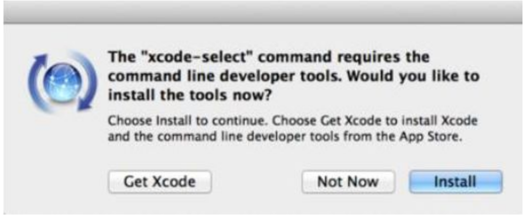
SETTING UP UG4 ON iMac

1. **DEVELOPER TOOLS:** Go to terminal and check first if xcode is installed by typing:

```
xcode-select --version
```

   If it is installed it will show the version number.
   If it is not installed it will prompt you to install as shown to the right, CLICK INSTALL.


The "xcode-select" command requires the command line developer tools. Would you like to install the tools now?

Choose Install to continue. Choose Get Xcode to install Xcode and the command line developer tools from the App Store.

2. **INSTALL BREW:** You will need brew to install cmake. Go to the website https://brew.sh/ and copy the command on the screen into terminal or type directly into terminal:

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

   this will install homebrew, homebrew is a package manager for macOS. To check that it is installed type in the terminal: *brew --version*

3. **LOADING UGHUB:** Now go to the website: https://github.com/UG4/ughub this will give the directions to install ughub onto the computer. I will outline the steps below
   a) Please clone the github repository, e.g. by executing the following commands:

```
cd $HOME
git clone https://github.com/UG4/ughub
```

   b) This will download ughub to *$HOME/ughub*. You may of course clone the repository to any other location. Simply replace *$HOME/ughub* with your path for the remainder of this introduction.
   c) Please make sure to update your ughub installation from time to time by executing

```
cd $HOME/ughub
git pull
```

   This will update your ughub repository on your computer.

4. **CMAKE:** In terminal type *brew install cmake.* This will install cmake, check to see that it is installed by typing *cmake --version*, if it is installed it will tell you what version is installed.

5. **BASH_PROFILE:** Now we set up the bash_profile, this is a text file which tells the computer to open/run applications in any directory:
   a) Type *cd $HOME* to go to to your home direction.
   b) Type *vim .bash_profile*, this will create a file for you to edit

c) In order to execute ughub from any path, you should either edit your PATH environment variable by adding the following line to your $HOME/.bash_profile or $HOME/.bashrc file: *export PATH=$PATH:$HOME/ughub*

d) Close the window and then type in terminal: *source $HOME/.bash_profile*

6. **INSTALLING UG4:** ughub serves as a package management tool for the UG4 simulation framework. Use the following command to get some help on ughub's usage: *ughub help*

a) One typically starts by creating a directory into which all UG4 related packages shall be installed (again, the exact path is arbitrary):

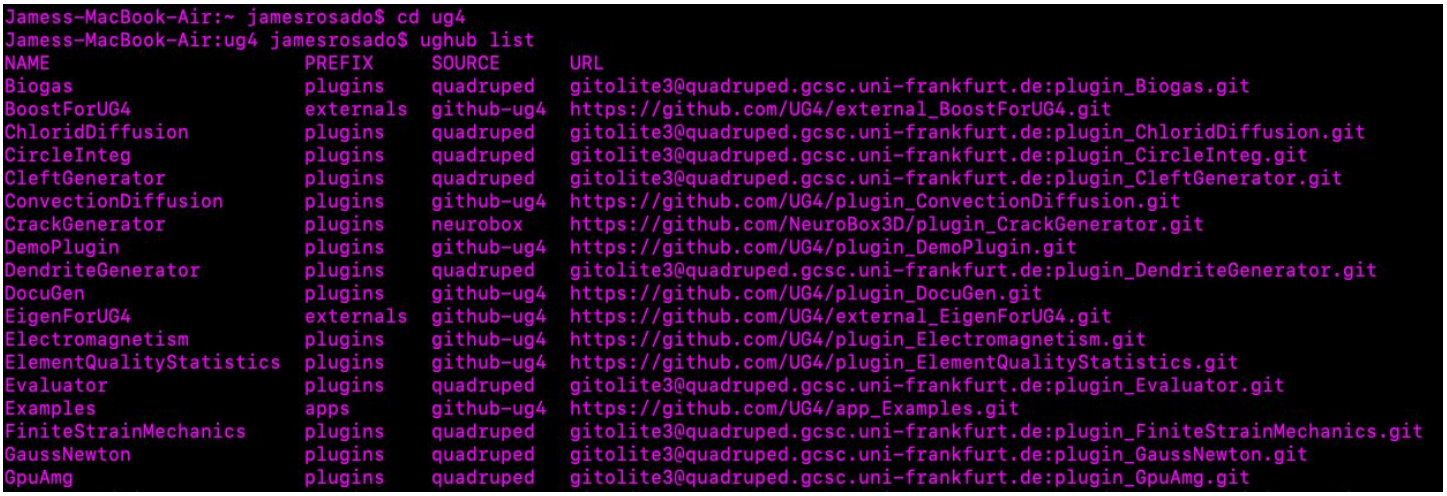
cd $HOME
mkdir ug4

b) Now we can initialize this directory for usage with ughub:

cd $HOME/ug4
ughub init

This creates a local CMakeLists.txt file and creates a hidden folder '.ughub' in which package-sources are maintained. All further calls to ughub from within the directory tree located at $HOME/ug4 will now relate to this main directory.

c) To view a list of available packages, execute: *ughub list*



```
Jamess-MacBook-Air:~ jamesrosado$ cd ug4
Jamess-MacBook-Air:ug4 jamesrosado$ ughub list
NAME                      PREFIX     SOURCE      URL
Biogas                    plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_Biogas.git
BoostForUG4               externals  github-ug4  https://github.com/UG4/external_BoostForUG4.git
ChloridDiffusion          plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_ChloridDiffusion.git
CircleInteg               plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_CircleInteg.git
CleftGenerator            plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_CleftGenerator.git
ConvectionDiffusion       plugins    github-ug4  https://github.com/UG4/plugin_ConvectionDiffusion.git
CrackGenerator            plugins    neurobox    https://github.com/NeuroBox3D/plugin_CrackGenerator.git
DemoPlugin                plugins    github-ug4  https://github.com/UG4/plugin_DemoPlugin.git
DendriteGenerator         plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_DendriteGenerator.git
DocuGen                   plugins    github-ug4  https://github.com/UG4/plugin_DocuGen.git
EigenForUG4               externals  github-ug4  https://github.com/UG4/external_EigenForUG4.git
Electromagnetism          plugins    github-ug4  https://github.com/UG4/plugin_Electromagnetism.git
ElementQualityStatistics  plugins    github-ug4  https://github.com/UG4/plugin_ElementQualityStatistics.git
Evaluator                 plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_Evaluator.git
Examples                  apps       github-ug4  https://github.com/UG4/app_Examples.git
FiniteStrainMechanics     plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_FiniteStrainMechanics.git
GaussNewton               plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_GaussNewton.git
GpuAmg                    plugins    quadruped   gitolite3@quadruped.gcsc.uni-frankfurt.de:plugin_GpuAmg.git
```

This is just a sample snippet of the packages that maybe available, your list may be different. Or you can execute '*ughub listpackages*'. Which and how many packages are displayed depends on the registered package-sources. To view all registered sources execute.

ughub listsources

To install a package, simply execute 'ughub install' with one or more of the available packages: *ughub install Examples*
Dependent packages will now be automatically installed.

7. **COMPILING:** Before compiling we need to update our bash_profile so that we can run ugshell from any directory.

   a) Type *vim .bash_profile* and add the following line:

   ```
   source $HOME/ug4/ugcore/scripts/shell/ugbash
   ```

   b) Then reload your .bashrc or .bash_profile files, e.g.: *source $HOME/.bash_profile*

   c) Then we setup the build directory:

   ```
   cd $HOME/ug4
   mkdir build
   cd build
   ```

   d) This next step always make sure you are in the build folder!

   ```
   cmake -DCMAKE_BUILD_TYPE=Release ..
   ```

   You may also activate or deactivate all installed plugins at once:

   ```
   cmake -DENABLE_ALL_PLUGINS=ON .
   cmake -DDISABLE_ALL_PLUGINS=ON .
   ```

   e) Then type: *make -j2*

   This will take a while 20~30 minutes to build for the first time.

   f) When complete type *ugshell* and it should access the shell for running commands with the ug4 framework.

8. **RUNNING EXAMPLES:** Starting from *UG4's* root directory, please execute the following,

```
mkdir runs
cd runs
ugshell -ex Examples/poisson.lua
ugshell -ex Examples/poisson.lua -dim 3
ugshell -ex Examples/solmech.lua
ugshell -ex Examples/elder_adapt.lua
ugshell -ex Examples/navier_stokes.lua
ugshell -ex Examples/electromagnetism_pan.lua -numRefs 3
```

The first line is to make the ‘runs’ direction, ‘cd runs’ takes you to that directory. The remaining six commands execute Example scripts of simulations.

9. **INSTALLING PROMESH AND PARAVIEW**
    a. See this link: https://www.paraview.org/download/
    b. See this link: http://www.promesh3d.com/downloads/
10. **INSTALL LUA**
    a. Type in terminal: brew install lua
    b. Then type: lua and you will be in compiler. Press control + D to exit

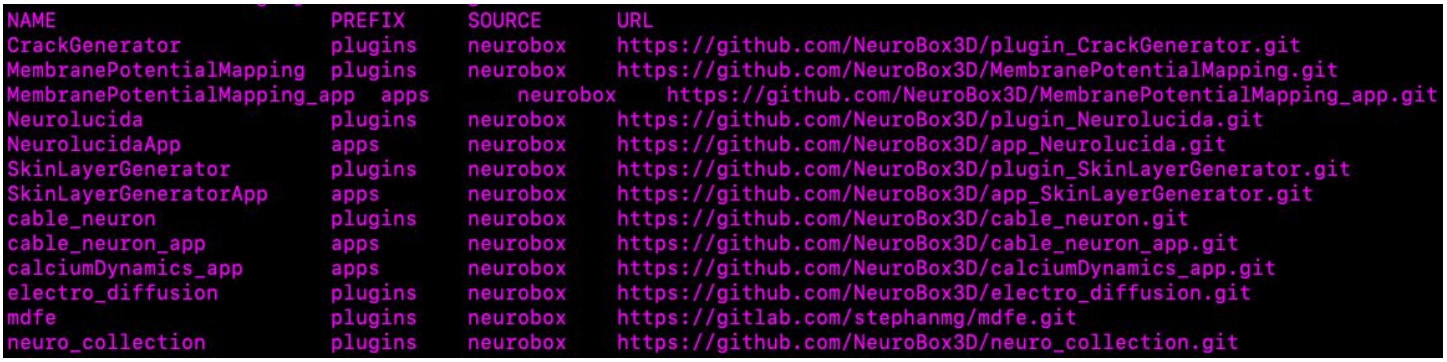
SETTING UP NEURO-BOX PLUGINS

11. Go to the ug4 root directory: *cd $HOME/ug4*

Execute: `ughub addsource neurobox https://github.com/NeuroBox3D/neurobox-packages.git`

This will add the neurobox repositories to your local drive.

12. Execute: `ughub list neurobox`



```
NAME                      PREFIX     SOURCE      URL
CrackGenerator            plugins    neurobox    https://github.com/NeuroBox3D/plugin_CrackGenerator.git
MembranePotentialMapping  plugins    neurobox    https://github.com/NeuroBox3D/MembranePotentialMapping.git
MembranePotentialMapping_app  apps       neurobox    https://github.com/NeuroBox3D/MembranePotentialMapping_app.git
Neurolucida               plugins    neurobox    https://github.com/NeuroBox3D/plugin_Neurolucida.git
NeurolucidaApp            apps       neurobox    https://github.com/NeuroBox3D/app_Neurolucida.git
SkinLayerGenerator        plugins    neurobox    https://github.com/NeuroBox3D/plugin_SkinLayerGenerator.git
SkinLayerGeneratorApp     apps       neurobox    https://github.com/NeuroBox3D/app_SkinLayerGenerator.git
cable_neuron              plugins    neurobox    https://github.com/NeuroBox3D/cable_neuron.git
cable_neuron_app          apps       neurobox    https://github.com/NeuroBox3D/cable_neuron_app.git
calciumDynamics_app       apps       neurobox    https://github.com/NeuroBox3D/calciumDynamics_app.git
electro_diffusion         plugins    neurobox    https://github.com/NeuroBox3D/electro_diffusion.git
mdfe                      plugins    neurobox    https://gitlab.com/stephanmg/mdfe.git
neuro_collection          plugins    neurobox    https://github.com/NeuroBox3D/neuro_collection.git
```

You want to install the plugins

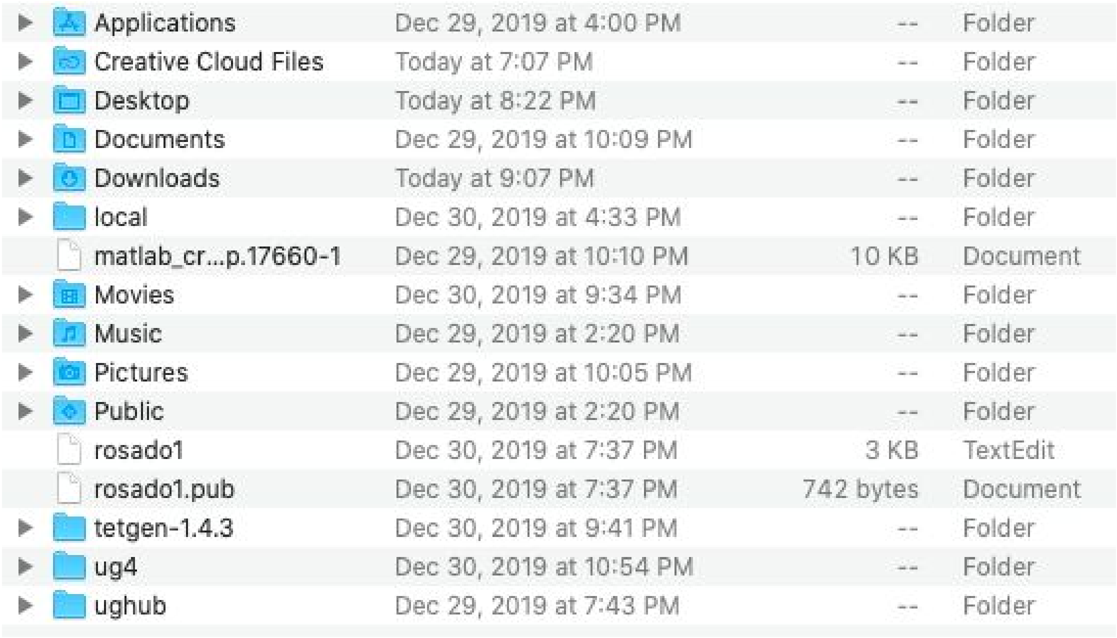
- neuro_collection
- cable_neuron
- cable_neuron_app
- calciumDynamics_app
- MembranePotentialMapping
- MembranePotentialMapping_app
- Neurolucida
- NeurolucidaApp
- SkinLayerGenerator (not necessary)
- SkinLayerGeneratorApp (not necessary)

By executing: `ughub install <plugin name>`

For plugins you need to be in the ‘plugins’ folder, for apps you need to be in the ‘apps’ folder.

13. **INSTALLING TETGEN**:

a) Download the tetgen zip file from http://wias-berlin.de/software/tetgen/tetgen143.html and unzip it into the same directory as UG4. The $HOME directory should look something like this:



▶ Applications	Dec 29, 2019 at 4:00 PM	--	Folder
▶ Creative Cloud Files	Today at 7:07 PM	--	Folder
▶ Desktop	Today at 8:22 PM	--	Folder
▶ Documents	Dec 29, 2019 at 10:09 PM	--	Folder
▶ Downloads	Today at 9:07 PM	--	Folder
▶ local	Dec 30, 2019 at 4:33 PM	--	Folder
matlab_cr...p.17660-1	Dec 29, 2019 at 10:10 PM	10 KB	Document
▶ Movies	Dec 30, 2019 at 9:34 PM	--	Folder
▶ Music	Dec 29, 2019 at 2:20 PM	--	Folder
▶ Pictures	Dec 29, 2019 at 10:05 PM	--	Folder
▶ Public	Dec 29, 2019 at 2:20 PM	--	Folder
rosado1	Dec 30, 2019 at 7:37 PM	3 KB	TextEdit
rosado1.pub	Dec 30, 2019 at 7:37 PM	742 bytes	Document
▶ tetgen-1.4.3	Dec 30, 2019 at 9:41 PM	--	Folder
▶ ug4	Dec 30, 2019 at 10:54 PM	--	Folder
▶ ughub	Dec 29, 2019 at 7:43 PM	--	Folder

(disregard rosado1 files for now)

b) Then type the following in terminal:

cd <TETGEN_DIR> && mkdir build && cd build && cmake .. && make

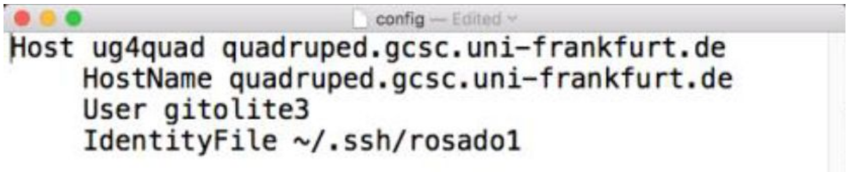
this should complete very quickly

c) This will build the TetGen library. When that is finished (should be quite fast) change to your UG4 build folder and reconfigure and rebuild using:

cmake -DTETGEN=<TETGEN_DIR> ..
make -j4
This will take ~20-30 minutes

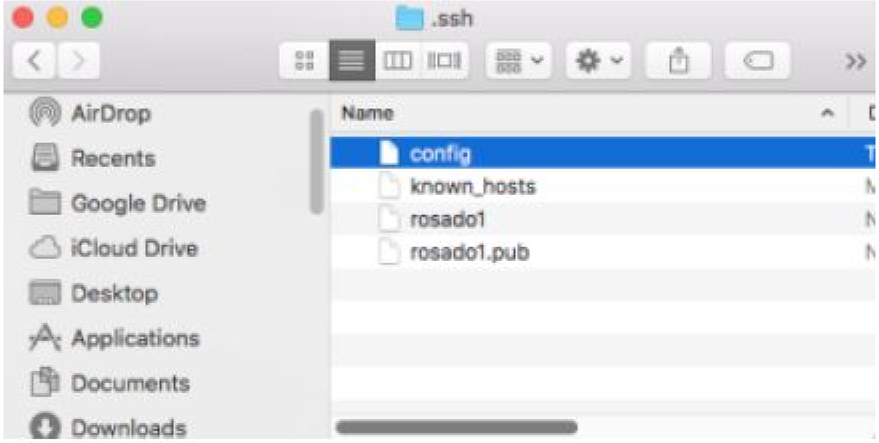
14. **INSTALLING SUPERLU & PARMETIS:** For this part you will need the **rosado1.pub** and **rosado1** files in order to access the packages at uni-frankfurt.de. (CONTACT James for this)

a) Create an SSH alias for the server to make further access easier. You may need to create the hidden .ssh folder in the $HOME folder. Create a file ~/.ssh/config (or simply open it if it already exists) and save the entry shown to the right.



```
Host ug4quad quadruped.gcsc.uni-frankfurt.de
    HostName quadruped.gcsc.uni-frankfurt.de
    User gitolite3
    IdentityFile ~/.ssh/rosado1
```

b) The .ssh folder should have in it the rosado1 files and config (disregard known_hosts):



config
known_hosts
rosado1
rosado1.pub

c) Next you have to set proper access permissions at least to the private key file (ssh is quite restrictive), go back to $HOME directory and execute:

```
chmod 600 ~/.ssh/rosado1
```

Where rosado1 is the private key.

d) To be on the safe side, also set the following access permissions:

```
chmod 644 ~/.ssh/rosado1.pub
chmod 700 ~/.ssh
```

e) Now we can add the quadruped source and install SuperLU and Parmetis. Go back to $HOME/ug4/plugins and execute:

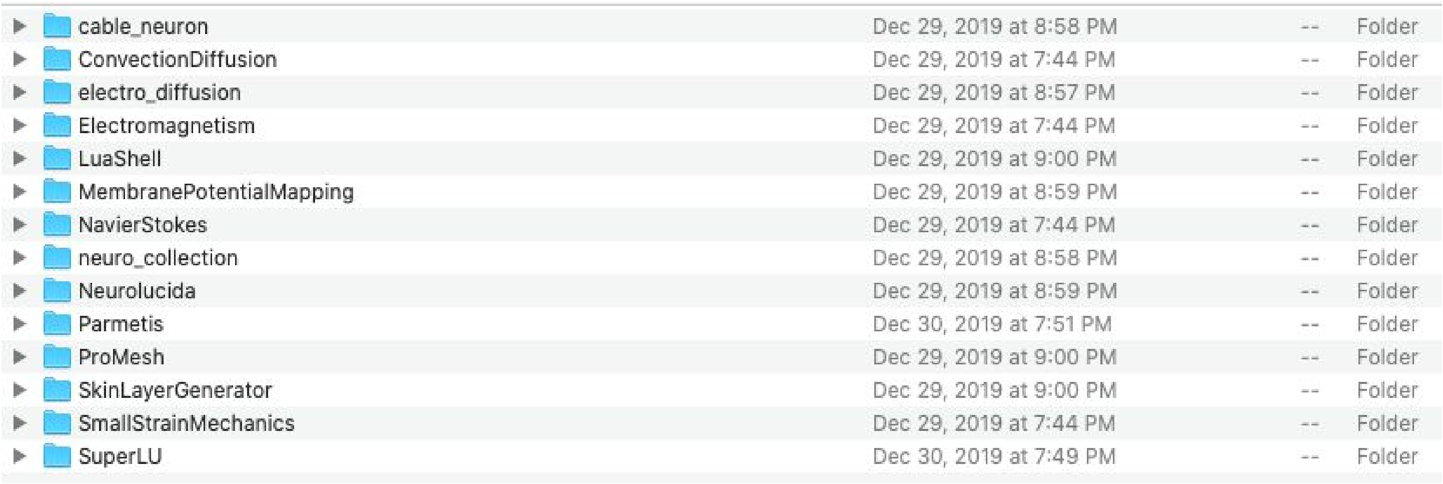
```
ughub addsource quadruped ug4quad:ug4-packages
```

It may ask for the password: ________________________ (contact James for this)

Then execute

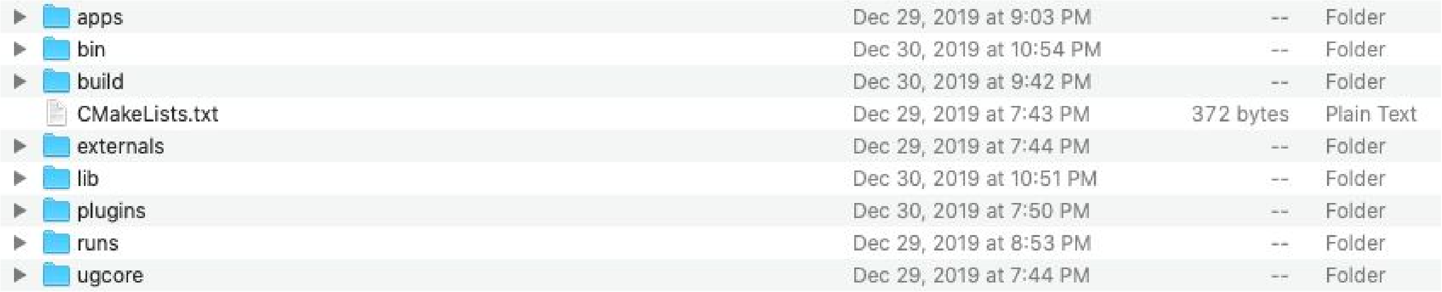
```
ughub install SuperLU Parmetis
```

The plugins folder should have:



cable_neuron	Dec 29, 2019 at 8:58 PM	--	Folder
ConvectionDiffusion	Dec 29, 2019 at 7:44 PM	--	Folder
electro_diffusion	Dec 29, 2019 at 8:57 PM	--	Folder
Electromagnetism	Dec 29, 2019 at 7:44 PM	--	Folder
LuaShell	Dec 29, 2019 at 9:00 PM	--	Folder
MembranePotentialMapping	Dec 29, 2019 at 8:59 PM	--	Folder
NavierStokes	Dec 29, 2019 at 7:44 PM	--	Folder
neuro_collection	Dec 29, 2019 at 8:58 PM	--	Folder
Neurolucida	Dec 29, 2019 at 8:59 PM	--	Folder
Parmetis	Dec 30, 2019 at 7:51 PM	--	Folder
ProMesh	Dec 29, 2019 at 9:00 PM	--	Folder
SkinLayerGenerator	Dec 29, 2019 at 9:00 PM	--	Folder
SmallStrainMechanics	Dec 29, 2019 at 7:44 PM	--	Folder
SuperLU	Dec 30, 2019 at 7:49 PM	--	Folder

The ug4 folder should look like this:



apps	Dec 29, 2019 at 9:03 PM	--	Folder
bin	Dec 30, 2019 at 10:54 PM	--	Folder
build	Dec 30, 2019 at 9:42 PM	--	Folder
CMakeLists.txt	Dec 29, 2019 at 7:43 PM	372 bytes	Plain Text
externals	Dec 29, 2019 at 7:44 PM	--	Folder
lib	Dec 30, 2019 at 10:51 PM	--	Folder
plugins	Dec 30, 2019 at 7:50 PM	--	Folder
runs	Dec 29, 2019 at 8:53 PM	--	Folder
ugcore	Dec 29, 2019 at 7:44 PM	--	Folder

15. **COMPILING EVERYTHING:** Now we compile everything, if a package is not installed then it will show errors and won't compile. The order of compiling will also affect if there are errors.

a) First compile SuperLU and Parmetis, go to $HOME/ug4/build and execute:

```
cmake -DCPU="1" -DDIM="2;3" -DPARALLEL=ON -DPCL_DEBUG_BARRIER=ON
-DEMBEDDED_PLUGINS=ON
-DUSE_LUA2C=ON -DParmetis=ON -DSuperLU=ON ..
```

Make sure you have the “DPARALLEL=ON” if not it will not compile and generate errors. This will take a while. Then execute *make -j4*

b) Then execute:

```
cmake -DConvectionDiffusion=ON -DcalciumDynamics=ON -Dneuro_collection=ON
-Dcable_neuron=ON -DMembranePotentialMapping=ON -DNeurolucida=ON ..
```

Then execute *make -j4*

c) When everything is compiled (hopefully without error) you are all set to test the execution of a script:

```
cdug bin
mkdir -p test/grid test/meas test/vtk
ugshell -ex calciumDynamics_app/spine/spine_dg.lua -outName test -tstep 0.0001 -endTime
0.002 -vtk -pstep 0.0001
```

This will start an example simulation of a dendritic spine that is chemically activated by a synaptic Ca2+ influx. The spine geometry is created according to some parameters set in the script and saved in test/grid.