Note: The following has been tested successfully on NU Discovery cluster

1. Connect to the Discovery cluster

2. Install/load Anaconda

We have two options here:

1. Obtain and install Miniconda executable specifically for python 2.7
   Obtain:
   ```
   $ wget https://repo.anaconda.com/miniconda/Miniconda2-py27_4.8.3-Linux-x86_64.sh
   ```
   Install:
   ```
   $ bash Miniconda2-py27_4.8.3-Linux-x86_64.sh
   ```
2. Load Anaconda module and activate it
   ```
   $ module load anaconda2/2018.12
   $ echo ". /shared/centos7/anaconda2/2018.12/etc/profile.d/conda.sh" >> ~/.bashrc
   $ source ~/.bashrc
   ```

3. Reserve a gpu/multigpu compute node

```
$ srun -p gpu --nodes=1 --pty --gres=gpu:1 --time=04:00:00 --export=ALL /bin/bash
```

4. Activate the environment

If you went with the first option:

```
$ source miniconda2/bin/activate
```

If you went with the second option:

```
$ conda activate
```

If everything goes well, `(base)` will show up next to your `username@computenode` in shell, as

```
(base)[username@c0001]$
```

5. Load the required modules

```
$ module load boost/1.63.0
$ module load cuda/10.2
```

6. Create a conda environemnt for PyGBe (takes a while)

First, delete the previously created `pygbe-env` environment from the old installation. You might have named it differently. If that's the case, you can see your environments at

```
$ ls ~/.conda/envs
```

To remove the old environment

```
$ conda remove -n pygbe-env --all
```

or delete the env folder

```
$ rm -rf ~/.conda/envs/pygbe-env
```

To create the new environment

```
$ conda create -n pygbe-env python=2.7
```

7. Activate the PyGBe environment

```
$ conda activate pygbe-env
```

8. Install the remaining dependencies

8.1 `numpy` and `scipy`

```
$ conda install numpy scipy
```

8.2 `pycuda`

before installing pycuda, we want to make sure there is absolutely no trace of previous installations left on your machine. So we delete them from the potential locations.

```
$ rm -rf ~/.conda/pkgs/pycuda*
$ rm -rf ~/.local/lib/python2.7/site-packages/pycuda*
```

Now install `pycuda`

```
$ pip install -Iv pycuda
```

9. Clone the repository

```
$ git clone https://github.com/alineu/pygbe.git
```

or

```
$ git clone git@github.com:alineu/pygbe.git
$ cd pygbe
```

10. checkout the asymmetric branch

```
$ git checkout asymmetric
```

11. Download and install SWIG

```
$ mkdir src
$ cd src
$ wget https://versaweb.dl.sourceforge.net/project/swig/swig/swig-3.0.12/swig-3.0.12.tar.gz
$ tar -xvzf swig-3.0.12.tar.gz
$ cd swig-3.0.12
$ ./configure --prefix=$PWD
$ make
$ make install
```

12. Install PyGBe and test the installation

12.1 Install

```
$ cd ../../bem_pycuda/
$ make all
```

12.2 Test

```
$ python main_asymmetric.py input_files/his.param input_files/his_stern.config --asymmetric --chargeForm
```

If the library is installed properly, you should see something like

```
Run started on:
Date: year/m/d
Time: hr:min:sec
Reading pqr for region 2 from ../geometry/his/his_prot.pqr
Reading surface 0 from file ../geometry/his/his_d01
Time load mesh: 0.036644
Removed areas=0: 0
Reading surface 1 from file ../geometry/his/his_d01_stern
Time load mesh: 0.041806
Removed areas=0: 81
Total elements : 1555
Total equations: 3110
...
...
...
Totals:
Esolv = -19.045255 kcal/mol
Esurf = 0.000000 kcal/mol
Ecoul = -119.087367 kcal/mol
Time = 15.916039 s
```

13. Whenever you're done:

```
$ conda deactivate
$ conda deactivate
```

twice!

Note

Next time, when you connect to a GPU compute node you only need to run the following to enable and use `PyGBe` library:

Activate the Anaconda Module

If you installed Miniconda locally (via first option):

```
$ source activate miniconda2
```

otherwise,

```
$ module load anaconda2/2018.12
$ conda activate
```

Activate the PyGBe Environment

```
$ conda activate pygbe-env
```

Load the Rest of the Modules

```
$ module load boost/1.63.0
$ module load cuda/10.2
```

The library should be ready to use!

Important

You might have to add the location of `pycuda` library to your `PATH` if `Python` fails to import the library. You can do this by adding

```
export PATH=$PATH:path_to_pycuda
```

to your `~/.bashrc` file (and then source it!). `path_to_pycuda` is the location where the `pycuda` library is installed, e.g. `$HOME/.conda/envs/pygbe-env/lib/python2.7/site-packages/pycuda` If you don't know the location of `pycuda` library you can find it using

```
find ~ -type d -name 'pycuda*'
```