



SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Matlab on SCW — Introduction



UNDEB EWROPEAIDD
EUROPEAN UNION



Llywodraeth Cymru
Welsh Government

**Cronfa Datblygu
Rhanbarthol Ewrop
European Regional
Development Fund**

Contents

1	Introduction	2
2	Matlab on SCW	2
2.1	Using Matlab	3
2.1.1	Running a Matlab Script	3
2.1.2	Matlab Command Line Arguments	4
2.2	Submitting Matlab Jobs via SLURM	5
2.3	SLURM Support	6

1 Introduction

This guide provides a brief introduction on using **Matlab** on Hawk/Sunbird, and assumes user knowledge/familiarity with **Matlab**.

If you have any issues using this guide, or would like further details on something, please contact support@supercomputingwales.ac.uk

2 Matlab on SCW

Before using **Matlab** we need to load it into our SCW environment by executing the following:

```
module load matlab/R2019a
```

You can confirm this has worked by executing:

```
which matlab
```

Which should return: `/apps/local/languages/matlab/2019a/bin/matlab`.

You are now ready to start using **Matlab** on Hawk/Sunbird.

2.1 Using Matlab

Please Note: `Matlab` is only available via the command line on Hawk/Sunbird. For SCW users unfamiliar with this process, we will run through a simple example script to document the procedure.

2.1.1 Running a Matlab Script

Consider the simple `Matlab` script below, which creates a 5x5 magic square, and has been saved ingeniously as `magic_square.m`:

```
% Creates a 5x5 Magic square
M = magic(5);
M
exit
```

To execute this script from the command line, we need to run the following:

```
matlab -nosplash -nodesktop -r "run('magic_square.m'); exit;"
#output:
      M =
      17    24     1     8    15
      23     5     7    14    16
       4     6    13    20    22
      10    12    19    21     3
      11    18    25     2     9
```

Please Note: The trailing `exit` in the above example is superfluous, as we ended our `magic_square.m` script with `exit`. However, it's good practice to include it to ensure the script does `exit` correctly.

This would have worked too:

```
matlab -nosplash -nodesktop -r "run('magic_square.m');"
```

2.1.2 Matlab Command Line Arguments

In the example above, we passed the following arguments to **Matlab**: `-nosplash` and `-nodesktop`. Below is a brief description of what these, and a couple of other **Matlab** command line arguments do:

<code>-nosplash</code>	prevents Matlab from displaying the GUI splash screen
<code>-nodesktop</code>	starts Matlab without the desktop environment
<code>-nodisplay</code>	suppresses displaying figures during Matlab execution
<code>-logfile <file></code>	copy output, including errors, into <file>

Please Note: There is a way of running our `magic_square.m` example script without using `-nodesktop`:

```
matlab < magic_square.m
```

This essentially reads the contents of `magic_square.m` and passes it to the `matlab` command. Implementing `<` automatically uses the `-nodesktop` argument.

2.2 Submitting Matlab Jobs via SLURM

To submit our `magic_square.m` script to **SLURM**, the following script can be written:

```
#!/bin/bash --login

#set job name:
#SBATCH --job-name=matlab_test
#set output file name:
#SBATCH --output=matlab_test.out.%J
#store any errors in this file name:
#SBATCH --error=matlab_test.err.%J
#set max wallclock time limit[D:HH:MM]:
#SBATCH --time=0-00:01
#set memory required per CPU (MB):
#SBATCH --mem-per-cpu=10
#set number of parallel processes/tasks you're requesting:
#SBATCH --ntasks=1
#set the account to be charged for resources used:
#SBATCH --account=scw1001

module load matlab/R2019a
matlab -nosplash -nodesktop -r "run('magic_square.m'); exit;"
```

If we pretend this **SLURM** script was saved as `matlab_slurm.sh` then we can submit this to the **SLURM** queue by executing:

```
sbatch matlab_slurm.sh
```

Please Note: These **SLURM** settings are only a guide. Anything more complicated than producing a magic square will require more time, memory, et cetera. So please bear that in mind.

2.3 SLURM Support

For further information regarding **SLURM**, please consult the SCW guide on submitting jobs, which can be found here: <https://portal.supercomputing.wales/index.php/index/submitting-jobs/>.

You can also visit <https://recon.swansea.ac.uk/> and chat with our team of Research Software Engineers/Sunbird Administrators for assistance/support on this and other matters.