**Step 1:**

+ Launch an Ubuntu 18.04 server on AWS.

**Step 2 (cmake):**

Install cmake: sudo apt-get update -y && sudo apt-get install cmake -y

Verify cmake installation: cmake –version

**Step 3 (ughub):**

1. Clone the Github repo:

Cd $HOME

Git clone <https://github.com/UG4/ughub>

1. Update ughub with:

cd $HOME

Git pull

**Step 4 (bash\_profile):**

1. For Ubuntu, every time a shell is loaded upon logging in. The file that the shell looks for is ~/.profile. So we will set up ughub bin path in this file.
2. Edit ~/.profile with the editor of your choice or you can use vim with: vim ~/.profile
3. Add the following line to your ~/.profile file: export PATH=$PATH:$HOME/ughub
4. Close it and type: source $HOME/.bash\_profile
5. Verify ughub is visible in your current shell by typing: which ughub and this should returns the executable path for ughub back to you.

**Step 5 (installing ug4):**

1. Cd $HOME

Mkdir ug4

1. Cd $HOME/ug4

Ughub init

1. We will need to install some example packages:

Ughub install Examples

**Step 6 (compiling ug4):**

1. Add the following line into your ~/.profile file: source $HOME/ug4/ugcore/scripts/shell/ugbash
2. Type: source ~/.profile
3. To verify that ugbash shell executable path is visible, type: which ugbash and it should returns ugbash path back to you.
4. Set up build directory:

Cd $HOME/ug4

Mkdir build

Cd build

1. Install important C-based code building software: sudo apt-get install build-essential or this will throws error when you run the cmake command because some of the directives in the makefile will attempt to look for C-compilers like gcc, g++.
2. Run cmake to create a makefile (cmake is simply a command that looks in the CMakeLists.txt file to create a makefile) and the .. in this command means that the CMakeLists.txt file is one directory above or backwards: make -DCMAKE\_BUILD\_TYPE=Release -DENABLE\_ALL\_PLUGINS=ON ..
3. Type: make -j4 (it will take about 20 to 30 minutes to build)
4. Type: ugshell and you can access the shell with ug4 framework.

**Step 7 (running examples):**

1. Cd $HOME/ug4
2. 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

**Step 8 (install lua):**

1. Simply type: sudo apt-get update -y && sudo apt install lua
2. This however will not automatically install the latest version of lua so it will returns a list of available lua installations and simply rerun the command again with the lua installation of your choice.
3. During the time of this documentation, my lua installation was: sudo apt install lua5.3

**Step 9 (set up neuro-box plugins):**

1. cd $HOME/ug4
2. Type: ughub addsource neurobox <https://github.com/NeuroBox3D/neurobox-packages.git>
3. Install the following plugins: ughub install neuro\_collection cable\_neuron cable\_neuron\_app calciumDynamics\_app MembranePotentialMapping MembranePotentialMapping\_app Neurolucida NeurolucidaApp
4. You can also install the optional SkinLayerGenerator and SkinLayerGeneratorApp as well.
5. This will add plugins and apps to your respective plugins/ and app/ directories in your ug4 root directory

**Step 10 (install tetgen):**

1. Go on tetgen website (<https://www.berlios.de/software/tetgen/>)and visit the download section to get the latest package.
2. When you get to the page where it lets you download either zip or tar files, make sure to download the file onto your local computer and use either rsync (if you are on Mac) or Windows scp to upload it to your ubuntu server on the cloud.
3. For this tutorial, I will use rsync to upload it to the cloud.
4. To make things easier, set up your server to allow password authentication instead of private key. Type: sudo vim /etc/ssh/sshd\_config and edit the line that says Password Authentication no to PasswordAuthentication yes
5. Afterwards, save the file and type: sudo service ssh restart.
6. To set up a password for your user, type: sudo passwd ubuntu
7. After setting up the password, go onto your local computer and type: rsync -arv /path/to/your/zipFileofTetgen ubuntu@IpAddress:~ (if you download it and did not move it, the path would be ~/Download/tetgen1.5.1.zip or it might have a different version number than what is shown here).
8. This will upload it to your ubuntu server’s home directory.
9. Make sure to either tar -xvfv or unzip. I will use unzip in this tutorial so type: unzip tetgen1.5.1.zip
10. But if unzip is not installed, simply type: sudo apt-get install unzip -y
11. To compile, you must first edit the makefile, so: cd $HOME/tetgenDir
12. Type: vim makefile
13. Look for 2 variables that says CXXFLAGS and PREDCXXFLAGS and add this option: -fPIC

So it would looks like: CXXFLAGS = -O3 -fPIC and PREDCXXFLAGS = -O0 -fPIC

1. This option is extremely critical for Ubuntu’s architecture.
2. Then type: make && make tetlib.
3. According this source (<http://wias-berlin.de/software/tetgen/compile.html>), make simply creates an executable and make tetlib creates the tetgen static library that will be linked when compiling with ug4.
4. Go back to ug4 build directory: cd $HOME/ug4/build
5. Type: cmake -DTETGEN=~/tetgenDir ..
6. This will create a new makefile that will link tetgen.a library to be re-compiled with ug4.
7. Type: make -j4 (this will take another 20-30 minutes)

**Step 11 (installing superlu and parmetis):**

1. Create an SSH alias for access to packages, by creating a file ~/.ssh/config and type in the following:

Host ug4quad quadruped.gcsc.uni-frankfurt.de

HostName quadruped.gcsc.uni-frankfurt.de

User gitolite3

IdentityFile ~/.ssh/rosado

1. Ask James for rosado and rosado.pub file.
2. Put these files in ~/.ssh directory.
3. You must set the right permission for these 2 key files, so type:

Chmod 600 ~/.ssh/rosado

Chmod 644 ~/.ssh/rosado.pub

Chmod 700 ~/.ssh

1. Cd $HOME/ug4/plugins
2. Type this to install superlu and parmetis: ughub addsource quadruped ug4quad:ug4-packages && ughub install SuperLU Parmetis
3. It will ask for a password, ask James for this.

**Step 12 (compiling everything):**

1. Make sure to install mpi before anything else:

sudo apt-get update -y && sudo apt install libopenmpi-dev -y && mpirun --version

1. Compile SuperLu and Parmetis by going 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 ..
2. Remember to have DPARALLEL=ON, because it uses the wrapper to compile with mpi.
3. Then run: make -j4
4. Then execute: cmake -DConvectionDiffusion=ON -DcalciumDynamics=ON -Dneuro\_collection=ON -D cable\_neuron=ON -DMembranePotentialMapping=ON -DNeurolucida=ON ..
5. Then run: make -j4
6. Afterwards, do the following to run a example simulation:

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

It 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.

This is with the help of James’ guide to install UG framework on a Mac as well.