Compiling Cytosim on cluster

05 June 2020

author: Alexandra (Sasha) Lamtyugina

The Linux environment of the cluster login nodes is missing several libraries and packages that are key to Cytosim. They cannot be installed without root privileges (probably for the best), so I found a workaround which involves using an Anaconda environment.

The procedure described below is quite "hacky," so if you are a C++ purist, avert your eyes.

Set up an Anaconda environment

First, load Anaconda with

```
module load anaconda3
```

Then, create and activate an environment for Cytosim and load the necessary packages:

```
conda create -n cytosim-env
conda activate cytosim-env

conda install -n cytosim-env python=3.7 numpy scipy matplotlib jupyter jupyterlab vtk pip
conda install -n cytosim-env -c conda-forge libblas lapack xorg-libxt freeglut zlib
conda install -n cytosim-env -c anaconda make cmake libpng
conda install -n cytosim-env -c psi4 gcc-5
conda install -n cytosim-env -c menpo glew
```

`-n` : `--name` , name of the environment to which the package should be installed

`-c` : `--channel` , channel from which package should be fetched

Modify `makefile.inc`

There are several modifications that need to be made to `makefile.inc` for compilation to run successfully. The modifications are in the same order that they appear in the file.

Compiler configuration

Copy the `GL` directory with `*.h` files from the `glew` (or was it `freeglut` ?) conda package directory to new directory `cytosim/src/extras` and add an include flag in the `makefile.inc` file under the `gcc` section of the compiler settings:

```
CXX := ... -Isrc/extras
```

Enable PNG export by changing the `HAS_PNG` variable:

```
HAS_PNG := 1
```

Linux-specific settings

Make sure to make the following edits in the Linux section of `makefile.inc`

Specify `$BLASDIR` in `makefile.inc` :

```
BLASDIR := /home/alexandral/.conda/pkgs/libblas*/lib/libblas.*
```

Change `GRAFIX` variable to

```
GRAFIX := -L$(LIBDIR) -lglut -lGL -lGLU -lXt -lX11
/home/alexandral/.conda/pkgs/glew-2.0.0-0/lib/libGLEW.so
```

For some reason, midway does not have GLEW, so this is a BYOGLEW kind of party.

Link to `libpng.a` of `libpng` package and to `libz.a` of `zlib` package in `makefile.inc`

```
LIB_PNG := /home/alexandral/.conda/pkgs/libpng-1.6.37-hbc83047_0/lib/libpng.a
/home/alexandral/.conda/pkgs/zlib-1.2.11-h516909a_1006/lib/libz.a
```

Edit the `LD_RUN_PATH` environment variable

Run the following command in the shell:

```
export LD_RUN_PATH="/home/alexandral/.conda/pkgs/glew-2.0.0-0/lib/"
```

As specified in the `GRAFIX` variable, the compiler relies on a GLEW shared object in a non-standard location (standard location is `/urs/lib/`). Therefore, we need to indicate where the executable should look for `libGLEW.so` whenever it is run. Adding the location of `libGLEW.so` to `LD_RUN_PATH` will compile the Cytosim executables with information on where to find the file.

(See https://homepages.inf.ed.ac.uk/imurray2/compnotes/library_linking.txt for more in-depth information about dynamic linking.)

Run `make`

If previously attempted to run `make` command in `cytosim` directory, first run

```
make clean
```

then

```
make alldim
```

The above command will compile executables for 1-, 2-, and 3D (to be found in `bin1`, `bin2`, and `bin3`, respectively). Though the executables are in different directories, they will have the same name. To add the executables to `$PATH`, create new `bin` directory and copy executables into it as `sim1`, `play1`, `report1`, etc. to have dedicated executables for each dimension which can be called from anywhere.