Submitting MATLAB jobs via SLURM

If you need to submit one or more single core jobs, and are not encountering problems with the lack of specific toolbox licenses, then you might wish to submit your jobs directly with SLURM rather than going through the MATLAB parallel computing toolbox.

1. Build a MATLAB .m code file

Include all of the MATLAB commands that you need to execute in a MATLAB .m file, such as *sample.m* and place it somewhere in your home directory or subdirectory. The creation and contents of a .m file are beyond the scope of this document. Consult the MATLAB documentation for information on .m files.

2. Create a SLURM batch script

You will need to execute MATLAB from within a SLURM batch script, such as *sample.slurm* as follows. In this example, this file is saved in the same directory as *sample.m*.

sample.slurm

```
#!/bin/bash
#SBATCH --job-name=JobNameHere
#SBATCH --partition=partitionNameHere
#SBATCH --ntasks=1
#SBATCH --time=24:00:00
#SBATCH --mail-user=YourEmailAddressHere
#SBATCH --mail-type=ALL
#
echo "I ran on:"
cd $SLURM_SUBMIT_DIR
echo $SLURM_NODELIST
#
matlab -nodisplay -r "sample; quit"
```

In this example the *sample.slurm* script calls *matlab* with the -r option followed by the MATLAB script name that was created in step #1. Leave off the trailing .m from the script name when calling MATLAB this way.

Click here for more information about SLURM job scripts.

3. Submit the Job

After you have created *sample.m* and *sample.SLURM*, go to the directory where *sample.SLURM* resides, load the MATLAB module (if you have not already done so) and submit the job to the job scheduler:

```
[user@host ~]$  module load matlab/2013a
[user@host ~]$  sbatch ./sample.slurm
```

You should now be able to see your job in the partition by going to your Linux terminal window and using the *squeue* command.

> This type of job submission will compete with all other jobs on the system for run time. If no processors are available then the job submission will wait in the partition until it is able to run.

> Look at the list of MATLAB modules available. Different versions of MATLAB are available on respective clusters with some overlap.

> **Possible License Errors**
> This type of job may fail due to license checkout errors. MATLAB shares a site license with the rest of the campus. If your job is using a toolbox for which there are no licenses available for checkout, the job will fail when it starts to run. You will need to use the parallel

computing toolbox to avoid this problem.