Matlab is a commercial programming language and computing environment that is widely popular in many areas of engineering and science. In order to run Matlab on the cluster, you must first purchase a Matlab license from the Vanderbilt Software Store . Be sure you select the correct version of Matlab (you should see one that is specifically for the ACCRE cluster environment). Once you have supplied proof of purchase for the appropriate license, you will be added to a group of users on the cluster that has the necessary permissions to run the software. You can verify that you are in this group by running the `groups` command while logged into the cluster:

```
[jill@vmps12 ~]$ groups
science_lab matlab
```

Here, the user `jill` is in `science_lab` as her primary group and `matlab` as a secondary group, so she could run Matlab on the cluster.

1 Versions of Matlab on the ACCRE Cluster

To see a list of installed versions of Matlab on the cluster, use `pkginfo`:

```
[jill@vmps12 ~]$ pkginfo | grep matlab
       matlab   Matlab (r2014b) [matlab]
matlab_r2012a   Matlab (r2012a) [matlab]
matlab_r2013a   Matlab (r2013a) [matlab]
matlab_r2014a   Matlab (r2014a) [matlab]
matlab_r2015a   Matlab (r2015a) [matlab]
```

Multiple versions of Matlab are available on the cluster. We encourage users to use the most recent version of Matlab (r2015a) installed, if possible, as we will not offer support for older versions indefinitely.

To load a particular version of Matlab, use `setpkgs`:

```
[jill@vmps12 ~]$ setpkgs -a matlab_r2015a
[jill@vmps12 ~]$ which matlab
/usr/local/matlab/2015a/bin/matlab
```

In order to run Matlab interactively, simply type `matlab` from the Linux command line:

```
[jill@vmps12 ~]$ matlab
MATLAB is selecting SOFTWARE OPENGL rendering.

                       < M A T L A B (R) >
```

```
    Copyright 1984-2015 The MathWorks, Inc.
       R2015a (8.5.0.197613) 64-bit (glnxa64)
                February 12, 2015

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

    Academic License

>>
```

Note that the Matlab command prompt may take ~30-60 seconds to completely load, depending on which version you are using and whether you've loaded it previously in your current shell session. You can also load the full graphical user interface (GUI) to your local machine by logging into the cluster with X11 forwarding enabled (`ssh -X <vunetid>@login.accre.vanderbilt.edu`). In general, using the Matlab GUI from the cluster will be very slow, especially if you are outside the Vanderbilt network, so we do not recommend it. Instead, most users develop their Matlab code locally (from a laptop or desktop environment) for testing and then submit jobs to the cluster for batch (non-interactive) processing.

2 Checking Installed Packages

To see a list of the available Matlab toolboxes on the cluster, use the `ver` command from the Matlab command prompt:

```
>>ver
-----------------------------------------------------------------------------------------------------
MATLAB Version: 8.5.0.197613 (R2015a)
MATLAB License Number: 299681
Operating System: Linux 2.6.32-220.2.1.el6.x86_64 #1 SMP Fri Dec 23 02:21:33 CST 2011 x86_64
Java Version: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed
-----------------------------------------------------------------------------------------------------
MATLAB                                          Version 8.5      (R2015a)
Simulink                                        Version 8.5      (R2015a)
Aerospace Blockset                              Version 3.15     (R2015a)
Aerospace Toolbox                               Version 2.15     (R2015a)
Bioinformatics Toolbox                          Version 4.5.1    (R2015a)
Communications System Toolbox                   Version 6.0      (R2015a)
Computer Vision System Toolbox                  Version 6.2      (R2015a)
Control System Toolbox                          Version 9.9      (R2015a)
Curve Fitting Toolbox                           Version 3.5.1    (R2015a)
DSP System Toolbox                              Version 9.0      (R2015a)
Database Toolbox                                Version 5.2.1    (R2015a)
```

```
Datafeed Toolbox                                  Version 5.1       (R2015a)
Econometrics Toolbox                              Version 3.2       (R2015a)
Embedded Coder                                    Version 6.8       (R2015a)
Filter Design HDL Coder                           Version 2.9.7     (R2015a)
Financial Instruments Toolbox                     Version 2.1       (R2015a)
Financial Toolbox                                 Version 5.5       (R2015a)
Fixed-Point Designer                              Version 5.0       (R2015a)
Fuzzy Logic Toolbox                               Version 2.2.21    (R2015a)
Global Optimization Toolbox                       Version 3.3.1     (R2015a)
Image Acquisition Toolbox                         Version 4.9       (R2015a)
Image Processing Toolbox                          Version 9.2       (R2015a)
Instrument Control Toolbox                        Version 3.7       (R2015a)
MATLAB Coder                                      Version 2.8       (R2015a)
MATLAB Compiler                                   Version 6.0       (R2015a)
MATLAB Compiler SDK                               Version 6.0       (R2015a)
Mapping Toolbox                                   Version 4.1       (R2015a)
Model Predictive Control Toolbox                  Version 5.0.1     (R2015a)
Neural Network Toolbox                            Version 8.3       (R2015a)
Optimization Toolbox                              Version 7.2       (R2015a)
Parallel Computing Toolbox                        Version 6.6       (R2015a)
Partial Differential Equation Toolbox             Version 2.0       (R2015a)
RF Toolbox                                        Version 2.16      (R2015a)
Robust Control Toolbox                            Version 5.3       (R2015a)
Signal Processing Toolbox                         Version 7.0       (R2015a)
SimBiology                                        Version 5.2       (R2015a)
SimDriveline                                      Version 2.8       (R2015a)
SimElectronics                                    Version 2.7       (R2015a)
SimEvents                                         Version 4.4       (R2015a)
SimMechanics                                      Version 4.6       (R2015a)
SimPowerSystems                                   Version 6.3       (R2015a)
SimRF                                             Version 4.4       (R2015a)
Simscape                                          Version 3.13      (R2015a)
Simulink 3D Animation                             Version 7.3       (R2015a)
Simulink Coder                                    Version 8.8       (R2015a)
Simulink Control Design                           Version 4.2       (R2015a)
Simulink Design Optimization                      Version 2.7       (R2015a)
Stateflow                                         Version 8.5       (R2015a)
Statistics and Machine Learning Toolbox           Version 10.0      (R2015a)
Symbolic Math Toolbox                             Version 6.2       (R2015a)
System Identification Toolbox                     Version 9.2       (R2015a)
Wavelet Toolbox                                   Version 4.14.1    (R2015a)
```

Note that ACCRE has a license for the Parallel Computing Toolbox , which allows for processing across multiple CPU cores and/or GPUs, and can therefore enable better performance (faster execution time) depending on the application.

3 Example Scripts

Running a Matlab script within a SLURM job is generally straightforward. Unless you are attempting to run across multiple CPU cores using Matlab's Parallel Computing Toolbox, you will want to request a single task, load the appropriate version of Matlab from your SLURM script, and then launch your Matlab job. The following example runs a simple Matlab script that demonstrates the utility of writing vectorized Matlab code:

```
[jill@vmps12 run1]$ ls
matlab.slurm  vectorization.m
```

```
[jill@vmps12 run1]$ cat matlab.slurm
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00
#SBATCH --mem=500M
#SBATCH --output=matlab_job_slurm.out

setpkgs -a matlab_r2015a

matlab -nodisplay -nosplash < vectorization.m
```

The `-nodisplay` flag informs Matlab you are operating in batch mode, while the `-nosplash` flag will prevent the splash screen from being displayed during startup. Additionally, you might try passing the `-nojvm` flag, which informs Matlab that you do not need Java features for processing. Passing this flag often leads to faster Matlab load-up times, but some I/O operations may depend on Java support so use this flag with caution. More information can be found on this page .

```
[jill@vmps12 run1]$ cat vectorization.m
% surrounding a block of code with tic and toc
% will time its execution

% non-vectorized code
tic
i = 0;
for t = 0:.00001:10
i = i + 1;
y(i) = sin(t);
end
toc

% vectorized code
tic
t = 0:.00001:10;
```

```
y = sin(t);
toc
```

```
[jill@vmps12 run1]$ sbatch matlab.slurm
Submitted batch job 2135971
```

After waiting a few minutes:

```
[jill@vmps12 run1]$ ls
matlab_job_slurm.out  matlab.slurm  vectorization.m
```

```
[jill@vmps12 run1]$ cat matlab_job_slurm.out

                 < M A T L A B (R) >
        Copyright 1984-2015 The MathWorks, Inc.
         R2015a (8.5.0.197613) 64-bit (glnxa64)
                   February 12, 2015


To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.


    Academic License

>> >> >> >> >> >> >> >> >> Elapsed time is 0.477166 seconds.
>> >> >> >> >> >> Elapsed time is 0.057248 seconds.
```

In this particular example no functions were defined in the vectorization.m file. Often users write scripts with functions, and need to call a function from the Linux command line. This can be accomplished by passing the Matlab interpreter the -r option. For example:

```
matlab -nodisplay -nosplash -r "myFunc(1),quit()"
```

Here we are calling myFunc() and passing a single argument (1) to the function. It's also necessary to call quit() afterwards to ensure your job ends once Matlab processing has completed. You may also need to update your Matlab path to include directories containing .m files and their function definitions. There are a few ways to accomplish this. The first way is to update your MATLABPATH Bash environment variable. Something like the following can be done within a SLURM script before Matlab is launched:

```
export MATLABPATH=/home/jill/myDir
```

Alternatively, the path can be updated at runtime like so:

```
matlab -nodisplay -nosplash -r "addpath(genpath('/home/jill/myDir')),myFunc(1),quit()"
```

4 Contributing New Examples

In order to foster collaboration and develop local Matlab expertise at Vanderbilt, we encourage users to submit examples of their own to ACCRE's Matlab Github repository . Instructions for doing this can be found on this page .