Background

The original documentation for CitcomCU can be found here.

This documentation will walk you through

- Installation
- Input File
- Provides and Example from Busse et al. [1993]

Running on NOTS

For the purposes of this document I will assume that I am running a model that I will call **ModelA1**. This model will have an input file (*input.ModelA1*) and a submit file (*quefile.ModelA1*).

Submitting a job

To submit a job, copy the input and submit files to the work directory

```
foo@bar:~$ scp input.ModelA1 quefile.ModelA1 user@nots.rice.edu:$WORK/user/
```

Using a terminal of your choice, log into nots and navigate to the scratch directory. (NOTE: If off campus, you will need a VPN connection.)

```
foo@bar:~$ ssh -Y user@nots.rice.edu
user@nots's password: PASSWORD
user@nots.rice.edu:~$ cd $WORK/user
```

Create a folder that will hold the model outputs

```
user@nots.rice.edu:~$ mkdir CaseModelA1
```

Submit the model to run

```
user@nots.rice.edu:~$ sbatch quefile.ModelA1
```

Resubmitting a job

Resubmitting a job in NOTS can be done a few different ways. The first is to find the last time step of the previous run and changing it in the input file. This can be accomplished using the command

```
user@nots.rice.edu:~$ ls/CaseModelA1 *temp.1.*
```

And noting the largest time step. Alternatively, you can simply uncomment the penultimate line within the quefile. When the quefile is submitted, the program will be called from $HOME (NOTE: make sure to update the path). This script will automatically update the last time step).

Diagnostics, Visualization and Analysis

This section assumes that you have the visualization scripts loaded onto NOTS. If you do not, create a folder within your projects direction on NOTS using the commands

```
user@nots.rice.edu:~$ cd $PROJECTS/ajns/user/
user@nots.rice.edu:~$ mkdir PythonScripts
```

Upload the scripts from your local computer

```
foo@bar:~$ scp PythonScripts/* user@nots.rice.edu:$PROJECTS/ajns/user/PythonScripts/
```

You should now have the scripts on NOTS and be ready to proceed.

Checking for steady state

The first step is to save a copy of the input file to the projects directory.

```
user@nots.rice.edu:~$ cd $PROJECTS/ajns/user/
user@nots.rice.edu:~$ cp $WORK/user/input.ModelA1 .
```

To check for steady state, we will extract a time series of the top and bottom nusselt numbers using the commands

```
user@nots.rice.edu:~$ python PythonScripts/CitcomCU6b.py -t 'timehistory' -d 'nusselt' -f
'input-ModelA1'
```

This will create a folder *CaseModelA1* within the current working directory and store the output file *output-Model1A.nusselt.dat*

In my workflow, I copy this output file to my local computer and analyze it there using

```
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/CaseModel1A/*nusselt* .
foo@bar:~$ python plotnu.py output-Model1A
```

This script produces a file (output-Model1A.nusselt_SORTED.png) that you can open and check whether the model is in a (statistical-)steady state.

Gathering Outputs

Once the model reaches a steady state, you can gather the remaining diagnostics using the following code within the NOTS project directory

```
user@nots.rice.edu:~$ python PythonScripts/CitcomCU6b.py -t 'timehistory' -d 'all_data' -f
'input-ModelA1'
```

This command will produce four outputs

- output-Model1A.meta.ave.dat -> time series of top and bottom nusselt number
- output-Model1A.temp.ave.dat -> geotherm at each time step
- output-Model1A.nusselt.dat -> time series of top and bottom nusselt values
- output-Model1A.vrms.ave.dat -> depth-averaged velocity profile for each time step

After transferring each of the **.dat* files to your local workspace, you can process the *meta*, *temp* and *vrms* averages using the Jupiter notebook **Depth_Profiles.ipynb**

Visualizing in 3D

To transform the CitcomCU output into output useable in Paraview, we can do the following:

Confirm that the needed script is in the projects folder or copy them from your local computer using the command

```
foo@bar:~$ scp cicomcu_write_vtk cu2vtk  user@nots.rice.edu:$PROJECTS/ajns/user/
user@nots's password: PASSWORD
```

Transfer the coordinate data from the run directory to the projects directory

```
user@nots.rice.edu:~$ cd $PROJECTS/ajns/user/CaseModelA1
user@nots.rice.edu:~$ cp $WORK/user/CaseModelA1/*coord* $PROJECTS/ajns/user/CaseModelA1/
```

Transfer whatever time steps you will be analyzing in 3D. For example, if you want to evaluate the last time step, and it is has a value of 105000, you will transfer it using the command

```
user@nots.rice.edu:~$ cp $WORK/user/CaseModelA1/*105000* $PROJECTS/ajns/user/CaseModelA1/
```

To transform this output from within the CaseModelA1 folder, use the command

```
user@nots.rice.edu:~$ .../citcomcu_write_vtk 'output-Model1A' 16 105000
```

Where I have assumed that 16 processors were used in the calculation. To transform more than one time step, transfer the needed model output files to the working directory and use the script *cu2vtk*.

Once you have transformed the data you can copy it to your local desktop using the command

```
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/CaseModel1A/*vtr .
user@nots's password: PASSWORD
```

And then use Paraview to visualize the results.

Plume analysis

This type of analysis was based on Leng and Zhong [2008]. The anaysis can be performed by:

- Loading model PVD into Paraview.
- Creating a slice at the desired depth
- Clicking *File* > *save data* and save as *data.csv*
- Analyze in *PlumeAnalysis.ipynb*

NB: I recommend extending the code to analyze the last 30 time steps and multiple, if not all depths to get an accurate picture.

Data Storage

You should save the:

- Input file
- quefile
- Last time step
- Coordinate files

Which can be accomplished using

```
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/input-ModelA1 .
user@nots's password: PASSWORD
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/quefile-ModelA1 .
user@nots's password: PASSWORD
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/CaseModelA1/*coord* .
user@nots's password: PASSWORD
foo@bar:~$ scp user@nots.rice.edu:$PROJECTS/ajns/user/CaseModelA1/*105000* .
user@nots's password: PASSWORD
```