Deployment Guide

yedtoss

May 10, 2016

Contents

1 INTRODUCTION

OpenWarp is a web application that can mesh and simulate a body of an offshore structure. It computes the first-order forces (added mass, radiation damping, and diffraction forces) on the body and solves the radiation-diffraction problem of first order.

The OpenWarp software package is composed of three main parts. The first part is the meshing tool. The tool can take as input a raw body in STEP, STL, or IGS format and generate its corresponding shell. The second part of OpenWarp, called Nemoh Solver, is an application that solves the problem of seakeeping in hydrodynamics. It uses the boundary element method to solve for bodies partially immersed (floating) or completely immersed in a fluid of infinite or constant finite depth, with or without forward speed subject to sinusoidal waves. It can be used with the first-order or higher-order panel method. The third part postprocesses the results obtained by Nemoh Solver and generates TECPLOT files of diffraction, radiation, and excitation forces. Added mass and damping coefficients are also generated in TECPLOT format.

OpenWarp operates on an input mesh. To get the input mesh in the correct format, one should use the meshing tool, which can generate a thin or non-thin mesh from an existing body described in STEP, STL, or IGS format. To solve the radiation and diffraction problem, we must represent the body with multiple (thin) quadrilateral elements or shells. OpenWarp includes a web application which can be used to mesh, simulate, and postprocess a body. This web application has a graphical interface that is powered by a server which the user runs locally.

OpenWarp has been developed by the Department of Energy and Topcoder. It is expected to be used by people developing Wave Energy Conversion (WEC) devices in the United States. Until now the testing of new WEC devices has been expensive and time-consuming. OpenWarp will spur innovation by enabling wave-energy startups to develop, analyze, and optimize their devices more quickly.

2 OVERVIEW

Immediately below, we list the notations used in this document and describe the software dependencies for OpenWarp deployment. In subsequent sections, we give detailed instructions for deploying OpenWarp on Windows and on OS X. Finally, we describe how to configure OpenWarp and give a practical usage example

3 NOTATIONS

The following notations are used in this document.

$ROOT : the top-level directory of the Nemoh software package.

$NEMOH_FORTRAN : the directory **$ROOT**/NemohImproved/Nemoh/

$FORTRAN_BUILD : the build directory for the FORTRAN version of Nemoh.

$MINGW_ROOT : the directory where MinGW will be installed.

$INSTALL_DIR is used in the following to mean the root directory containing the files you get when you install the currently available installer for each os.

$NEMOH_PYTHON : the directory **$ROOT**/openwarpgui/nemoh

4 DEPENDENCIES

OpenWarp is compatible with Windows and OS X. Below are the software packages needed to install the application. Section 5 provides step-by-step instructions for Windows installation and section 6 does likewise for OS X.

4.1 Windows

- **Windows 7/8 64 bits** (Other versions might work but not tested)
- MinGW 4.8.1 `http://sourceforge.net/projects/MinGWbuilds/files/host-windows/releases/4.8.1/`
- BLAS `http://icl.cs.utk.edu/lapack-for-windows/lapack/#libraries`
- LAPACK `http://icl.cs.utk.edu/lapack-for-windows/lapack/#libraries`
- OpenMP provided by MinGW
- HDF5 >= 1.8.11 `http://www.hdfgroup.org/HDF5/` Optionally provided by Anaconda
- HDFView `http://www.hdfgroup.org/products/java/release/download.html` Python 2.7 Optionally provided by Anaconda
- H5py >= 2.3.1 Optionally provided by Anaconda
- NumPy Optionally provided by Anaconda
- CMake >= 2.8 `http://www.cmake.org/cmake/resources/software.html`
- Anaconda (with Python 2.7) >= 2.1.0 `http://continuum.io/downloads`
- ParaView >= 4.1 `http://www.paraview.org/download/`

4.2 OS X

- **MAC OS X 10.10 64 bits** (Other versions might work but not tested)
- Homebrew (brew command) `http://brew.sh/`
- GCC >= 4.8 installed by brew
- Xcode Command Line Tools installed by brew

- BLAS provided by XCode commands
- LAPACK Provided by XCode commands
- OpenMP provided by GCC
- HDF5 >= 1.8.11 `http://www.hdfgroup.org/HDF5/` Optionally provided by Anaconda
- HDFView `http://www.hdfgroup.org/products/java/release/download.html`
- Python 2.7 Optionally provided by Anaconda
- H5py >= 2.3.1 Optionally provided by Anaconda
- NumPy Optionally provided by Anaconda
- CMake >= 2.8 installed by brew
- Anaconda (with Python2.7) >= 2.1.0 `http://continuum.io/downloads`
- ParaView >= 4.1 `http://www.paraview.org/download/`

4.3 Linux

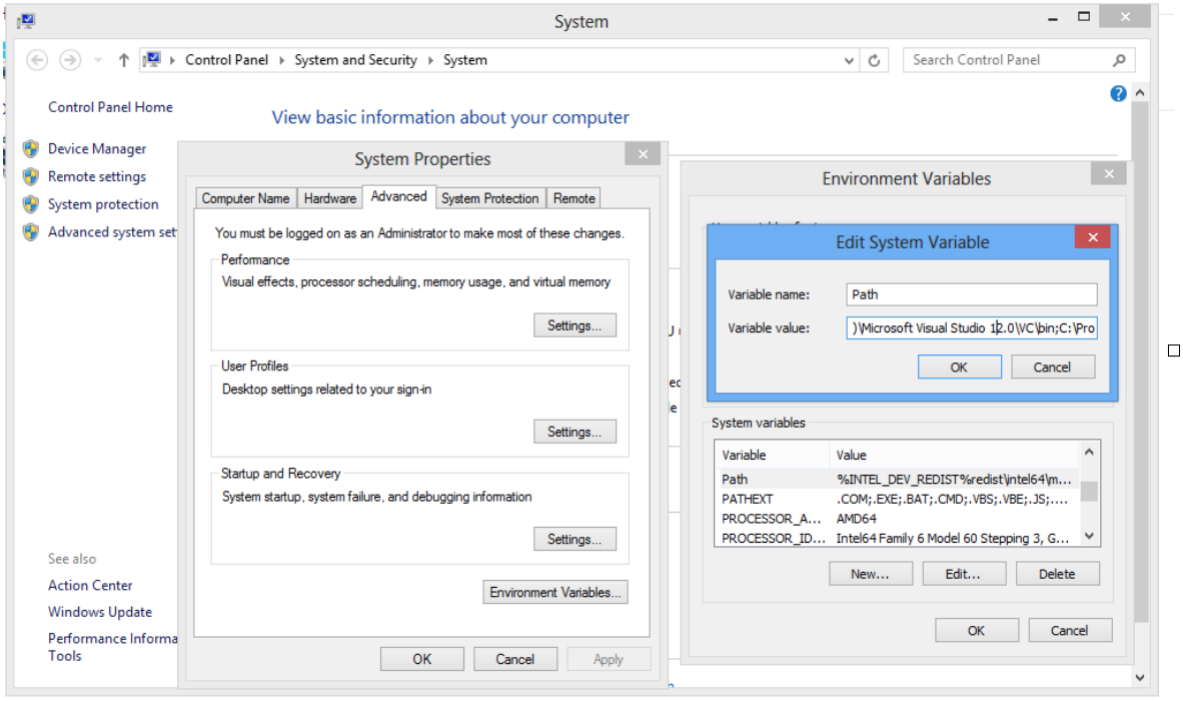
- Linux/Ubuntu 14.04 64 bits (Other versions might work but not tested)
- Except from Homebrew and XCode, Linux shared the same dependencies as MAC OSX as listed in section 4.2.

5 Building from scratch

5.1 WINDOWS DEPLOYMENT FROM SCRATCH

Only 64-bit versions of Windows 7 and Windows 8 are supported. To deploy the application on Windows, perform the following steps:

- Download MinGW (64-bit) from `http://sourceforge.net/projects/mingwbuilds/files/host-windows/releases/4.8.1/64-bit/threads-posix/sjlj/x64-4.8.1-release-posix-sjlj-rev5.7z/download`.
- Extract it to a directory, making sure that the directory name contains no spaces. Let's denote this directory as **$MINGW_ROOT**.
- Add **$MINGW_ROOT**\bin and **$MINGW_ROOT**\lib to the beginning of your Windows PATH by opening the System control panel and editing Properties → Advanced System Settings → Advanced tab → Environment Variables → Path as shown in this screenshot:



- Copy **$ROOT**/src/bundled/simulation/libs/libnemoh.dll, **$ROOT**/src/bundled/simulation/libs/libnemoh.dll.a, **$ROOT**/src/bundled/simulation/libs/libblas.dll, and **$ROOT**/src/bundled/simulation/libs/liblapack.dll to **$MINGW_ROOT**/lib.

- Download and install Anaconda 2.10 with Python 2.7 for Windows (64-bit, graphical installer) from `http://continuum.io/downloads`. If you install it to C:/Users/bob/Anaconda for example, make sure that you manually add C:/Users/bob/Anaconda and C:/Users/bob/Anaconda/Scripts to the beginning of your Windows PATH variable. Don't forget to replace bob with your own user name.

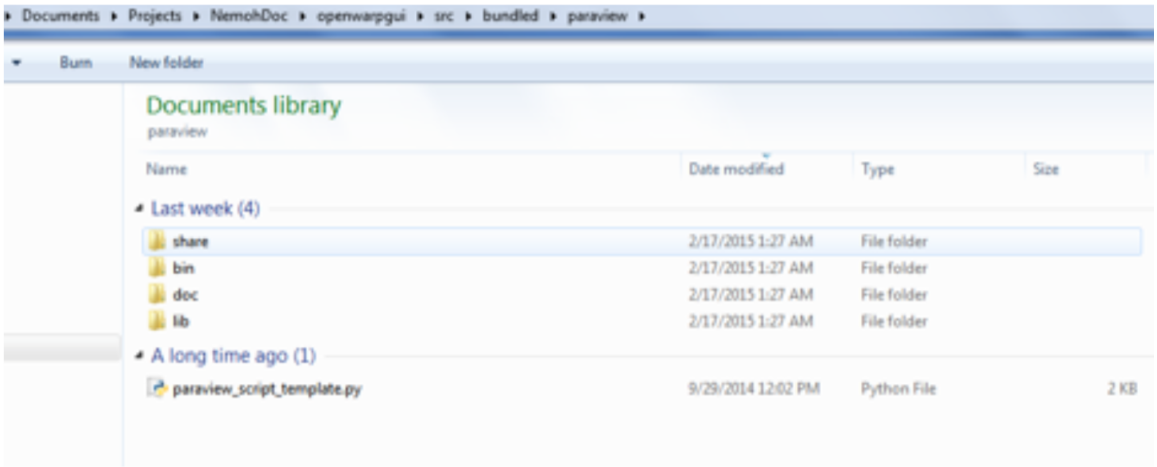
- Once the preceding steps are done, start PowerShell and install the CherryPy dependencies by executing this command in the **$ROOT**/src/ directory: pip install -r requirements.txt

```
PS C:\Users\yedtoss> cd C:\Users\yedtoss\Documents\Projects\NemohDoc\openwarpgui\src
PS C:\Users\yedtoss\Documents\Projects\NemohDoc\openwarpgui\src> pip install -r requirements.txt
Requirement already satisfied (use --upgrade to upgrade): CherryPy in c:\users\yedtoss\anaconda\lib\site-packages (from
-r requirements.txt (line 1))
Requirement already satisfied (use --upgrade to upgrade): numpy in c:\users\yedtoss\anaconda\lib\site-packages (from -r
requirements.txt (line 2))
Requirement already satisfied (use --upgrade to upgrade): cython in c:\users\yedtoss\anaconda\lib\site-packages (from -r
 requirements.txt (line 3))
Requirement already satisfied (use --upgrade to upgrade): h5py in c:\users\yedtoss\anaconda\lib\site-packages (from -r r
equirements.txt (line 4))
Cleaning up...
PS C:\Users\yedtoss\Documents\Projects\NemohDoc\openwarpgui\src>
```

- Start the server by executing this in the src/ directory: python main.py

```
PS C:\Users\yedtoss\Documents\Projects\NemohDoc\openwarpgui\src> python .\main.py
running build_ext
[24/Feb/2015:16:27:33] ENGINE Listening for SIGTERM.
[24/Feb/2015:16:27:33] ENGINE Bus STARTING
[24/Feb/2015:16:27:33] ENGINE Set handler for console events.
[24/Feb/2015:16:27:33] ENGINE Started monitor thread '_TimeoutMonitor'.
[24/Feb/2015:16:27:33] ENGINE Started monitor thread 'Autoreloader'.
[24/Feb/2015:16:27:33] ENGINE Serving on http://127.0.0.1
[24/Feb/2015:16:27:33] ENGINE Bus STARTED
```

- Optionally, download the .zip version of ParaView from `http://www.paraview.org/download/` and copy it to src/bundled/paraview so that src/bundled/paraview/bin exists. This is only needed for testing the visualization of the results in ParaView.



- If the provided libnemoh.dll and libnemoh.dll.a do not work, you can generate them manually by doing the following:
  - Download CMake from `http://www.cmake.org/files/v2.8/cmake-2.8.12.2-win32-x86.exe` , install it and make sure it is in your PATH variable.
  - Start PowerShell and enter an empty directory. Then run:
  - cmake -DCMAKE_Fortran_COMPILER="gfortran" "**$NEMOH_FORTRAN**" -G "MinGW Makefiles" And finally run: mingw32-make

5.2 OS X DEPLOYMENT FROM SCRATCH

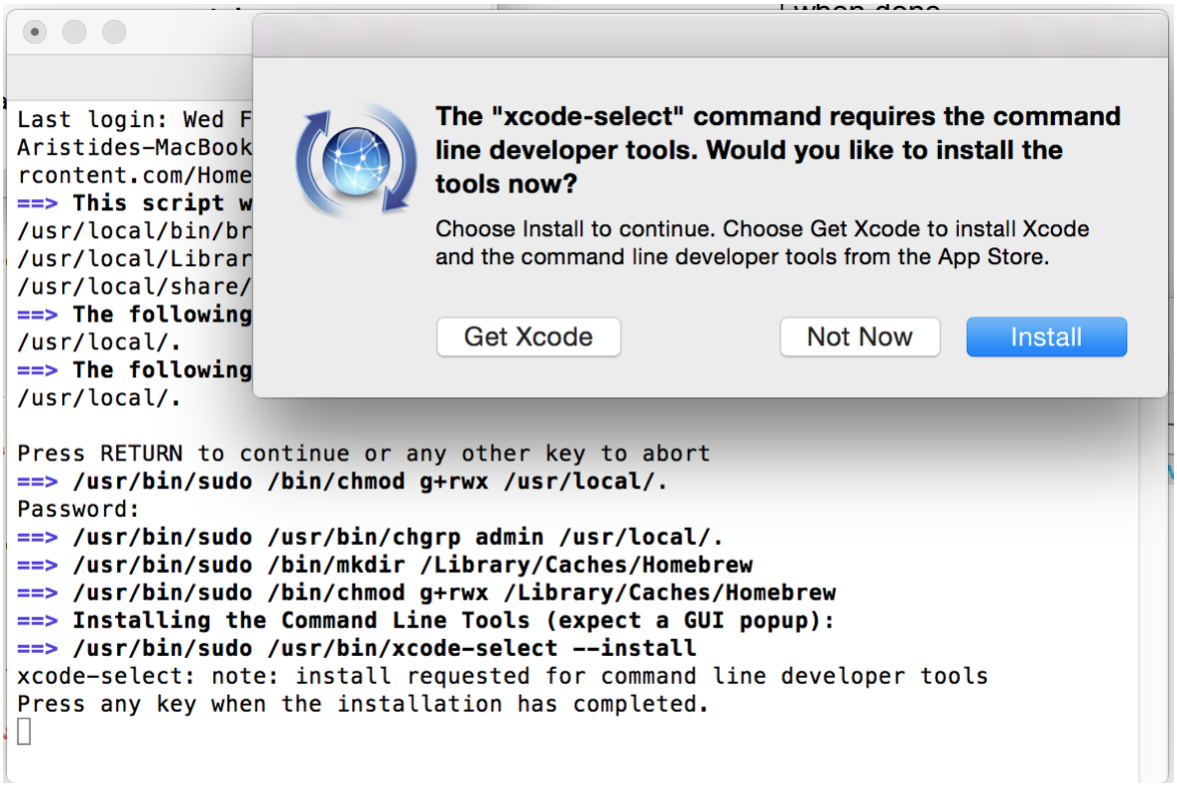
The following instructions are known to work on OS X versions 10.9 and 10.10. To deploy the application in OS X, perform the following step:

- Install brew: ruby -e "$(curl -fsSL `https://raw.githubusercontent.com/Homebrew/install/master/install`)".

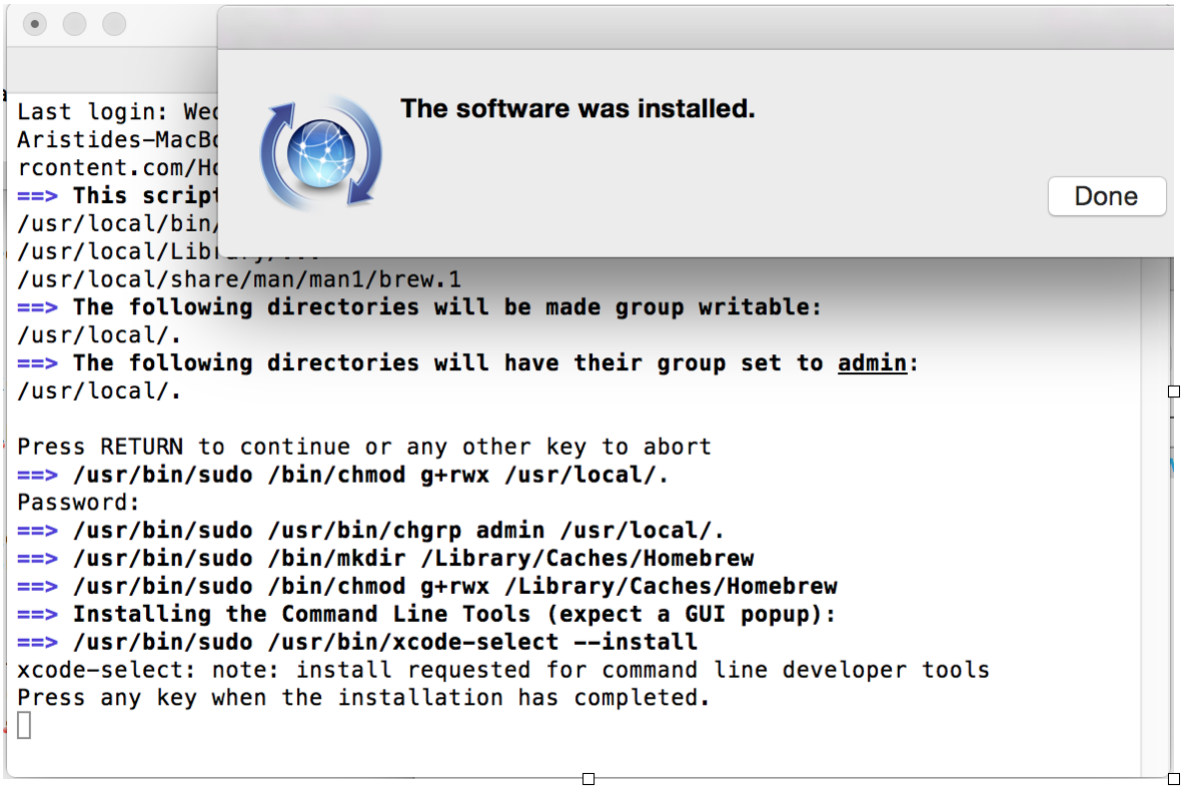
```
Aristides-MacBook-Pro:build yedtoss$ ruby -e "$(curl -fsSL https://raw.githubuse
rcontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/Library/...
/usr/local/share/man/man1/brew.1
==> The following directories will be made group writable:
/usr/local/.
==> The following directories will have their group set to admin:
/usr/local/.

Press RETURN to continue or any other key to abort
```

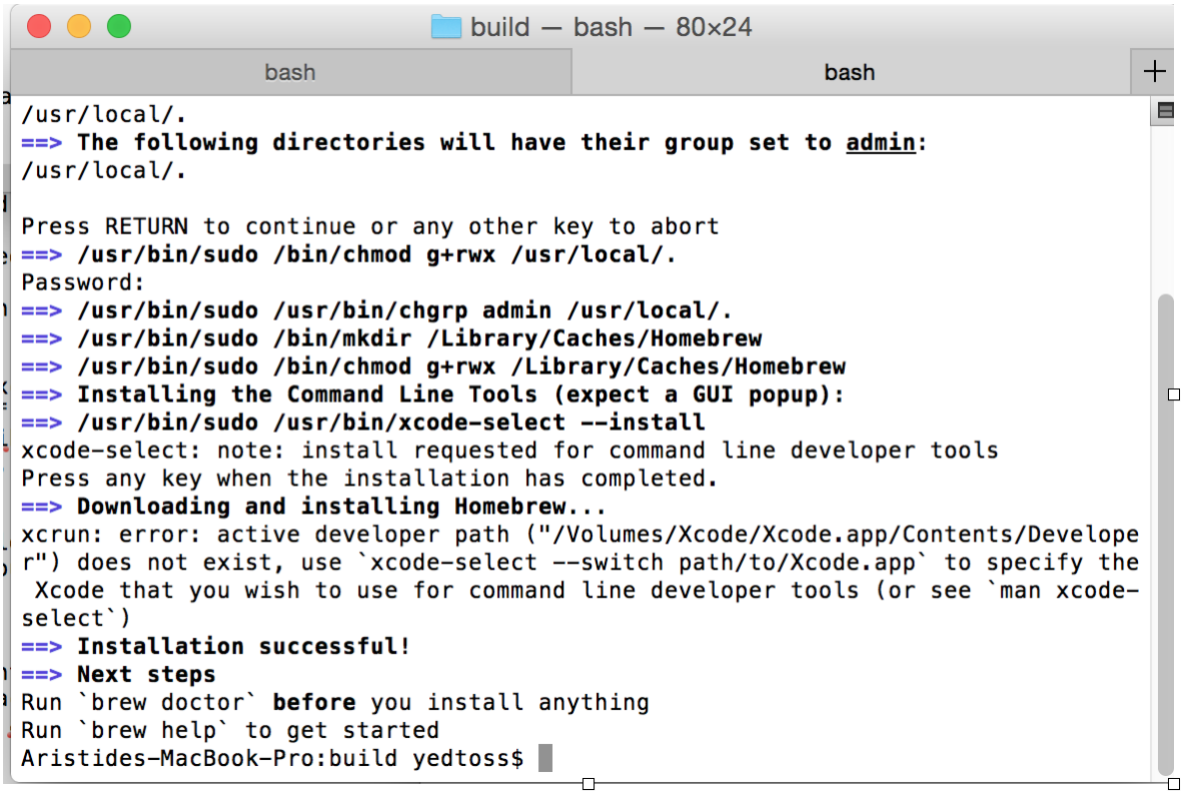
- If you do not have the Xcode command-line tools, you will be prompted to install them as shown below.



- Install the software and make sure the installation is successful as shown below.



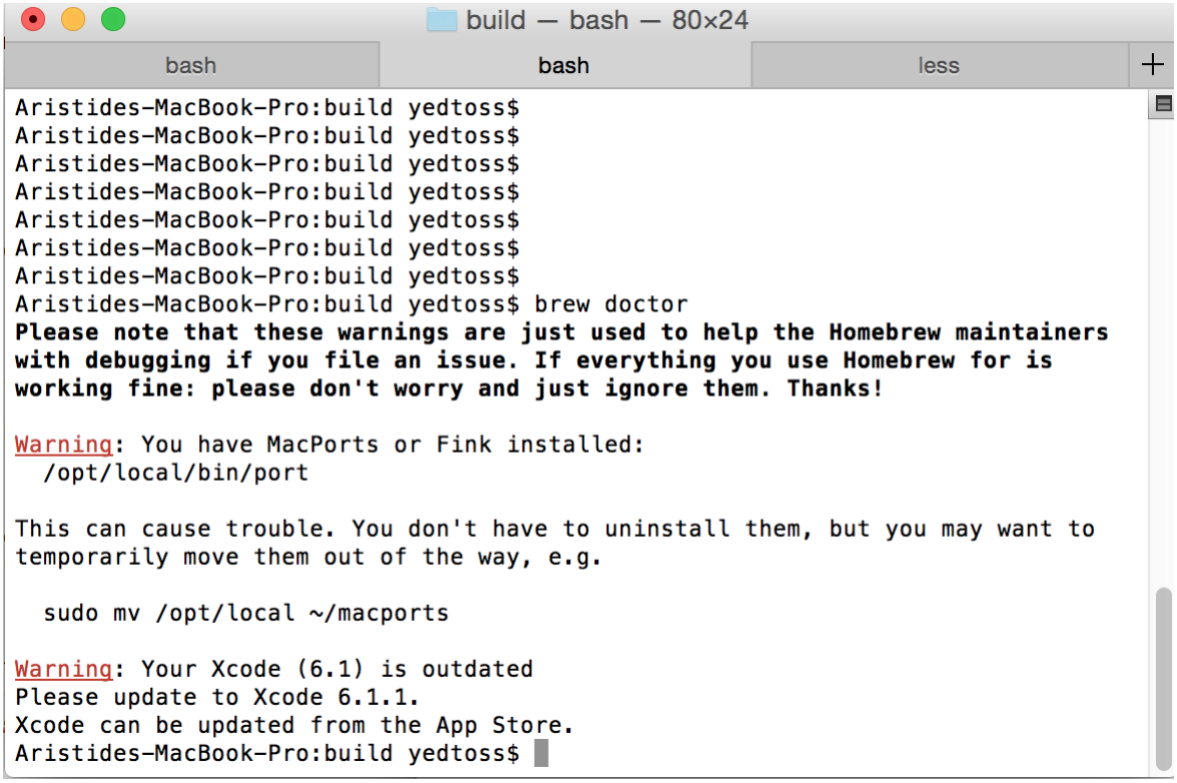
- Press any key. It is possible that you will see an error about the active developer path:



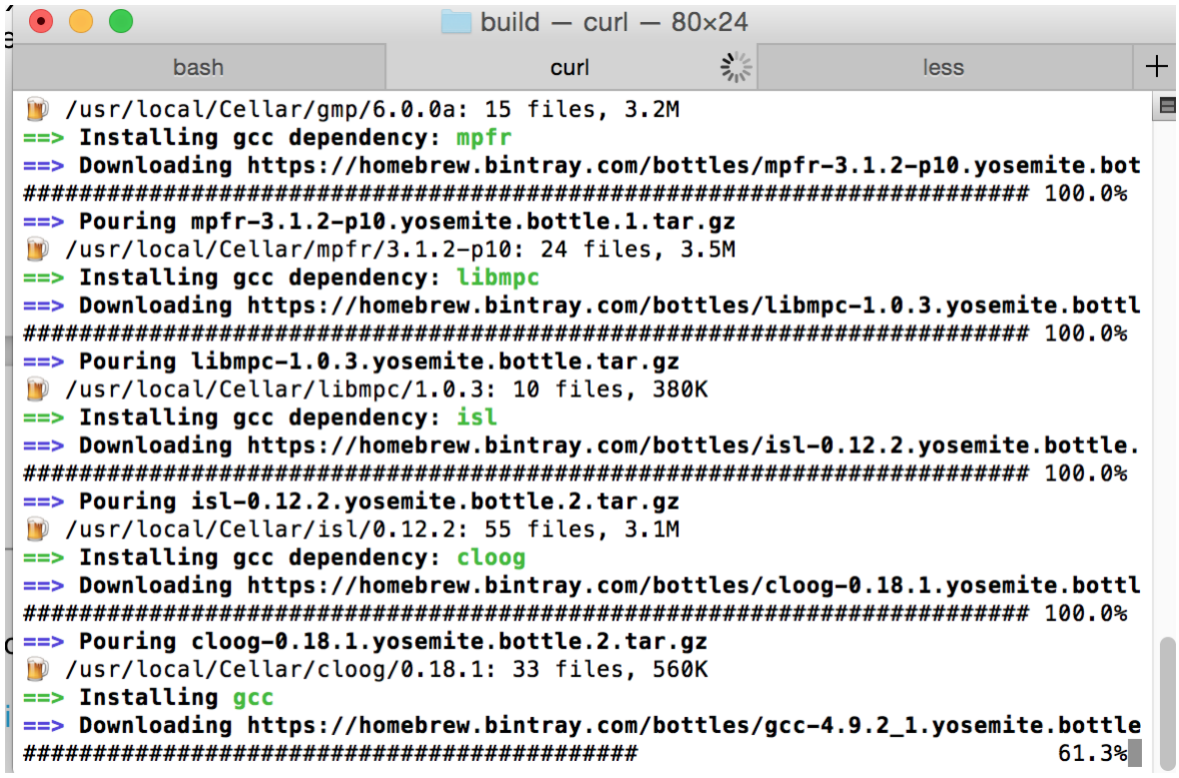
- If you do get that error, you can correct it by specifying the version of Xcode to use: sudo xcode-select –switch /Applications/Xcode.app

```
Aristides-MacBook-Pro:build yedtoss$ sudo xcode-select --switch /Applications/Xcode.app
Password:
Aristides-MacBook-Pro:build yedtoss$
```

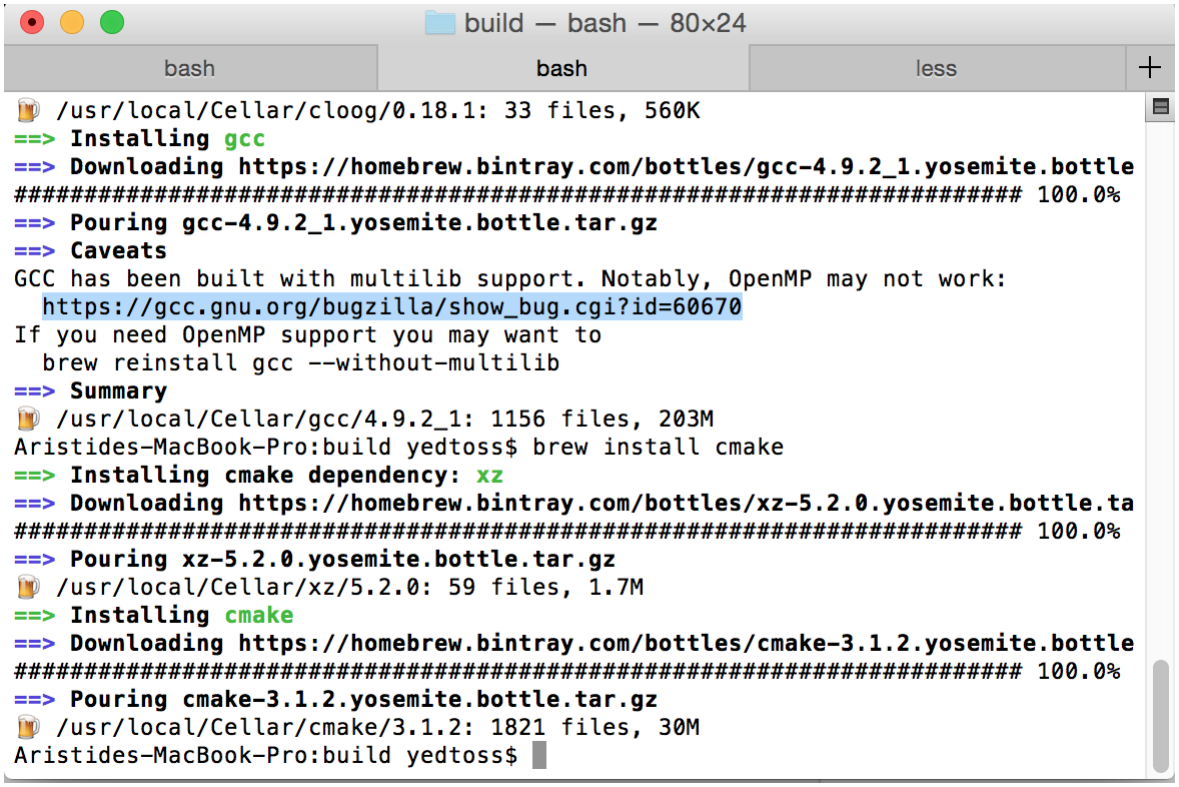
- Now run this command: brew doctor



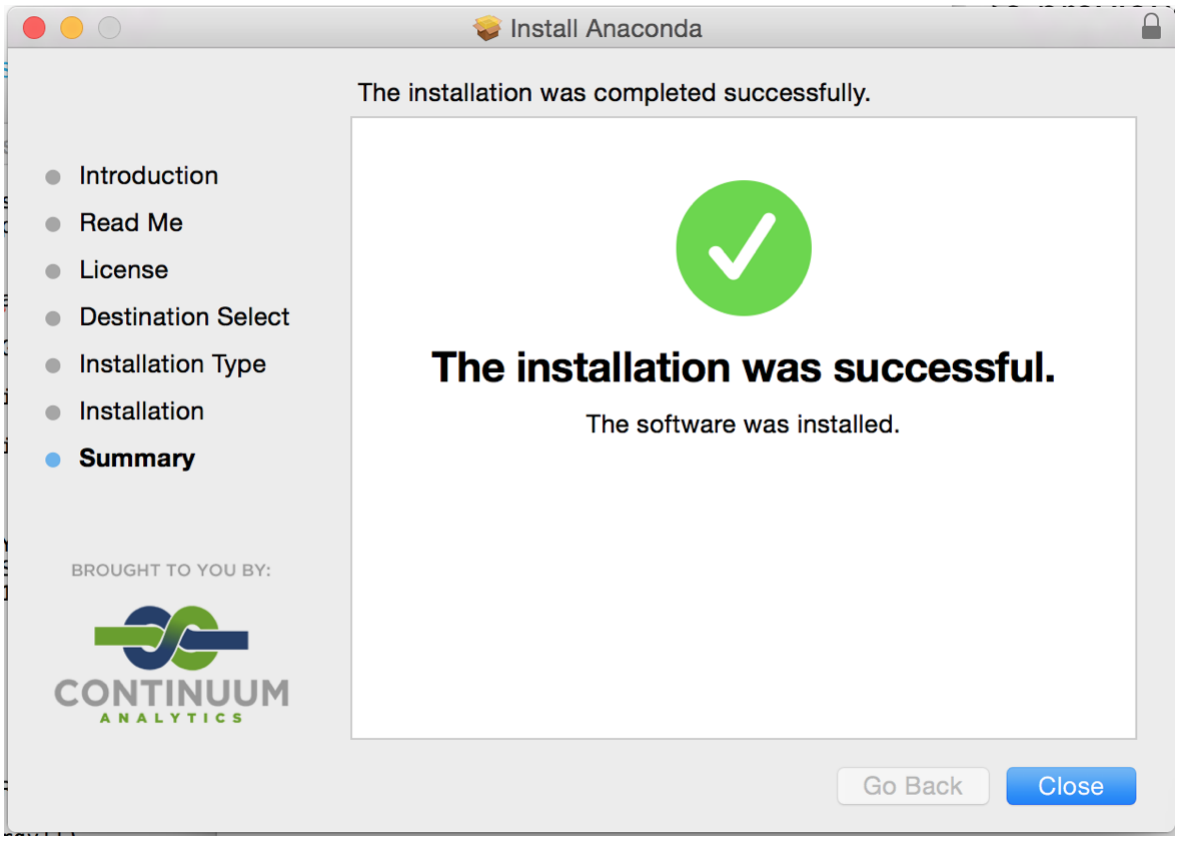
- Ignore any warnings that may appear. Install GCC and GFortran: brew install gcc
  You can ignore warnings about multilib.



- Install CMake: brew install cmake



- Download and install Anaconda for OS X, 64-bit version with Python 2.7 and graphical installer, from `http://continuum.io/downloads`.

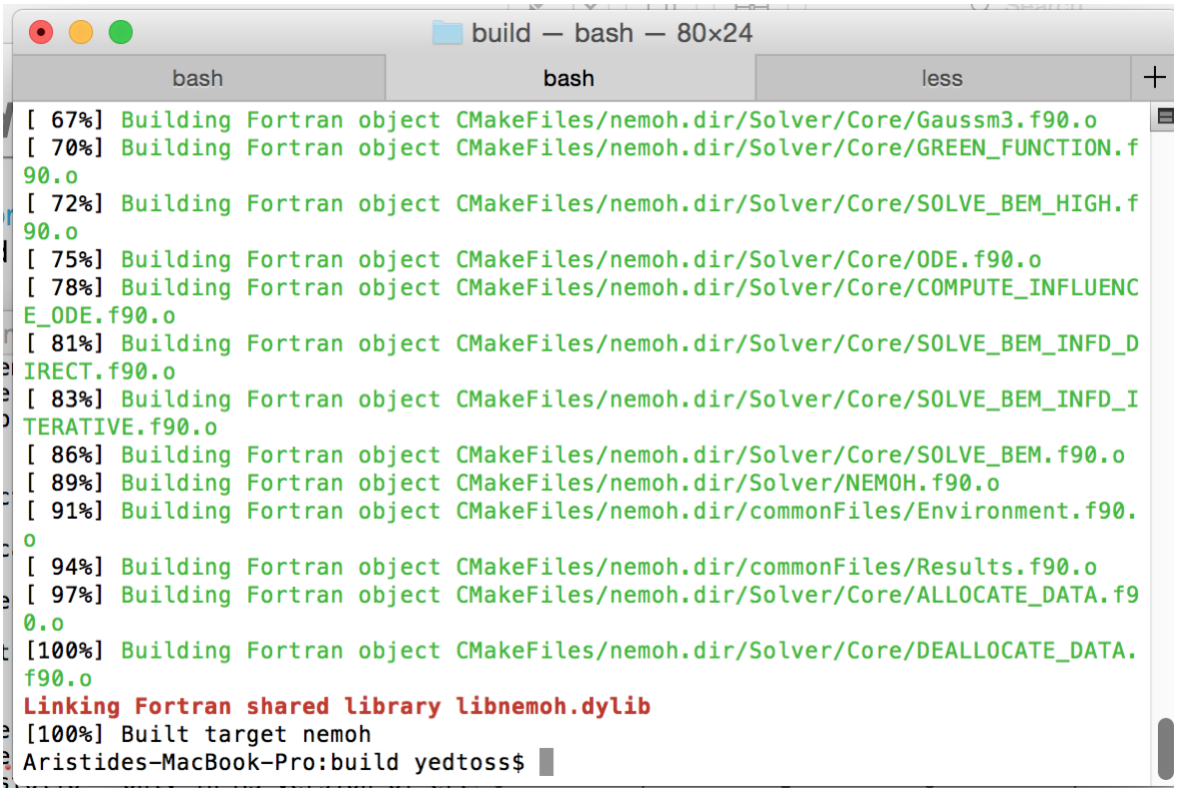


- Now we are going to compile Nemoh Fortran in order to generate the libraries used by the OpenWarp web application. In the following, **$NEMOH_FORTRAN**is the directory NemohMerged/Nemoh/ within the top-level OpenWarp directory.
- Create a new directory different from **$NEMOH_FORTRAN**. Let's call this directory **$FORTRAN_BUILD**.
- Go to **$FORTRAN_BUILD**: cd **$FORTRAN_BUILD**
- This step is optional for a local testing. However if you want to redistribute or create an installer you MUST do it.

  Make sure that the dynamic version of quadmath library is not in your path by running the following command. You may have to adapt 4.9 to the version of gcc installed by brew.

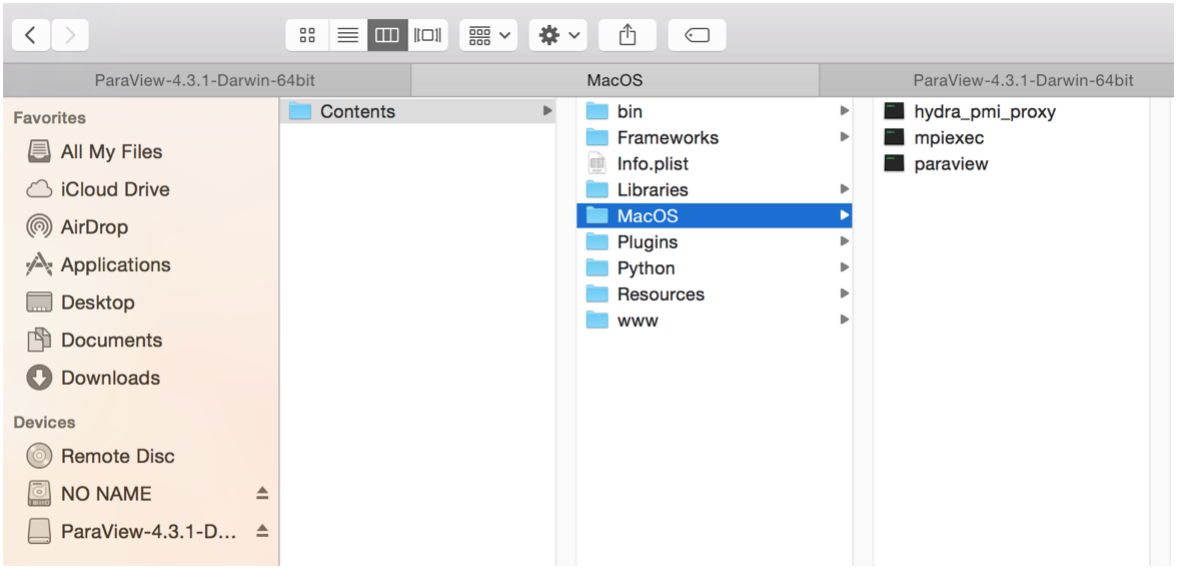
  - (One line command) mv /usr/local/lib/gcc/4.9/libquadmath.0.dylib /usr/local/lib/gcc/4.9/disable_libquadmath.0.dylib
  - (One line command) mv /usr/local/lib/gcc/4.9/libquadmath.dylib /usr/local/lib/gcc/4.9/disable_libquadmath.dylib

- Run this command to compile Nemoh Fortran: cmake -DCMAKE_FortranE_COMPILER="gfortran" **$NEMOH_FORTRAN** If you get a warning about CMake policy, ignore it. Generate the Nemoh Fortran library by running: make This causes libnemoh.dylib to be created.



- Copy libnemoh.dylib from **$FORTRAN_BUILD**to the lib/ directory inside the Anaconda installation root: cp **$FORTRAN_BUILD**/libnemoh.dylib /Users/bob/anaconda/lib (Replace "bob" with your own user name.)

- Download the binary installer of ParaView version >= 4.1 for OS X from `http://www.paraview.org/download/` and install it. Next, copy the paraview.app/ folder from the ParaView directory to **$ROOT**/src/bundled/ so that we can invoke ParaView to do visualization. (You may need to use sudo if you are performing the copy operation on the command line.) The files under **$ROOT**/src/bundled/paraview.app/ should be organized as below:



- We have to start the OpenWarp server before using the web application. To start the server: Make sure you are using Anaconda's version of Python by running: export PATH=/Users/bob/anaconda/bin:$PATH (Replace "bob" with your own user name.) Now run: python -version You should see Anaconda in the output as shown here:

- Run the following command to verify that the library path is correctly set up:

  (test -e /Users/bob/anaconda/lib/libnemoh.dylib && echo 'Success') || echo 'Error: Nemoh library not found'. (As before, replace "bob" with your own user name.)

```
Aristides-MacBook-Pro:src yedtoss$
Aristides-MacBook-Pro:src yedtoss$ export LD_LIBRARY_PATH=/Users/yedtoss/anacond
a/lib
Aristides-MacBook-Pro:src yedtoss$ export LDFLAGS="-L/Users/yedtoss/anaconda/lib
"
Aristides-MacBook-Pro:src yedtoss$ echo $LD_LIBRARY_PATH
/Users/yedtoss/anaconda/lib
Aristides-MacBook-Pro:src yedtoss$ echo $LDFLAGS
-L/Users/yedtoss/anaconda/lib
Aristides-MacBook-Pro:src yedtoss$ (test -e /Users/yedtoss/anaconda/lib/libnemoh
.dylib && echo 'Nemoh is successfully found' ) || echo 'Error Nemoh Library not
found'
Nemoh is successfully found
Aristides-MacBook-Pro:src yedtoss$
```

  If you get an error message here, it means you did not correctly set up the path to the Nemoh library.

- Make sure the nglib-mesh binary is runnable: chmod +x **$ROOT**/src/bundled/mesh-generator/build/nglib-mesh (Recall that **$ROOT**is the top-level directory of the OpenWarp package.)

- To prevent dylib errors, go to the directory **$ROOT**/src/bundled/mesh-generator/lib/ and run:

  python update_dylib.py

```
Aristides-MacBook-Pro:openwarpgui yedtoss$ chmod +x src/bundled/mesh-generator/b
uild/nglib-mesh
Aristides-MacBook-Pro:openwarpgui yedtoss$ cd src/bundled/mesh-generator/lib
Aristides-MacBook-Pro:lib yedtoss$ python update_dylib.py
Aristides-MacBook-Pro:lib yedtoss$
```

- Install the remaining Python dependencies by going to **$ROOT**/src/ and running: pip install -r requirements.txt This step is required before running the OpenWarp server for the first time. You can skip it when running the server on subsequent occasions.

```
Aristides-MacBook-Pro:src yedtoss$ pip install -r requirements.txt
Downloading/unpacking CherryPy (from -r requirements.txt (line 1))
  Downloading CherryPy-3.6.0.tar.gz (432kB): 432kB downloaded
  Running setup.py (path:/private/var/folders/81/0r8qm66x40d88d523jj7t6c80000gn/
T/pip_build_yedtoss/CherryPy/setup.py) egg_info for package CherryPy

Requirement already satisfied (use --upgrade to upgrade): numpy in /Users/yedtos
s/anaconda/lib/python2.7/site-packages (from -r requirements.txt (line 2))
Requirement already satisfied (use --upgrade to upgrade): cython in /Users/yedto
ss/anaconda/lib/python2.7/site-packages (from -r requirements.txt (line 3))
Requirement already satisfied (use --upgrade to upgrade): h5py in /Users/yedtoss
/anaconda/lib/python2.7/site-packages (from -r requirements.txt (line 4))
Installing collected packages: CherryPy
  Running setup.py install for CherryPy
    changing mode of build/scripts-2.7/cherryd from 644 to 755

    changing mode of /Users/yedtoss/anaconda/bin/cherryd to 755
Successfully installed CherryPy
Cleaning up...
Aristides-MacBook-Pro:src yedtoss$
```

- To start the server, run: sudo python main.py If you get an error, make sure that port 80 is not being used by another process.

```
Aristides-MacBook-Pro:openwarpgui yedtoss$ sudo python main.py
Password:
Found logging configuration file at /Applications/OpenWarp/openwarpgui/nemoh/logging.json

Writing log at /Users/yedtoss/OpenWarpFiles/logs/logs.log

running build_ext
```

  If you get an error message reading "library not found for -lnemoh" or "No module named solver_fortran", make sure you have correctly exported the path to Nemoh in the current terminal session as explained earlier.

5.3 Linux deployment from scratch

It is enough to run the script install.sh inside **$ROOT** while being connected to the Internet. Note that you are supposed to use that script from the pre-built package of Linux as explained in 6.3; otherwise you should download and install Paraview yourself before running the script. Only tested on Ubuntu 14.04 64 bits.

6 Installing the pre built package

Get the OpenWarp Installer from `https://drive.google.com/folderview?id=0B56WkDEeFyNHfmRZNmcyZjNWOTFfTzZabGgxZVdsQUg3Y3d6SXloWk8xMXo0bXhnSUVTOGM&usp=sharing`

The windows one is in the Windows directory. The MAC one is in the MAC directory. Note that those two installers are outdated. The Linux one is in the Linux directory.

6.1 OS X

- Download the MAC OS X packages and right-click on it. Then select Open in the contextual menu to install it.
- To run the OpenWarp GUI later, go to the directory where you install it and double-click "run" or "run.sh". See section 9 of this document for more information about using the installed application. NB: If you get a cookie error open the url (http://127.0.0.0.1:8386/index.html) in private navigation

6.2 Windows

- Download the Windows binary of OpenWarp and double-click on it to start the installation procedure.
- To run the OpenWarp GUI later go to the directory where you install it and double-click "run.bat". See section 9 of this document for more information about using the installed application.
- NB: OpenWarp GUI is known not to work with Internet Explorer. So you must use Chrome or Firefox with the url http://127.0.0.0.1:8386/index.html. If you get a cookie error open the url in private navigation

6.3 Linux

- The Linux installation is a little different from other OS. You need to make sure you are connected to Internet the first time you want to install it at least.
- Extract the OpenWarp GUI in a folder and in a terminal run "bash installer.sh". (You can double-click to install if you extracted it to an ext4 file system). You will be asked for your sudo password. Make sure to enter it.

- To run the installer later on, enter the folder where you had extracted the OpenWarp and run "bash run.sh". See section 9 of this document for more information about using the installed application. NB: If you get a cookie error open the url (http://127.0.0.0.1:8386/index.html) in private navigation This is known to happen in some configuration of Firefox

7 Deployment from existing package

Instead of building from scratch, you can first install the old installer using section 6
Then follow these steps:

7.1 Windows

7.1.1 Recompiling NemohFortran and NemohPython

- You need to make sure to replace all new codes in **$INSTALL_DIR**/OpenWarp by the one in the submission (the new one.)
- Follow instructions of section 5.1 to build a local copy. Replace **$NEMOH_FORTRAN**by **$INSTALL_DIR**/OpenWarp/NemohImproved/Nemoh;

  $ROOT**/src should be replaced by **$INSTALL_DIR/OpenWarp/openwargui The python version of nemoh should be **$INSTALL_DIR**/OpenWarp/openwargui/nemoh Anaconda should be **$INSTALL_DIR**/OpenWarp/Anaconda
- Now copy the newly generated libnemoh.dll and libnemoh.dll.a to **$INSTALL_DIR**/OpenWarp/Anaconda/DLLs.
- If your code needs any additional DLLS make sure to put them there

7.1.2 Repackaging the installer

- Copy the directory src/Windows/installer to **$INSTALL_DIR**such that **$INSTALL_DIR**/OpenWarp and **$INSTALL_DIR**/installer exists.
- Download and Install Inno Setup version equal or greater than 5.5.5 from `http://www.jrsoftware.org/isdl.php`
- Open **$INSTALL_DIR**/installer/openwarp.iss with Inno Setup (You may double-click on the file)
- Inno Setup will open. Click on Build –> Compile (menu). Once done the installer will be in **$INSTALL_DIR**/OpenWarp/installer/Output/OpenWarp.exe

7.2 OS X

7.2.1 Recompiling NemohFortran and NemohPython

- You need to make sure to replace all new codes in **$INSTALL_DIR**/OpenWarp by the one in the submission (the new one.)
- Follow instructions of section 5.2 to build a local copy. Replace **$NEMOH_FORTRAN**by **$INSTALL_DIR**/OpenWarp/NemohImproved/Nemoh;

  $ROOT**/src should be replaced by **$INSTALL_DIR/OpenWarp/openwargui The python version of nemoh should be **$INSTALL_DIR**/OpenWarp/openwargui/nemoh Anaconda should be **$INSTALL_DIR**/OpenWarp/Anaconda
- Now copy the newly generated libnemoh.dylib to **$INSTALL_DIR**/OpenWarp/anaconda/lib
- Then VERY IMPORTANTLY run in a terminal install_name_tool -change "libnemoh.dylib" "../anaconda/lib/libnemoh.dylib" **$INSTALL_DIR**/OpenWarp/openwarpgui/nemoh/solver_fortran.so from **$INSTALL_DIR**/OpenWarp/openwargui/nemoh

7.2.2 Packaging the installer

- Copy the directory src/MAC/installer to **$INSTALL_DIR**such that **$INSTALL_DIR**/OpenWarp and$INSTALLDIR/installer exists.
- Download and install Packages (version >= 1.1.1) from `http://www.macupdate.com/app/mac/34613/packages`
- Open **$INSTALL_DIR**/OpenWarp/installer/OpenWarp with Packages (You can double-click on it)
- In the menu click on Build –> Build or Type Command +B
- The installer will start building and once done the MAC OS X packages will be in **$INSTALL_DIR**/OpenWarp/installer/build/OpenWarp.pkg

7.3 Linux

Here the file installer.sh is taking care of everything. In most cases you don't have to do anything extras in case the code is updated. Read section **??** to see the cases where you might need to update this file. In any case just make sure installer.sh and run.sh are accurate and can be used to install and run the new code in a clean system.

7.3.1 Recompiling NemohFortran and NemohPython

Not needed as the installer.sh take care of that automatically.

7.3.2 Packaging the installer

You just need to compress the directory **$INSTALL_DIR**.

7.4 General Guide about the installer build/packaging procedure

OpenWarp is composed of three main parts: the Fortran part of Nemoh whose source is available in **$INSTALL_DIR**/OpenWarp/NemohImproved/Nemoh; the Python part of Nemoh whose source is available in **$INSTALL_DIR**/OpenWarp/openwarpgui/nemoh and the OpenWarp GUI part whose source is available in **$INSTALL_DIR**/OpenWarp/openwarpgui/openwarp. If you do not update the Fortran part of Nemoh or the file **$INSTALL_DIR**/OpenWarp/openwarpgui/nemoh/solver_fortran.pyx then the only step you have to do to rebuild the installer is to package it directly. There is no need to do any re-compilation steps. When you update the Fortran or **$INSTALL_DIR**/OpenWarp/openwarpgui/nemoh/solver_fortran.pyx you need to recompile them and copy the generated library in a well-defined location. See the relevant OS guide for more information.

In case you added some python dependencies not currently available, you would need to package them in the installer. For Linux you can update the relevant **$INSTALL_DIR**/OpenWarp/openwarpgui /requirements.txt and it will be installed automatically by **$INSTALL_DIR**/OpenWarp/installer.sh

For Windows and MAC OS X you would need to install those dependencies in the provided Anaconda site-packages available in **$INSTALL_DIR**/OpenWarp/Anaconda and **$INSTALL_DIR**/OpenWarp/anaconda respectively. If you add some external libraries as dependencies you need to make sure they are available to end users. In Windows you can just copy them to **$INSTALL_DIR**/OpenWarp/Anaconda/DLLs.

In MAC OS X you can just copy them to **$INSTALL_DIR**/OpenWarp/anaconda/lib. Make sure there are static or update their dependencies using install_name_tool (You can see an example in the MAC OS X specific instructions). For Linux/Ubuntu, it is preferred to use apt-get install to make those external libraries available to end users.

8 Configuration

The three parts of the application can all be configured using the web application GUI. In this section, we describe the configurable parameters for each part.

8.1 Meshing parameters

Input File : This is the input file geometry for which we want to generate quadrilateral elements. The accepted input file formats are IGS, STEP, and STL.

Output File : Enter the name of the output file without any file extension. This should be a bare file name without a directory path preceding it.

Fineness : The target fineness of the mesh.

MaxH : The maximum height of the quadrilateral elements in the mesh.

MinH : The minimum height of the quadrilateral elements in the mesh.

Grading : The grading of the mesh.

Use Tolerance ?: Whether or not to use tolerance when generating the mesh. When the absolute value of the difference between two values is less than the tolerance, we consider them to be equal.

Tolerance : Volume tolerance expressed as a number from 0 to 1.

8.2 Simulation parameters

Rho : The fluid's specific volume in KG/m^3.

G : Gravity in m/s^2.

Depth : The depth of the water in meters. Enter 0 to indicate infinite depth.

XEFF, YEFF : The x and y coordinates, in meters, of the point where the wave is measured.

Name of Mesh File : The name of the mesh file generated in the meshing step.

Number of Points and Number of Panels : The number of points in the mesh and the total number of panel or quadrilaterals in the mesh. These values are automatically filled in.

Surge: The freedom of translation along the X axis (moving forward and backward).

Sway : The freedom of translation along the Y axis (moving left and right).

Heave : The freedom of translation along the Z axis (moving up and down).

Roll About A Point : The freedom of rotation along the X axis (pivoting side to side).

Pitch About A Point : The freedom of rotation along the Y axis (tilting forward and backward).

Yaw About A Point : The freedom of rotation along the Z axis (swiveling left and right).

Force In X, Y, Z Direction : Three-dimensional coordinates of force.

Moment Force In X, Y, Z Direction About A Point : Three-dimensional coordinates of moment force.

Number of Lines of Additional Information Currently not supported. Should be set to 0.

Number of Wave Frequencies, MIN, MAX : Respectively the number of wave frequencies, the minimum value of each wave frequency, and the maximum value of each wave frequency.

Number of Wave Directions, MIN, MAX : Respectively the number of wave directions, the minimum value of each wave direction, and the maximum value of each wave direction.

Indiq_solver : The method to use for solving linear equation. Use 0 to indicate the Gauss Method, 1 to indicate the GMRES method, and 2 to indicate the GMRES method with FMM acceleration. Option 2 is not yet supported.

TOL_GMRES : The value of the tolerance to use for the GMRES method. The tolerance is used to determine convergence. IRES : Restart parameter for GMRES. MAXIT: Maximum iterations for GMRES.

Sav_potential : Use 0 or 1 to indicate whether or not the potential should be saved in the output.

GREEN_TABULATION_NUMX : Represents the number of points in the X direction in the tabulated data.

GREEN_TABULATION_NUMZ : Represents the number of points in the Z direction in the tabulated data.

GREEN_TABULATION_SIMPSON_NPOINTS : Represents the number of subintervals used to approximate the Green's function integral using Simpson's rule.

USE_ODE_INFLUENCE_COEFFICIENTS Indicate whether or not to use the ODE method to compute the influence coefficients.

USE_HIGHER_ORDER : Whether or not to use the higher-order panel method.

NUM_PANEL_HIGHER_ORDER : The number of panels per patch in the higher-order method.

B_SPLINE_ORDER : The order of the B-spline for the potential in the higher-order panel method.

USE_DIPOLES_IMPLEMENTATION : Whether or not to use the dipole implementation.

THIN_PANELS : A list containing the indices of panels which are thin dipoles. Indices are zero-based. Set the list to [-1] to indicate that all panels are thin dipoles.

COMPUTE_DRIFT_FORCES : Whether or not to compute the drift forces.

COMPUTE_YAW_MOMENT : Whether or not to compute the yaw moment.

8.3 Postprocessing parameters

IRF : Whether or not to compute the IRF, i.e., the infinite frequency added mass and the impulse response function for the radiation force.

TIME Step : The time step used for the IRF computation.

Duration : The duration used for the IRF computation.

Show Pressure : Whether or not to output the pressure forces.

Kochin Function : The number of angles to use in computing the Kochin function. Set 0 to disable this computation.

Min Angle, Max Angle : The minimum and maximum angle to use in computing the Kochin function.

Number of Points In X, Y Direction : Number of points in each direction for the free-surface visualization. Set 0 to disable the computation.

Dimensions of Domain In X, Y Direction : The domain dimension in each direction, in degrees.

8.4 Other and Logging parameters

There are only two parameters currently for the Configuration screen and they all support logging. We have:

Logging level : Pick "DEBUG" or "INFO" to set the logging level to the respective value

Clear old logs : Whether or not to delete the old log files

The logging level can also be configured using the configuration file **$NEMOH_PYTHON**/logging.json

When run from the web interface the logs are located by default in the user home in OpenWarpFiles/logs

When run from the command line the logs are located in the configured base test directory in **$NEMOH_PYTHON**/settings.py

9 Usage

9.1 Start the application

Make sure the server has been started: python main.py This command must be executed from the **$ROOT**/source/openwarpgui directory Open a web browser and go to: `http://127.0.0.1/index.html` You will see the following.

- To launch the application on Windows, go to where you have installed it (by default C:/Program Files (x86)/OpenWarp) and double-click "run.bat"

  You can then access the GUI using the url `http://127.0.0.0.1:8386/index.html`

  Don't use Internet Explorer as it is known not to work with the GUI.

- To launch the application on Linux, go to where you have installed it (The directory where you extracted the archive) from a terminal and run "bash run.sh" You can then access the GUI using the url http://127.0.0.0.1:8386/index.html

- To launch the application on MAC OS X, go to where you have installed it (by default /Applications/OpenWarp) and double-click on "run" file You can then access the GUI using the url http://127.0.0.0.1:8386/index.html

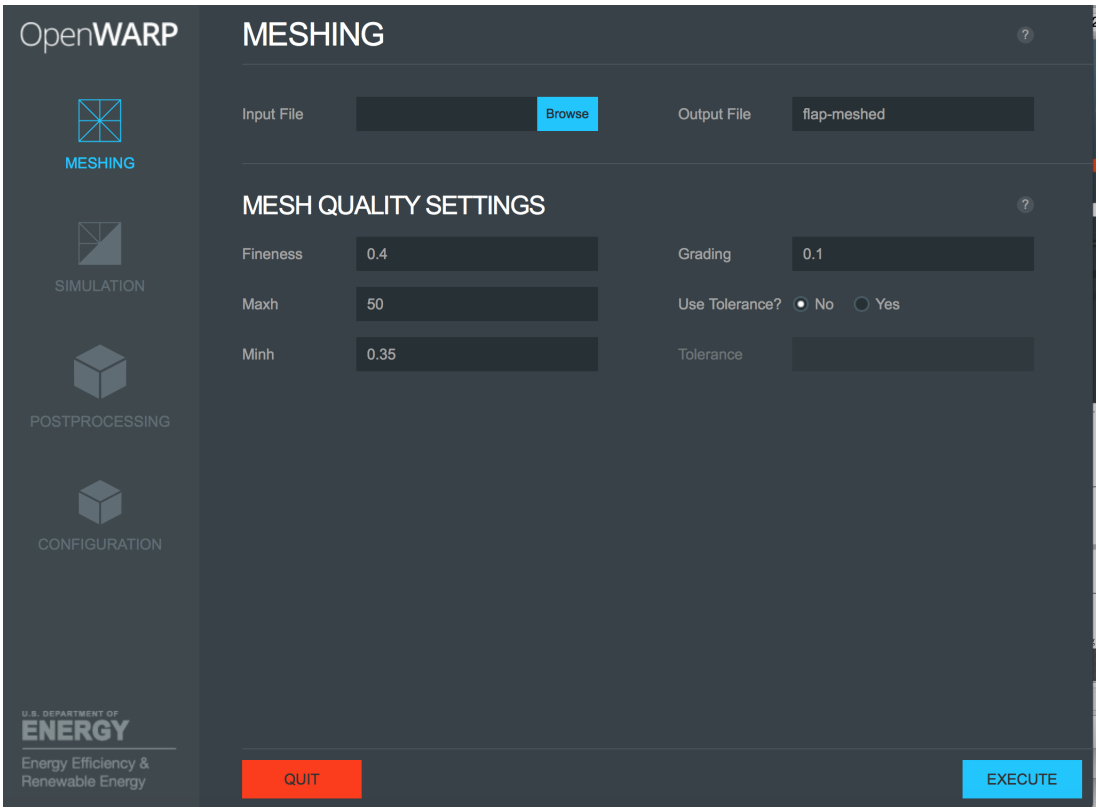
You can quit the application by clicking the orange Quit button. Quitting results in:

You must restart the Python server before using the application again.

9.2 Start meshing

Click the Meshing button:

This is the meshing page:



Select the input file by clicking Browse and selecting a file. Set the other parameters

according to your needs.

9.3 Execute meshing

Click the blue Execute button.

9.4 View the meshing results

Once meshing is complete, you will see a popup displaying a log:

The meshing converts the input file to all formats recognised by the application. It salso convert the input file to a mesh containing only quadrilateral elements without any triangle. The output directory is **$ROOT**/src/user_data/meshing_TIMESTAMP_HASH where **$ROOT**is the top-level directory of the OpenWarp package. The timestamp is in the format YYYYMMDDHHMM. Navigate to the directory with the latest timestamp. Inside the directory, you will find:

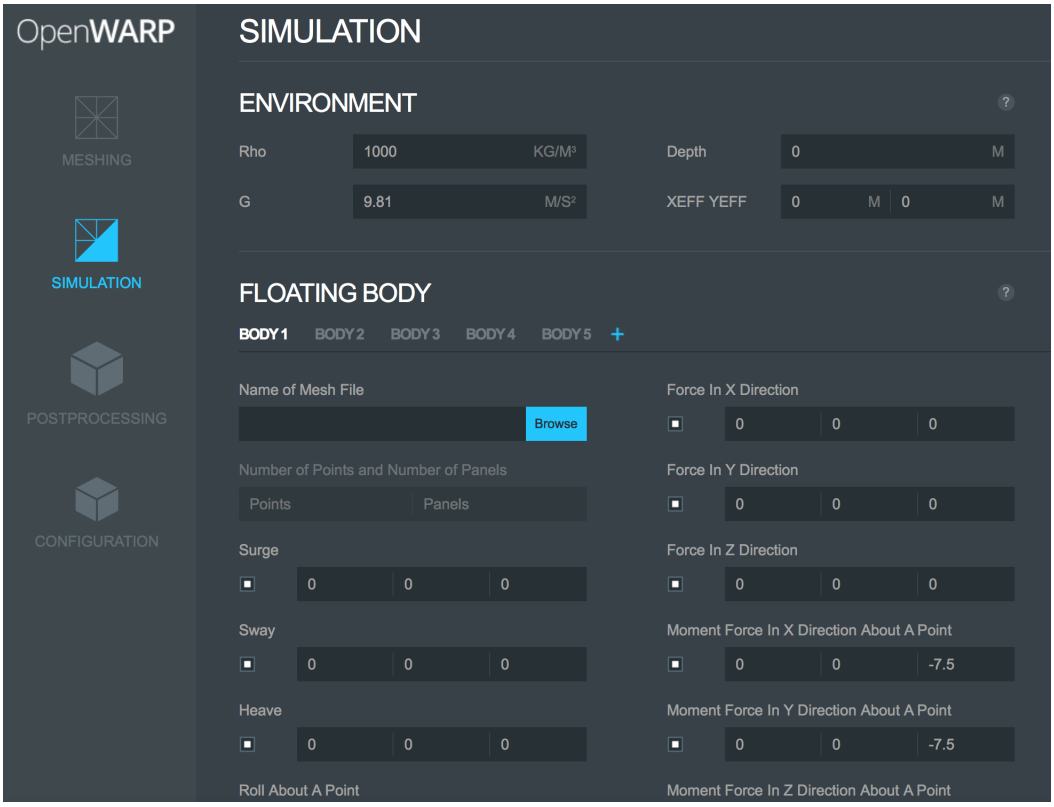
- Two files ending in .vtp, containing the mesh in VTP format. The "quad" version contains only quadrilateral elements.
- Two files ending in .vtk, containing the mesh in VTK format.
- Two files ending in .dat, the format recognized by Nemoh.
- Two files ending in .gdf, containing the mesh in GDF format.
- Two files ending in .stl, containing the mesh in stl format.

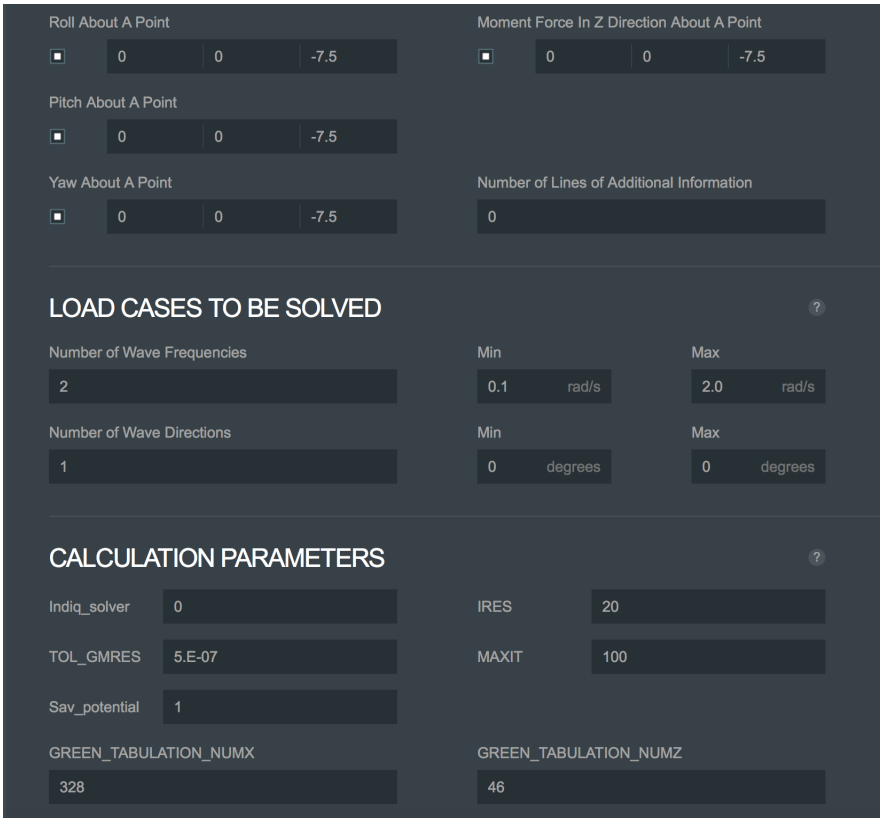
Each file is named OUTPUT_FILE_NAME.EXTENSION or OUTPUT_FILE_NAME-quad.EXTENSION, where OUTPUT_FILE_NAME is the output file name configured before meshing and EXTENSION corresponds to the file format. For each pair of files, the "quad" version contains only quadrilateral elements.

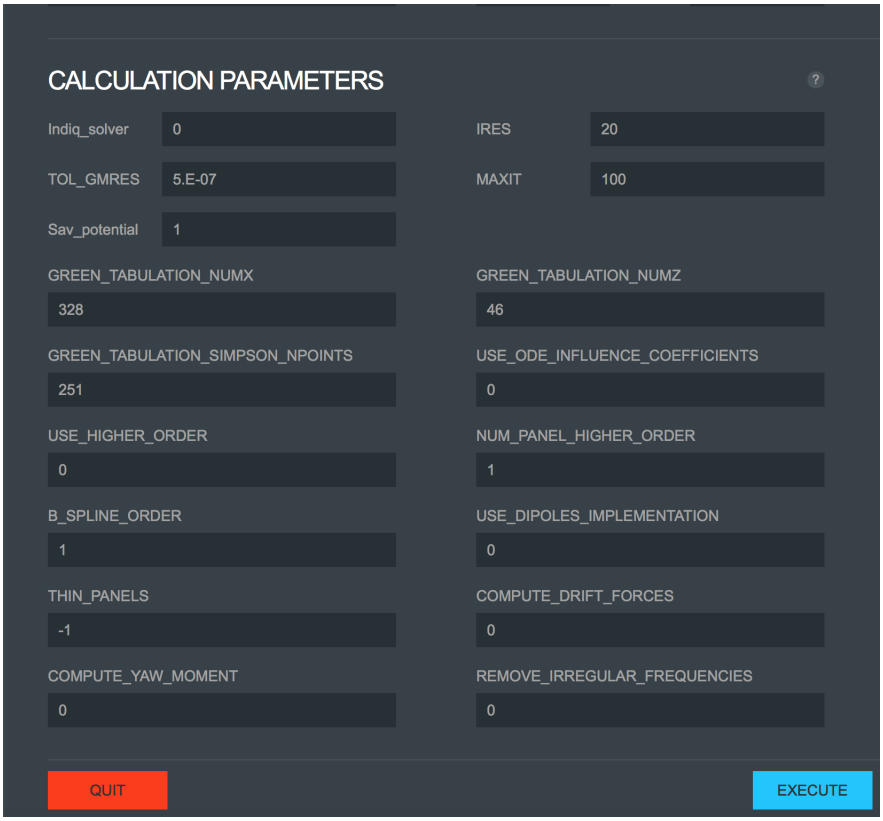
9.5 Start the simulation

Click the Simulation button:

This is the resulting page:

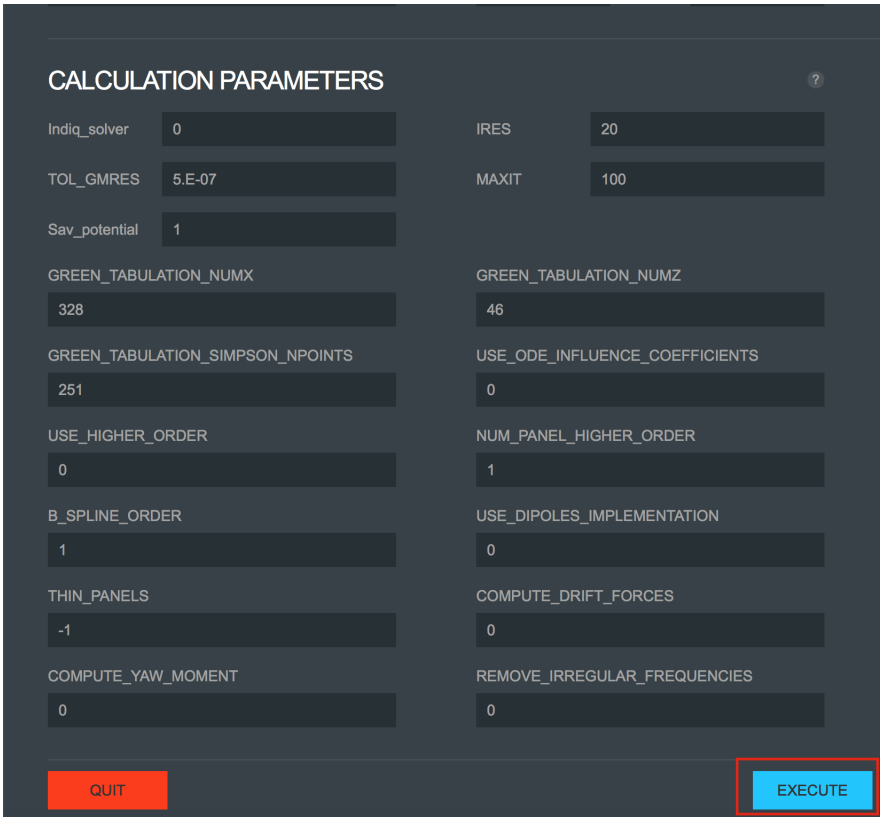

OpenWARP
SIMULATION
MESHING
SIMULATION
POSTPROCESSING
CONFIGURATION
ENVIRONMENT
Rho 1000 KG/M³
G 9.81 M/S²
Depth 0 M
XEFF YEFF 0 M 0 M
FLOATING BODY
BODY 1 BODY 2 BODY 3 BODY 4 BODY 5 +
Name of Mesh File
Browse
Number of Points and Number of Panels
Points Panels
Surge 0 0 0
Sway 0 0 0
Heave 0 0 0
Roll About A Point
Force In X Direction 0 0 0
Force In Y Direction 0 0 0
Force In Z Direction 0 0 0
Moment Force In X Direction About A Point 0 0 -7.5
Moment Force In Y Direction About A Point 0 0 -7.5
Moment Force In Z Direction About A Point


Roll About A Point 0 0 -7.5
Pitch About A Point 0 0 -7.5
Yaw About A Point 0 0 -7.5
Moment Force In Z Direction About A Point 0 0 -7.5
Number of Lines of Additional Information 0
LOAD CASES TO BE SOLVED
Number of Wave Frequencies 2
Min 0.1 rad/s
Max 2.0 rad/s
Number of Wave Directions 1
Min 0 degrees
Max 0 degrees
CALCULATION PARAMETERS
Indiq_solver 0
IRES 20
TOL_GMRES 5.E-07
MAXIT 100
Sav_potential 1
GREEN_TABULATION_NUMX 328
GREEN_TABULATION_NUMZ 46



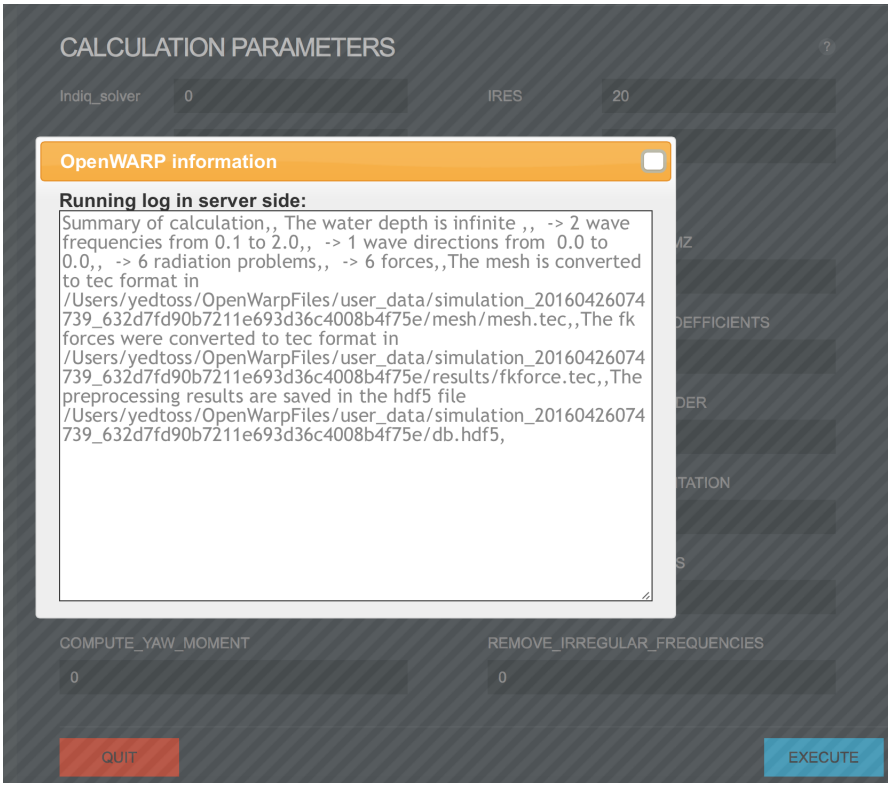
9.6 Execute the simulation

Click the Execute button at the bottom right of the page.

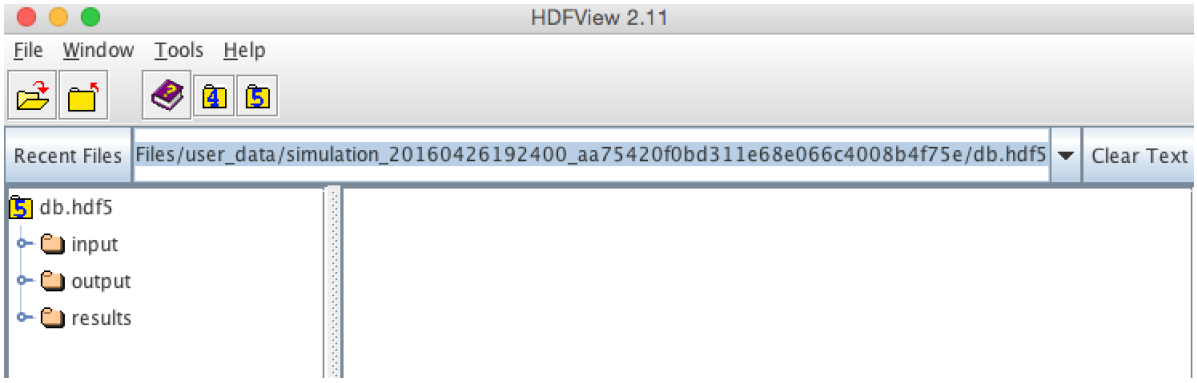


9.7 View the simulation results

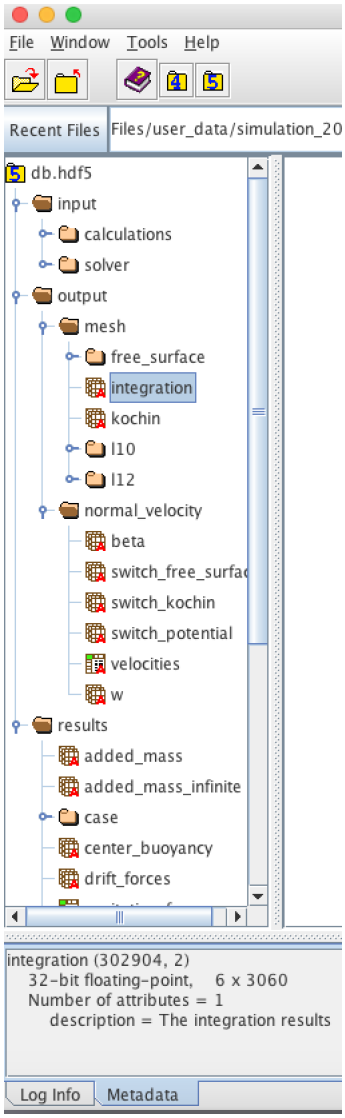
Once the simulation has completed, you will see a popup containing the calculation log:



The output directory is **$ROOT**/src/user_data/simulation_TIMESTAMP_HASH where **$ROOT**is the top-level directory of the OpenWarp package. The timestamp is in the format YYYYMMDDHHMM. Navigate to the directory with the latest timestamp. The main result at this step is the db.hdf5 file inside that directory. You can open it with HDFView:



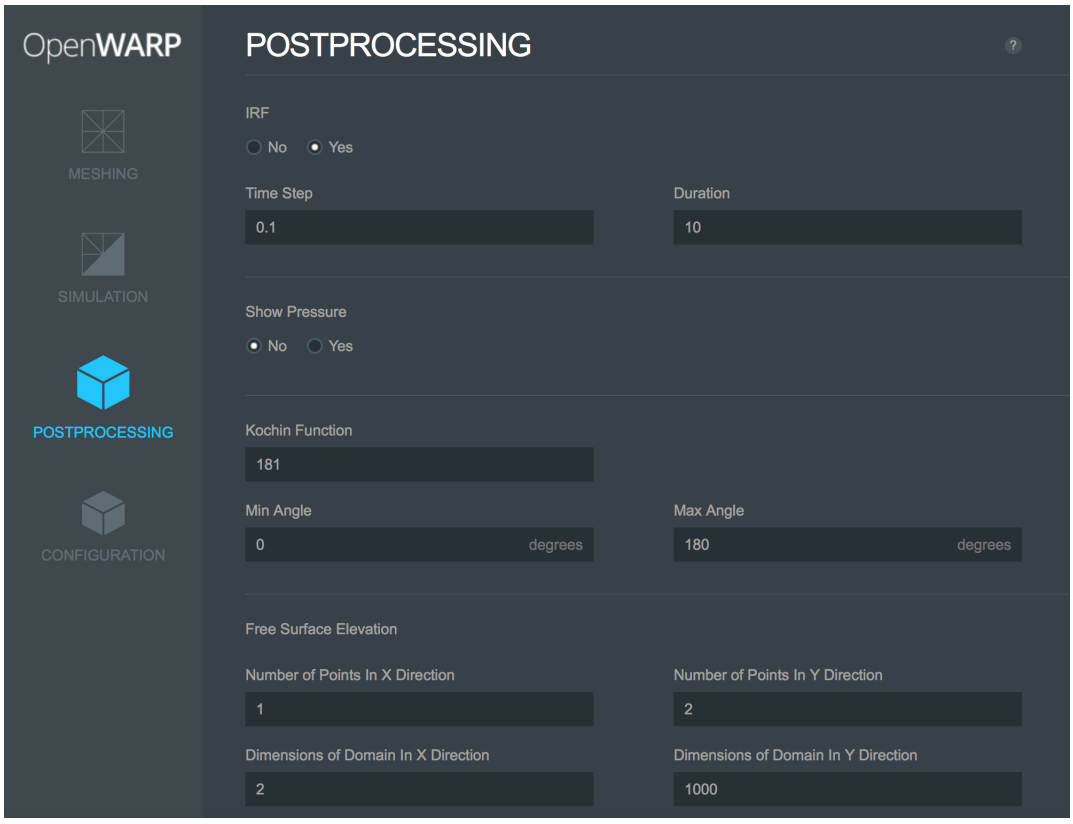
You will see that it contains three main groups: the input group, the output group containing intermediate results generated by the application, and the results group containing final results. In each group, you can see datasets with their description as shown in the figure below.

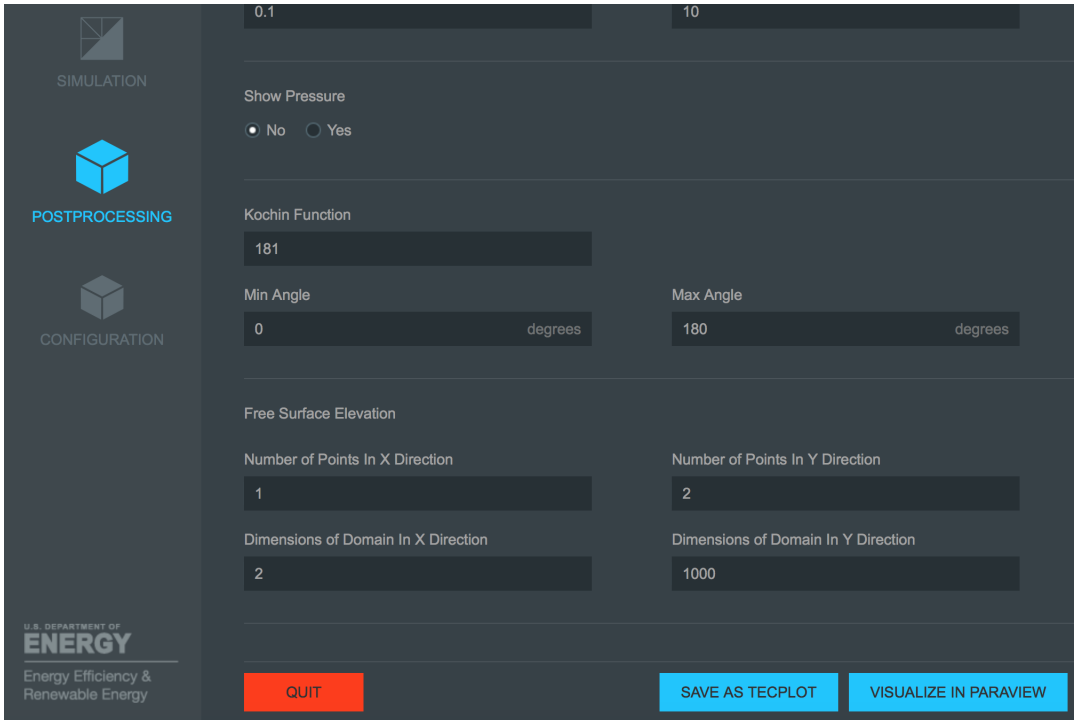


9.8 Postprocessing

Click the Postprocessing button:

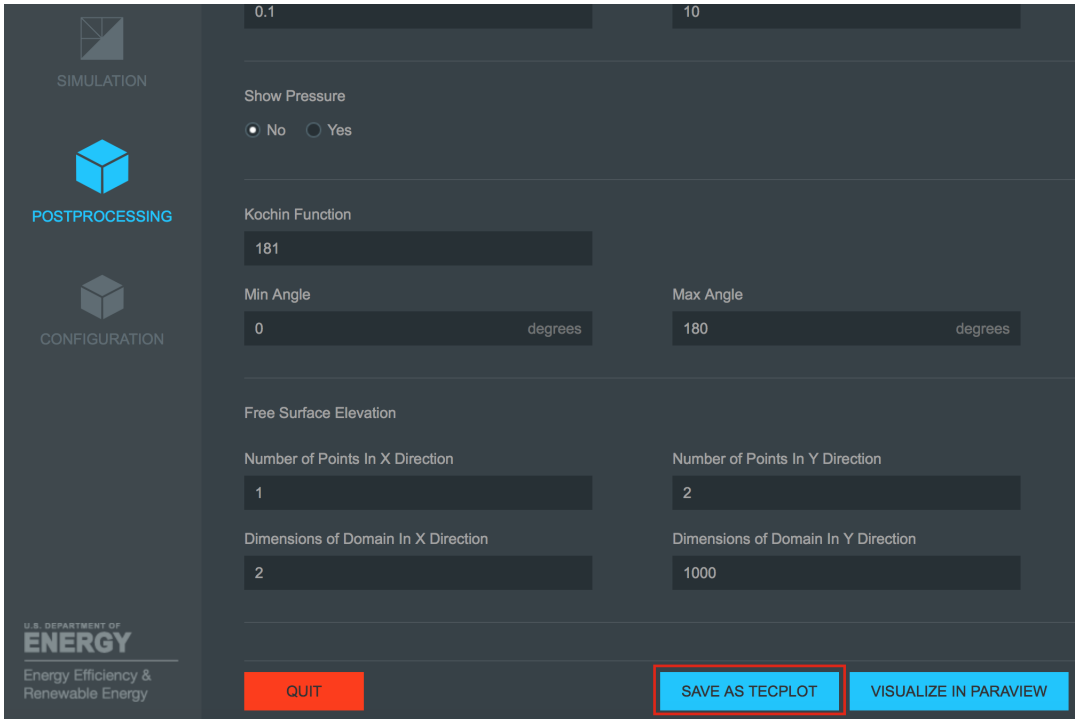
You should see the following:





9.9 Generate TEC outputs

Generate the TEC file by clicking on "SAVE AS TECPLOT". Note that before generating a TEC file, you must have executed a simulation in the current session.



9.10 Postprocessing results

Once the postprocessing is done, you can manually examine the results in the output directory **$ROOT**/src/user_data/simulation_TIMESTAMP_HASH where **$ROOT**is the top-level directory of the OpenWarp package. The timestamp is in the format YYYYMMDDHHMM. Navigate to the directory with the latest timestamp. You will find the following output files:

results/irf.tec : This file contains the infinite frequency added mass and the impulse response function for the radiation force.

results/WaveField.tec : The wave field in TECPLOT format.

results/diffractionforce.tec : The diffraction force for the diffraction problems.

results/excitationforce.tec : The excitation force for the diffraction problems.

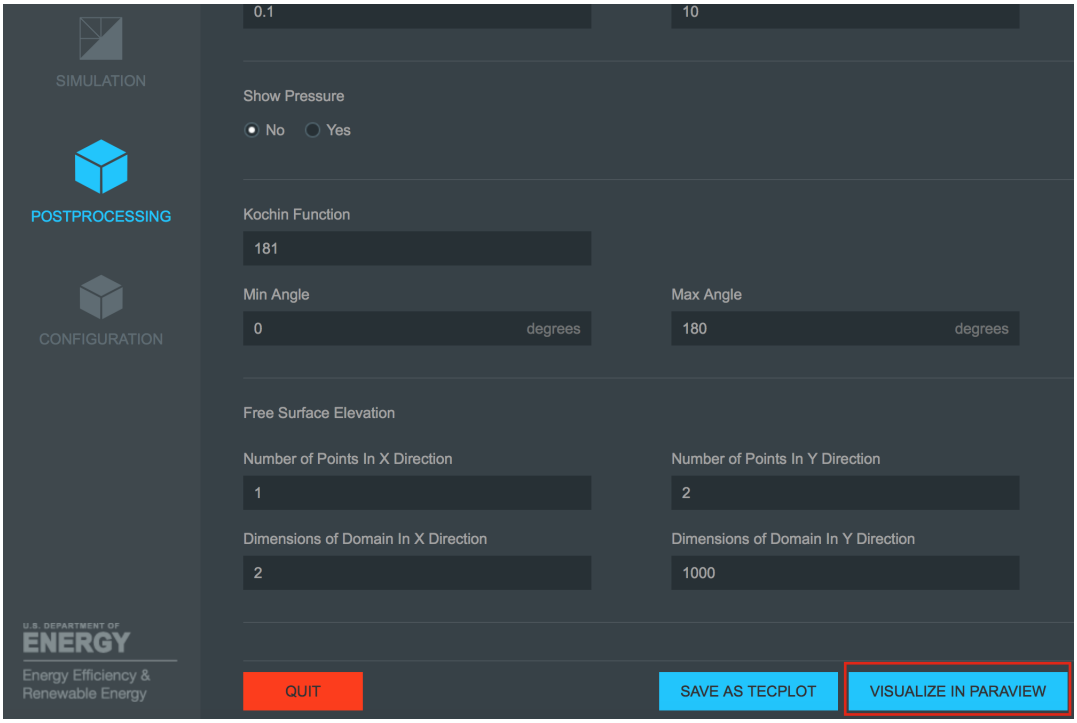
results/radiationcoefficients.tec : The added mass and damping forces for the radiation problems.

results/fkforce.tec : The Froude-Krylov forces for the diffraction problems.

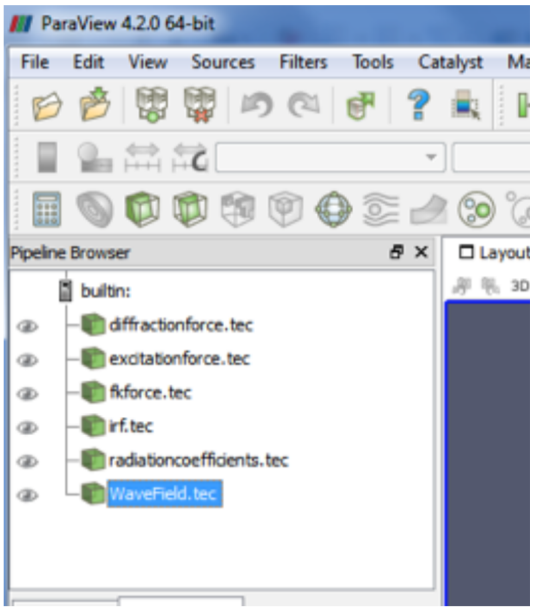
mesh/mesh.tec : The mesh file in TECPLOT format. It contains tables of nodes and connections.

9.11 Visualize the generated TEC files

Instead of manually viewing the postprocessing output files, you can visualize them with ParaView. After generating the TEC files, click on "VISUALIZE IN PARAVIEW":

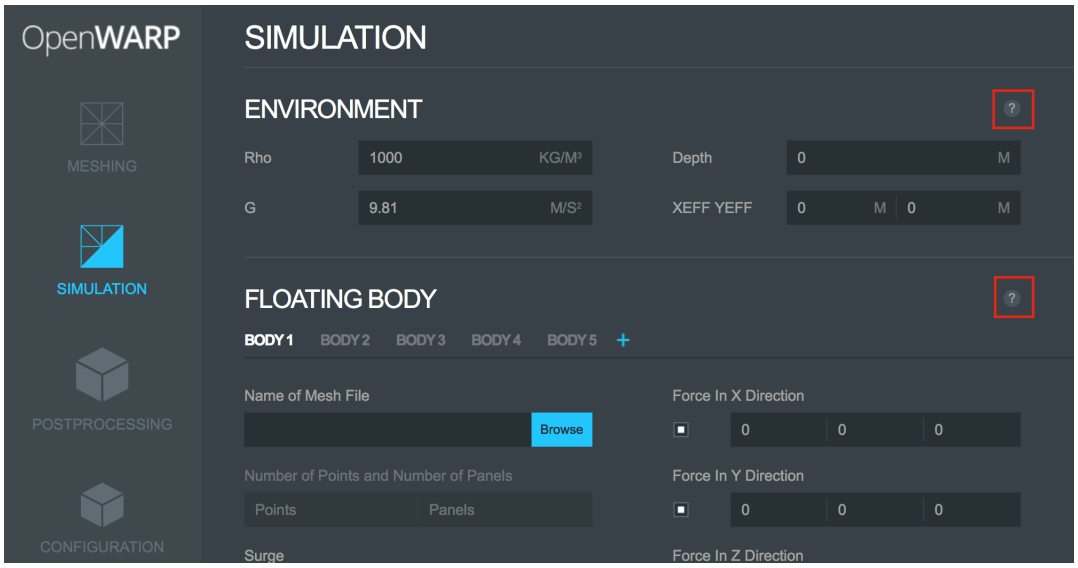


ParaView will start and you will find the resulting forces on the left side:



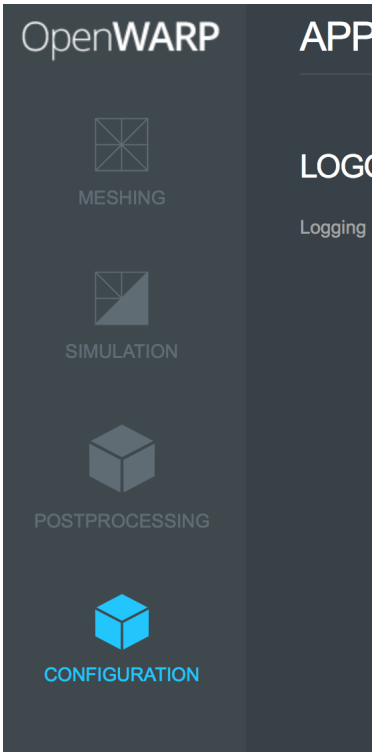
9.12 Getting help

You can click on the Help icons displayed on each page:

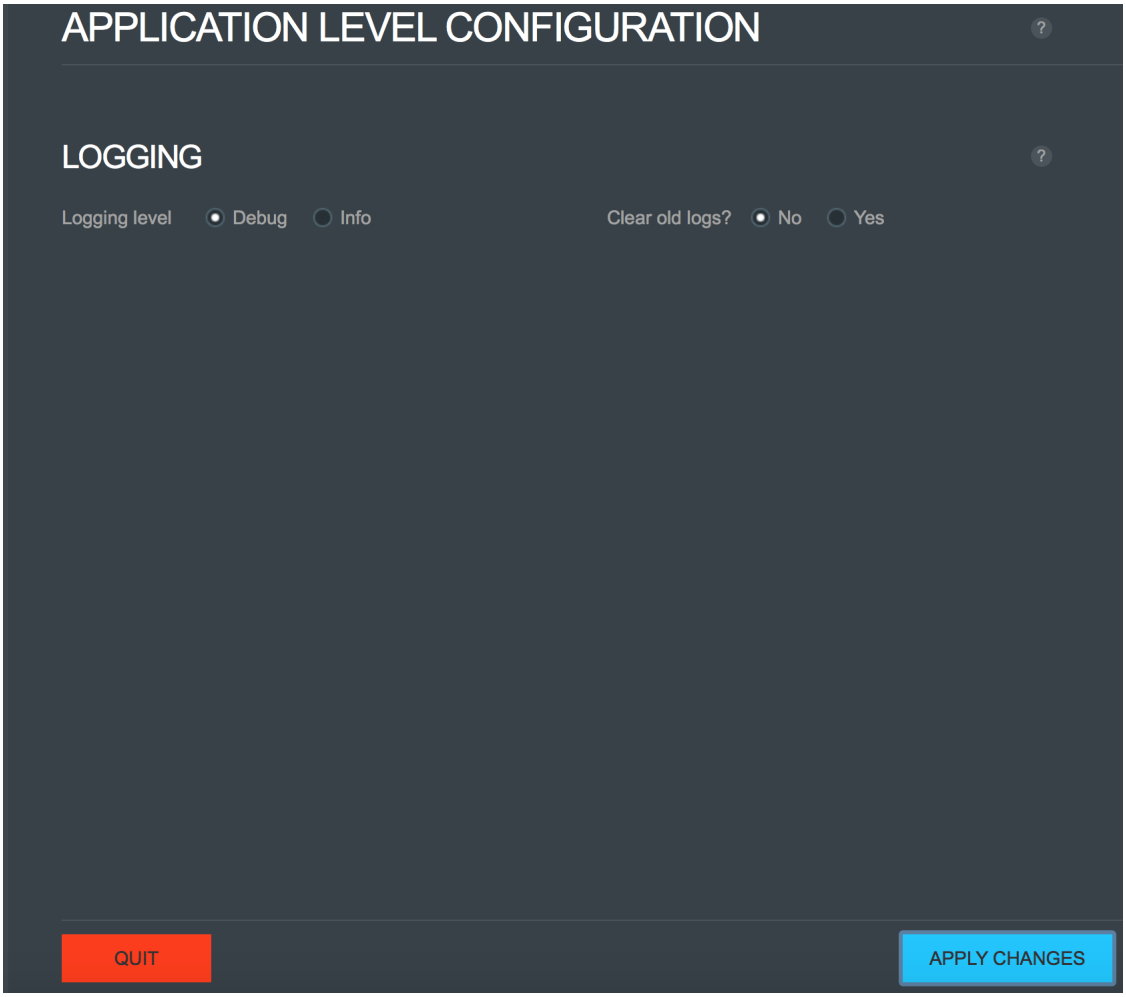


9.13 Logging Support

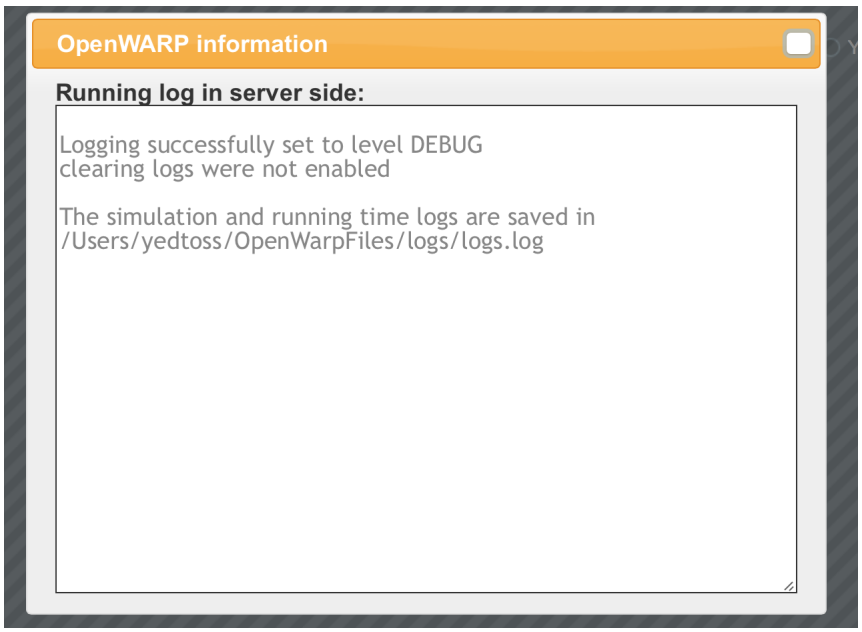
To configure the logging click on the CONFIGURATION screen as shown below:



After setting the configuration as your wish, you should click on "APPLY CHANGES" to make sure your changes are taken into account.



Once done, you will get a status of the change request as well as the location where the logs are saved.

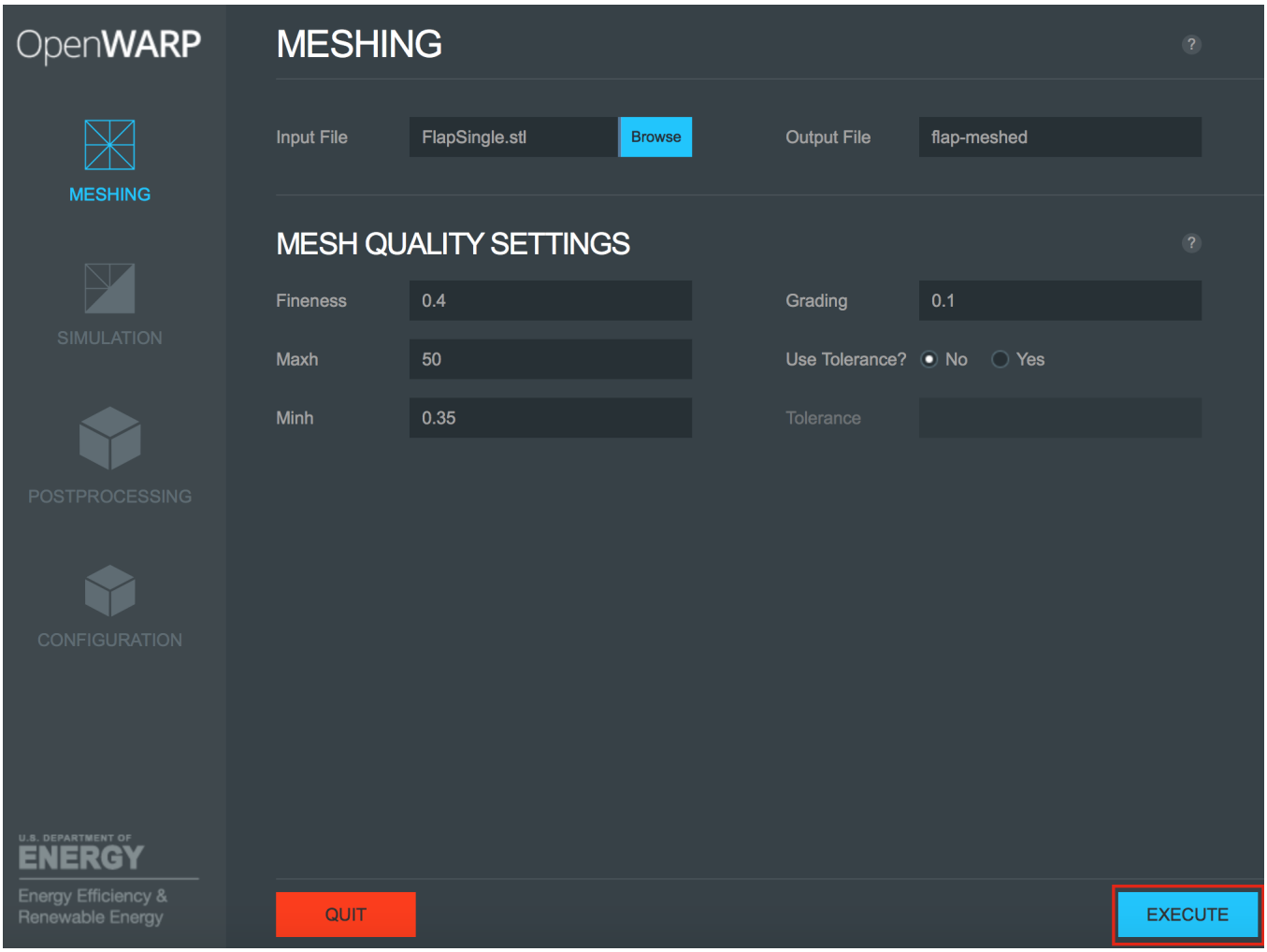


10 Practical Example

We are going to use the FlapSingle example available in **$ROOT**/test_files where **$ROOT**is the top-level directory of the OpenWarp package. It is also available in **$INSTALL_DIR**/OpenWarp/test_files First, we will generate the mesh of the body. Next, we will run the simulation. Finally, we will generate the TEC files and visualize them.

10.1 Mesh Generation

Go to the Meshing page, click on Browse, and choose **$ROOT**/test_files/mesh/FlapSingle.stl where **$ROOT**is the top-level directory of the OpenWarp package and Execute the meshing:

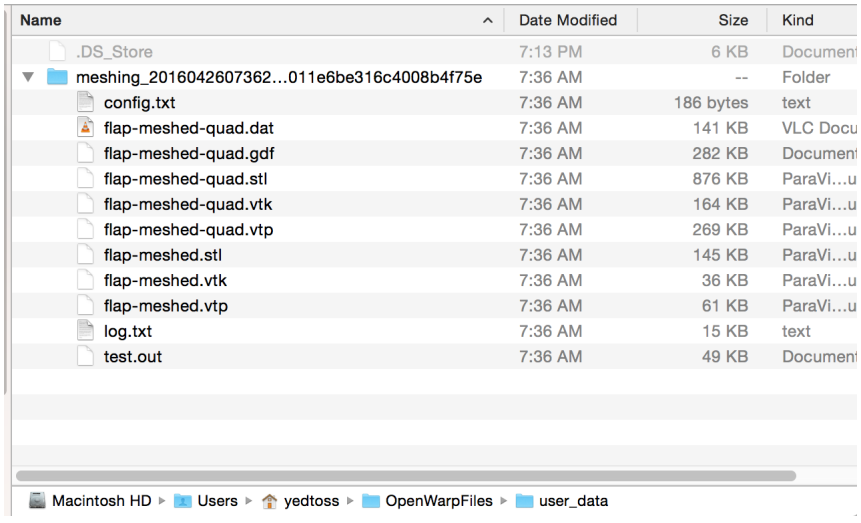


You will see the results shown like this :

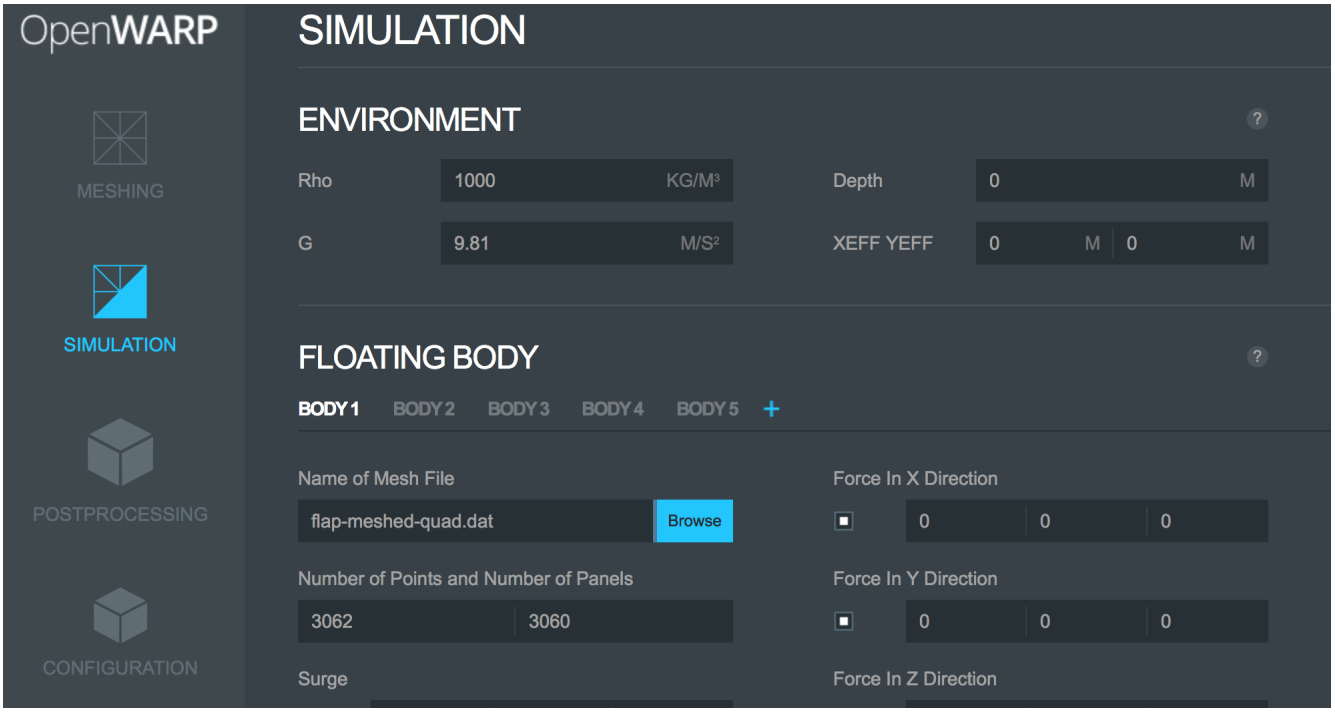
Click the button in the corner of the popup to close it:

10.2 Run the simulation

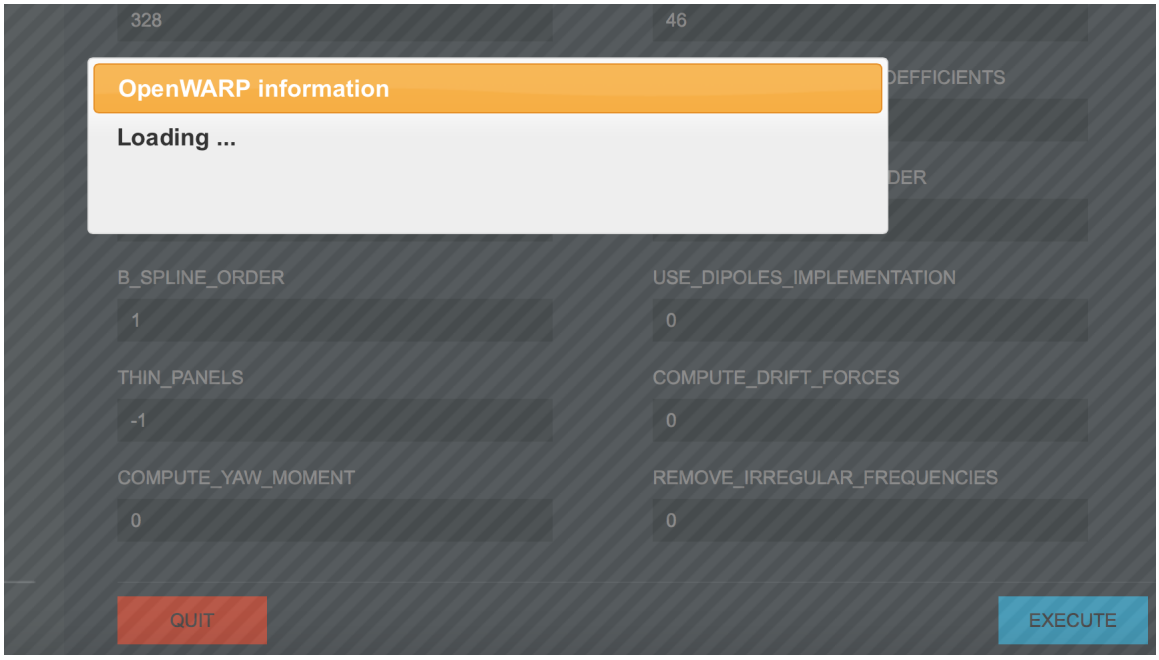
Go to the simulation page and click on Browse to load the generated mesh file. The generated mesh file is inside the $HOME/OpenWarpFiles/user_data/meshing_TIMESTAMP_HASH where $HOME is your home directory. Go to the directory with the latest timestamp.



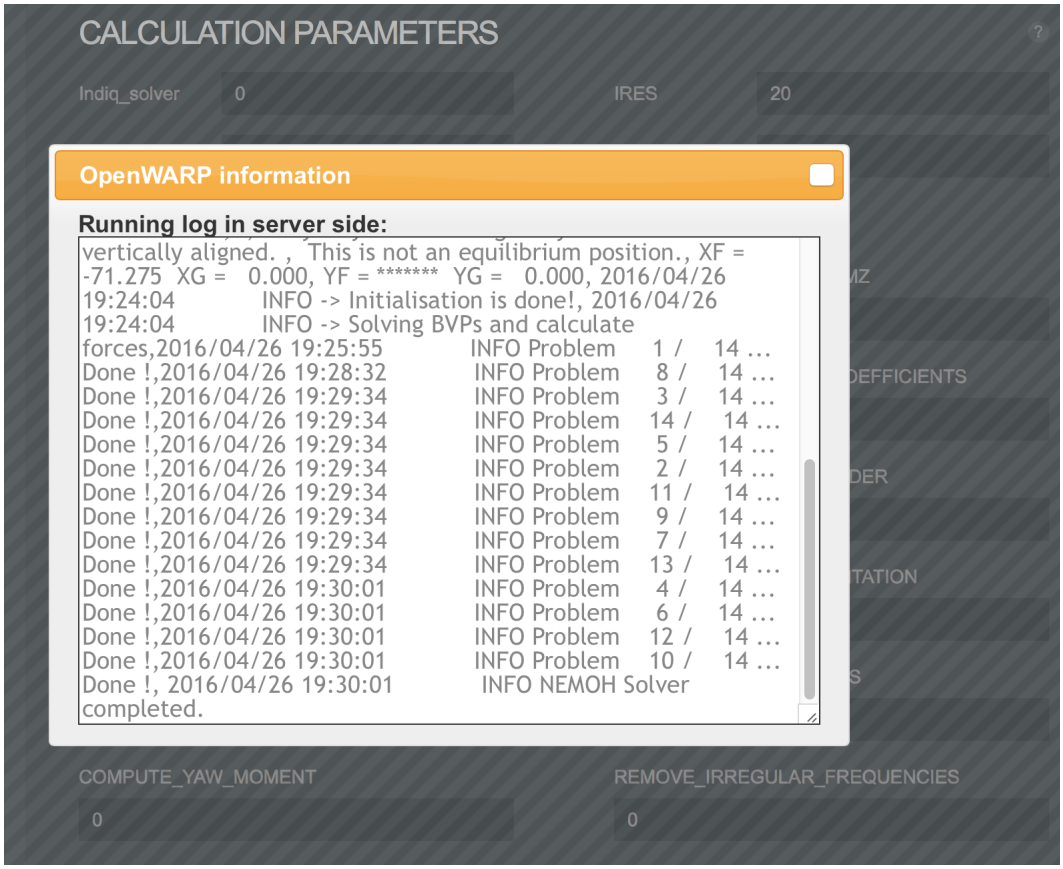
Look for a .dat file. It is named flap-meshed-quad.dat in our case.



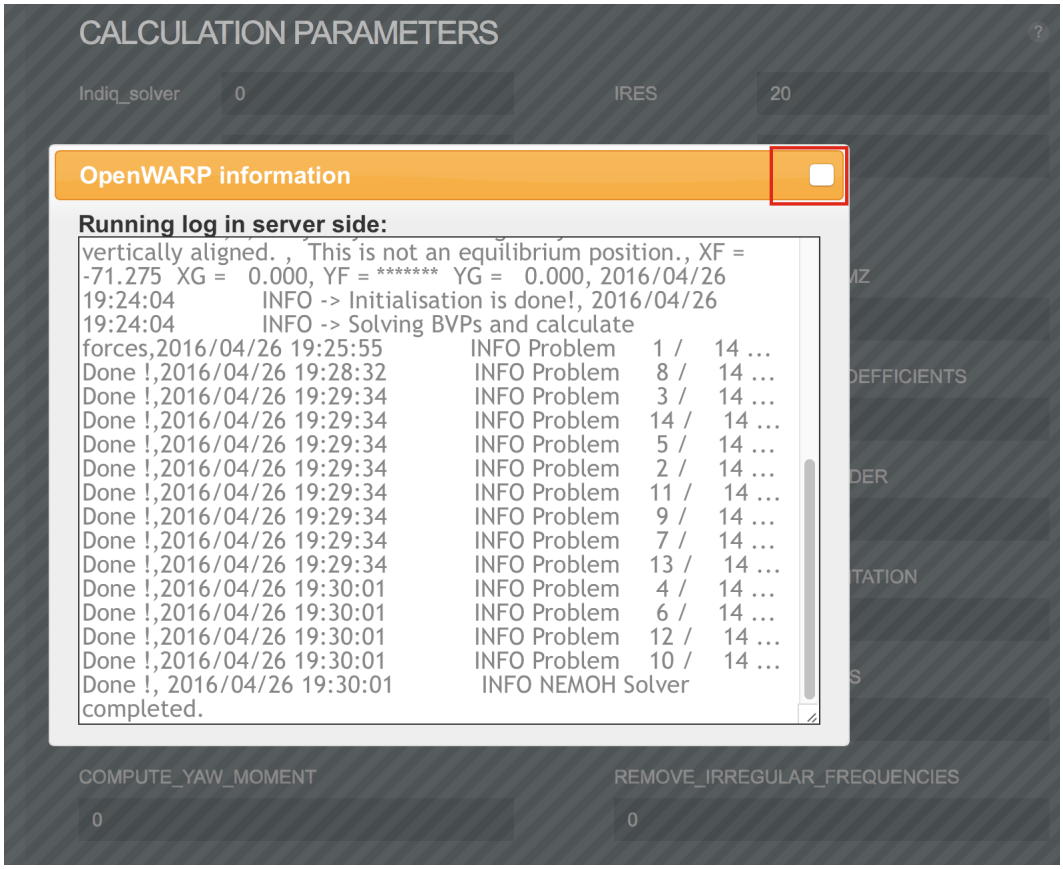
Run the simulation by clicking on "EXECUTE". You should see this:



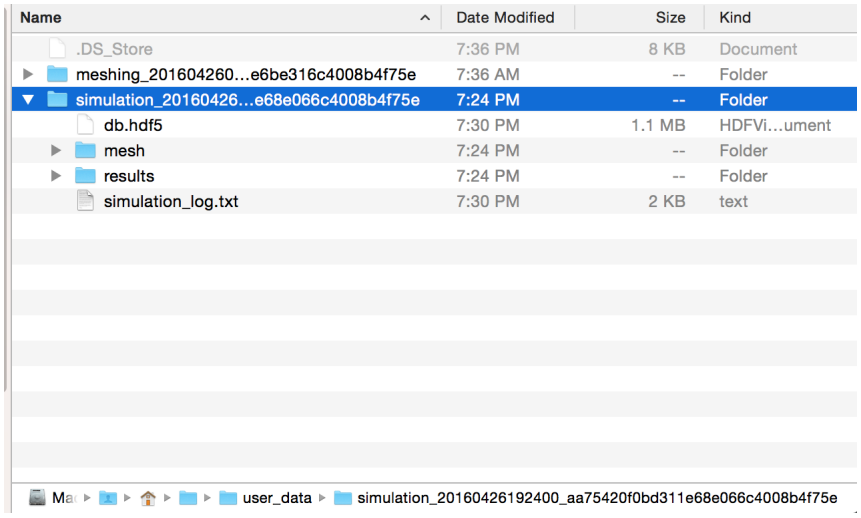
When the simulation is done (You should be patient as it will take a few minutes), the results are summarized in a popup:



Close the popup:

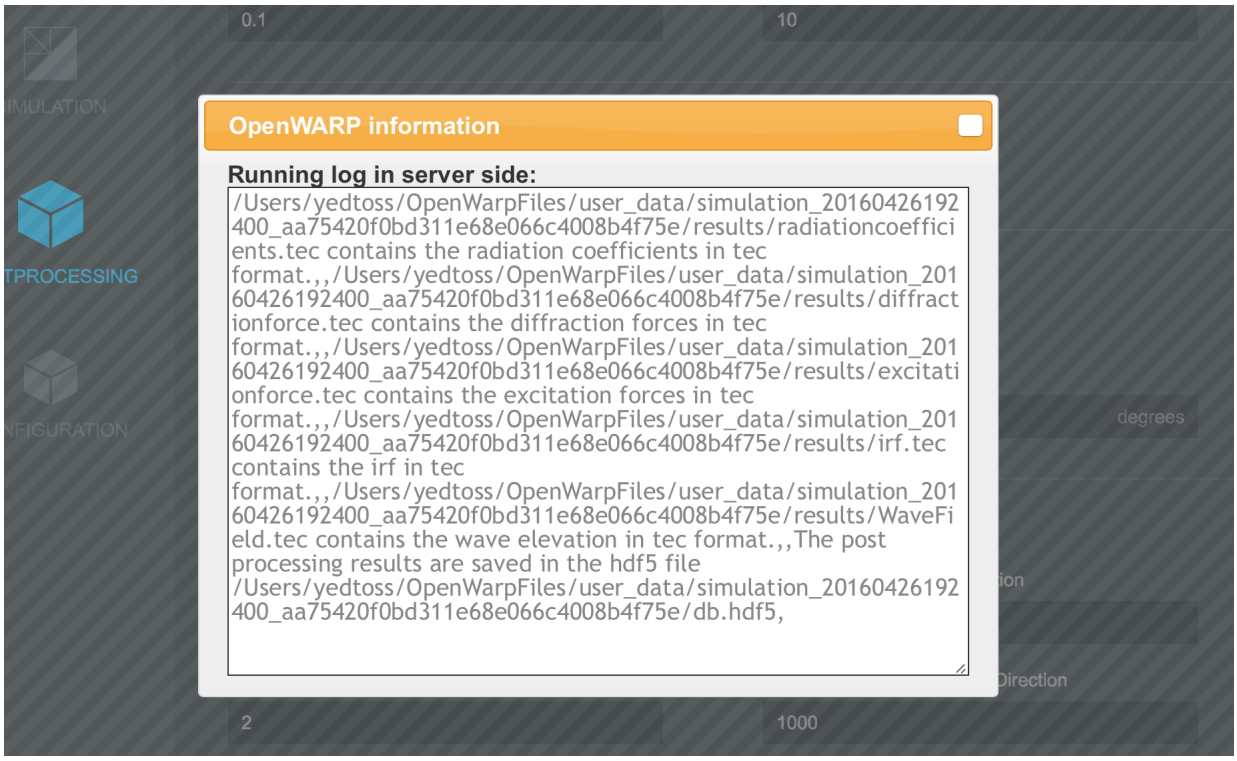


Inside the directory $HOME/OpenWarpFiles/user_data/simulation_TIMESTAMP_HASH, you will see the generated simulation results where $HOME is the home directory of the current user. Find db.hdf5 inside the directory with the latest timestamp; it contains the results.



10.3 Generate TEC files and run postprocessing

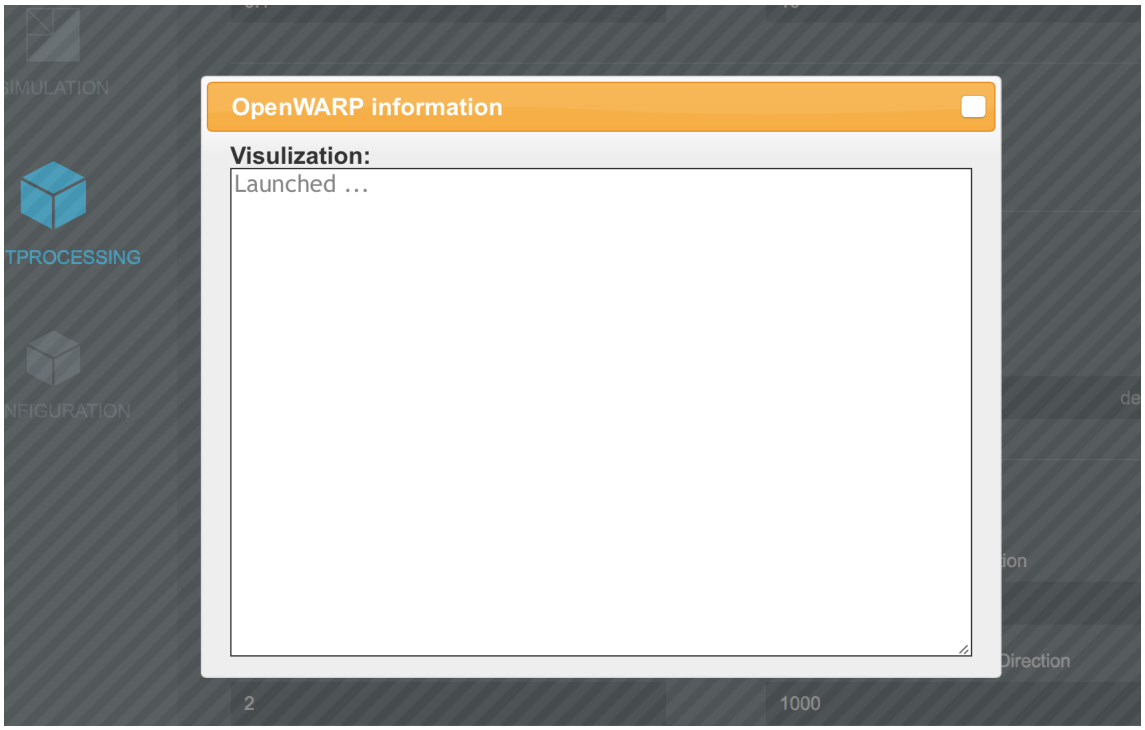
Open the postprocessing page. Leave the default values. Generate the TEC files by clicking "Save as TECPLOT". When postprocessing is done, you get a popup:



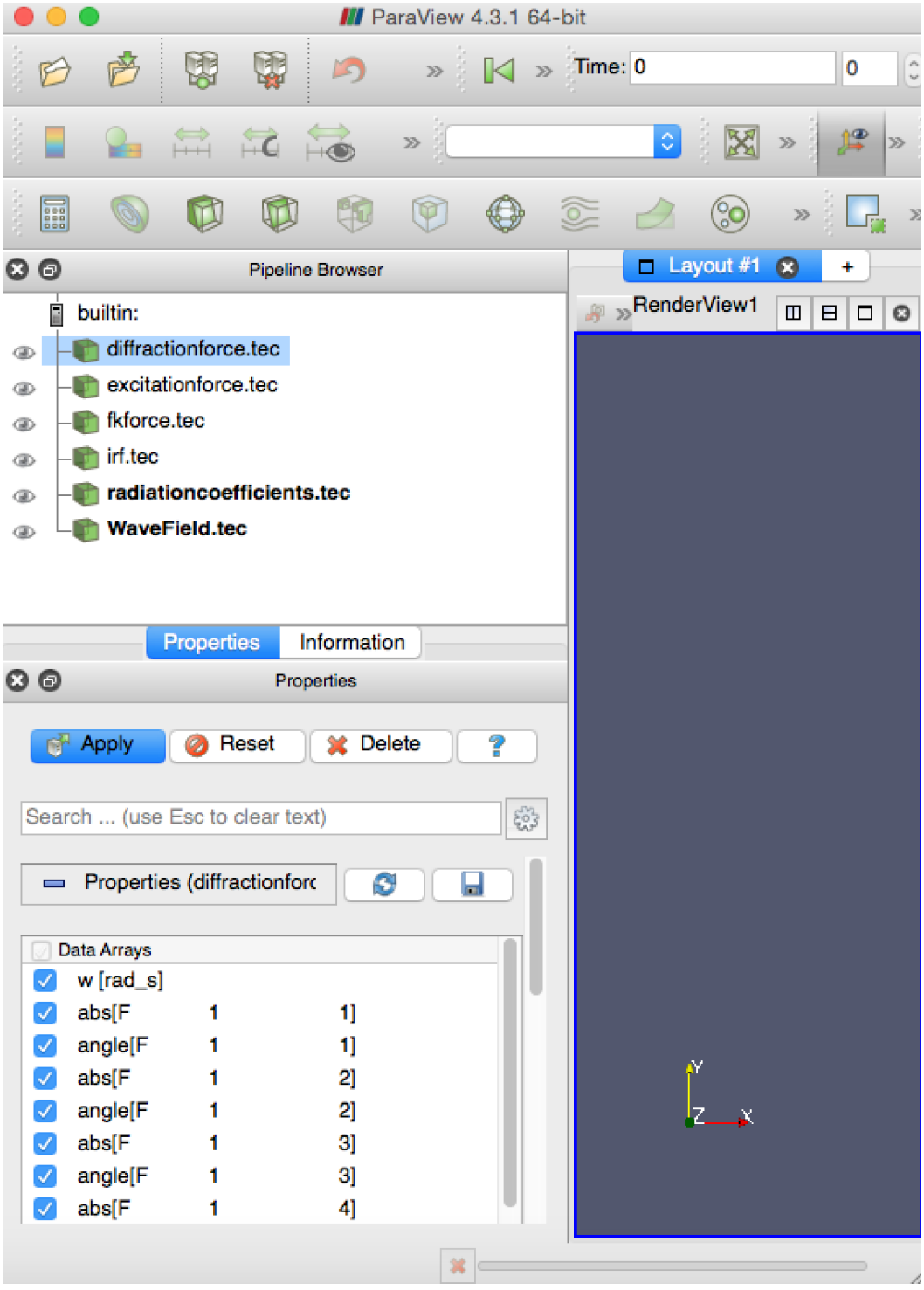
Close it.

10.4 Visualize the generated tec files

Click on "VISUALIZE IN PARAVIEW". You will see this while ParaView opens:



Once ParaView is open, you will see the generated forces on the left:


ParaView 4.3.1 64-bit
Time: 0
0
Pipeline Browser
builtin:
diffractionforce.tec
excitationforce.tec
fkforce.tec
irf.tec
radiationcoefficients.tec
WaveField.tec
Layout #1
RenderView1
Properties
Information
Properties
Apply
Reset
Delete
Search ... (use Esc to clear text)
Properties (diffractionforc
Data Arrays
w [rad_s]
abs[F 1 1]
angle[F 1 1]
abs[F 1 2]
angle[F 1 2]
abs[F 1 3]
angle[F 1 3]
abs[F 1 4]
Y
Z
X