Copy of FAQ - How do I use Matlab on the HPC clusters?

<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC at NYU</td>
</tr>
<tr>
<td>HPC Newsletter</td>
</tr>
<tr>
<td>Getting started on Prince</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Prince How-to Articles</td>
</tr>
<tr>
<td>Logging in</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Mac / Linux</td>
</tr>
<tr>
<td>Clusters and Storage July 2017</td>
</tr>
<tr>
<td>Mercer (HPC, retired)</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
</tbody>
</table>
Why does running "ls" on /scratch take so long?

I can't login

When trying to login, I get warning about "HOST IDENTIFICATION HAS CHANGED"

What happened to my data on /scratch?

In the library, my wireless connection keeps dropping out. How can I fix it?

I'm getting a "module: command not found" error
Warning: no access to tty (Bad file descriptor), thus no job control in this shell.

I get an error: "Warning: no display specified." when I use -X flag with ssh.

Who killed my job, and why?

I got an email: "Please do not run jobs on login nodes."

Running jobs

What resources can and should I request?

Can I make sure a job gets executed only after another one completes?

How do I log in to a specific node?

How can I make sure my job is running smoothly?
My job will take longer than 48 hours, what should I do?

My job needs (MySQL, some other service) to be running

I want to run a job at 9am every day

Using software

How do I run ...
(esp, needs a license)

A STATTA job?

A Gaussian job?

A Matlab job?
I can't find (some software package)
Can you install (some software package)?
How can I view a PDF file on Prince?

Managing data
How much of my file/space quota have I used?
How do I give my colleague access to my files?
How do I get the best transfer speed to or from BuTinah?
I have a huge amount of data that I want to compress for storage or transfer.

### Detecting failed Matlab jobs

By default, Matlab returns 0 even when it stops due to an error. The batch system (and Unix generally) interprets a 0 exit code as "completed successfully", any other return code means "failed with error". This makes discovering failed Matlab jobs difficult, and can cause problems if you are using job dependencies for a workflow.

To return a meaningful error code from Matlab you can wrap the Matlab code in a "try ... catch" block. There is an example of how to do this at /share/apps/examples/matlab/basic

### Running serial Matlab jobs with a single thread

You may run serial Matlab jobs from an interactive session or the batch mode. In both ways, you need to either load the module as,

```bash
$ module load matlab/2014a
$ matlab
```

By default, MATLAB makes use of the multithreading capabilities of the computer on which it is running. Mercer uses cpusets so that your job sees only the cpus allocated to it, therefore you no longer need to specify "-mutlipleCompThreads" as was the case on Bowery.

Please also declare the correct CPU cores for your Matlab job, which should agree with the thread number. A multi-threading Matlab job script example is,

```bash
#!/bin/csh -f
#PBS -V
#PBS -S /bin/tcsh
#PBS -N Matlab-test
#PBS -l nodes=1:ppn=8,walltime=01:00:00
#PBS -M NetID@nyu.edu
#PBS -m abe
cd /scratch/NetID/m matlab-workdir
/share/apps/matlab/R2009b/bin/matlab <INPUT >OUTPUT
```

**Note**

If you need fewer CPU cores than the total CPU number in a node (8 in most cases), you must also declare the correct number in the "OMP_NUM_THREADS" variable.

**Note**

Be sure to substitute your own "NetID" for NetID.

Your PBS script will be like this if you only need two CPU cores,
Running parallel Matlab jobs with PCT (on one node)

Because the MATLAB Distributed Computing Server is not available on the NYU HPC clusters, you cannot run parallel Matlab jobs across multiple compute nodes but only on one node. This can be realized with the Parallel Computing Toolbox (PCT) in Matlab. You may refer to the Matlab website for more information on PCT.

Please note that the PCT in Matlab tends to write scratch files to the same directory (e.g., /home/NetID/matlab/R2014a). As a result, these files can be overwritten or cause a conflict if you run multiple PCT jobs at the same time. The solution is to define a private folder for each job.

To do so, add this part to your PBS script below any "#PBS" lines and before executing Matlab if you use tcsh/csh,

```bash
set tmp_folder_base = $PBS_JOBTMP
mkdir -p $tmp_folder_base
set data_location = `mktemp -d "$tmp_folder_base/MATLAB-data-XXXXXXXXXX"`
setenv DATA_LOCATION $data_location
setenv NTHREADS `cat $PBS_NODEFILE | wc -l`
```

Or this part if you use bash,

```bash
tmp_folder_base=$PBS_JOBTMP
mkdir -p $tmp_folder_base
data_location=$(mktemp -d "$tmp_folder_base/MATLAB-data-XXXXXXXXXX")
export DATA_LOCATION=data_location
export NTHREADS=$(cat $PBS_NODEFILE | wc -l)
```

For either shell, please also add "rm -rf $data_location" right before the "exit" in your PBS script.

Then modify the .m input file to include these lines before calling the "matlabpool" function,

```matlab
data_location = getenv('DATA_LOCATION');
nthreads = getenv('NTHREADS');
scheduler = findResource('scheduler', 'type', 'local')
scheduler.DataLocation = data_location
```

Now you can start PCT by calling "matlabpool" like this in your .m input file,

```matlab
matlabpool('open', 'local', nthreads)
```

**Special Notes on Running Matlab 2010b**

Matlab 2010b was installed on USQ and Bowery, which have since been decommissioned.
Matlab 2010b was also installed on USQ and Bowery. However, possibly due to a bug, the program will hang after printing the first line (lineA) if you run this script,

```bash
#!/bin/bash -f
/share/apps/matlab/R2010b/bin/matlab -nodisplay -r "
fprintf('lineA\n');
fprintf('lineB\n');
fprintf('lineC\n');
exit"
```

This may also prevent you from executing any further Matlab commands. As an alternative, you can change your script to this format,

```bash
#!/bin/bash -f
cat <<EOF | /share/apps/matlab/R2010b/bin/matlab -nodisplay
fprintf('lineA\n');
fprintf('lineB\n');
fprintf('lineC\n');
exit
EOF
```

which can finish correctly with an output,

```
lineA
lineB
lineC
```

**Further help**

NYU Data Services also provides Matlab support and tutorials