**Project Chrono Installation on Agave**

As technical supporters for cluster have failed to install chrono module on agave, the alternative is to install the package in your home dir.

**Step1:**

Use source tree or GitHub desktop to clone chrono initial source file from GitHub(<https://github.com/projectchrono/chrono/tree/release/7.0>). Notice to clone the latest branch such as V.7.

**Step2.**

Based on the module you are using different libraries may be required. You can easily download these libraries. The requires library for core engine is EIGEN 3.4. (download here: <https://github.com/projectchrono/chrono/tree/release/7.0>). In addition, if you enable IRRLICHT module you need to download and input its dir. in cmake.

So far, the modules that resulted in a successful installation was module eigen/3.3.7-gcc-stock and gcc/7.3.0 loaded from cluster!

There have been issues with eigen 3.4 when added manually. You can load module eigen 3.4 from cluster but the eigen 3.4 lib format mises some include files such as core. So, if you want to use eigen 3.4 you have to manually add the lib to your home dir.

**Step3.**

Use Fila Zila or WIN SCP to transfer source file and other dependencies(libraries) to a dir. E.G, my case the dir. for source file is home/sshahho1/”Project CHRONO”/Initial

**Step4.**

Now you must use cmake to compile. Before doing so we need to load module cmake and gcc for newer version. I am using cmake/20.2 and gcc/7.3. Following commands are used to load, unload and…:

Module load gcc/7.3.0

Module unload gcc

Gcc –version : gcc/7.3.0

After compiling the modules go to the build dir. Then use cmake to give the dir. for source file using -S. In addition, you can use -D to simply add the dir for build dir. or value for chrono modules or libraries. See below:

cmake -S /home/sshahho1/"Project CHRONO"/initiall -DEIGEN3\_INCLUDE\_DIR=/home/sshahho1/"Project CHRONO"/chrono\_dependencies/eigen-3.4.0 ( OR simply load module before using cmake, no need for eigen dir after loading the module) -DIRRLICHT\_ROOT= -DIRRLICHT\_LIBRARY= -DENABLE\_MODULE\_GPU=on -DENABLE\_MODULE\_IRRLICHT=off -DBUILD\_DEMOS=on -DBUILD\_DEMOS\_GPU=on -DBUILD\_DEMOS\_IRRLICHT=off

In this example only core and gpu module have been enables. The problem now is the dir. for IRRLICHT since irrlicht lib for Linux must be complied locally and there are no .lib file as there is for win64

**Step5.**

Use “make” to build libraries.

**Step6.**

Now make the file for template including the source file and build dir. Go to build di rans run cmake by including the dir. for chrono:

cmake -S /home/sshahho1/Project\_CHRONO/my-projects-final/test1/initial/ -DChrono\_DIR=/home/sshahho1/Project\_CHRONO/buildv7/cmake/

**Step6.**

Generate the files using make.

Notes:

* DO not use file name that have space, instead use –
* Do not forget to Load anaconda.py3 when using interactive session
* GPU\_ballcosim.json not JSON!
* Include the absolute path for data folder starting /home/…, other wise the defaulte data path is ../data/gpu from exe dir
* If you load gcc 8x: The gcc version included in Ubuntu 18.04 is gcc-8 as seen [here](https://packages.ubuntu.com/bionic/libstdc++6), which only supports up to libstdc++.so.6.0.25.: load gcc 7x

“Illegal instruction” error: the processer you are running is different the one you used to install chrono. Check modules do not forger anaconda and run the case in an interactive session where at least one gpu exists.