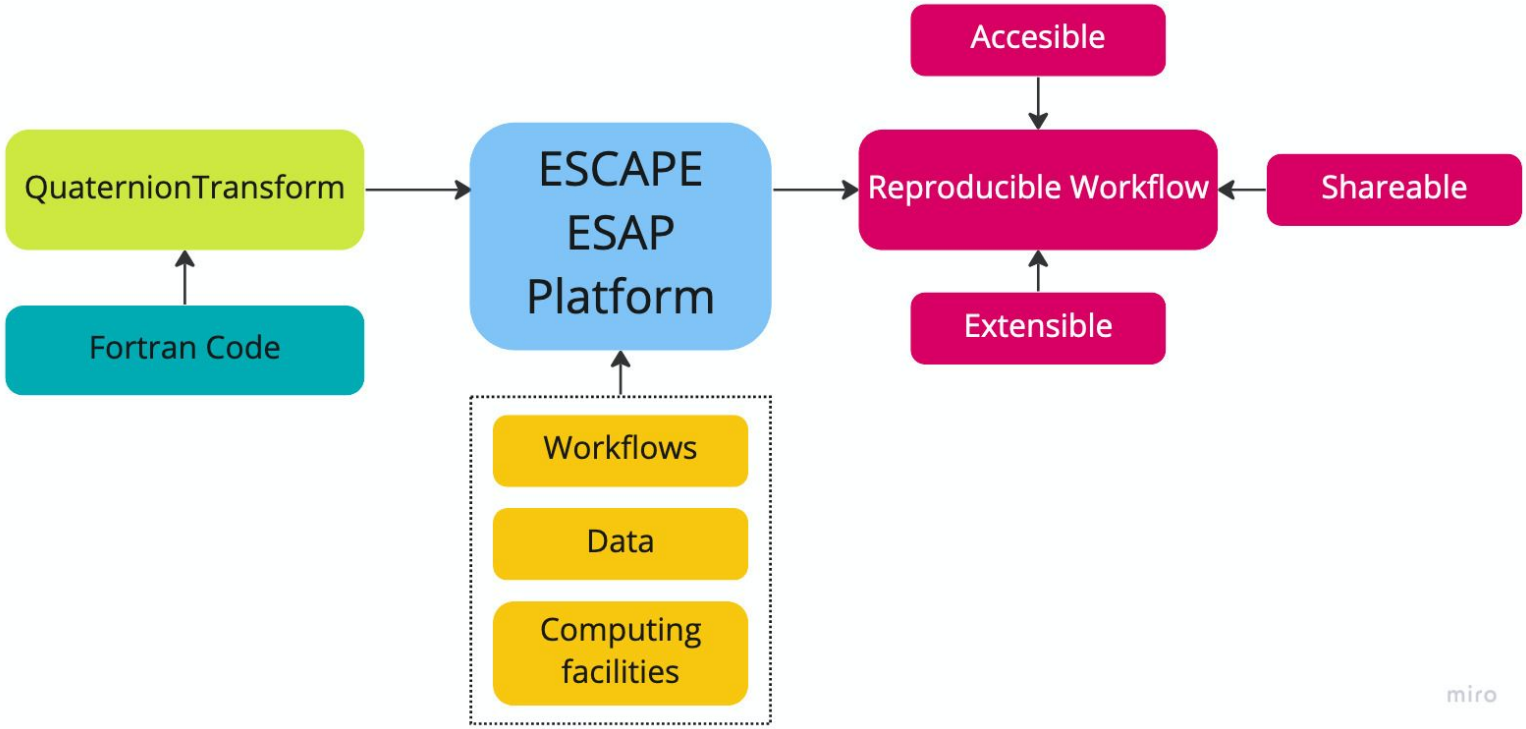
>> We need a tool that integrates well with ESAP

>> How about using a workflow within Jupyter notebooks?

FORTRAN code inside a Notebook ?

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction



Funded by the European Union's
Horizon 2020 - Grant N° 824064

Architecture and structure of the project and workflow

Funded by the European Union's Horizon 2020 - Grant N° 824064

Architecture and Structure

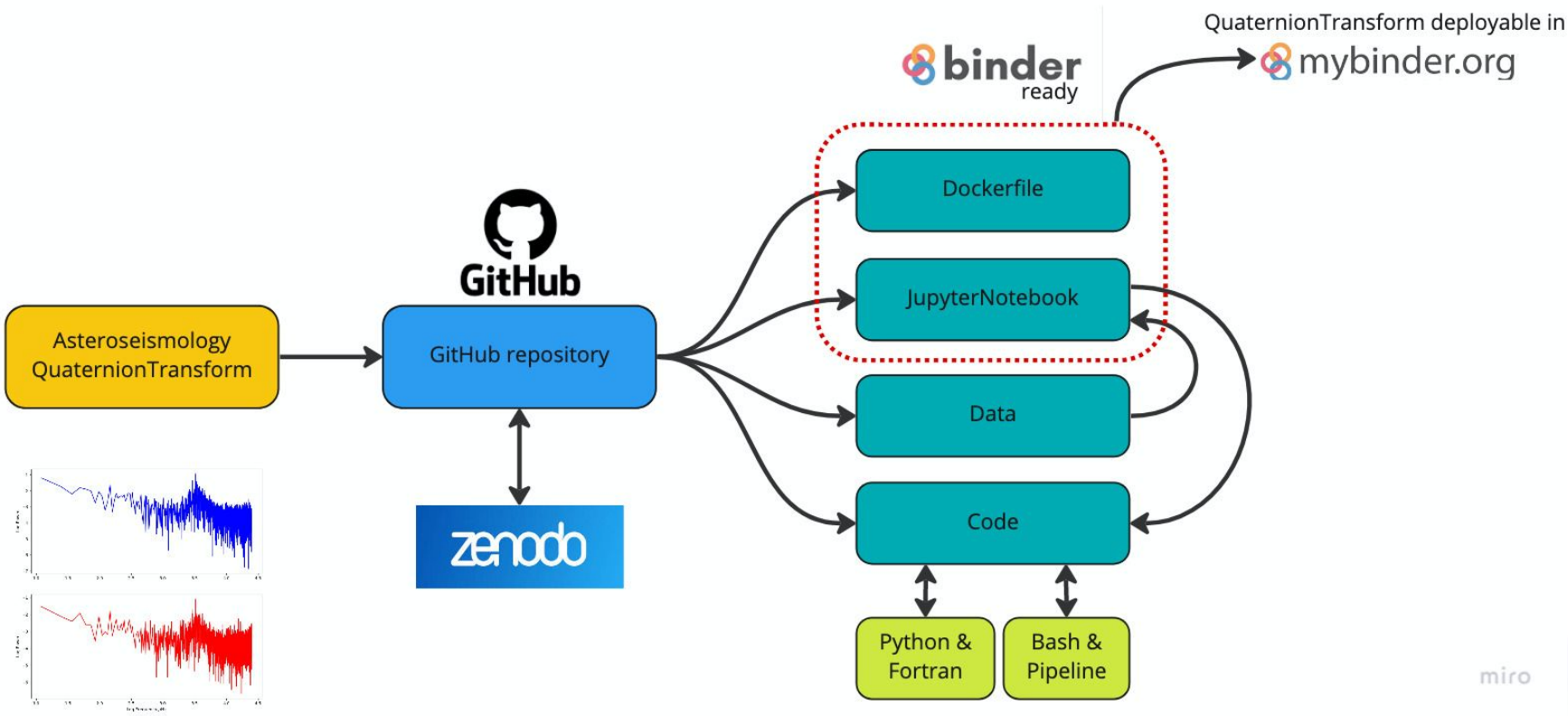
>> Four main parts

- A) Docker definition (Dockerfile) with a custom, containerised image that supports Fortran, Python Notebooks and a wrapper for Fortran-Python.
- B) Jupyter Notebook with sample code including Fortran code to load Quaternionic Transform and Python code to play with this function and create plots.
- C) Input data and native code
- D) A repository with all this content, necessary for a deployment on BinderHub.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Architecture and Structure

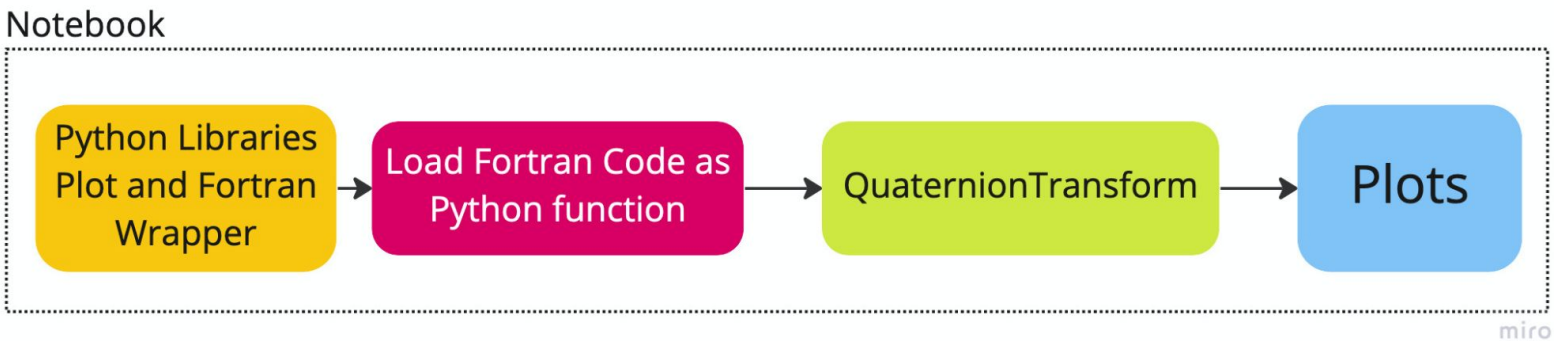
>> Overall diagram of the project



Funded by the European Union's Horizon 2020 - Grant N° 824064

Architecture and Structure

>> Overall diagram of the workflow



Funded by the European Union's Horizon 2020 - Grant N° 824064

Repository and documentation

Funded by the European Union's Horizon 2020 - Grant N° 824064

Repository and Documentation

>> Repository with Quaternion Transform

https://github.com/AsteroSeismologyIAA/QuaternionTransform

- Repository ready for BinderHub → Dockerfile to create an environment to deploy all the components ready to launch Quaternion Transform and a basic NoteBook with this workflow.

>> Documentation

- A Notebook has been added with the necessary steps for a basic use of the Quaternionic Transform function with which the user can work interactively.

Funded by the European Union's Horizon 2020 - Grant N° 824064

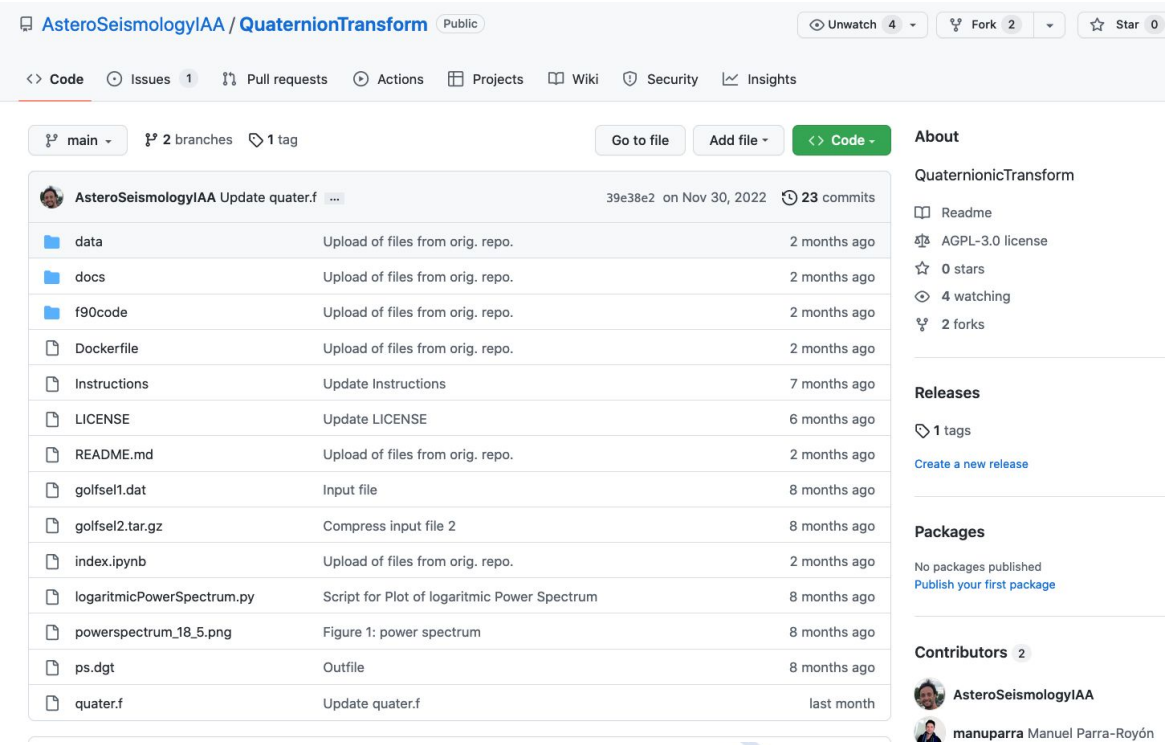
Demo

Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> Repository with Quaternion Transform

https://github.com/AsteroSeismologyIAA/QuaternionTransform



Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> Dockerfile

- Allows any Python workflow to be adapted to a BinderHub environment.

manuparra Upload of files from orig. repo. Latest commit a2e1d90 on Oct 31, 2022 History

1 contributor

27 lines (22 sloc) | 578 Bytes Raw Blame

```
FROM python:3.9-slim
RUN apt-get update
RUN apt-get -y install git gfortran
RUN pip install --no-cache --upgrade pip
RUN pip install --no-cache notebook jupyterlab
RUN pip install --no-cache matplotlib
RUN git clone https://github.com/mgaitan/fortran_magic && \
    cd fortran_magic && \
    python setup.py install

ARG NB_USER
ARG NB_UID
ENV USER ${NB_USER}
ENV HOME /home/${NB_USER}

RUN adduser --disabled-password \
    --gecos "Default user" \
    --uid ${NB_UID} \
    ${NB_USER}


COPY . ${HOME}
USER root
RUN chown -R ${NB_UID} ${HOME}
USER ${NB_USER}

WORKDIR ${HOME}
```


Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> Python Notebook

- Notebook that includes Fortran parts needed to load the Quaternion function as a callable function from Python.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Load extension for Fortran. It will enable fortran code within this notebook

In [ ]:
```
%load_ext fortranmagic
```

Load data Fortran routines (two ways)

First way: adding all the routines as python objects

In [ ]:
```
!f2py -c f90code/quater.f -m quater
```

Second way: Running the code directy from a cell (remove if production environment)

In [ ]:
```
%%fortran
```


Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> Python Notebook

- Notebook that includes Fortran parts needed to load the Quaternion function as a callable function from Python.

Execute quat

```
quat("data/golfsel1.dat",0,25000,16,20,"data/ps.dgt")
```

Plotting results

```
c1,c2,c3 = np.loadtxt("data/ps.dgt",unpack=True)

fig, (ax1, ax2) = plt.subplots(2, 1,figsize=(10,10))


ax1.plot(np.log10(c1), np.log10(c2), 'b-')
ax1.set_ylabel('Log Power')

ax2.plot(np.log10(c1), np.log10(c3), 'r-')
ax2.set_xlabel('Log Frequency $\mu$Hz')
ax2.set_ylabel('Log Power')
plt.savefig("powerspectrum_18_5.png")
plt.show()
```


Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> BinderHub ready !

- Dockerfile, Notebook and the data provide a solution ready to be deployed in BinderHub and therefore included in the ESAP catalogue.

Build and launch a repository

GitHub repository name or URL

GitHub ▾ https://github.com/AsteroSeismologyIAA/QuaternionTransform

Git ref (branch, tag, or commit) HEAD

Path to a notebook file (optional) Path to a notebook file (optional) File ▾

launch

Copy the URL below and share your Binder with others:

https://mybinder.org/v2/gh/AsteroSeismologyIAA/QuaternionTransform/HEAD

Expand to see the text below, paste it into your README to show a binder badge: launch binder

Already built Launching

Build logs view raw hide

```
Found built image, launching...
Launching server...
Server requested
2023-01-13T06:51:03.092415Z [Normal] Successfully assigned prod/jupyter-asteroseismolog-2d
erniontransform-2dbpurnn94 to gke-prod-user-202201-0769dbf8-gsvk
2023-01-13T06:51:03Z [Normal] Container image "jupyterhub/mybinder.org-tc-init:2020.12.4-0
.dev.git.4289.h140cef5" already present on machine
2023-01-13T06:51:03Z [Normal] Created container tc-init
2023-01-13T06:51:04Z [Normal] Started container tc-init
```

Demo

>> BinderHub ready !

- Dockerfile, Notebook and the data provide a solution ready to be

https://hub.gke2.mybinder.org/user/asteroseismolog-erniontransform-bpurnn94/lab/tree/index.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/

Name	Last Modified
data	a month ago
docs	a month ago
f90code	a month ago
Dockerfile	a month ago
golfsel1.dat	a month ago
golfsel2.ta...	a month ago
index.ipynb	a month ago
Instructions	a month ago
LICENSE	a month ago
logaritmic...	a month ago
powerspec...	a month ago
ps.dgt	a month ago
quater.f	a month ago
README.md	a month ago

Launcher | index.ipynb

Code

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Load extension for Fortran. It will enable fortran code within this notebook

[ ]:
```
%load_ext fortranmagic
```

Load data Fortran routines (two ways)

First way: adding all the routines as python objects

[ ]:
```
!f2py -c f90code/quater.f -m quater
```

Second way: Running the code directy from a cell (remove if production environment)

[ ]:
```
%%fortran

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c Objetive: Program to generate quaternion transform power spectrum, from a starting frequency (f
c Input Parameters:
c        file: Input file with data column
c        fmin: starting frequency
c        fmax: maximum frequency (fmax)
c        days: number of days recalculted in based of a power 2
```

Demo

>> BinderHub ready !

- Dockerfile, Notebook and the data provide a solution ready to be deployed in BinderHub and therefore included in the ESAP catalogue.

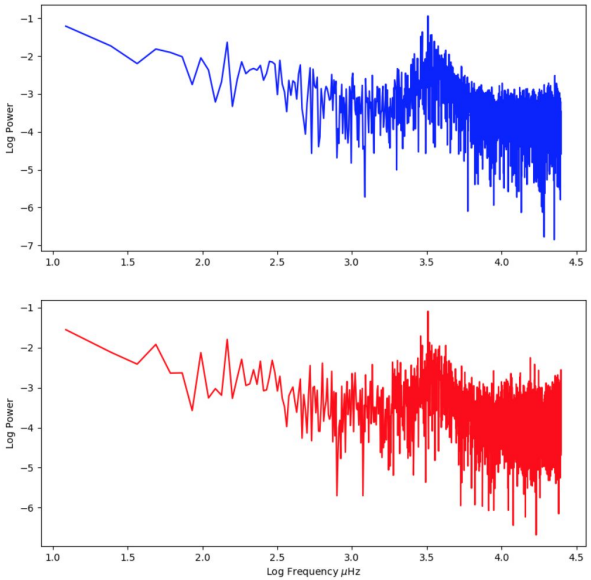
Plotting results

```
[ ]: c1,c2,c3 = np.loadtxt("data/ps.dgt",unpack=True)

fig, (ax1, ax2) = plt.subplots(2, 1,figsize=(10,10))


ax1.plot(np.log10(c1), np.log10(c2), 'b-')
ax1.set_ylabel('Log Power')

ax2.plot(np.log10(c1), np.log10(c3), 'r-')
ax2.set_xlabel('Log Frequency $\mu$Hz')
ax2.set_ylabel('Log Power')
plt.savefig("powerspectrum_18_5.png")
plt.show()
```



Funded by the European Union's Horizon 2020 - Grant N° 824064

Final remarks and future developments

Funded by the European Union's Horizon 2020 - Grant N° 824064

Demo

>> Integration with ESCAPE ESAP platform

>> Interactive notebook to run Fortran functions within Python Code

>> Quaternion Trans. workflow easily customisable

>> Access to BinderHub from the repository

>> Deployable in standalone mode with docker and from the cloud with BinderHub

>> Future work:

- a) workflow integration with Singularity containers,
- b) publishing Quaternion transform workflow container images to DockerHub and ArtifactsHub
- c) Improvements in efficiency and function design

Funded by the European Union's Horizon 2020 - Grant N° 824064