

AQUA_{gpusph} - 1.5.03 .

Developers guide

JL Cercos-Pita <jl.cercos@upm.es>

September 10, 2014



Contents

1	Introduction	5
1.1	About AQUAghusph	5
1.2	License notes	5
1.3	Objectives of the present document	6
1.4	Required background	6
2	AQUAghusph strcuture	7
2.1	General	7
2.2	Kernel	7
2.3	Computational tools	9
2.4	Helper tools	9
2.5	Examples	9
2.6	Documentation	10
2.7	CMake	11
	Appendices	13
A	Directory tree	15

Chapter 1

Introduction

1.1 About AQUAagpusph

AQUAagpusph (Another QUALity GPU-SPH) is a free CFD software licensed under GPLv3, and developed by J.L. Cercos-Pita as part of his PhD at CEHINAV-UPM group ¹. AQUAagpusph is based on Lagrangian meshfree SPH (Smoothed Particles Hydrodynamics) method.

AQUAagpusph has been accelerated with OpenCL, that allows you to execute it over CPU based platforms, over GPUs based platforms, and eventually, over every platform developed in the future adapted to the OpenCL standard. For the moment the simultaneous use of several platforms in the same simulation is not supported.

The code has been widely validated, used and documented in several publications, for instance ?] where boundary integrals are tested or ?] where SPHERIC benchmark test case number 9 was simulated. Some examples provided within the package have been studied in these publications.

AQUAagpusph has been designed for UNIX like operative system, and tested on GNU/Debian Linux distributions. For the moment no ports has been developed for Windows or Mac operative systems. AQUAagpusph has been developed in C++ and OpenCL². Also Python extensions has been developed.

1.2 License notes

AQUAagpusph is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

¹Model Basin Research group, Technical University of Madrid (UPM), <http://canal.etsin.upm.es>

²OpenCL is so quite similar to C

AQUAgnusph is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License in 'LICENSE' file with AQUAgnusph package. If not, see <http://www.gnu.org/licenses/>.

1.3 Objectives of the present document

This document has been created aiming to provide a tool that allows the AQUAgnusph developers to know:

1. The structure of AQUAgnusph source code.
2. The location of the source codes and helper files.
3. The entry points to the main parts of AQUAgnusph

This document is trying to complement the Doxygen documentation. Such documentation can be get, either building it with CMake (see the user manual) or in the following web page (corresponding to the last stable AQUAgnusph package):

<http://canal.etsin.upm.es/aquagnusph/doc/doxygen/stable>

1.4 Required background

The background required for the developers depends on the section of the code to be read/modified:

1. The kernel of AQUAgnusph is written C++.
2. The computational tools are written in OpenCL.
3. The examples are built using XML files.
4. Also Python extension may be used in some examples.
5. The documentation (this document as well as the developers guide) is written in \LaTeX .
6. The configuration and compilation is carried out with CMake.

Also, depending on the modifications to be done, some background in CFD, and more specifically in SPH, may be required.

Chapter 2

AQUAAgpusph strcuture

2.1 General

AQUAAgpusph package is divided in several parts:

1. AQUAAgpusph kernel.
2. Computational tools.
3. Helper tools.
4. Examples.
5. Documentation.
6. CMake configuration and compiling files.

2.2 Kernel

The kernel on AQUAAgpusph is basically the main program which will be executed in the CPU. When it is launched the following operations will be owned by the main program:

1. The input XML files loading.
2. The computational device setup.
3. The OpenCL kernels loading.
4. The OpenCL codes execution querying (this codes will be ran in the computational device).
5. The Python scripts loading and executing.

6. The output files writing.

The source code files are placed in both **src/** and **include/** package folders.

The kernel is divided in two main parts, widely denominated “host” and “server”. Such division can be appreciated in figure 2.1. Developers interested in the “host” side, mainly responsible of the input and

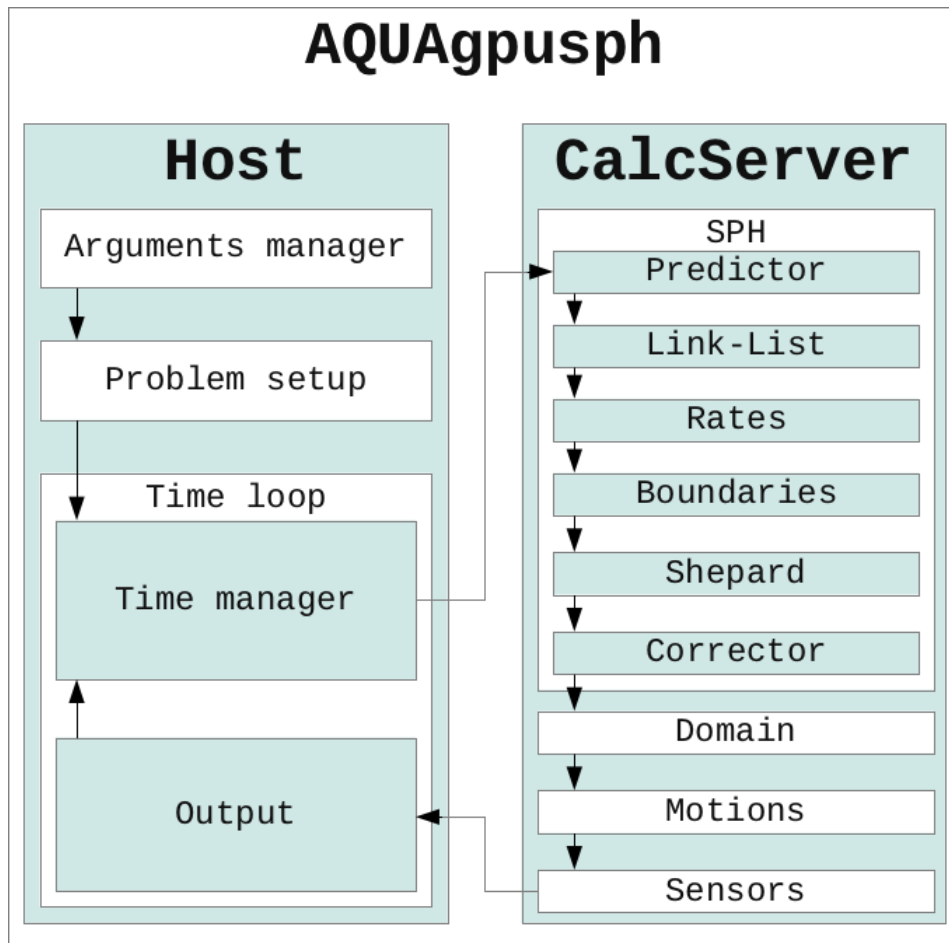


Figure 2.1: General AQUAgpupsh flux diagram

output, may start with the file **src/main.cpp**.

Developers interested in the “server side”, designed to control the simulation querying the OpenCL codes execution, may start with the class **CalcServer**.

You can learn more about the source code files location and objective in the Doxygen documentation:

<http://canal.etsin.upm.es/aquagnusph/doc/doxygen/stable/files.html>

2.3 Computational tools

The computational tools are OpenCL codes placed in **resources/OpenCL** folder. These codes are running in the computational device, and therefore are widely related with the kernel “server” side tools.

You can learn more about the OpenCL codes in the Doxygen documentation:

<http://canal.etsin.upm.es/aquagnusph/doc/doxygen/stable/files.html>

It should be mentioned that, being these codes the provided with AQUAgnusph package, they can be replaced by other ones for specific simulations just setting it in the input XML files.

2.4 Helper tools

With AQUAgnusph package several helper tools are provided. These tools are placed in the **tools/** folder.

Probably the most important tools are placed in **tools/aquagnusph_preprocessing**, being the particles generators. The generators are scripts that can read a mesh and generate the boundary particles in the mesh and the fluid ones inside it.

2.5 Examples

The examples are placed in the **examples/** folder. The examples available depends on whether the 2D or the 3D version of AQUAgnusph is built. In AQUAgnusph package each example has its files divided in 2 folders, namely:

1. **examples/*example_name***
2. **examples/cMake/*example_name***

where *example_name* is replaced by the example folder name. In the first folder the files which does not require any specific configuration are placed, while in the latter folder the files which require to be modified according to the cMake configuration are placed. When the examples are built, all the files (conveniently configured) are placed in the first folder.

In the built examples following files may be found:

1. Create.py: Python script to generate the particles input file.

2. Fluid.dat: Particles file generated by Create.py.
3. Main.xml: Main input XML file, it is just including other XML files.
4. Settings.xml: General program settings: http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/d8/dfe/structAqua_1_1InputOutput_1_1ProblemSetup_1_1sphSettings.html
5. Time.xml: Simulation time flow options: http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/de/d49/structAqua_1_1InputOutput_1_1ProblemSetup_1_1sphTimingParameters.html
6. SPH.xml: Physics simulation parameters: http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/d6/d08/structAqua_1_1InputOutput_1_1ProblemSetup_1_1sphSPHParameters.html
7. GhostParticles.xml: Ghost particles boundaries (just when Ghost particles boundary condition is applied): http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/d5/d89/classAqua_1_1InputOutput_1_1ProblemSetup_1_1sphGhostParticles.html
8. Fluids.xml: Fluids physics properties: http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/d7/da8/structAqua_1_1InputOutput_1_1ProblemSetup_1_1sphFluidParameters.html
9. Movements.xml: Motions to be parsed: http://canal.etsin.upm.es/aquagpusph/doc/doxygen/stable/dd/df7/classAqua_1_1InputOutput_1_1ProblemSetup_1_1sphMoveParameters.html
10. Move/6DOF.xml: Specific motion type parameters.
11. Move/move.py: Python script which controls the solid motion (just in Python script controlled motions).
12. run.sh: Helper running script, type “./run.sh –help” in the example folder to learn more.

Some other files may be found, for instance, to plot the results in real time, or to load the prescribed solid motions.

Main.xml file is also including the shared XML file **resources/OpenCLMain.xml**, which is automatically setting all the computational tools discussed in the section 2.3.

2.6 Documentation

The documentation files, which include the user manual, the developers guide and the Doxygen documentation, are placed in the folder **doc/**.

In AQUAagpusph it is expected that the Doxygen documentation is built using cMake, so please do not try

to build it manually.

Regarding the user manual and the developers guide, AQUAopus is providing already compiled version of them, however they can be edited and recompiled using the **buildPDF.sh** script provided with each one.

2.7 CMake

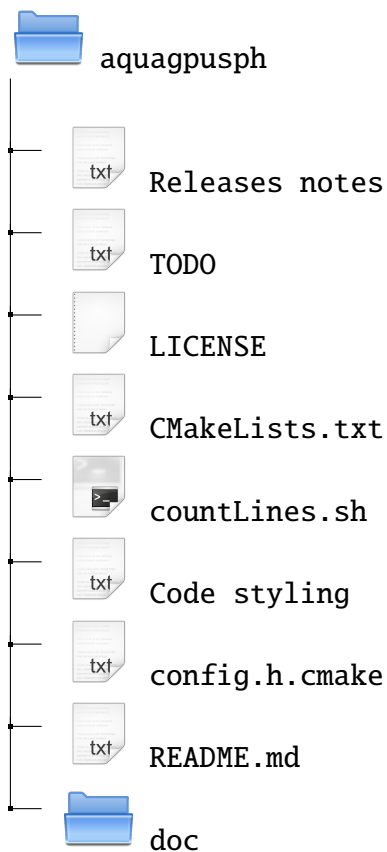
All the files called “CMakeLists.txt”, “*.cmake”, or included in a “cMake” folder are designed to let CMake to configure the package for the build and installation process. More specifically, the files called “CMakeLists.txt” are the ones where the instructions for CMake are written, while the other ones are templates that CMake is conveniently reading and editing.

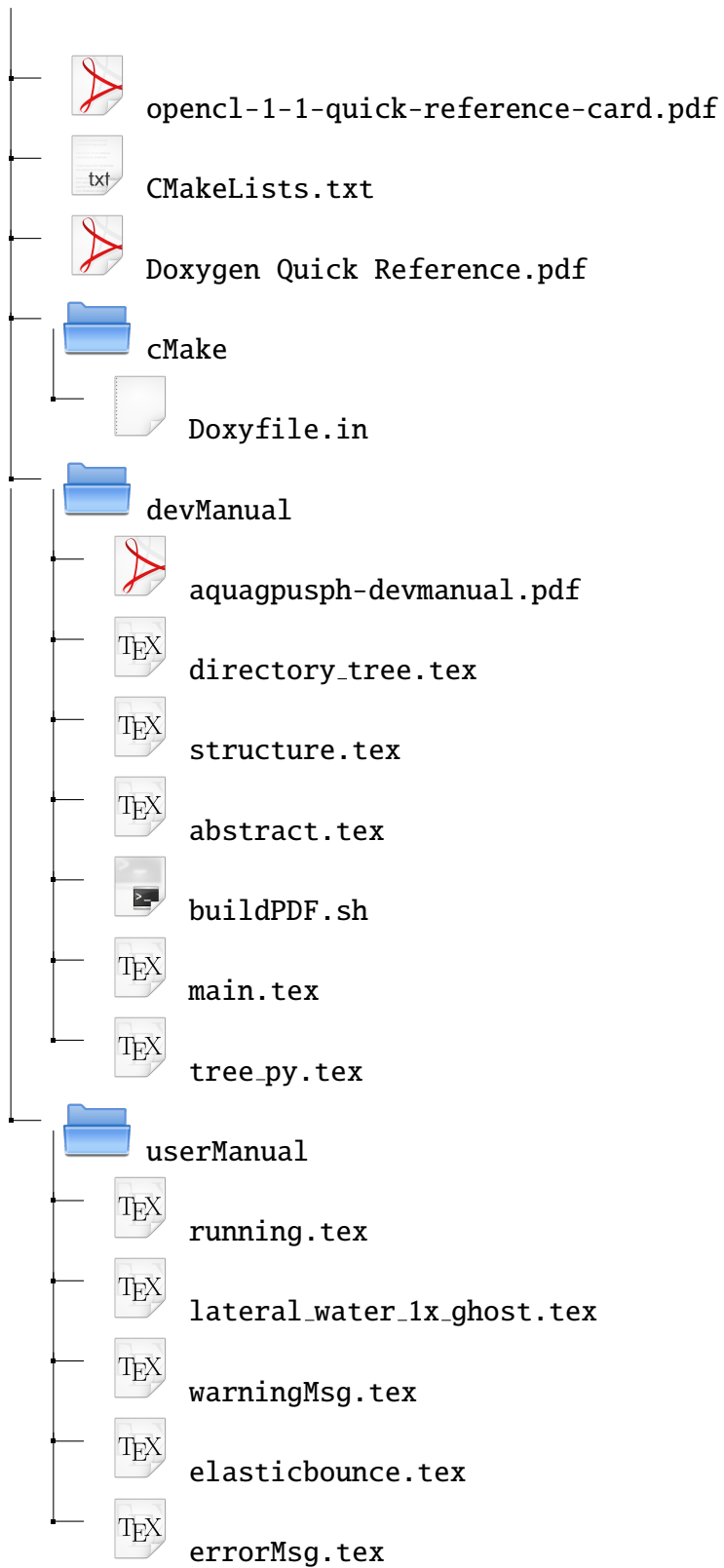
Appendices




















Appendix A

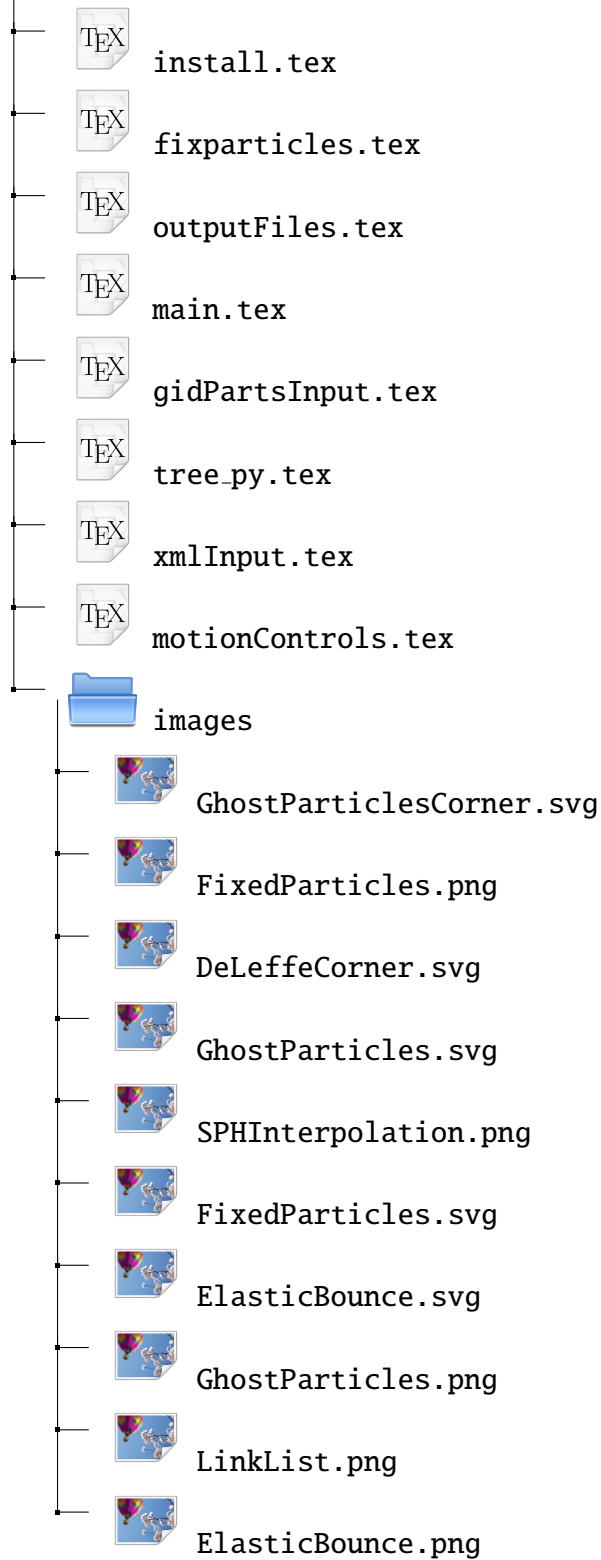
Directory tree

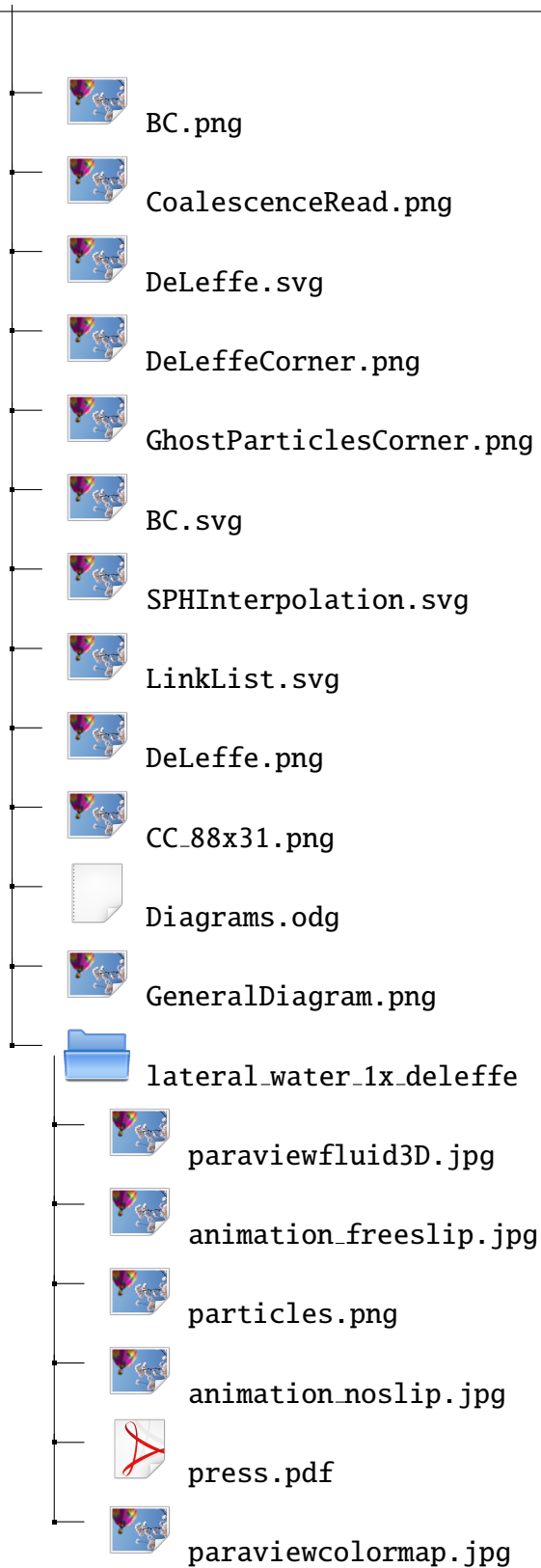
Following all the files provided with AQUAopusph package, with its location, is shown. The files autogenerated during the configuration, build and install processes are not shown.

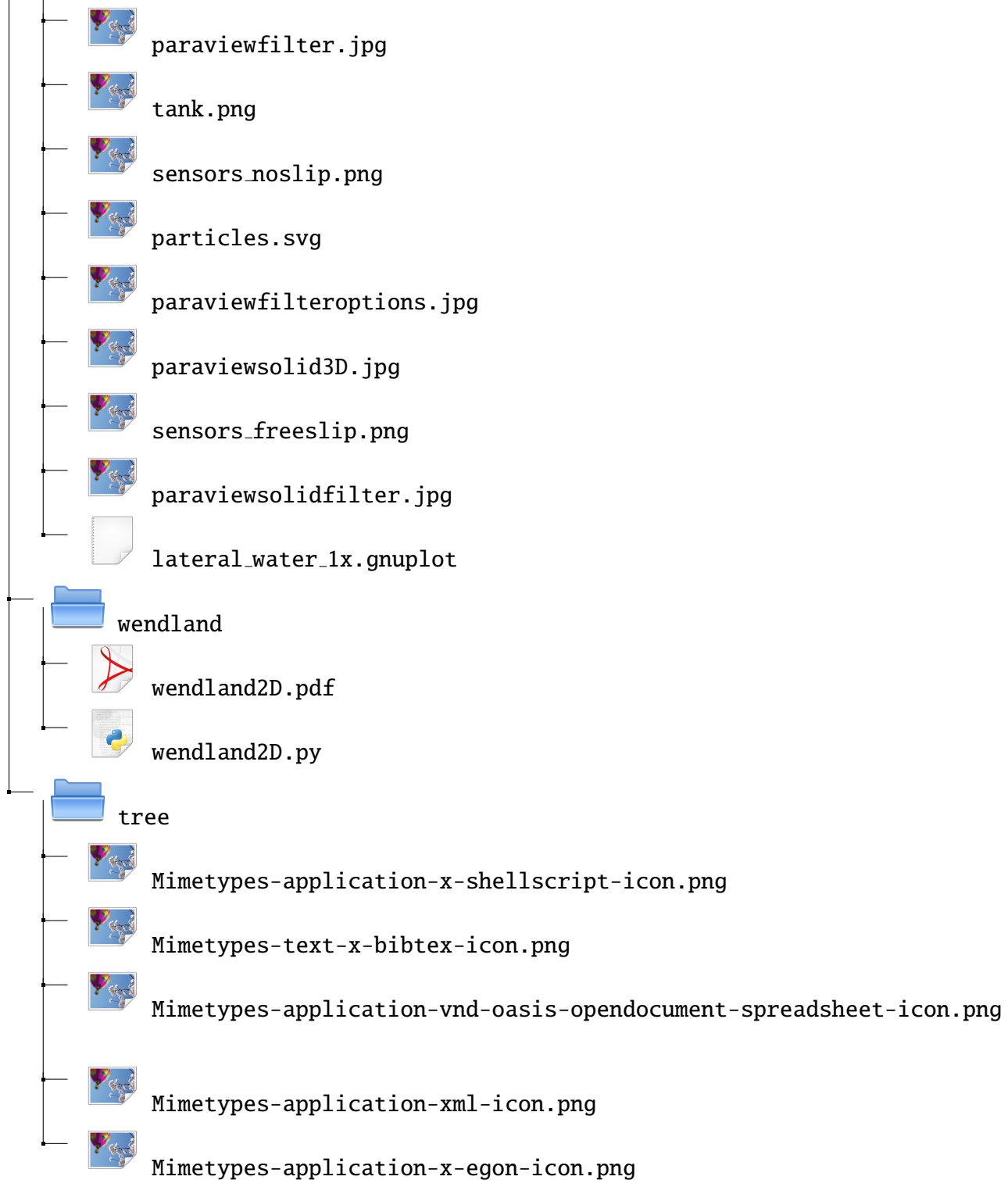


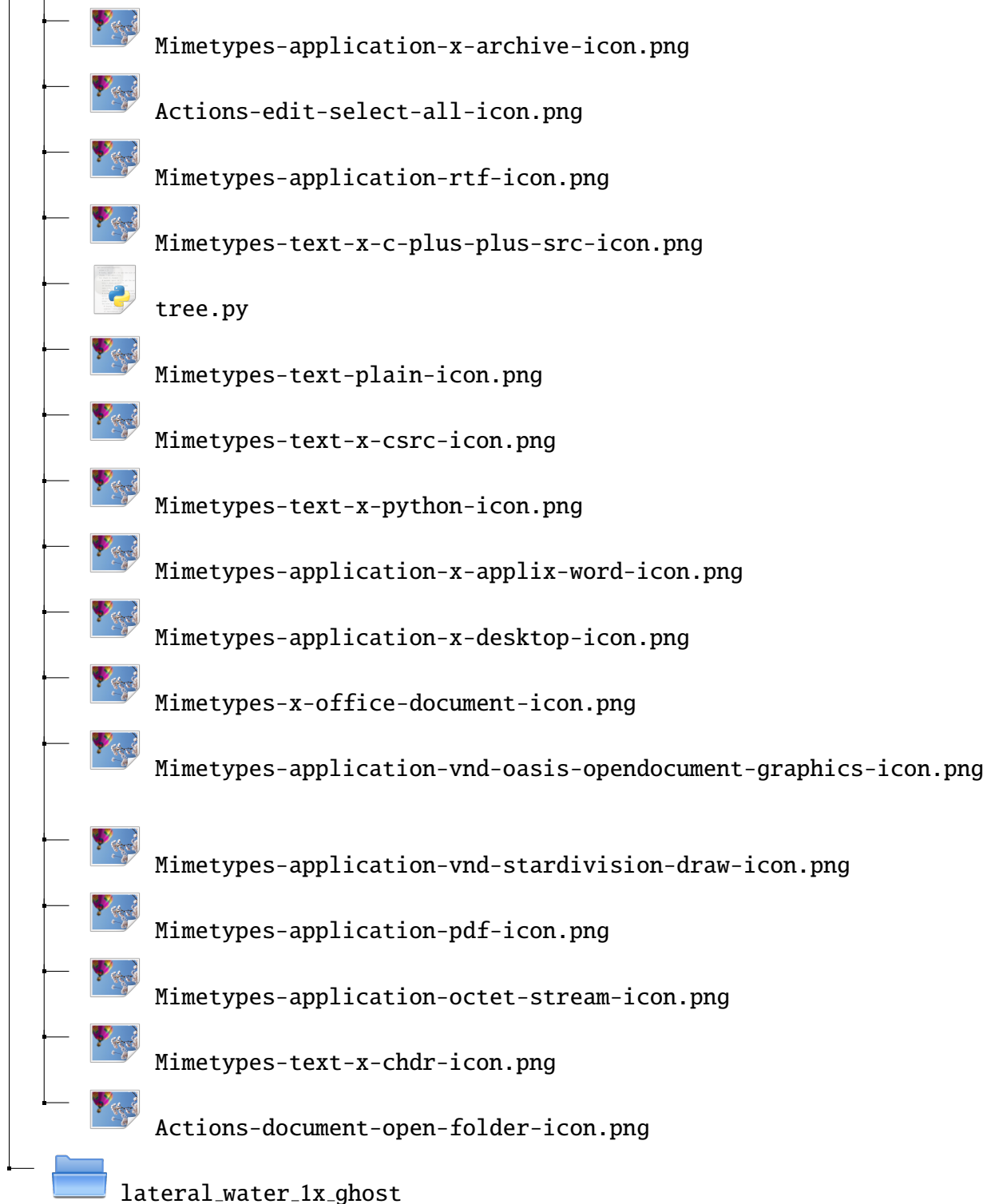


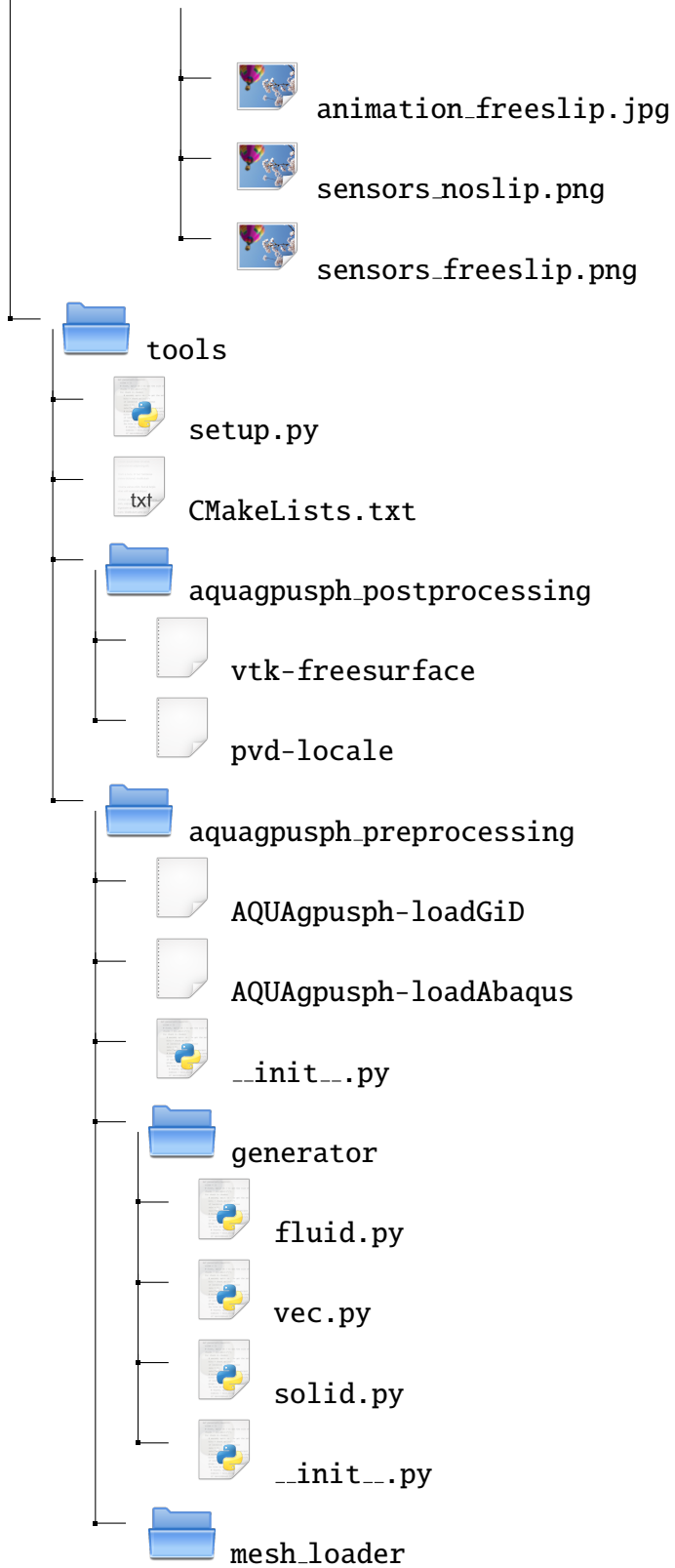
	directory_tree.tex
	examples.tex
	asciiPartsInput.tex
	motionTypes.tex
	paraview.tex
	texput.log
	sensors.tex
	DeLeffe.tex
	infoMsg.tex
	ghostparticles.tex
	partsInput.tex
	aquagpusph.tex
	abstract.tex
	xmlPartsInput.tex
	lateral_water_1x_deleffe.tex
	buildPDF.sh
	bib.bib
	aquagpusph-usermanual.pdf
	physics.tex

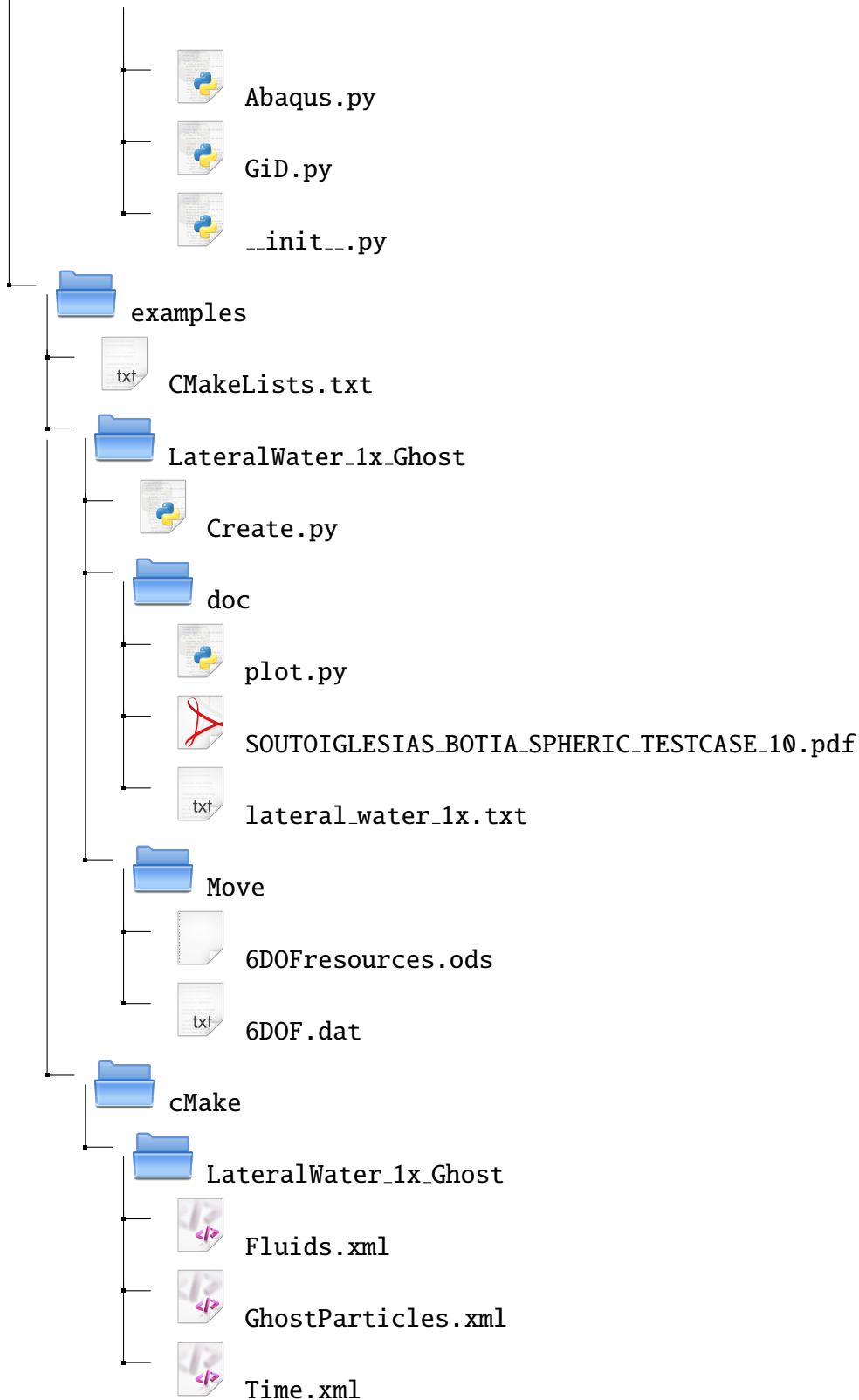


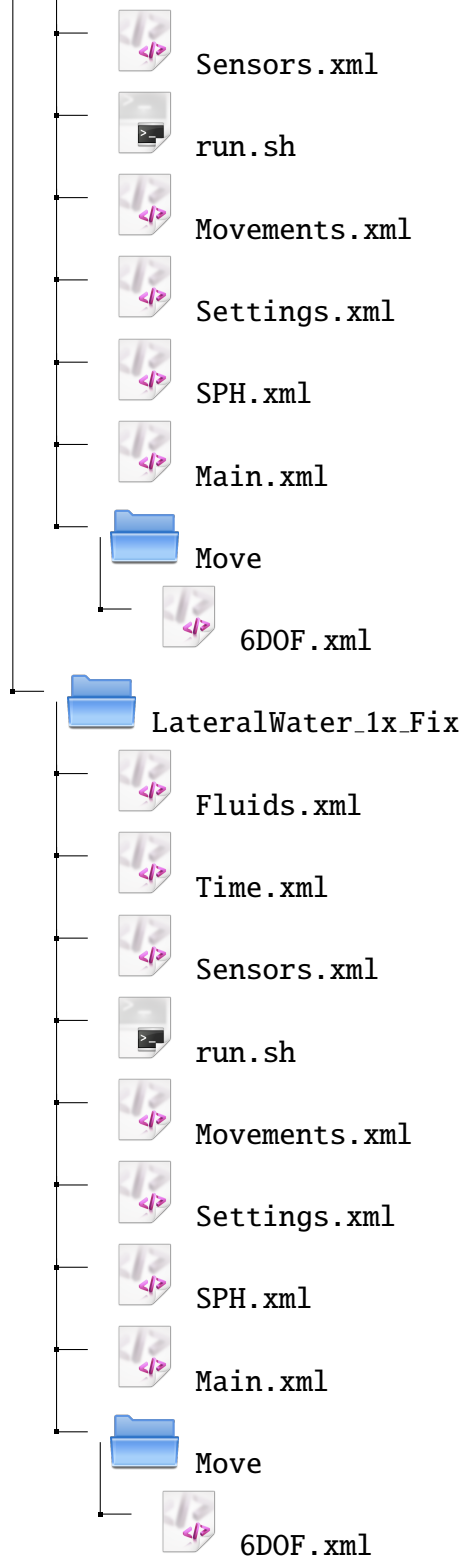


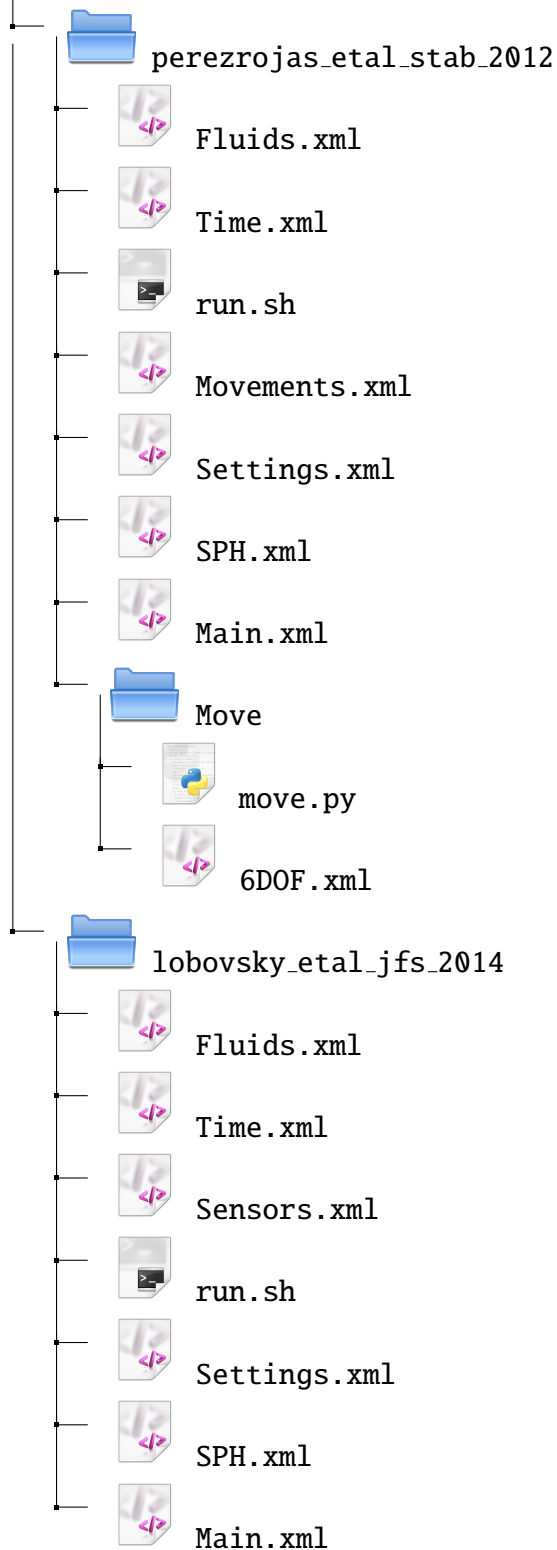


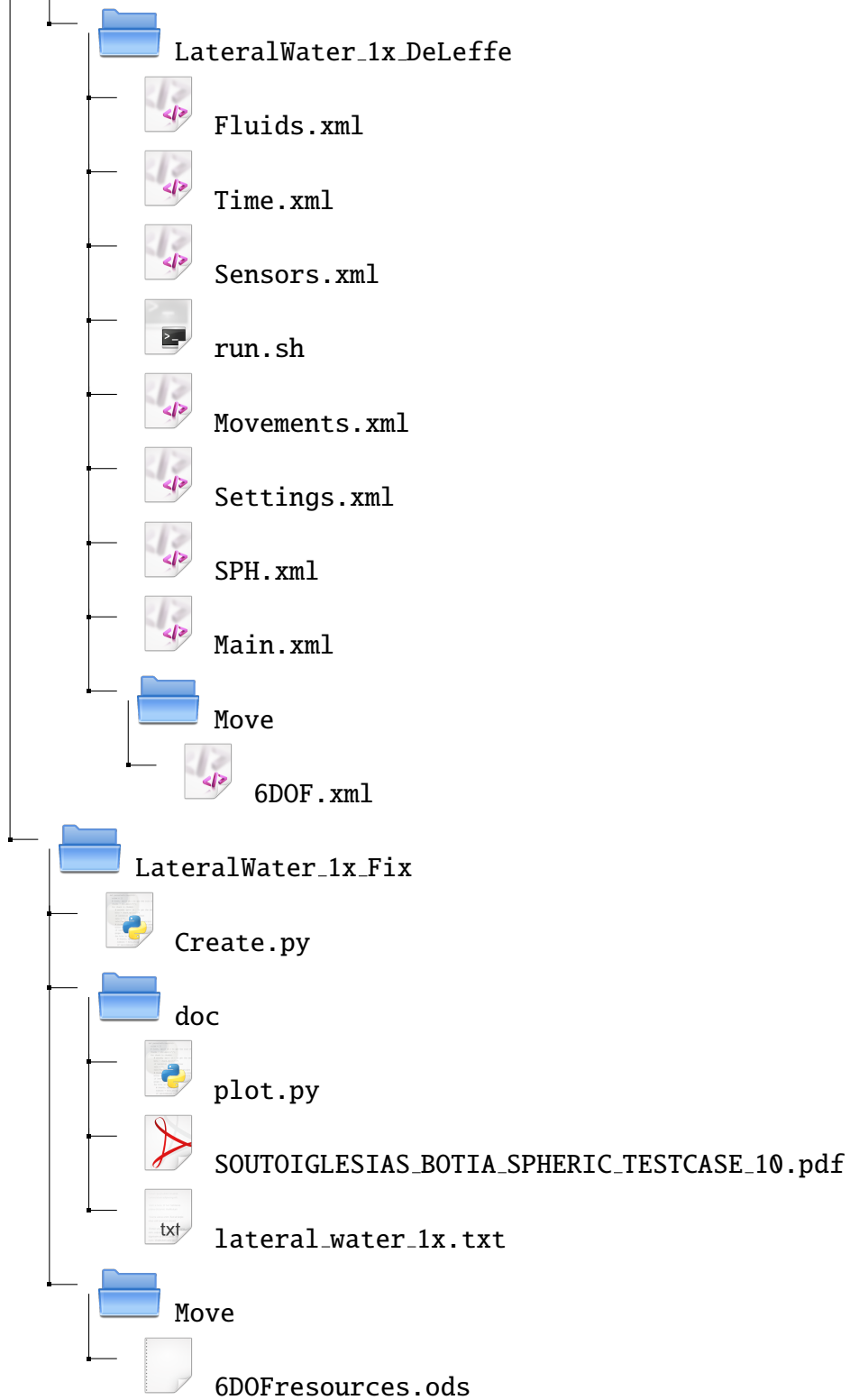


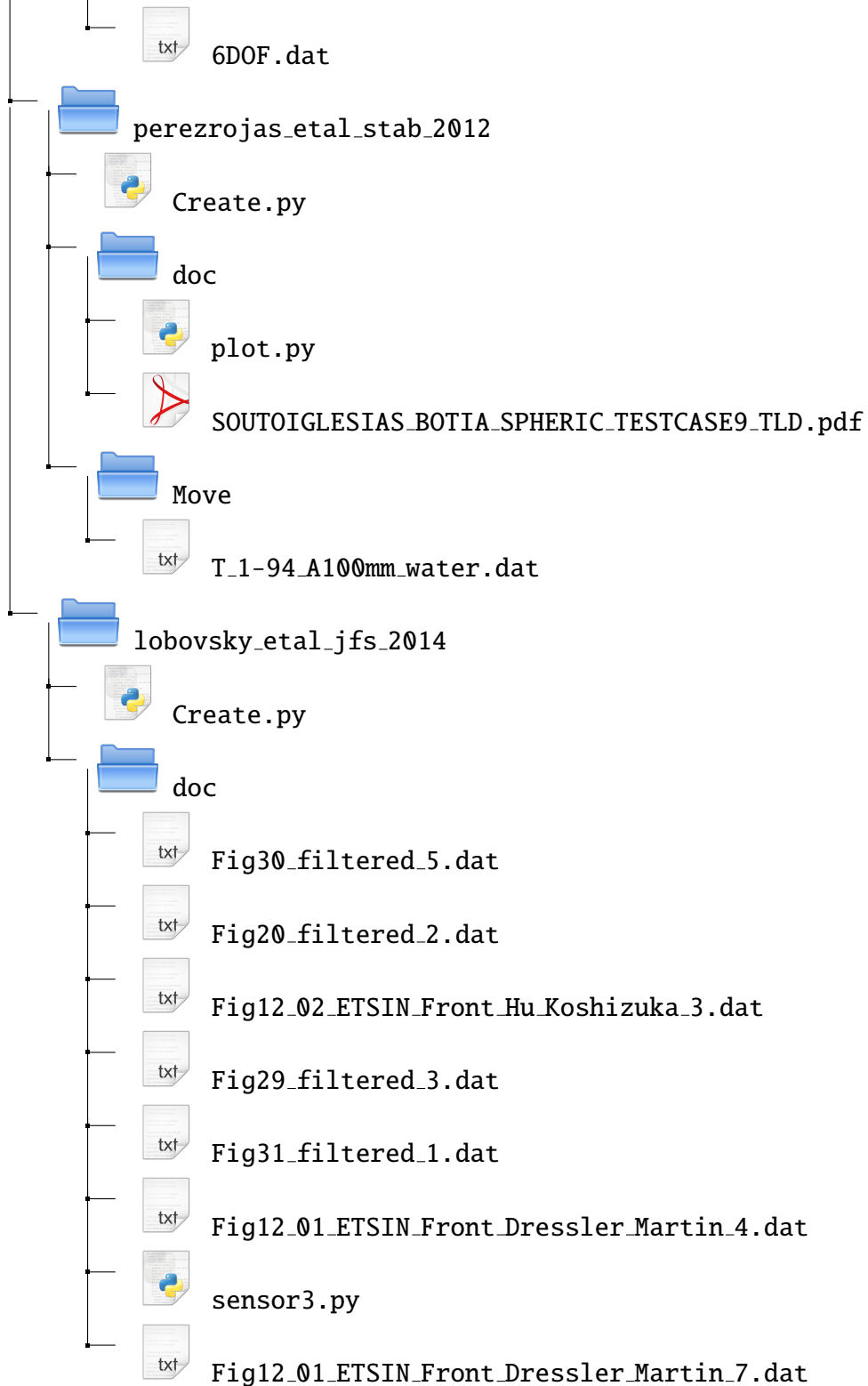























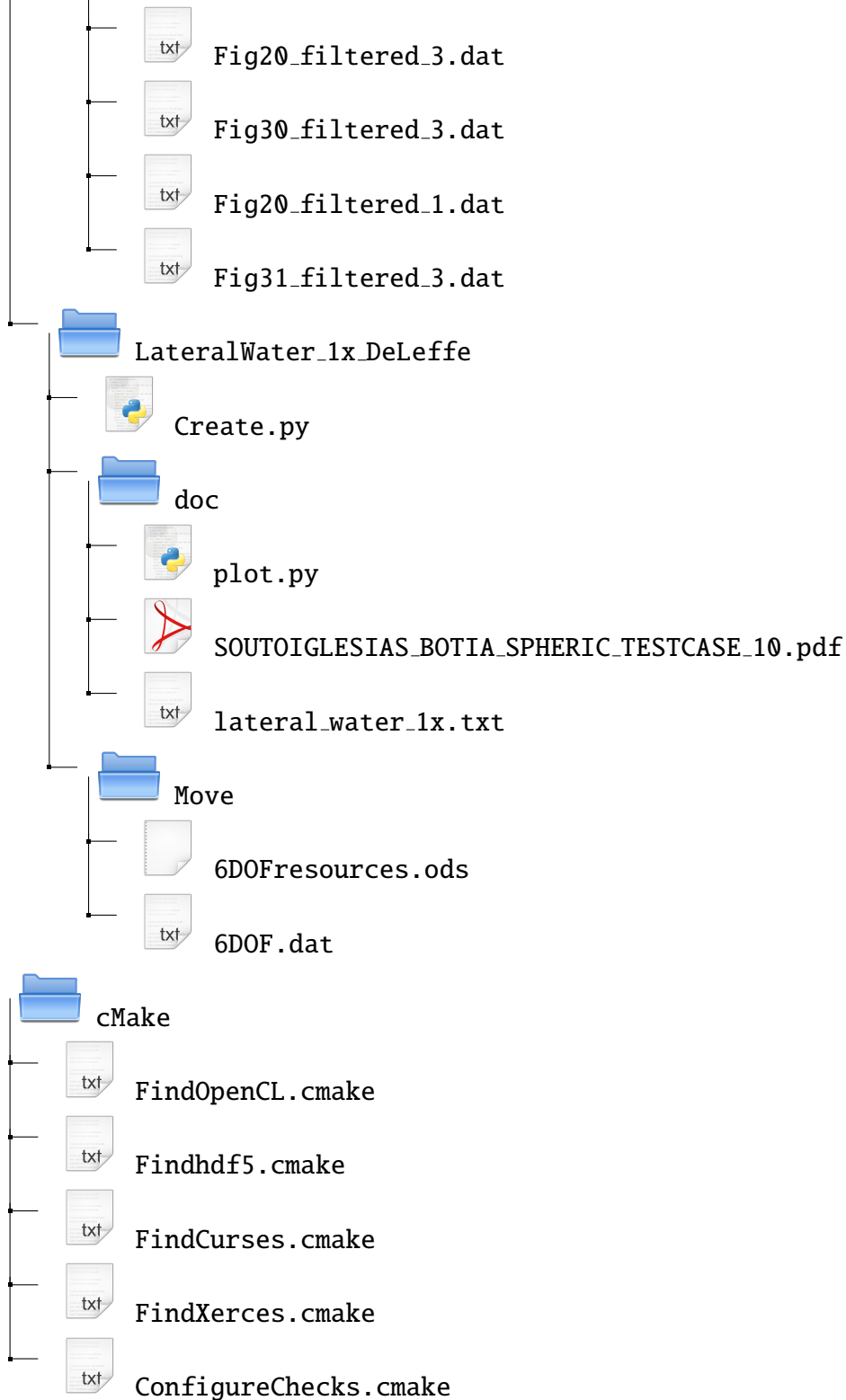


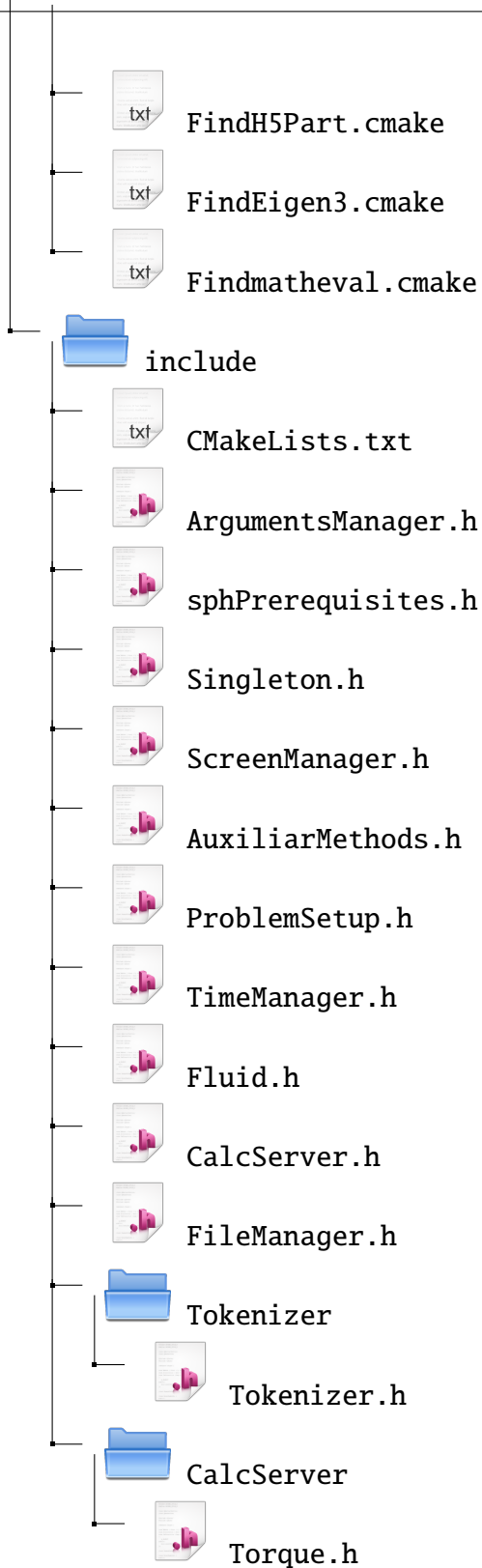


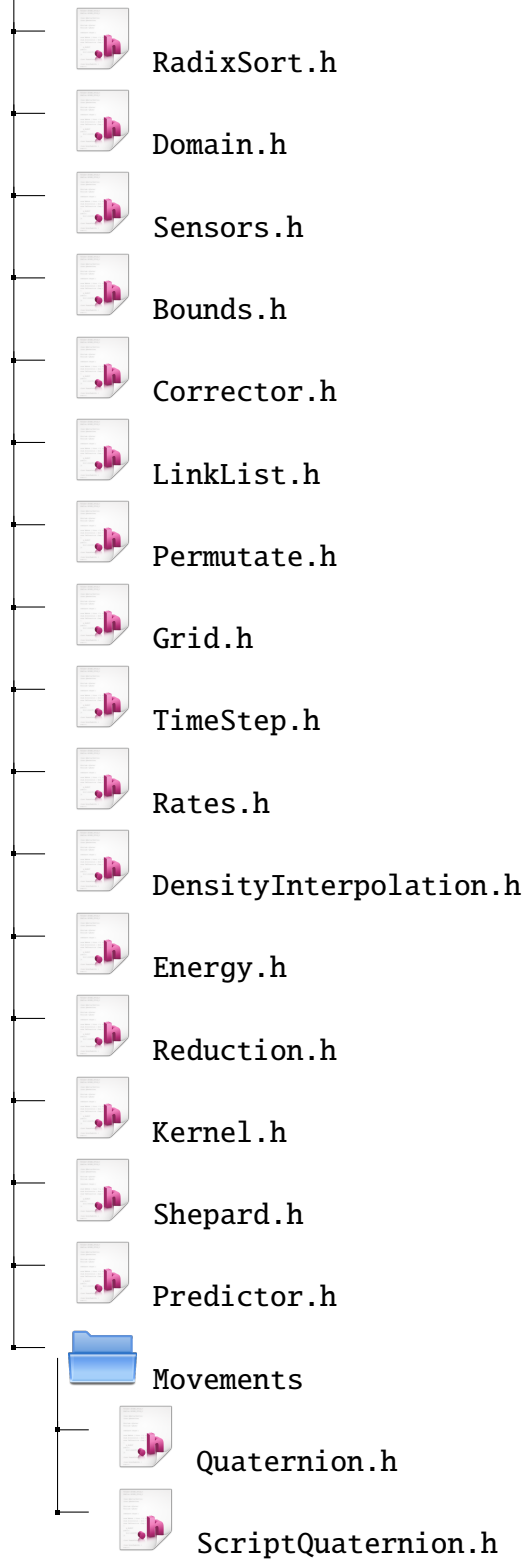


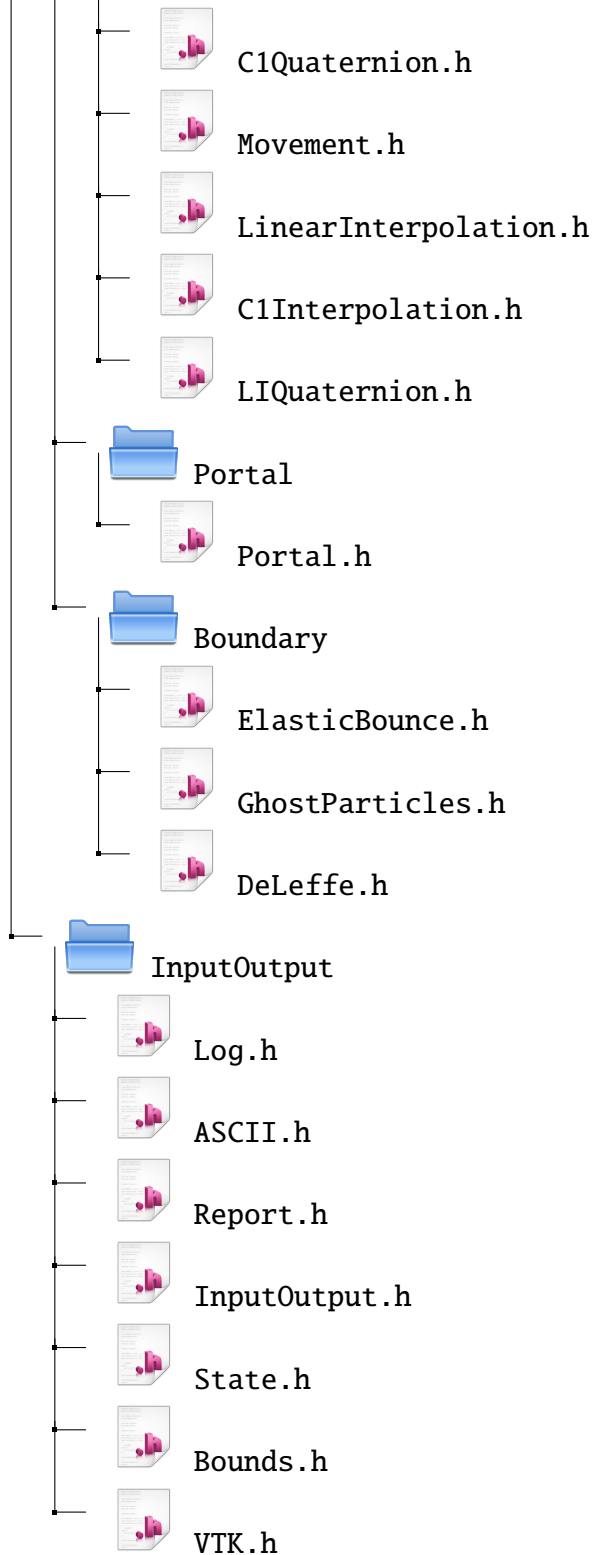


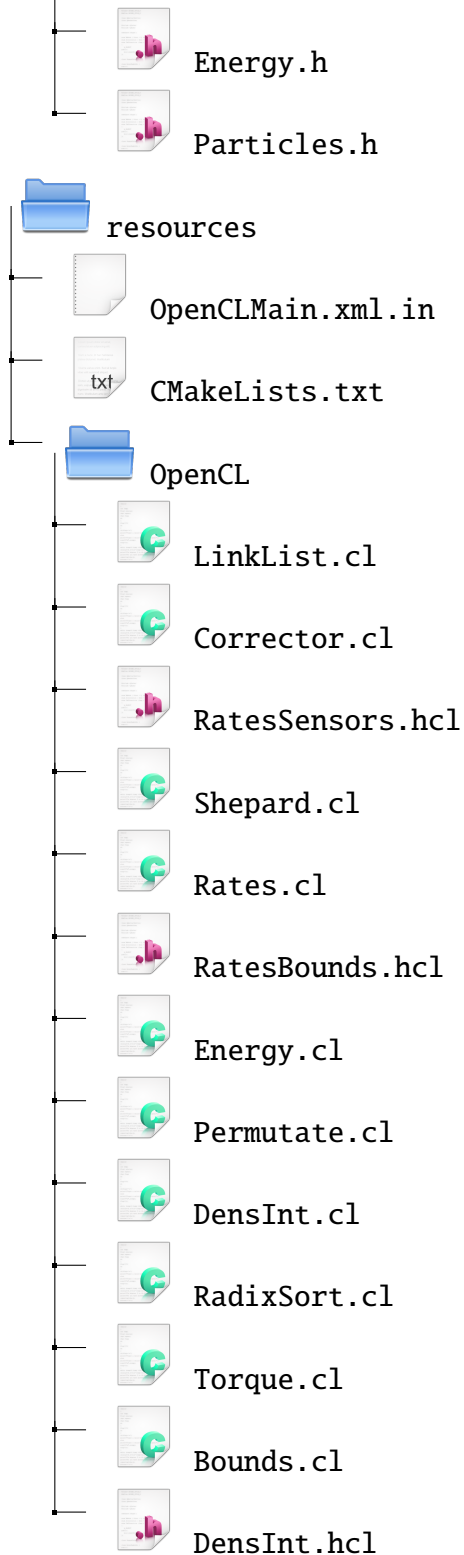
	Fig12_01_ET SIN_Front_Dressler_Martin.6.dat
	Fig30_filtered.2.dat
	Fig12_01_ET SIN_Front_Dressler_Martin.3.dat
	Fig29_filtered.2.dat
	Fig31_filtered.4.dat
	Fig12_01_ET SIN_Front_Dressler_Martin.5.dat
	Fig31_filtered.2.dat
	Fig30_filtered.4.dat
	performance.py
	Fig12_01_ET SIN_Front_Dressler_Martin.2.dat
	Fig29_filtered.4.dat
	Fig12_02_ET SIN_Front_Hu_Koshizuka.5.dat
	sensor2.py
	sensor1.py
	sensor4.py
	Fig30_filtered.1.dat
	Fig29_filtered.1.dat
	Fig12_02_ET SIN_Front_Hu_Koshizuka.4.dat
	sensors.py

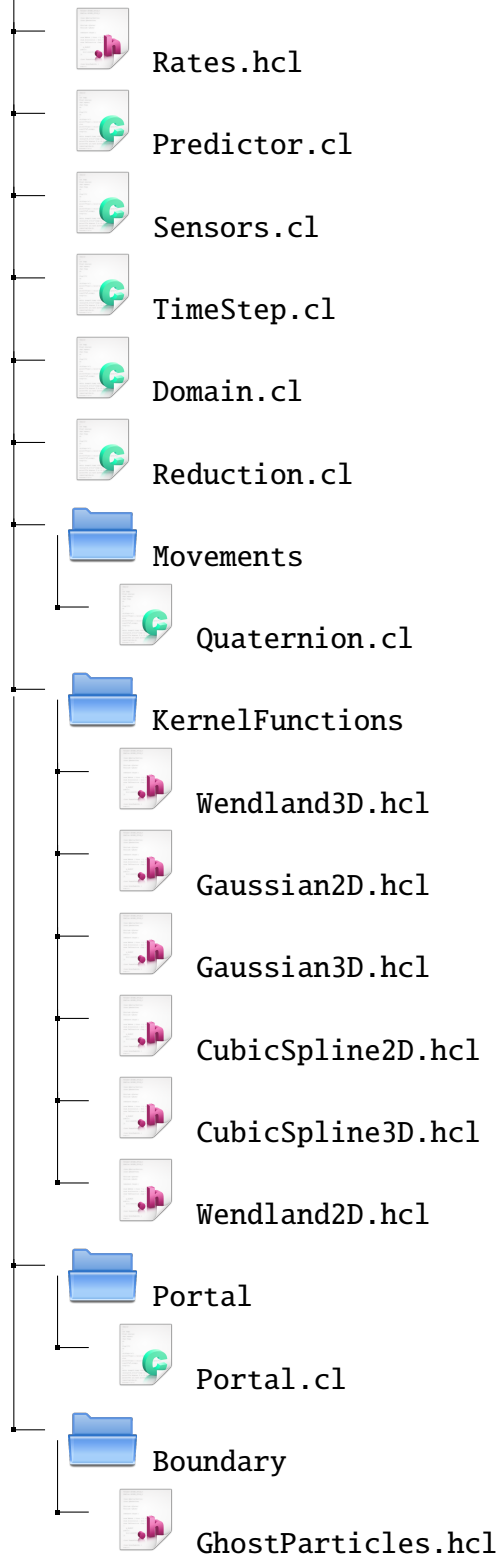


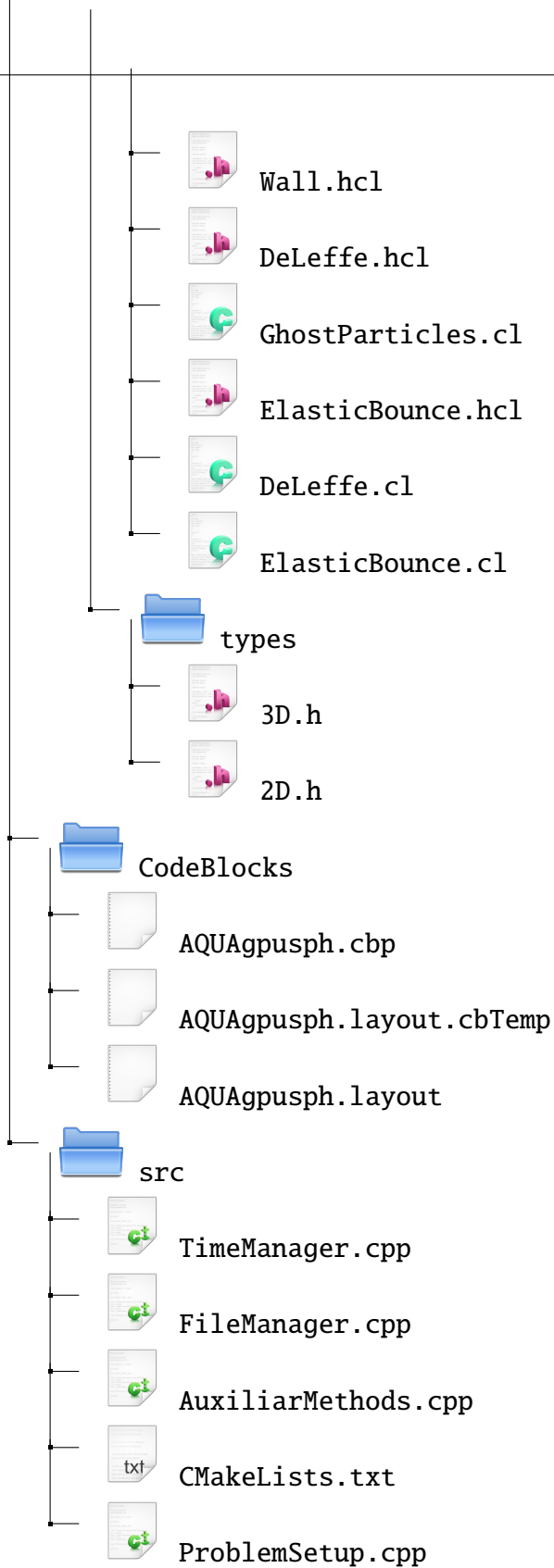


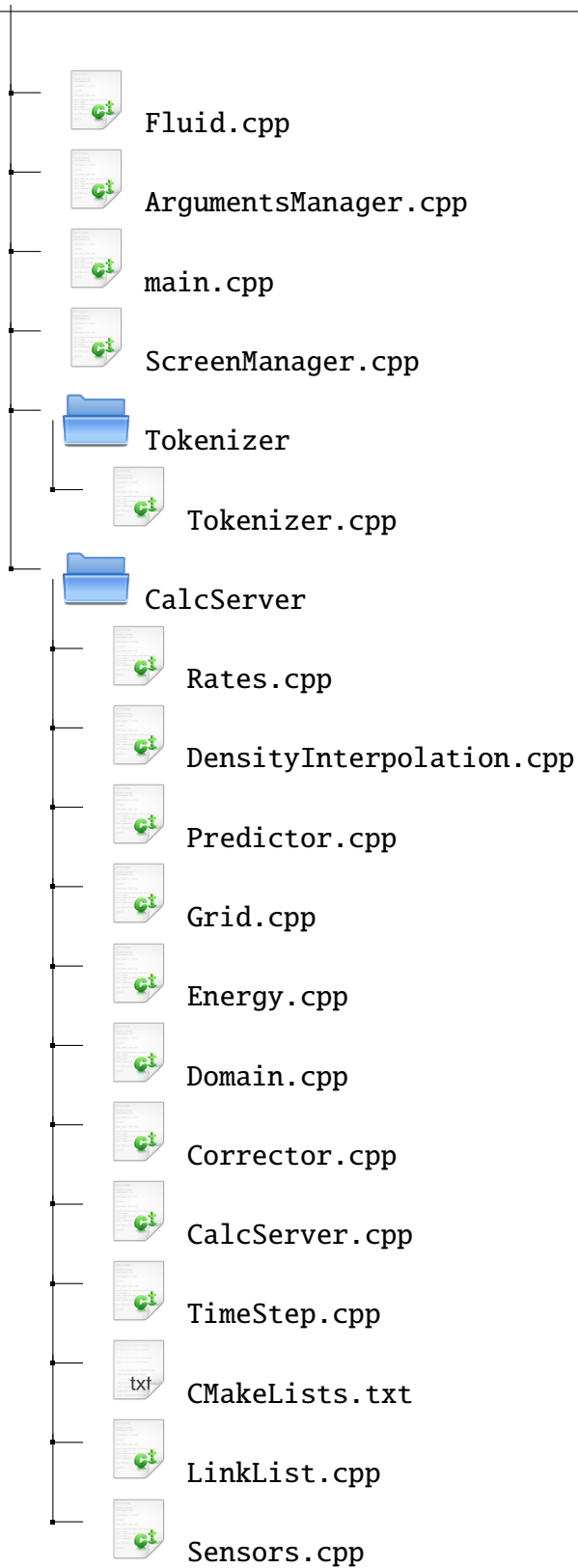


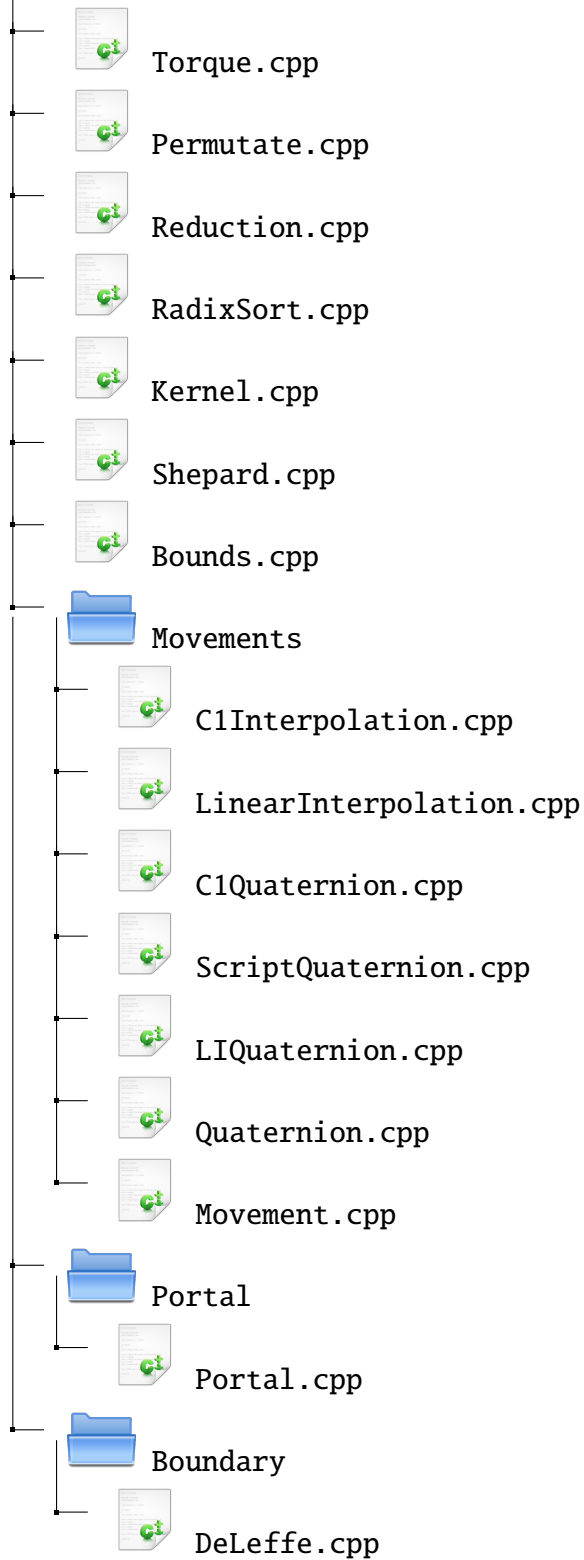


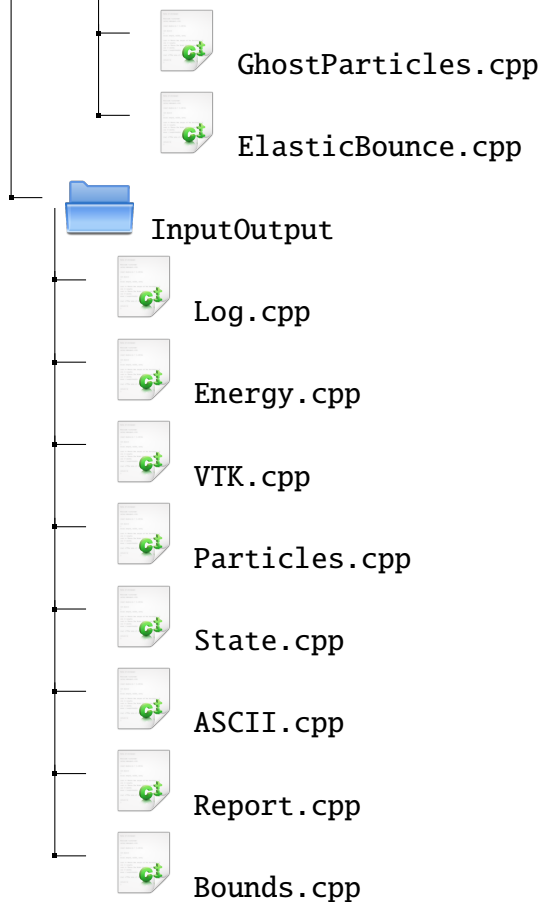












All the files called “CMakeLists.txt”, “*.cmake”, or included in a “cMake” folder are designed to let cmake to configure the package for the build and installation process. More specifically, the files called “CMakeLists.txt” are the ones where the instructions for cmake are written, while the other ones are templates that cmake is conveniently reading and editing.

A CodeBlocks project is provided for the developers in the file “CodeBlocks/AQUAghusph.cbp”. However, it is strongly recommended to use cmake to perform compilations.