MATLAB: How to speed up your code and run jobs on Viking from MATLAB

Philip Harrison

Research Coding Club
3rd May 2023

Today's talk

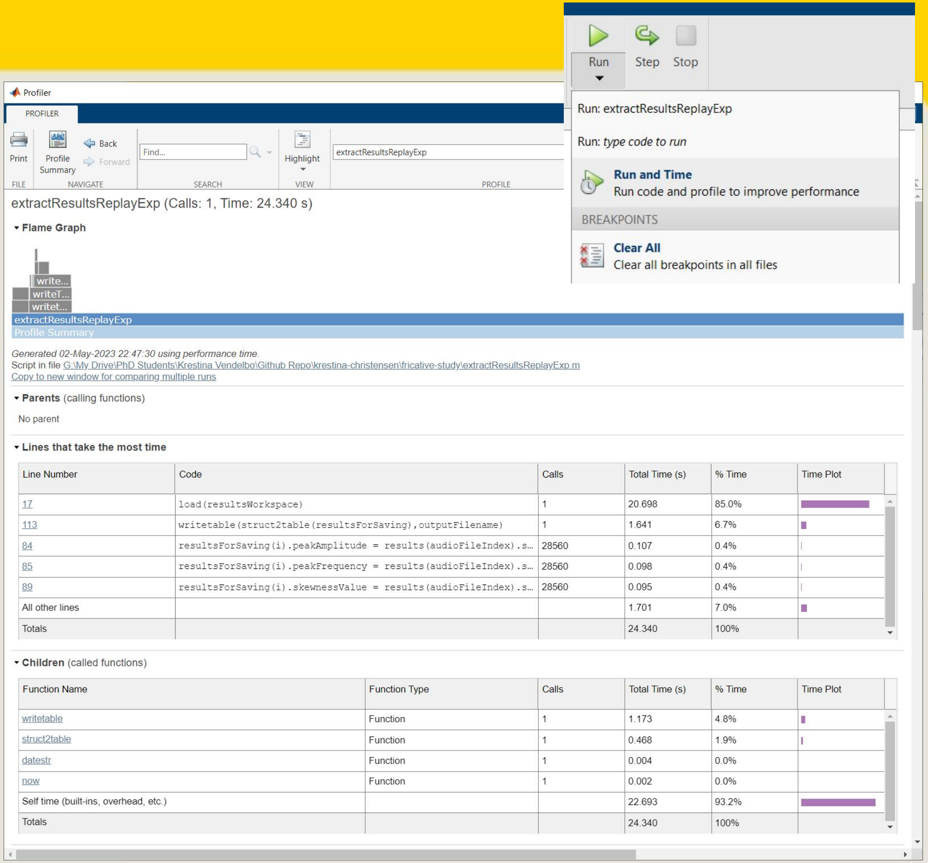
- Topic: Introduction on how to speed up your MATLAB code
- How to measure execution time
- Simple steps
  - Pre-assign arrays, vectorisation, built-in functions
- Use parallelisation
  - On your computer
- Use Viking
  - What is Viking?
  - Prerequisites and setup
  - Using Viking from within MATLAB
  - Getting data onto Viking
- Final comments

Why speed up your code?

- Who wants to wait for code to run?
- Faster code equals
    - More efficient use of time and resources
    - Get more research done
    - Use your computer for longer
    - Good habit to get into - might not make much difference now, but could in the future
- Get the low hanging fruit first
- Strike a balance - is the time you're spending to speed up your code longer than the time saved?

How to measure execution time

- How long does it take my code to run?
- Simple approach
  - Stopwatch - `tic` and `toc`
- More complex approach
  - Code profiler
  - Simple to use 'Run and time'
  - Produces interactive report:
    - Execution time
    - Number of calls to a function
- Mathworks help: [Measure performance of your program]()



extractResultsReplayExp (Calls: 1, Time: 24.340 s)

Flame Graph

extractResultsReplayExp
Profile Summary

Generated 02-May-2023 22:47:30 using performance time.
Script in file G:\My Drive\PhD Students\Krestina Vendelbo\Github Repo\krestina-christensen\fricative-study\extractResultsReplayExp.m
Copy to new window for comparing multiple runs

Parents (calling functions)

No parent

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
17	`load(resultsWorkspace)`	1	20.698	85.0%	▬
113	`writetable(struct2table(resultsForSaving),outputFilename)`	1	1.641	6.7%	▮
84	`resultsForSaving(i).peakAmplitude = results(audioFileIndex).s...`	28560	0.107	0.4%	|
85	`resultsForSaving(i).peakFrequency = results(audioFileIndex).s...`	28560	0.098	0.4%	|
89	`resultsForSaving(i).skewnessValue = results(audioFileIndex).s...`	28560	0.095	0.4%	|
All other lines			1.701	7.0%	▮
Totals			24.340	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
writetable	Function	1	1.173	4.8%	▮
struct2table	Function	1	0.468	1.9%	|
datestr	Function	1	0.004	0.0%	
now	Function	1	0.002	0.0%	
Self time (built-ins, overhead, etc.)			22.693	93.2%	▬
Totals			24.340	100%	

Simple ways to speed up MATLAB code: Preallocate arrays

- Resizing arrays within a loop takes extra time to find more memory

```
tic
x = 0;
for k = 2:1000000
  x(k) = x(k-1) + 5;
end
toc

Elapsed time is 0.107429 seconds.
```

vs

```
tic
x = zeros(1,1000000);
for k = 2:1000000
  x(k) = x(k-1) + 5;
end
toc

Elapsed time is 0.017111 seconds.
```

- Version on right with preallocation approximately 6 times faster
- MATLAB will warn you in the code editor, code profiler and code analyser
- Mathworks help - Preallocating arrays

Simple ways to speed up MATLAB code: Vectorisation

- MATLAB optimised for vector and matrix operations
- Vector and matrix algebra and functions

```
tic
i = 0;
for t = 0:.01:10
   i = i + 1;
   y(i) = sin(t);
end
toc


Elapsed time is 0.013842 seconds.
```

vs

```
tic
t = 0:.01:10;
y = sin(t);
toc


Elapsed time is 0.006431 seconds.
```

- Version with vectorisation approximately 2 times faster
- Code is neater, more readable, fewer chances for bugs
- Mathworks help - Vectorization

Simple ways to speed up MATLAB code: Use built-in functions

- Don't reinvent the wheel!
- Optimised for speed - preallocation, vectorisation
- Written by MATLAB experts, refined over time
- Check available Toolboxes
  - All licensed ones should be installed on managed PCs
  - Add relevant toolboxes on unmanaged/personal devices
- Search Mathworks FileExchange
  - Community repository of code examples, functions, applications
  - https://uk.mathworks.com/matlabcentral/fileexchange/
- Search internet

Simple ways to speed up MATLAB code: Parallelisation

- Normal `for` loops executes the code sequentially
- `parfor` loops execute in parallel - simultaneous execution!
- Can significantly speed up execution
- Can only use if each loop execution is independent of the others
- Useful for analysing multiple input data files or independent simulations
- Just swap `for` with `parfor`
- Uses the multiple cores in CPU of your computer
- Mathworks help - Decide when to use `parfor`

Simple ways to speed up MATLAB code: Parallelisation

```
tic
n = 200;
A = 500;
a = zeros(1,n);
for i = 1:n
   a(i) = max(abs(eig(rand(A))));
end
toc

Elapsed time is 21.236543 seconds
```

vs

```
tic
n = 200;
A = 500;
a = zeros(1,n);
parfor i = 1:n
   a(i) = max(abs(eig(rand(A))));
end
toc

Elapsed time is 9.479539
```

- Parallel version is 2 times faster with 4 cores
- Creates a parallel pool - but this takes time to create for the first run (extra 60 seconds)

What is Viking?

- University of York's Research Computing Cluster
- Cluster = lots of computers working as a single system
- Free at the point of use
- Offload code execution from local computer to the cluster
- Typically use Linux command line to interact with it
- Submit 'jobs' requesting specific resources
    - Managed by Slurm - workload manager and job scheduler
- Lots more information - Viking Wiki pages

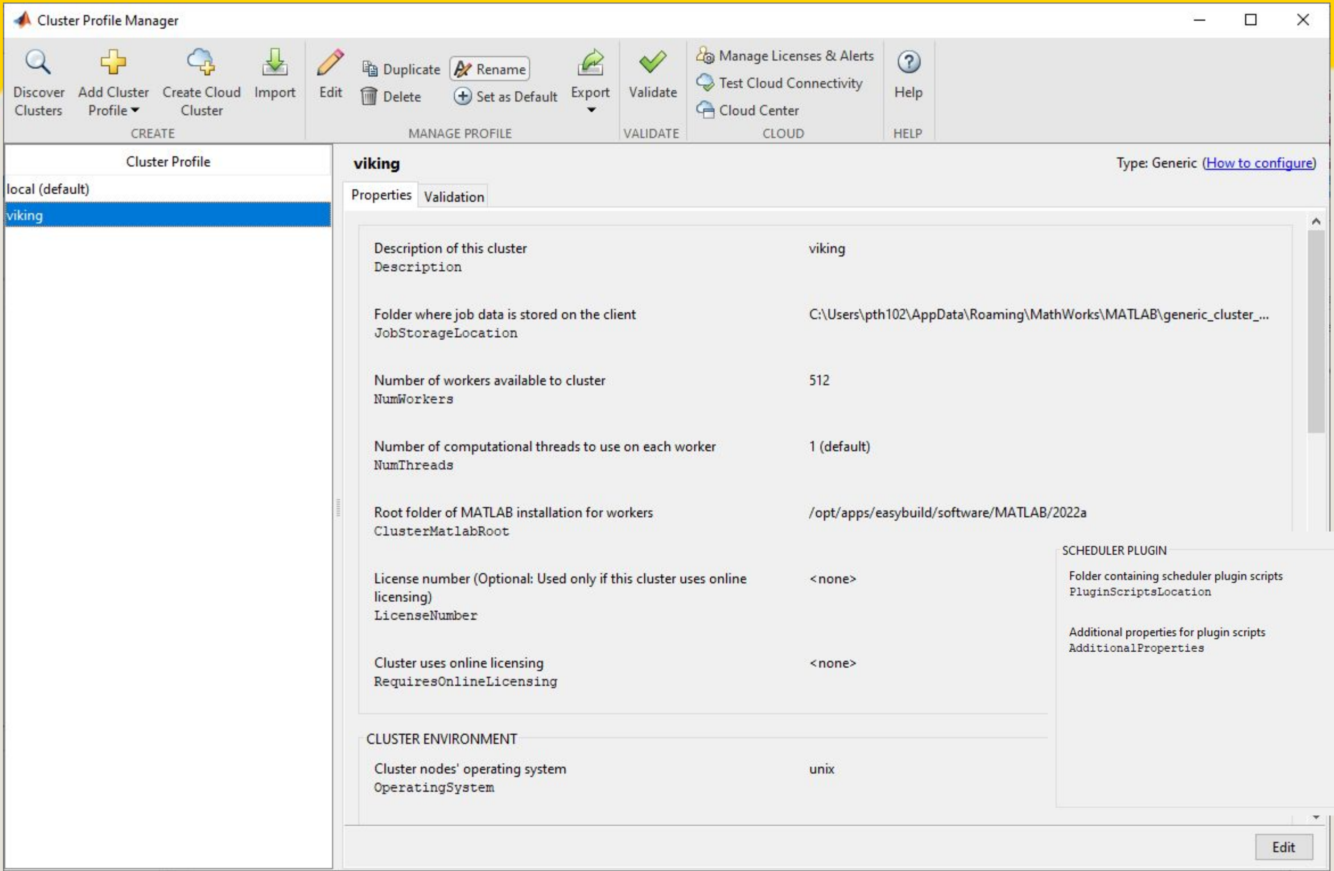
Viking & MATLAB Prerequisites

- Need to request access - user to complete the Viking User Application Form
- But need a project code first - supervisor/PI completes the Viking Project Application Form
- Can only connect to Viking from on-campus or via the VPN
- MATLAB Parallel Computing Toolbox on local MATLAB instance
  - Installed by default on managed devices and via Software Center
  - Personal devices - make sure to add it
- Local MATLAB version that matches a version on Viking
  - Currently: 2018a, 2020a, 2020b, 2021a, 2022a

Cluster Profiles

- Need a Cluster Profile to tell MATLAB how to communicate with Viking
- Automatically generated using scripts - download scripts from the Viking MATLAB Wiki page
    - Only tested on Windows so far, but should work for Linux and Mac
- Put scripts on local machine, make sure the location is in MATLAB's path, e.g.
    - `addpath(genpath('C:\Users\abc123\Documents\MATLAB\Viking'))`
- Run creation script
    - `configCluster`
- Only information required is University username, e.g. abc123
- Creates a cluster profile called 'viking'

Cluster Profiles



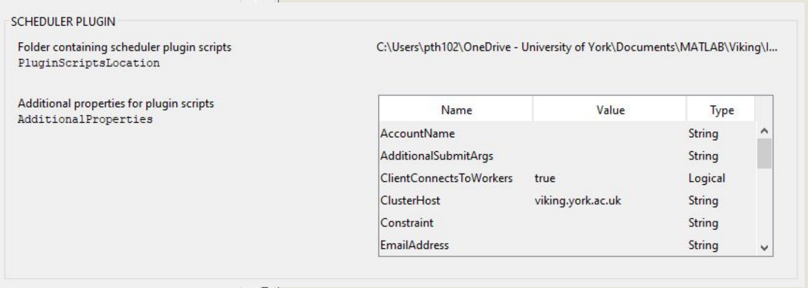
viking

Type: Generic (How to configure)

Description of this cluster `Description`	viking
Folder where job data is stored on the client `JobStorageLocation`	C:\Users\pth102\AppData\Roaming\MathWorks\MATLAB\generic_cluster_...
Number of workers available to cluster `NumWorkers`	512
Number of computational threads to use on each worker `NumThreads`	1 (default)
Root folder of MATLAB installation for workers `ClusterMatlabRoot`	/opt/apps/easybuild/software/MATLAB/2022a
License number (Optional: Used only if this cluster uses online licensing) `LicenseNumber`	<none>
Cluster uses online licensing `RequiresOnlineLicensing`	<none>

CLUSTER ENVIRONMENT

Cluster nodes' operating system `OperatingSystem`	unix



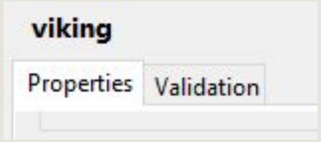
SCHEDULER PLUGIN

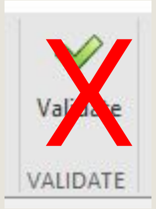
Folder containing scheduler plugin scripts `PluginScriptsLocation`	C:\Users\pth102\OneDrive - University of York\Documents\MATLAB\Viking\I...
Additional properties for plugin scripts `AdditionalProperties`	

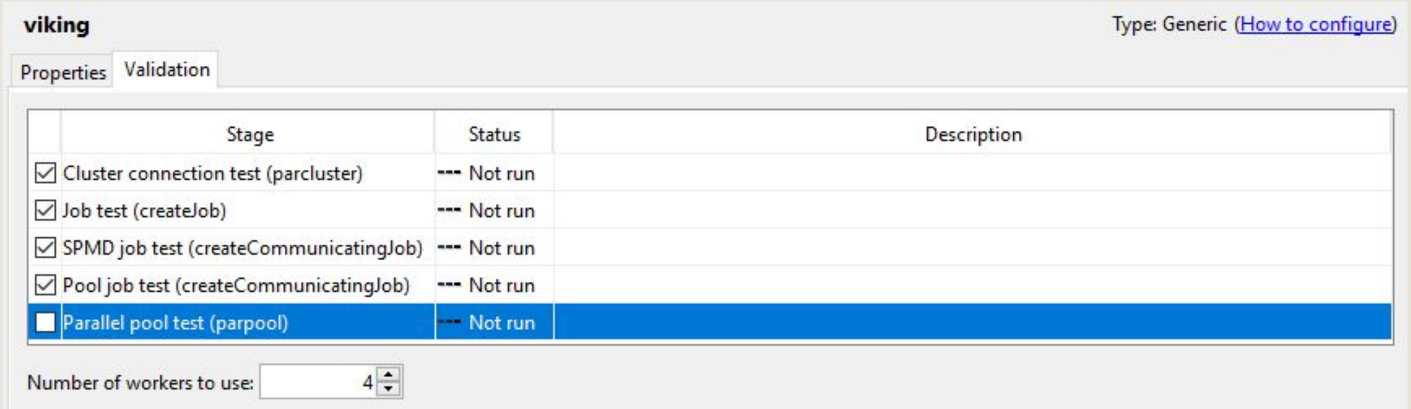
Name	Value	Type
AccountName		String
AdditionalSubmitArgs		String
ClientConnectsToWorkers	true	Logical
ClusterHost	viking.york.ac.uk	String
Constraint		String
EmailAddress		String

Cluster Profile Validation

- Need to Check that the profile, your account and connection are working
- Parallel > Create and Manage Clusters
- Click 'Validation' tab not the 'Validate button'
- Update 'Number of workers to use' to 4
- Untick 'Parallel pool test (parpool)






viking

Type: Generic (How to configure)

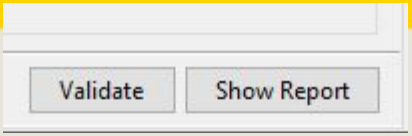
Properties | Validation

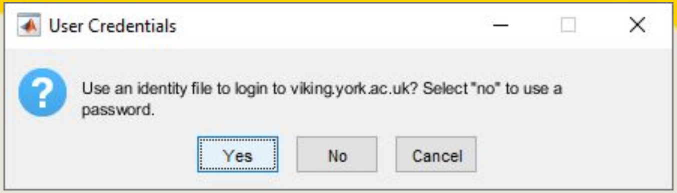
	Stage	Status	Description
☑	Cluster connection test (parcluster)	--- Not run	
☑	Job test (createJob)	--- Not run	
☑	SPMD job test (createCommunicatingJob)	--- Not run	
☑	Pool job test (createCommunicatingJob)	--- Not run	
☐	Parallel pool test (parpool)	--- Not run	

Number of workers to use: 4

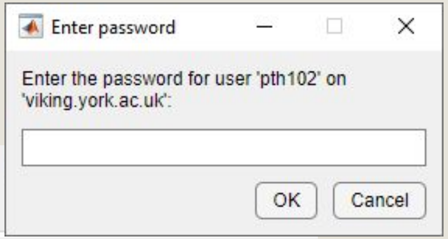
Cluster Profile Validation

- Click 'Validate' button
- Will be asked if using an identity file: No
- Will be asked for your university password
- Then wait! Successful validation looks like:

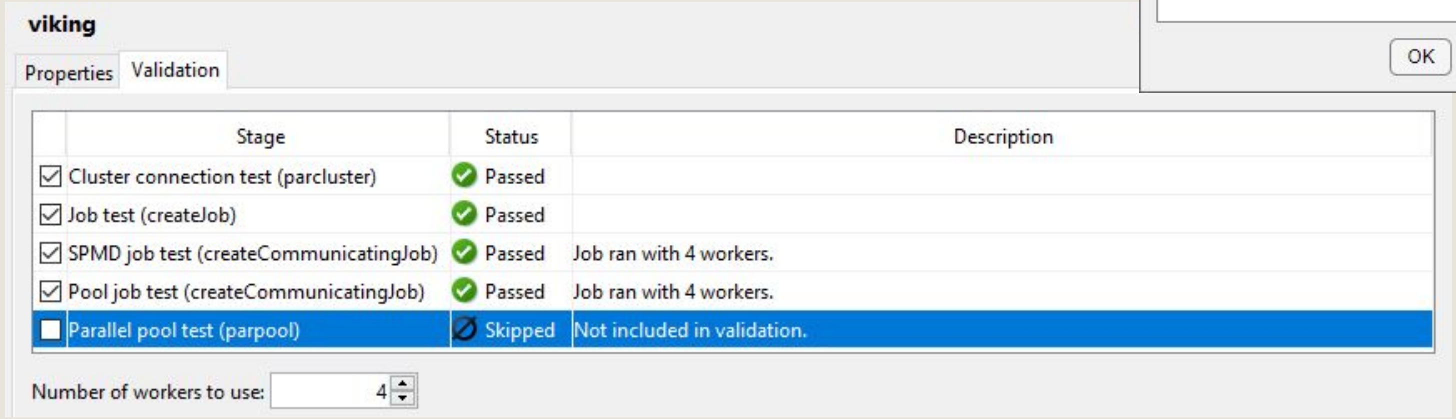



User Credentials

Use an identity file to login to viking.york.ac.uk? Select "no" to use a password.


Enter password

Enter the password for user 'pth102' on 'viking.york.ac.uk':


viking

	Stage	Status	Description
☑	Cluster connection test (parcluster)	✓ Passed	
☑	Job test (createJob)	✓ Passed	
☑	SPMD job test (createCommunicatingJob)	✓ Passed	Job ran with 4 workers.
☑	Pool job test (createCommunicatingJob)	✓ Passed	Job ran with 4 workers.
☐	Parallel pool test (parpool)	Skipped	Not included in validation.

Number of workers to use: 4

Demo time…

Useful commands

- `c = parcluster('viking')` - creates a cluster object using the viking profile
- Modify and add properties with:
    - `c.AdditionalProperties.NumNodes = 1;`
    - `c.AdditionalProperties.ProcsPerNode = 9;`
- Submit jobs with `batch`
    - `myjob = batch(c, 'scriptname', 'pool', 8)`
    - Number of procs/workers requested must be 1 greater than specified with `pool`
    - Scripts are on local device and sent to Viking
    - Mathworks `batch` examples

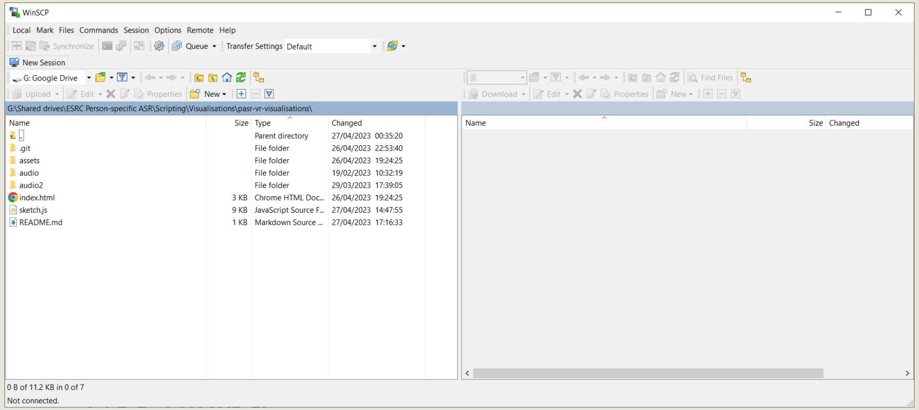
Useful commands

- `diary(myjob)` - returns elapsed time
- `load(myjob)` - loads all workspace variables from specified job
- Job Monitor (Environment Toolbar > Parallel > Job Monitor)
  - Shows status of jobs
- Can submit job(s) then close MATLAB and jobs will run on Viking
- After reopening MATLAB get the results back using either:
  - Right-click on job ID in the Job Monitor window > Load Variables
  - Or

```
c = parcluster('viking');
job8 = findJob(c, 'ID', 8);
load(job8);
```

Getting data on and off Viking

- Windows: WinSCP
  - In Software Center on managed PCs
  - Personal devices download from https://winscp.net/eng/download.php
- Mac: Filezilla
  - Download from: https://filezilla-project.org/
- Use /scratch folder to store data
  - Fast
  - No limit on number of files
  - 3 TB by default
  - WARNING - scratch is not backed up



G:\Shared drives\ESRC Person-specific ASR\Scripting\Visualisations\pasr-vr-visualisations\

Name	Size	Type	Changed
..		Parent directory	27/04/2023 00:35:20
.git		File folder	26/04/2023 22:53:40
assets		File folder	26/04/2023 19:24:25
audio		File folder	19/02/2023 10:32:19
audio2		File folder	29/03/2023 17:39:05
index.html	3 KB	Chrome HTML Doc...	26/04/2023 19:24:25
sketch.js	9 KB	JavaScript Source F...	27/04/2023 14:47:55
README.md	1 KB	Markdown Source ...	27/04/2023 17:16:33

Name	Size	Changed

0 B of 11.2 KB in 0 of 7
Not connected.

Final comments

- Just an outline and introduction
- Lots of options and configurations
- Experiment - find out what works/doesn't work
- Viking 2 is coming later in the year
    - Hopefully with automatic cluster discover!
- Using Linux commands are a useful complement to check on job progress, files etc

Sources of help & information

- Research Coding Club Slack channel and drop-in sessions
  - Webpages with previous talks - e.g. Parallelisation
- Email to IT Support - itsupport@york.ac.uk
- Mathworks help - Techniques for Improving Performance
- MATLAB training:
  - MATLAB Onramp (2 hours)
  - MATLAB Fundamentals (16.5 hours)
  - MATLAB for Data Processing and Visualization (8 hours)
  - MATLAB Programming Techniques (16 hours)
  - Object-Oriented Programming Onramp (2 hours)

Questions?