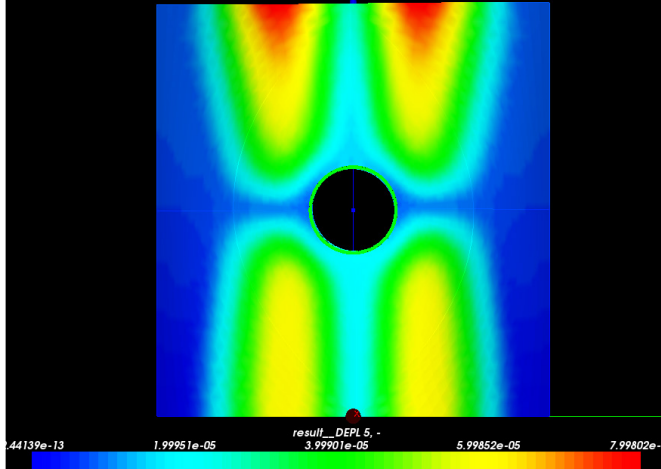
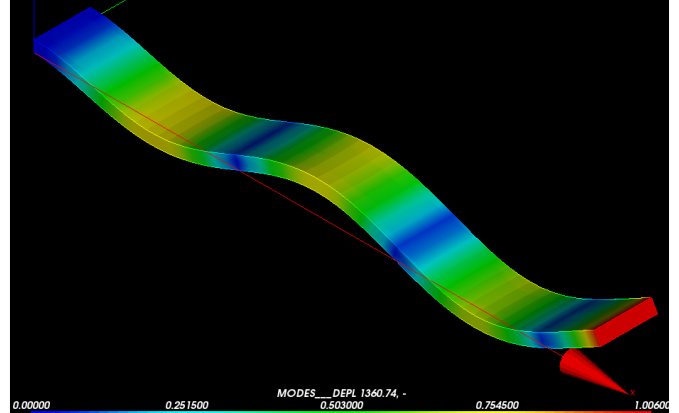


Intermediate Finite Element Analysis using Open source Software

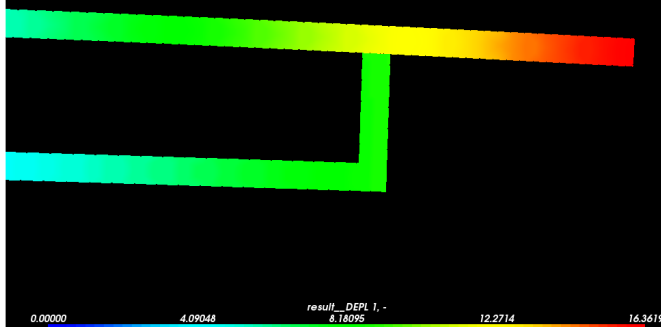
Moment on Nozzle



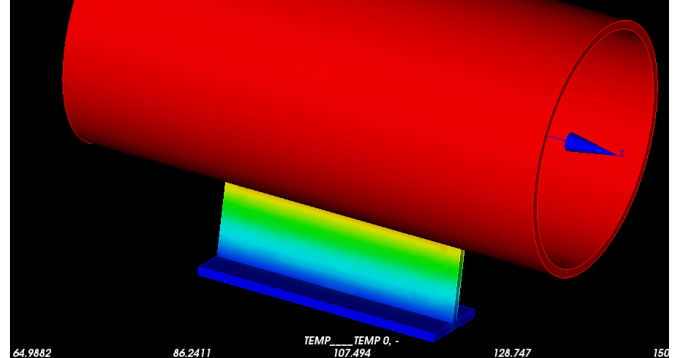
Natural frequency



Non-Linear Contact



Thermal



- by Dharmit A. Thakore

Intermediate Finite Element Analysis with Open Source Software

First Edition

Intermediate Finite Element Analysis with Open Source Software

First Edition

Dharmit Thakore, CPEng, RPEQ

Moonish Ent. Pty. Ltd.

Brisbane, QLD, Australia

Moonish Enterprises Pty Ltd

GPO Box 1299, Brisbane, QLD 4001, Australia

2014

Credits and Copyright

Written by: Dharmit Thakore

dharmit@moonish.biz

Publisher: Moonish Ent. Pty. Ltd

engineering@moonish.biz

<http://engineering.moonish.biz>

Graphic Design / Layout: Lomesha Thakore

lomesha27@gmail.com

Edition 1 ©2014 Dharmit Thakore

No part of this publication may be reproduced, stored or transmitted in any form or by any means, electronic, mechanical or otherwise, without prior written consent from the publisher, except for the inclusion of brief quotations in a review. You may store the pdf on your computer and backups. You may print one copy of this book for your own personal use.

Disclaimer: The information contained in this book is based on the author's experience, knowledge and opinions. The author and publisher will not be held liable for the use or misuse of the information in this book.

To
My wife,
Our beloved son
&
Open Source Software

About the Author

Dharmit Thakore is the Director of Moonish Enterprises Pty Ltd at Brisbane, Queensland, Australia. He practices as a Mechanical / Piping Engineer in Queensland. He received his Bachelor's degree from Birla Vishwakarma Mahavidhyalaya, Vallabh Vidhyanagar, Gujarat, India which was affiliated with Sardar Patel University. He started his engineering career as a young Graduate in Larsen & Toubro – Sargent & Lundy, Vadodara. He came to Australia for further studies and settled here. He received his Registered Professional Engineer in Queensland (RPEQ) recognition early in his career and subsequently obtained his Chartered Professional Engineer (CPEng) as a Mechanical / Piping Engineer.

Dharmit has broad interests, which include finite element analysis, design, optimization and Open Source software. He is a member of ASME, Engineers Australia and Board of Professional Engineers in Queensland.

Table of Contents

Foreword	xiii
This book is written for	xiii
This book is not written for	xiii
What are the steps in Finite Element Analysis	xv
Study Cases	xvii
Parametric Modelling in Salome for Geometry and Mesh generation	xvii
Combining element types in a single FE Analysis	xvii
Non Linear Material Analysis	xvii
Contact FE analysis	xvii
Modal Analysis	xvii
Thermal Analysis	xviii
Parametric Modelling of a Pressure Vessel Geometry	1
Step 1: Description of the problem	2
Step 2: Input values for the FE analysis	2
Step 3: Parametric Modelling by direct editing of .py file	2
Step 4: Meshing using Notebook	30
Change Parameter in Notebook	34
Summary	36
Adding Moments to a face in 3D model	37
Step 1: Description of the problem	38
Step 2: Input values for the FE analysis	38
Step 3 and 4: Generating Geometry by Parametric Modelling and editing mesh	38
Step 5, 6, 7, 8 and 9: Reusing .comm file from Case9 and editing it	44
Step 10: Run the analysis	49
Step 11: Post Processing of the Results	50
Summary	55
Combining Element types in a single FE Analysis	56
Step 1: Purpose of the FE Analysis / Description of the problem	57
Step 2: Input values for the FE analysis	57
Step 3: Model Geometry	57
Step 4: Meshing Geometry	66
Step 5, 6, 7, 8 and 9: Generating command file by hand	69
Step 10: Run the analysis	73
Step 11: Post Processing of the Results	75
Summary	80
Non-Linear Material FE Analysis	81
Step 1: Purpose of the FE Analysis / Description of the problem	82
Step 2: Input values for the FE analysis	82
Step 3: Model Geometry	83
Step 4: Meshing Geometry	85

Step 5, 6, 7, 8 and 9: Modifying comm file created by Efficient	88
Step 10: Run the analysis	96
Step 11: Post Processing of the Results	102
Summary	106
Non-Linear Material – Real Curve FE Analysis	108
Step 1: Purpose of the FE Analysis / Description of the problem	109
Step 2: Input values for the FE analysis	109
Step 3: Model Geometry	110
Step 4: Meshing Geometry	110
Step 5, 6, 7, 8 and 9: Modifying comm file created by Efficient	110
Step 10: Run the analysis	114
Step 11: Post Processing of the Results	114
Summary	116
Non-Linear FE Analysis with Contact	117
Step 1: Purpose of the FE Analysis / Description of the problem	118
Step 2: Input values for the FE analysis	118
Step 3: Model Geometry	119
Step 4: Meshing Geometry	123
Step 5, 6, 7, 8 and 9: Modifying comm file created by Efficient	126
Step 10: Run the analysis	134
Step 11: Post Processing of the Results	139
Summary	142
Non-Linear FE Analysis with Contact and Non-Linear Material	143
Step 1: Purpose of the FE Analysis / Description of the problem	144
Step 2: Input values for the FE analysis	144
Step 3: Model Geometry	145
Step 4: Meshing Geometry	145
Step 5, 6, 7, 8 and 9: Modifying comm file created by Efficient	145
Step 10: Run the analysis	148
Step 11: Post Processing of the Results	149
Summary	150
FE Analysis of 3D Plate for Mode Shapes	151
Step 1: Purpose of the FE Analysis / Description of the problem	152
Step 2: Input values for the FE analysis	152
Step 3: Model Geometry	154
Step 4: Meshing Geometry	156
Step 5, 6, 7, 8 and 9: Creating command file by Wizard	159
Step 10: Run the analysis	162
Step 11: Post Processing of the Results	162
Summary	167
FE Analysis of 1D Beam for Mode Shapes	168

Step 1: Purpose of the FE Analysis / Description of the problem	169
Step 2: Input values for the FE analysis	169
Step 3: Model Geometry	171
Step 4: Meshing Geometry	173
Step 5, 6, 7, 8 and 9: Modifying comm file created by Wizard	175
Step 10: Run the analysis	178
Step 11: Post Processing of the Results	178
Summary	183
FE Analysis of 2D Plate for Mode Shapes	184
Step 1: Purpose of the FE Analysis / Description of the problem	185
Step 2: Input values for the FE analysis	185
Step 3: Model Geometry	186
Step 4: Meshing Geometry	188
Step 5, 6, 7, 8 and 9: Modifying comm file created by Wizard	190
Step 10: Run the analysis	192
Step 11: Post Processing of the Results	195
Summary	199
Thermal Conduction FE Analysis	200
Step 1: Purpose of the FE Analysis / Description of the problem	201
Step 2: Input values for the FE analysis	201
Step 3: Model Geometry	202
Step 4: Meshing Geometry	203
Step 5, 6, 7, 8 and 9: Creating command file by Wizard	206
Step 10: Run the analysis	209
Step 11: Post Processing of the Results	210
Summary	213
Thermal Convection FE Analysis	214
Step 1: Purpose of the FE Analysis / Description of the problem	215
Step 2: Input values for the FE analysis	215
Step 3: Model Geometry	216
Step 4: Meshing Geometry	222
Step 5, 6, 7, 8 and 9: Creating command file by Wizard	225
Step 10: Run the analysis	227
Step 11: Post Processing of the Results	227
Summary	229
What will be covered in Volume 3	230
Using Python for Parametric Modelling and FE Analysis	230
Using Hommard for adaptive meshing	230
Advanced Thermal FE Analysis	230
Thermo-Mechanical FE Analysis	230
Pipe Stress Analysis	230

Fluid Structure Integration	231
Appendix A	232
Other sources of information	232
Appendix B	233
Installing Software required for this book	233
Ubuntu 12.04 Configuration	233
Salome-Meca 2013.2 installation	234
Efficient Install	236

Foreword

After the success I received by writing my first Book “Finite Element Analysis using Open Source Software”, I received many email communications congratulating me and telling me how easy it was to use my book. Readers had found an easy to use, easy to read and easy to follow documentation for Open Source Software that can be used for Finite Element analysis. Some said that they completed the entire book with the exercise within one single weekend and I doubt if they had taken any sleep in between.

The users of my book were fresh graduates from the university who knew the fundamentals and had been using proprietor software in their university and now as they were out of university, they wanted to use something that doesn't hurt their hip pocket. Others were seasoned professionals who knew other proprietary software but wanted to know how to perform FEA using Open Source Software.

This book starts with updated examples in version 11.x for Code_Aster from previous book. It then goes on and adds on advanced analysis.

This book is written for

Those who have a passion for learning Open Source software, particularly CAD and FEA software. This book is written for those who are new to software like Salome and Code_Aster.

If you are having trouble understanding where to start with Salome and Code_Aster, this book is written for you. If you are having troubles understanding the computer translated Code_Aster User Documents (which are rich in information), this book is written for you. If you want easy reference to 75% of FEA problems that are encountered by engineers in day to day life and want to do that by Open Source Software, this book is written for you.

This book is for those who don't want to waste their time in finding tutorials online and trying to make logical and sequential sense. This book starts with a very basic introduction of what to do to perform FE Analysis, and then, with each new Chapter, it introduces new concepts in an easy to understand format. If you want to learn how to do FE Analysis with Open Source software in a week's time, than this book is for you.

This book is not written for

If you are advanced user of Salome and Code_Aster and after reading the Table of Content you can say to yourself that “the information covered in this book is something that I already know”, this book is not for you.

This book is also not written for someone who does not know what Finite Element Analysis is. FE Analysis, as a fundamental, should be known to the user of this book.

If you are a beginner, it is advised that you purchase our first book “Finite Element Analysis with Open Source Software” where you will be able to gain more insights into the fundamental analysis that can be done with Salome-Meca and Code_Aster.

Once you are familiar with those concepts, this book will be easy to follow. Some of the chapters in this book rely on the information given in the first book. Though it is not necessary, it is recommended to complete the first book before you start with this one.

What software would you need to follow through

Operating System used:

1. Ubuntu 12.04

Software used for this book are

1. Salome-Meca version 2013.2
2. Code Aster version 11.3
3. Efficient version 0.1.1

All of the above software (except latest version of Efficient) are available in CAELinux 2013 DVD so if you have installed it, don't worry about any more installations. If you want to use latest software, install the above versions or latest versions of these software on your computer. Please note that if you install a software that is of higher version than that mentioned above, the screenshots may differ, but the fundamental concepts remain the same.

What are the steps in Finite Element Analysis

This book does not teach you what Finite Element Analysis is. You are nearly ready if you are familiar with the general form of Hook's law which states that "For small deformations of the object, the amount of deformation / displacement (Δx) is directly proportional to the deforming force or load (F)". The constant of proportionality in the above equation is the stiffness (k) of the object.

Generally the stiffness of the object is known due to the fact that we would have its shape and material properties as a given. If not, Either we would be optimising its shape by finding stresses generated in the object due to applied loads (e.g. objective of the study can be "optimise web thickness of gussets or optimise thickness of a pressure vessel Nozzle saddle"), or we would be checking which material is most suitable for the given object (e.g. objective of the study can be "Can Aluminium alloy be used to reduce the weight of the object?"). So stiffness " k " would be fixed for the given analysis based on shape and material selected.

The next step is boundary conditions. Any given object has to be sufficiently supported in the real world and FEA will emulate these supports, either there is a fixed support (e.g. bolted or welded joint), sliding support (e.g. shaft in hub or pipe shoe on structural steel). By applying these boundary conditions of supports, we are providing / fixing values of displacement. Then there would be forces applied to the object, either by gravity (self-weight) or by pressure applied on a surface or force applied on the object. By adding these boundary conditions of loads, we are providing / fixing values of force / load.

After all of the above is given to Code Aster (FE analysis software of choice for this book), it tries to solve the equation which will be in matrix form with the given input values of displacement and forces and obtain displacement for the entire object. These displacements (Strains) are converted to Stresses (Stress = Strain multiplied by Young's Modulus of the material, for elastic case) and are displayed as a coloured model or exported as a table.

Figure on the next page shows general steps to perform FEA.



In the coming chapters of this book, we will be using above philosophy to carry out Finite Element Analysis. Interpretation of results will be left to the reader.

Study Cases

After Esha learnt how to perform FEA using Open Source Software like Salome-Meca and Code_Aster, she started her professional career in the same firm that she did her internship. She was happy that John was her mentor all the way long. Esha started gaining more experience with Linear Static Finite Element Analysis, and with more experience she needed less reliance on John's guidance. Esha started doing her FE Analysis with more confidence.

After several months, John caught up with Esha to find out how she was feeling regarding the use of Open Source Software for Finite Element Analysis. Esha was very excited to tell John all about her experiences while they met for coffee. Then Esha told John that every now and then a different type of FE Analysis comes to her for which she is not ready yet. She has to pass them to her other colleagues as she is not feeling confident and she is feeling a bit lost. As usual John was listening to her words carefully and asked her if he could help. Esha was waiting for John to say that and she listed the analysis which she felt were a bit tough for her to do.

Parametric Modelling in Salome for Geometry and Mesh generation

Esha said that John had taught her how to “Dump the study” so that if she wants to recreate the geometry and mesh, it becomes easy for her. But what if she wants to change some of the parameters. What if Esha wants to generate geometry of the Pressure Vessel and Nozzle junction with different PV Diameter and Nozzle Diameter? What if she wants to change the mesh density in the PV Shell thickness or in the Nozzle thickness?

Combining element types in a single FE Analysis

Esha said that sometimes the models are too big and it would really help her if she could combine 3D elements with shell elements and Beam elements. This would make FE Analysis run faster without compromising her results. Was there a way to do this in Salome and Code_Aster?

Non Linear Material Analysis

Esha said that what John taught her in last book in Chapter 9 for checking against Allowable stress of the material was sufficient at that time, but she wanted to know if she could put the Graph of the Material Properties in Salome or Code-Aster and if the Analysis could take care of checking when the Allowable stress has been reached and stop the analysis?

Contact FE analysis

Esha said that she was happy to perform Assembly FE Analysis, but sometimes there is a requirement where she needs a gap between two parts and that load can only be translated once there is contact between surfaces. Was there a way to do it in Code_Aster?

Modal Analysis

Sometimes in her career, Esha had come across a FE problem where she needed to find the Natural frequencies of a given shape of the object. Esha asked John, if it was possible in Code Aster to find out the Natural frequencies of

the object by performing Modal Analysis?

Thermal Analysis

Esha remembered that once she was asked if there was a way to perform Thermal analysis in Code_Aster. As Esha didn't know, she had to again pass it along to her colleague. Is there a way to conduct Thermal Analysis in Code_Aster?

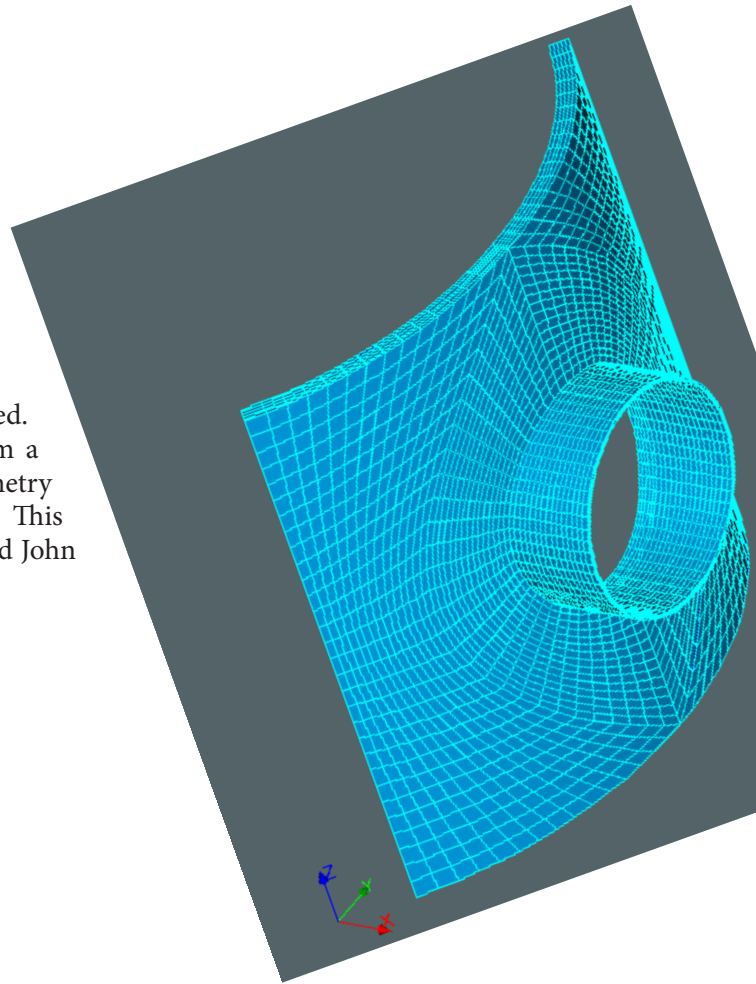
These were some of the example problems that she had the opportunity to do but was not able to do them due to her limited knowledge. There were some more problems that she wanted to discuss with John but first she wanted to know if her existing problems can be solved.

John told Esha that both Salome and Code_Aster were capable of conducting the analysis she asked for and much more.

Chapter 1 - Case12

Parametric Modelling of a Pressure Vessel Geometry

John was glad that Esha asked for help when she needed. John said that first he will show Esha, how to perform a Parametric Modelling of the same Pressure Vessel Geometry that they created in Chapter 9 / Case 9 of previous book. This time Esha was the client as she had a specific request and John was the Engineer.



Step 1: Description of the problem

John asked Esha, what the exact outcome that Esha was looking for was. Esha said that in previous book in Chapter 9 John had showed her how to save a Python file from which she could generate the geometry again and again. But editing that file was time consuming and sometimes she made a mistake in entering the data and was confused in looking through series of lines of Python in the “.py” file. She wanted to know if there was an easy way in which she could just enter the values of the Parameters and Salome could do the modelling and meshing by itself.

John said that Salome indeed was very powerful and the request that Esha made was easily achievable in Salome. John told Esha that there are two methods in which the request that Esha has made can be accomplished. One method is to edit the “.py” file manually, enter additional parameters and use them in the file to generate geometry and mesh it. The second method was to use something called “Notebook” in Salome that stores parameters and which can be edited. John will show Esha both of this methods.

Step 2: Input values for the FE analysis

Dimensions of the Pressure Vessel are similar to the previous book Chapter 9.

ID of Cylindrical shell of PV: 2000mm (2m)

Thickness of Cylindrical shell of PV: 10mm (0.01m)

Half Length of Cylindrical shell of PV: 1500mm (1.5m)

Dimensions of the Nozzle are

ID of the Nozzle: 300mm (0.3m)

Thickness of the Nozzle: 10mm (0.01m)

Projection of Nozzle: 300mm (0.3m)

Step 3: Parametric Modelling by direct editing of .py file

John copied Python file “Case9.py” from Case9 folder and saved it as “Case12.py” in a separate folder named Case12.

Then John double clicked the file to open it. He told Esha that each and every line in this file is either a code that tells Salome about the geometry and mesh or is a comment. Then he scrolled down the file till he could see the following code.

```
# -*- coding: iso-8859-1 -*-

###

### This file is generated automatically by SALOME v6.3.0 with dump python functionality

###
```

Just below the above statements we will write our code for making this file do parametric modelling. This code can be written anywhere in the file, but as these lines of code are for initialising variables that store information they will be used in modelling they should be initialised before the values are used.

Python code is really easy and if you have done programming with any other language, you will feel Python to be a fun and easy language.

Things to remember in Python language:

1. You don't need to end the statement with a semi-colon “;”
2. Python ends the statement when “Enter” (Carriage Return / Line Feed) is encountered
3. If you are initialising a variable, Variable Name will be on the left side of the = sign and value will be on the right side
4. Mathematical operations have regular meaning and precedence
5. Everything after “#” will be considered as a comment

With this in mind let's add variables that will hold information for Geometry and Mesh.

Add the following lines:

```
#####
#Parameters for Geometry#
#####
#Pressure Vessel Outside Radius
PV_OR = 1010
#Pressure Vessel Inside Radius
PV_IR = 1000
#Pressure Vessel Height
PV_Height = 1500
#Nozzle Outside Radius
Nozz_OR = 160
#Nozzle Inside Radius
Nozz_IR = 150
#Nozzle Protrusion Length
Nozz_Protrusion = 200
#Height of Cylinder for Nozzle Construction
Nozz_Len = PV_OR + Nozz_Protrusion
```

```
#####

#Parameters for Mesh#

#####

#Number of Segments for Global Meshing

GlobalSeg = 15

#Number of Segments for Pressure Vessel Shell Thickness Meshing

PVSeg = 5

#Number of Segments for Nozzle Shell Thickness Meshing

NozSeg = 5

#####

#End of Parameters#

#####
```

After adding the parameters that will be used in the creation of geometry and mesh, lets add these in the python file.

Find the lines shown below and enter the parameters defined earlier in their respective locations. Parameters entered manually are shown with Red colour.

```
Cyl_PV_OD = geompy.MakeCylinderRH(PV_OR, PV_Height)

Cyl_PV_ID = geompy.MakeCylinderRH(PV_IR, PV_Height)

Cyl_Noz_OD = geompy.MakeCylinderRH(Nozz_OR, Nozz_Len)

Cyl_Noz_ID = geompy.MakeCylinderRH(Nozz_IR, Nozz_Len)

Vx = geompy.MakeVectorDXDYDZ(1, 0, 0)

Vy = geompy.MakeVectorDXDYDZ(0, 1, 0)

Vz = geompy.MakeVectorDXDYDZ(0, 0, 1)

geompy.Rotate(Cyl_Noz_OD, Vy, 90*math.pi/180.0)

geompy.Rotate(Cyl_Noz_ID, Vy, 90*math.pi/180.0)

geompy.TranslateDXDYDZ(Cyl_Noz_OD, 0, 0, PV_Height / 2)

geompy.TranslateDXDYDZ(Cyl_Noz_ID, 0, 0, PV_Height / 2)

Geom_OD = geompy.MakeFuse(Cyl_PV_OD, Cyl_Noz_OD)

Geom_ID = geompy.MakeFuse(Cyl_PV_ID, Cyl_Noz_ID)

PV_Whole = geompy.MakeCut(Geom_OD, Geom_ID)

#Please note that $ is entered here for clarity purpose

# to show that line continues below without any break
```

```

# Please do NOT enter $ sign in your code.

a3D_Sketcher_1 = geompy.Make3DSketcher([0, 0, 0, PV_OR, PV_OR, 0, $
PV_OR, PV_OR, PV_Height, 0, 0, PV_Height, 0, 0, 0])

Face_1 = geompy.MakeFaceWires([a3D_Sketcher_1], 1)

Revolution_1 = geompy.MakeRevolution(Face_1, Vz, 270*math.pi/180.0)

PV = geompy.MakeCut(PV_Whole, Revolution_1)

P0 = geompy.MakeVertex(0, 0, PV_Height / 2)

Plane_Vy = geompy.MakePlane(P0, Vy, PV_OR * 5)

Plane_Vz = geompy.MakePlane(P0, Vz, PV_OR * 5)

PV_1 = geompy.MakePartition([PV], [Plane_Vy, Plane_Vz], [], [], $
geompy.ShapeType["SOLID"], 0, [], 0)

P1 = geompy.MakeVertex(0, 0, 0)

P2 = geompy.MakeVertex(0, (PV_OR * math.sin(45*math.pi/180.0) + $
Nozz_OR) / 2, PV_Height / 2)

P3 = geompy.MakeVertex(0, 0, PV_Height)

```

By doing above changes we have made geometry creation parametric.

Now to make meshing parametric, find the lines shown below and then we need to make following changes

```

Global_Mesh_Seg = Regular_1D.NumberOfSegments(GlobalSeg)

Global_Mesh_Seg.SetDistrType( 0 )

Quadrangle_2D = PV_Mesh.Quadrangle()

Hexa_3D = smesh.CreateHypothesis('Hexa_3D')

status = PV_Mesh.AddHypothesis(Hexa_3D)

Regular_1D_1 = PV_Mesh.Segment(geom=PV_SubMsh)

PV_Thk_Seg = Regular_1D_1.NumberOfSegments(PVSeg)

PV_Thk_Seg.SetDistrType( 0 )

Propagation_of_1D_Hyp_on_opposite_edges_1 = $

Regular_1D_1.Propagation()

Regular_1D_2 = PV_Mesh.Segment(geom=Noz_SubMsh)

Noz_Thk_Seg = Regular_1D_2.NumberOfSegments(NozSeg)

```

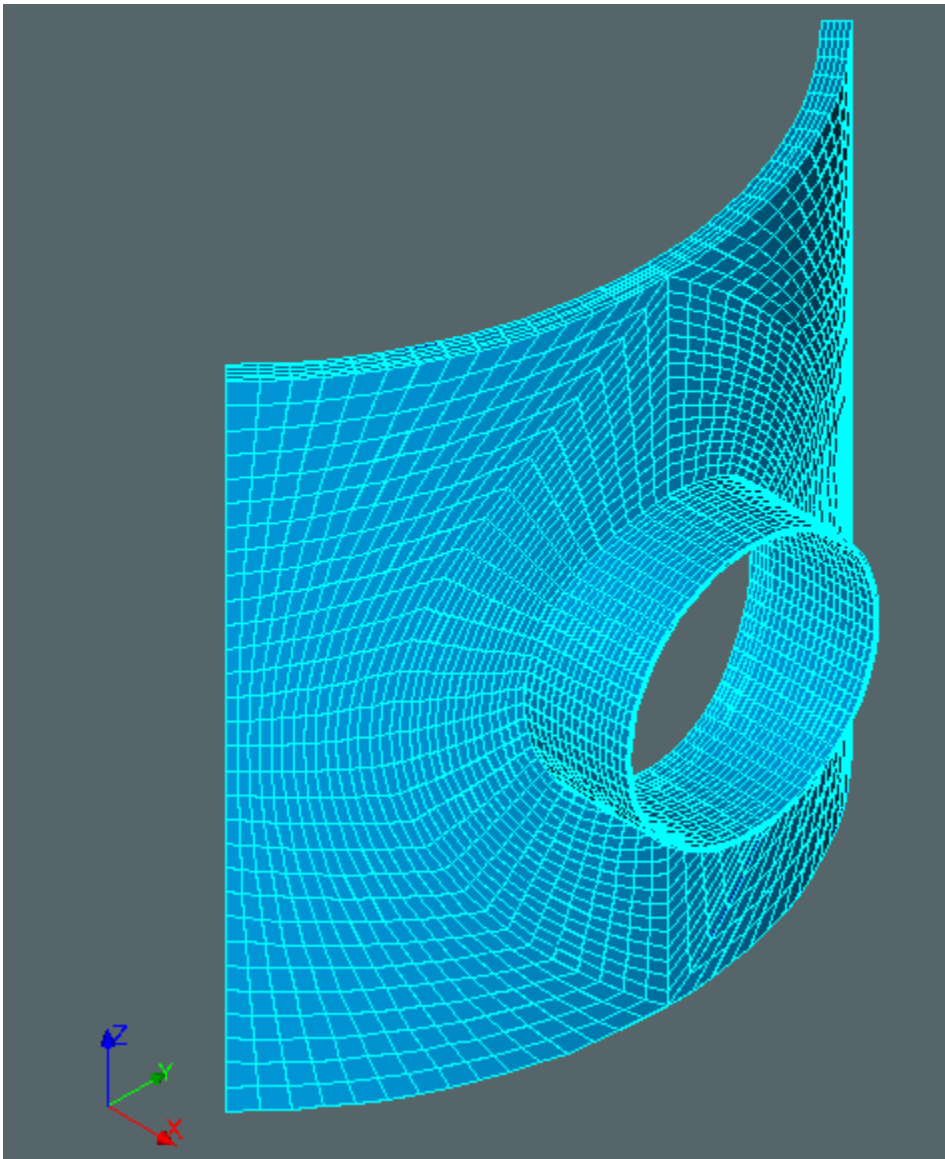
By doing above changes we have made the meshing as parametric too.

Now Start Salome Meca and Click File -> Load Script, then select the python file we saved earlier, after some time mesh of the Pressure vessel with the dimensions you entered in parameters will be displayed.

Just to check if the python file and parameters are working properly, John entered following parameters (Changes marked in Red)

```
#####  
#Parameters for Geometry#  
#####  
#Pressure Vessel Outside Radius  
PV_OR = 1050  
#Pressure Vessel Inside Radius  
PV_IR = 1000  
#Pressure Vessel Height  
PV_Height = 1500  
#Nozzle Outside Radius  
Nozz_OR = 300  
#Nozzle Inside Radius  
Nozz_IR = 290  
#Nozzle Protrusion Length  
Nozz_Protrusion = 200  
#Height of Cylinder for Nozzle Construction  
Nozz_Len = PV_OR + Nozz_Protrusion  
#####  
#Parameters for Mesh#  
#####  
#Number of Segments for Global Meshing  
GlobalSeg = 15  
#Number of Segments for Pressure Vessel Shell Thickness Meshing  
PVSeg = 5  
#Number of Segments for Nozzle Shell Thickness Meshing  
NozSeg = 5  
#End of Parameters#  
#####
```

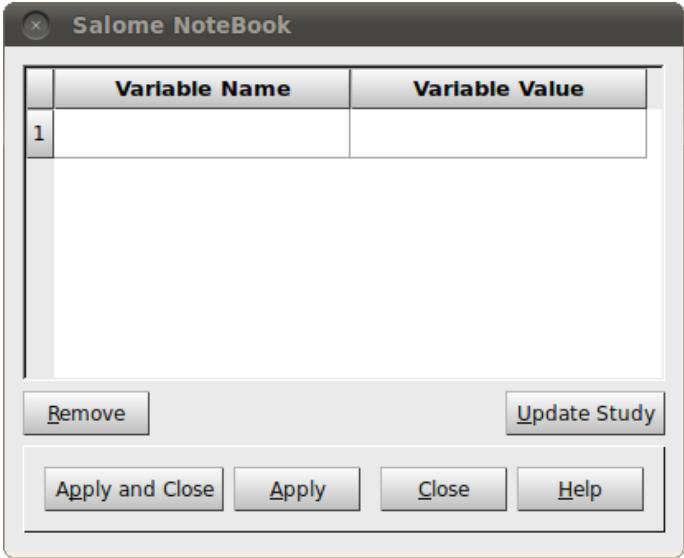
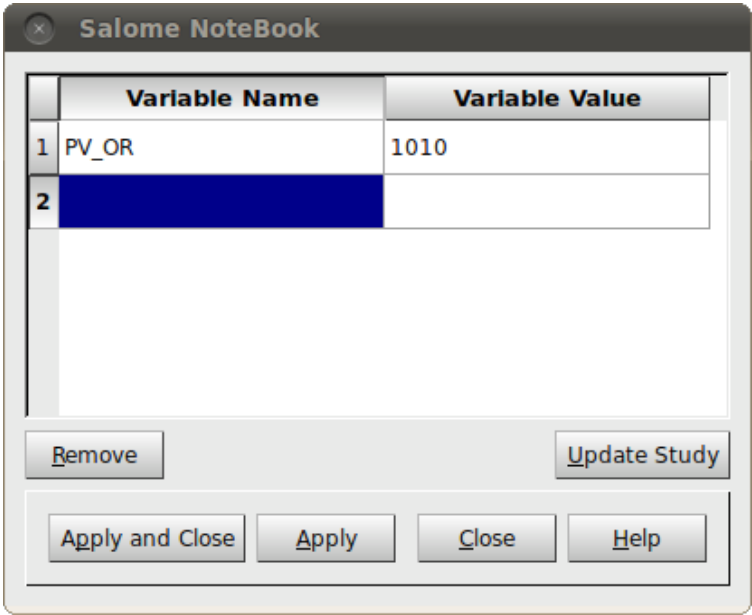

And he got following as a result

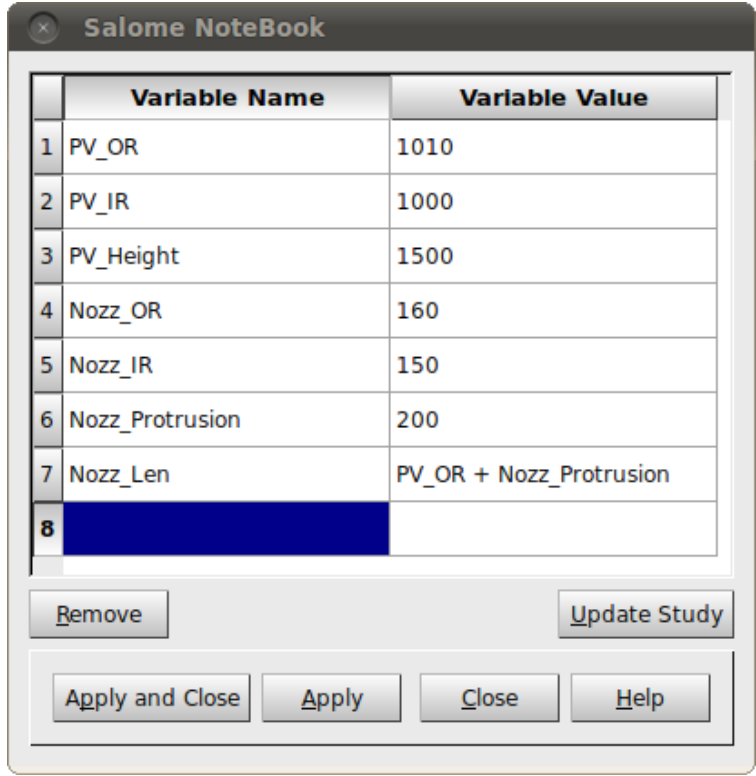


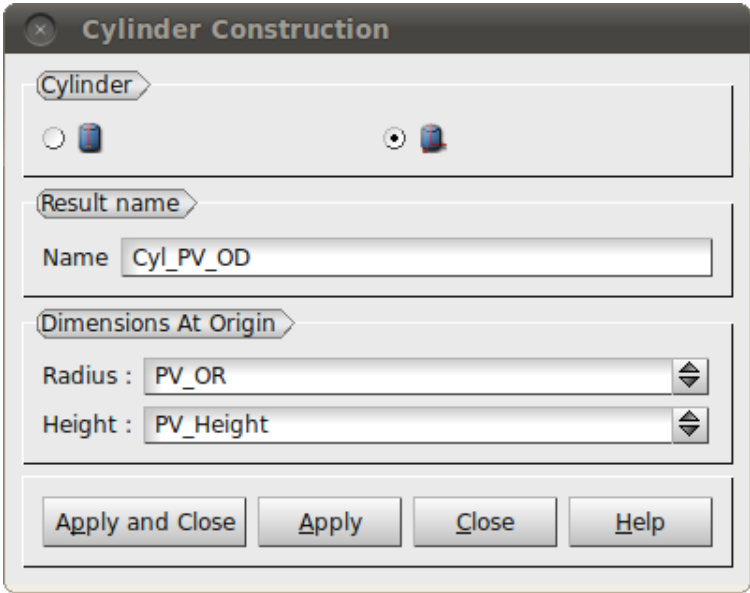
As can be seen, the thickness of Pressure vessel has increased and the Nozzle diameter has also increased.

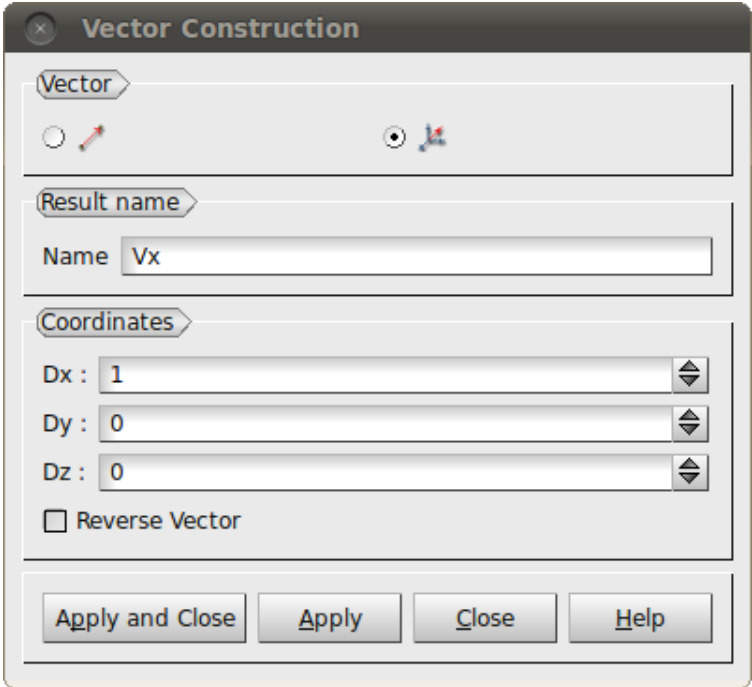
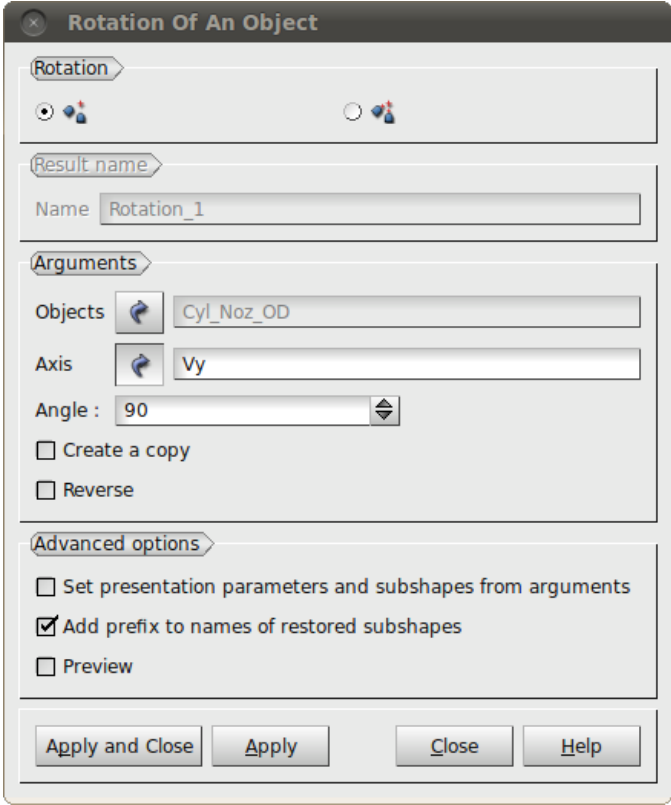
Step 3 again: Parametric Modelling by using Notebook

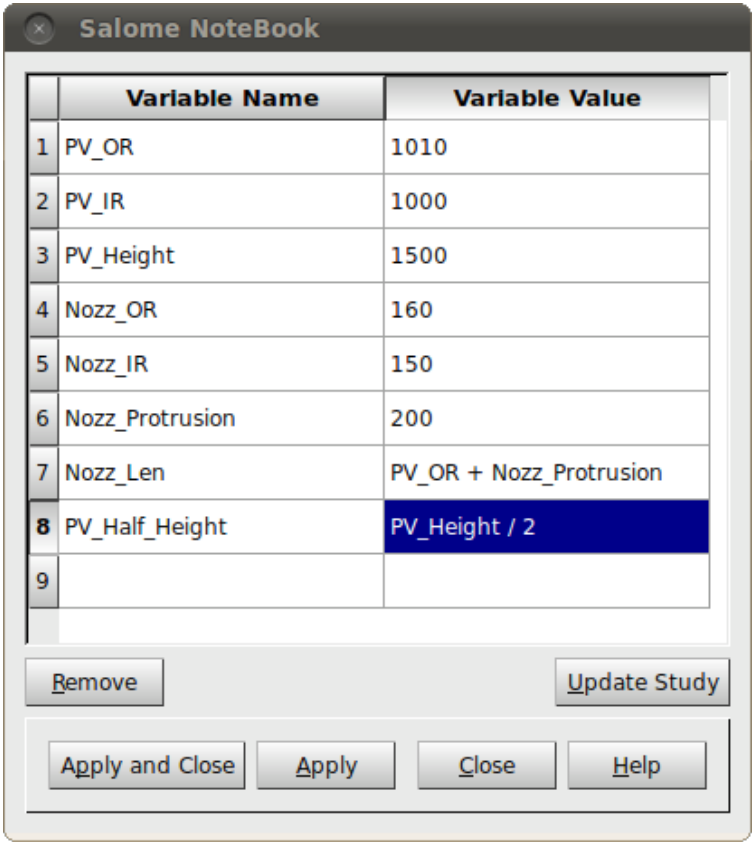
John told Esha that Salome provides a feature called Notebook which is very powerful and that is what we are going to use next.

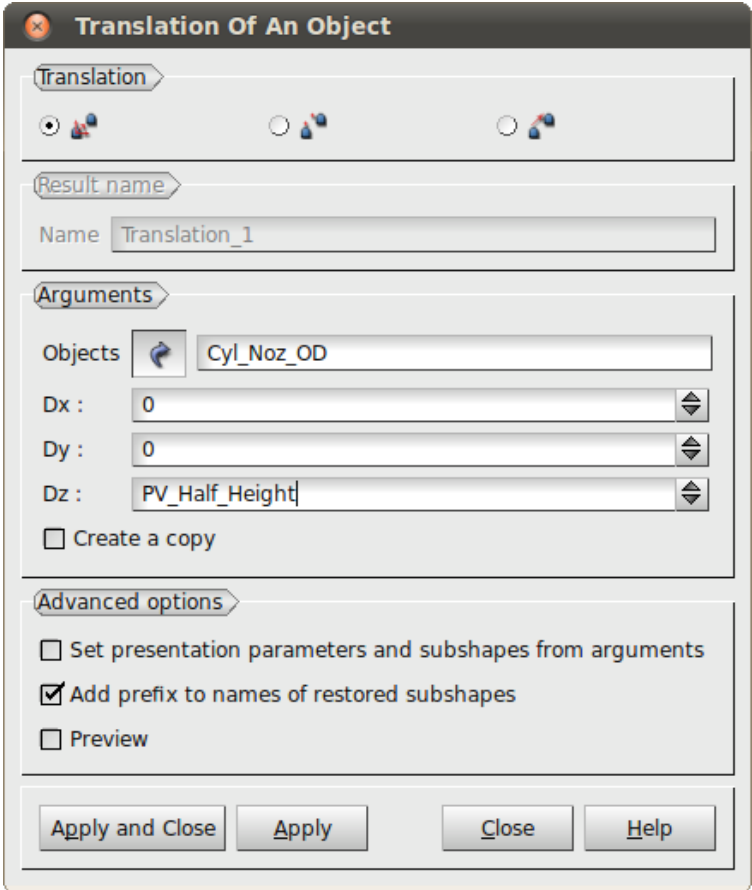
#	Description	Figure
1	Modelling steps shown here are exactly same as shown in previous book in Chapter 9, with the only difference of using Notebook to add parameters. First we will add Parameters to the Notebook and then use those while generating geometry and mesh. Open Salome-Meca and start Geometry Module.	
2	Click File -> Notebook and a window will popup as shown in adjacent figure. This is Salome Notebook. It is blank right now, but we will add Parameters ¹ soon.	
3	Click below Variable Name and the field will be ready to accept Variable Name. Enter “PV_OR” as the variable name, Press “Tab” and the focus will move to Variable Value, enter “1010” as the Variable Value. PV_OR is Outside Radius of Pressure Vessel with a value of 1010mm.	

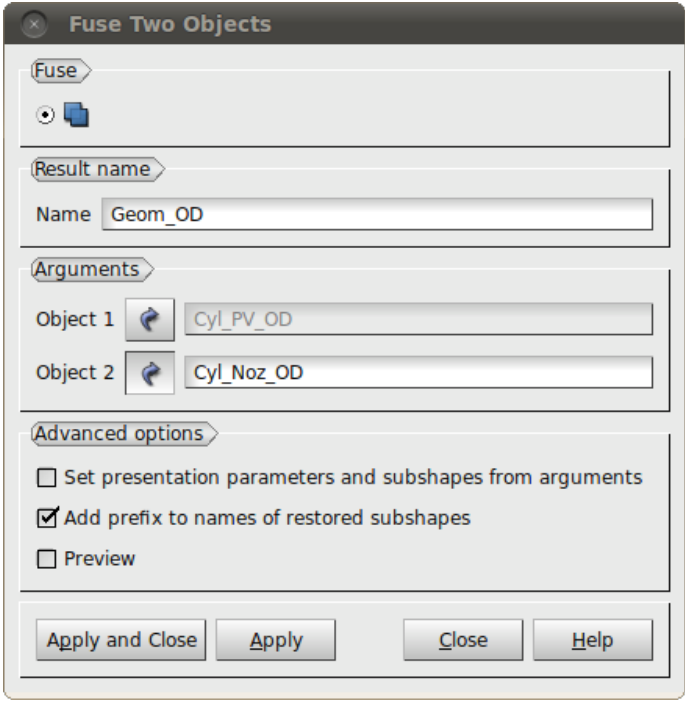
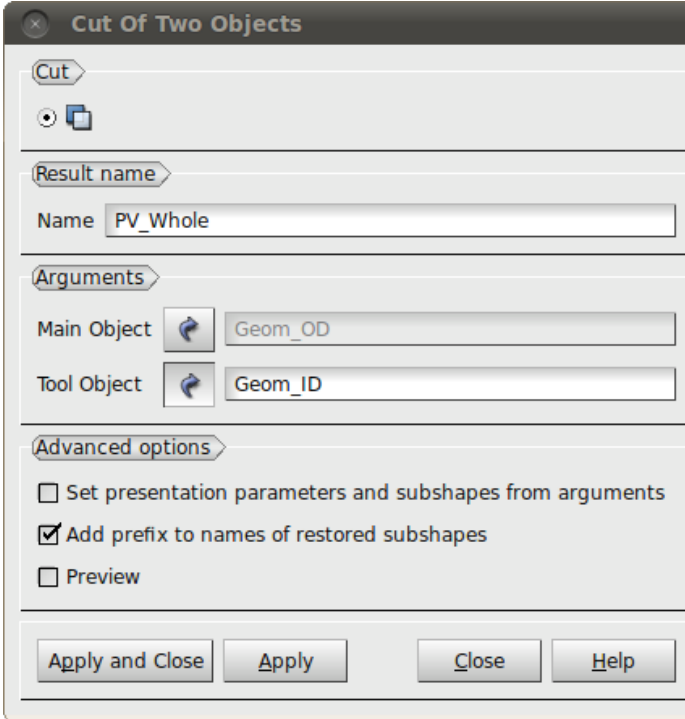
#	Description	Figure																											
4	<p>Enter other parameters as shown in adjacent figure.</p> <p>One thing to note here is the “Nozz_Len” which is given a Variable Value as addition of a two Variables.²</p>	 <p>The screenshot shows the Salome NoteBook dialog box with a table of variable definitions. The table has two columns: 'Variable Name' and 'Variable Value'. The variables are listed as follows:</p> <table border="1"> <thead> <tr> <th></th> <th>Variable Name</th> <th>Variable Value</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>PV_OR</td> <td>1010</td> </tr> <tr> <td>2</td> <td>PV_IR</td> <td>1000</td> </tr> <tr> <td>3</td> <td>PV_Height</td> <td>1500</td> </tr> <tr> <td>4</td> <td>Nozz_OR</td> <td>160</td> </tr> <tr> <td>5</td> <td>Nozz_IR</td> <td>150</td> </tr> <tr> <td>6</td> <td>Nozz_Protrusion</td> <td>200</td> </tr> <tr> <td>7</td> <td>Nozz_Len</td> <td>PV_OR + Nozz_Protrusion</td> </tr> <tr> <td>8</td> <td></td> <td></td> </tr> </tbody> </table> <p>Below the table are buttons for 'Remove', 'Update Study', 'Apply and Close', 'Apply', 'Close', and 'Help'.</p>		Variable Name	Variable Value	1	PV_OR	1010	2	PV_IR	1000	3	PV_Height	1500	4	Nozz_OR	160	5	Nozz_IR	150	6	Nozz_Protrusion	200	7	Nozz_Len	PV_OR + Nozz_Protrusion	8		
	Variable Name	Variable Value																											
1	PV_OR	1010																											
2	PV_IR	1000																											
3	PV_Height	1500																											
4	Nozz_OR	160																											
5	Nozz_IR	150																											
6	Nozz_Protrusion	200																											
7	Nozz_Len	PV_OR + Nozz_Protrusion																											
8																													
5	<p>We are done adding Geometric Parameters. Click “Apply and Close”</p> <p>Next we will start generating our geometry.</p>																												

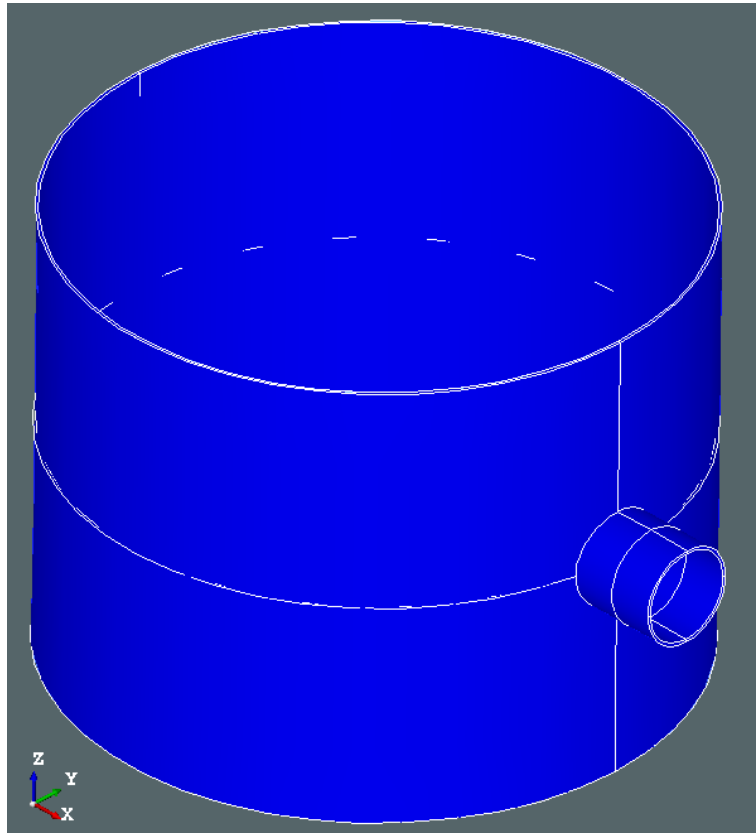
#	Description	Figure
6	<p>Click New Entity -> Primitives -> Cylinder.</p> <p>Select the second Option for cylinder, give it a Name “Cyl_ PV_OD” and enter “PV_OR” for Radius and “PV_Height” for Height.</p> <p>Click “Apply”</p> <p>For second Cylinder, use “Cyl_ PV_ID” as Name and enter “PV_IR” for Radius and “PV_ Height” for Height.</p> <p>Click “Apply”</p> <p>Use “Cyl_Noz_OD” as Name and enter “Nozz_OR” for Radius and “Nozz_Len” for Height.</p> <p>Click “Apply”</p> <p>Use “Cyl_Noz_ID” as Name and enter “Nozz_IR” for Radius and “Nozz_Len” for Height.</p> <p>Click “Apply and Close”</p>	
7	<p>As can be seen from the Cylinders that have been created, we need to Translate and Rotate the cylinders created for Nozzle and place them in their proper location and orientation.</p> <p>For that we need to create Vectors in all three directions. Then use Vector in Y direction to rotate the cylinders for nozzle and then translate them.</p>	

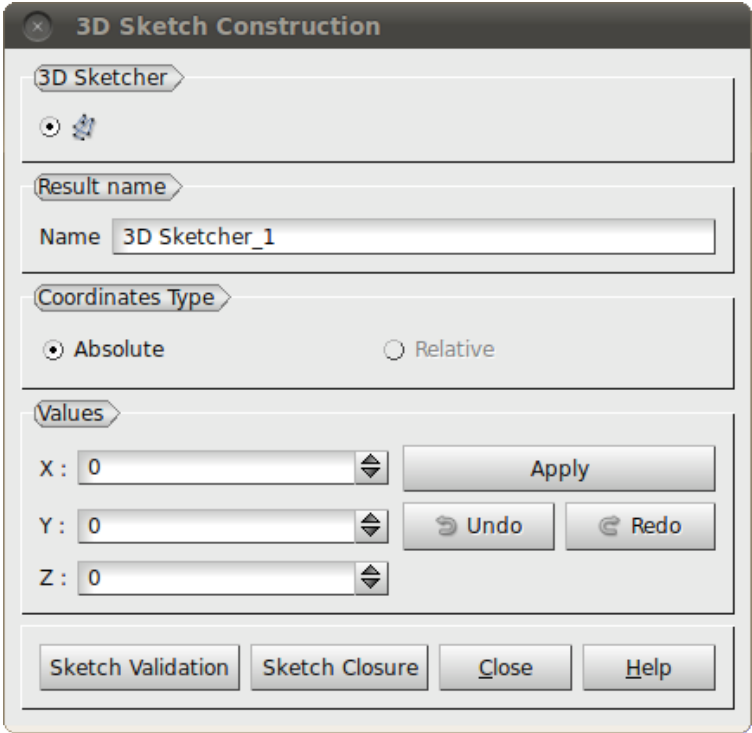
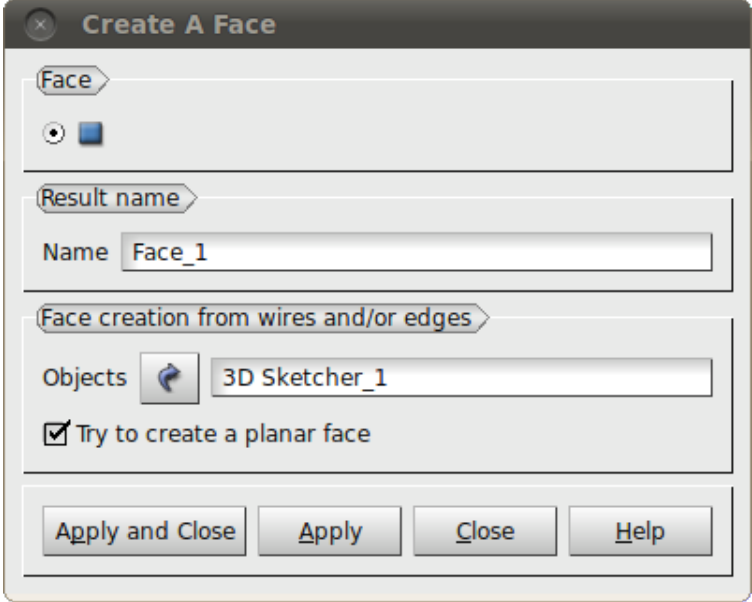
#	Description	Figure
8	<p>Click New Entity -> Basic -> Vector.</p> <p>Select the Second Option for Vector, Give it a name "Vx" (Vector in X direction) and enter 1, 0, 0 for X, Y and Z.</p> <p>Click "Apply"</p> <p>Next give the name "Vy" and enter 0, 1, 0 for X, Y and Z.</p> <p>Click "Apply"</p> <p>Next give the name "Vz" and enter 0, 0, 1 for X, Y and Z.</p> <p>Click "Apply and Close"</p>	
9	<p>Click Operations -> Transformation -> Rotation.</p> <p>Select "Cyl_NoZ_OD" as Objects, "Vy" as Axis, enter "90" as Angle and Untick "Create a copy" as we do not want to create a new cylinder.</p> <p>Click "Apply"</p> <p>In the similar manner, Rotate "Cyl_NoZ_ID" as shown above as well.</p> <p>Click "Apply and Close"</p>	

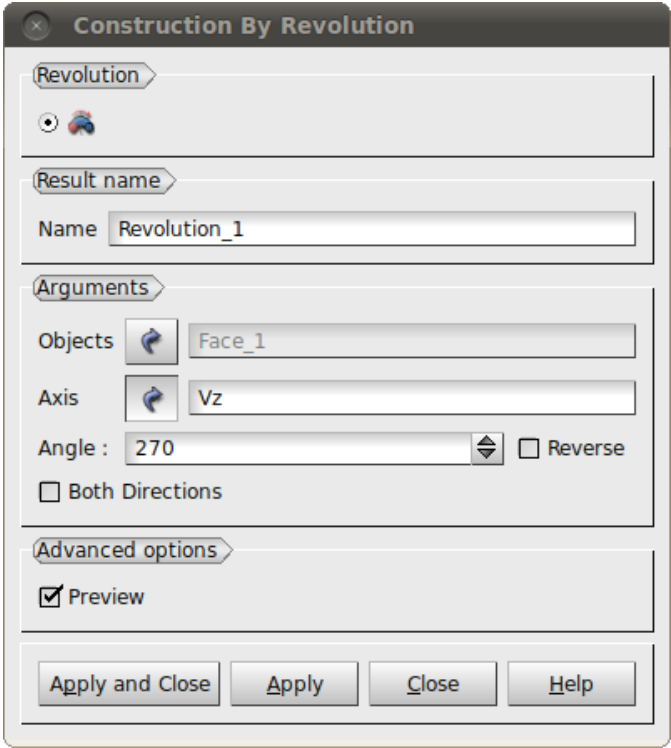
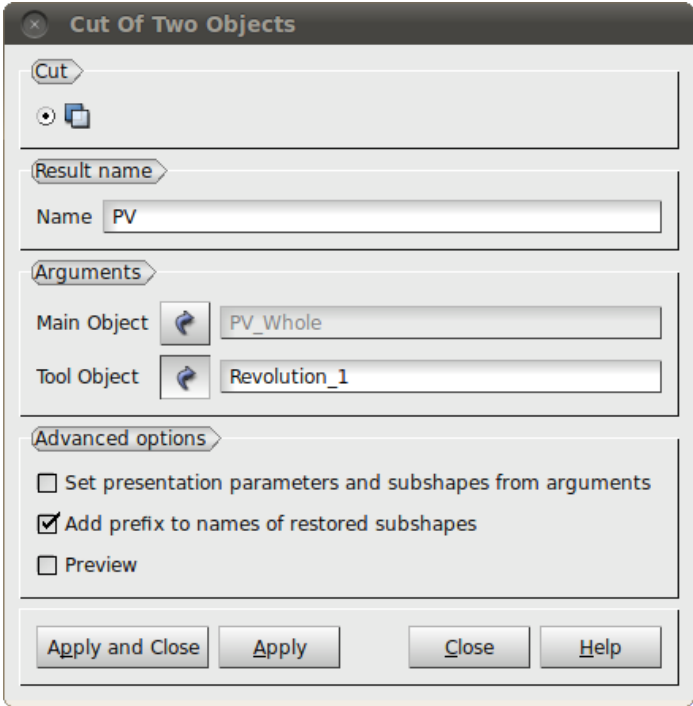
#	Description	Figure																														
10	Next to transform the Cylinders we need Half height of the Pressure Vessel, As we did not create it earlier, we need to create it now. Note that Salome does not allow Arithmetic operations in the field where you need to enter Values. You can only perform arithmetic operations in Notebook.																															
11	<p>Click File -> Notebook and add “PV_Half_Height” as a new Variable as shown in adjacent figure.</p> <p>Click “Apply and Close”.</p> <p>As you can see, if you forgot to add a Variable at the start of the geometry creation, you can do that at a later stage.</p>	 <p>The screenshot shows the Salome NoteBook interface. It contains a table with two columns: 'Variable Name' and 'Variable Value'. The table lists several variables, with 'PV_Half_Height' highlighted in blue. Below the table are buttons for 'Remove', 'Update Study', 'Apply and Close', 'Apply', 'Close', and 'Help'.</p> <table><thead><tr><th></th><th>Variable Name</th><th>Variable Value</th></tr></thead><tbody><tr><td>1</td><td>PV_OR</td><td>1010</td></tr><tr><td>2</td><td>PV_IR</td><td>1000</td></tr><tr><td>3</td><td>PV_Height</td><td>1500</td></tr><tr><td>4</td><td>Nozz_OR</td><td>160</td></tr><tr><td>5</td><td>Nozz_IR</td><td>150</td></tr><tr><td>6</td><td>Nozz_Protrusion</td><td>200</td></tr><tr><td>7</td><td>Nozz_Len</td><td>PV_OR + Nozz_Protrusion</td></tr><tr><td>8</td><td>PV_Half_Height</td><td>PV_Height / 2</td></tr><tr><td>9</td><td></td><td></td></tr></tbody></table>		Variable Name	Variable Value	1	PV_OR	1010	2	PV_IR	1000	3	PV_Height	1500	4	Nozz_OR	160	5	Nozz_IR	150	6	Nozz_Protrusion	200	7	Nozz_Len	PV_OR + Nozz_Protrusion	8	PV_Half_Height	PV_Height / 2	9		
	Variable Name	Variable Value																														
1	PV_OR	1010																														
2	PV_IR	1000																														
3	PV_Height	1500																														
4	Nozz_OR	160																														
5	Nozz_IR	150																														
6	Nozz_Protrusion	200																														
7	Nozz_Len	PV_OR + Nozz_Protrusion																														
8	PV_Half_Height	PV_Height / 2																														
9																																

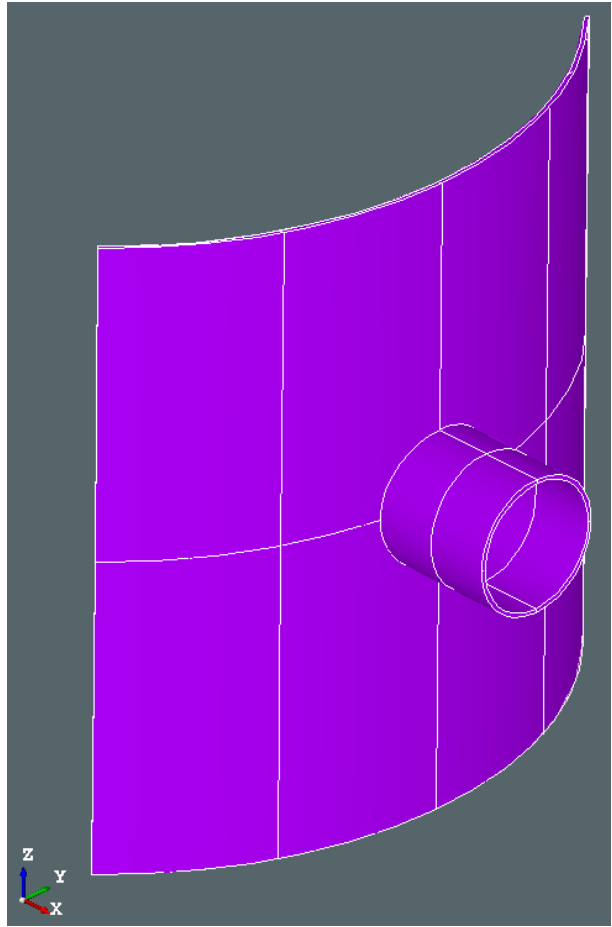
#	Description	Figure
12	<p>Click Operations -> Transformation -> Translation.</p> <p>Select “Cyl_NoZ_OD” as Objects, and enter “PV_Half_Height” for Dz, Untick “Create a copy” as we do not want a new copy of translated cylinder.</p> <p>Click “Apply”.</p> <p>Translate “Cyl_NoZ_ID” as shown above as well.</p> <p>Click “Apply and Close”.</p>	
13	<p>Now we need to fuse the Pressure Vessel and Nozzle OD Cylinders to generate a Geometry for the OD and then we need to fuse ID Cylinders to generate a Geometry for ID.</p> <p>After that we need to Remove the ID geometry from OD geometry to generate a Geometry that will be our Pressure Vessel.</p>	

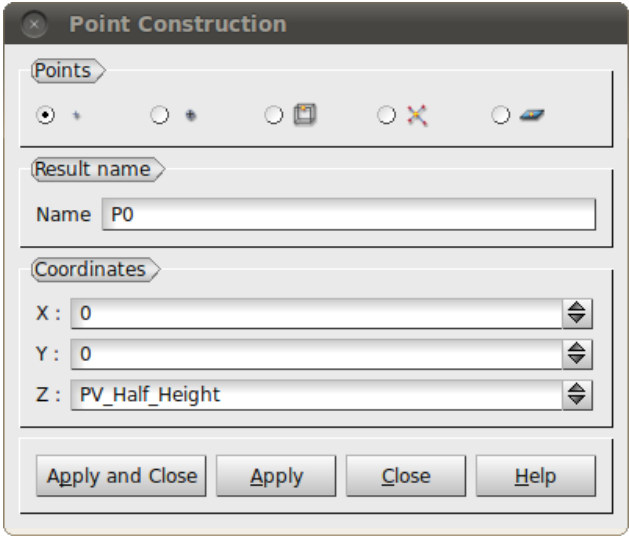
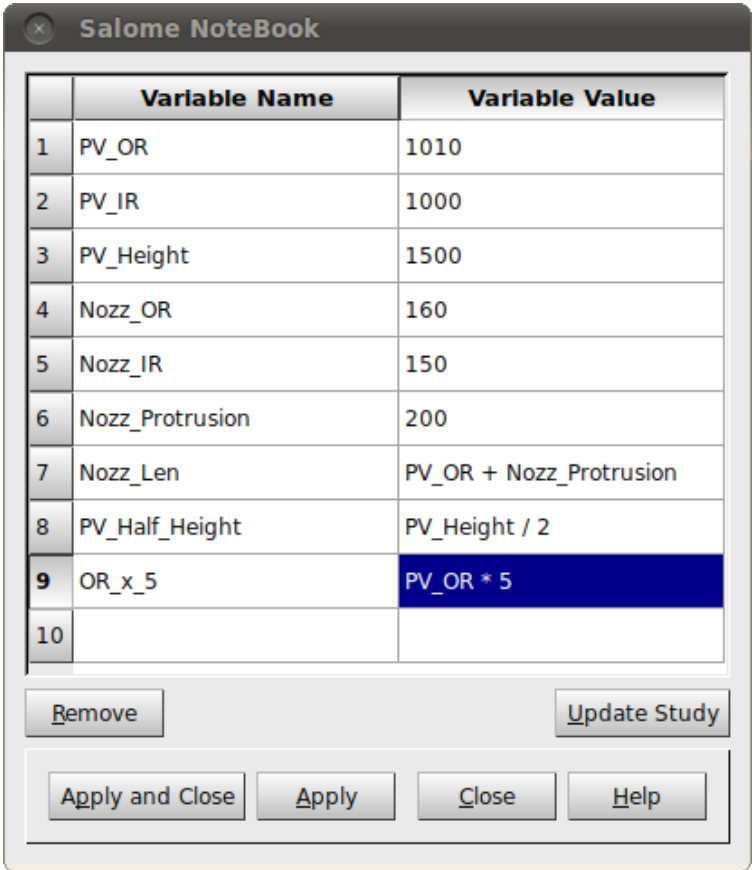
#	Description	Figure
14	<p>Click Operations -> Boolean -> Fuse.</p> <p>Give it a Name “Geom_OD” and select “Cyl_PV_OD” as Object 1 and “Cyl_Noz_OD” as Object 2.</p> <p>Click “Apply”</p> <p>Enter a new Name “Geom_ID” and select “Cyl_PV_ID” as Object 1 and “Cyl_Noz_ID” as Object 2.</p> <p>Click “Apply and Close”</p>	
15	<p>Click Operations -> Boolean -> Cut.</p> <p>Give it a Name “PV_Whole” and select “Geom_OD” as Main Object and “Geom_ID” as Tool Object.</p> <p>Click “Apply and Close”</p>	

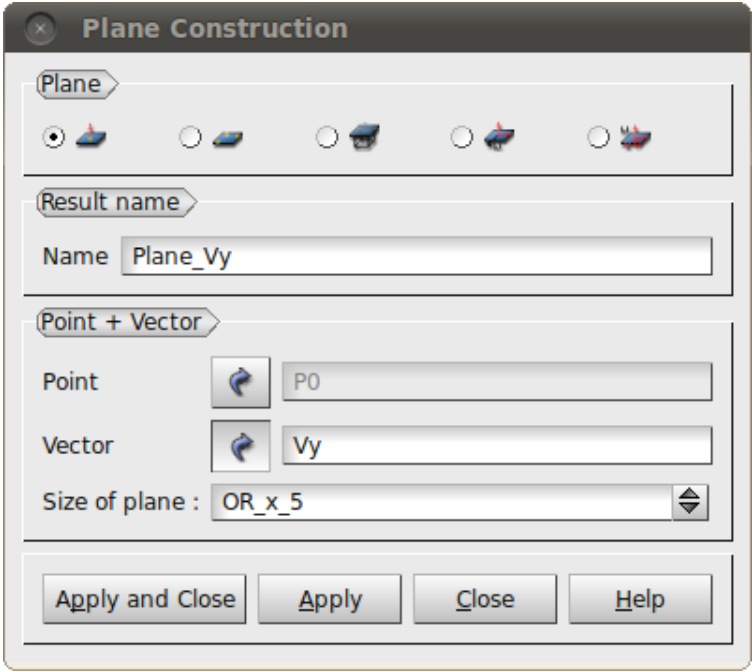
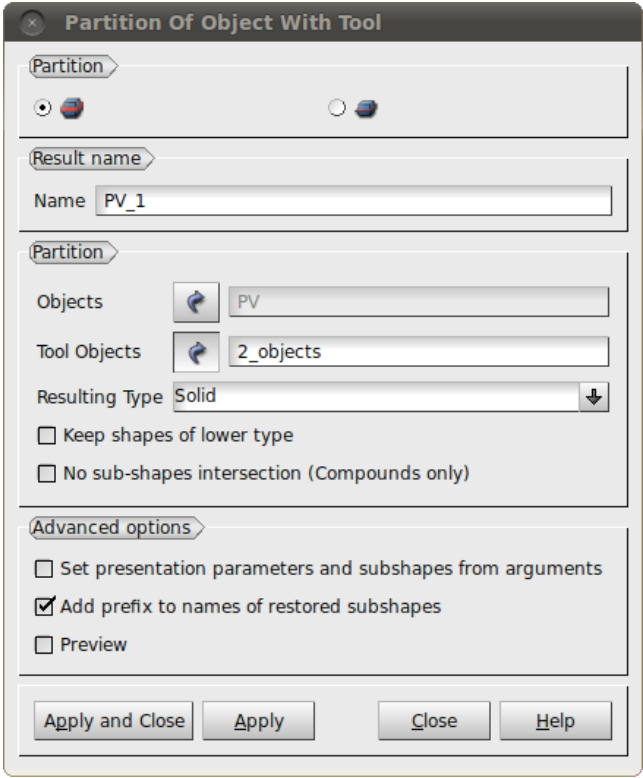
#	Description	Figure
16	<p>Right Click “PV_Whole” in Object Browser and Select Show Only. Change the mode to “Shading”.</p> <p>You should be able to see something similar to adjacent figure.</p> <p>Rotate it, zoom it and see the Pressure Vessel shell that you have created.</p> <p>Next we will remove excess geometry as we only need 90 degrees of this whole PV.</p>	

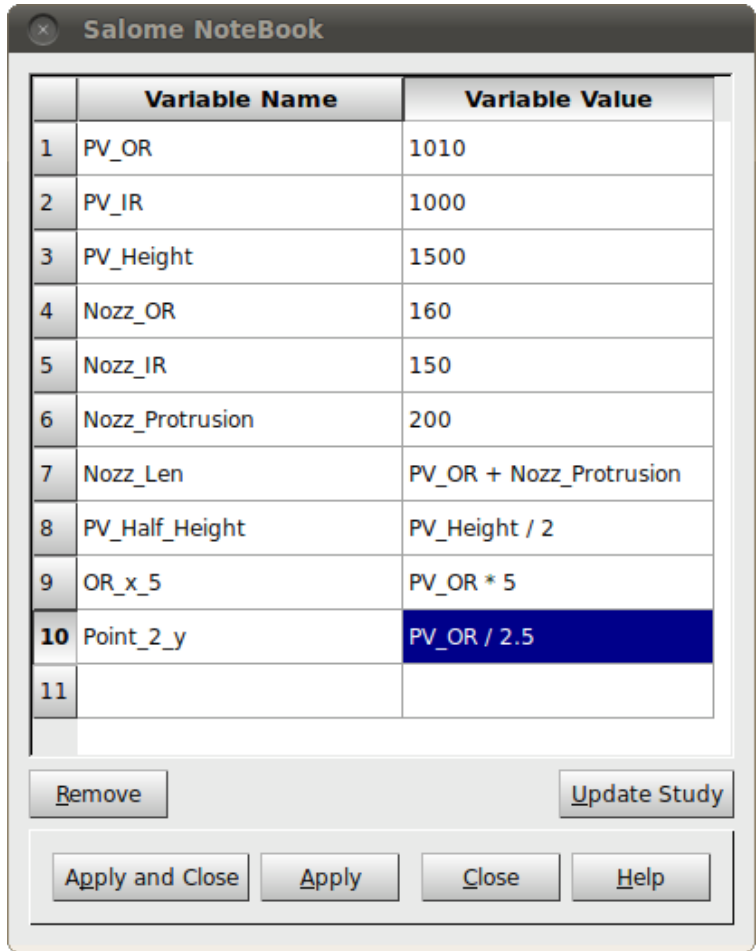
#	Description	Figure
17	<p>Click New Entity -> 3D Sketch.</p> <p>In the window that opens, keep default name and “Absolute” Coordinates Type.</p> <p>Enter following Numbers for X, Y and Z respectively:</p> <p>0, 0, 0 -> “Apply”</p> <p>PV_OR, PV_OR, 0 -> “Apply”</p> <p>PV_OR, PV_OR, PV_Height -> “Apply”</p> <p>0, 0, PV_Height -> “Apply”</p> <p>Next, Click “Sketch Closure”.</p> <p>This will create a frame for the face we will create in next step.</p>	
18	<p>Now we need to create a Face out of the 3D sketch we just created.</p> <p>Click New Entity -> Build -> Face.</p> <p>Keep default Name “Face_1”, select “3D Sketcher_1” as Objects and make sure that “Try to create a planar face” is ticked.</p> <p>Click “Apply and Close”</p>	

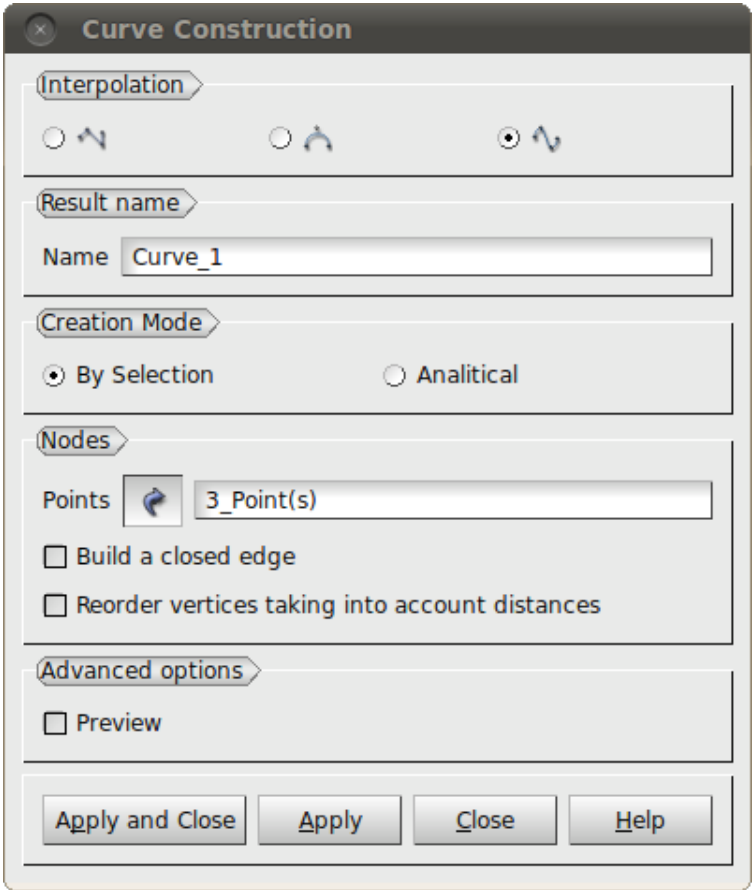
#	Description	Figure
19	<p>Now we will revolve this face to create the geometry.</p> <p>Click New Entity -> Generation -> Revolution.</p> <p>Keep default Name as "Revolution_1", Select "Face_1" for Objects and "Vz" for Axis. Enter "270" for Angle, Untick "Both Directions".</p> <p>Click "Apply and Close"</p>	
20	<p>Click Operations -> Boolean -> Cut.</p> <p>Give it a Name "PV" and select "PV_Whole" as Main Object and "Revolution_1" as Tool Object.</p> <p>Click "Apply and Close"</p>	

#	Description	Figure
21	<p>Right Click “PV_Whole” in Object Browser and Select Show Only. Change the mode to “Shading”.</p> <p>You should be able to see something similar to adjacent figure.</p>	
22	<p>We will need to partition this geometry such that each partitioned solid has only 6 edges.</p> <p>We need to create two planes that will cut the Geometry in 4 equal pieces. For this we will need to create two planes perpendicular to each other and that intersect at the centre of the nozzle.</p>	

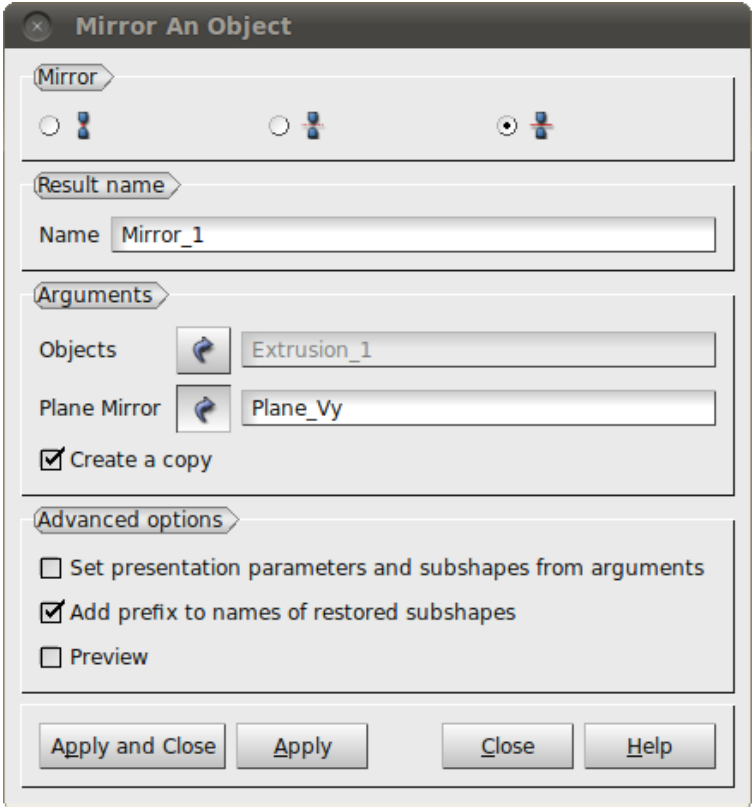
#	Description	Figure																																	
23	<p>First we will create a point from which the plane will pass.</p> <p>Click New Entity -> Basic -> Point.</p> <p>Click on First constructor of Points, give it a Name "P0" and enter co-ordinates as shown in the adjacent figure.</p> <p>Click "Apply and Close"</p>																																		
24	<p>Now we need to create two Planes with sizes equal to 5 times the Diameter of Pressure Vessel. So we will go back to Notebook and add another Parameter "OR_x_5" as shown in adjacent figure</p>	 <table border="1"> <thead> <tr> <th></th><th>Variable Name</th><th>Variable Value</th></tr> </thead> <tbody> <tr><td>1</td><td>PV_OR</td><td>1010</td></tr> <tr><td>2</td><td>PV_IR</td><td>1000</td></tr> <tr><td>3</td><td>PV_Height</td><td>1500</td></tr> <tr><td>4</td><td>Nozz_OR</td><td>160</td></tr> <tr><td>5</td><td>Nozz_IR</td><td>150</td></tr> <tr><td>6</td><td>Nozz_Protrusion</td><td>200</td></tr> <tr><td>7</td><td>Nozz_Len</td><td>PV_OR + Nozz_Protrusion</td></tr> <tr><td>8</td><td>PV_Half_Height</td><td>PV_Height / 2</td></tr> <tr><td>9</td><td>OR_x_5</td><td>PV_OR * 5</td></tr> <tr><td>10</td><td></td><td></td></tr> </tbody> </table>		Variable Name	Variable Value	1	PV_OR	1010	2	PV_IR	1000	3	PV_Height	1500	4	Nozz_OR	160	5	Nozz_IR	150	6	Nozz_Protrusion	200	7	Nozz_Len	PV_OR + Nozz_Protrusion	8	PV_Half_Height	PV_Height / 2	9	OR_x_5	PV_OR * 5	10		
	Variable Name	Variable Value																																	
1	PV_OR	1010																																	
2	PV_IR	1000																																	
3	PV_Height	1500																																	
4	Nozz_OR	160																																	
5	Nozz_IR	150																																	
6	Nozz_Protrusion	200																																	
7	Nozz_Len	PV_OR + Nozz_Protrusion																																	
8	PV_Half_Height	PV_Height / 2																																	
9	OR_x_5	PV_OR * 5																																	
10																																			

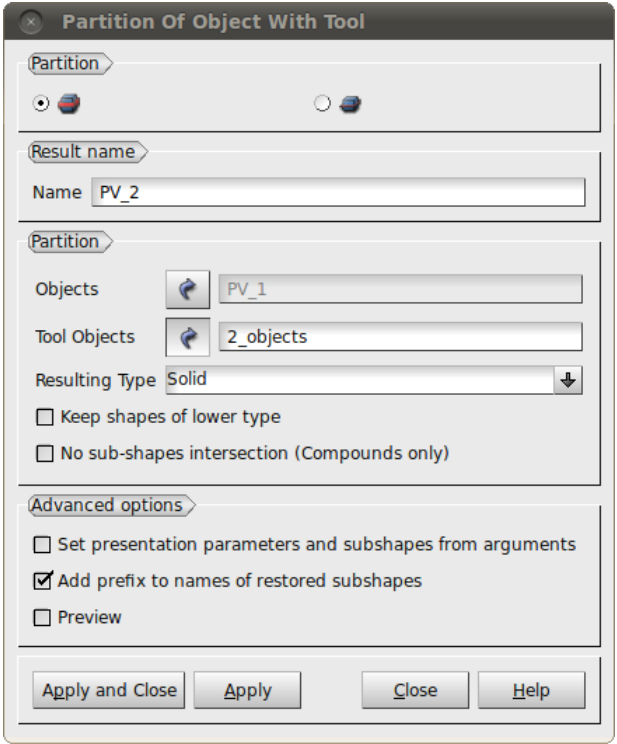
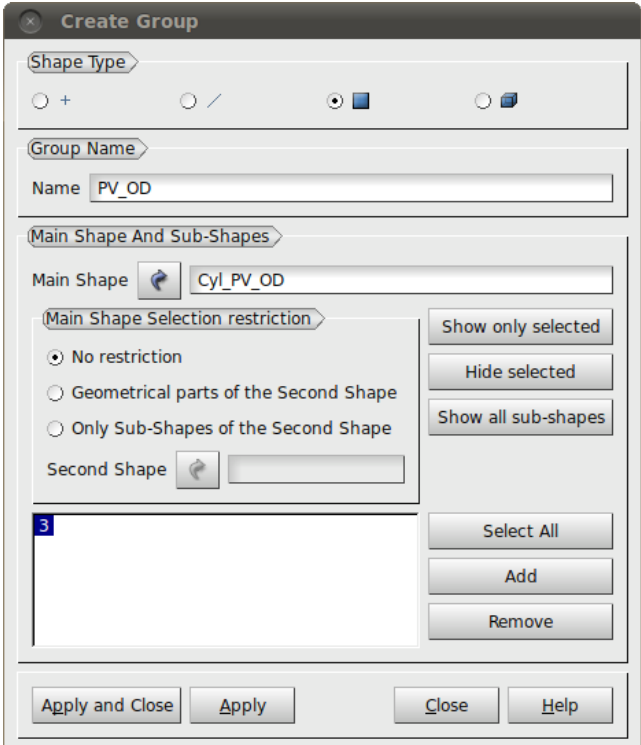
#	Description	Figure
25	<p>Click New Entity -> Basic -> Plane.</p> <p>In the window, Select first constructor for Plane, give it a Name "Plane_Vy", Select "P0" as Point and "Vy" as Vector. Enter "OR_x_5" as the "Size of plane".</p> <p>Click "Apply"</p> <p>For the second Plane, give it a Name "Plane_Vz", Select "P0" as Point and "Vz" as Vector. Enter "OR_x_5" as the "Size of plane".</p> <p>Click "Apply and Close"</p>	
26	<p>Now we partition the geometry.</p> <p>Click Operations -> Partition.</p> <p>Give it a Name "PV_1", select "PV" as Objects and for the Tool Objects select both Planes "Plane_Vy" and "Plane_Vz" in the Object Browser by Clicking on them while pressing "Ctrl".</p> <p>Make sure Resulting Type is "Solid".</p> <p>Click "Apply and Close"</p>	

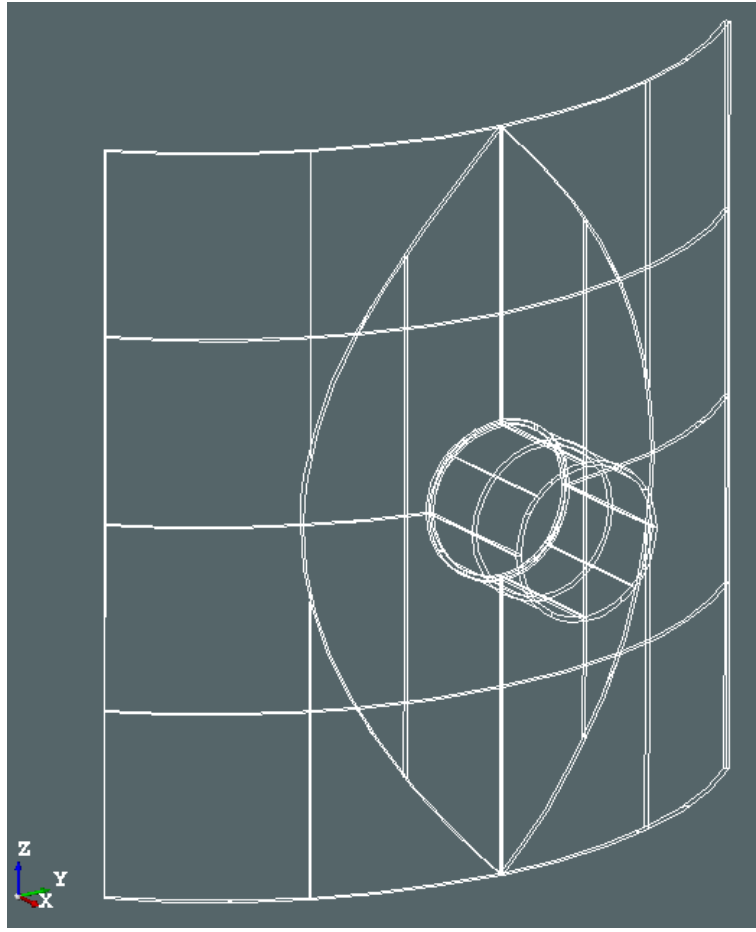
#	Description	Figure																																				
27	<p>We are not done yet, we still need to partition this geometry further such that each individual solids will have only 6 sides.</p> <p>Next we will create a curve edge that will aid in generating a curved surface to partition the geometry further.</p> <p>At this point in time, Notebook is not able to handle advanced python code like the one we added manually in “Case12.py”</p> $(PV_OR * \sin(45 * \pi / 180.0) + Nozz_OR) / 2$ <p>so we will just add a simple formula for Point P2 as $PV_OR / 2.5$</p>																																					
28	Add another variable named “Point_2_y” with the value as shown in adjacent figure	 <p>The screenshot shows the Salome Notebook interface. It contains a table with 11 rows and 2 columns: 'Variable Name' and 'Variable Value'. The 10th row, 'Point_2_y', is highlighted in blue and shows the formula 'PV_OR / 2.5'. Below the table are buttons for 'Remove', 'Update Study', 'Apply and Close', 'Apply', 'Close', and 'Help'.</p> <table border="1"> <thead> <tr> <th></th><th>Variable Name</th><th>Variable Value</th></tr> </thead> <tbody> <tr><td>1</td><td>PV_OR</td><td>1010</td></tr> <tr><td>2</td><td>PV_IR</td><td>1000</td></tr> <tr><td>3</td><td>PV_Height</td><td>1500</td></tr> <tr><td>4</td><td>Nozz_OR</td><td>160</td></tr> <tr><td>5</td><td>Nozz_IR</td><td>150</td></tr> <tr><td>6</td><td>Nozz_Protrusion</td><td>200</td></tr> <tr><td>7</td><td>Nozz_Len</td><td>PV_OR + Nozz_Protrusion</td></tr> <tr><td>8</td><td>PV_Half_Height</td><td>PV_Height / 2</td></tr> <tr><td>9</td><td>OR_x_5</td><td>PV_OR * 5</td></tr> <tr><td>10</td><td>Point_2_y</td><td>PV_OR / 2.5</td></tr> <tr><td>11</td><td></td><td></td></tr> </tbody> </table>		Variable Name	Variable Value	1	PV_OR	1010	2	PV_IR	1000	3	PV_Height	1500	4	Nozz_OR	160	5	Nozz_IR	150	6	Nozz_Protrusion	200	7	Nozz_Len	PV_OR + Nozz_Protrusion	8	PV_Half_Height	PV_Height / 2	9	OR_x_5	PV_OR * 5	10	Point_2_y	PV_OR / 2.5	11		
	Variable Name	Variable Value																																				
1	PV_OR	1010																																				
2	PV_IR	1000																																				
3	PV_Height	1500																																				
4	Nozz_OR	160																																				
5	Nozz_IR	150																																				
6	Nozz_Protrusion	200																																				
7	Nozz_Len	PV_OR + Nozz_Protrusion																																				
8	PV_Half_Height	PV_Height / 2																																				
9	OR_x_5	PV_OR * 5																																				
10	Point_2_y	PV_OR / 2.5																																				
11																																						

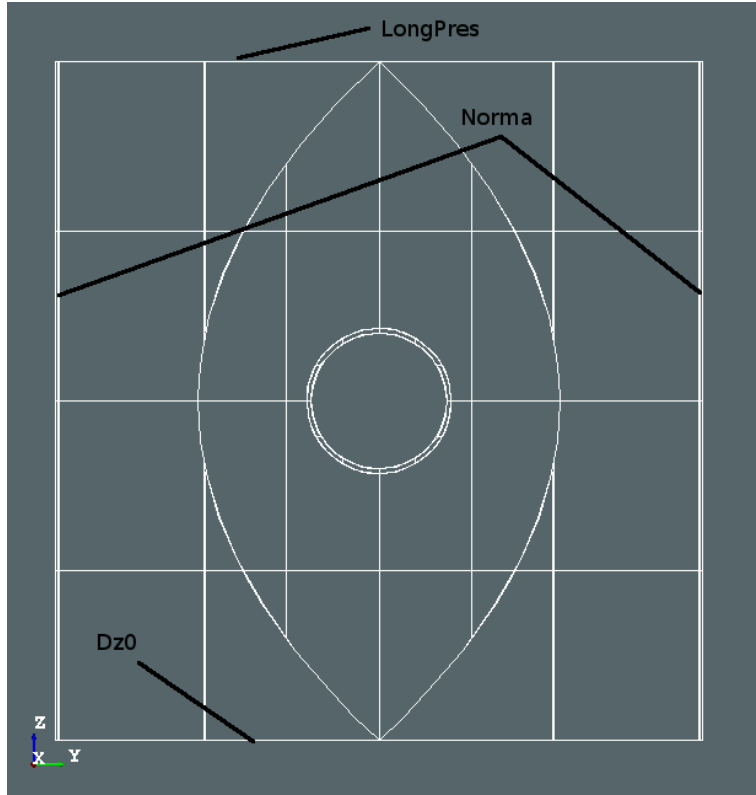
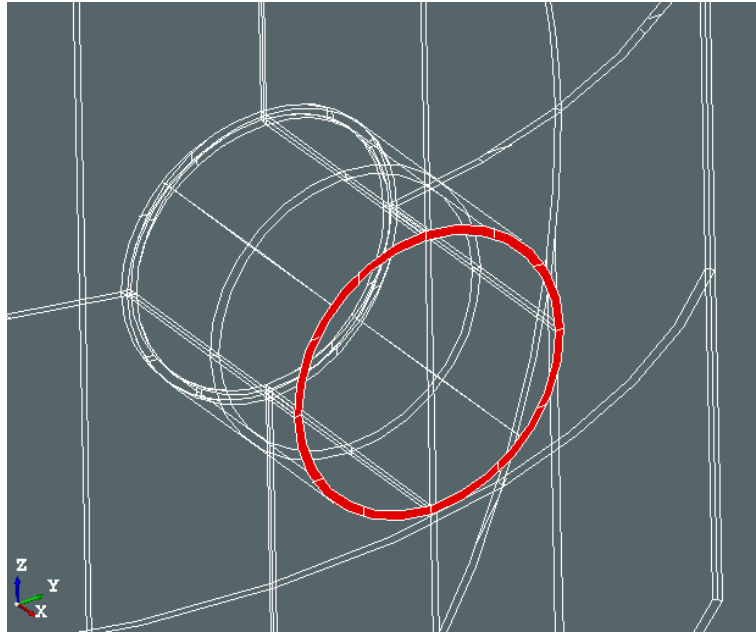
#	Description	Figure
29	<p>Click New Entity -> Basic -> Point.</p> <p>Click on First constructor of Points, give it a Name “P1” and enter co-ordinates as shown below.</p> <p>For Point “P1” enter 0, 0, 0 for X, Y, Z -> Click “Apply”</p> <p>For Point “P2” enter 0, Point_2_y, PV_Half_Height for X, Y, Z -> Click “Apply”</p> <p>For Point “P3” enter 0, 0, PV_Height for X, Y, Z -> Click “Apply and Close”</p>	
30	<p>Next we generate a Curve from these three points.</p> <p>Click New Entity -> Basic -> Curve</p> <p>Select third constructor for Interpolation, keep default Name of “Curve_1”, Creation Mode should be “By Selection”.</p> <p>Now for Points we will directly select them from Object Browser by Clicking on them while holding “Ctrl” key. The order in which you select the points is critical as the interpolation will occur based on that selection.</p> <p>Select Point “P1” then “P2” and last “P3”.</p> <p>Click “Apply and Close”</p>	

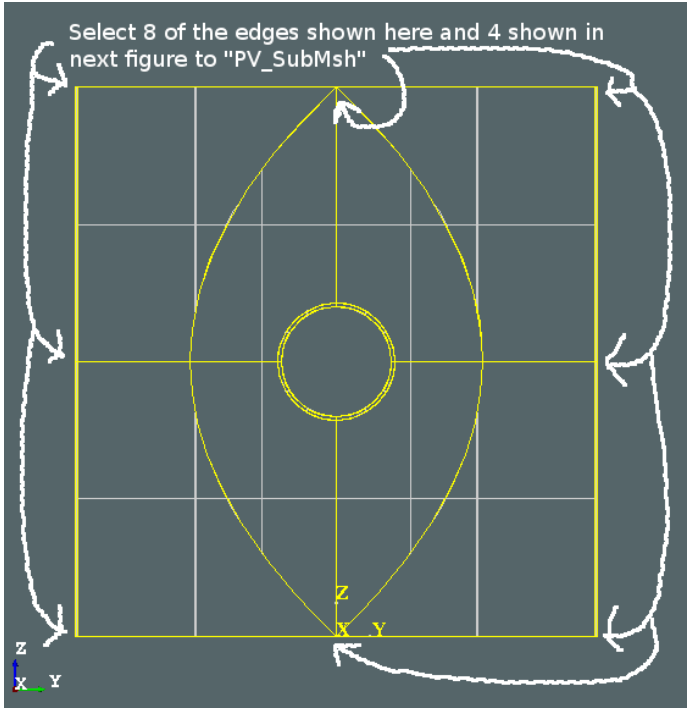
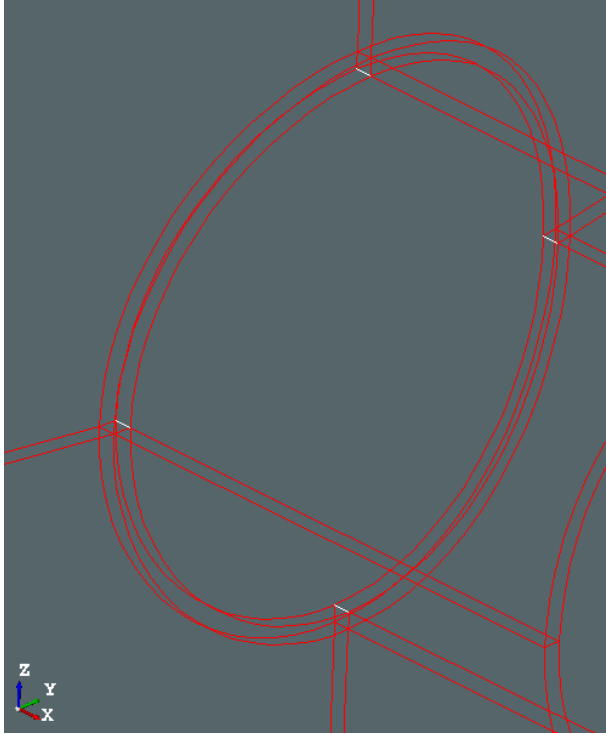
#	Description	Figure
31	<p>We will extrude this “Curve_1” to generate our curved plane.</p> <p>Click New Entity -> Generation -> Extrusion.</p> <p>Select first constructor for Extrusion, keep default Name of “Extrusion_1”, select “Curve_1” as Base and “Vx” as Vector, enter “Nozz_Len” for Height.</p> <p>Click “Apply and Close”</p>	

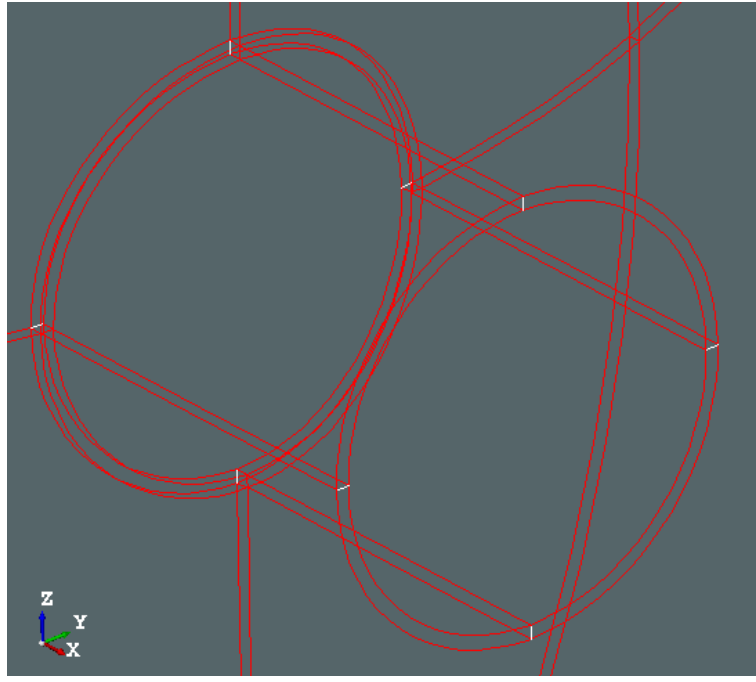
#	Description	Figure
32	<p>Now we will mirror this Extrusion to form our second Tool object for Partition.</p> <p>Click Operations -> Transformation -> Mirror Image.</p> <p>Select third constructor for Mirror, keep default Name of “Mirror_1”, select “Extrusion_1” for Objects and “Plane_Vy” for Plane Mirror.</p> <p>Make sure “Create a copy” is ticked as we want a copy of the existing extrusion.</p> <p>Click “Apply and Close”</p>	

#	Description	Figure
33	<p>Now we partition the geometry.</p> <p>Click Operations -> Partition.</p> <p>Give it a Name “PV_2”, select “PV_1” as Objects and for the Tool Objects select Planes “Extrusion_1” and “Mirror_1”.</p> <p>Make sure Resulting Type is “Solid”.</p> <p>Click “Apply and Close”</p>	
34	<p>We still need more partitions to make sure that every face has 4 edges and every solid has 6 faces.</p> <p>Right Click Cylinder “Cyl_PV_OD” and Select “Show Only”</p> <p>Click New Entity -> Group -> Create.</p> <p>Select third constructor for Shape Type, change the Name to “PV_OD”, the Main Shape should be “Cyl_PV_OD” and select the Cylindrical Surface and Click “Add”</p> <p>Click “Apply and Close”</p> <p>Next do the same for Cylinder “Cyl_NoZ_OD” and give it a name “Noz_OD”.</p>	

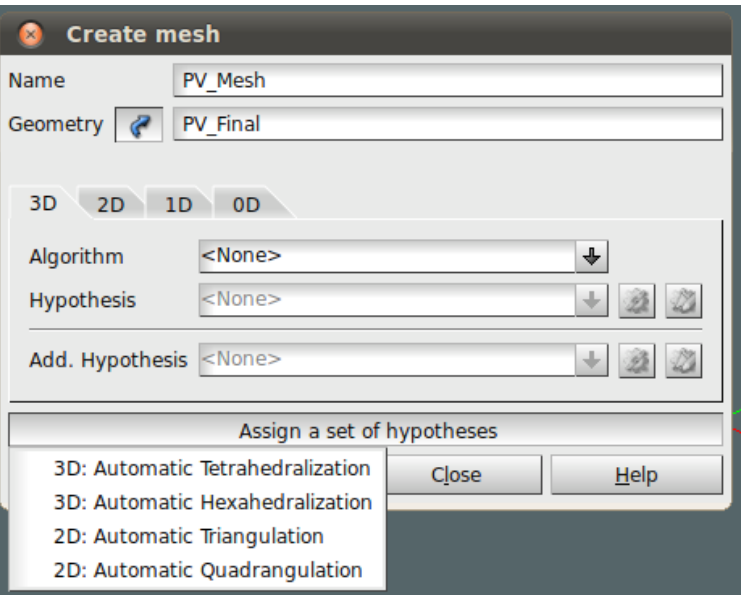
#	Description	Figure
35	<p>Now we partition the geometry one last time.</p> <p>Click Operations -> Partition.</p> <p>Give it a Name "PV_Final", select "PV_2" as Objects and for the Tool Objects select Planes "PV_OD" and "Noz_OD".</p> <p>Make sure Resulting Type is "Solid".</p> <p>Click "Apply and Close".</p> <p>Right Click "PV_Final" and Select "Show Only" and you should be able to see something similar to the adjacent figure.</p>	
36	<p>Now that the Geometry modelling is complete, we need to create Groups of Faces where we will apply Boundary Condition and Loads.</p> <p>Boundary Conditions will remain same as Previous Study we did for Pressure Vessel Shell for most of the part.</p>	

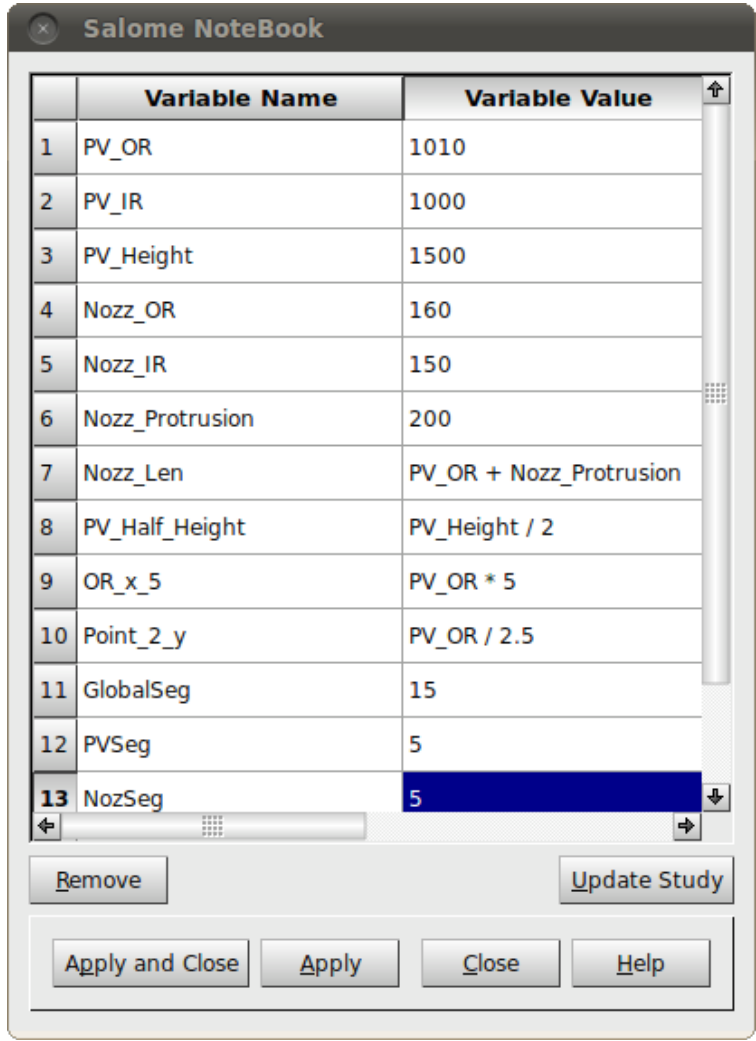
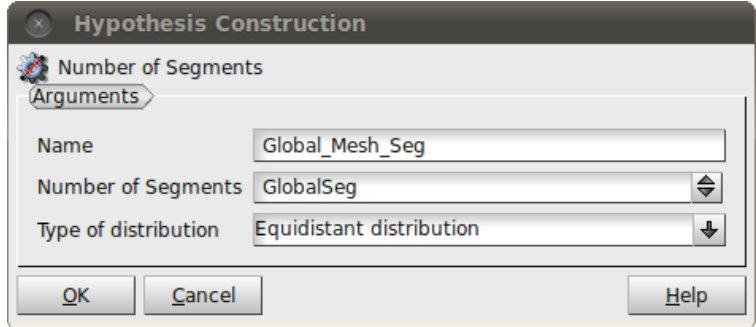
#	Description	Figure
37	<p>Click New Entity -> Create -> Group, Select the Third button to Add Face.</p> <p>Give Name “Dz0” to the bottom face of the shell, “LongPres” to the top face and “Norma” to the two vertical faces.</p> <p>Give Name “Press” to the Inside Face of the Pressure Vessel Shell and also to the Inside face of Nozzle.</p>	
38	<p>We will need a Face Group on the Nozzle to apply Force.</p> <p>Give it a Name “Force” and select 4 faces as highlighted in Red in adjacent figure. Click “Add” to add them and then Click “Apply”</p>	

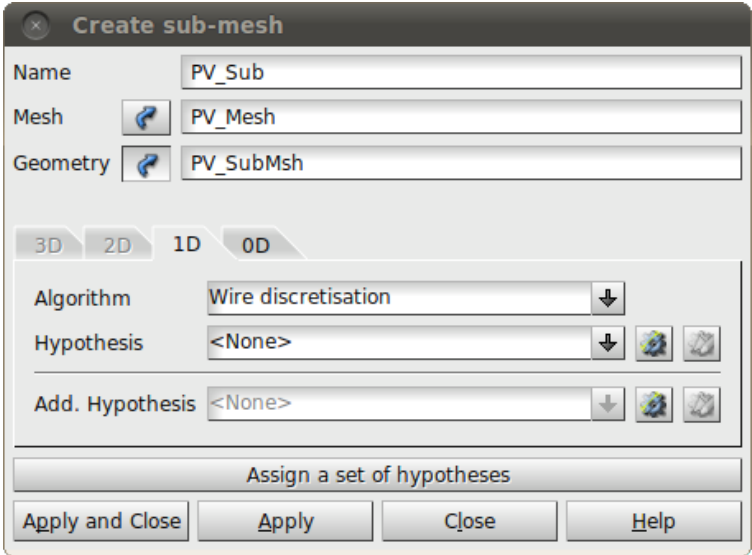
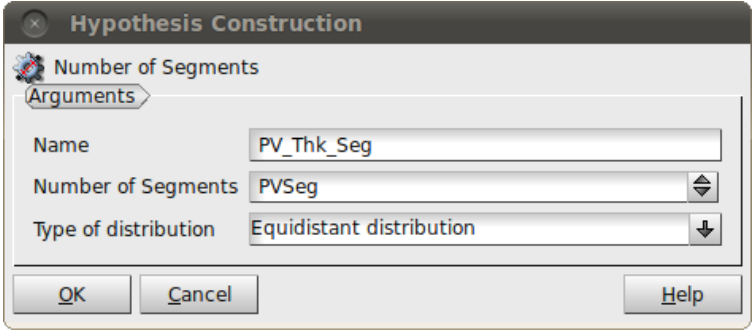
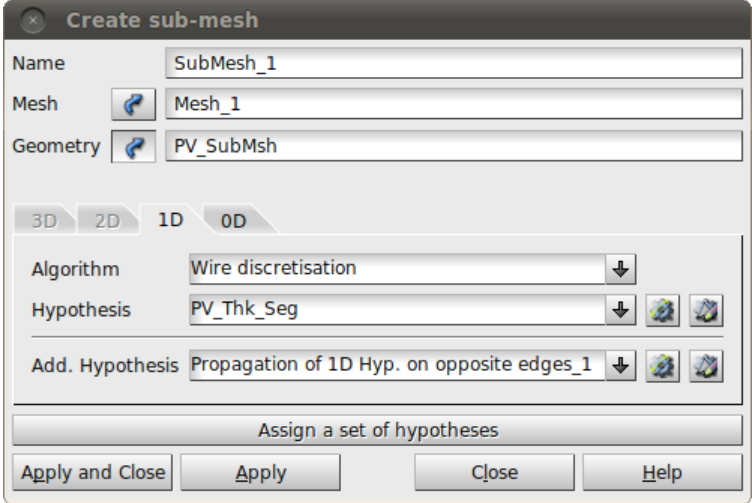
#	Description	Figure
39	<p>Now we need to create Edge Group for Sub Meshes that we will create in the thickness of the Pressure Vessel and Nozzle.</p> <p>In the Create Group Window, Select the Second button to Add Edge.</p> <p>Add all four edges that we had selected in previous chapter for the PV Shell and additional edges that are created due to partitioning of the PV in the middle of the Length.</p>	
40	<p>Also Add four edges as shown in the adjacent figure which are on the Inside of Nozzle and on PV Shell, and give it a Name "PV_SubMsh".</p> <p>"PV_SubMsh" should have 12 edges in it.</p>	

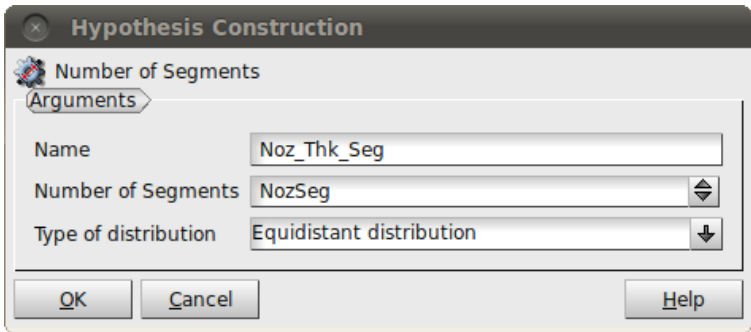
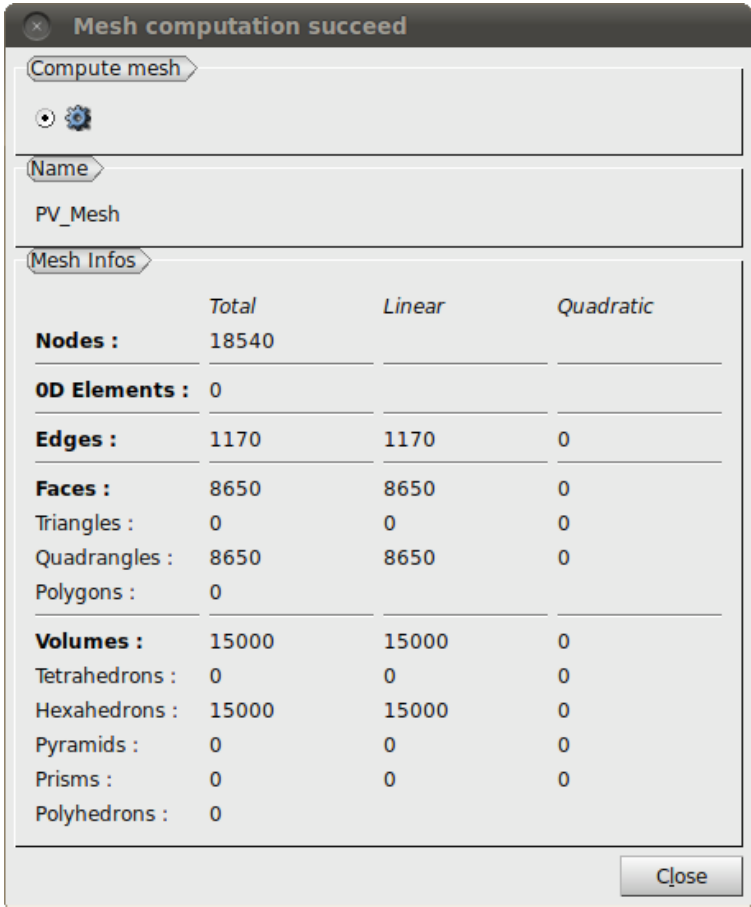
#	Description	Figure
41	<p>Similarly we will create Edge group for “Noz_SubMsh” as shown in adjacent figure.</p> <p>There will be 8 edges in “Noz_SubMsh”</p>	
42	<p>This concludes creating Geometry.</p> <p>Save this file.</p>	

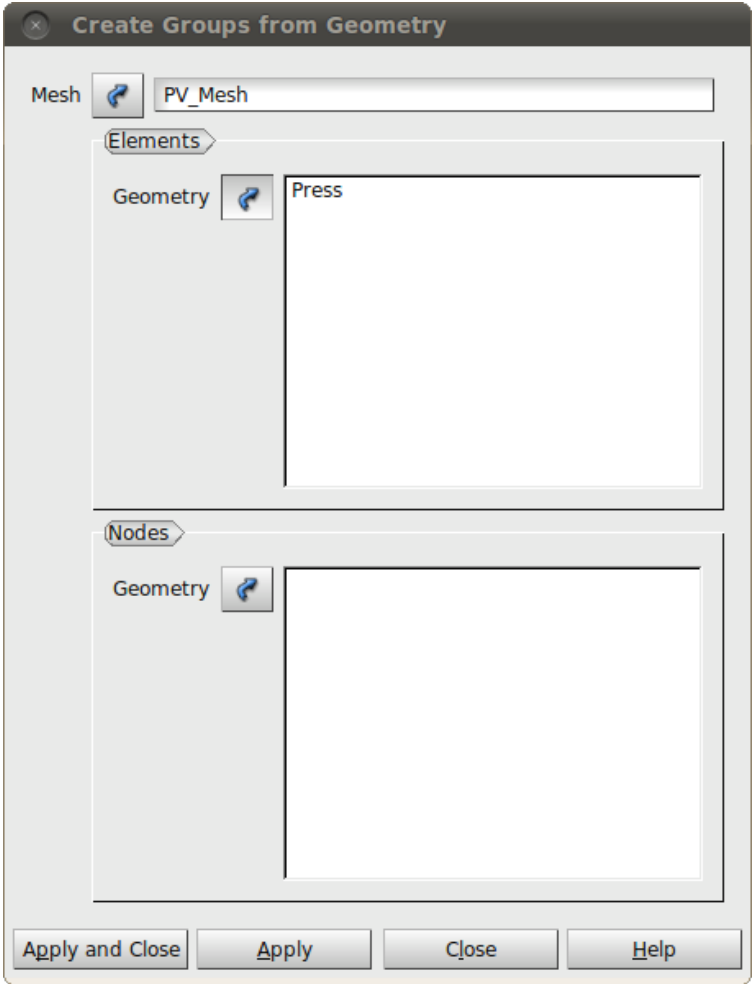
Step 4: Meshing using Notebook

#	Description	Figure
1	<p>We spend quiet a long time Modelling the Pressure vessel and Nozzle junction but it is well worth.</p> <p>Due to the effort spent earlier, we will have to do less effort in Meshing the geometry.</p> <p>Select Mesh module in Salome-Meca.</p>	
2	<p>Click on Mesh -> Create Mesh</p> <p>Change the Name to “PV_Mesh”.</p> <p>Click on the Button “Assign a set of hypotheses” and Select “3D: Automatic Hexahedralization”</p>	

#	Description	Figure																																										
3	<p>Now in the next window it asks for Number of Segments for Global Mesh. We need to define Parameters for the Mesh Number of Segments in Notebook.</p> <p>We will define:</p> <p>GlobalSeg with Value 15</p> <p>PVSeg with Value 5</p> <p>NozSeg with Value 5</p>	 <p>The Salome NoteBook window displays a table of variables and their values. The table has two columns: 'Variable Name' and 'Variable Value'. The variables are listed in a numbered list from 1 to 13. The 'NozSeg' variable (row 13) is highlighted in blue. Below the table are buttons for 'Remove', 'Update Study', 'Apply and Close', 'Apply', 'Close', and 'Help'.</p> <table border="1"> <thead> <tr> <th></th> <th>Variable Name</th> <th>Variable Value</th> </tr> </thead> <tbody> <tr><td>1</td><td>PV_OR</td><td>1010</td></tr> <tr><td>2</td><td>PV_IR</td><td>1000</td></tr> <tr><td>3</td><td>PV_Height</td><td>1500</td></tr> <tr><td>4</td><td>Nozz_OR</td><td>160</td></tr> <tr><td>5</td><td>Nozz_IR</td><td>150</td></tr> <tr><td>6</td><td>Nozz_Protrusion</td><td>200</td></tr> <tr><td>7</td><td>Nozz_Len</td><td>PV_OR + Nozz_Protrusion</td></tr> <tr><td>8</td><td>PV_Half_Height</td><td>PV_Height / 2</td></tr> <tr><td>9</td><td>OR_x_5</td><td>PV_OR * 5</td></tr> <tr><td>10</td><td>Point_2_y</td><td>PV_OR / 2.5</td></tr> <tr><td>11</td><td>GlobalSeg</td><td>15</td></tr> <tr><td>12</td><td>PVSeg</td><td>5</td></tr> <tr><td>13</td><td>NozSeg</td><td>5</td></tr> </tbody> </table>		Variable Name	Variable Value	1	PV_OR	1010	2	PV_IR	1000	3	PV_Height	1500	4	Nozz_OR	160	5	Nozz_IR	150	6	Nozz_Protrusion	200	7	Nozz_Len	PV_OR + Nozz_Protrusion	8	PV_Half_Height	PV_Height / 2	9	OR_x_5	PV_OR * 5	10	Point_2_y	PV_OR / 2.5	11	GlobalSeg	15	12	PVSeg	5	13	NozSeg	5
	Variable Name	Variable Value																																										
1	PV_OR	1010																																										
2	PV_IR	1000																																										
3	PV_Height	1500																																										
4	Nozz_OR	160																																										
5	Nozz_IR	150																																										
6	Nozz_Protrusion	200																																										
7	Nozz_Len	PV_OR + Nozz_Protrusion																																										
8	PV_Half_Height	PV_Height / 2																																										
9	OR_x_5	PV_OR * 5																																										
10	Point_2_y	PV_OR / 2.5																																										
11	GlobalSeg	15																																										
12	PVSeg	5																																										
13	NozSeg	5																																										
4	<p>In the window that pops up, change the Name to “Global Mesh Seg” and enter “GlobalSeg” as Number of Segments.</p> <p>Click “OK”</p> <p>Click “Apply and Close”</p>	 <p>The Hypothesis Construction window shows the 'Number of Segments' dialog. The 'Name' field contains 'Global_Mesh_Seg'. The 'Number of Segments' dropdown menu is set to 'GlobalSeg'. The 'Type of distribution' dropdown menu is set to 'Equidistant distribution'. At the bottom are buttons for 'OK', 'Cancel', and 'Help'.</p>																																										

#	Description	Figure
5	<p>Right Click “PV_Mesh” and Select “Create Sub-Mesh”.</p> <p>Change the Name to “PV_Sub”, select “PV_SubMsh” for Geometry.</p> <p>Select “Wire discretisation” for Algorithm.</p> <p>Click on the Gear button next to Hypothesis and Select “Nb. Segments”</p>	
6	<p>In the window that pops up change the Name to “PV Thk Seg” and Number of Segments to “PVSeg”.</p> <p>Click “OK”</p>	
7	<p>Back to Create sub-mesh window.</p> <p>Click on the Gear button next to “Add. Hypothesis” and Select “Propagation of 1D Hyp. On opposite edges”.</p> <p>This will ensure that the sub mesh is propagated throughout the thickness.</p> <p>Click “Apply and Close”</p>	

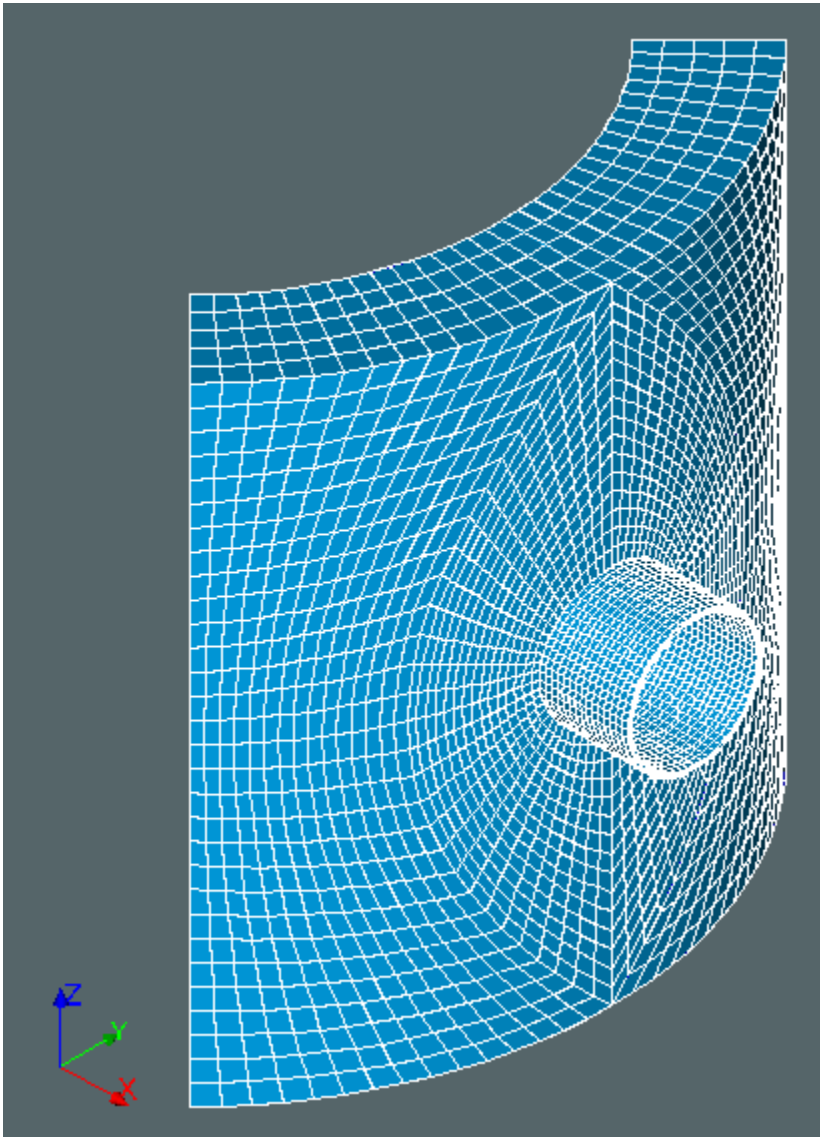
#	Description	Figure
8	<p>Do the same with “Noz_SubMsh” and for the Number of Segments use “NozSeg”.</p> <p>Make sure that “Propagation of 1D Hyp. On opposite edges” has been selected.</p> <p>Click on “Apply and Close”</p>	
9	<p>Right Click on “PV_Mesh” and Select “Compute”.</p> <p>As can be seen the total number of Edges, Faces and Volumes has been reduced.</p> <p>Number of Volumes reduced from 54000 to 15000.</p> <p>This will reduce the time taken by Code_Aster to carry out FE Analysis.</p>	

#	Description	Figure
10	<p>We need to create Mesh Groups from Geometry Group.</p> <p>Right Click “PV_Mesh” and Select “Create Groups from Geometry”.</p> <p>Click on the Curved Arrow next to Geometry and Select “Press” from Object Browser. Click “Apply”</p> <p>Do the same with “Dz0”, “LongPres”, “Norma” and “Force”</p>	
11	<p>This concludes generation of Mesh.</p> <p>Save this study.</p>	

Change Parameter in Notebook

Now is the real test of Notebook. Once geometry and mesh is generated and the file has been saved, Click on File -> Notebook and change the parameters and see if the study updates.

For example, change the value of PV_IR from 1000 to 750 and Click “Update Study”. Salome will be performing the steps that we carried out to generate geometry and mesh again and when done, it will present the final result. Our result is shown below.



John told Esha that using Notebook is very easy and updating geometry and mesh is done by just a click of a button, but as we noticed, Notebook is not powerful yet and complex formulas cannot be entered. If you want to enter complex formulas, you better edit the python file manually.

Summary

John asked Esha to summarise what she learnt by doing this Parametric Modelling. Esha summarised it as below

- ▶ How to edit a Python file.
- ▶ How to enter Parameters in Python file.
- ▶ How to Use Parameters in Geometry and Mesh creation.
- ▶ How to use Notebook in Salome.
- ▶ How to add Parameters and its Value in the Notebook.
- ▶ What are the limitations of Notebook.
- ▶ How to update study by changing Parameters in the Notebook.

Endnotes

- 1 Parameters are called Variables
- 2 You can perform arithmetic calculations in Variable Value field.

What will be covered in Volume 3

Learning new topics from John, Esha was equipped with new enthusiasm. She started working again and the projects that formerly she had to forward to her colleagues due to lack of understanding, she could do them with ease.

Nearly after a year, John caught up with Esha to find out how she was feeling regarding the use of new and a bit more advanced use of Open Source Software for Finite Element Analysis. Esha, now a seasoned engineer, told John that she was very happy to get new information for the help that he had provided her as it helped her in her job. She could do the projects with ease and now was also able to teach others the use of Open Source Software.

Then Esha told John that every now and then an advanced type of FE Analysis comes to her for which she is not ready yet. She has to pass them to her other colleagues as she is not feeling confident and she is feeling a bit lost. As usual John was listening to her words carefully and asked her if he could help. Esha was waiting for John to say that and she listed the analysis which she felt were a bit tough for her to do.

Using Python for Parametric Modelling and FE Analysis

Esha said that using Python for Parametric modelling was good, but every now and then when she has to re-run the study, with more refined mesh, she had to re-do all the steps again. She wanted to know if using Python, can she automate the phase of Modelling, Meshing and running the analysis.

Using Homnard for adaptive meshing

Esha had heard about Homnard and that it can be used for adaptive meshing. She wanted to learn how to use it.

Advanced Thermal FE Analysis

After learning about two types of Thermal FE Analysis, conduction and convection, Esha wanted more. She asked John if he could teach her advanced Thermal Analysis. She wanted to learn Thermal Radiation, Thermal Analysis of an Assembly and so on.

Thermo-Mechanical FE Analysis

Finding out the temperature of different parts in Thermal analysis is good, but there was something missing. Esha wanted to learn, how to find out thermal deformation in a part or assembly. She wanted to know if the thermal Stress Strain curve of the material can be attached to thermal and mechanical analysis.

Pipe Stress Analysis

Esha knew that standard Pipe Stress Analysis programs available in the market use Beam model for Stress analysis. She wanted to learn how to perform Pipe Stress Analysis using Salome-Meca and Code_Aster.

Fluid Structure Integration

Esha remembered that doing Computational Fluid Dynamics, you can find out the pressure on the boundary walls, she wanted to know if there was a way to project these on to the mechanical parts and carry out FE analysis. In short she wanted to learn Fluid Structure Integration of FE Analysis.

After listing the above analysis, Esha looked at John who was smiling and had a sparkle in his eyes. John was ready to teach Advanced FE Analysis to Esha.

John told Esha that both Salome and Code_Aster were capable of conducting the analysis she asked for and much more.

Appendix A

Other sources of information

This section is provided here for motivated users who are hungry for more information on the software covered in this book. Links are provided to the Software home page and also to the User Forums. Links provided to the websites are correct at the time of writing this; if by any chance these links don't work, just use any search engine to find relevant software.

1. Salome website <http://www.salome-platform.org/>
2. Code Aster website <http://www.code-aster.org/V2/spip.php?rubrique2>
3. CAELinux website <http://www.caelinux.com/CMS/>
4. Efficient website <http://engineering.moonish.biz/efficient/>
5. Salome forum <http://www.salome-platform.org/forum>
6. Code_Aster forum <http://www.code-aster.org/forum2/>
7. CAELinux wiki http://www.caelinux.org/wiki/index.php/Main_Page

Appendix B

Installing Software required for this book

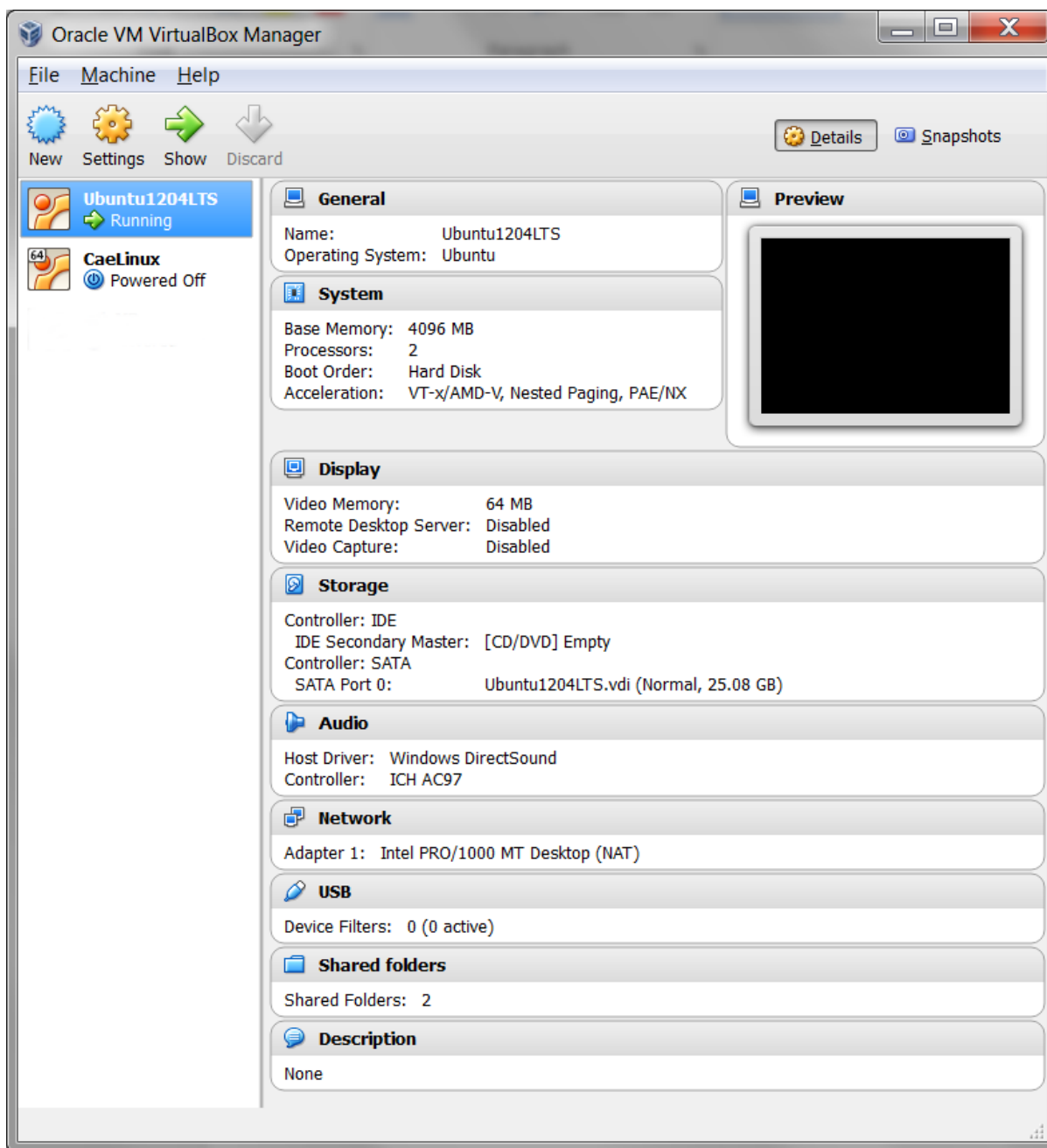
All case studies mentioned in this book can be performed on a computer that has Ubuntu 12.04 Operating System and which has Salome-Meca and Efficient installed on it.

Ubuntu 12.04 Configuration

Ubuntu 12.04 can be installed on a computer alongside Windows 7 or it can be installed as a Virtual Operating System. Decision of how to install Ubuntu is left to the readers.

I have installed Ubuntu 12.04 in a virtual environment using Virtual Box.

Configuration of Ubuntu on Virtual Box is shown below

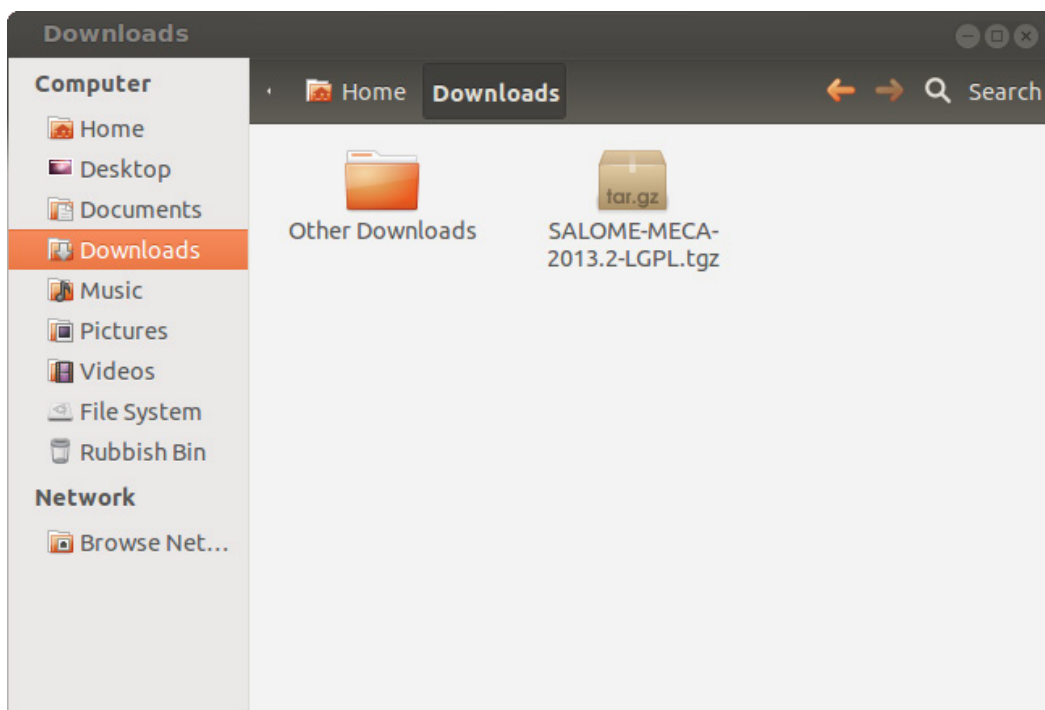


Salome-Meca 2013.2 installation

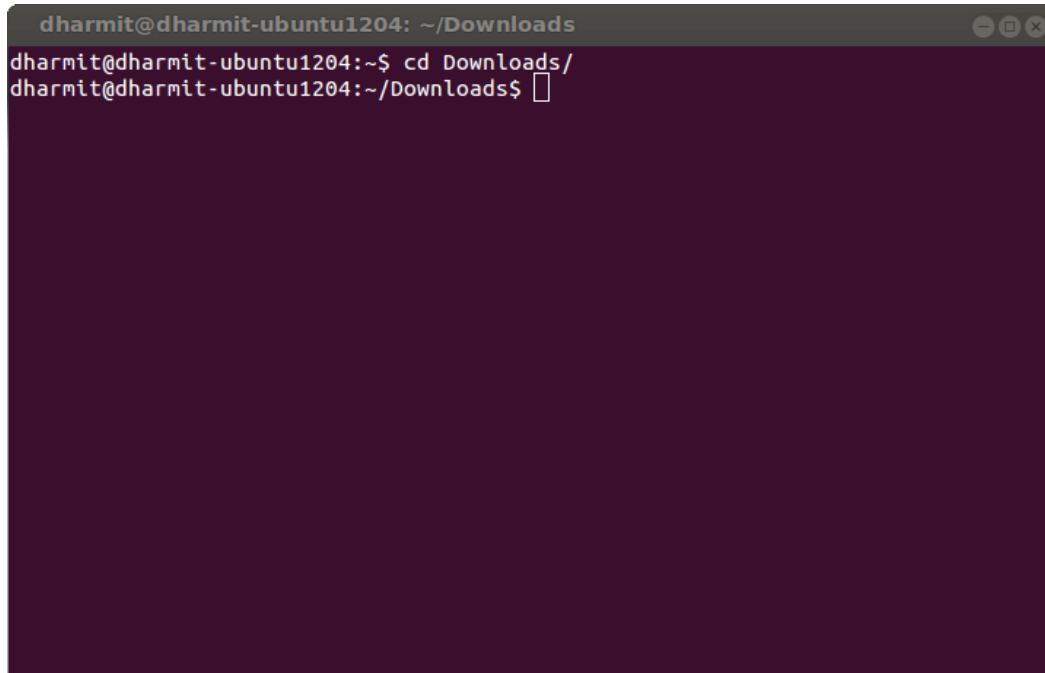
Download Salome-Meca 2013.2 from their website

<http://www.code-aster.org/V2/spip.php?article303>

It downloads as a rar file and it is in my Downloads folder as shown below.



Now Open Terminal and go to Downloads Folder



Enter the command by typing in Terminal

```
tar xvf SALOME-MECA-2013.2-LGPL.tgz && ./SALOME-MECA-2013.2-LGPL.run
```

Follow the prompt and install Salome-Meca 2013.2 in desired location.

Once Salome-Meca installation is complete, a desktop icon will be created which can be double clicked to start Salome-Meca 2013.2

Have fun.

If you have any trouble installing the software, consult Code_Aster Forums on

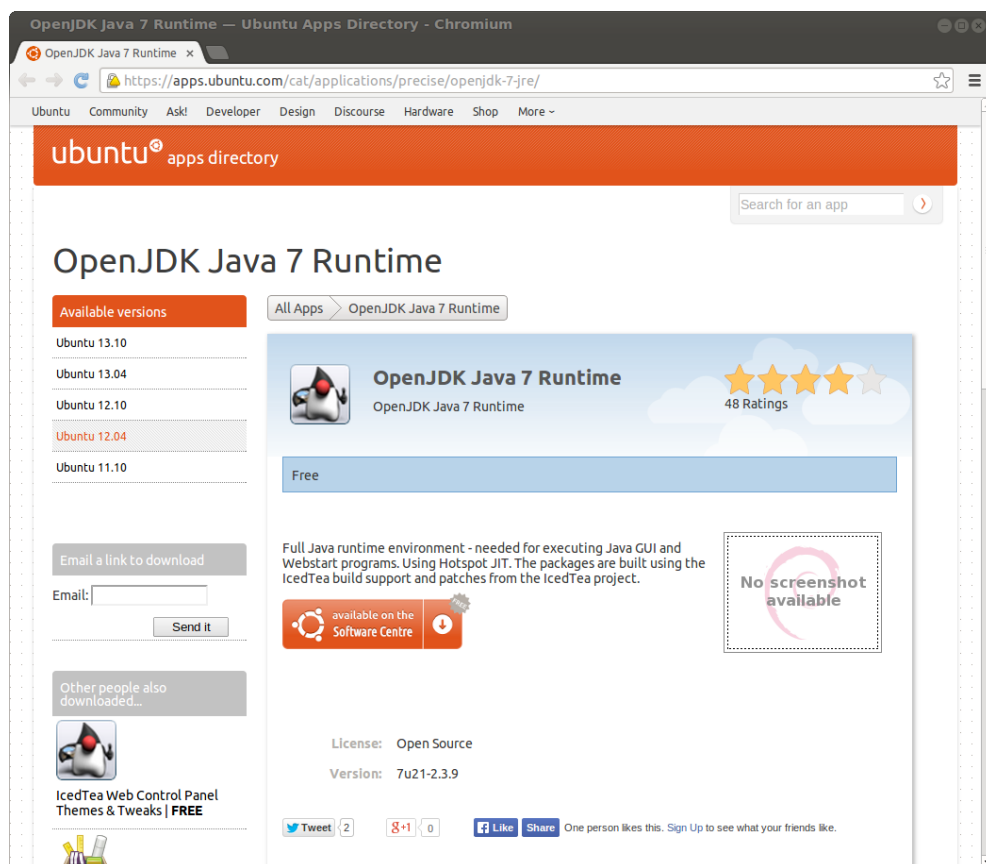
<http://www.code-aster.org/forum2/viewforum.php?id=26>

Efficient Install

Efficient is developed in Java and to run it in Ubuntu, you will need OpenJDK Java 7 runtime.

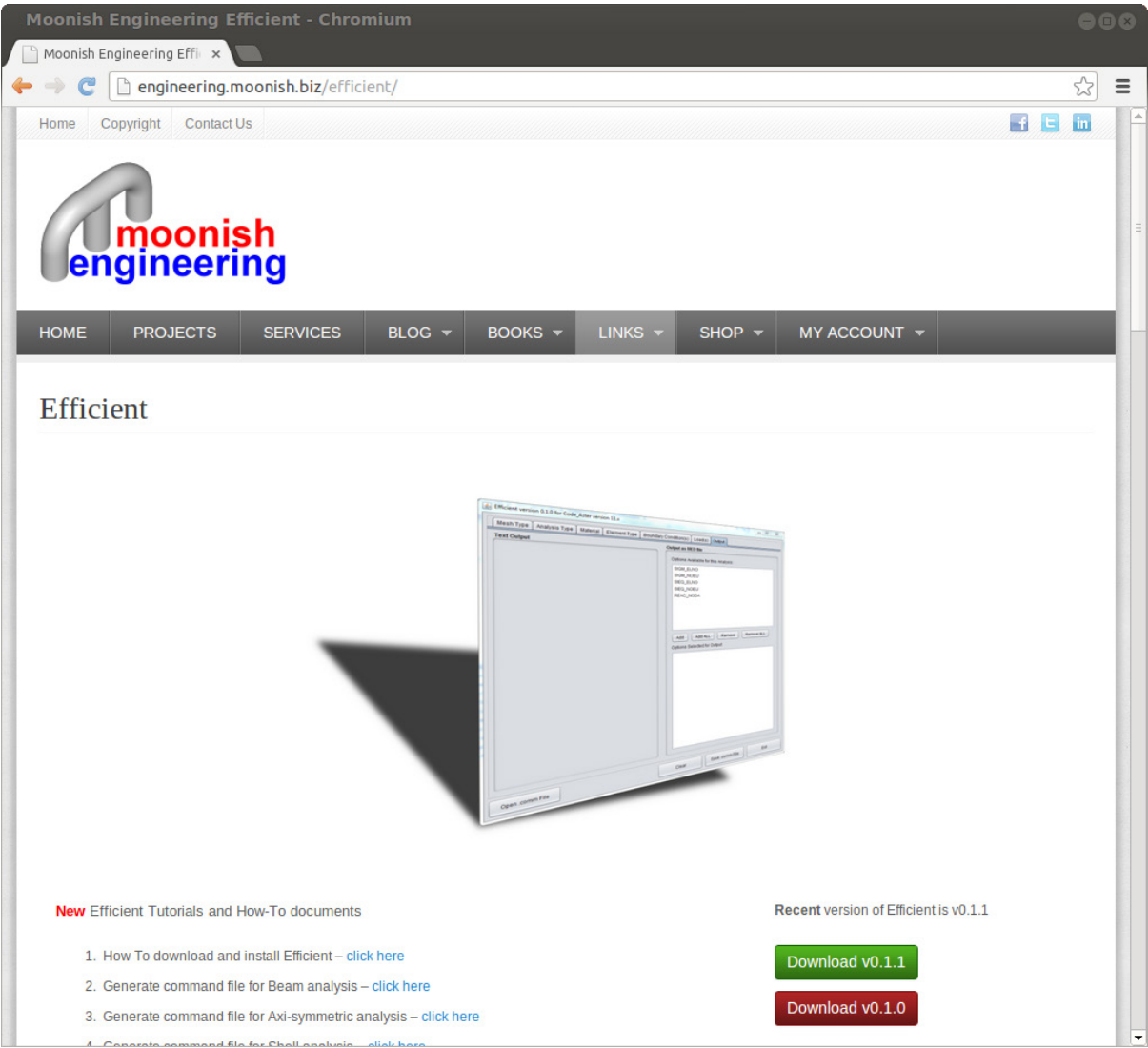
To download and install it go to

<https://apps.ubuntu.com/cat/applications/precise/openjdk-7-jre/>



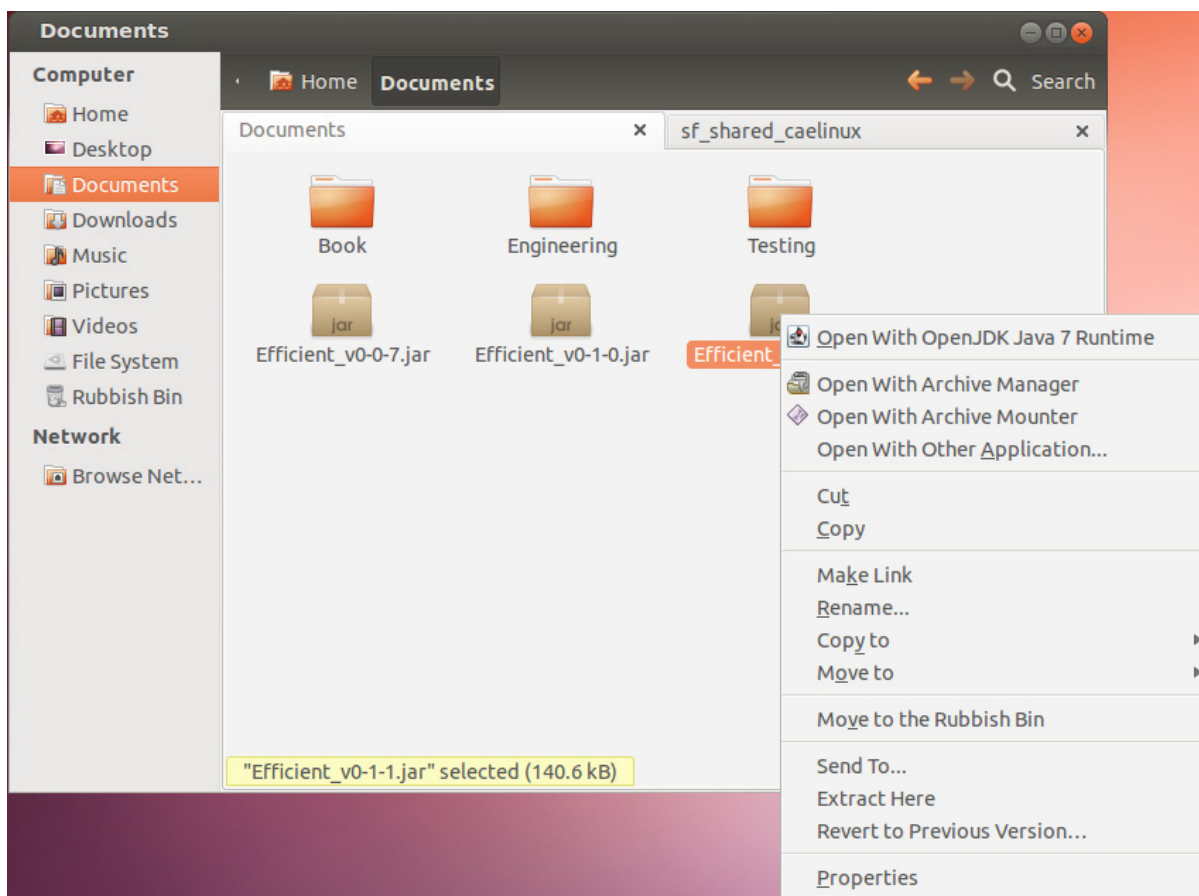
Install it on your Ubuntu installation.

Download Efficient version 0.1.1 from <http://engineering.moonish.biz/efficient/>



On the right hand side latest Efficient version can be downloaded. Click on “Download v0.1.1” and a Java Jar file named “Efficient_v0-1-1.jar” will be downloaded on your computer. Copy it to a proper location.

To run Efficient, Right Click on “Efficient_v0-1-1.jar” and Select “Open With OpenJDK Java 7 Runtime”



Intermediate Finite Element Analysis with Open Source Software

First Edition

By

Dharmit Thakore

CPEng, RPEQ, Brisbane, QLD, Australia

Features:

- ▶ Builds on the foundation created in Finite Element Analysis with Open Source Software
- ▶ Still using easy to read and understand language loved by many users
- ▶ Screen shots to help users visualise the procedure and understand efficiently
- ▶ Real world examples used in the book to make users confident of their knowledge and the power of Open Source software being used in real world
- ▶ Each chapter introduces a new concept of conducting FE Analysis so that the user is not bombarded with too much information in one Chapter.