How to add an new inversion method

Iñaki Martín Soroa

22 October 2019

1 Naming conventions and structure

In order to keep the code well structured, follow the structure used for all the other inversion methods, for this example we consider that we have created the method fooBar:

- Your inversion method must implement *inversionInterface*, found in `libraries/core/inversionInterface.h`.
- Use an original name with camelCase ending with "Inversion". For example: *fooBarInversion*. Please avoid unoriginal names such as: *fooBarInversion*, *myInversion* or *fasterConjugateGradientInversion*.
- Use the same folder structure as other inversion methods. Create a folder in the directory `libraries/inversion/` with the name of you inversion method, e.g. *fooBarInversion*. That folder must contain a `CMakeLists.txt` file, a *src* folder with all the `.cpp` files and a *include* folder with all the `.h` files.
- All the files and classes related to your method should include the inversion method name at the beginning, i.e. an input card reader can be called *fooBarInversionInputCardReader*. The name is long but much more understandable for anyone else using the code later

2 Setting up the Factory

The *inversionFactory* class takes care of instantiating the desired inversion method. It is the only C++ file that you need to modify outside of your folder. The file is in
`libraries/inversion/inversionFactory/src/inversionFactory.cpp`

- Add your method to the includes, i.e. `#include "fooBarInversion.h"`.
- Add an if condition for your method following the same structure as the other methods, namely:
  ```
  if (desired_method == "fooBarInversion"){
  inversion = new fooBarInversion(forwardModel, gInput);
  return inversion;}
  ```

3 Setting up CMake

We need to edit 3 different `CMakeLists.txt` files to ensure that the libraries are properly compiled and linked.

- In `libraries/inversion/CMakeLists.txt`, add a line BEFORE `add_subdirectory ( inversionFactory )` with the name of your folder, i.e. `add_subdirectory ( fooBarInversion )` . This adds the folder containing your method to the compilation path.
- In your folder, i.e. `libraries/inversion/fooBarInversion`, the file must compile your method and need to include the libraries that it depends on. The simplest option is to copy the `CMakeLists.txt` file in another method, i.e. *randomInversion*, and change where it says RANDOM or random for FOO_BAR or foo_bar respectively. If you have to add dependencies to some other library, you can add it also in that file.
- In `libraries/inversion/inversionFactory/CMakeLists.txt` we need to make the library with your method available to the *inversionFactory*. In order to do that add your library in the same way that the other inversion methods have been added.
  - Add a line adding your directory using `include_directories`, i.e. `include_directories( ${LIBRARY_INCLUDE_DIRS_FOO_BAR_INVERSION})`
  - Add a line linking your library to the *inversionFactory* library. This is added at the end in the command `target_link_libraries(...)`. Add your library in the same way that the other inversion methods are added, i.e. `foo_bar_inversion_library` separated by a space from the other linked libraries.