

Using Matlab



- [Overview](#)
- [Running Matlab Analyses](#)
- [Parallel Computing Toolbox](#)
- [Matlab and SLURM](#)
- [Resources](#)

Overview

There are four ways to run Matlab:

1. Using the Matlab graphical application through [Open OnDemand](#).
2. Graphical mode using the [Open OnDemand Desktops](#).
3. The command line version using modules. This is the most common as you will typically submit a job using SLURM.
4. Python

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 the command `module avail matlab`.

Running Matlab Analyses

The typical procedure for performing calculations on UArizona 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 SLURM script and execute specified statement using `-r` option. For details, refer to the [manual page for the `matlab` command](#).

Example Matlab batch script

```
#!/bin/bash
#SBATCH --job-name=matlab
#SBATCH --account=group_name
#SBATCH --partition=standard
#SBATCH --ntasks=20
#SBATCH --nodes
#SBATCH --mem-per-cpu=5gb
#SBATCH --time=01:00:00

module load matlab/<version>

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

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

Parallel Computing Toolbox

By default, Matlab PCT will dump files to `~/matlab/MATLAB_VERSION`. This causes problems when multiple Matlab PCT jobs are running simultaneously. Users should always define the environment variable `MATLAB_PREFDIR` so each job uses a unique temporary folder. Files there will be cleaned after the job finishes. For example:

```
export MATLAB_PREFDIR=$(mktemp -d $SLURM_JOBTMP/matlab-XXXX)
```

Matlab and SLURM

If you are trying to run Matlab in parallel interactively, you may encounter the following error:

```
>> Starting parallel pool (parpool) using the 'local' profile ...
Error using parpool (line 149)
You requested a minimum of <n> workers, but the cluster "local" has the NumWorkers property set to allow a
maximum of 1 workers. To run a communicating job on more workers than this
(up to a maximum of 512 for the Local cluster), increase the value of the NumWorkers property for the
cluster. The default value of NumWorkers for a Local cluster is the number of
physical cores on the local machine.
```

This is caused by an interaction between SLURM and Matlab. To resolve this issue, when requesting <n> cores for your interactive job, you will need to set SLURM's `--ntasks` directive to 1 and `--cpus-per-task` to the number of cores you need. For example:

```
$ salloc --nodes=1 --ntasks=1 --cpus-per-task=6 --mem-per-cpu=5GB --time=01:00:00 --job-name=interactive --
account=<GROUP> --partition=standard
```

Resources

If you are getting started with Matlab or think there might be a better way, check out the training resources.

Self-Paced Online Courses

<https://matlabacademy.mathworks.com/>

Matlab Parallel Server

<https://www.mathworks.com/products/matlab-parallel-server.html#resources>

Natural Language Processing

<https://www.mathworks.com/discovery/natural-language-processing.html>

Matlab Videos

<https://www.mathworks.com/videos.html>