

Parallel Computing in Matlab on BASS

File names:

1. myParallelJob.qsub
2. myParallelScript.m
3. myParallelFunction.m

File contents:

1. myParallelJob.qsub: Save into /home1/username

```
# Submit MatLab job.
# ref http://pages.stern.nyu.edu/~nwhite/scrc/matjobs.html
#
#$ -S /bin/bash
# Request 8 processors
#$ -pe smp 8
# Request a machine with MATLAB on it
#$ -l matlab
# Where to put the output
#$ -o $JOB_NAME.$JOB_ID
#
matlab -nodisplay < myParallelScript.m
```

2. myParallelScript.m: Save into /home1/username

```
% cd to the folder where myParallelFunction.m resides. Since this script
% assumes you are at /home1/username, it means the full path to the m-file
% should be /home1/username/MyFolder/myParallelFunction.m.
cd MyFolder/
% Allocate 8 CPU's
matlabpool 8
% Generate a 1x8 vector of values, each to be used by one of the eight
% threads.
thetaVector = rand(1,8);
% Parallel for [multithread]
parfor k = 1:8
    myParallelFunction(k,thetaVector(k))
end
% Deallocate the CPU's
matlabpool close
```

3. myParallelFunction.m: Save into /home1/username/MyFolder

```
function myParallelFunction(thread,theta)

for k = 1:5
    % Display thread, loop and parameter info. All these are going to be
    % output in the command window, and hence be saved in $JOB_NAME.$JOB_ID
    % [Recall: -o option in .qsub file]
    disp(['Thread: ' num2str(thread) ', k = ' num2str(k) ', theta = ' num2str(theta)]);
    % Pause for 5 seconds to simulate long running times.
    pause(5);
end
```

How to run:

Open an X-Win32 session. As soon as you see `-bash-3.2$`, input:

```
qsub myParallelJob.qsub
```

This is going to submit your job.

How to watch progress:

If you have put **disp** or **fprintf** statements into your `myParallelFunction.m`, you should be able to gauge the progress of your job. In the `.qsub` file, we had stated an option like `-o $JOB_NAME.$JOB_ID`. This is where you are going to see what has been printed into Matlab's command window [i.e. the window where you see the `>>` sign]. The name of this file is going to be something like **myParallalJob.qsub.xxxxxxx** where **xxxxxxx** is a system-assigned number which is going to be printed right after you submit your job as follows:

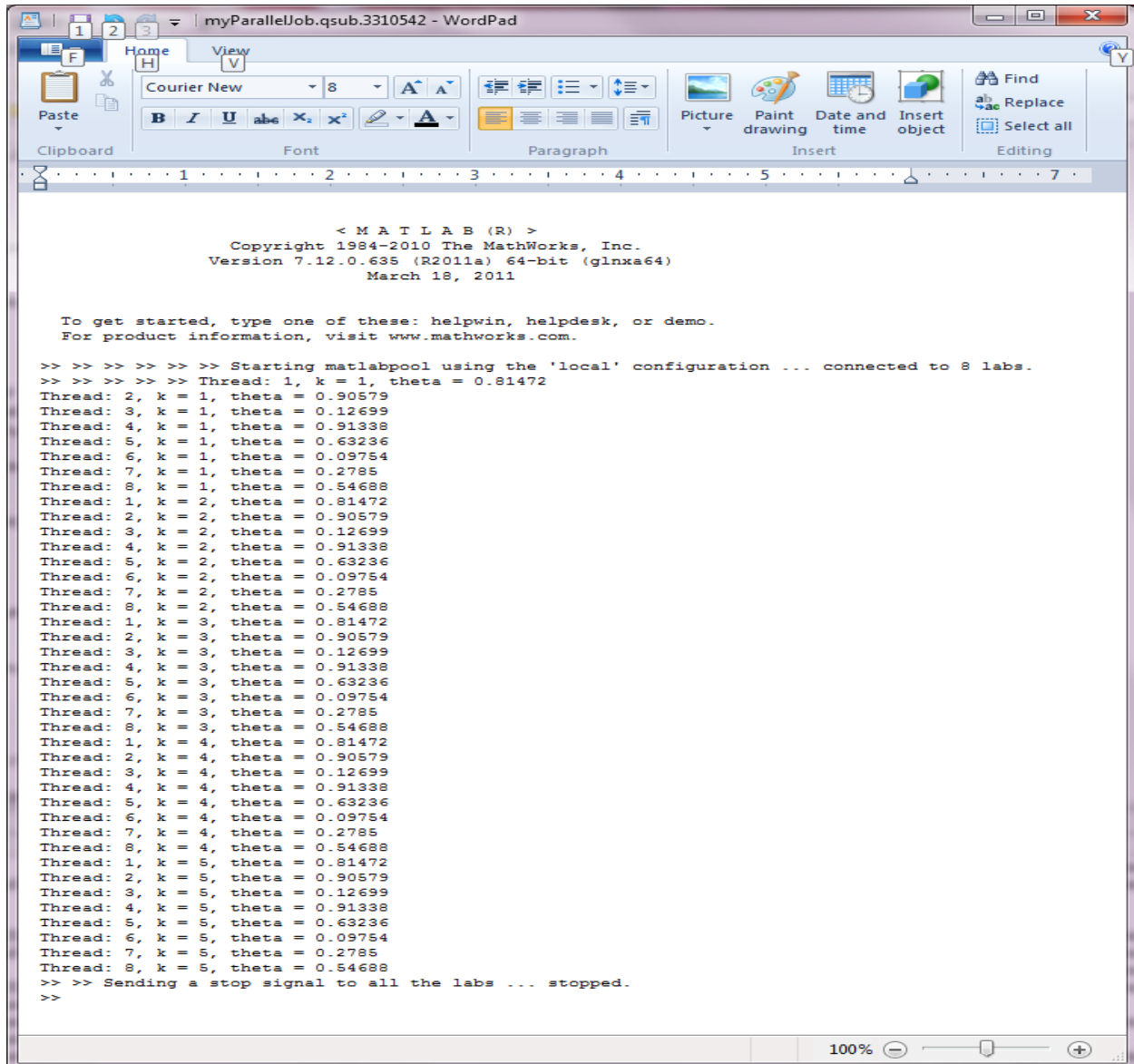
```
Your job xxxxxxxx ("myParallelJob.qsub") has been submitted.
```

As the job is executed, the size of this file is going to get larger and larger due to the increasing amount of output in the command window. In order to open this file, it is recommended that you use wordpad or other text editors that recognize the escape character so that you can see the output line by line.

Output of an example run:

We can see that the first loop of the `for` statement in `myParallelFunction.m` is executed once for each of the 8 threads, and then the second loop runs, and this continues until the fifth and the last loop. This output is a proof that 8 different `myParallelFunction.m`'s, are running in parallel threads.

Note the file name in the window title.



```
< M A T L A B (R) >
Copyright 1984-2010 The MathWorks, Inc.
Version 7.12.0.635 (R2011a) 64-bit (glnxa64)
March 18, 2011

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

>> >> >> >> >> Starting matlabpool using the 'local' configuration ... connected to 8 labs.
>> >> >> >> >> Thread: 1, k = 1, theta = 0.81472
Thread: 2, k = 1, theta = 0.90579
Thread: 3, k = 1, theta = 0.12699
Thread: 4, k = 1, theta = 0.91338
Thread: 5, k = 1, theta = 0.63236
Thread: 6, k = 1, theta = 0.09754
Thread: 7, k = 1, theta = 0.2785
Thread: 8, k = 1, theta = 0.54688
Thread: 1, k = 2, theta = 0.81472
Thread: 2, k = 2, theta = 0.90579
Thread: 3, k = 2, theta = 0.12699
Thread: 4, k = 2, theta = 0.91338
Thread: 5, k = 2, theta = 0.63236
Thread: 6, k = 2, theta = 0.09754
Thread: 7, k = 2, theta = 0.2785
Thread: 8, k = 2, theta = 0.54688
Thread: 1, k = 3, theta = 0.81472
Thread: 2, k = 3, theta = 0.90579
Thread: 3, k = 3, theta = 0.12699
Thread: 4, k = 3, theta = 0.91338
Thread: 5, k = 3, theta = 0.63236
Thread: 6, k = 3, theta = 0.09754
Thread: 7, k = 3, theta = 0.2785
Thread: 8, k = 3, theta = 0.54688
Thread: 1, k = 4, theta = 0.81472
Thread: 2, k = 4, theta = 0.90579
Thread: 3, k = 4, theta = 0.12699
Thread: 4, k = 4, theta = 0.91338
Thread: 5, k = 4, theta = 0.63236
Thread: 6, k = 4, theta = 0.09754
Thread: 7, k = 4, theta = 0.2785
Thread: 8, k = 4, theta = 0.54688
Thread: 1, k = 5, theta = 0.81472
Thread: 2, k = 5, theta = 0.90579
Thread: 3, k = 5, theta = 0.12699
Thread: 4, k = 5, theta = 0.91338
Thread: 5, k = 5, theta = 0.63236
Thread: 6, k = 5, theta = 0.09754
Thread: 7, k = 5, theta = 0.2785
Thread: 8, k = 5, theta = 0.54688
>> >> Sending a stop signal to all the labs ... stopped.
>>
```

For Questions: You should not have any questions after this clear tutorial. :D