Universitat Politecnica de Catalunya

Barcelona,Spain

Guidelines to use Abaqus, Ansys and OpenFoam for LaCaN Users

Author:
Anil Bettadahalli
MSc.Compuatational
Mechanics

Supervisor:
Professor Pedro Diez
Director of LaCaN
UPC-BarcelonaTech

July 6, 2016

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

1 Introduction

In this manual it has been clearly explained the usage of different software's like Abaqus, Ansys and OpenFoam in Linux Computers of LaCaN from different server's/cluster's with examples and also the limitations of the software's. Different examples are shown for respective software's from which a new user can clearly understand how to get access to different software's and how to call scripts(like Python and Matlab) in software's without using Graphical User Interface (GUI). The scripts which are used in different examples are attached in *Appendices* section.

Versions of different software's installed:

- Abaqus V6.11-3
- Ansys V15
- OpenFoam V2.4.0
- ParaView V4.1.0
- Python V2.6.2
- Matlab V2015

Different Clusters where software's installed:

- ALIEN
- Lordvader
- Clonetroop

2 How to use Abaqus ?

Abaqus is a calculation program for finite elements analysis. It is oriented to solve problems such as solid mechanics, linear and non linear problems in static and dynamic ranges.

Abaqus is installed in ***'ALIEN Cluster'*** in UPC and its version is 'Abaqus V6.11-3' and right now there are only '4' licenses(Academic Research) of Abaqus available in the *'ALIEN'*. One has to get the access('username' and 'password') to *'ALIEN'* first to use Abaqus and get the works done. To get the access mail to **'serveistic.camins@upc.edu'** stating your objective.

ALIEN: *ALIEN* is a high performance cluster formed by a set of Dell computers. It is a multitasking and multiuser environment with a Red Hat operative system using the set of tools known as cluster Oscar. This cluster is oriented for those users that execute programs with high performance requirements of memory as well as processor time.

One access node and six nodes for calculation form this cluster. The six calculation nodes are the ones that will execute the jobs the users send. Four of them have Intel processors meanwhile the other two have AMD processors. Adding all of them this cluster provides 64 Intel cores and 16 AMD. It is recommended not to send more than two jobs per user using Abaqus/Standard and Abaqus/Explicit, and just one job for Abaqus/CAE.

For more details refer to : http://www.camins.upc.edu/pdf/pdf-serveis/recursos-per-a-la-recerca/manual-utilitzacio-alien-english

>> Following are the steps need to follow to use Abaqus software from LaCaN Computer's:

1. **Get the 'username' and 'password' for *Lordvader* server (lordvader.upc.es) from LaCaN sytems administrator.**

   ⇓

2. **Login with 'username' and 'password' in Lordvader**

   ⇓

3. **Open the terminal 'CTRL+ALT+T'**

   ⇓

4. **Type 'ssh -X <username>@alien.upc.edu' ('username' of Alien server)**

   ⇓

5. **Password: <Type password> ('password' of Alien server)**

   ⇓

6. **Create your own folder using command 'mkdir <folder name>'**

   ⇓

7. **Create your input files [.inp] format (Normally written in Python scripts for Abaqus) and save in folder.**

   ⇓

8. **To Launch jobs to the Queues using Abaqus:**

   - Use command - **'abaqus job=filename [.inp] queue=queue_name'**
   - OR If you have python scripts just to create Model Database/Output Database to call in Abaqus, use command: **'abaqus cae script = <filename>.py'**

   ⇓

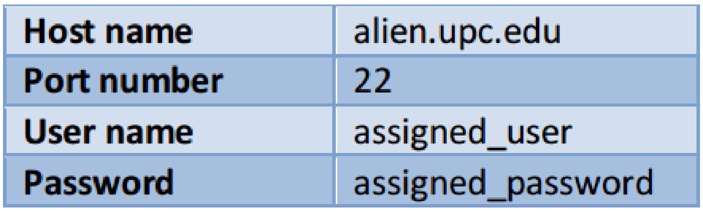
9. **To Execute the Post-process/to open GUI of Abaqus:**

   - Use command - **'abaqus cae'**

>> Some of the other useful things need to know which will be helpful are listed below:

<u>To transfer files from 'Lordvader' to 'Alien'</u>

- Use ***'FILEZILLA'*** software which is already installed in LaCaN Computers.



Host name	alien.upc.edu
Port number	22
User name	assigned_user
Password	assigned_password

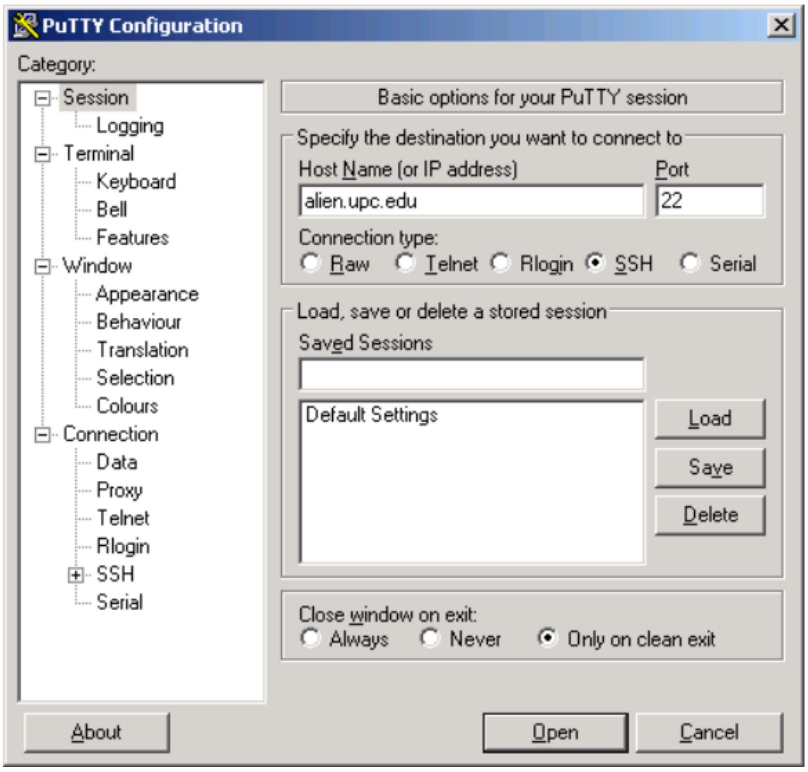
To have remote access from Windows/Linux OS of your personal computer:

- Use ***'MOBAXTERM'*** software which allows to connect remotely to *Lordvader* through **'secured shell(ssh protocol)'** when you want to work from home.

  You can download here : http://mobaxterm.mobatek.net/download.html

- Or if you have just Windows OS, you can use ***'puTTY'*** software which also allows you to connect remotely to server's through **'ssh protocol'**.

  You can download here : http://www.putty.org/



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): alien.upc.edu

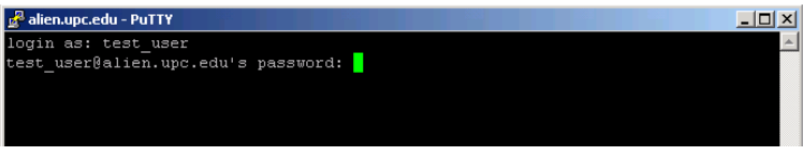
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit



alien.upc.edu - PuTTY

```
login as: test_user
test_user@alien.upc.edu's password:
```

For more detailed information take a look at this document : http://www.camins.upc.edu/pdf/pdf-serveis/recursos-per-a-la-recerca/Manual%20de%20PuTTY.pdf

Some of the basic commands in a Unix environment you will find useful :

> pwd - shows the current working directory.
> mkdir - creates a new directory.
> cd - changes from one directory to the one specified.
> ls - lists the files.
> mv - move files or subdirectories from one directory to any other specified directory.
> cp - copies files into the specified directory.
> rmdir - deletes the specified directory which must be empty.

For more details : http://www.mediacollege.com/linux/command/linux-command.html

EXAMPLE 1 : Parametric Study of Cantilever Beam

To explain calling *Python scripts* in *Abaqus*, I am considering example of parametric study of simple 'Cantilever Beam' of square cross section with length and width = 30m, depth=250 m which is subjected to uniform load of 50000N as shown in *Figure* 1 below ('A' refers surface upon which load is subjected uniformly and surface 'B' refers to fixed surface). Here I am conducting parametric study by varying Material parameters like Young's Modulus(E) and Poisson's ratio (ν).

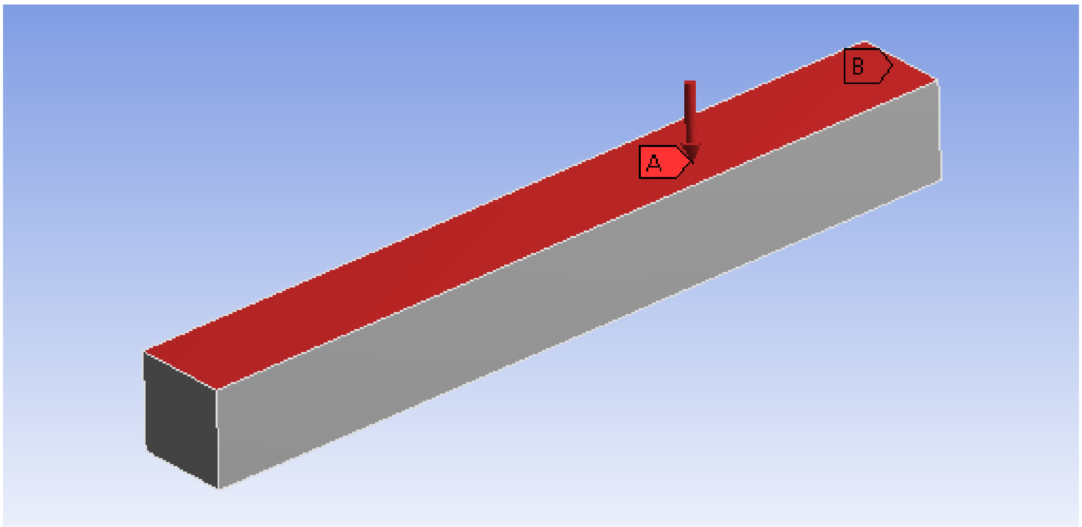


Figure 1: Cantilever beam subjected to uniform load of 50000N

>> To perform a Parametric study one must follow these 4 major steps :

1. **Create the model in Abaqus CAE Environment**

   ⇓

2. **Save the input file and modify it.**

   ⇓

3. **Construct the Python Script**

   ⇓

4. **Run the Python Script**

>>To explain calling *Python script* in *Abaqus* to perform Parametric study in LacaN Computer's follow below steps :

1. **Get the 'username' and 'password' for Lordvader server (lordvader.upc.es) from LaCaN sytems administrator.**

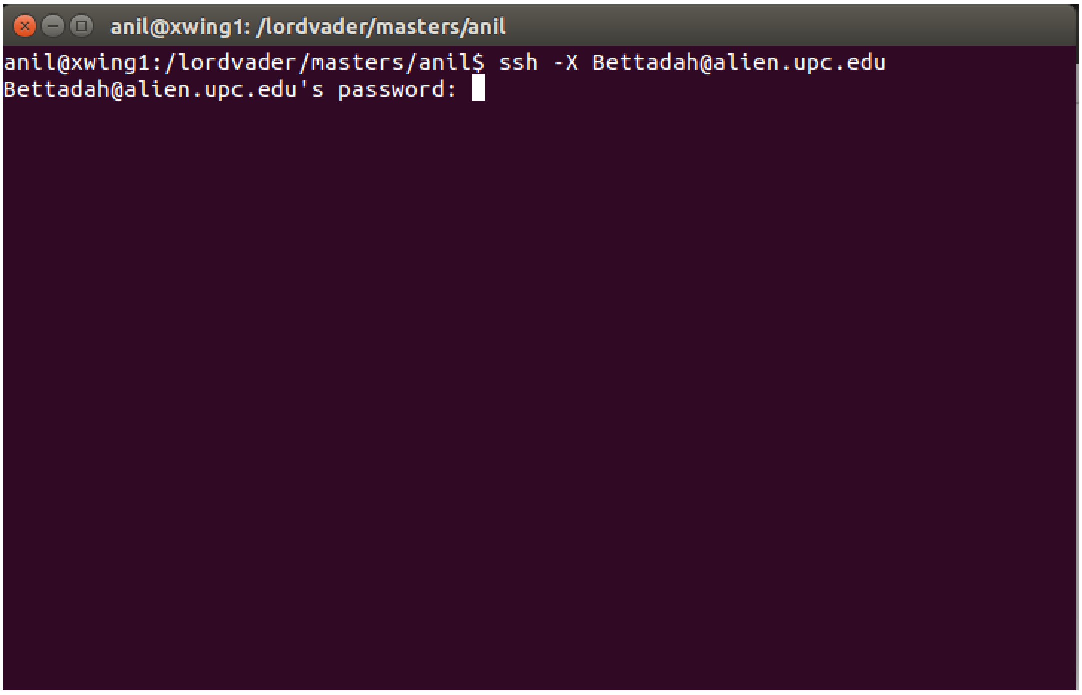
   ⇓

2. **Login with 'username' and 'password' in Lordvader**

   ⇓

3. **Open the terminal 'CTRL+ALT+T'**

   ⇓



```
anil@xwing1:/lordvader/masters/anil$ ssh -X Bettadah@alien.upc.edu
Bettadah@alien.upc.edu's password: 
```

4. **Type 'ssh -X <username>@alien.upc.edu' ('username' of Alien server)**

   ⇓

5. **Password: <Type password> ('password' of Alien server)**

   ⇓

6. **Create your own folder using command 'mkdir <folder name>'**

   ⇓

7. **Create your input files [.inp] format (Normally written in Python scripts for Abaqus) and save in folder.**

   - Create the model in Abaqus CAE environment.
   - Generate the input file and save it in folder.
   - Modify input file and save it in folder (attached in *Appendix 1* named **'PSE.inp'**)
   - Create the Python Script and save it in same folder(named as **'PSE.psf'**, attached in *Appendix 2*). Below lines attached is simple Python Script, where we can observe that 'E' and 'ν' varied by discrete and continuous values to understand the vertical displacements and stress distribution for different values of material properties for the given load. And the final result is saved in file name given **'cbeam.PSE.psr'**

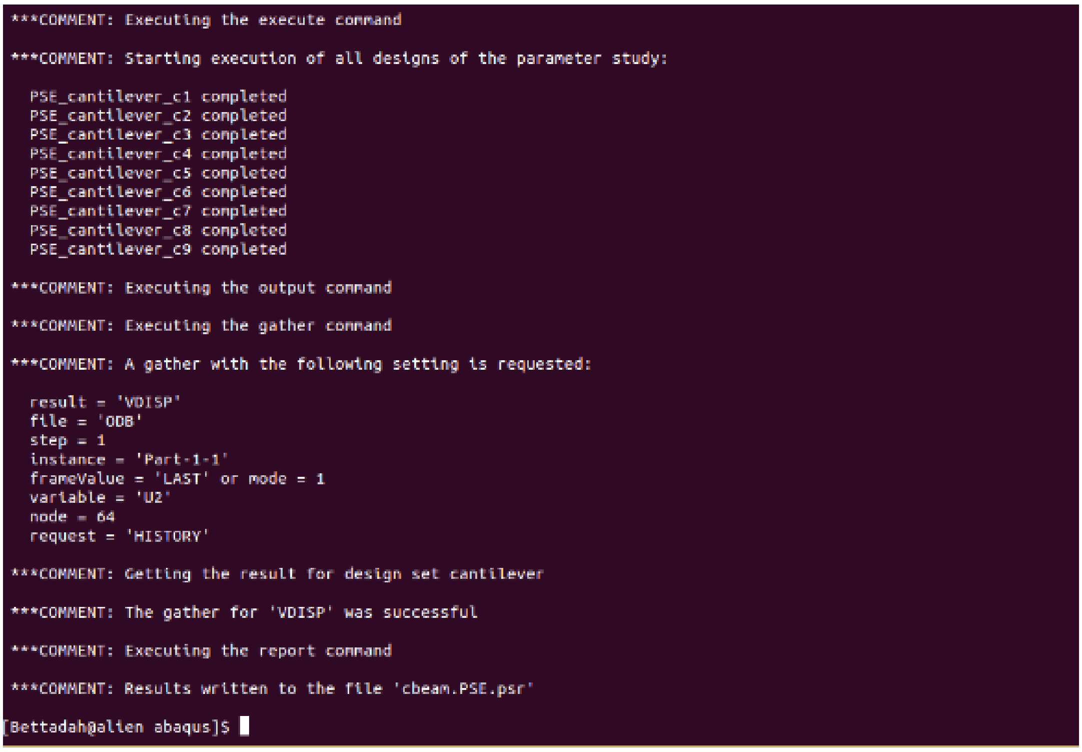
```
#create study
cbeam=ParStudy(par=('E','nu'))
#defining the parameters
cbeam.define(DISCRETE,par='E',domain=(100e9,200e9,300e9))
cbeam.define(CONTINUOUS,par='nu',domain=(0.15,0.35))
#sampling
cbeam.sample(INTERVAL,par='E',interval=1)
cbeam.sample(NUMBER,par='nu',number=3)
#combine the samples
cbeam.combine(MESH,name='cantilever')
```

```
#generate the analysis job data
cbeam.generate(template='PSE')
#execute the jobs
cbeam.execute(ALL)
#read the results from ODB and Report
cbeam.output(step=1,file=ODB)
cbeam.gather(results='VDISP',variable='U2',node=64,...
    request=HISTORY,instance="Part-1-1")
cbeam.report(FILE,results=('VDISP'),par=('E','nu'),file='cbeam.PSE.psr')
```

⇓

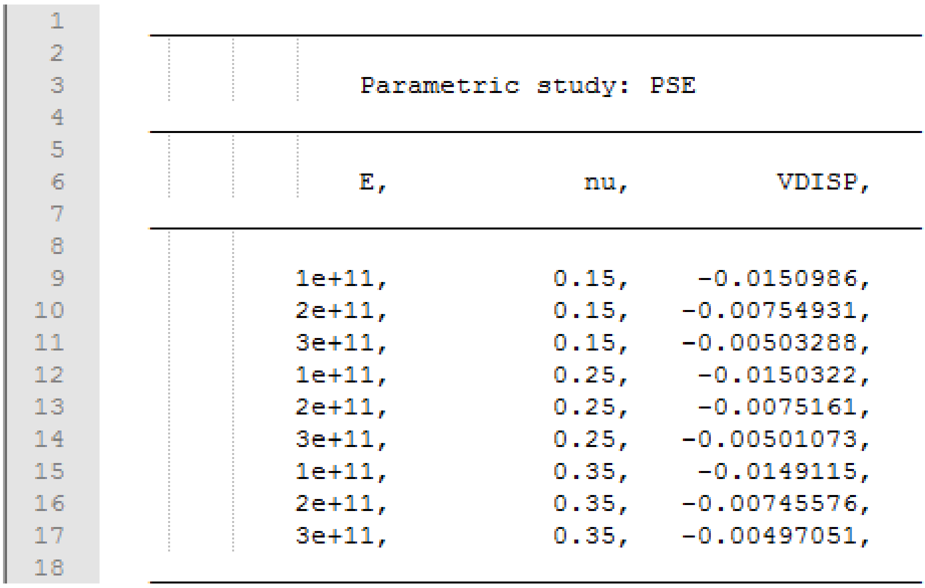
8. **Execute/Run the Python Script:**

   - Using Command - **'abaqus script = PSE.psf'** , we obtain following results as shown below.



```
***COMMENT: Executing the execute command

***COMMENT: Starting execution of all designs of the parameter study:

  PSE_cantilever_c1 completed
  PSE_cantilever_c2 completed
  PSE_cantilever_c3 completed
  PSE_cantilever_c4 completed
  PSE_cantilever_c5 completed
  PSE_cantilever_c6 completed
  PSE_cantilever_c7 completed
  PSE_cantilever_c8 completed
  PSE_cantilever_c9 completed

***COMMENT: Executing the output command

***COMMENT: Executing the gather command

***COMMENT: A gather with the following setting is requested:

  result = 'VDISP'
  file = 'ODB'
  step = 1
  instance = 'Part-1-1'
  frameValue = 'LAST' or mode = 1
  variable = 'U2'
  node = 64
  request = 'HISTORY'

***COMMENT: Getting the result for design set cantilever

***COMMENT: The gather for 'VDISP' was successful

***COMMENT: Executing the report command

***COMMENT: Results written to the file 'cbeam.PSE.psr'

[Bettadah@alien abaqus]$ 
```

We can observe after running the scripts, 9 different 'Output Database(.odb)' files have been created which shows the results as per our variation of material properties and saved automatically in the same folder (Open 'ODB' files in GUI to view visualization). The result file named **'cbeam.PSE.psr'** shows the vertical displacements for different values of material properties ('E', 'ν') since in our script file we have asked to obtain the result files of vertical displacement selecting certain node number for generating history. The results are shown in below figures.

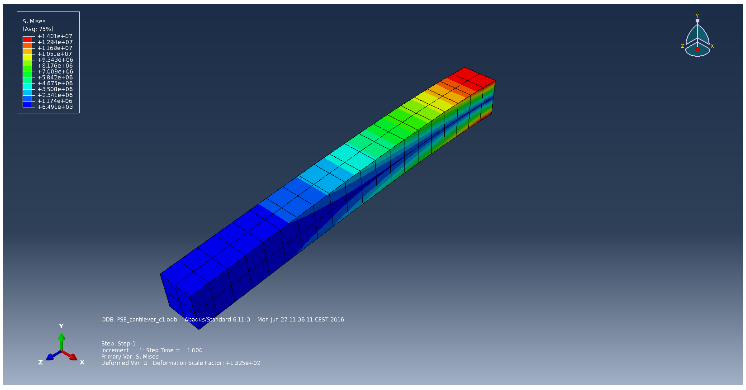


```
---------------------------------------------------------------
            Parametric study:  PSE
---------------------------------------------------------------
            E,              nu,            VDISP,
---------------------------------------------------------------
          1e+11,          0.15,      -0.0150986,
          2e+11,          0.15,     -0.00754931,
          3e+11,          0.15,     -0.00503288,
          1e+11,          0.25,      -0.0150322,
          2e+11,          0.25,      -0.0075161,
          3e+11,          0.25,     -0.00501073,
          1e+11,          0.35,      -0.0149115,
          2e+11,          0.35,     -0.00745576,
          3e+11,          0.35,     -0.00497051,
---------------------------------------------------------------
```

⇓

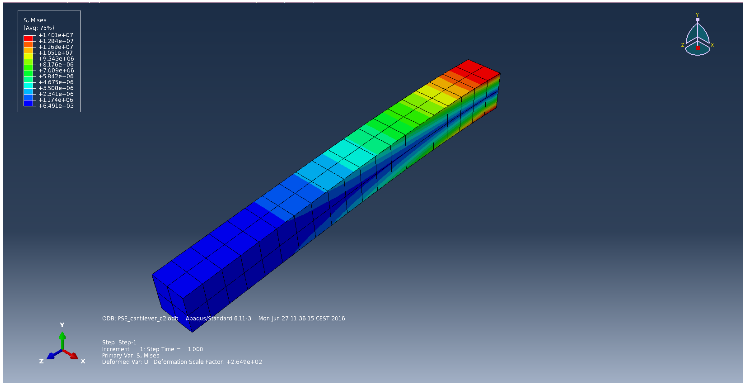
9. **To Execute the Post-process/to open GUI of Abaqus:**

- Use command - **’abaqus cae’**
- To view the output results, open different ’ODB’ files which is saved in folder. Here I am just showing the Von-Mises stress distribution for 4 different material properties of cantiliver beam for reference. (files named - *PSE_cantilever_c1.odb*, *PSE_cantilever_c2.odb*, *PSE_cantilever_c3.odb*, *PSE_cantilever_c4.odb*).



```
ODB: PSE_cantilever_c1.odb    Abaqus/Standard 6.11-3    Mon Jun 27 11:36:11 CEST 2016

Step: Step-1
Increment      1: Step Time =    1.000
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +1.325e+02
```

Figure 2: Von-mises stress distribution of cantilever 1



```
ODB: PSE_cantilever_c2.odb    Abaqus/Standard 6.11-3    Mon Jun 27 11:36:15 CEST 2016

Step: Step-1
Increment      1: Step Time =    1.000
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +2.649e+02
```

Figure 3: Von-mises stress distribution of cantilever 2

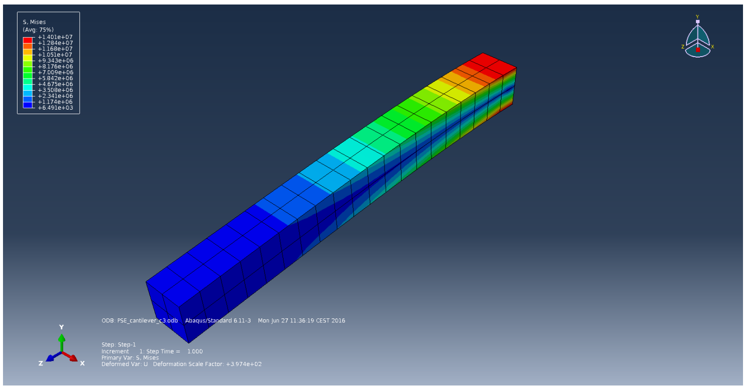


Figure 4: Von-mises stress distribution of cantilever 3

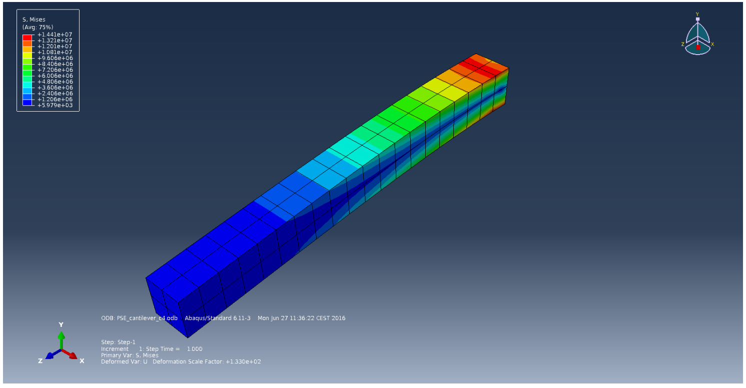


Figure 5: Von-mises stress distribution of cantilever 4

EXAMPLE 2 : 2D Planar shell

One more way to explain calling *Python scripts* in *Abaqus* is creating Model database or Output database of model manually by writing python scripts to call in GUI without having any interaction with the GUI. Here I have written python scripts for creating a simple model of 2d rectangular planar shell of length and width of 10m and 2m, with linear elastic material properties of E=1e9 N/m^2, ν=0.3 which is fixed at one end and subjected to uniform distributed load of 100000 N, which is saved in the folder by name 'EXAMPLE_MDB.py'(attached in *Appendix 3*).

Below lines gives the clear explanation of python scripting step by step for the 2d planar shell to create model, apply material, meshing, boundary conditions, to apply load and to execute job for postprocess without interaction of GUI .

>> Python script for creating model, apply BC's and load :

```
#load modulus
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
```

- Including these lines will import some of the Abaqus modulus used in this script file.

```
### PART ###
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=20.0)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-5.0, -1.0),
point2=(5.0, 1.0))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1', type=
DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
```

- With these lines the sketch and part is created.

```
### MATERIAL & SECTION ###
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((1000000000.0,
0.3), ))
mdb.models['Model-1'].HomogeneousSolidSection(material='Material-1', name=
'Section-1', thickness=None)
mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
offsetField='', offsetType=MIDDLE_SURFACE, region=Region(
faces=mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667,
-0.333333, 0.0), (0.0, 0.0, 1.0)), )), sectionName='Section-1')
```

- With these lines Material, section and section assignment are done.

```
### SET & SURFACE ###
mdb.models['Model-1'].parts['Part-1'].Set(edges=
mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-5.0, -0.5, 0.0), )),
name='Set-1')
mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=
mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-2.5, 1.0, 0.0), )))
```

- Here we create the set and surface. Note findAt was used instead of getSequenceFromMask (which is a numbering system Abaqus used), because we entered the line in the script window in the GUI before creating the model.

```
### MESH ###
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=QUAD, regions=
mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667, -0.333333,
0.0), )), technique=STRUCTURED)
mdb.models['Model-1'].parts['Part-1'].setElementType(elemTypes=(ElemType(
elemCode=CPS8R, elemLibrary=STANDARD), ElemType(elemCode=CPS6M,
elemLibrary=STANDARD)), regions=(
mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667, -0.333333,
0.0), )), ))
```

```
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=0.5)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
```

- *These lines will create the mesh control, element type and mesh.*

```
### ASSEMBLY ###
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-1-1',
part=mdb.models['Model-1'].parts['Part-1'])
mdb.models['Model-1'].rootAssembly.regenerate()
```

- *The assembly process is given above. Note that the word rootAssembly is used and you don't have to give a name of the assembly.*

```
### STEP, BC & LOAD ###
mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1',
previous='Initial')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
'BC-1', region=
mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].sets['Set-1'], u1=
0.0, u2=0.0, ur3=0.0)
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
distributionType=UNIFORM, field='', magnitude=-100000.0, name='Load-1',
region=
mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])
```

- *Above lines will create step, BC and application of load.*

```
### JOB & CALCULATE ###
mdb.Job(contactPrint=OFF, description='', echoPrint=OFF, explicitPrecision=
SINGLE, historyPrint=OFF, memory=90, memoryUnits=PERCENTAGE, model=
'Model-1', modelPrint=OFF, multiprocessingMode=DEFAULT, name='EXAMPLE',
nodalOutputPrecision=SINGLE, numCpus=1, numDomains=1,
parallelizationMethodExplicit=DOMAIN, scratch='', type=ANALYSIS,
userSubroutine='')
mdb.jobs['EXAMPLE'].submit(consistencyChecking=OFF)
```

- *This creates the job and submits the job for analysis.*

>> Now to run this script simply follow these steps :

1. **Login with 'username' and 'password' in Lordvader**

   ⇓

2. **Open the terminal 'CTRL+ALT+T'**

   ⇓

3. **Type 'ssh -X <username>@alien.upc.edu' ('username' of Alien server)**

   ⇓

4. **Password: <Type password> ('password' of Alien server)**

   ⇓

5. **Create your own folder using command 'mkdir <folder name>'**

   ⇓

6. **Save your script file inside the folder created.**

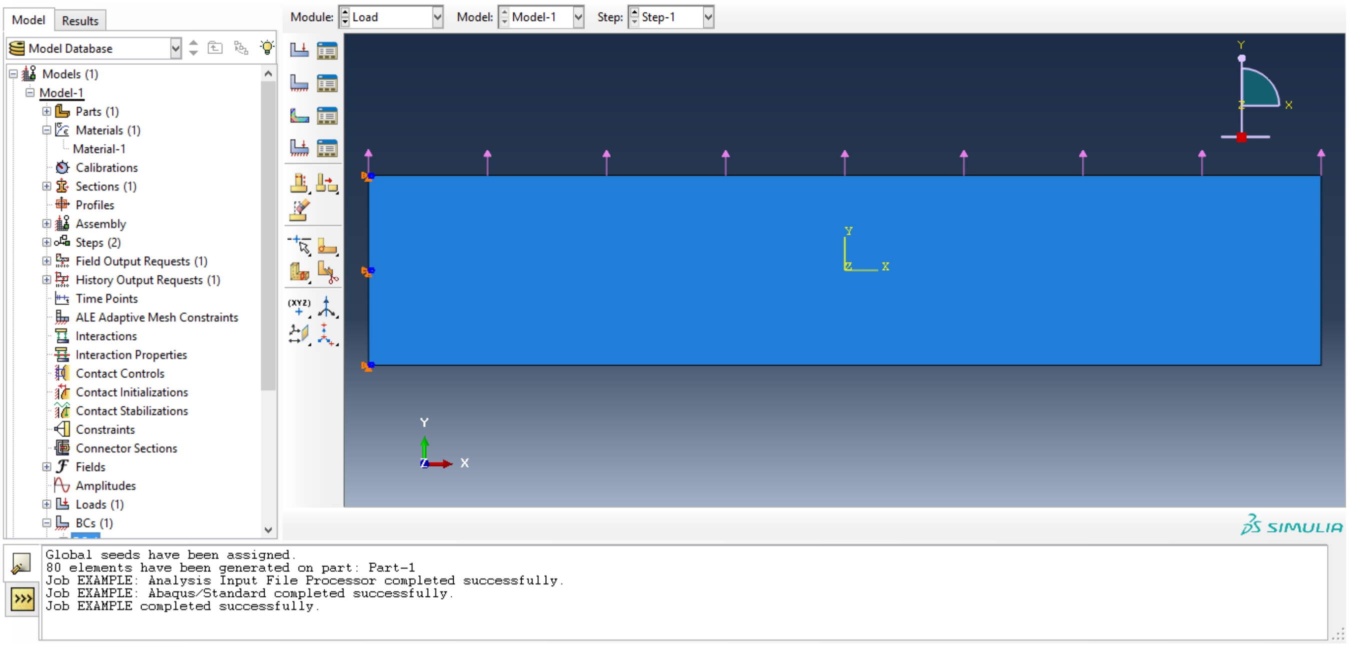
   ⇓

7. **Run the script file**

   - Use Command - **'abaqus cae script=**`EXAMPLE_MDB.py`**'** (after you are inside folder where script is saved)
   - OR else you can also give directly the path name where your script is saved soon after you login to Alien. **Login to Alien ⇒ 'pwd'⇒ 'abaqus cae script=/pathname/'**

   ⇓

8. **After the scripts get executed, Abaqus cae GUI will open automatically calling python scripts and we can see the generated model(as shown in below *Figure 6*).**



```
Global seeds have been assigned.
80 elements have been generated on part: Part-1
Job EXAMPLE: Analysis Input File Processor completed successfully.
Job EXAMPLE: Abaqus/Standard completed successfully.
Job EXAMPLE completed successfully.
```

Figure 6: Model created from python script

>>Python scripts for creating Output database :

The content of this script file is given in the *Appendix 4* named `'EXAMPLE_ODB.py'` . Below is a short ordering of the script file in which output like 'stress in deformed shape' is requested to display.

```
#open modulus, create viewport and open odb
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=268.952117919922,
height=154.15299987793)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
o1 = session.openOdb(name='anil/abaqus/EXAMPLE.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
```

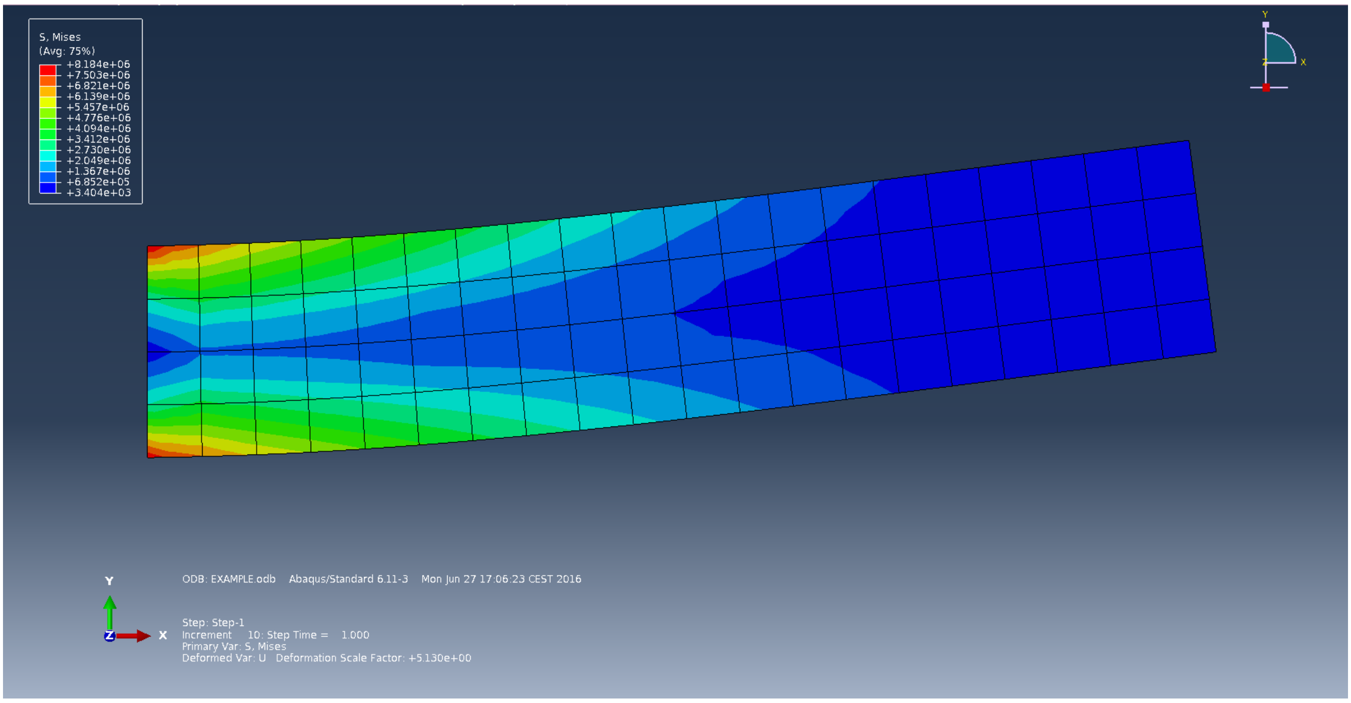
- *This opens some modulus, creates the viewport and open odb. It is to be noted that in line 11 ’pathname’ should be provided correctly.*

```
### CREATE OUTPUT ###
session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(
CONTOURS_ON_DEF, ))
session.printToFile(fileName='EXAMPLE', format=TIFF, canvasObjects=(
session.viewports['Viewport: 1'], ))
```

- *These lines set the display to the stress in deformed shape and it saves the figure.*

>> To run this script :

Follow the same steps as previous until step 6 and in step 7 - Use Command : **’abaqus cae script=**`EXAMPLE_ODB.py`’ (after you are inside folder where script is saved). Or else you can also give directly the path name where your script is saved soon after you login to Alien. **Login to Alien ⇒ ’pwd’⇒ ’abaqus cae script=/pathname/’**. After the scripts get executed, Abaqus cae GUI will open automatically calling python scripts and we can see the generated model with output result(stress in deformed shape) as shown in below *Figure 7*.



```
ODB: EXAMPLE.odb    Abaqus/Standard 6.11-3    Mon Jun 27 17:06:23 CEST 2016

Step: Step-1
Increment     10: Step Time =    1.000
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +5.130e+00
```

Figure 7: Von-Mises stress distribution in deformed shape

3 How to use Ansys ?

ANSYS is a general purpose FEA software, used to simulate interactions of all disciplines of physics, structural, vibration, fluid dynamics, heat transfer and electromagnetic for engineers. Normally there are two platforms of ANSYS. One is *'ANSYS Mechanical APDL'* environment and other one is *' ANSYS Workbench'* environment. The advantage of *Workbench* platform is that it is more user friendly than *Mechanical APDL* platform due to interaction with GUI.

Ansys is installed in ***'Lordavder Cluster'*** in LaCaN and its version is 'Ansys V15' . But its license is available in ***'Clonetroop Cluster'*** and right now there are only '1' license(Academic Research) of Ansys available in the *'Clonetroop'*. One has to get the access('username' and 'password') to *'Lordvader'* and *'Clonetroop'* first to use Ansys in LaCaN Computer's. To get the access mail to **'david.ortin@upc.edu'** stating your objective.

Limitations of ANSYS installed in LaCaN :

- One has to be aware that ANSYS License available currently in LaCaN is only for CFD analysis (CFX,Fluent,IC Engine).
- No License avaialable for Structural analysis.
- License is available only for *Workbench* platform not for *Mechanical APDL* platform.
- Dealing with the problems like calling scripts(Matlab and Python) in ANSYS is complicated due to its limited availability of license.
- Software crashes sometimes.
- Overall calculation time while getting results of analysis is slow sometimes.

>> Following are the steps need to follow to use Ansys software from LaCaN Computer's:

1. **Get the 'username' and 'password' for *Lordvader* and *Clonetroop* server from LaCaN sytems administrator.**

   ⇓

2. **Login with 'username' and 'password' in Lordvader**

   ⇓

3. **Open the terminal 'CTRL+ALT+T'**

   ⇓

4. **Type 'ssh -X <username>@clonetroop.upc.es' ('username' of Clonetroop server)**

   ⇓

5. **Password: <Type password> ('password' of Clonetroop server)**

   ⇓

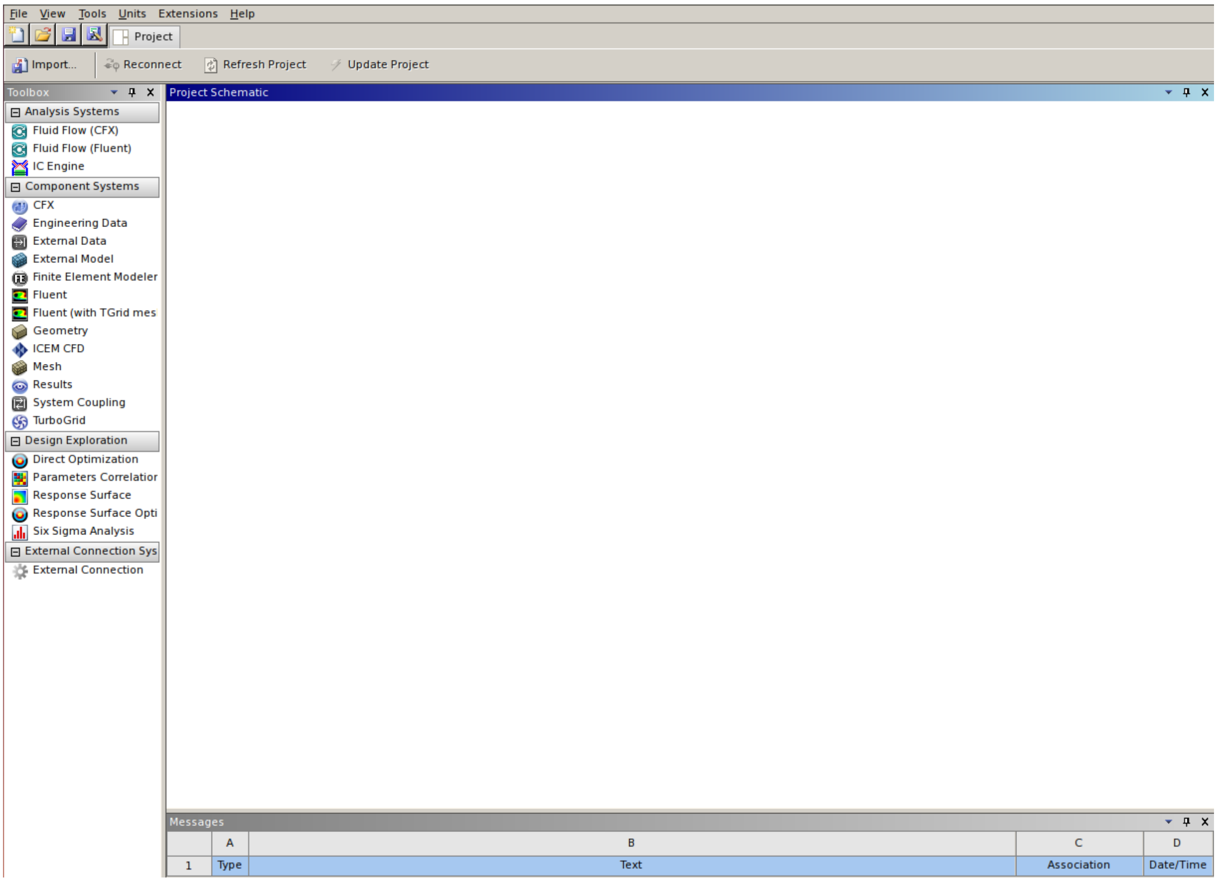
6. **Create your own folder using command 'mkdir <folder name>'**

   ⇓

7. **To open Ansys Workbench software use command - *'runbw2'***

   ⇓

8. **Soon after you run the command this window of workbench should appear as shown below.**

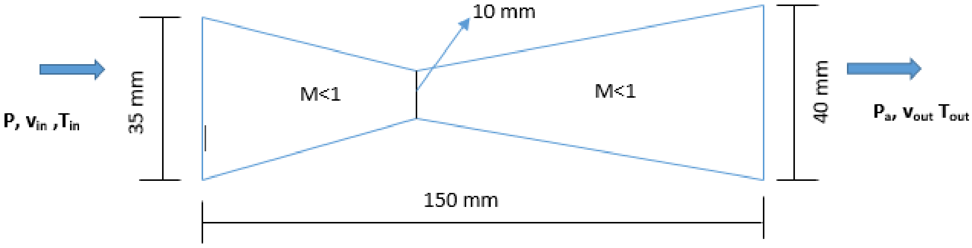


   ⇓

9. **Start with your workbench project.**

EXAMPLE : Convergent-Divergent Nozzle(Subsonic Flow)

To show the usage of ANSYS in LaCaN, I have considered example of *Convergent-Divergent Nozzle* and solved in *'Fluent'* as shown in *Figure* below with flow of ideal gas(air) at subsonic speeds(M<1) and viscid. And also assumed gas flow through nozzle is isentropic with no-slip condition.



Boundary Conditions :

- Inlet - For inlet boundary condition the mass-flow-inlet with the mass flow rate of 0.05 kg/s is used and the inlet temperature of 300K is used. And also it is to be noted that standard K-ϵ model with enhanced wall function is chosen in *'Fluent'*. Operating pressure is kept at 1atm.
- Outlet - The outlet boundary condition is set to pressure-outlet with the gauge pressure of 28675 Pa and temperature is set to 298.9K.

The geometry and mesh is created and the problem is solved with the above BC's using *'Density based solvers'* since we are dealing with subsonic flow . And the convergence criteria for all the residuals are set to 10^{-7}, finally run the calculations for 3000 iterations and obtained the converged results, which is shown in below figures.

>> Following are the steps need to follow to solve this problem in Ansys software from LaCaN Computer's:

1. **Get the 'username' and 'password' for *Lordvader* and *Clonetroop* server from LaCaN sytems administrator.**

   $\Downarrow$

2. **Login with 'username' and 'password' in Lordvader**

   $\Downarrow$

3. **Open the terminal 'CTRL+ALT+T'**

   $\Downarrow$

4. **Type 'ssh -X <username>@clonetroop.upc.es' ('username' of Clonetroop server)**

   $\Downarrow$

5. **Password: <Type password> ('password' of Clonetroop server)**

   $\Downarrow$

6. **Create your own folder using command 'mkdir <folder name>'**

   $\Downarrow$

7. **To open Ansys Workbench software use command - *'runbw2'***

   $\Downarrow$

8. **Open *'Fluent'*(for this example) in workbench and solve the problem.**

   $\Downarrow$

9. **Save results in folder.**

The results obtained after solving *Convergent-Divergnet Nozzle* (subsonic flow) problem in *'ANSYS Fluent'* is shown in below *Figures.*

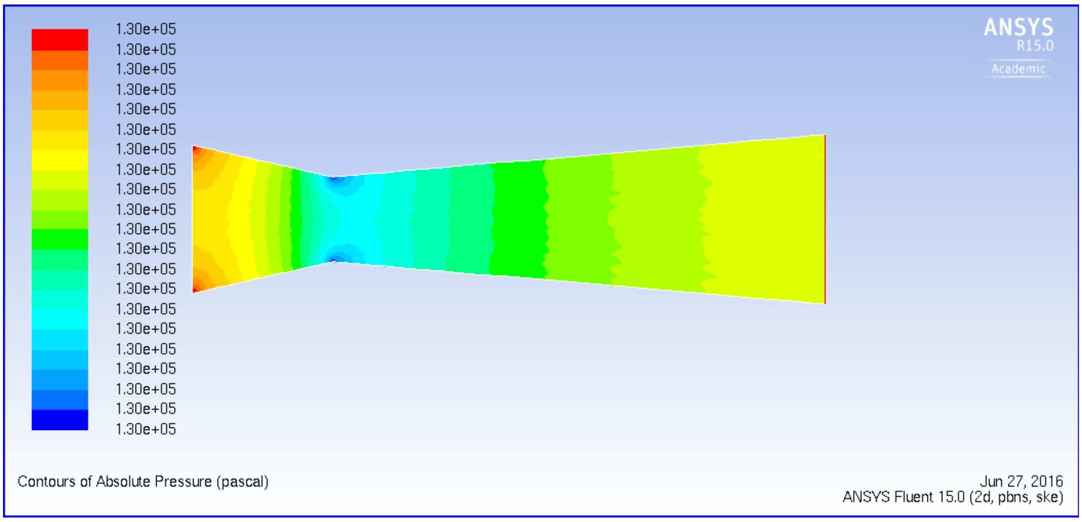


Figure 8: Pressure Distribution in (Pa)

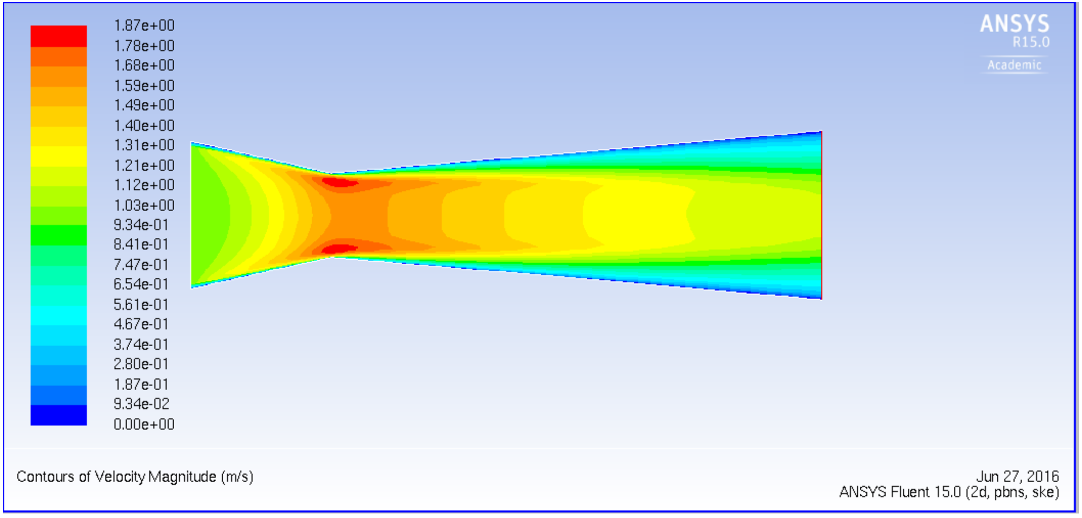


Figure 9: Velocity Distribution in (m/s)

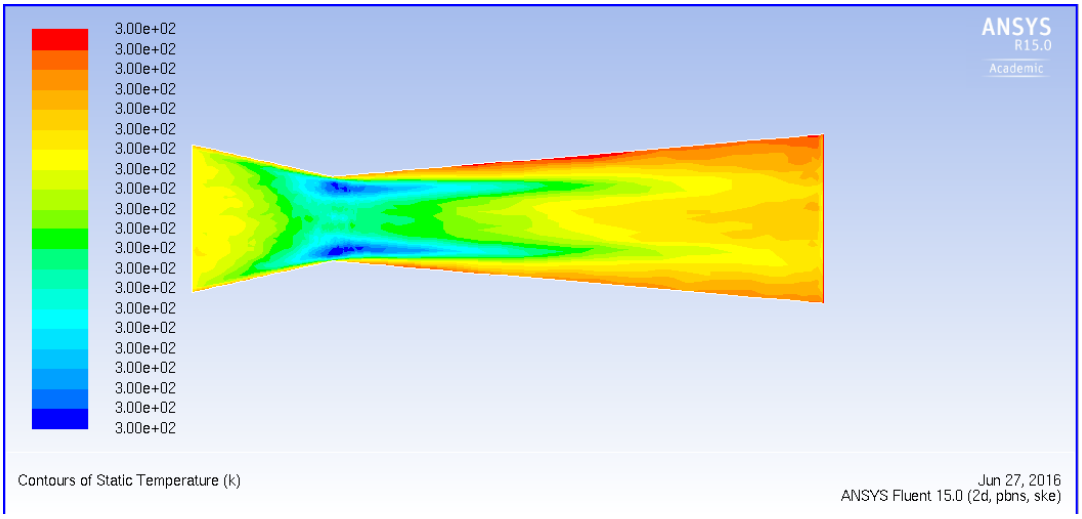


Figure 10: Temperature Distribution in (K)

4 How to use OpenFoam ?

OpenFOAM (**O**pen **F**ield **O**peration **A**nd **M**anipulation) is a free, open source CFD software developed primarily by OpenCFD Ltd since 2004, distributed by OpenCFD Ltd and the OpenFOAM Foundation. It is C++ library of highest quality of programming for solving computational continuum mechanics problems utilising all the features of Objected Oriented Programming (OOP).

Advantages:

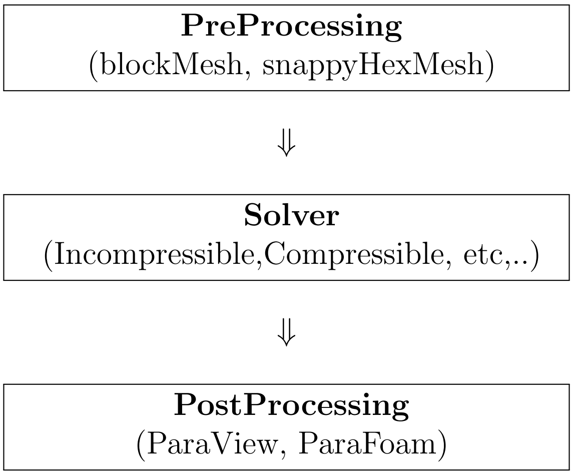
- Friendly syntax for partial differential equations.
- Unstructured polyhedral grid capabilities.
- Automatic parallelization of applications written using OpenFOAM high-level syntax.
- Wide range of applications and models ready to use.
- Commercial support and training provided by the developers.
- No license costs.

Disadvantages:

- Absence of an integrated graphical user interface.
- The Programmer's guide does not provide sufficient details, making the learning curve very steep if you need to write new applications or add functionality.

Source : https://en.wikipedia.org/wiki/OpenFOAM , http://www.openfoam.com/

Simulation Workflow of OpenFoam Follows these 3 main steps as shown below :



PreProcessing
(blockMesh, snappyHexMesh)

⇓

Solver
(Incompressible,Compressible, etc,..)

⇓

PostProcessing
(ParaView, ParaFoam)

For more details of OpenFoam Solvers, Tutorials and examples refer to this very useful links :
https://drive.google.com/file/d/0B5Kx6wTb8bgIUzhiT21TZmxJQkk/view
http://jibranhaider.weebly.com/openfoam.html
http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf
http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf

OpenFoam is installed in ***'Clonetroop Cluster'*** and ***'Lordvader Cluster'*** in LaCaN and its version is 'OpenFoam V2.4.0'. One has to get the access ('username' and 'password') for the two clusters *'Lordvader'* and *'Clonetroop'* first to use OpenFoam in LaCaN Computer's. To get the access mail to **'david.ortin@upc.edu'** stating your objective.

>> Following are the steps need to follow to use OpenFoam software from LaCaN Computer's:

1. **Get the 'username' and 'password' for *Lordvader* and *Clonetroop* server from LaCaN sytems administrator.**

   ⇓

2. **Login with 'username' and 'password' in Lordvader**

   ⇓

3. **Open the terminal 'CTRL+ALT+T'**

   ⇓

4. **The directory '/lordvader/<username>/OpenFOAM/OpenFOAM-2.4.0' must exist. It should be done initially when using first time :**

   - Use command - **'mkdir -p ∼/OpenFOAM/<username>-2.4.0'**

   ⇓

5. **Load the environment (every time you use it)**

   - Use Command : **'source /opt/openfoam240/etc/bashrc'**
   - Alternatively for regular OpenFOAM use, edit your bashrc file.
     Use command : '**gedit ∼/.bashrc**' and once it opens add following lines at end of file :

     ```
     source /opt/openfoam240/etc/bashrc
     export PYTHONPATH=/opt/python/lib/python2.7/site−packages:$PYTHONPATH
     export PATH=/opt/python/bin:$PATH
     ```

   ⇓

6. **Test the OpenFoam**

   - Copy the tutorial examples directory in the OpenFOAM distribution to the run directory. If the OpenFOAM environment variables are set correctly, then the following command will be correct:
     ' **cp -r** $FOAM_TUTORIALS $FOAM_RUN '

   - Run the first example case of incompressible laminar flow in a cavity:
     ' **cd** $FOAM_RUN**/tutorials/incompressible/icoFoam/cavity** '
     ↓
     blockMesh
     icoFoam
     paraFoam

For more detailed information open a copy of presentation by Mr.Gibran Hyder (copy paste the below link in 'Terminal window' after you logged into lordavder) :
- **evince /lordvader/courses/OpenFoam/presentation/presentation.pdf** &

EXAMPLE : Convergent-Divergent Nozzle(Subsonic Flow)

To show the usage of OpenFoam in LaCaN, I have considered same example *Convergent-Divergent Nozzle*, which I have used previously to show in *ANSYS Fluent.* After running the scripts written, obtained the results and then compared with *ANSYS Fluent* results.

All the conditions assumed are same as previous and OpenFoam scripts are written accordingly (attached in *Appendix 5*). Mesh file (.msh) is saved and imported from *Fluent.* Later it is then transferred in OpenFoam using command **'fluentMeshToFoam'** and built the grid with the OpenFOAM using command '**buildMesh**'.

>> Following are the steps need to follow to solve this problem in OpenFoam software from LaCaN Computer's :

1. **Get the 'username' and 'password' for *Lordvader* and *Clonetroop* server from LaCaN sytems administrator.**

   ⇓

2. **Login with 'username' and 'password' in Lordvader**

   ⇓

3. **Open the terminal 'CTRL+ALT+T'**

   ⇓

4. **Since I have already created my directory, I have saved my OpenFoam scripts(written in C++) in my folder of path :**

   ' **/lordvader/masters/anil/OpenFOAM/anil-2.4.0/tutorials/nozzle/** `$FOAM_RUN` '

   ⇓

5. **Now as my OpenFoam scripts are already in my folder, just need to follow these steps :**

   - **echo** `$FOAM_RUN`
   - **run**
   - **cd nozzle**
   - **Create and View mesh**
     -blockMesh
     -checkMesh
     -paraFoam
   - **Run the Solver**
     'rhoSimpleFoam' (Compressible steady state density based solver)
   - **Post Processing**
     -paraFoam

The post-processing results (pressure, velocity and Temperature distributions) after running for 1000 iterations are shown in below *Figures* due to limitations of software installed, from which we can observe that we obtained almost same contours and results as we obtained from *Ansys Fluent.* If we run for 3000 iterations we may obtain exact same results as *Ansys Fluent.*

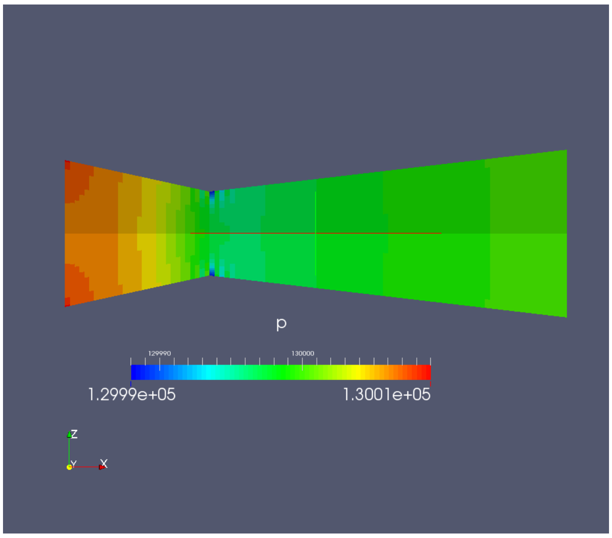


Figure 11: Pressure Distribution in (Pa)

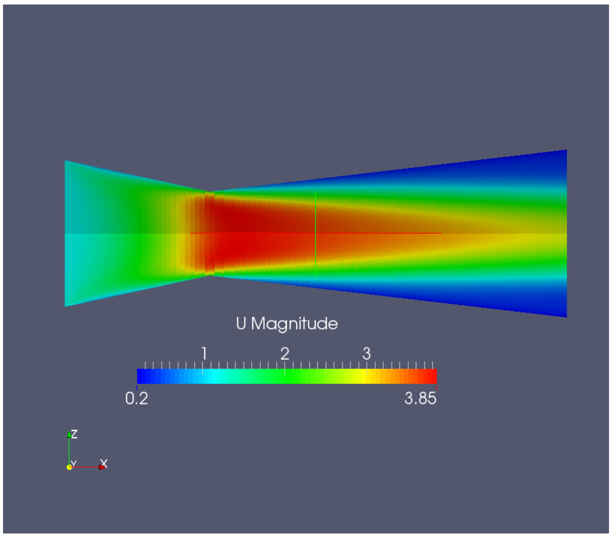


Figure 12: Velocity Distribution in (m/s)

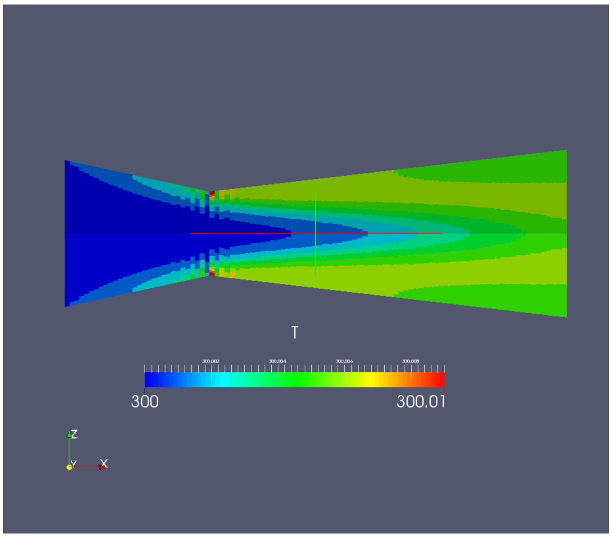


Figure 13: Temperature Distribution in (K)

References : http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf
http://cfdblogvienna.blogspot.com.es/p/computational-fluid-dynamics-advanced-a.html

>>**Some of the useful Solvers of OpenFoam :**

- *potentialFoam* - Solves for potential flow.
- *icoFoam* - Transient solver for incompressible, laminar flow of Newtonian fluids.
- *simpleFoam* - Steady state solver for incompressible, turbulent flow.
- *rhoCentralFoam* - Density based compressible flow solver based on central upwind scheme.
- *sonicFoam* - Transient solver for trans-sonic/supersonic, laminar/turbulent flow of a compressible gas.
- *multiPhaseEulerFoam* - Solver for multiple compressible fluid phases with heat transfer.

Appendices

Appendix 1: Input file of Cantilever beam 'PSE.inp'

```
*Heading
** Job name: PSE Model name: Model-1
** Generated by: Abaqus/CAE Student Edition 6.14-2
*Preprint, echo=NO, model=YES, history=NO, contact=NO
**
*PARAMETER
E=210e9
nu=0.3
** PARTS
**
*Part, name=Part-1
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1
*Node
      1,           2.,           2.,          20.
      2,           2.,           1.,          20.
      3,           2.,           0.,          20.
      4,           2.,           2.,          19.
      5,           2.,           1.,          19.
      6,           2.,           0.,          19.
      7,           2.,           2.,          18.
      8,           2.,           1.,          18.
      9,           2.,           0.,          18.
     10,           2.,           2.,          17.
     11,           2.,           1.,          17.
     12,           2.,           0.,          17.
     13,           2.,           2.,          16.
     14,           2.,           1.,          16.
     15,           2.,           0.,          16.
     16,           2.,           2.,          15.
     17,           2.,           1.,          15.
     18,           2.,           0.,          15.
     19,           2.,           2.,          14.
     20,           2.,           1.,          14.
     21,           2.,           0.,          14.
     22,           2.,           2.,          13.
     23,           2.,           1.,          13.
     24,           2.,           0.,          13.
     25,           2.,           2.,          12.
```

```
26,            2.,            1.,            12.
27,            2.,            0.,            12.
28,            2.,            2.,            11.
29,            2.,            1.,            11.
30,            2.,            0.,            11.
31,            2.,            2.,            10.
32,            2.,            1.,            10.
33,            2.,            0.,            10.
34,            2.,            2.,             9.
35,            2.,            1.,             9.
36,            2.,            0.,             9.
37,            2.,            2.,             8.
38,            2.,            1.,             8.
39,            2.,            0.,             8.
40,            2.,            2.,             7.
41,            2.,            1.,             7.
42,            2.,            0.,             7.
43,            2.,            2.,             6.
44,            2.,            1.,             6.
45,            2.,            0.,             6.
46,            2.,            2.,             5.
47,            2.,            1.,             5.
48,            2.,            0.,             5.
49,            2.,            2.,             4.
50,            2.,            1.,             4.
51,            2.,            0.,             4.
52,            2.,            2.,             3.
53,            2.,            1.,             3.
54,            2.,            0.,             3.
55,            2.,            2.,             2.
56,            2.,            1.,             2.
57,            2.,            0.,             2.
58,            2.,            2.,             1.
59,            2.,            1.,             1.
60,            2.,            0.,             1.
61,            2.,            2.,             0.
62,            2.,            1.,             0.
63,            2.,            0.,             0.
64,            1.,            2.,            20.
65,            1.,            1.,            20.
66,            1.,            0.,            20.
67,            1.,            2.,            19.
68,            1.,            1.,            19.
69,            1.,            0.,            19.
70,            1.,            2.,            18.
71,            1.,            1.,            18.
72,            1.,            0.,            18.
73,            1.,            2.,            17.
74,            1.,            1.,            17.
75,            1.,            0.,            17.
```

```
    76,           1.,           2.,          16.
    77,           1.,           1.,          16.
    78,           1.,           0.,          16.
    79,           1.,           2.,          15.
    80,           1.,           1.,          15.
    81,           1.,           0.,          15.
    82,           1.,           2.,          14.
    83,           1.,           1.,          14.
    84,           1.,           0.,          14.
    85,           1.,           2.,          13.
    86,           1.,           1.,          13.
    87,           1.,           0.,          13.
    88,           1.,           2.,          12.
    89,           1.,           1.,          12.
    90,           1.,           0.,          12.
    91,           1.,           2.,          11.
    92,           1.,           1.,          11.
    93,           1.,           0.,          11.
    94,           1.,           2.,          10.
    95,           1.,           1.,          10.
    96,           1.,           0.,          10.
    97,           1.,           2.,           9.
    98,           1.,           1.,           9.
    99,           1.,           0.,           9.
   100,           1.,           2.,           8.
   101,           1.,           1.,           8.
   102,           1.,           0.,           8.
   103,           1.,           2.,           7.
   104,           1.,           1.,           7.
   105,           1.,           0.,           7.
   106,           1.,           2.,           6.
   107,           1.,           1.,           6.
   108,           1.,           0.,           6.
   109,           1.,           2.,           5.
   110,           1.,           1.,           5.
   111,           1.,           0.,           5.
   112,           1.,           2.,           4.
   113,           1.,           1.,           4.
   114,           1.,           0.,           4.
   115,           1.,           2.,           3.
   116,           1.,           1.,           3.
   117,           1.,           0.,           3.
   118,           1.,           2.,           2.
   119,           1.,           1.,           2.
   120,           1.,           0.,           2.
   *Element, type=C3D20R
1,  64,  65,  68,  67,   1,   2,   5,   4, 193, 192, 191, 190, 194, 195, 196,
   197, 199, 198, 200, 201
2,  65,  66,  69,  68,   2,   3,   6,   5, 204, 203, 202, 192, 205, 206, 207,
   195, 198, 208, 209, 200
```

```
 3,  67,  68,  71,  70,   4,   5,   8,   7, 191, 212, 211, 210, 196, 213, 214,
    215, 201, 200, 216, 217
 4,  68,  69,  72,  71,   5,   6,   9,   8, 202, 219, 218, 212, 207, 220, 221,
    213, 200, 209, 222, 216
 5,  70,  71,  74,  73,   7,   8,  11,  10, 211, 225, 224, 223, 214, 226, 227,
    228, 217, 216, 229, 230
 6,  71,  72,  75,  74,   8,   9,  12,  11, 218, 232, 231, 225, 221, 233, 234,
    226, 216, 222, 235, 229
 7,  73,  74,  77,  76,  10,  11,  14,  13, 224, 238, 237, 236, 227, 239, 240,
    241, 230, 229, 242, 243
 8,  74,  75,  78,  77,  11,  12,  15,  14, 231, 245, 244, 238, 234, 246, 247,
    239, 229, 235, 248, 242
 9,  76,  77,  80,  79,  13,  14,  17,  16, 237, 251, 250, 249, 240, 252, 253,
    254, 243, 242, 255, 256
10,  77,  78,  81,  80,  14,  15,  18,  17, 244, 258, 257, 251, 247, 259, 260,
    252, 242, 248, 261, 255
11,  79,  80,  83,  82,  16,  17,  20,  19, 250, 264, 263, 262, 253, 265, 266,
    267, 256, 255, 268, 269
12,  80,  81,  84,  83,  17,  18,  21,  20, 257, 271, 270, 264, 260, 272, 273,
    265, 255, 261, 274, 268
13,  82,  83,  86,  85,  19,  20,  23,  22, 263, 277, 276, 275, 266, 278, 279,
    280, 269, 268, 281, 282
14,  83,  84,  87,  86,  20,  21,  24,  23, 270, 284, 283, 277, 273, 285, 286,
    278, 268, 274, 287, 281
15,  85,  86,  89,  88,  22,  23,  26,  25, 276, 290, 289, 288, 279, 291, 292,
    293, 282, 281, 294, 295
16,  86,  87,  90,  89,  23,  24,  27,  26, 283, 297, 296, 290, 286, 298, 299,
    291, 281, 287, 300, 294
17,  88,  89,  92,  91,  25,  26,  29,  28, 289, 303, 302, 301, 292, 304, 305,
    306, 295, 294, 307, 308
18,  89,  90,  93,  92,  26,  27,  30,  29, 296, 310, 309, 303, 299, 311, 312,
    304, 294, 300, 313, 307
19,  91,  92,  95,  94,  28,  29,  32,  31, 302, 316, 315, 314, 305, 317, 318,
    319, 308, 307, 320, 321
20,  92,  93,  96,  95,  29,  30,  33,  32, 309, 323, 322, 316, 312, 324, 325,
    317, 307, 313, 326, 320
21,  94,  95,  98,  97,  31,  32,  35,  34, 315, 329, 328, 327, 318, 330, 331,
    332, 321, 320, 333, 334
22,  95,  96,  99,  98,  32,  33,  36,  35, 322, 336, 335, 329, 325, 337, 338,
    330, 320, 326, 339, 333
23,  97,  98, 101, 100,  34,  35,  38,  37, 328, 342, 341, 340, 331, 343, 344,
    345, 334, 333, 346, 347
24,  98,  99, 102, 101,  35,  36,  39,  38, 335, 349, 348, 342, 338, 350, 351,
    343, 333, 339, 352, 346
25, 100, 101, 104, 103,  37,  38,  41,  40, 341, 355, 354, 353, 344, 356, 357,
    358, 347, 346, 359, 360
26, 101, 102, 105, 104,  38,  39,  42,  41, 348, 362, 361, 355, 351, 363, 364,
    356, 346, 352, 365, 359
27, 103, 104, 107, 106,  40,  41,  44,  43, 354, 368, 367, 366, 357, 369, 370,
    371, 360, 359, 372, 373
```

```
28, 104, 105, 108, 107,  41,  42,  45,  44, 361, 375, 374, 368, 364, 376, 377,
    369, 359, 365, 378, 372
29, 106, 107, 110, 109,  43,  44,  47,  46, 367, 381, 380, 379, 370, 382, 383,
    384, 373, 372, 385, 386
30, 107, 108, 111, 110,  44,  45,  48,  47, 374, 388, 387, 381, 377, 389, 390,
    382, 372, 378, 391, 385
31, 109, 110, 113, 112,  46,  47,  50,  49, 380, 394, 393, 392, 383, 395, 396,
    397, 386, 385, 398, 399
32, 110, 111, 114, 113,  47,  48,  51,  50, 387, 401, 400, 394, 390, 402, 403,
    395, 385, 391, 404, 398
33, 112, 113, 116, 115,  49,  50,  53,  52, 393, 407, 406, 405, 396, 408, 409,
    410, 399, 398, 411, 412
34, 113, 114, 117, 116,  50,  51,  54,  53, 400, 414, 413, 407, 403, 415, 416,
    408, 398, 404, 417, 411
35, 115, 116, 119, 118,  52,  53,  56,  55, 406, 420, 419, 418, 409, 421, 422,
    423, 412, 411, 424, 425
36, 116, 117, 120, 119,  53,  54,  57,  56, 413, 427, 426, 420, 416, 428, 429,
    421, 411, 417, 430, 424
37, 118, 119, 122, 121,  55,  56,  59,  58, 419, 433, 432, 431, 422, 434, 435,
    436, 425, 424, 437, 438
38, 119, 120, 123, 122,  56,  57,  60,  59, 426, 440, 439, 433, 429, 441, 442,
    434, 424, 430, 443, 437
39, 121, 122, 125, 124,  58,  59,  62,  61, 432, 446, 445, 444, 435, 447, 448,
    449, 438, 437, 450, 451
40, 122, 123, 126, 125,  59,  60,  63,  62, 439, 453, 452, 446, 442, 454, 455,
    447, 437, 443, 456, 450
41, 127, 128, 131, 130,  64,  65,  68,  67, 460, 459, 458, 457, 193, 192, 191,
    190, 462, 461, 463, 464
42, 128, 129, 132, 131,  65,  66,  69,  68, 467, 466, 465, 459, 204, 203, 202,
    192, 461, 468, 469, 463
43, 130, 131, 134, 133,  67,  68,  71,  70, 458, 472, 471, 470, 191, 212, 211,
    210, 464, 463, 473, 474
44, 131, 132, 135, 134,  68,  69,  72,  71, 465, 476, 475, 472, 202, 219, 218,
    212, 463, 469, 477, 473
45, 133, 134, 137, 136,  70,  71,  74,  73, 471, 480, 479, 478, 211, 225, 224,
    223, 474, 473, 481, 482
46, 134, 135, 138, 137,  71,  72,  75,  74, 475, 484, 483, 480, 218, 232, 231,
    225, 473, 477, 485, 481
47, 136, 137, 140, 139,  73,  74,  77,  76, 479, 488, 487, 486, 224, 238, 237,
    236, 482, 481, 489, 490
48, 137, 138, 141, 140,  74,  75,  78,  77, 483, 492, 491, 488, 231, 245, 244,
    238, 481, 485, 493, 489
49, 139, 140, 143, 142,  76,  77,  80,  79, 487, 496, 495, 494, 237, 251, 250,
    249, 490, 489, 497, 498
50, 140, 141, 144, 143,  77,  78,  81,  80, 491, 500, 499, 496, 244, 258, 257,
    251, 489, 493, 501, 497
51, 142, 143, 146, 145,  79,  80,  83,  82, 495, 504, 503, 502, 250, 264, 263,
    262, 498, 497, 505, 506
52, 143, 144, 147, 146,  80,  81,  84,  83, 499, 508, 507, 504, 257, 271, 270,
    264, 497, 501, 509, 505
```

```
53, 145, 146, 149, 148,  82,  83,  86,  85, 503, 512, 511, 510, 263, 277, 276,
    275, 506, 505, 513, 514
54, 146, 147, 150, 149,  83,  84,  87,  86, 507, 516, 515, 512, 270, 284, 283,
    277, 505, 509, 517, 513
55, 148, 149, 152, 151,  85,  86,  89,  88, 511, 520, 519, 518, 276, 290, 289,
    288, 514, 513, 521, 522
56, 149, 150, 153, 152,  86,  87,  90,  89, 515, 524, 523, 520, 283, 297, 296,
    290, 513, 517, 525, 521
57, 151, 152, 155, 154,  88,  89,  92,  91, 519, 528, 527, 526, 289, 303, 302,
    301, 522, 521, 529, 530
58, 152, 153, 156, 155,  89,  90,  93,  92, 523, 532, 531, 528, 296, 310, 309,
    303, 521, 525, 533, 529
59, 154, 155, 158, 157,  91,  92,  95,  94, 527, 536, 535, 534, 302, 316, 315,
    314, 530, 529, 537, 538
60, 155, 156, 159, 158,  92,  93,  96,  95, 531, 540, 539, 536, 309, 323, 322,
    316, 529, 533, 541, 537
61, 157, 158, 161, 160,  94,  95,  98,  97, 535, 544, 543, 542, 315, 329, 328,
    327, 538, 537, 545, 546
62, 158, 159, 162, 161,  95,  96,  99,  98, 539, 548, 547, 544, 322, 336, 335,
    329, 537, 541, 549, 545
63, 160, 161, 164, 163,  97,  98, 101, 100, 543, 552, 551, 550, 328, 342, 341,
    340, 546, 545, 553, 554
64, 161, 162, 165, 164,  98,  99, 102, 101, 547, 556, 555, 552, 335, 349, 348,
    342, 545, 549, 557, 553
65, 163, 164, 167, 166, 100, 101, 104, 103, 551, 560, 559, 558, 341, 355, 354,
    353, 554, 553, 561, 562
66, 164, 165, 168, 167, 101, 102, 105, 104, 555, 564, 563, 560, 348, 362, 361,
    355, 553, 557, 565, 561
67, 166, 167, 170, 169, 103, 104, 107, 106, 559, 568, 567, 566, 354, 368, 367,
    366, 562, 561, 569, 570
68, 167, 168, 171, 170, 104, 105, 108, 107, 563, 572, 571, 568, 361, 375, 374,
    368, 561, 565, 573, 569
69, 169, 170, 173, 172, 106, 107, 110, 109, 567, 576, 575, 574, 367, 381, 380,
    379, 570, 569, 577, 578
70, 170, 171, 174, 173, 107, 108, 111, 110, 571, 580, 579, 576, 374, 388, 387,
    381, 569, 573, 581, 577
71, 172, 173, 176, 175, 109, 110, 113, 112, 575, 584, 583, 582, 380, 394, 393,
    392, 578, 577, 585, 586
72, 173, 174, 177, 176, 110, 111, 114, 113, 579, 588, 587, 584, 387, 401, 400,
    394, 577, 581, 589, 585
73, 175, 176, 179, 178, 112, 113, 116, 115, 583, 592, 591, 590, 393, 407, 406,
    405, 586, 585, 593, 594
74, 176, 177, 180, 179, 113, 114, 117, 116, 587, 596, 595, 592, 400, 414, 413,
    407, 585, 589, 597, 593
75, 178, 179, 182, 181, 115, 116, 119, 118, 591, 600, 599, 598, 406, 420, 419,
    418, 594, 593, 601, 602
76, 179, 180, 183, 182, 116, 117, 120, 119, 595, 604, 603, 600, 413, 427, 426,
    420, 593, 597, 605, 601
77, 181, 182, 185, 184, 118, 119, 122, 121, 599, 608, 607, 606, 419, 433, 432,
    431, 602, 601, 609, 610
```

```
78, 182, 183, 186, 185, 119, 120, 123, 122, 603, 612, 611, 608, 426, 440, 439,
    433, 601, 605, 613, 609
79, 184, 185, 188, 187, 121, 122, 125, 124, 607, 616, 615, 614, 432, 446, 445,
    444, 610, 609, 617, 618
80, 185, 186, 189, 188, 122, 123, 126, 125, 611, 620, 619, 616, 439, 453, 452,
    446, 609, 613, 621, 617
*Nset, nset=Set-1, generate
   1,  621,    1
*Elset, elset=Set-1, generate
  1,  80,   1
** Section: Section-1
*Solid Section, elset=Set-1, material=Material-1
,
*End Instance
**
*Nset, nset=Set-1, instance=Part-1-1
 64,
*Nset, nset=_PickedSet8, internal, instance=Part-1-1
  61,  62,  63, 124, 125, 126, 187, 188, 189, 445, 448, 450, 451, 452, 455, 456
 615, 617, 618, 619, 621
*Elset, elset=_PickedSet8, internal, instance=Part-1-1
 39, 40, 79, 80
*Elset, elset=_Surf-1_S6, internal, instance=Part-1-1, generate
  1,  79,   2
*Surface, type=ELEMENT, name=Surf-1
_Surf-1_S6, S6
*End Assembly
**
** MATERIALS
**
*Material, name=Material-1
*Density
7850.,
*Elastic
 <E>,<nu>
*Plastic
 2.4e+08,0.
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
_PickedSet8, 1, 1
_PickedSet8, 2, 2
_PickedSet8, 3, 3
_PickedSet8, 4, 4
_PickedSet8, 5, 5
_PickedSet8, 6, 6
** ----------------------------------------------------------------
**
```

```
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=NO
*Static
1., 1., 1e-05, 1.
**
** LOADS
**
** Name: Load-1   Type: Pressure
*Dsload
Surf-1, P, 50000.
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-2
**
*Output, history
*Node Output, nset=Set-1
U1, U2, U3, UR1, UR2, UR3
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

Appendix 2: Python script for parametric study(material properties) of Cantilever beam 'PSE.psf'

```
#create study
cbeam=ParStudy(par=('E','nu'))
#defining the parameters
cbeam.define(DISCRETE,par='E',domain=(100e9,200e9,300e9))
cbeam.define(CONTINUOUS,par='nu',domain=(0.15,0.35))
#sampling
cbeam.sample(INTERVAL,par='E',interval=1)
cbeam.sample(NUMBER,par='nu',number=3)
#combine the samples
cbeam.combine(MESH,name='cantilever')
#generate the analysis job data
cbeam.generate(template='PSE')
#execute the jobs
cbeam.execute(ALL)
#read the results from ODB and Report
cbeam.output(step=1,file=ODB)
cbeam.gather(results='VDISP',variable='U2',node=64,request=HISTORY,instance="Part-1-1")
cbeam.report(FILE,results=('VDISP'),par=('E','nu'),file='cbeam.PSE.psr')
```

Appendix 3: Python script for creating model database of 2D Planar Shell 'EXAMPLE_MDB.py'

```
#load modulus
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
### PART ###
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=20.0)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-5.0, -1.0),
point2=(5.0, 1.0))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1', type=
DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
### MATERIAL & SECTION ###
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((1000000000.0,
0.3), ))
mdb.models['Model-1'].HomogeneousSolidSection(material='Material-1', name=
'Section-1', thickness=None)
mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
offsetField='', offsetType=MIDDLE_SURFACE, region=Region(
faces=mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667,
-0.333333, 0.0), (0.0, 0.0, 1.0)), )), sectionName='Section-1')
### SET & SURFACE ###
mdb.models['Model-1'].parts['Part-1'].Set(edges=
mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-5.0, -0.5, 0.0), )),
name='Set-1')
mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=
mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-2.5, 1.0, 0.0), )))
### MESH ###
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=QUAD, regions=
mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667, -0.333333,
0.0), )), technique=STRUCTURED)
mdb.models['Model-1'].parts['Part-1'].setElementType(elemTypes=(ElemType(
elemCode=CPS8R, elemLibrary=STANDARD), ElemType(elemCode=CPS6M,
elemLibrary=STANDARD)), regions=(
mdb.models['Model-1'].parts['Part-1'].faces.findAt(((-1.666667, -0.333333,
0.0), )), ))
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=0.5)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
### ASSEMBLY ###
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-1-1',
part=mdb.models['Model-1'].parts['Part-1'])
mdb.models['Model-1'].rootAssembly.regenerate()
### STEP, BC & LOAD ###
mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1',
previous='Initial')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
'BC-1', region=
mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].sets['Set-1'], u1=
0.0, u2=0.0, ur3=0.0)
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
distributionType=UNIFORM, field='', magnitude=-100000.0, name='Load-1',
region=
mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])
### JOB & CALCULATE ###
mdb.Job(contactPrint=OFF, description='', echoPrint=OFF, explicitPrecision=
SINGLE, historyPrint=OFF, memory=90, memoryUnits=PERCENTAGE, model=
'Model-1', modelPrint=OFF, multiprocessingMode=DEFAULT, name='EXAMPLE',
nodalOutputPrecision=SINGLE, numCpus=1, numDomains=1,
parallelizationMethodExplicit=DOMAIN, scratch='', type=ANALYSIS,
userSubroutine='')
mdb.jobs['EXAMPLE'].submit(consistencyChecking=OFF)
```

Appendix 4: Python script for creating Output Database of 2D Planar Shell 'EXAMPLE_ODB.py'

```
#open modulus, create viewport and open odb
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=268.952117919922,
height=154.15299987793)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
o1 = session.openOdb(name='anil/abaqus/EXAMPLE.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
### CREATE OUTPUT ###
session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(
CONTOURS_ON_DEF, ))
session.printToFile(fileName='EXAMPLE', format=TIFF, canvasObjects=(
session.viewports['Viewport: 1'], ))
```

Appendix 5: OpenFoam scripts for Converging-Diverging Nozzle(Subsonic Flow).

1. Mesh Generation - 'blockMeshDict' file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class dictionary;
object blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters 0.001;
vertices
(
(0 0 0)
(68.68 0 0)
(0 -5.543455411358772 35)
(68.68 -3.167688806490727 20)
(68.68 3.167688806490727 20)
(0 5.543455411358772 35)
(238.8 0 0)
(238.8 -6.335377612981453 40)
(238.8 6.335377612981453 40)
);
blocks
(
hex (0 1 1 0 2 3 4 5) (60 1 100) simpleGrading (1 1 0.18)
hex (1 6 6 1 3 7 8 4) (90 1 100) simpleGrading (1 1 0.18)
);
edges
(
arc 5 2 (0 0 -35.436279402580105)
arc 4 3 (68.68 0 -20.249302515760061)
arc 4 3 (68.68 0 -20.249302515760061)
arc 8 7 (238.8 0 -40.498605031520121)
);
patches
(
patch
inlet
(
(0 2 5 0)
)
patch
- B1 -
outlet
(
(6 7 8 6)
)
wedge
axi_symm-f
(
(0 1 3 2)
(1 6 7 3)
)
wedge
axi_symm-r
(
(0 1 4 5)
(1 6 8 4)
)
wall
wall
(
(2 3 4 5)
(3 7 8 4)
)
);
mergePatchPairs
(
);
// ************************************************************************* //
```

2. Initial and Boundary Conditions - Pressure - 'p' file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 130000;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            fixedValue;
        value           uniform 130000;
    }

    wall
    {
        type            zeroGradient;
    }

    axi_symm-f
    {
        type            wedge;
    }

    axi_symm-r
    {
        type            wedge;
    }
}

// ************************************************************************* //
```

3. Initial and Boundary Conditions - Temperature - 'T' file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 300;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 300;
    }

    outlet
    {
        type            zeroGradient;
    }

    wall
    {
        type            zeroGradient;
    }

    axi_symm-f
    {
        type            wedge;
    }

    axi_symm-r
    {
        type            wedge;
    }
}

// ************************************************************************* //
```

4. Initial and Boundary Conditions - Velocity - 'U' file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (7.68 0 0);

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (7.68 0 0);
    }

    outlet
    {
        type            zeroGradient;
    }

wall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    axi_symm-f
    {
        type            wedge;
    }

    axi_symm-r
    {
        type            wedge;
    }

}

// ************************************************************************** //
```

5. Initial and Boundary Conditions - Turbulent Kinetic Energy - 'k' file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0.0001;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 0.0001;
    }
    outlet
    {
        type            zeroGradient;
    }
    wall
    {
        type            compressible::kqRWallFunction;
        value           uniform 0.0001;
    }
    axi_symm-f
    {
        type            wedge;
    }

    axi_symm-r
    {
        type            wedge;
    }
}
```

```
// ************************************************************************* //
```

6. RAS Properties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      RASProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

RASModel        kEpsilon;

turbulence      off;

printCoeffs     off;


// ************************************************************************* //
```

7. Thermophysical Properties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

thermoType
{
    type            hePsiThermo;
    mixture         pureMixture;
    transport       sutherland;
    thermo          hConst;
    equationOfState perfectGas;
    specie          specie;
    energy          sensibleInternalEnergy;
}

mixture
{
    specie
    {
        nMoles      1;
        molWeight   28.9;
    }
    thermodynamics
    {
        Cp          1005;
        Hf          0;
    }
    transport
    {
        As          1.4792e-06;
        Ts          116;
mu          1.79e-05;
        Pr          1;
    }
}


// ************************************************************************* //
```

8. controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     rhoSimpleFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         5000;

deltaT          1;

writeControl    timeStep;

writeInterval   5000;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

graphFormat     raw;

runTimeModifiable true;


// ************************************************************************* //
```

9. fVSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         steadyState;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    div(phi,U)      bounded Gauss upwind;
    div((muEff*dev2(T(grad(U))))) Gauss linear;
    div(phi,e)      bounded Gauss upwind;
    div(phi,epsilon) bounded Gauss upwind;
    div(phi,k)      bounded Gauss upwind;
    div(phi,Ekp)    bounded Gauss upwind;
}

laplacianSchemes
{
    laplacian(muEff,U) Gauss linear corrected;
    laplacian(alphaEff,e) Gauss linear corrected;
    laplacian((rho*(1|A(U))),p) Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DkEff,k) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p               ;
}


// ************************************************************************* //
```

10. fVSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-07;
        relTol          0.01;
        smoother        GaussSeidel;
        cacheAgglomeration off;
        nCellsInCoarsestLevel 20;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }
    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }

    e
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-07;
        relTol          0.1;
    }

    "(k|epsilon)"
    {
        $U;
        tolerance       1e-07;
        relTol          0.1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;
    rhoMin          rhoMin [ 1 -3 0 0 0 ] 0.5;
    rhoMax          rhoMax [ 1 -3 0 0 0 ] 1.5;

    residualControl
    {
        p               1e-2;
        U               1e-3;
        e               1e-3;

        // possibly check turbulence fields
        "(k|epsilon|omega)" 1e-3;
    }
}

relaxationFactors
{
    fields
    {
        p               0.3;
        rho             0.05;
    }
    equations
    {
        U               0.7;
        k               0.7;
        epsilon         0.7;
        R               0.7;
        nuTilda         0.7;
    }
}


// ************************************************************************* //
```