

BUILD-RUN-DOCUMENT

Saurabh Sharma, Iñaki Martín Soroa

Last update: 17 October 2019

This document shows the steps needed to clone, build and run the FWI code and to install the prerequisite packages.

1 Pre-requisites

The prerequisite development tools needed can be installed using the following commands.

```
sudo apt install git qt5-default libeigen3-dev python3.7 python3.7-dev python3-tk
sudo apt install python3-numpy python3-matplotlib eog cmake
```

2 Cloning the Repository

To clone the FWI repository using git,

```
git clone -o redmine https://git.alten.nl/parallelized-fwi.git
```

This will create a copy of the repository, in a folder named *parallelized-fwi*

Any branch as needed can then be checked out from inside the *parallelized-fwi* folder. This document has been verified for the *gradientDescent* branch, later branches have added OpenCL dependencies, making this guide obsolete. In this example we use the *gradientDescent* branch.

```
git checkout gradientDescent
```

3 Build/Run

In this section the process to build the code is explained in two steps, namely downloading and installing Google Test and thereafter building the code and finally running it.

3.1 Install Google Test

First we need to install Google Test using the following commands:

```
sudo apt-get install libgtest-dev
cd /usr/src/gtest
sudo cmake CMakeLists.txt
sudo make
sudo cp *.a /usr/lib
```

3.2 Build

To build the project, first create a folder titled *build* outside the *parallelized-fwi* folder. Afterwards the code is built and run. NOTE: This folder should be exactly 1 level outside the *parallelized-fwi* folder.

```
cd ~
mkdir Build
cd Build
sudo cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX= ~/FWIInstall ../parallelized-fwi/
sudo make install
```

The first flag in the cmake command above enforces that the release version of the code be built and the 2nd flag implies that all the executables of the program will be placed in a folder *FWIInstall* (executables are in *FWIInstall/bin*)

3.3 Run

First check out the *inputFiles* folder in the parallelized-fwi folder and copy the *default* folder to the *FWIInstall* folder. This can be done by issuing the following command:

```
cp -r ../parallelized-fwi/inputFiles/default/ ~/FWIInstall
cp -r ../Build/runtime/bin/ ~/FWIInstall/bin
```

Now, the individual scripts for the preProcessing and the processing part can be run as shown below:

```
cd ~/FWIInstall/bin/
./FWI_PreProcess ../default/
./FWI_Process ../default/
```

For both executables we have to pass one argument which is the path to the case to be run (in this case *../default/*). Note that the arguments are the locations relative to the executable position. The default input parameters are stored in the *default/* which is located in *~/parallelized-fwi/inputFiles/*. Users can create their own set of input cards by copying the default folder into a new folder, modifying the input cards and giving the new folder's location argument to both PreProcess and Process.

For post-processing (i.e. generation of image using the estimated chi values and the residual plot), the python script *postProcessing.py* can be used. This script is located inside the *parallelized-fwi/pythonScripts* folder. This can be copied to the FWIInstall folder and then executed using the following commands:

```
cd ~/FWIInstall
cp ~/parallelized-fwi/pythonScripts/postProcessing-python3.py .
python3 postProcessing-python3.py default/
```

The run case folder is provided as the argument for the python script. The pre-processing, processing and the post-processing can all be grouped together using the python wrapper *wrapper.py* located inside the *parallelized-fwi/pythonScripts* folder. The postprocessed data can then be visualized using EOG from the output folder:

```
cd default/output/
eog defaultResult.png
```

4 Unit Test

The unit test executable is stored in the *FWIInstall/bin* folder after the make install command. The following commands can be used to run the test.

```
cd ../../bin
./unittest
```

5 Regression Test

This section details the comparison of a *default* run with the *fast* regression data. In order to run a regression test the following steps need to be taken. First, a *test* folder is created, then the contents of the regression data folder, the default folder, the input cards and the regression test python scripts are copied to this *test* folder. Then finally the python scripts can be run in succession.

```
cd ..
mkdir test
cp -r ~/parallelized-fwi/tests/regression.data/fast/ test/
cp ~/parallelized-fwi/tests/testScripts/*.py test/
cp -r default/ test/
cd test/
python3 regressionTestPreProcessing-python3.py fast default
python3 regressionTestProcessing-python3.py fast default
```