3. WORKFLOW

Users are recommended to read System Coupling User Manual, but they MUST read the Command Manual before performing setting changes within the run.py script.

Overview

3.1 Open Workbench on local system.

3.2 Setup Mechanical in Workbench.

3.3 Setup Fluent in Workbench.

3.4 Add Systems Coupling Module and connect Mechanical and Fluent setups.

3.5 Add Data transfer and analysis settings.

3.6 Export System Coupling Setup.

3.7 Add Job file in directory and edit based on resources needed and file directories.

3.8 Add Script file and edit settings, and names based on exported Settings.h5 file.

3.9 Upload exported files (with Job and Script files) to scp.chpc.ac.za server via the WinSCP.

3.10 Open lengau.chpc.ac.za server and submit job.

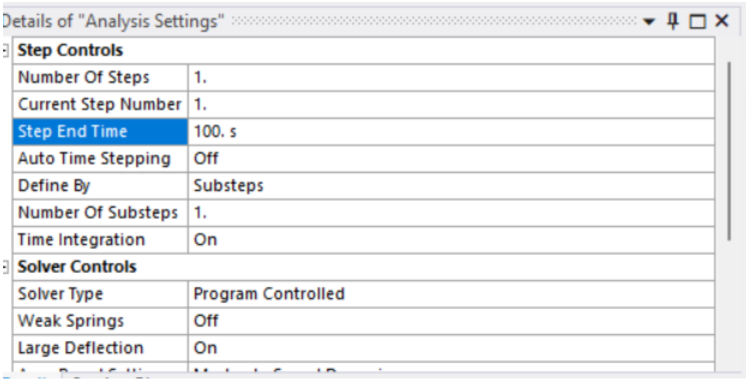
3.1 Open WorkBench on local system

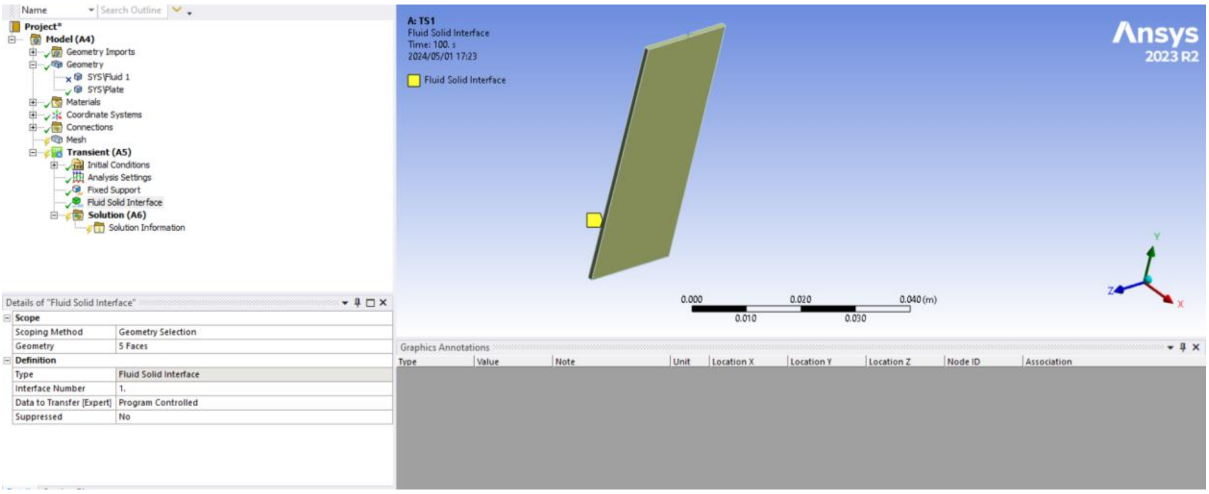
Open WorkBench on local system and add mechanical and fluent modules. Share the geometries with each other. Fluent and Mechanical modules will be setup separately before being added to the Systems Coupling module.

3.2 Setup Mechanical in Workbench

Supress fluid geometry.

- Must set step end time in Mechanical to be greater than or equal to the one specified in System Coupling (set it to 100s for convenience).
- Auto time stepping set to off, and 1 substep per iteration is used.
- A single **Fluid-solid Interface** is applied to the area in contact with the fluid (e.g., 5 sides of a blade which contact the fluid)
- Setup the rest of the simulation as normal: create mesh and assign BCs.





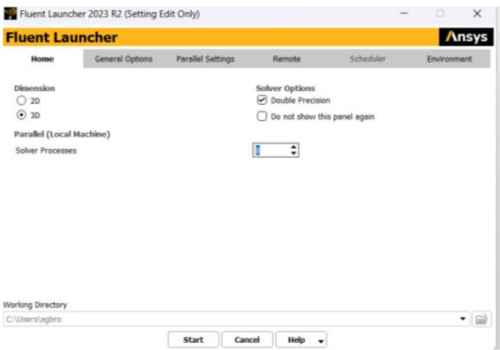
3.3 Setup Fluent in Workbench

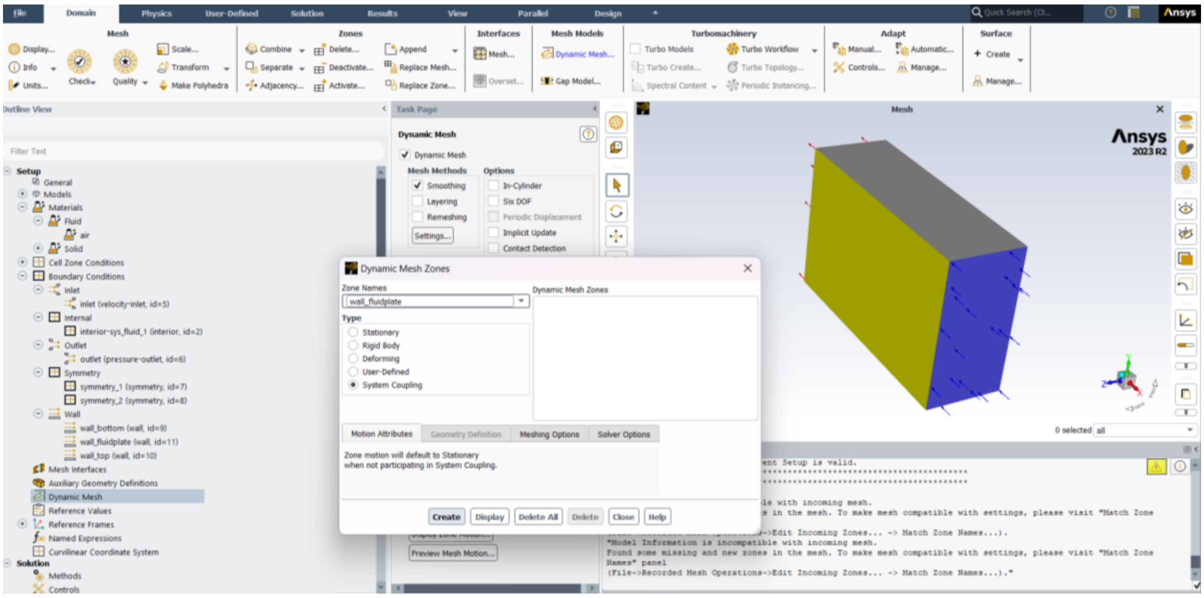
Create mesh.

Open Fluent and ensure that double precision is selected before completing setup.

Supress the solid geometry.

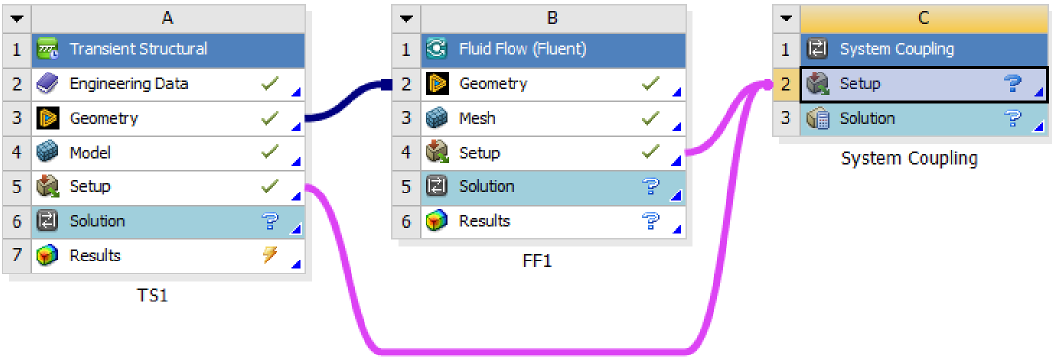
- Set up dynamic mesh. For the fluid walls which are coincident with the solid interface, the dynamic mesh is set to System Coupling.
- Recommend to keep the mesh setting as diffusion. Increase the diffusion parameter if you want to preserve the mesh close to the coupling wall.
- System coupling will override the fluent set time step size and time settings, so can set arbitrary values.
- Setup the rest of the analysis as normal.



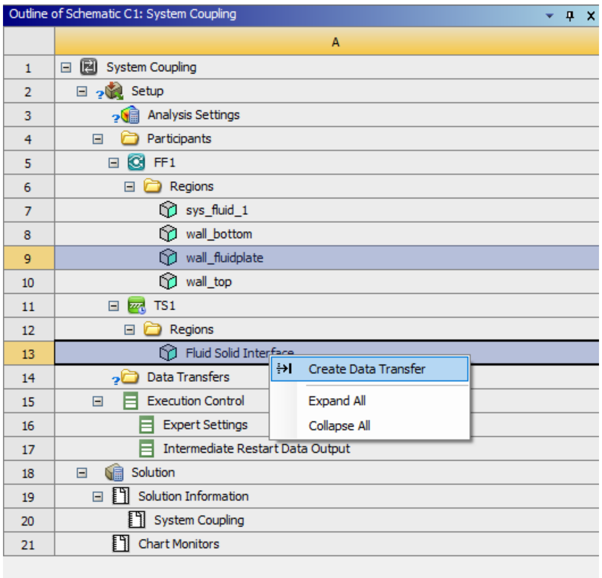


3.4 Add Systems Coupling Module and connect Mechanical and Fluent setups

Once Fluent and Mechanical setups have been completed, connect them to the System Coupling module.



3.5 Add Data transfer and analysis settings



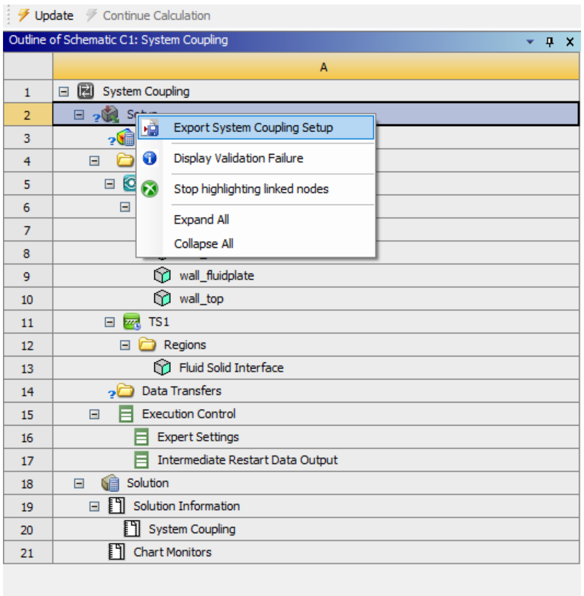
To add data transfer, select the two surfaces which will transfer data, then right click and select Create Data Transfer.

roperties of DataTransfer : Data Transfer

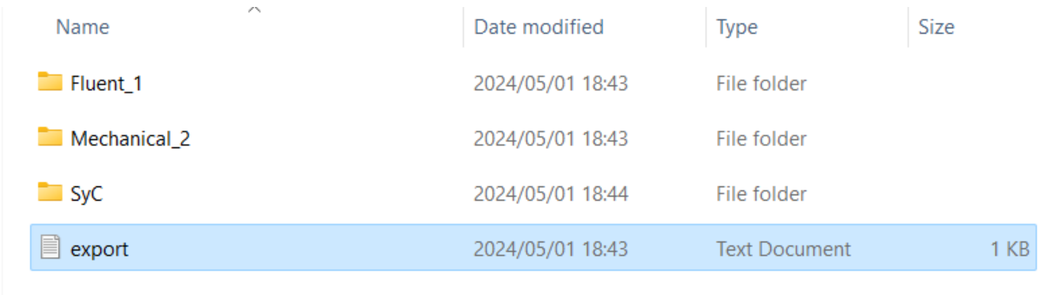
	A	B
1	Property	Value
2	Source	
3	Participant	FF1
4	Region	wall_fluidplate
5	Variable	force
6	Target	
7	Participant	TS1
8	Region	Fluid Solid Interface
9	Variable	Force
10	Data Transfer Control	
11	Transfer At	Start Of Iteration
12	Under Relaxation Factor	1
13	RMS Convergence Target	0.01
14	Ramping	None

	A	B
1	Property	Value
2	Source	
3	Participant	TS1
4	Region	Fluid Solid Interface
5	Variable	Incremental Displacement
6	Target	
7	Participant	FF1
8	Region	wall_fluidplate
9	Variable	displacement
10	Data Transfer Control	
11	Transfer At	Start Of Iteration
12	Under Relaxation Factor	1
13	RMS Convergence Target	0.01
14	Ramping	None

3.6 Export System Coupling Setup



Once data transfers and analysis settings are set, you right click the setup and export the System Coupling setup.



3.7 Add Job file in directory and edit based on resources needed and file directories

#!/bin/bash

#PBS -l select=2:ncpus=24:mpiprocs=24

#PBS -l cfd_base=1

#PBS -l anshpc=44

#PBS -q normal

#PBS -P MECH1663

#PBS -l walltime=0:20:00

#PBS -o /mnt/lustre/users/abrocco/SystemCouplingTest/PBS_Out.out

#PBS -e /mnt/lustre/users/abrocco/SystemCouplingTest/PBS_Err.err

#PBS -m abe

#PBS -M broccorose53@gmail.com

```
##### Running commands

export LM_LICENSE_FILE=1055@login1

export ANSYSLMD_LICENSE_FILE=1055@login1

export FLUENT_ARCH=lnamd64

### You may need the Intel compiler to be available

module load chpc/parallel_studio_xe/18.0.2/2018.2.046

### Mesa is needed to provide libGLU.so

module load chpc/compmech/mesa/18.1.9

export PBS_JOBDIR=/mnt/lustre/users/abrocco/SystemCouplingTest

cd $PBS_JOBDIR

nproc=`cat $PBS_NODEFILE | wc -l`

AWP_ROOT232=/home/apps/chpc/compmech/ansys_inc/v232

cat $PBS_NODEFILE > hosts.txt

$AWP_ROOT232/SystemCoupling/bin/systemcoupling --connectiontimeout=200 --logLevel=2 --ic=infiniband --cnf=hosts.txt --sycnprocs=6 --nprocs=$nproc --runScript=run.py
```

NOTES:

- Must specify the AWP_ROOT232 variable as the absolute path to the required version of ANSYS.
- --connectiontimeout specifies the number of seconds the program is allowed to wait while the participants connect.
- --logLevel specifies the verbosity of the diagnostic files created
- --syncnprocs is the number of processors that the System Coupling module (excluding participants) will use.
- --runScript runs the python script ‘run.py’ that is in the file directory

3.8 Add Script file and edit settings, and names based on exported Settings.h5 file

```
##############################################################

# Example Script for System Coupling v232

# 5 May 2024

# By Alex Brocco

##############################################################


############################ Notes ##########################
```

```
# This script is used to run fully setup Systems Coupling WorkBench module.
# The user only needs to define the data transfer and solution controls, then
# export the full Systems Coupling setup (must have green check mark).
# The export automatically creates scp files, settings.h5 file and relevant subdirectories.
# The automatically generated settings that do not require changing are NOT changed by this
# script (e.g., interfaces, regions, variables etc.).

# Refer to the settings.h5 file for applicable names of participants and interfaces
# (not the display names).
# For this example, the fluent participant is referred to as 'Fluent_2'  with input
# file 'fluent.cas.h5'.
# Mechanical participant is referred to as 'Mechanical_1' with input file 'Mechanical.dat".
# There is only one interface called 'interface-1'.
# 'Data Transfer' is force from fluent to mechanical.
# 'Data Transfer 2' is displacement from mechanical to fluent.

##################### 0. Names ##############################
# (Definitions not display names from Settings.h5 file)
FluentParticipant_Name = 'Solution 1'
MechnaicalParticipant_Name = 'Solution'
Fluent_CouplingWall_Name1 = 'wall_fluid_plate'
Mechanical_CouplingWall_Name1 = 'FSIN_1'
Fluent_Working_Directory = 'Fluent_2'
Mechanical_Working_Directory = 'Mechanical_1'
Fluent_CaseFile = 'fluent.cas.h5'
Fluent_DataFile = 'fluent.dat.h5'
#Fluent_JournalFile =
Mechanical_File = 'Mechanical.dat'

Open() # reads Settings.h5 file and restart files in SyC directory, optional 'CouplingStep='
# input allows you to start from an intermediate restart point. With no commands it will run
```

```
# as normal or use the latest restart point if there is one.
dr = DatamodelRoot()

############################ 1. Library ########################

##################### 2. Coupling Participant #################

# Fluent General Settings
Fluent_Par = dr.CouplingParticipant[FluentParticipant_Name]
Fluent_Par.Dimension = '3D'
Fluent_Par.ParticipantType = 'FLUENT'
Fluent_Par.ParticipantAnalysisType = 'Transient'
Fluent_Par.RestartsSupported = True
Fluent_Par.UseNewAPIs = True

# Fluent Participant Excecution Control
FluentEC = Fluent_Par.ExecutionControl
FluentEC.Option = 'ProgramControlled' # ProgramControlled, UserDefined, ExternallyManaged
FluentEC.AutoDistributionSettings = True # If False, add additional resource distribution settings
# Setting Fluent input to Case and Data allows initialization from SS solution
FluentEC.FluentInput.Option = 'InitialCaseAndData' #InitialCaseFile, InitialCaseAndData, JournalFile
FluentEC.FluentInput.CaseFile = Fluent_CaseFile
FluentEC.FluentInput.DataFile = Fluent_DataFile
#FluentEC.FluentInput.JournalFile =
FluentEC.ParallelFraction = 1.0 #Using all available parallel resources
FluentEC.AdditionalArguments = '3ddp' #must run 3D double precision
FluentEC.WorkingDirectory = Fluent_Working_Directory

# Mechanical General Settings
Mechanical_Par = dr.CouplingParticipant[MechnaicalParticipant_Name]
Mechanical_Par.Dimension = '3D'
```

Mechanical_Par.ParticipantType = 'MAPDL'

Mechanical_Par.ParticipantAnalysisType = 'Transient'

Mechanical_Par.RestartsSupported = True

Mechanical Participant Execution Control

MechanicalEC = Mechanical_Par.ExecutionControl

MechanicalEC.Option = 'ProgramControlled' # ProgramControlled, UserDefined, ExternallyManaged

MechanicalEC.AutoDistributionSettings = True # If False, add additional resource distribution settings

MechanicalEC.InitialInput = Mechanical_File #input file name (located in subdirectory)

MechanicalEC.ParallelFraction = 1.0/5.0 # Using one fifth of total available parallel resources

MechanicalEC.WorkingDirectory = Mechanical_Working_Directory

#MechanicalEC.AdditionalArguments = None

#MechanicalEC.AdditionalRestartInputFile restart file for mechanical to allow additional commands

###################### 3. Analysis Control ###################

Not changing the autogenerated settings that do not need changing

AC = dr.AnalysisControl

AC.AllowSimultaneousUpdate = False

AC.PartitioningAlgorithm = 'SharedAllocateMachines' # Parellel resource allocation algorithm, only change if there is not enough RAM, consider 'DistributedAllocateCores'

AC.GlobalStabilization.Option = 'None'

Add Additional AC.GlobalStabilization if Quasi-Newton selected

##################### 4. Coupling Interface #################

Mapping Controls

CoupInt1 = dr.CouplingInterface['interface-1']

CoupInt1.MappingControl.FaceAlignment = 'ProgramControlled' #Might need to change if using porous zones or shell elements

CoupInt1.MappingControl.StopIfPoorIntersection = True

CoupInt1.MappingControl.PoorIntersectionThreshold = 0.5 # 0 to 1, as a fraction

CoupInt1.MappingControl.AbsoluteGapTolerance = 0.1 # value in [m]

CoupInt1.MappingControl.RelativeGapTolerance = 1 # fraction relative to face size

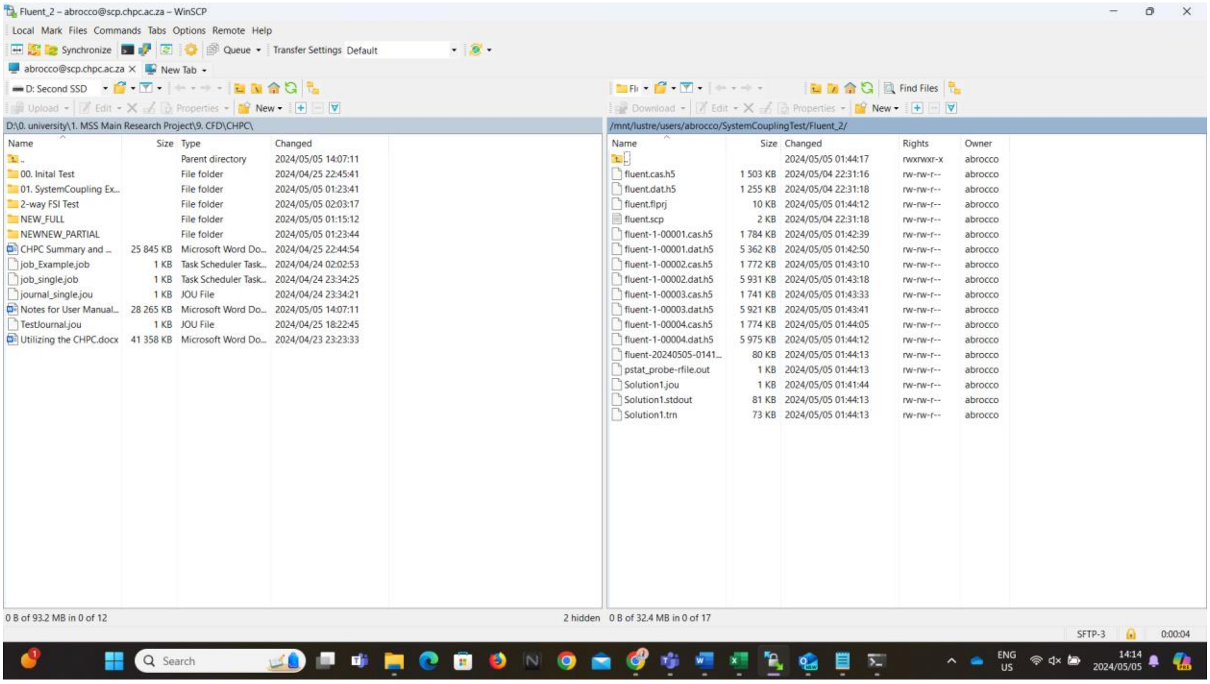
```
# Data Transfer Settings
CI1_DT1 = CoupInt1.DataTransfer['Data Transfer'] #force from fluent to mech
CI1_DT2 = CoupInt1.DataTransfer['Data Transfer 2'] #disp from mech to fluent
CI1_DT1.RampingOption = 'Linear' #None or Linear
CI1_DT2.RampingOption = 'Linear'
CI1_DT1.RelaxationFactor = 0.95 # Between 0 and 1, 1 is default
CI1_DT2.RelaxationFactor = 0.95
CI1_DT1.ConvergenceTarget = 1e-2 #Defualt is 1e-2
CI1_DT2.ConvergenceTarget = 1e-2
CI1_DT1.UnmappedValueOption = 'Extrapolation' #Nearest Value or Extrapolation
CI1_DT2.UnmappedValueOption = 'Extrapolation'
# Add Quasi-Newton settings if AC.GlobalStabilization.option set to Quasi-Newton


###################### 5. Solution Control #########################
SC = dr.SolutionControl
SC.DurationOption = 'EndTime' #NumberOfSteps or EndTime
SC.EndTime = 2.0 # [s]
#SC.NumberOfSteps =
SC.TimeStepSize = 0.025 # [s]
SC.MinimumIterations = 3 #per time step
SC.MaximumIterations = 6 #per time step


##################### 6. Output Control ###########################
OC = dr.OutputControl
OC.Option = 'StepInterval' #LastStep, EveryStep, StepInterval
OC.GenerateCSVChartOutput = True
OC.OutputFrequency = 1 #Remove if Option set to something else
OC.WriteInitialSnapshot = False
#OC.IncludeInstances = 'ProgramControlled' #ProgramControlled, ReferenceInstance, All
```

```
OC.Results.Option = 'StepInterval'

OC.Results.OutputFrequency = 1 #Remove if option set to something else


######################### Additional Commands #####################

GetErrors()

#get the execution command used for the participants

GetExecutionCommand(ParticipantName = MechnaicalParticipant_Name)

GetExecutionCommand(ParticipantName = FluentParticipant_Name)

GetMachines()

PrintSetup()

Solve()

# Shutdown() to interupt solution, use if you can implement an if statement based on available walltime
```

NOTES:

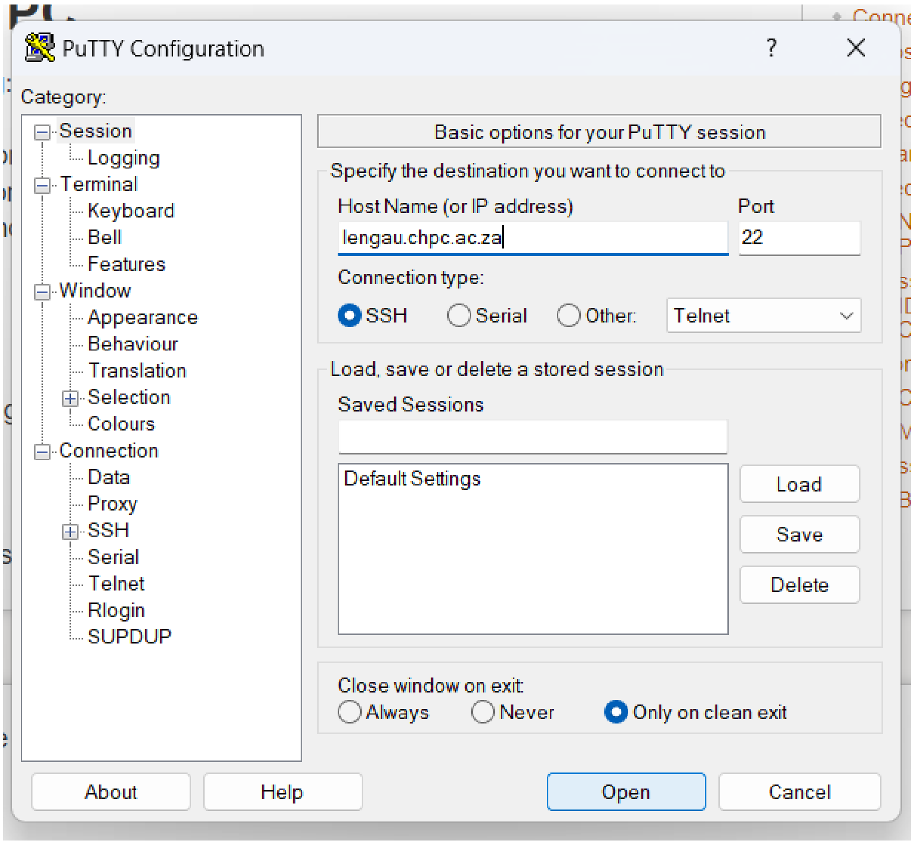
- Used 2 nodes (48 cores) for the example. SharedAllocateMachines algorithm used. ParallelFraction set to 1/5 for Mechanical so 10 cores were used. Set ParallelFraction set to 1 for Fluent so 48 cores were used. Set --sycnprocs=6 so 6 cores were used for System Coupling. This means some cores have multiple processes running from different applications.
- If no settings need to be changed, then can simply have a script with only Open() and Solve() commands
- It is possible to use a partial setup and manually add the participants, variables and interfaces. Tried it and it can work using the Add<>() commands (refer to command manual), but it is unnecessary and a waste of time since WorkBench can do it in a fraction of the time automatically, especially since there is no benefit to doing it manually.

3.9 Upload exported files (with Job and Script files) to scp.chpc.ac.za server via the WinSCP

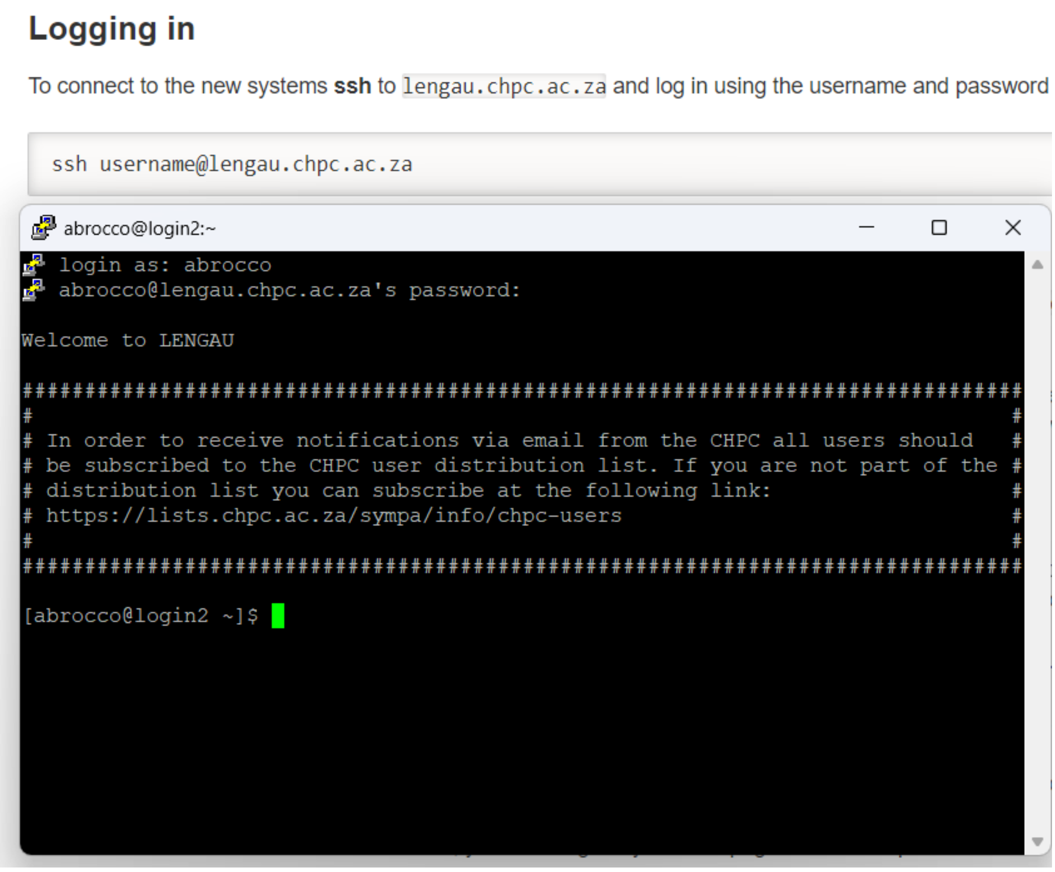


Left hand side is your file directory and right-hand side is the home and lustre file directories on the CHPC.

3.10 Open lengau.chpc.ac.za server and submit job



Open PuTTY for submitting jobs, connect to **lengau.chpc.ac.za server.**



Then this command window opens when you select open.

Login as abrocco do not have to say ssh and the @ part because you did that when opening PuTTY

Enter password

To copy in Putty, cntrl c, then right click.

Used the command ll /apps/chpc/compmech/CFD/ansys_inc/v232/ to check all the ANSYS products that they have.

Once in the correct directory which has all the files (using cd command), use qsub {job file name} to submit the job

Can track with qstat -awu abrocco or qstat -f {job number}

3.11 Inspect the output files

During the solve, use the update icon in top ribbon in WinSCP to get live file updates. Inspect logs for convergence and inspect the mapping summary.

```
+-----------------------------------------------------------------------------+
|                              MAPPING SUMMARY                                |
+-----------------------------------------------------------------------------+
|                                     |      Source           Target         |
+-----------------------------------------------------------------------------+
| interface-1                         |                                       |
|   Data Transfer                     |                                       |
|     Mapped Area [%]                 |       100              100            |
|     Mapped Elements [%]             |       100              100            |
|     Mapped Nodes [%]                |       100              100            |
|   Data Transfer 2                   |                                       |
|     Mapped Area [%]                 |       100              100            |
|     Mapped Elements [%]             |       100              100            |
|     Mapped Nodes [%]                |       100              100            |
+-----------------------------------------------------------------------------+
```