Quick start guide for installing and running Diablo on the Cambridge HPC

(Dated: December 10, 2014)

I. INSTALLING THE CODE

1. Create a new directory where you want to store the code. When you install the code, a Diablo subdirectory will be created inside this directory.

2. Inside this directory, install the code using the Mercurial archive

   - type: `touch .hgrc` at the command prompt
   - open up `.hgrc` in a text editor and add the following lines:
     `[ui]`
     `username = myUserName < mymail@mail.com >`
     with your Darwin username and a valid email address
   - Now, download the latest version of the code from the mercurial archive with the following command:
     `hg clone https://edeusebio85@bitbucket.org/edeusebio85/diablo`

3. The code should now be installed in the `diablo` directory.

4. The FFTW libraries on Darwin have a different name to those on other systems, so you will need to edit the link options in the Makefile to use the correct library name. To do this, move into the `for` subdirectory within the `diablo` directory. Edit the Makefile here and change `-ldfftw` to `-lrfftw` in the definition of the LINKOPTS variable.

5. To compile the code, load the appropriate module files with the following commands:
   `module add fftw/intel/64/double/2.1.5`
   `module add hdf5/impi/1.8.12`

6. Still inside the `for` subdirectory, compile the code with the following command:
   `make PARALLEL=TRUE HDF5=TRUE`

7. Create a new directory somewhere where you will store the executable and input and output files from your simulation. This should be in your `/scratch` directory since the output files will quickly up your `/home` directory quota.

8. Copy the files from the `input_files` directory of the Diablo directory into the run directory that you just created.

9. Edit the `grid_def` file to set the size of your computational grid. This is the size of the full domain.

10. Edit `grid_def.all`. Here, NX and NZ should match those in `grid_def`. With NY_{full} equal to the value of NY set inside `grid_def`, the value of NY in `grid_def.all` should be:

$$NY = \frac{NY_{full} - 1}{nprocs + 1},$$

where *nprocs* is the number of processors for your simulation.

11. Copy the executable `diablo` from the `diablo/for` subdirectory into your run directory.

12. Create a new directory called `bin` in your home directory. Copy `submitjob.sh.sand` into this folder.

13. Edit the input files according to your simulation. These are described in the Diablo User's guide.

II. RUNNING THE CODE

Note, If you aren't sure what the name of your account is, run `mybalance`.

To run a simulation from inside your run directory, type: `submitjob.shsand -n NPROCS -t HH:MM:SS -c ACCOUNT -e diab`
where NPROCS is the number of processors (eg 16), `HH:MM:SS` is the runtime in hours, minutes, and seconds, and ACCOUNT is the name of your account.

To check to see if your simulation is running type: `qstat -u USERNAME` with your user name.

To cancel a job, type: `qdel JOB_ID` where `JOB_ID` is the JOB ID number listed in qstat.

An optional argument to `submitjob.shsand` is `-p PROCS_NODE`, where `PROCS_NODE` is the number of cores per node (the default is 8 which is the number of cores per node on Darwin).