Merge Code and Update GUI Deployment Guide

Revision History

Author	Revision Number	Date
TCSASSEMBLER	1.1	02/09/2015

1. Organization of Submission

The content of submission follows directory structure as below:

docs – Contains this deployment guide.

docs/Nemoh Contains all Nemoh deployment guides

docs/OpenWarp Contains all OpenWarp deployment guides

src – Contains the whole tool suite of this assembly.

src/nemoh - Contains the modified python version of Nemoh

src/openwarp - Contains the modified openwarp gui

NemohMerged - Contains the modified merged Nemoh

test_files– Contains files that help verify this submission.

QuickWindowsInstallStep.txt - Quick Installation instructions for Windows

README.txt - README file. Please reviewer, read it. It contains useful information.

ChangeLog.txt The changelogs

2. Application Setup

2.1 Environment

- Windows 7 64bit
- Mac OS X 10.9 64bit
- Anaconda (with Python 2.7) >= 2.1.0 http://continuum.io/downloads
- Python 2.7.8 X64 provided by Anaconda
- CherryPy 3.5.0 http://www.cherrypy.org/
- Chrome/Firefox/Safari
- ParaView 4.1 http://www.paraview.org/download/

3. Deployment

3.1 Common steps

We need to compile the Nemoh library as well as the python module. For this, follow the deployment instructions (Section 5) of **docs/Nemoh/Dipoles Implementation in NEMOH.pdf** (The new Nemoh deployment guide)
Note that in the new Nemoh deployment guide,
$NEMOH_FORTRAN will now be the directory Nemoh/ in the root of this submission directory
$NEMOH_PYTHON the directory src/nemoh

Once this is done, you need to install the remaining python dependencies, currently only CherryPy. To do so, go to src/ and run pip install -r requirements.txt

Make sure before running any remaining command that the path are correctly setup in your shell according to the new nemoh deployment guide (docs/Nemoh/Dipoles Implementation in NEMOH.pdf)

In this DG we will also borrow the environment variable **$MINGW_ROOT** from the new Nemoh DG.

3.2 Steps for windows

On windows, you would need having the **nglib-mesh.exe** compiled according to the previous assemblies for Windows provided at
http://apps.topcoder.com/forums/?module=Thread&threadID=829570&start=0
.

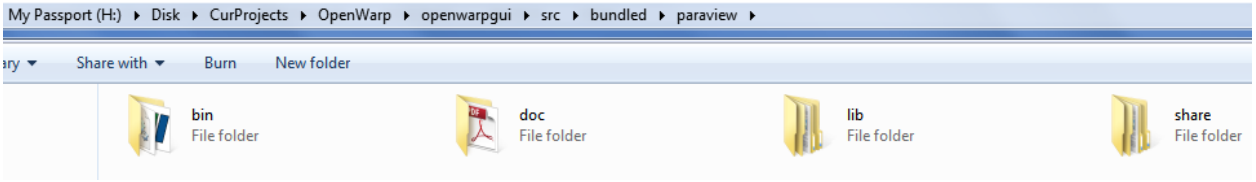
However, I have provided pre-compiled executable files for Windows 7(64 bit). Check them in folder "src\bundled\mesh-generator\bin". So that you don't need to compile yourself.

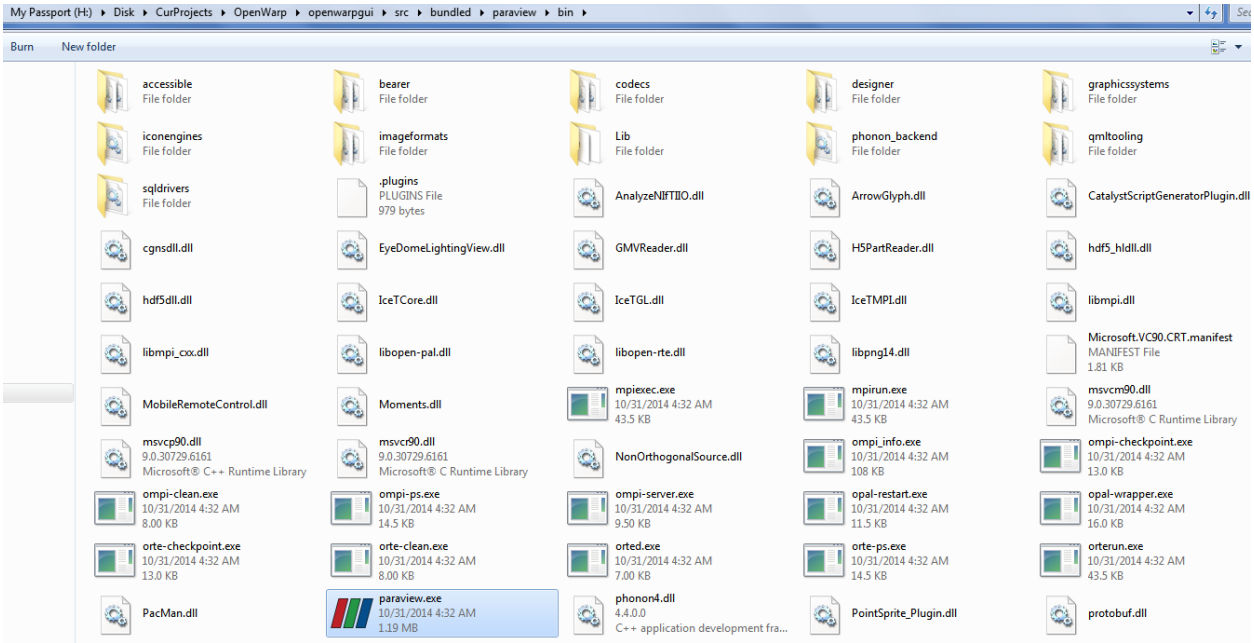
*Note that in case you met visual studio dependency issues, please download and install VS2010/VS2012 redistribution package for **X64** here:*
http://www.microsoft.com/en-us/download/details.aspx?id=14632
http://www.microsoft.com/en-us/download/details.aspx?id=30679#

Note also that I have provided precompiled version of the Nemoh library in "src\bundled\simulation\libs". You need to copy src\bundled\simulation\libs\libnemoh.dll and src\bundled\simulation\libs\libnemoh.dll.a src\bundled\simulation\libs\libblas.dll and src\bundled\simulation\libs\liblapack.dll to **$MINGW_ROOT\lib** and make sure **$MINGW_ROOT/lib** and **$MINGW_ROOT/bin** are in your PATH. See section 5.3.1.3 of **docs/ Dipoles Implementation in NEMOH.pdf** for more information on this

Please do note that this submission does NOT work well with IE browser. I suggest you install browsers like Chrome/Firefox/Safari and set one of them as the default browser before verify this assembly.

IMPORTANT: Please download the whole ParaView 4.1 X64 version as **ZIP** format from http://www.paraview.org/download/ . Extract the files under folder "src/bundled/paraview" so that we can invoke ParaView to do visualization. The files should be organized as below:





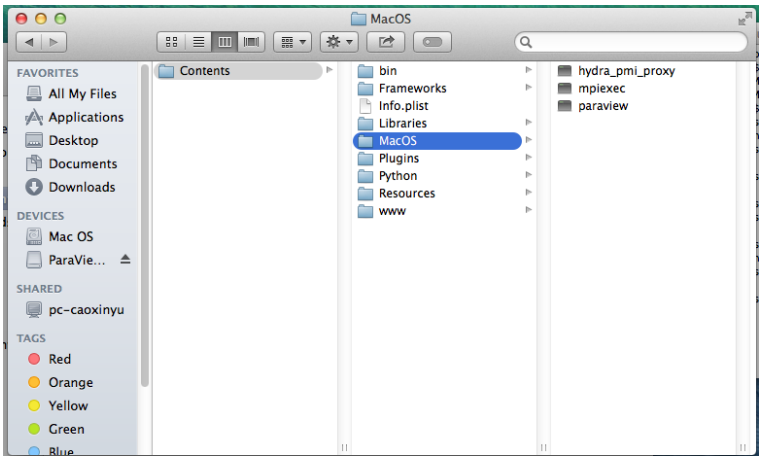
3.3 Steps for MAC OS X

On MAC, you would need to have the **nglib-mesh** compiled according to the previous assemblies for MAC provided at
http://apps.topcoder.com/forums/?module=Thread&threadID=829570&start=0
in forum.

There're **important** two tips for compiling on MAC OS X:

- You may need to configure "src/openwarp/settings.py" to change the listening port to some value other than 80, say 8000. Otherwise you would meet permission issues.
- You should first try to run "nglib-mesh" via terminal to ensure that the configuration and library dependencies are all right. In case you met **.dylib** dependency or loading issues, a short hack python script named "src/bundled/mesh-generator/lib/update_dylib.py" might help.

IMPORTANT: Please download the whole ParaView 4.1 For MAC OS X version from http://www.paraview.org/download/ and install it to your MAC OS X.
After that copy the whole "paraview.app" folder from your MAC OS to folder "src/bundled/" of this submission so that we could invoke ParaView to do visualization. The files under "src/bundled/paraview.app" should be organized as below:



3.4 Complete Quick Deployment steps for Windows 64 bits

Most of the tools are already compiled for Windows 7 64 bits.
(Not that they should also work for Windows 8 64 bits)
All you need to do to start testing in windows is to follow these steps:

1- Download MINGW 64 bits 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

   a) Extract it to a directory not containing spaces. Let's note the root as $MINGW_ROOT.
   b) Add $MINGW_ROOT\bin and $MINGW_ROOT\lib to the beginning of your windows PATH in Properties --> Advanced System Settings --> Advanced Tabs --> Environment Variables --> Path

2- Copy src\bundled\simulation\libs\libnemoh.dll, src\bundled\simulation\libs\libnemoh.dll.a, src\bundled\simulation\libs\libblas.dll and src\bundled\simulation\libs\liblapack.dll to $MINGW_ROOT\lib

3- Download and install Ananconda 2.10 with Python 2.7 for Windows 64 bits from at http://09c8d0b2229f813c1b93-c95ac804525aac4b6dba79b00b39d1d3.r79.cf1.rackcdn.com/Anaconda-2.1.0-Windows-x86_64.exe

If you install it to C:\Users\tcs\Anaconda for example, make sure that you manually add C:\Users\tcs\Anaconda and C:\Users\tcs\Anaconda\Scripts to the beginning of your windows Path

4- Once all previous steps are done, Start Powershell and install cherrypy (and other dependencies) by executing
   pip install -r requirements.txt (from the src/ directory)

5- Start the server by running python main.py (from the src/ directory)

6- You can also download the zip paraview from http://www.paraview.org/download/ and copy it to src/bundled/paraview such that src/bundled/paraview/bin exists. Needed only for testing Visualize

4. Verification

Open one terminal with the path correctly setup (See section 3.1), go to folder "**src/**" of this submission. Run following command to start the application:
python main.py

```
PS H:\Disk\CurProjects\OpenWarp\openwarpgui\src> python main.py
running build_ext
[31/Oct/2014:04:01:18] ENGINE Listening for SIGTERM.
[31/Oct/2014:04:01:18] ENGINE Bus STARTING
[31/Oct/2014:04:01:18] ENGINE Set handler for console events.
[31/Oct/2014:04:01:18] ENGINE Started monitor thread '_TimeoutMonitor'.
[31/Oct/2014:04:01:18] ENGINE Started monitor thread 'Autoreloader'.
[31/Oct/2014:04:01:18] ENGINE Serving on http://127.0.0.1
[31/Oct/2014:04:01:18] ENGINE Bus STARTED
```

After seconds, the default browser would open page http://127.0.0.1:80/index.html .

4.1 Integration Strategy

The new Nemoh code use hdf5 data for input and output and offer some convenience methods for converting from old text input to new hdf5 input.
This integration has not used these convenience methods as it would have been wrong. Instead, the python variables are passes directly to the new Nemoh code which are run as function instead of external program.

There are some important configurations variables in the new Nemoh which are now made available in the GUI when running a simulation

4.2 Mesh generator - Windows

Go to "MESHING" page. Please refer to the configuration parameters in "test_files/mesh/config.txt" of this submission as below to fill in the forms:

Note that most values are already pre-filled. All you need to fill is the path to the file test_files/mesh/FlapSingle.stl

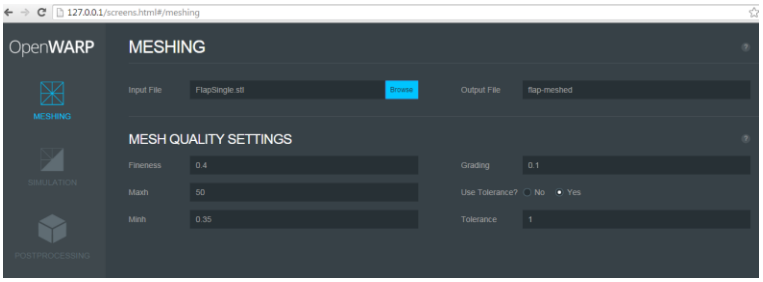
infile: FlapSingle.stl

 # outfile without extension
outfile: flap-meshed

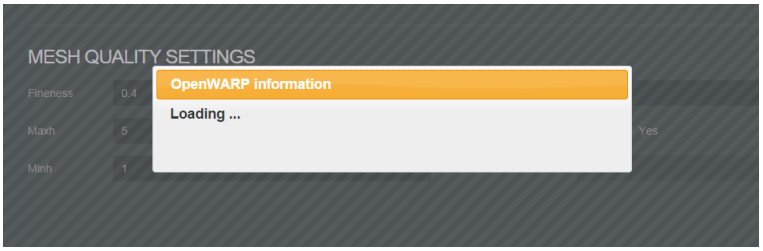
maxh: 50
minh: 0.35
fineness: 0.4
grading: 0.1

 # set one to use tolerace, otherwise 0
usetolerance: 0
 # volume tolerance in percent
tolerance: 1

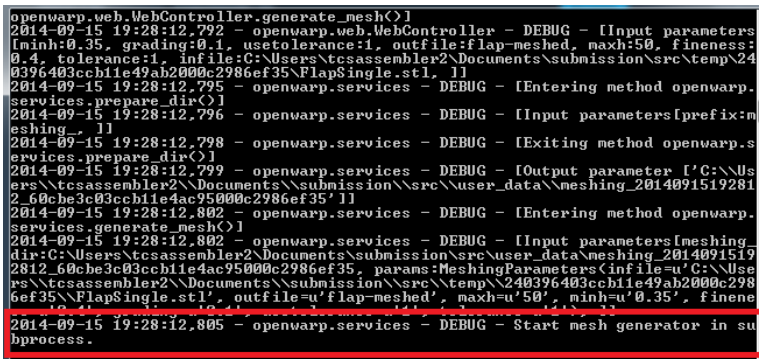
See the screen shot below:



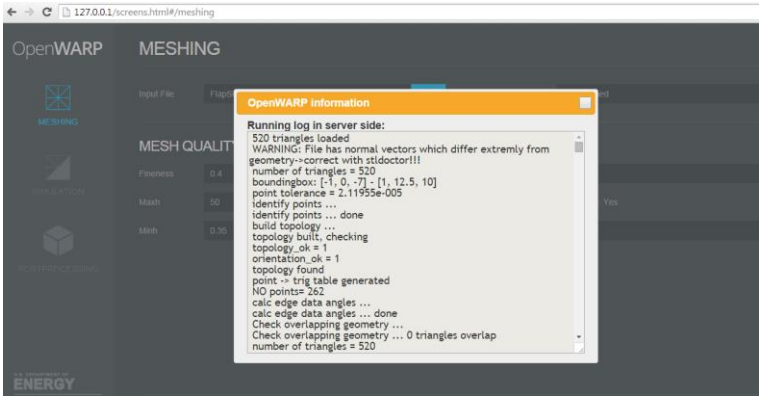
Click "**Browse**" button to upload the correct 3d files to server. Choose "test_files/mesh/FlapSingle.stl" for example.
Click the "**EXECUTE**" button to submit the parameters to server side and generate mesh files. A "Loading ..." dialog pops up to prevent you from editing and clicking anything of this page.



Notice the python log information, which shows that the **nglib-mesh** process has started and not finished yet.

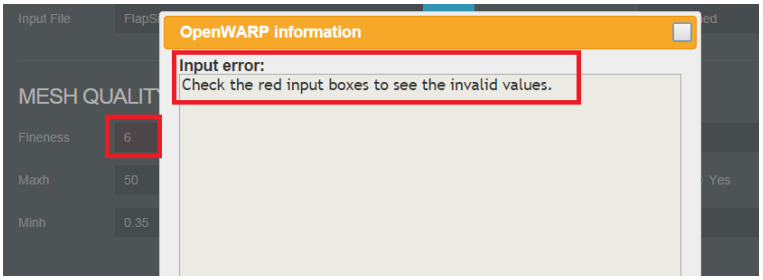


After the mesh generator finished, see the client side got the running log. The message was displayed in a popup dialog.



You may also check the generated mesh files in some path like "**src/user_data/meshing_20140915192812_60cbe3c03ccb11e4ac95000c2986ef35**". The really path may depends since a UUID was used to avoid conflicts.

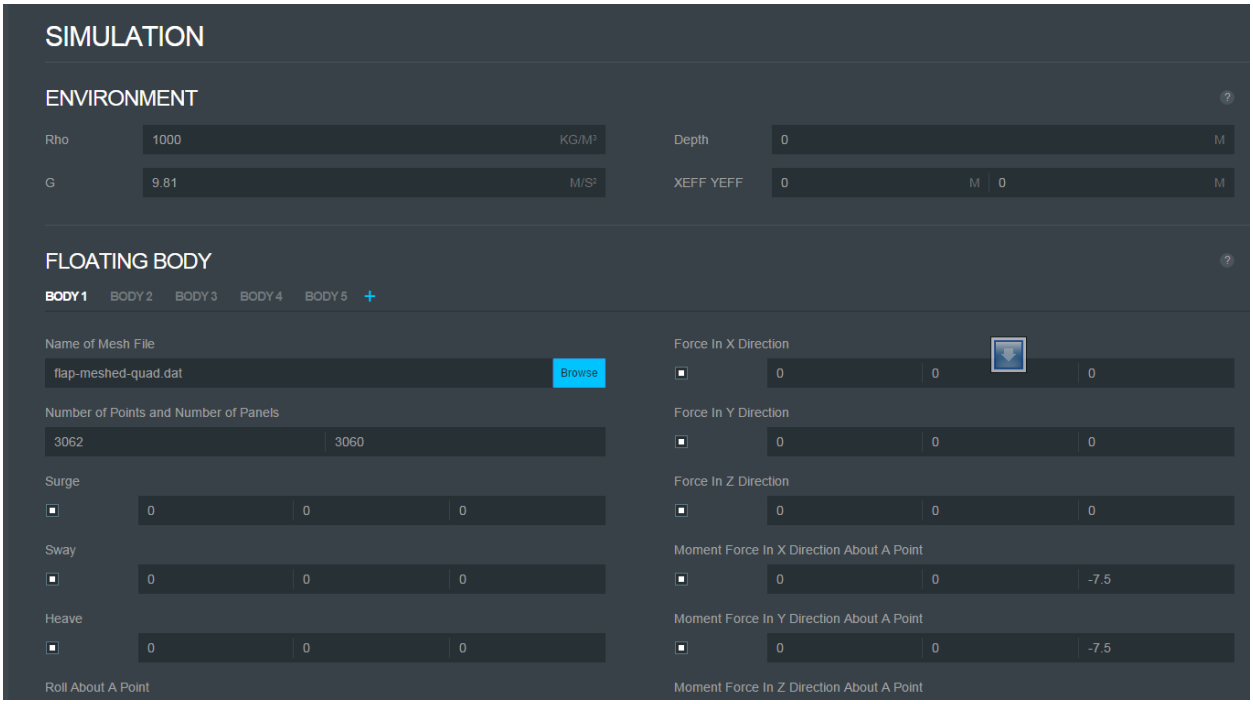
You may try entering some invalid input value to see that the application is able to recognize and report the error. For the example below, I input a value "**6**" for "Fineness" while it should be no larger than 1.

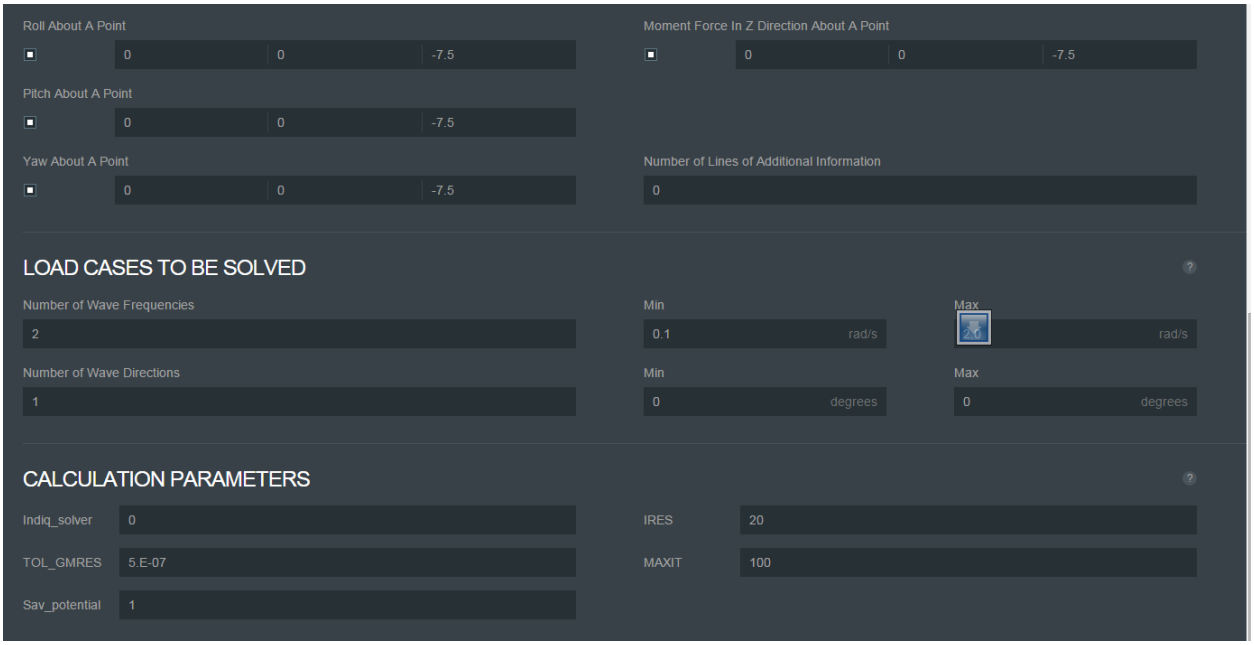


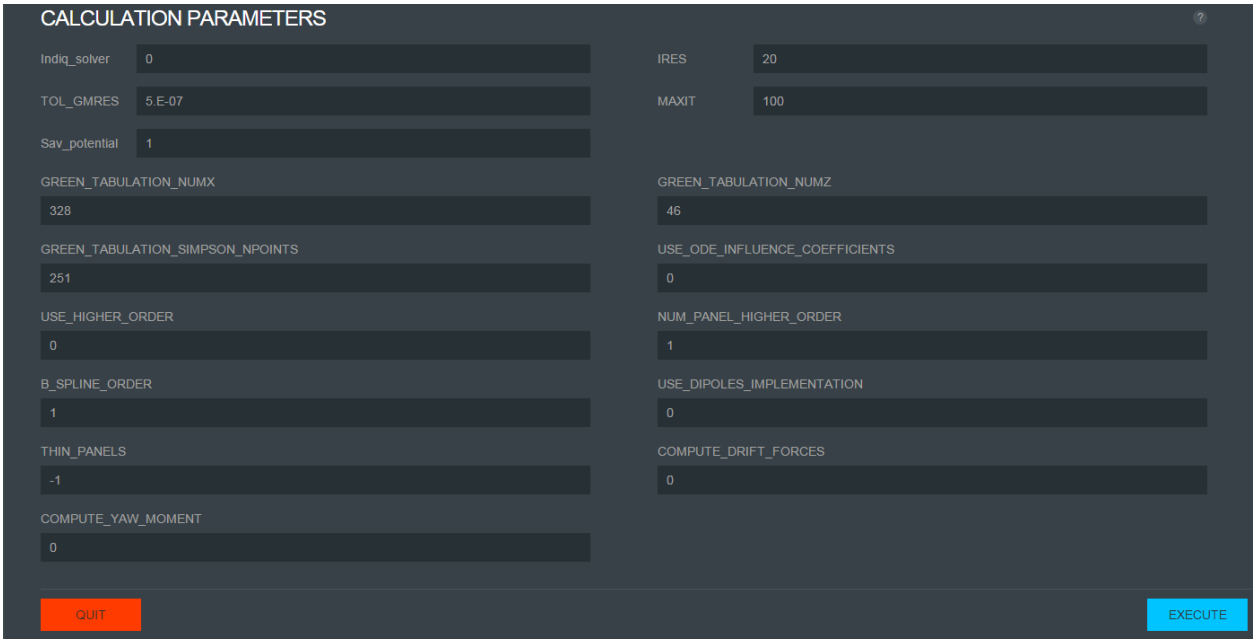
4.3 Simulation - Windows

Go to "SIMULATION" page. Please reference the configuration parameters in "**test_files/simulation/flap-meshed-quad/Nemoh.cal**" and "**test_files/simulation/flap-meshed-quad/input.txt**" to fill in the forms:

Note that most values are already pre-filled. All you need to fill is the path to the mesh file test_files/simulation/flap-meshed-quad/mesh/flap-meshed-quad.dat


SIMULATION
ENVIRONMENT
Rho
1000
KG/M³
Depth
0
M
G
9.81
M/S²
XEFF YEFF
0
M
0
M
FLOATING BODY
BODY 1
BODY 2
BODY 3
BODY 4
BODY 5
+
Name of Mesh File
flap-meshed-quad.dat
Browse
Force In X Direction
0
0
0
Number of Points and Number of Panels
3062
3060
Force In Y Direction
0
0
0
Surge
0
0
0
Force In Z Direction
0
0
0
Sway
0
0
0
Moment Force In X Direction About A Point
0
0
-7.5
Heave
0
0
0
Moment Force In Y Direction About A Point
0
0
-7.5
Roll About A Point
Moment Force In Z Direction About A Point

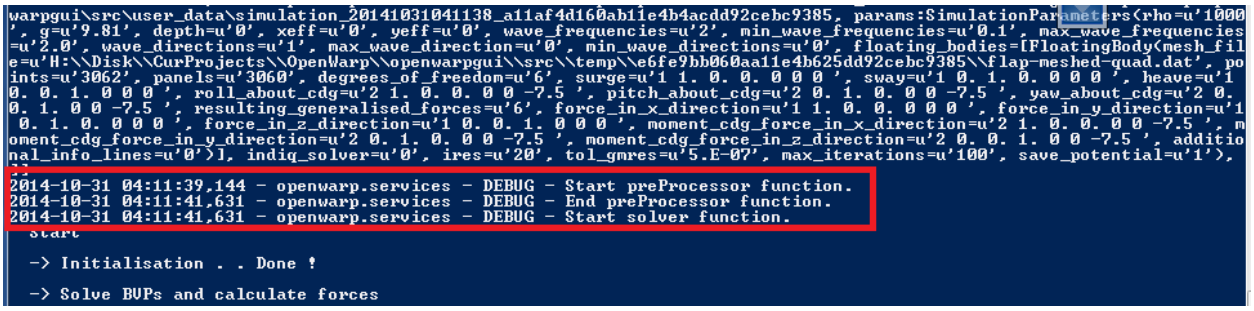

Roll About A Point
0
0
-7.5
Moment Force In Z Direction About A Point
0
0
-7.5
Pitch About A Point
0
0
-7.5
Yaw About A Point
0
0
-7.5
Number of Lines of Additional Information
0
LOAD CASES TO BE SOLVED
Number of Wave Frequencies
2
Min
0.1
rad/s
Max
2.6
rad/s
Number of Wave Directions
1
Min
0
degrees
Max
0
degrees
CALCULATION PARAMETERS
Indiq_solver
0
IRES
20
TOL_GMRES
5.E-07
MAXIT
100
Sav_potential
1



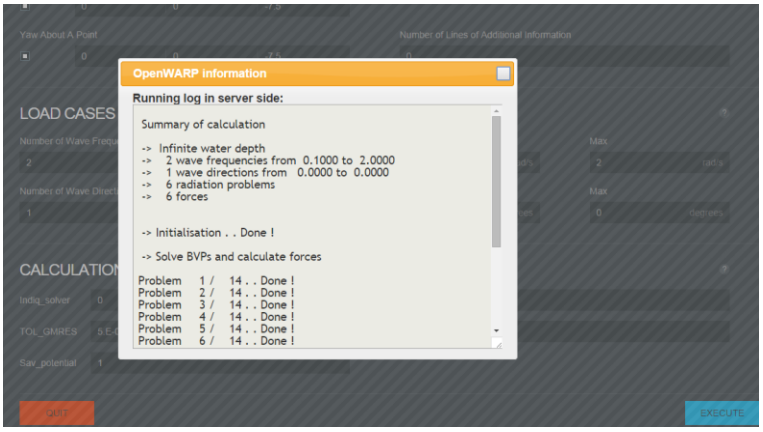
Click "**Browse**" button to choose the corresponding mesh file. E.g. "**test_files/simulation/flap-meshed-quad/mesh/flap-meshed-quad.dat**" in this submission.

*Note that the value **3062** (points) and **3060** (panels) are calculated and returned from server automatically after the mesh file were uploaded.. You do **NOT** need to input them manually.*

Then click the "**EXECUTE**" button to start simulation. You may check the python output to see that the "solver" process has started and not completed yet.



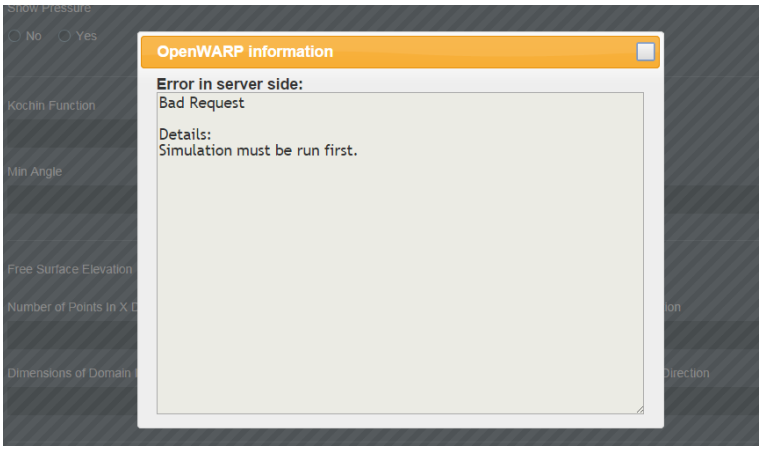
After solver finished working, you may check the running log information displayed within browser:



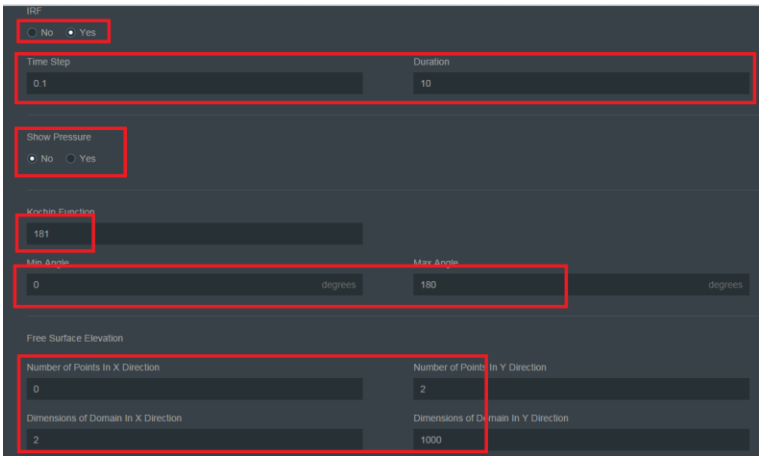
You may also try some invalid input values to see that browser will detect client side errors.

4.4 Postprocessing - Windows

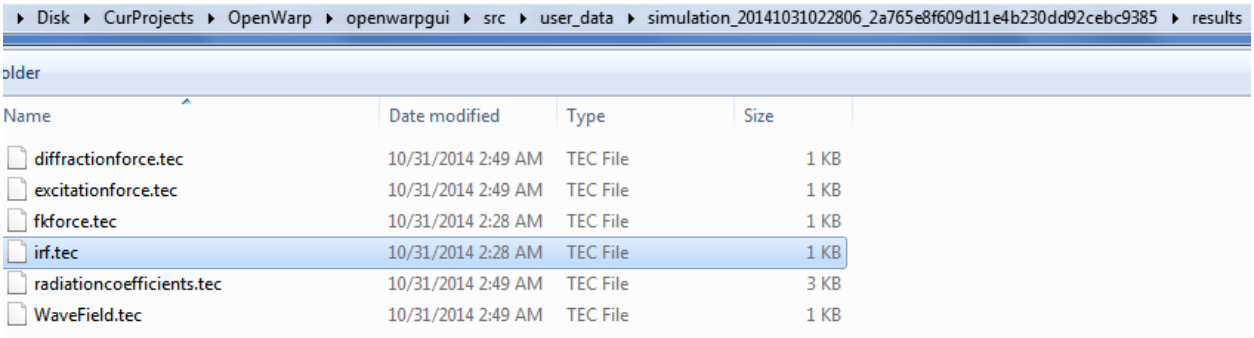
Navigate to "POSTPROCESSING" page and click "**SAVE AS TECPLOT**" button directly. If you did not run "**Simulation**" before, you will get following error dialog:



Ensure that you have ran simulation correctly as section 4.2 described.
Now configure the fields as below and then click "**SAVE AS TECPLOT**" button:
Note that all values are already pre-filled.



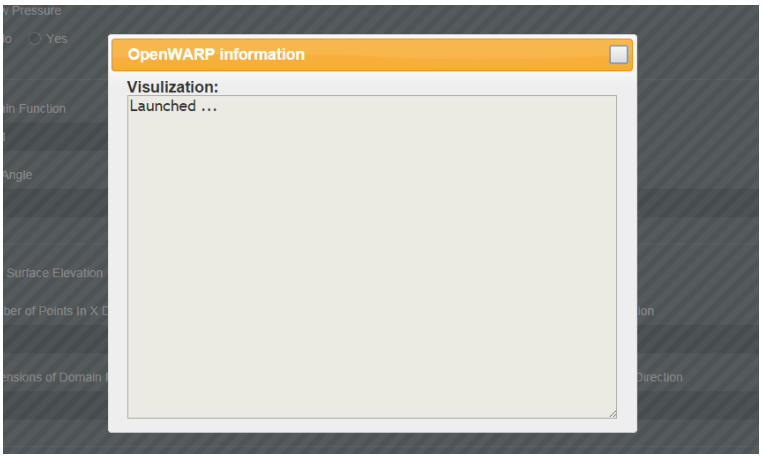
You will see the running log in server side displayed in a dialog. And find the simulation results from some folder like:
"**src/user_data/simulation_20140924011009_791b58f0434411e4bbb0000c2986ef35/results**".
This may depends since the running folder was generated randomly.

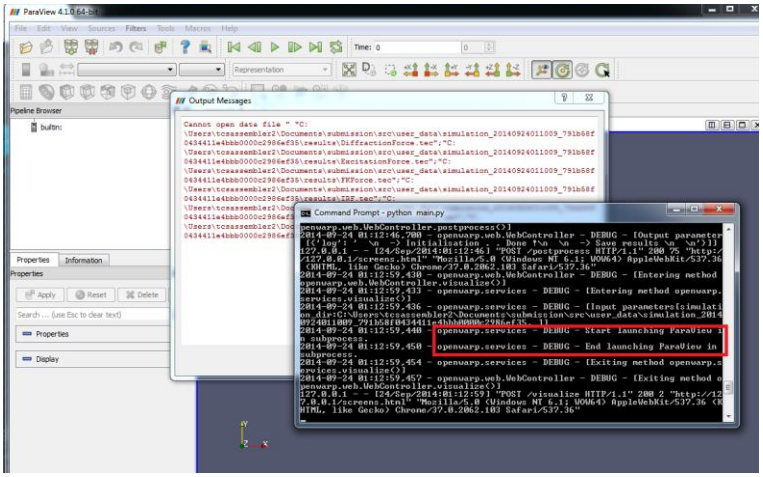


4.5 Visualization - Windows

Navigate to "POSTPROCESSING" page and click "**VISUALIZE IN PARAVIEW**" button directly. If you did not run "**Simulation**" before, you will get some error information like "Simulation must be run first." If you did not run "**SAVE AS TECPLOT**" before, you will get some error information like "SAVE AS TECPLOT must be run right after a successful simulation".

Ensure you've run "**Simulation**" and "**SAVE AS TECPLOT**" in order, now click "**VISUALIZE IN PARAVIEW**" button, you would see the success dialog and paraview started as a subprocess.

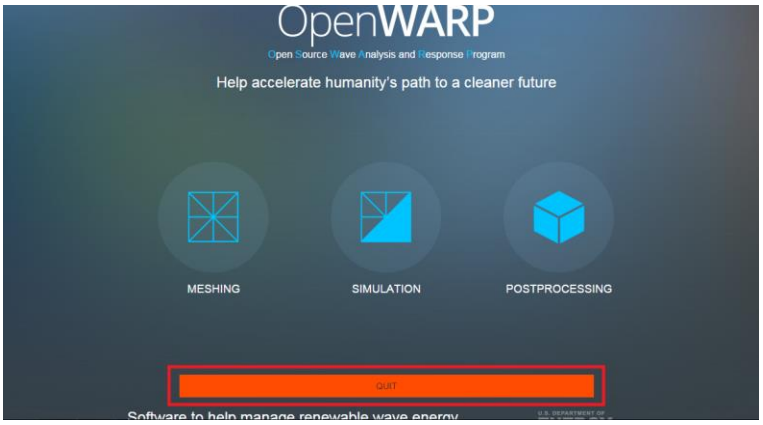


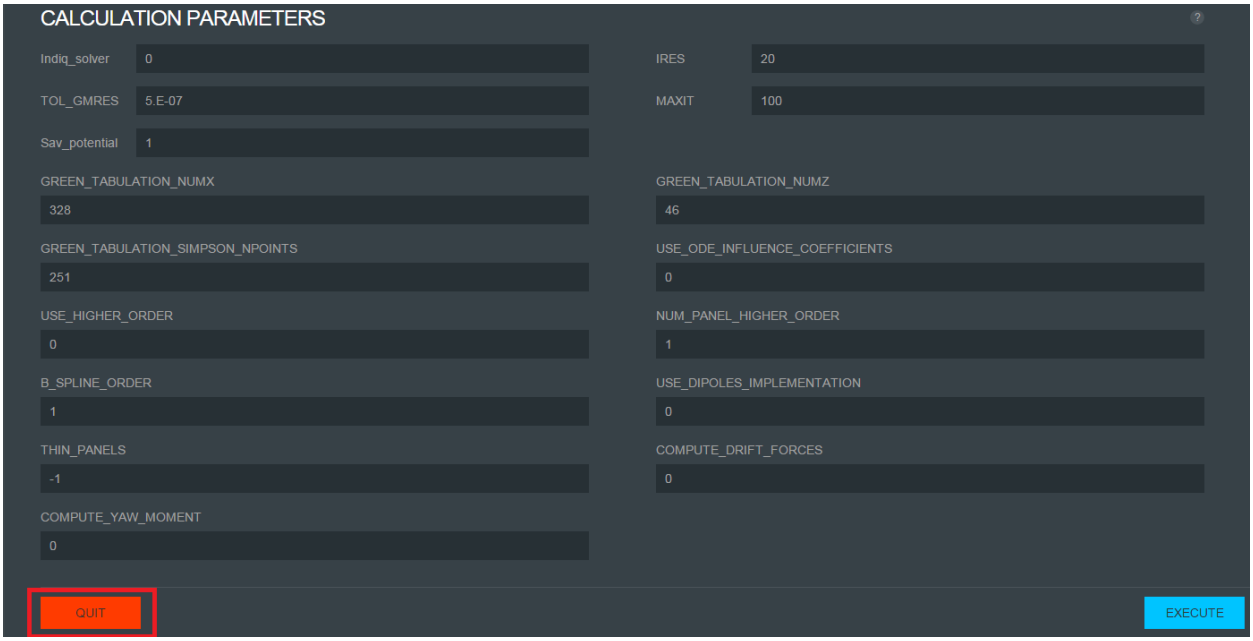


*Note that It's confusing that ParaView could not load and visualize the generated *.tec files correctly. Actually it could not load multiple *.tec files in one single launching. We need to have this fixed later. But this doesn't affect the whole logic of this assembly. The most important thing is that the ParaView was invoked successfully as the program desires.*

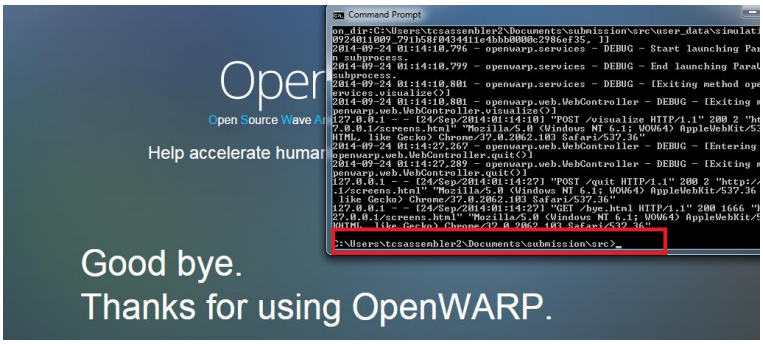
4.6 Quit - Windows

Now there's a QUIT button in **every** web page of the submission. See them as below:



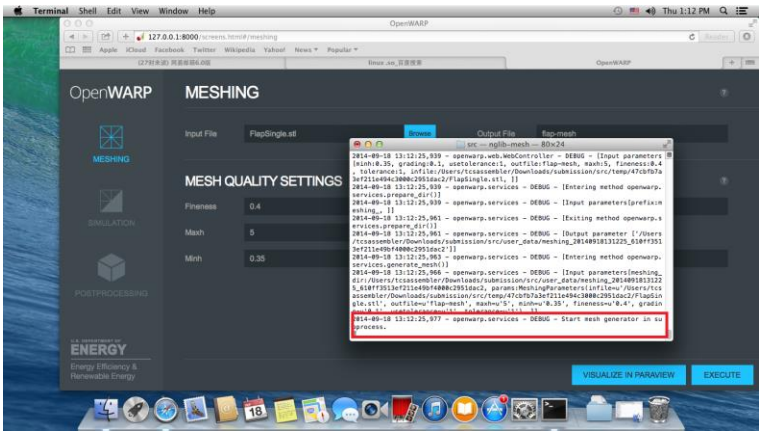


Try to click the "**QUIT**" button, you will be redirected to a "goodbye" page and check to see that the main program has been terminated.

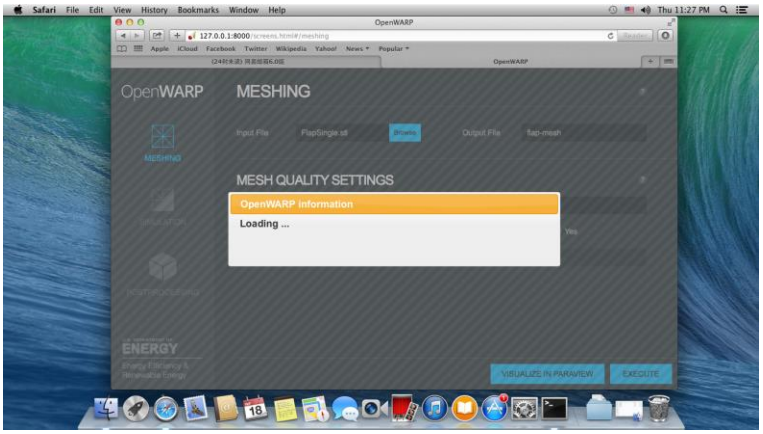


4.7 Mesh generator - MAC OS X

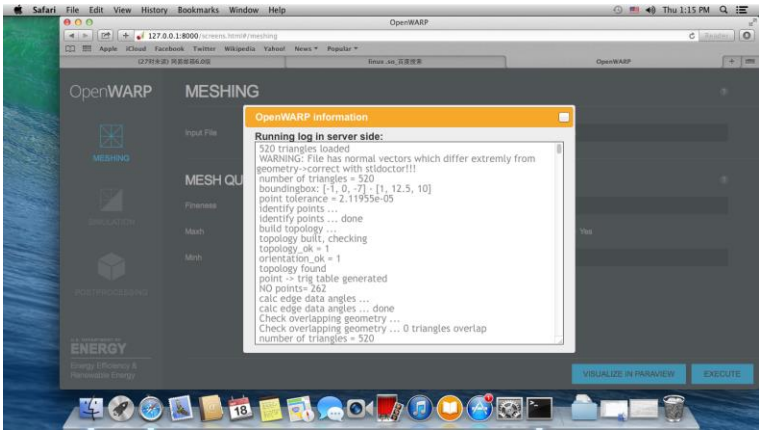
The operations on MAC OS X are all the same with that of Windows.
See the successful screenshots below:



The nglib-mesh was correctly invoked.



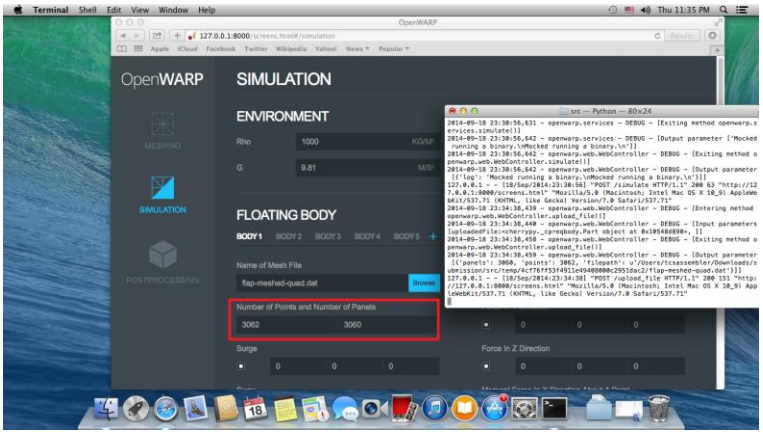
"**Loading ...**" dialog prevents user from clicking buttons and editing fields.



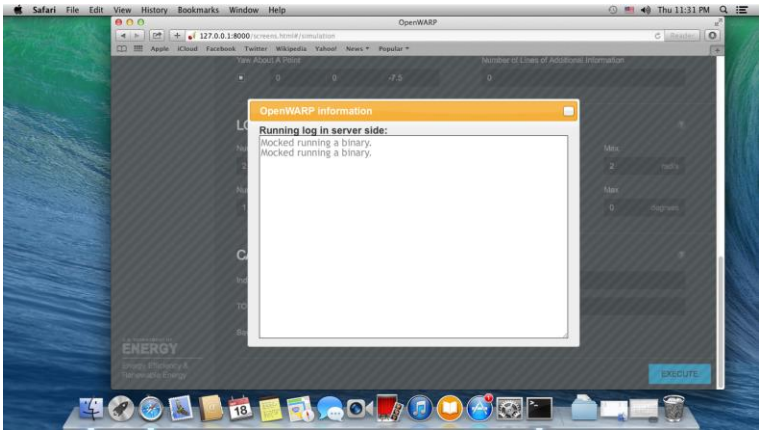
The correct result was logged and returned to client.

4.8 Simulation - MAC OS X

The operation on MAC OS X is all the same with that of Windows.
See the successful screen shots below:



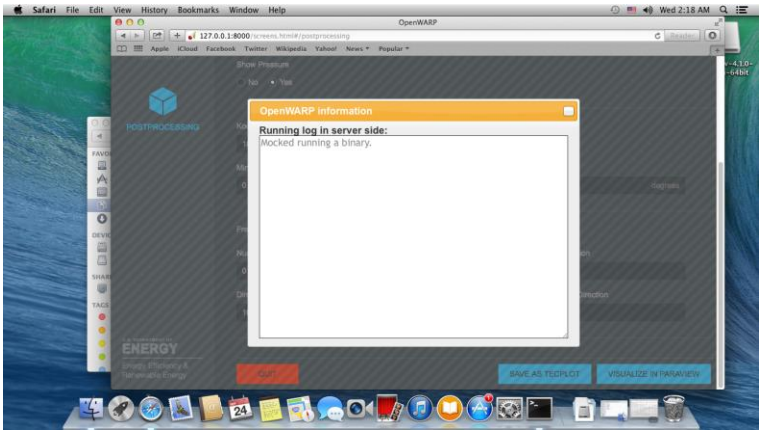
The mesh **points** and **panels** were determined by server and posted back to client.



Mocked Nemoh executables were invoked.

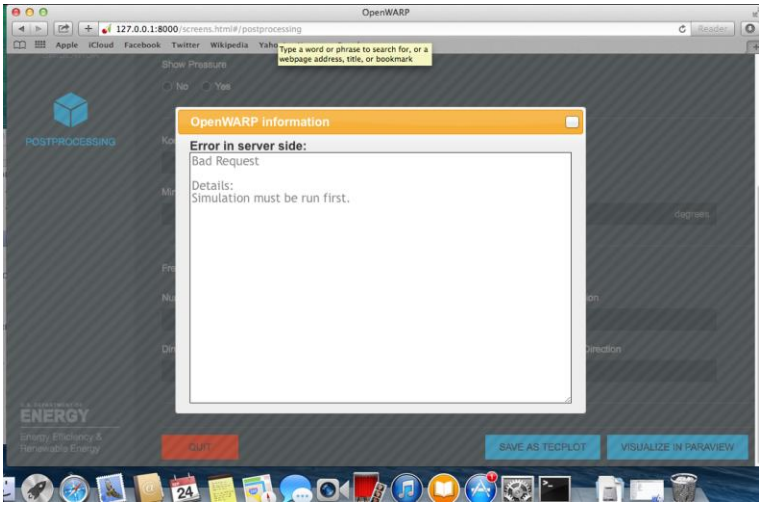
4.9 Postprocessing - MAX OS X

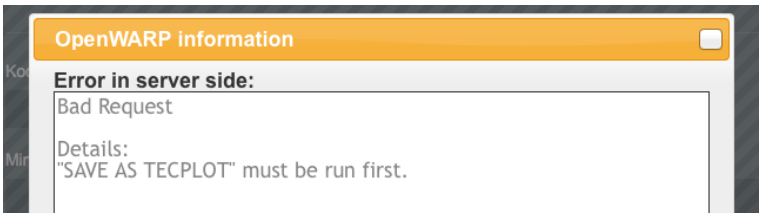
The operations on MAC OS X are all the same with those of Windows.
After you run "**Simulation**" correctly, go to page "POSTPROCESSING" and click "**SAVE AS TECPLOT**" button to see the output of the mocked **postProcessor**.



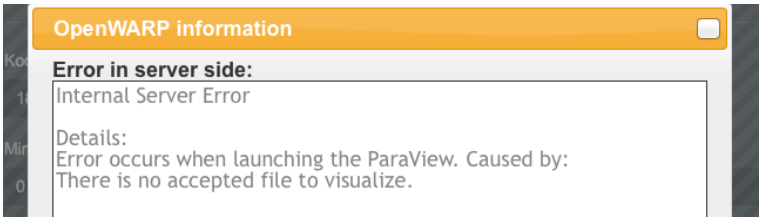
4.10 Visualization - MAC OS X

The operations on MAC OS X are all the same with those of Windows.
If the "**Simulation**" or "**SAVE AS TECPLOT**" has not been executed correctly, the error dialog would pop up to inform you.





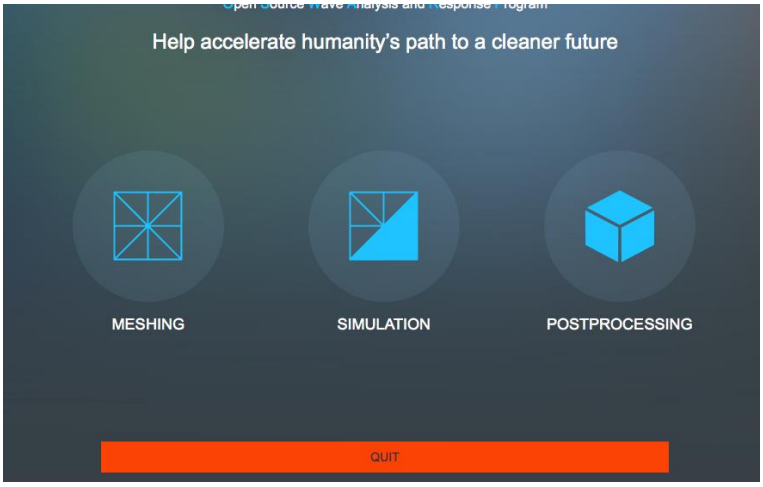
If there's no bad operations before, i.e. The simulation and postprocess has been run. Then we click "**VISUALIZE IN PARAVIEW**" button, a dialog will popup to warn you that there's no accepted file to visualize.



*This is because the **postProcessor** was mocked and there're no *.tec files in "Results" folder.*

4.11 Quit - MAC OS X

The operations on MAC OS X are all the same with those of Windows. See the screen shots below:



Click “QUIT” button to see the python program been terminated.

5. Resource Contact List

Name	Resource Email
TCSASSEMBLER	