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 python3.7-dev`
5. `sudo apt-get install python3.7`
6. `sudo apt-get install python3-tk`
7. `sudo apt-get install python3-numpy`
8. `sudo apt-get install python3-matplotlib`
9. `sudo apt-get install eog`
10. `sudo apt-get install 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, 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
```
In order to set the `GTEST_ROOT` reference more permanent one, one has to add the `GTEST_ROOT` to bashrc file by using vim or another editior.
```
cd
vim ~/.bashrc
```
Then, press i or insert to modify the file and add the following line to the file:
```
export GTEST_ROOT=/googletest/install
```
Then, exit insert mode by pressing esc. To save bashrc and exit vim, enter the `:wq` command. Then, in order to take the changes in bashrc into effect enter:
```
source ~/.bashrc
```
Another way of installing GTest
```
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.
mkdir Build
cd Build
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX= ~/FWIInstall ../parallelized-fwi/
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.
```
cd ~/FWIInstall
cp -r ../parallelized-fwi/inputFiles/default/ .
```
Now, the individual scripts for the preProcessing and the processing part can be run as shown below:
```
cd 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.

Note: It is very useful to add the *~/FWIInstall/bin/* folder to the PATH environment variable. This means that both the `FWI_PreProcess` and `FW_Process` commands can be run from any folder without having to give the path to where they can be found. This can be achieved by issuing the following commands.
```
vim ~/.bashrc
```
Add the following line at the end of this file (don't forget the colon!).
```
export PATH=$PATH:~/FWIInstall/bin/
```
Now execute the following command in your terminal.
```
source ~/.bashrc
```
Now you can run `FWI_PreProcess` and `FW_Process` commands from anywhere in your terminal!

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 used as,

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

The folder where all the output files are located 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 ~/FWIInstall/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 command can be used to run the test.

```
cd ~/FWIInstall/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 a regression data folder, the output folder, the input card (default.in) and the regression test python scripts are copied to this *test* folder. Then finally the python scripts can be run in succession.

```
cd ~/FWIInstall/
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
```