There are three ways to run Matlab

1. Using the Matlab integration with Python in a Jupyter notebook.
2. Graphical mode using the Ocelote Desktop of OnDemand
3. The command line version using modules. This is the most common as you will typically submit a job using PBS.

MATLAB performs its own hardware discovery and it might try to access all the cores and the memory of the node even if the full node wasn’t allocated. That will result in scheduler killing the job. To prevent that the full Ocelote node of 28 cores and 168GB of memory should be allocated to run a MATLAB job.

Like any other application, MATLAB has to be loaded as a module before you can use it. To see all the installed versions of the MATLAB use command `module avail matlab`.

The typical procedure for performing calculations on UA HPC systems is to run your program non-interactively on compute nodes. The easiest way to run MATLAB non-interactively is to use input/output redirection. This method uses Linux operators `<` and `>` to point MATLAB to the input file and tell where to write the output (see the example script). The other method is to invoke MATLAB from the PBS script and execute specified statement using `-r` option. For details please refer to the manual page of `matlab` command:

https://www.mathworks.com/help/matlab/ref/matlablinux.html

```
#!/bin/bash
#PBS -N job_name
#PBS -W group_list=group_name
#PBS -q standard
#PBS -l select=1:ncpus=28:mem=168gb:pcmem=6gb
#PBS -l walltime=01:00:00
#PBS -l cput=28:00:00

cd $PBS_O_WORKDIR

module load matlab

matlab -nodisplay -nosplash < script_name.m > output.txt
```

The options `-nodisplay` and `-nosplash` in the example prevent MATLAB from opening elements of GUI. To view the full list of options for `matlab` command load the MATLAB module and type `matlab -h` in Linux prompt, or use the link above to the manual page on MathWorks website.

2.

Use the OnDemand interface to start an Ocelote Desktop session.