Example MATLAB main script: “**ZZvsModel.m**”

The link below contains the basic steps of running MATLAB on the CHTC.
https://chtc.cs.wisc.edu/matlab-jobs.shtml

There are two parts to running MATLAB on the CHTC:

1. Compilation
2. Submission

Two things are needed for the compilation step.

1. Main MATLAB script/function
2. Functions folder

The first is simply the main MATLAB script to be run. Note that script files should be converted to function files if multiple inputs or combinations of inputs are to be put in to make use of the parallelism of the CHTC. Make sure to **convert input variables from str type to num type with str2num()** since all inputs will be in str. Also, remove all semicolons for the final output variables as they do not automatically appear in the CHTC output, but rather need to be printed to the command line.

‘Functions’ is a folder that contains all the functions required to run the main script. There are some key things to note about this:

- Any external toolbox used needs to be converted to a folder containing all its functions (in MATLAB, Add-Ons -> Manage Add-Ons -> right click ToolBox -> open folder). These need to be added to an existing directory called ‘functions’ (can be renamed but this is to suit the above example code)

- Add all other functions required to run the main script to the ‘functions’ folder

- Remove all non-functions from the folder. For example, in the HHXT MATLAB toolbox, remove the ‘doc’ and ‘resources’ directories.

- The structure of this folder doesn’t matter. The CHTC will recursively search over subfolders and compile all the functions it finds in the compiling step.

The compilation step is described in the link above. Refer to https://github.com/IanJentz/HHXT/blob/master/Jeremy/ZZvsModel_2/buildMatlab.sub for an example of a build file to get to the Compilation step. To summarize the Compilation step:

1. With the build file ready, run the following in the CHTC terminal:

```
condor_submit -i build.sub
```

Example build file:

2. Then run the following:

```
module load MATLAB/R2018b

mcc -m -R -singleCompThread -R -nodisplay -R -nojvm ZZvsModel.m -a functions

exit
```

Note that this is specific for running ‘ZZvsModel.m’. Change that part of the code as needed.

After exiting, a .sh file will be created. If the main MATLAB script is ‘ZZvsModel.m’, this file will be named ‘run_ZZvsModel.sh’. Open this file and add the following:]

```
tar -xzf r2018b.tar.gz

mkdir cache

export MCR_CACHE_ROOT=$PWD/cache
```

Next, moving on the submission step:

1)

Prepare a csv or txt file with all the input combinations. For example, ZZvsModel has 2 inputs: j_C and f_D (in this order)

In this case, the csv file might look like:

```
0, 0.1, 0.3
1, 0.2, 0.3
2, 0.1, 0.4
3, 0.2, 0.4
```

Note that header information is not needed. Make sure the order of inputs in a row matches the order of the function call (not necessary but makes things more simple). The first column refers to the row number, which will help with the outputs later.

2)

Prepare a Submit file (example: https://github.com/IanJentz/HHXT/blob/master/Jeremy/ZZvsModel_2/submitMatlab.sub)

In this file, note that there is a 'transfer_input_files' row. Make sure to add in:

1. The compiled script (for ZZvsModel.m, this will be called 'ZZvsModel')

2. Any .mat files needed to run the main script

3. http://proxy.chtc.wisc.edu/SQUID/r2018b.tar.gz (as a text)

3)

At the bottom of the submit file, queue arguments/inputs from the input text file with the following:

'queue run, j_C, f_D from input_data.txt'

The input 'run' will be helpful in varying the output data.

4)

Edit the arguments line as needed. In the ZZvsModel.m example:

'arguments = v95 $(run) $(j_C) $(f_d)'

5)

Formatting the outputs:

'output = OUTPUT/run$(run).out'

Where, OUTPUT is an existing directory

For any other detail, refer to https://chtc.cs.wisc.edu/helloworld.shtml.

Once the submit file is ready, run:

```
condor_submit submitMatlab.sub
```

Where submitMatlab.sub is the submit file. Run 'condor_q' to check the progress of your nodes.

Handling Outputs

- Outputs can be stored in a .mat file (from the main script) but this must be executed from the MATLAB side of things.

- Outputs also appear in a txt file from the CHTC, but the format is often quite messy. What appears in the CHTC output file is whatever appears in the command line in MATLAB.

- Use a python script or module to clean and save the data to a csv. An example of this can be found in

https://github.com/IanJentz/HHXT/blob/master/Jeremy/ZZvsModel_2/Cases/run_020/Create_Dataframe_ZZvsModel_2.ipynb

Other

- Avoid generating plots for now. Will try to find a way around this soon.