MATLAB on TRUBA: Q&A

What if I do not have a Parallel Computing Toolbox license, could I still see the parallel option?

You will need Parallel Computing Toolbox installed and licensed to run parallel code. See your MATLAB license administrator for more information on installing and licensing Parallel Computing Toolbox.

Are Fuzzy Logic Toolbox for ANFIS training and Simulink supported on TRUBA?

Yes. All MathWorks products are supported on TRUBA.

Our Python code invokes MATLAB as follows:

```
import matlab.engine

eng = matlab.engine.start_matlab()
```

Does the MATLAB license provided on TRUBA support the MATLAB Engine for Python interface when called from within a Slurm batch script?

Yes

What is the recommended method for installing the matlabengine Python package? Should we install it inside a container (Apptainer) or within a virtual environment under /arf/scratch?

See https://www.mathworks.com/matlabcentral/answers/2086093-can-i-use-the-matlab-engine-api-for-python-with-a-virtual-environment for how to install MATLAB Engine API for Python. Contact the TRUBA help desk for more information.

Is it possible to use TRUBA ARF with any 3rd party application like ANSYS or ABAQUS called from within MATLAB?

All academics have access to MATLAB running on TRUBA. Depending on the 3rd party application, you might need to be properly licensed to run it on TRUBA. Examples include ANSYS and Abaqus.

How do we request the orfoz partition to run MATLAB Parallel Server jobs?

```
c.AdditionalProperties.Partition = "orfoz";
```

Is there any good way to estimate wall time beforehand?

Initially, we can guestimate that the code will run no longer than it ran on your local machine. To get a better approximation, you will need to time how long it takes your serial code to run plus the amount of time it takes your parallel code to run divided by the number of requested cores. For example,

$S_t + P_t/nW$

Where

S_t is the amount of time it takes to run all serial code (except parallelizable code)

P_t is the amount of time it takes to run code that could be parallelizable (e.g., for-loops that could be rewritten as parfor-loops, GPU code, etc.)

nW is the number of workers that will run the parallel code

Is it better to run many small simulations at the same time or one large simulation in HPC?

There should be a balance between the number of workers requested and the amount of time each worker will run for.

Running a significant number of workers, each running a trivial amount of work is not efficient. It takes a longer time to start a job with many workers and the overhead of starting the workers may outweigh the performance.

Running the code with only one worker may start sooner, but you're limiting the resources and, in the end, may take longer to complete.

How do I debug the following error message?

Error using parallel.Job/fetchOutputs (line 1411)

Job failed to run and did not return a specific error message. This is likely due to the scheduler failing to start MATLAB correctly on the cluster or the unavailability of the files required for job execution on the cluster. For more information, use the getDebugLog function to read the debug log for the job.

Run

```
c.getDebugLog(job)
```

How do I debug the following error message?

This job was submitted with version 25.1 of the Parallel Computing Toolbox. However the current MATLAB Parallel Server is version 25.2. You must run jobs from clients on servers with the same version."

Users running MATLAB R2025b must install at least Update 2. To verify, run the following

```
matlabRelease
```

The MATLAB code that I want to run uses large files that are already on TRUBA in scratch. Rather than transferring the files with each job submission, is it possible to tell MATLAB to use the files on TRUBA instead of transferring them from my local machine?

For each job submission, MATLAB will transfer over MATLAB files (e.g., .m, .mlx, .mat, .slx, etc.). External files (e.g., jpeg, csv, etc.) either need to be explicitly added to the job or already exist on TRUBA.

For local files, use the *AttachedFiles* parameter of batch for a small number of small files.

For large files, or many files, stage the files on TRUBA and parameterize your MATLAB code, pointing to where the files are located (rather than hardcoding paths to file locations). This will require that your top-level MATLAB file is a function and not a script.

For instance, assume your function needs to read in a file from a specific location.

```
function myCode(folder_location)
if nargin==0
   folder_location = pwd;
end
X = imread(fullfile(folder_location,"image.jpg"));
. . .
```

Now, to submit this job, run the following

```
job = c.batch(@myCode,0,{"/arf/scratch/my-username/images"});
```

This way we can run the code locally and not be required to specify the folder location or run it on the cluster and specify where the files have been hosted.

Reinforcement Learning Toolbox supports a parallel option. Is TRUBA suitable for RL + parallel computing? Are there any adjustments needed in the settings?

If you run MATLAB on TRUBA, then start a parallel pool of workers and read more about training options https://www.mathworks.com/help/reinforcement-learning/ug/train-agents-using-parallel-computing-and-gpu.html

If you are running MATLAB on your local desktop, then contact the TRUBA help desk for more information on how to use the MATLAB `batch` command to submit your jobs to TRUBA.

Is it free to use GPU on TRUBA? Do we need additional permissions to use the Cuda partitions from TRUBA?

You can use the barbun-cuda and akya-cuda partitions on ARF cluster. Cuda partitions are for projects for now. See more on https://docs.truba.gov.tr. Alternatively, you may want to purchase it if you want to use it.