FLUENT LES Simulations on Oswald

Ravindu Ranaweera

Department of Mechanical and Civil Engineering

University of Northumbria at Newcastle

This is a guide for;

The submission and simulation of an LES case on ANSYS FLUENT using the HPC (Oswald)

Northumbria University 2022

Contents

List of Figures

Chapter 1 Introduction

This document is a general guide detailing the process of submitting and running a job on Oswald, the HPC at Northumbria University at Newcastle. It utilises a simple but a complete method for running a successful LES simulation on an HPC (running CentOS) running SLURM workload manager. The script may need to be developed by the user for additional user specific requirements.

High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business. For more information on Oswald and SLURM, please refer to Oswald documentation and SLURM documentation.

The process of simulating a case on Oswald would include the following steps;

- Creating an SSH session with Oswald using a terminal.
- Creating “ready-to-run” case files on ANSYS FLUENT locally (PC).
- Creating supporting journal and script files which executes fluent and simulates the case as per instructed.
- Submit the job on to the queue for simulation.
- Download the completed simulation files from the cluster.

1.1 Prerequisites

1.1.1 Oswald Account

To access Oswald, the user would first need to set-up an account on Oswald. This can be done by contacting the HPC/Linux Technology Specialist via e-mail or raising an IT ticket.

Once the registration/induction is complete, the user will receive a unique username and a password (which should be changed on first login).

1.1.2 SSH and SFTP Compatible Terminal

The cluster would be operated remotely via command line using the Secure Shell Protocol (SSH) and use the Secure File Transfer Protocol (SFTP) to transfer files to and from the cluster. Therefore, the user will require an SSH and SFTP compatible client such as MobaXterm (used in this guide), PuTTY or similar software on Windows computers. The native Terminal on Mac OS and Linux operating systems will support SSH and SFTP by default.

1.1.3 Connection to the Cluster

Oswald can only be accessed from the university network. It cannot be accessed from outside connections. Therefore, the user must access from a computer on the university network. Alternatively, VPN or Remote Desktop solutions (Such as MyAccess via Citrix or other approved RDP solutions) can be used.

1.1.4 ANSYS Fluent on a PC

Although a complete case can be constructed, simulated and post processed on the cluster via ANSYS Text User Interface (TUI), a command-based user interface, it can be very complex. Hence, it is advisable to set-up the case on a PC using the graphical user interface of FLUENT.

It is advisable to use the same version of FLUENT on both the cluster and the PC to avoid incompatibility issues.

1.1.5 Fluent Case File

Please create a "ready to calculate" case file from FLUENT on the PC. Ensure the case is setup with all the required models activated, boundary conditions set, result outputs figures/files/animations configured, and any other configurations have been set.

Care should be taken to avoid absolute paths, instead, relative paths should be used to avoid errors and loss of outputs (such as animation frames, data files, autosaves, etc.).

Save the case and data files in the .cas (or .cas.h5 in newer versions of FLUENT) format along with the data file following the same naming convention (except for the format .dat)

It is advisable to create a folder and keep these files together as more files will be added (such as the journal file and the job submission script).

1.1.6 Code Editor (Visual Studio Code)

A code editor is required to in case the user is on a Windows operating systems. The recommended editor is Visual Studio Code (VS Code), which can be installed without admin privileges on the university computers under student user accounts.

VS Code is required because operating systems have special characters denoting the start of a new line. For example, in Linux a new line is denoted by "\n", also called a Line Feed (LF). In Windows, a new line is denoted using "\r\n", sometimes called a Carriage Return and Line Feed (CRFL). Since Oswald runs Linux, the job scripts detailed in section 1.1.8 require the Linux notation of a new line. Code-editor such as VS Code allows users to switch between the new line notations.

Please follow the steps found on the documentation found at https://code.visualstudio.com/docs/setup/windows to install VS Code.

Linux and MAC users can use VS Code or the native text editor.

Alternatively, the built in text editor on MobaXterm or the "nano" text editor on the terminal can be used to draft scripts but may not be as powerful as a dedicated code-editor such as VS Code.

Once installed, when editing scripts for Oswald, please switch to "LF" from "CRLF" from the bottom right of the screen as shown in Figure 1.1: Change to LF on VS Code.

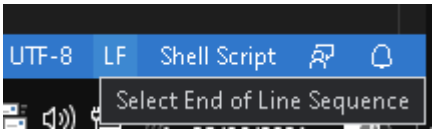

Figure 1.1: Change to LF on VS Code.

1.1.7 TUI Journal File

As the simulation process will be run in headless mode (or batch mode), a journal file consisting of all the commands needed to execute the case must be created. This Text User Interface (TUI) file will replace the traditional inputs in the Graphical User Interface (GUI).

Please refer to Appendix A for an example journal file for simulating an LES case. The lines preceding ";" are comments inserted to explain the following command to the reader. The case name, number of iterations and timestep size should be changed as required by the user.

This file must be placed in the same parent directory as the case and data files created in section 1.1.5.

For additional functions, please refer to the ANSYS TUI documentation.

1.1.8 Job Submission Script

Oswald utilises the SLURM workload manager to schedule in jobs as per the resource availability and priority. SLURM is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. To submit a job

via SLURM, a job script detailing the resources required, job information, the list of commands to execute, etc. is required.

Please refer to Appendix B.1 for an example job submission script for submitting and simulating a LES case. The lines preceding ”#<space>” and “##” are comments, not to be confused with “#SBATCH” which are part of the script. The file must be saved as a “.sh” file.

The user may change the job name, maximum iteration time, submission queue, e-mail address, nodes, tasks per node, journal filename as required.

Please Note: You must use the “LF” option on VS Code or your code-editor as explained in section 1.1.6.

Chapter 2 Configuring MobaXterm

MobaXterm is an enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more. It has an easy-to-use GUI and tools that would help the user interact with Oswald.

For more information and help on MobaXterm, please use the documentation and resources found on the website.

This guide will focus on setting up and using MobaXterm as it provides much easier SSH and SFTP access is a user-friendly manner. If you are using Putty, please check their documentation and guides and use the aforementioned settings for MobaXterm in their respective fields. Alternatively, if you are logging in from Linux, the type the below command on a Terminal window and press the return key;

```
ssh <username>@oswald
```

2.1 Installing MobaXterm

Please download the latest version (portable recommended) of the MobaXterm Home Edition from https://mobaxterm.mobatek.net/download-home-edition.html. Once downloaded, extract the Zip file to a preferred location (works from OneDrive synced folders as well as external storage devices as well).

2.2 Configuring MobaXterm

Please follow the steps outlined below to connect to create a SSH connection with Oswald.

1. Please launch “MobaXterm”. You will be presented with a similar screen as shown in Figure 2.1: Launch and creating a session on MobaXterm.

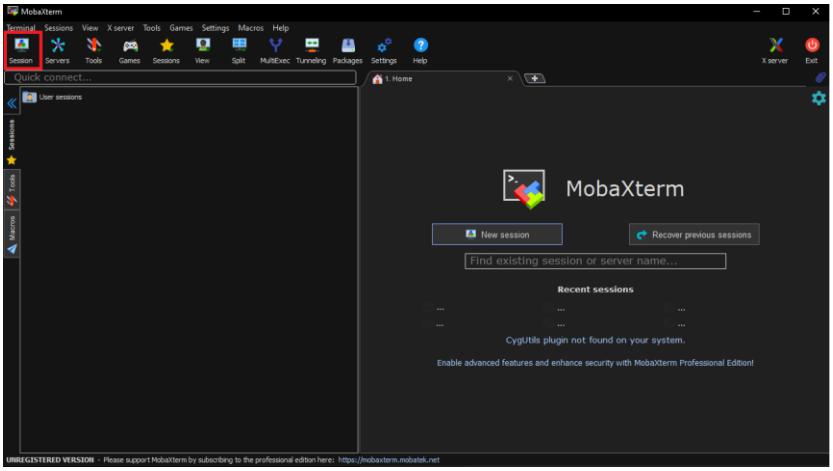

Figure 2.1: Launch and creating a session on MobaXterm.

2. Click on the “Sessions” button found on the top left of the window as marked in Red on Figure 2.1: Launch and creating a session on MobaXterm.
3. Click on the “SSH” button on the subsequent windows.
4. Select the “Specify Username” and entre your assigned username in the field next to it. The settings should look similar to the Figure 2.2: Setting for the SSH session.

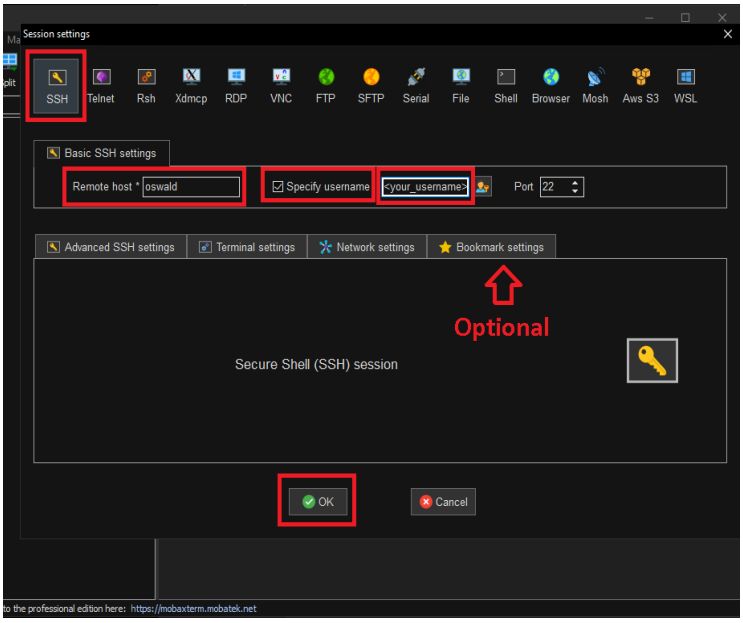

Figure 2.2: Setting for the SSH session.

5. You can choose to create a shortcut of your session on your desktop by selecting the "Bookmark settings" tab and clicking the "Create a desktop shortcut to this session".
6. The session can be given a customised name under the "Session name" as well. Click on the "Ok" button to start the session. You will now be asked to log in. This may take a few minutes depending on the internet connection and the status of Oswald.

   Please Note: When entering the password on the terminal, it will not be visible (no asterisks, etc.). Type the password and press the return key.
7. Once logged in, you will see a screen similar to Figure 2.3: Oswald session welcome screen. with any messages from system admin and your last log-in timestamp.

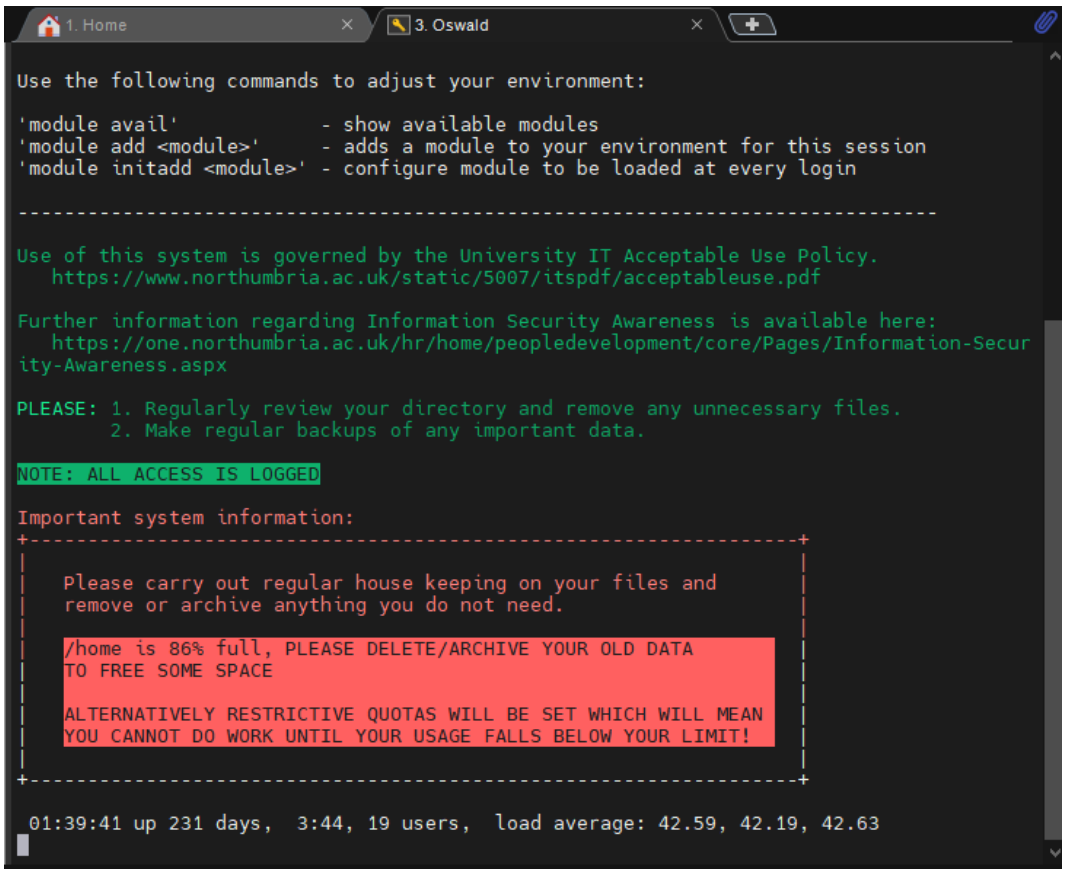

Figure 2.3: Oswald session welcome screen.

If you see any discrepancies, immediately change your password using the "passwd" command and notify the system admin.

2.3 Uploading Case Files to Oswald

For simplicity, this guide will follow the graphical SFTP protocol to manage files on the cluster. Putty and Linux users can use the traditional SFTP commands or setup a map/mount a remote drive using the SFTP protocol.

1) It is advisable to create a folder to upload your case under your "home" directory. This can be done using the traditional "mkdir <foldername>" command or the much simpler SFTP interface found next to the Terminal on MobaXterm.
2) Once the case folder has been setup, double click on it and navigate into it.
3) Now, simply drag and drop the fluent case files from your desktop into the SFTP area in your MobaXterm.
   The upload should automatically begin. The duration of the upload will depend on the file sizes and the network connection between your PC and Oswald.
8. Ensure you upload the job script and the TUI script onto the same folder as well.

2.4 Submitting Jobs to the Job Queue

One a case has been uploaded to the cluster using SFTP as detailed in section 2.3, please follow the steps below to submit the "job" to the "queue".

1) Using the "cd" (change directory) command, navigate to the folder where case files, journal files, job scripts, etc. are stored.
2) Now use the following command to submit the job to the queue.

   ```
   sbatch fluent_job.sh
   ```

3) Within a few seconds, you will see a job number for the submitted job on the queue.
4) You can view the queue by typing "squeue".
   Once the job starts running and upon completion, you will receive a notification on the nominated e-mail.
   You can also view the information on jobs using "sinfo <job_ID>" and cancel jobs by using "scancel <job_ID>".

2.5 Downloading Results

Once the simulation starts running, a series of files will be generated as a result of the FLUENT simulation and SLURM job execution (log files). The status of the simulation can be ascertained from double clicking and reading the “fluent.log” file.

Alternatively, the “tail fluent.log” (or the chosen log filename) command can be used to view the last 10 lines of the file in the terminal window.

Once the simulation has completed, you will receive a notification via e-mail. Unless you are doing further post processing on the cluster, results can be downloaded in a similar manner to section 2.3.

Appendix A Journal File for LES Simulation

The following journal file can be used to read, initialize from inlet, simulate and save data for a LES simulation case file on FLUENT. Please copy the below TUI commands onto a plain text editor such a Notepad or code editor such as VSCode and save as “run_case_les.jou” case sensitive). Alternatively, source code editors such as Visual Studio Code can be used.

```
;Run LES Case
;Journal file (TUI commands) for fluent LES case execution on a HPC (Linux) by Ravindu
;Ranaweera (https://github.com/ItsJustRav)
;Tested on ANSYS2021R1
;Lines starting with ";" are comments
;Batch loading options (confirm overwrite, hide questions, exit on error, re-display question)
/file/set-batch-options no yes yes no
;Read case and data
/file/read-case-data AB25_Iso_FW_init_2021R1_les_anim.cas.h5
;Mesh info and memory usage
/mesh/memory-usage
;Initialize using velocity-inlet <inlet_name>
/solve/initialize/compute-defaults/velocity-inlet velocity_inlet
/solve/initialize/initialize-flow
;File saving frequency and file naming
/file/auto-save data-frequency 1
/file/auto-save append-file-name-with time-step 6
;Timestep size
/solve/set/time-step 0.01
;Timesteps and max iterations per timestemp
/solve/dual-time-iterate 400 25
;Write the final output (also there with autosave)
/file/write-case-data AB25_Iso_FW_les_out.dat
;Performance information
/parallel/timer/usage
exit
```

Please refer to https://bit.ly/2SnMcoa for a copy of this script, sample RANs job submission and complete case build journals.

Appendix B Job Submission Script for LES Simulation

The following job submission script allows the user to run submit a job to Oswald HPC and execute the commands required for running FLUENT via command line and simulate in parallel using a journal file found in Appendix A. Please copy the script to a new file made using a VS Code (with “LF” as discussed in section 1.1.6) and save as “fluent_job.sh” (case sensitive).

B.1 Generic LES Job Script

```
#!/bin/bash

# Job Submission Script
# Job Submission Script for running ANYS Fluent CFD Case on an HPC Cluster running Linux
# and SLURM Workload Manager by Ravindu Ranaweera (https://github.com/ItsJustRav)
# Commands have been commented out with "##". Un-comment as needed.

# ###############################################################################
# Job Information
#SBATCH --job-name="<job_name>"                    ## Job name
#SBATCH --time=48:00:00                              ## Time limit hrs:mm:ss
#SBATCH --nodes=1                                    ## Number of nodes
#SBATCH --ntasks-per-node=56                         ## Number of tasks per per node
#SBATCH --error=error_log_%j.log                  ## Error log
#SBATCH --output=sbatch_output.log                  ## Output log
#SBATCH --mail-user<=email>@northumbria.ac.uk    ## E-mail for notifications
#SBATCH --mail-type=ALL                              ## Types of notification for e-mails
#SBATCH --partition=48hour                        ## Job que to submit <24hour, 48hour, 120hour>
# ###############################################################################

nodes=$SLURM_JOB_NUM_NODES          ## Number of nodes
cores=$SLURM_CPUS_ON_NODE          ## Number of cores

# Load Modules and setup
# Commented out due to these being included in .bashrc file for load at login.
```

```
# module purge  ## Remove all modules
module add openmpi/intel-opa/gcc-hfi/64/1.10.4  ## Load dedicated MPI module
module add gcc/5.2.0 ## Load GCC module
# module add slurm/15.08.6 ## Load the SLURM module
module load ANSYS/2021R1  ## Load the ANSYS module

# Information
echo "<----------Start---------->"
date

# Execute Solver (replace run_case.jou with the journal file name)
# /usr/bin/time -v >> Information such as time, cpu usage, memory usage, etc.
# fluent >> execute fluent
# 3d >> 3d case (2d for 2d case)
# -gu >> Run with graphics minimised
# -slurm >> Workload manager
# -t<x> >> number of processors to be used
# -mpi=<mpi_type> >> MPI implementation to use
# -i <journal_file_name.jou> >> Read and execute as per journal file
# -cflush >> Free the file cache buffer

echo "<--Executing Fluent-->"
/usr/bin/time -v fluent 3d -gu -slurm -t$((nodes*cores)) -mpi=openmpi -i run_case_les.jou -cflush> fluent.log

echo "<--Fluent Finished-->"
date
echo  ""
# Folder Size
echo "<--Directory Space-->"
pwd
du -sh
# End Time Stamp
echo "<----------End---------->"
date
exit
```

B.2 Scripts with Additional Functions

Please refer to https://bit.ly/2SnMcoa for updated scripts with custom functions such as encoding animations, file organisation, residual plotting and more.

Appendix C Job Submission Checklist

The following checklist in Table C.1 can be used as a generic check to avoid common errors.

Table C.1: Job submission checklist.

No.	Task	Y/N
1.	Ensure the outputs from simulation are saved onto “relative” paths as opposed to “absolute” paths when saving the case file.	
2.	Case and data files named properly (same except for the format)[1].	
3.	The journal file (run_case.jou), job submission script (fluent.sh), case file, data file and other required files are saved in the same folder or referenced file path.	
4.	Filenames and paths inside the scripts are up-to-date (such as case name, journal file name, etc.)[1].	
5.	The number of iterations and times steps has been updated in the journal file.	
6.	Job name, maximum time and the cluster queue has been updated in the job submission script.	
7.	Supported files are present (such as FFMPEG, Python, Numpy, Matplotlib, etc.)[2].	

[1] Linux is case sensitive, and it is advisable to avoid spaces in filenames and paths, instead, use underscores.
[2] These are required only if you are performing additional functions such as encoding animations, plotting from log files, etc..