**FFM Installation**

1. Go to the following folder with fortran executables:

**fortran\_code/bin\_inv...,**

**fortran\_code/bin\_str...**

**fortran\_code/src\_dc... .**

In each one of them:

**make clean**

**make**

2. The folder ‘tectonicsplates-master’ must be at this level

**~/production\_code/tectonicsplates-master**

3. Correct location of surface waves GF bank.

.Go to **production\_code/fortran\_code/gfs\_nm.**

.Open file **low.in,** change location of file **fd\_bank** according to your home.  
  
Note: make sure you have file **fd\_bank**. This file has the green functions for surface waves. Without this file, you can’t use long period surface waves for finite fault modelling.

4. Modify file config.ini

Change the paths that corresponds to your own home (SEBASTIAN in my case)

Example:

[PATHS]

surf\_gf\_bank = /home/SEBASTIAN/production\_code/fortran\_code/gfs\_nm/long/low.in

code\_path = /home/SEBASTIAN/production\_code

modelling = /home/SEBASTIAN/production\_code/fortran\_code/bin\_inversion\_gfortran\_f95

get\_near\_gf = /home/SEBASTIAN/production\_code/fortran\_code/bin\_str\_f95

compute\_near\_gf = /home/SEBASTIAN/production\_code/fortran\_code/src\_dc\_f95

info = /home/SEBASTIAN/production\_code/fortran\_code/info

cartopy\_files = /home/SEBASTIAN/production\_code/fortran\_code/tectonicplates-master

sac\_exec = /usr/local/sac/bin/sac

This replaces the need to set environment variables. Because this allows the python scripts to locate the binaries needed for plotting, creating green functions and running kinematic modelling automatically.

5. Install the virtual environment:

First, you must create the (initially empty) virtual environment

**$ python3 -m venv ~/venv/finitefault**

Second, activate virtual environments:

**$ source ~/venv/finitefault/bin/activate**

Third, load required python packages (found on file requirements.txt) on virtual environment:

**$ pip3 install -r requirements.txt**

In case this gives problems, you need to install the packages by hand:

**$ pip3 install numpy**

**$ pip3 install scipy**

**$ pip3 install matplotlib**

**$ pip3 install obspy**

**$ pip3 install shapely**

**$ pip3 install cartopy**

**$ pip3 install geographiclib  
$ pip3 install netcdf4**

**(optional : $ pip3 install pyqt5)**

Every time you want to run this code, you must activate the virtual environment:

**$ source ~/venv/finitefault/bin/activate**

6. In case user wants, user can work with LITHO velocity model instead of CRUST2.0 velocity model. To do this, however, user needs to download file Litho1.0.nc from <https://ds.iris.edu/ds/products/emc-litho10/> and store it in   
  
**~/production\_code/fortran\_code/info**  
  
On the other hand, if user only wants to work with CRUST2.0 velocity model, user should go to file **velocity\_models.py**. There, user should uncomment the line:  
  
# crust\_model = \_\_crust\_crust\_velmodel(tensor\_info, default\_dirs)  
  
and comment the line   
  
crust\_model = \_\_litho\_crust\_velmodel(tensor\_info, default\_dirs)  
  
  
7. It is good idea to test whether fortran binaries are working properly prior to run automatic inversions:

1. Go to folder binaries\_test.

6. It is good idea to test whether fortran binaries are working properly prior to run automatic inversions:

1. Go to folder binaries\_test.

2. Modify Green\_strong.txt wth directions in your system.

3. Execute

$ ~/folder/fortran\_code/src\_dc…/green\_bank\_fk\_openmp (test creation of Green functions bank)

4. Execute

$ ~/folder/fortran\_code/bin\_str.../get\_strong\_motion (test creation of near field green functions for all used stations)

5. Execute

$ ~/folder/fortran\_code/bin\_inversion.../green\_tele (test creation of body wave green functions)

6. Execute

$ ~/folder/fortran\_code/src\_dc.../gf\_static\_f95 gps (test creation of static GPS green functions)

7. Execute

$ ~/folder/fortran\_code/bin\_inversion.../run\_modelling strong

(test kinematic modelling with strong motion data)

8. Execute

$ ~/folder/fortran\_code/bin\_inversion.../run\_modelling body

(test kinematic modelling with body wave data)

9. Execute

$ ~/folder/fortran\_code/bin\_inversion.../run\_modelling strong gps

(test kinematic modelling with strong motion and static GPS data)

**REQUIREMENTS**

gfortran, sac, y python (v3.4).

C for the Green functions of strong motion and the Green functions GPS static

**cartopy (**<https://scitools.org.uk/cartopy/docs/latest/>**).**

**PROJ (**<https://proj.org/>**)**

**GEOS (**<https://trac.osgeo.org/geos/>**)**

**——————————————————————————————————————**

**1. Modify Green.in con wherever you are going to add Green functions of strong motion.**

**2. Execute**

**$ ~/folder/fortran\_code/src\_dc…/green\_bank\_fk\_openmp (Green functions bank)**

**3. Execute**

**$ ~/folder/fortran\_code/bin\_str.../get\_strong\_motion**

**4. Execute**

**$ ~/folder/fortran\_code/bin\_inversion.../run\_modelling -strong**