

BUILD-RUN-DOCUMENT

Saurabh Sharma

30 November 2018

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.

1. `sudo apt-get install git`
2. `sudo apt-get install qt5-default`
3. `sudo apt-get install libeigen3-dev`
4. `sudo apt-get install python2.7-dev`
5. `sudo apt-get install python2.7`
6. `sudo apt-get install python-tk`
7. `sudo apt-get install python-numpy`
8. `sudo apt-get install python-matplotlib`
9. `sudo apt-get install eog`

1.1 Install Google Test

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, e.g. the develop branch

```
git checkout develop
```

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 go back to your home directory, then make the googletest directory. Then download Google test from Github. Then, execute cmake, make and make install commands. Finally set the gtest root to the path of the working directory `$/googletest/install`.

```
cd ..
mkdir googletest
cd googletest
git clone https://github.com/google/googletest source
mkdir build
cd build
```

```

cmake = -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=~/.googletest/install/ ../source
make
make install
cd ..
cd install/
export GTEST_ROOT=$PWD
Finally, go back to the working directory for the next step:
cd

```

3.2 Build

To build the project, first create a folder titled **build** outside the **parallelized-fwi** folder. Afterwards the code is built and ran.

NOTE: This folder should be exactly 1 level outside the **parallelized-fwi** folder.

```

mkdir Build
cd Build
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX= ~/FWIInstall ../parallelized-fwi/
make install
Then go back back to the home directory by entering:
cd

```

3.3 Run

First enter the inputFiles folder parallelized-fwi folder and copy its contents to the FWIInstall folder. Then, create a output folder

```

cd parallelized-fwi/inputFiles/
cp * ../../FWIInstall
cd

```

```

cd FWIInstall/bin

```

```

mkdir output

```

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

```

cd bin
./FWI_PreProcess ../ ../output/ default
./FWI_Process ../ ../output/ default

```

The input parameters for the code are provided in the input card i.e. default.in. User can create his/her own input card with a new name e.g. newCard.in. To use this input card use the card name as an argument when running the executables, ./FWIPreProcess newCard and ./FWIProcess newCard.

For post-processing (i.e. generation of image using the estimated chi values), the python script `imageCreator_CMake.py` can be used. This script is located inside the **parallelized-fwi** folder and can be used as,

```
python imageCreator_CMake.py
```

The pre-processing, processing and the image creation step can all be grouped together using the python wrapper `wrap_FWI_CMake.py` located inside the **parallelized-fwi** folder.

```
python wrap_FWI_CMake.py
```

The data can then be visualized using EOG from the output folder:

```
eog defaultResult.png
```

4 Unit Test

5 Regression Test

In order to run a regression test the following steps need to be taken. First, the contents of a regression data test folder (in this example the fast example is shown) needs to be copied to the output folder. —Then the contents of the output folder are copied to the newly made test folder, as well as the regression data and the test scripts. Finally, the default input file is copied into test file. Then finally the python scripts can be ran in succession.

```
cd parallelized-fwi/tests/regression_data/fast
cp * ../../../../FWIInstall/output/
cd
cd FWIInstall
mkdir test
cd test
cp ../output/* .
cp ../../parallelized-fwi/tests/regression_data/fast/* .
cp ../../parallelized-fwi/tests/testScripts/regressionTestPreProcessing.py .
cp ../../parallelized-fwi/tests/testScripts/regressionTestProcessing.py .
cp ../default.in .
python regressionTestPreProcessing.py fast default
python regressionTestProcessing.py fast default
```