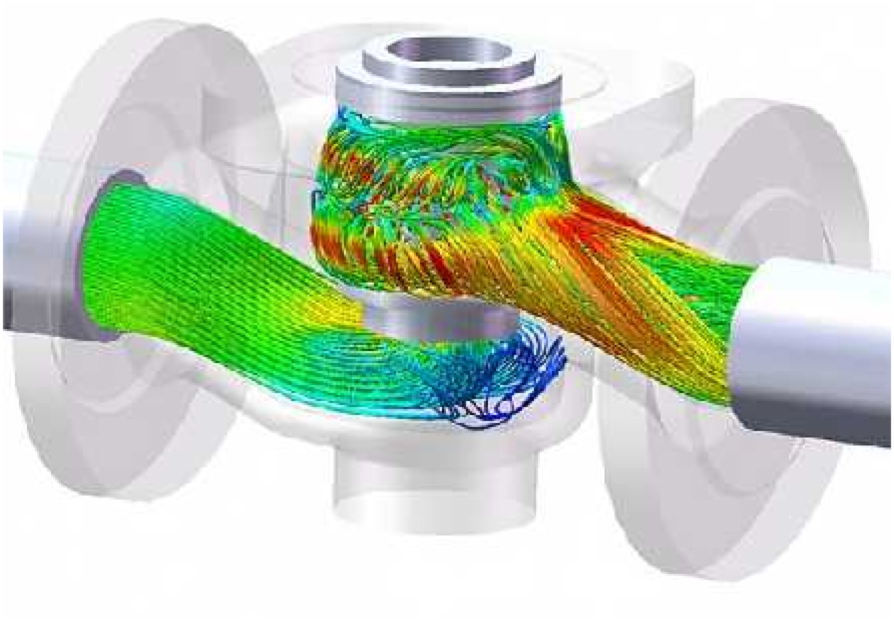
Sri Venkateshwaraa College of Engineering &Technology

(Approved by AICTE, New Delhi & Affiliated to Pondicherry University - Ariyur)
13-A, Pondy-Villupuram Main Road, Ariyur, Puducherry-605 102

## DEPARTMENT OF
## MECHANICAL ENGINEERING



# COMPUTATIONAL FLUID DYNAMICS LABORATORY (MEP63)

LAB MANUAL

**III-YEAR**

**VI – SEMESTER**

*Prepared by*
***Ravindran.S***
*Assistant Professor*

# Contents

# SYLLABUS

## MEP63 COMPUTATIONAL FLUID DYNAMICS LABORATORY (0 0 3 2)

### OBJECTIVES

- Introduce the students to the science of computational fluid dynamics and heat transfer.
- Familiarity with pre and post processing steps in CFD study.
- Using physics based simulation for computer aided design and engineering.
- Grid generation and boundary conditions for complex geometry.
- Understanding the multi physical simulation approach for phenomena under investigation
- Design Optimization using CFD

*Geometry and Grid Generation - SaloME*

*Solver - OpenFOAM*

*Analysis of Results - Para View*

1. 2D and 3D structured grid generation – flat plates, aerofoil
2. 3D unstructured grid generation – pipe, external aerodynamics
3. Incompressible internal laminar flows
4. Incompressible external laminar flows
5. Incompressible internal turbulent flows
6. Incompressible external turbulent flows
7. Forced Convection flows
8. Buoyancy driven flows
9. Multiphase flows - Capillary driven
10. Compressible flows
11. Non Newtonian and Biological flows
12. Fluid Structure Interaction

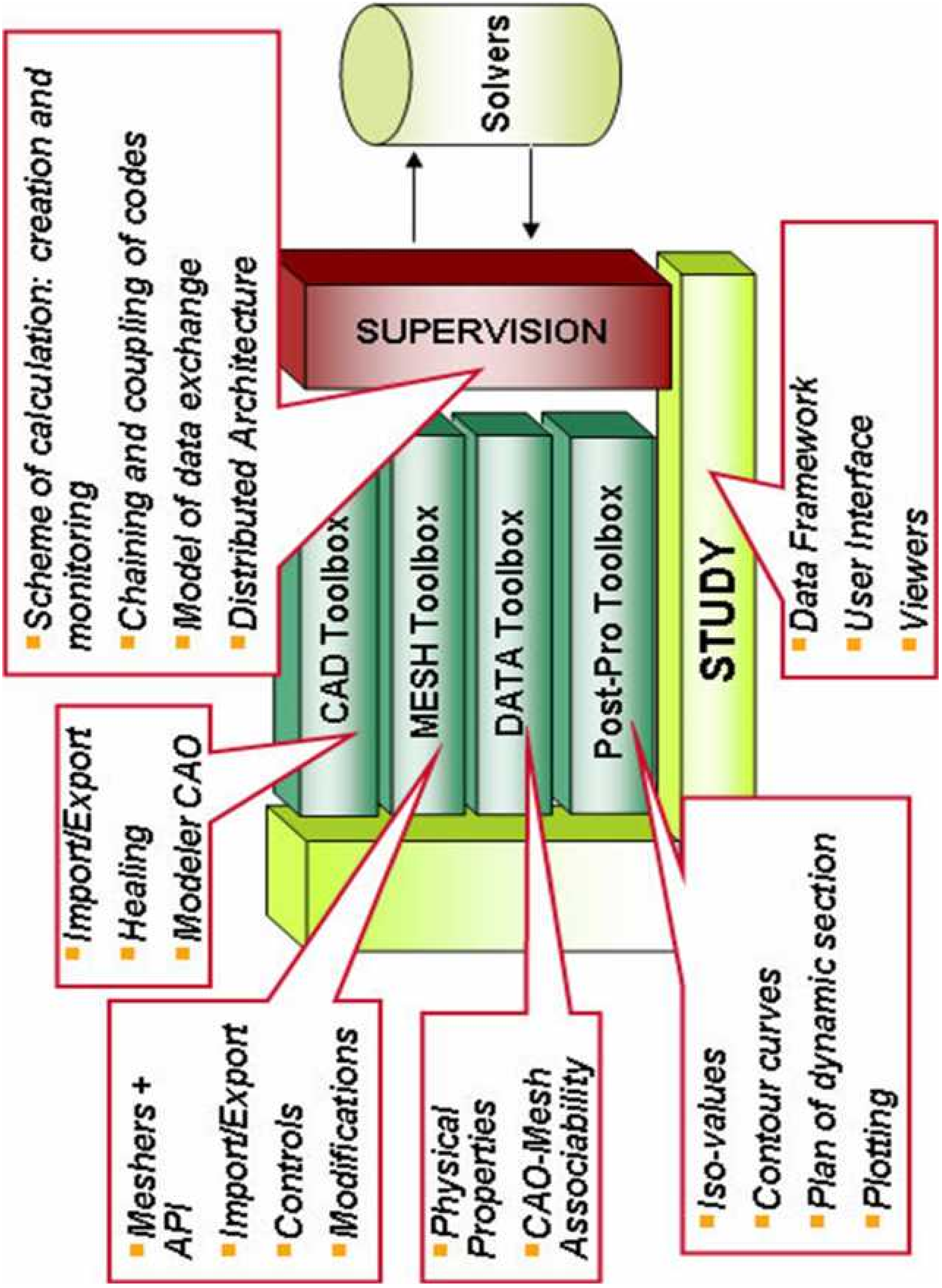
### Reference Books:

1. OpenFOAM 2.3.0 User Manual, 2014.
2. Salome 7.4.0 User Manual, 2014.
3. ParaView 4.2.0 User Manual, 2014.

### Web Resources:

1. https://www.coursera.org/course/spobuildaerodynamics
2. http://nptel.ac.in/courses/101106045
3. http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-100- aerodynamics-fall-2005
4. https://www.youtube.com/playlist?list=PLRuuYmz0Py9_VBgp1p4h84RaVNIelyWV-

**SALOME architecture:**



# SALOME

## Introduction

SALOME is a free software that provides a generic platform for Pre and Post-Processing for numerical simulation. It is based on an open and flexible architecture made of reusable components available as free software.

SALOME is a CAD/CAE integration platform. It provides reusable components for:

- 3D modeling (bottom-up construction, import, healing);
- Visualization;
- Computational schemas management;
- Post-processing.

SALOME is tailored for integration of custom components:

- CAD interfaces;
- Mesh generators;
- Finite Element solvers with specific pre-processors.

The SALOME platform is available in Open Source.

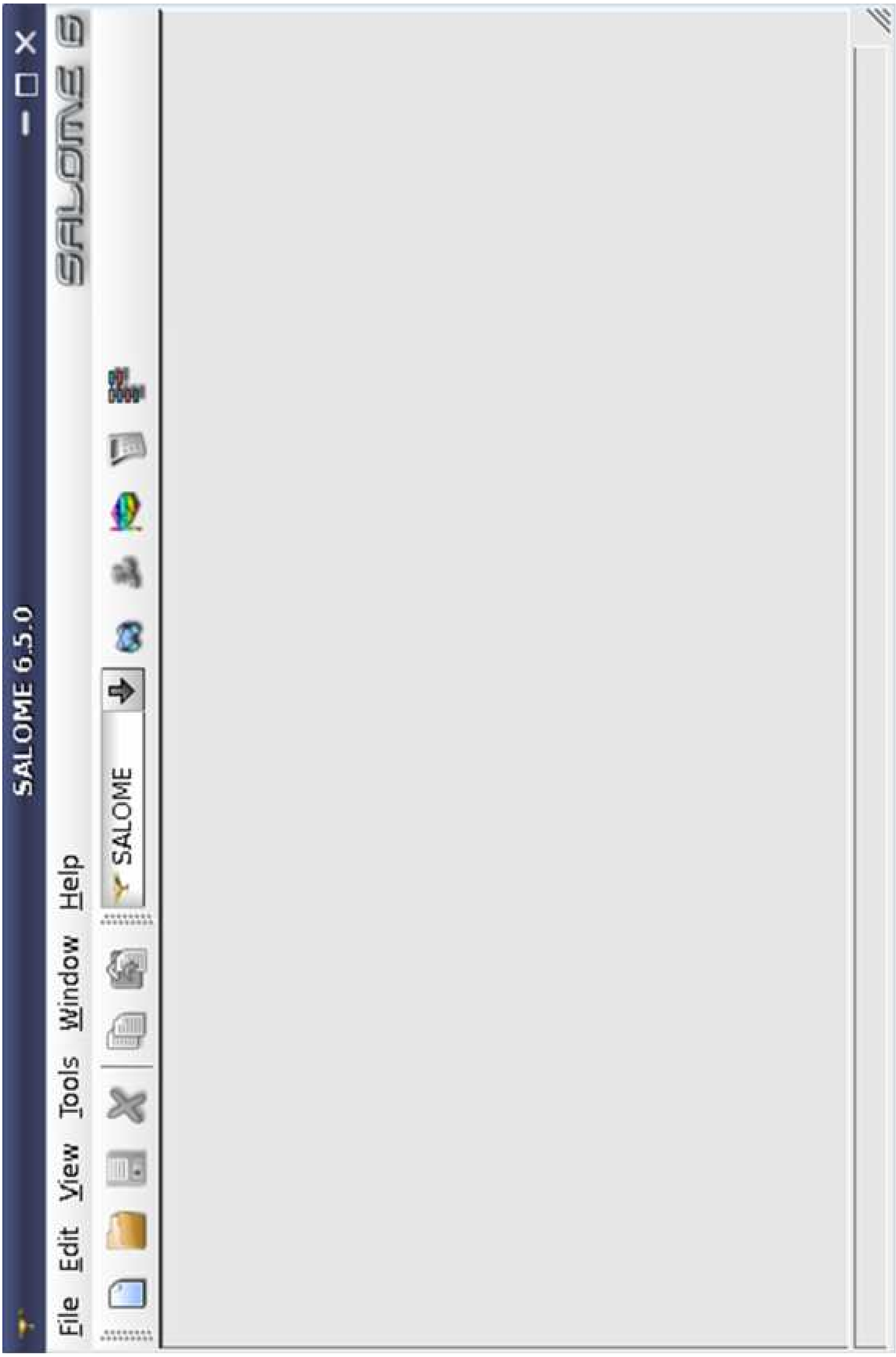
Salome Platform:

- Supports interoperability between CAD modeling and computation software (CAD-CAE link)
- Makes easier the integration of new components on heterogeneous systems for numerical computation
- Sets the priority to multi-physics coupling between computation software
- Provides a generic user interface, user-friendly and efficient, which helps to reduce the costs and delays of carrying out the studies
- Reduces training time to the specific time for learning the software solution which has been based on this platform
- All functionalities are accessible through the programmatic integrated Python console.

Some of SALOME main functions

- Define geometrical models (create/modify geometrical items), import and export them using the BREP, IGES and STEP formats;
- Define meshing of these geometrical items, import and export them;
- Handle physical properties and quantities attached to geometrical items, import and export them to a reusable format;
- Perform computations using a solver (optionally provided): read input data, configure the solver, and write calculation results);
- Visualize result fields in 3D, 2D and export images of their visualization to an appropriate format;
- Manage study schemes: definition, save/restore;
- Manage computation schemes: definition, execution.

**Salome GUI**



**SALOME architecture:**

SALOME architecture is based on CORBA technology using distributed system model of applications. SALOME combines several software components, which are built in such a way that it allows to integrate solvers and existing meshing algorithms along with the specification of physical properties for a given domain. The originality of this approach is that the various components must cooperate dynamically and be configurable.

SALOME platform integrates a number of modules each having its own function:

- KERNEL: provides a common shell for all components, which can be integrated into the SALOME platform.
- GUI: provides visual representation: basic widgets, viewers etc. Third party modules optionally can have GUI, but in any case they always connected with KERNEL.
- Geometry: facilitates construction and optimization of geometrical models using a wide range of CAD functions.
- Mesh: generates meshes on geometrical models previously created or imported by the Geometry component.
- ParaVis: performs data visualization and post-processing.
- MED: allows to work with MED files.

**Introduction to Salome GUI:**

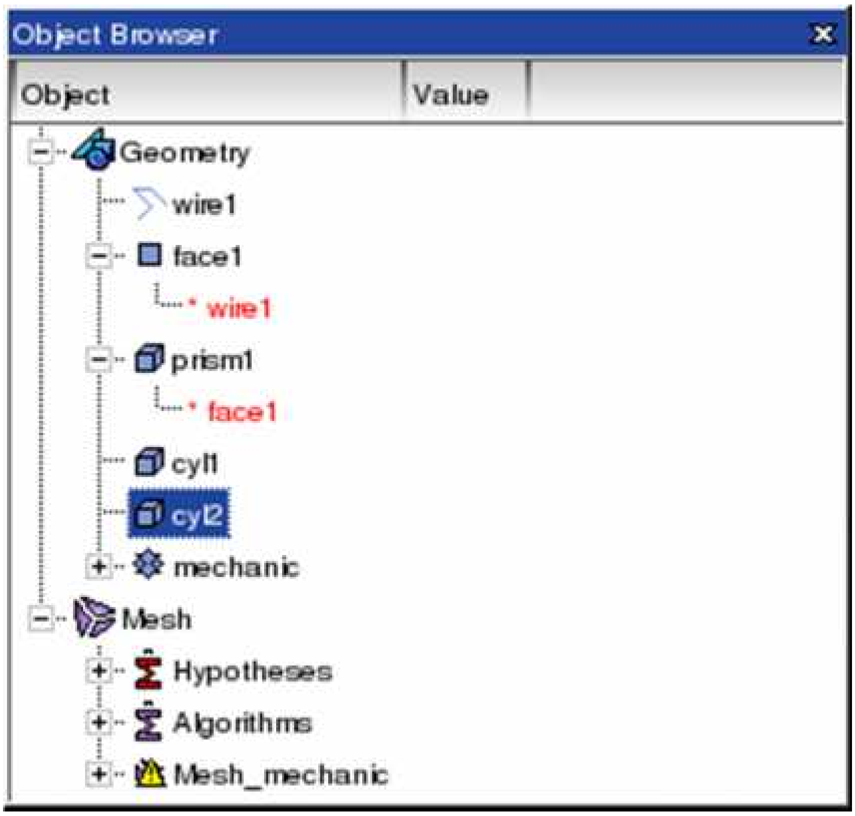
SALOME GUI is based on SUIT (Salome User Interface Toolkit) that proposes very flexible, powerful and safe mechanisms of interaction with SALOME tool components (both CORBA and standalone), resource management, viewers and selection handling, exception and signals processing.

All Graphical User Interfaces of standard modules have plug-in structure, i.e. each module is placed in a dynamic library which is loaded on demand. So, all modules add their own menu items, buttons in toolbar, windows etc. GUI (Graphical User Interface) provides a common shell for all components, which can be integrated into the SALOME platform and some basic GUI functionalities, common for all modules.

Menu bar:

- File menu corresponds to Study management functionalities (creation, saving, loading, connecting, disconnecting, editing studies, etc.)
- Edit menu gives access to Copy/Paste allowing to paste the objects from one study into the other. The availability of this functionality depends on the module and the nature of an operation or an object.
- View and Window menus provide functionalities of Desktop management. Basically, they allow to show/hide toolbars, activate and manage study windows, change the appearance of Salome, etc. Among key parts of study window are:
  - Viewer window, used for visualization of objects,
  - Object Browser, used for management of objects created or imported into the SALOME application, also providing search possibilities with the Find Tool, and
  - Notebook - allows to predefine study variables.
  - Python interpreter, used for direct input of python commands and dumping studies into Python scripts.
  - Almost all aspects of Salome look and feel can be tuned by the user in Themes dialog.
- Tools menu gives access to: Catalogue Generator, Registry tool and Python plugins
- Help menu gives access to the help on Salome modules. The appropriate help page can also be called from any operation dialog via Help button.

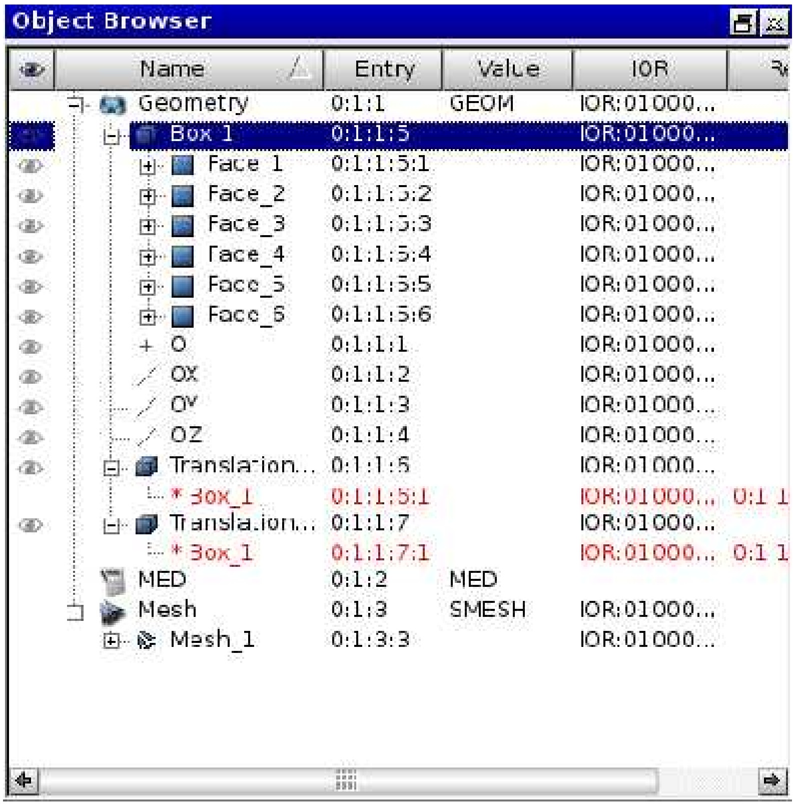
**Object Browser**



Object Browser

| Object | Value |
|---|---|
| Geometry | |
| wire1 | |
| face1 | |
| * wire1 | |
| prism1 | |
| * face1 | |
| cyl1 | |
| cyl2 | |
| mechanic | |
| Mesh | |
| Hypotheses | |
| Algorithms | |
| Mesh_mechanic | |

**Object Browser with supplementary attributes of the objects**



Object Browser

| Name | Entry | Value | IOR | Re |
|---|---|---|---|---|
| Geometry | 0:1:1 | GEOM | IOR:01000... | |
| Box 1 | 0:1:1:5 | | IOR:01000... | |
| Face 1 | 0:1:1:5:1 | | IOR:01000... | |
| Face_2 | 0:1:1:5:2 | | IOR:01000... | |
| Face_3 | 0:1:1:5:3 | | IOR:01000... | |
| Face_4 | 0:1:1:5:4 | | IOR:01000... | |
| Face_5 | 0:1:1:5:5 | | IOR:01000... | |
| Face_6 | 0:1:1:5:6 | | IOR:01000... | |
| O | 0:1:1:1 | | IOR:01000... | |
| OX | 0:1:1:2 | | IOR:01000... | |
| OY | 0:1:1:3 | | IOR:01000... | |
| OZ | 0:1:1:4 | | IOR:01000... | |
| Translation... | 0:1:1:6 | | IOR:01000... | |
| * Box_1 | 0:1:1:5:1 | | IOR:01000... | 0:1 1 |
| Translation... | 0:1:1:7 | | IOR:01000... | |
| * Box_1 | 0:1:1:7:1 | | IOR:01000... | 0:1 1 |
| MED | 0:1:2 | MED | | |
| Mesh | 0:1:3 | SMESH | IOR:01000... | |
| Mesh_1 | 0:1:3:3 | | IOR:01000... | |

**Using Object Browser**

The Object Browser in SALOME is destined for displaying the structure of the current study in a tree-like form. It contains:

- components, loaded during the current session
- objects created with the help of different components (The objects created with the help of a definite component are located in the folder having the name of this component)
- references to different objects (they are highlighted in red)

The Object Browser is destined to getting quick access to different objects created during SALOME session.

The Object Browser may contain supplementary attributes of the objects displayed in additional columns. By default, these columns are not displayed - displaying/hiding these columns is possible through setting study preferences or right-clicking on the attributes bar and toggling the necessary attributes.

**SALOME desktop**

SALOME applies Single Document Interface (SDI) approach. Only one study (document) can be opened at any given moment.

The front-end of the application is a single Desktop top-level window that embeds all GUI operations. The desktop window provides access to standard GUI elements, such as viewers, main menu, toolbars, dockable windows, dialog boxes, etc.

The main elements of the SALOME Desktop are represented on the picture below:

The main parts of the SALOME Desktop are:

- Main menu - This is a set of context-sensitive menus updated according to the functionality of the currently active component. In general, these menus contain all functional options of SALOME platform.
- Standard toolbar - contains standard actions that allow creating new or opening existing study, saving study, deleting objects, etc.
- Components toolbar - allows activating components in SALOME platform.
- Module toolbars - contain actions specific to the currently active module.
- Viewer toolbar - provides a set of tools destined for visualization of the presentations in the viewer.

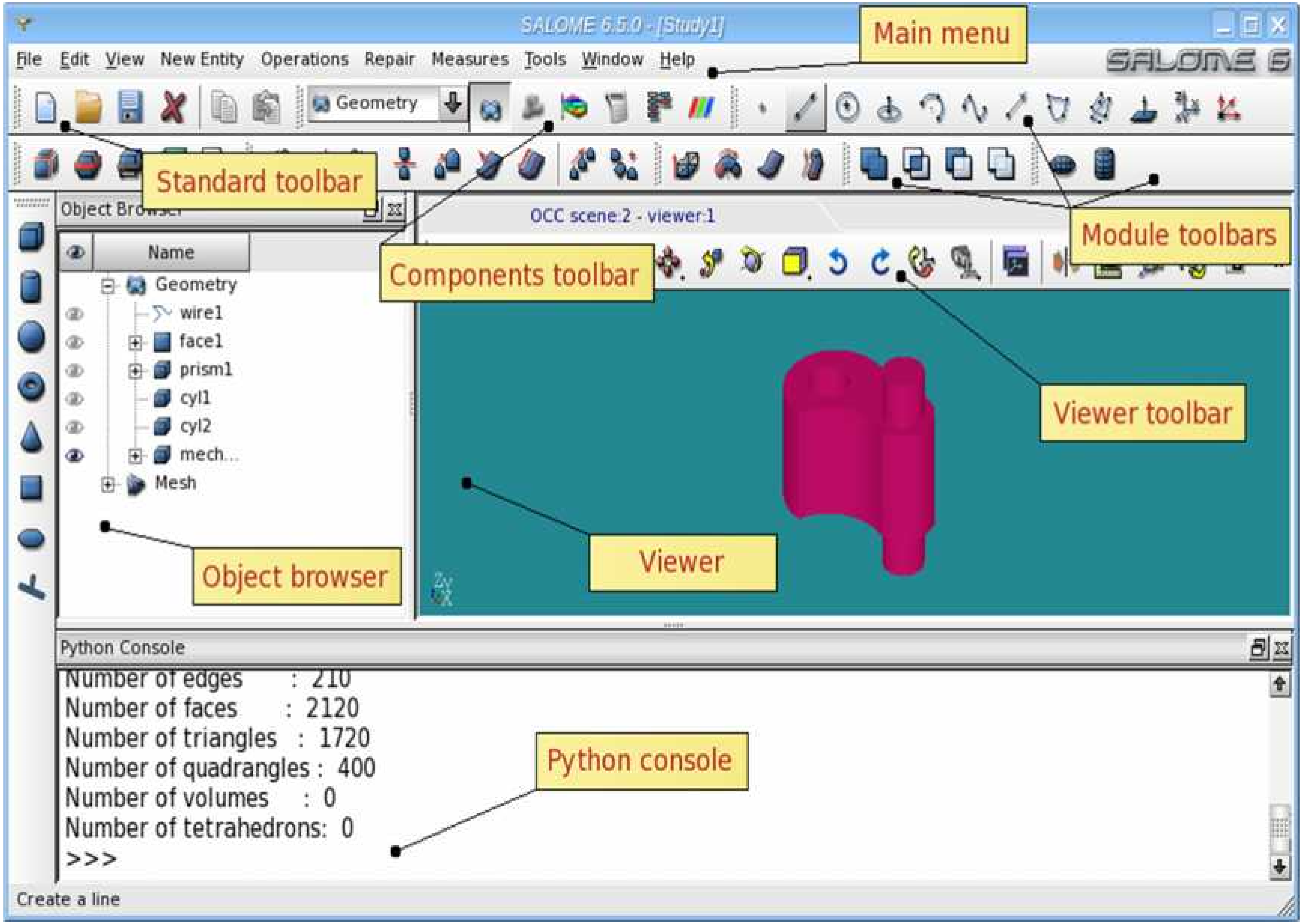
All toolbars in the SALOME Platform are context-sensitive. Loading of a definite component (Geometry, Mesh etc.) will automatically add some additional toolbars specific to this component. To display/hide a toolbar, in the main menu choose View > Toolbars, and from this submenu check or uncheck the toolbar that you would like to display/hide.

By default all toolbars are located on top of the SALOME desktop, but you can relocate a toolbar by dragging it with the mouse and dropping in any place of the window.

Parts of the study window:

- Object browser - Management of objects created or imported into SALOME application.
- Viewer - This window is destined for visualization of presentations.
- Python console - Window for Python interpreter. This window functions like a standard document: the pop-up menu invoked by right-click in this window gives access to Copy/Paste/SelectAll/ClearAll options.

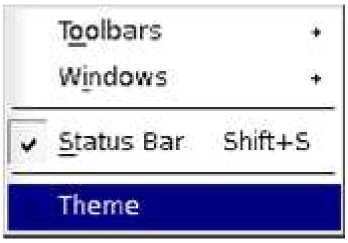
## SALOME desktop



```
Number of edges       :  210
Number of faces       :  2120
Number of triangles   :  1720
Number of quadrangles :  400
Number of volumes     :  0
Number of tetrahedrons:  0
>>>
```

Object Browser, Python Console, MenuBar, etc are, in fact, separate dockable windows. Dockable windows can be placed in special areas, on the borders of desktop like toolbars. You can change the size and the position of dockable windows, hide them or even place outside desktop. SALOME saves in the same file all positions and sizes of dockable windows. It means that the difference in settings of the same windows (for example Python console) is possible between different modules (the same window can have different size and position when other module is activated). Also each module can indicate which common windows are necessary for working, and only they will be displayed when this module is active.

All parts of Salome desktop and study window can be managed through View menu.



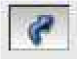
- Toolbars - allows to show and hide toolbars.
- Windows - allows to show and hide dockable windows, such as Object Browser and Python Console.
- Status Bar allows showing and hiding the Status Bar.
- Theme - allows to select and edit the style (colors and fonts) for Salome session.

**Introduction to Geometry Module**

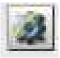
Geometry module of SALOME is destined for:

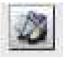
- import and export of geometrical models in IGES, BREP, STEP, STL, XAO and VTK formats;
- construction of geometrical objects using a wide range of functions;
- viewing geometrical objects in the OCC viewer;
- transformation of geometrical objects using various algorithms;
- optimization of geometrical objects;
- viewing information about geometrical objects using measurement tools;
- designing shapes from pictures;

**Various Buttons Icons:**

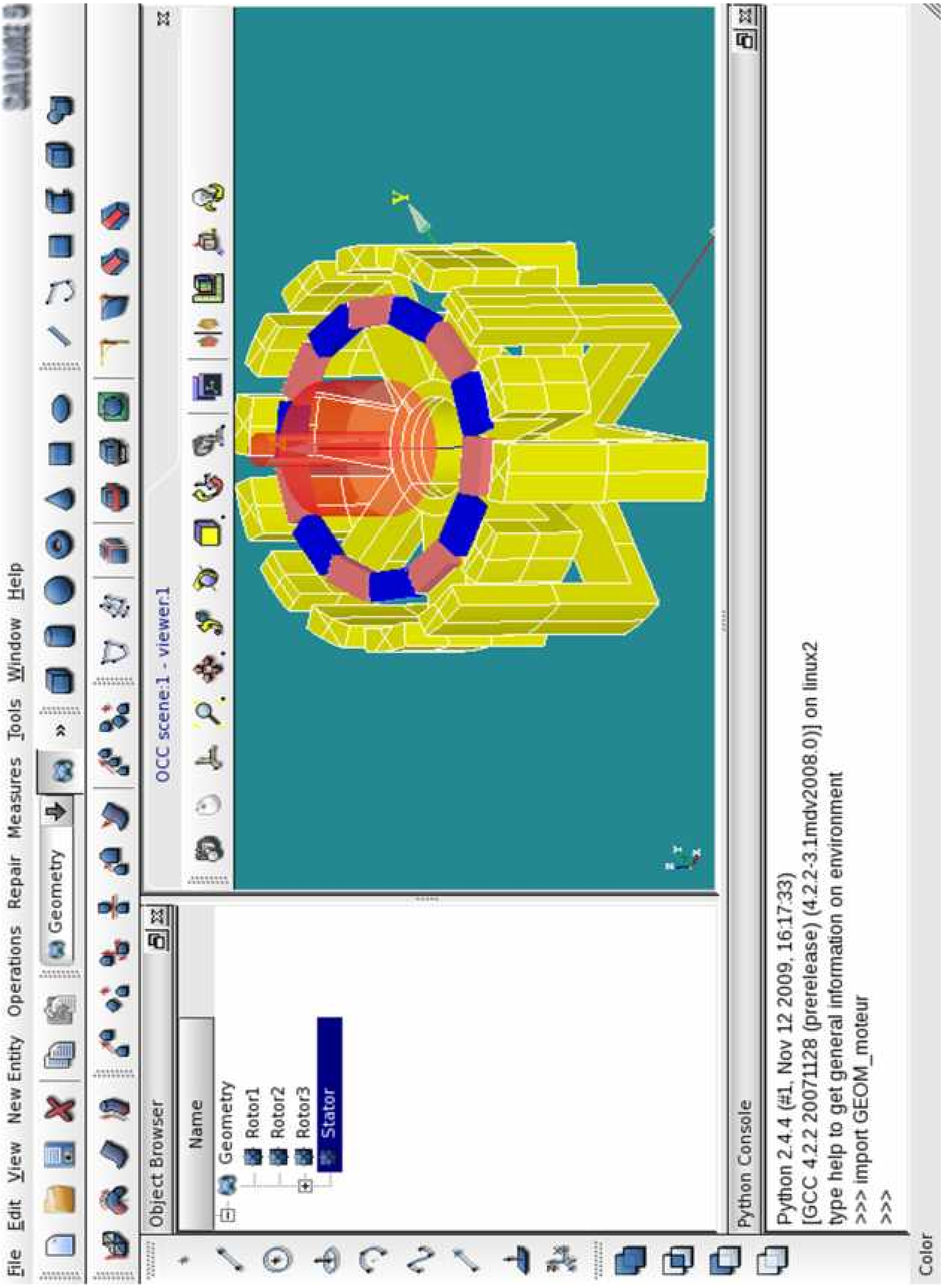
 - Select Button

- Radio Button

 - Add Hypothesis Button

 - Edit Hypothesis Button

## Introduction to Geometry Module



```
Python 2.4.4 (#1, Nov 12 2009, 16:17:33)
[GCC 4.2.2 20071128 (prerelease) (4.2.2-3.1mdv2008.0)] on linux2
type help to get general information on environment
>>> import GEOM_moteur
>>>
```

### Creating geometrical objects

Salome Geometry module provides the toolset allowing to create a vast range of geometrical objects, from points to complex extrusions:

- create Basic objects - points, lines, circles...
- create Primitives - cubes, spheres, cones...
- create Complex objects by extrusion, rotation, interpolation of other objects.
- create and edit Groups of objects of lower dimension, which belong to the objects of higher dimension.
- Build by blocks faces from edges and solids from faces.
- Explode objects of higher dimension into sub-objects of lower dimension.
- create Topological objects - edges, wires, shells.

### Creating Basic Geometric Objects

New Entity -> Basic submenu gives access to creation of such geometrical objects as:

- Point
- Line
- Circle
- Ellipse
- Arc
- Curve
- Isoline
- 2D Sketcher
- 3D Sketcher
- 2D Polyline
- Surface From Face
- Vector
- Plane
- Local Coordinate System

### Point

To create a Point in the Main Menu select New Entity - > Basic - > Point

There are five algorithms to create a Point in the 3D space.

Each time the Result of the operation will be a GEOM_Object (vertex).

I. Firstly, we can define a point by setting its X, Y and Z Coordinates.

II. Secondly, we can define a point by a Reference to another point and the shift of the coordinates of the new point regarding the coordinates of the old one.

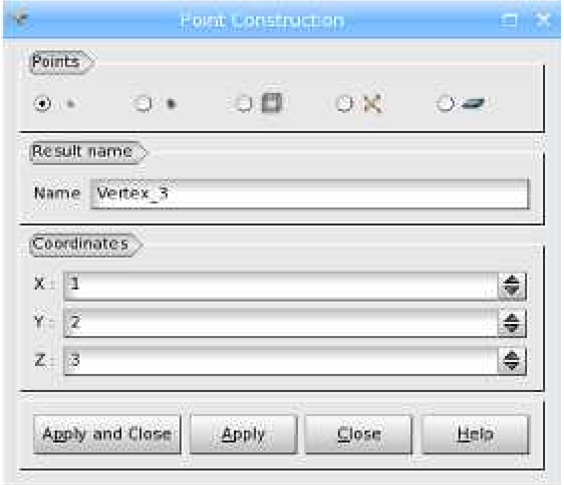
III. Thirdly, we can define a point by an Edge and a Parameter indicating its position on the Edge, ranging from 0.0 to 1.0. For example, 0.5 means that the point is located in the middle of the edge.

IV. Fourthly, we can define a point by intersection of two Lines or Wires (or a Wire and a Line). If they intersect only once, a point will be created. If there are several intersections, a compound of points will be created.

V. Finally, we can define a point located on a Face. The position of the point on it can be defined in one of two ways:
    - by two Parameters: U and V, ranging from 0.0 to 1.0. For example, (0.5; 0.5) means that the point is located in the middle of the face.
    - by three coordinates of the point projected on the given face.

## Creating Basic Geometric Objects

### Point Creation:

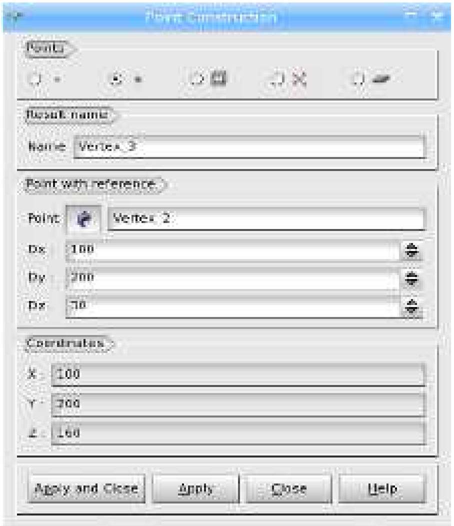


Name: Vertex_3

X: 1

Y: 2

Z: 3



Name: Vertex_3

Point: Vertex_2

Dx: 100

Dy: 200

Dz: 30

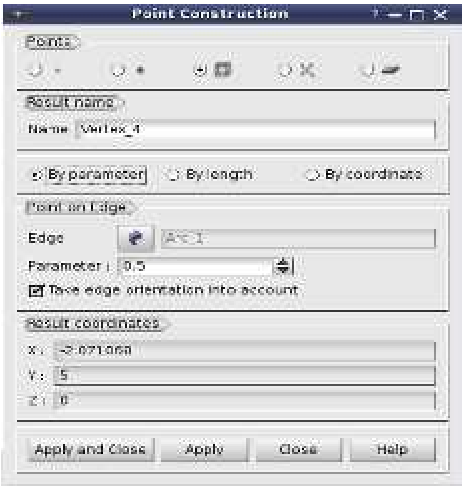
X: 100

Y: 200

Z: 160

Algorithm I

Algorithm II



Name: Vertex_4

● By parameter ○ By length ○ By coordinate

Edge: Arc_1

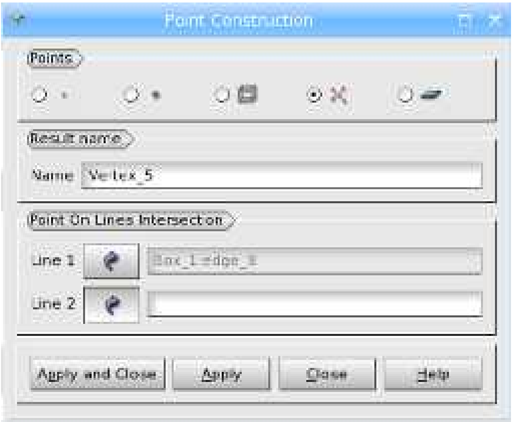
Parameter : 0.5

☑ Take edge orientation into account

X : -2.071068

Y : 5

Z : 0



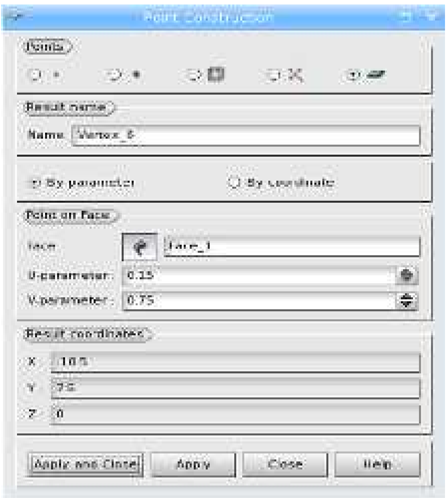
Name: Vertex_5

Line 1: Box_1:edge_3

Line 2:

Algorithm III

Algorithm IV



Name: Vertex_6

● By parameter ○ By coordinate

Face: Face_1

U-parameter: 0.15

V-parameter: 0.75

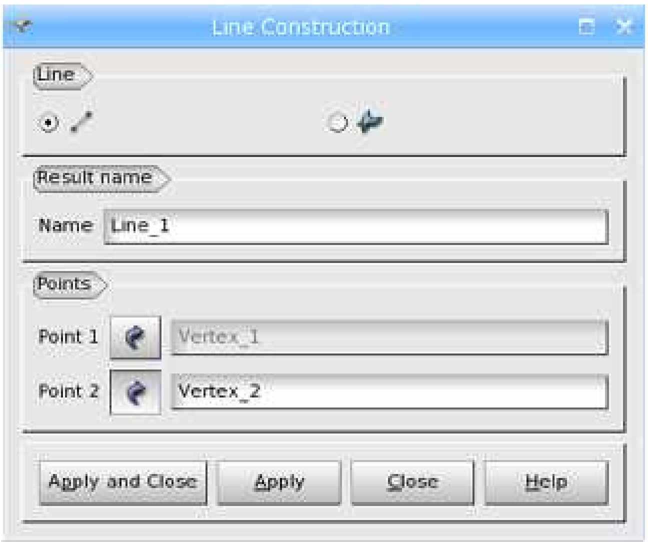
X: 105

Y: 75

Z: 0

Algorithm V

## Line Creation:

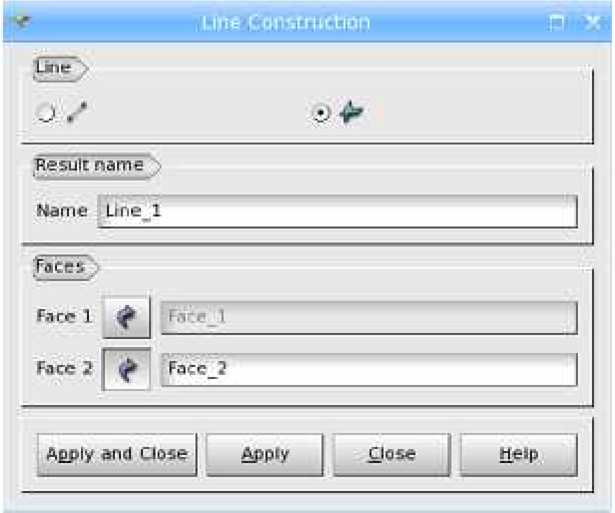


Name: Line_1

Point 1: Vertex_1

Point 2: Vertex_2

Algorithm I



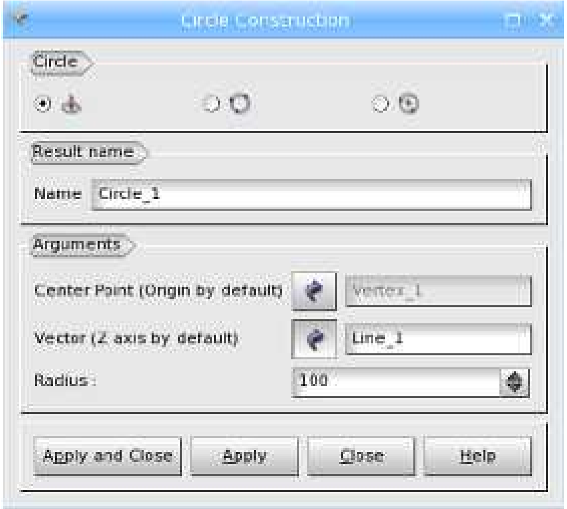
Name: Line_1

Face 1: Face_1

Face 2: Face_2

Algorithm II

## Circle Creation:



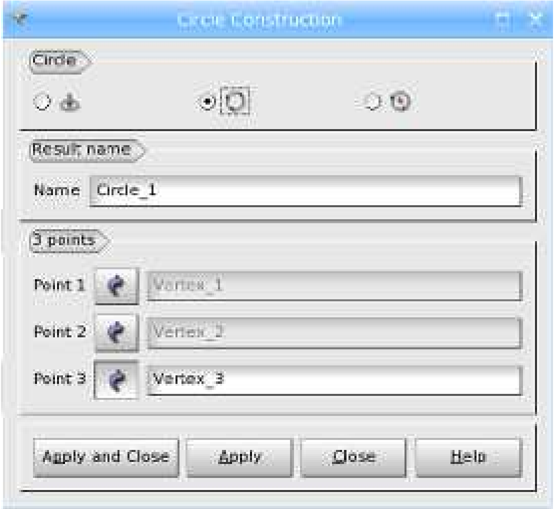
Name: Circle_1

Center Point (Origin by default): Vertex_1

Vector (Z axis by default): Line_1

Radius : 100

Algorithm I



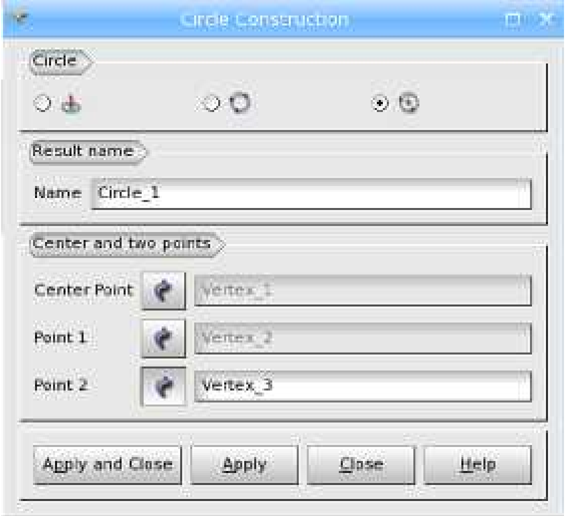
Name: Circle_1

Point 1: Vertex_1

Point 2: Vertex_2

Point 3: Vertex_3

Algorithm II



Name: Circle_1

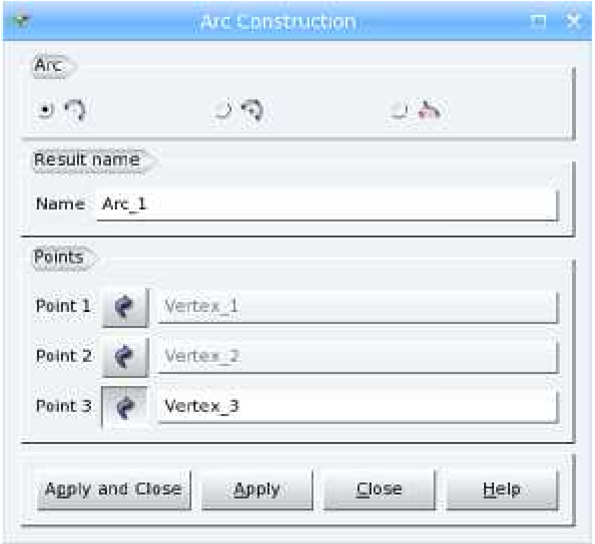
Center Point: Vertex_1

Point 1: Vertex_2

Point 2: Vertex_3

Algorithm III

## Arc



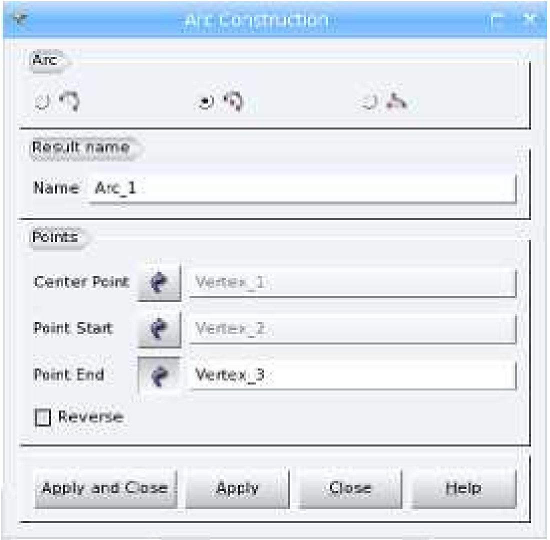
Name Arc_1

Point 1 Vertex_1

Point 2 Vertex_2

Point 3 Vertex_3

Algorithm I



Name Arc_1

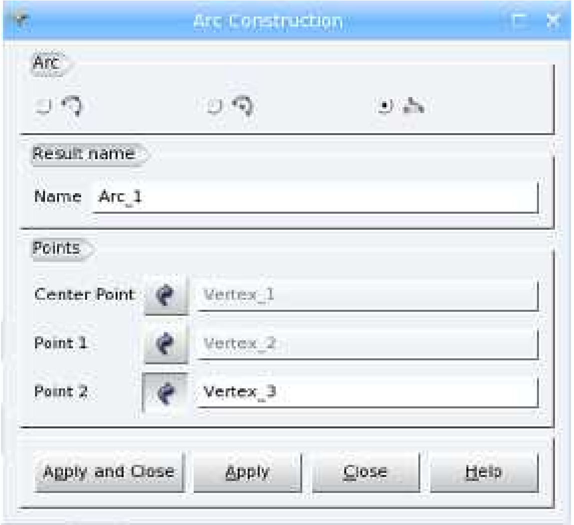
Center Point Vertex_1

Point Start Vertex_2

Point End Vertex_3

□ Reverse

Algorithm II



Name Arc_1

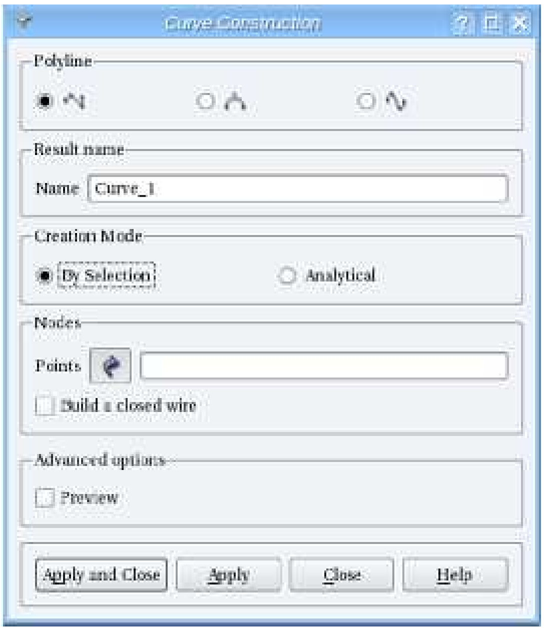
Center Point Vertex_1

Point 1 Vertex_2

Point 2 Vertex_3

Algorithm III

**Curve**



Polyline

● ○ ○

Result name

Name Curve_1

Creation Mode

● By Selection ○ Analytical

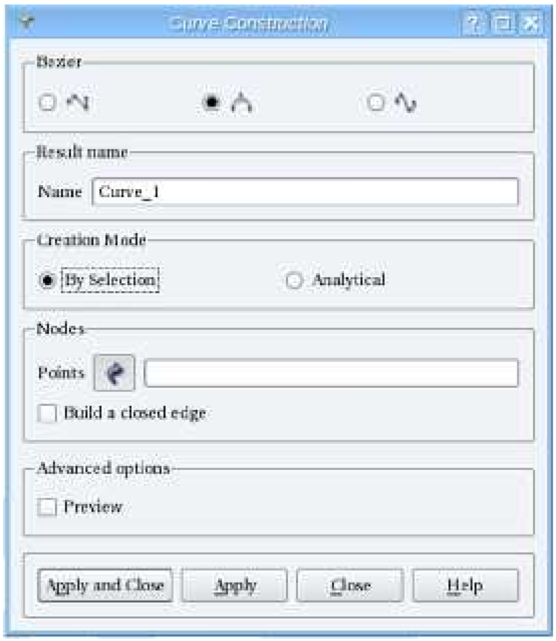
Nodes

Points

□ Build a closed wire

Advanced options

□ Preview



Bezier

○ ● ○

Result name

Name Curve_1

Creation Mode

● By Selection ○ Analytical

Nodes

Points

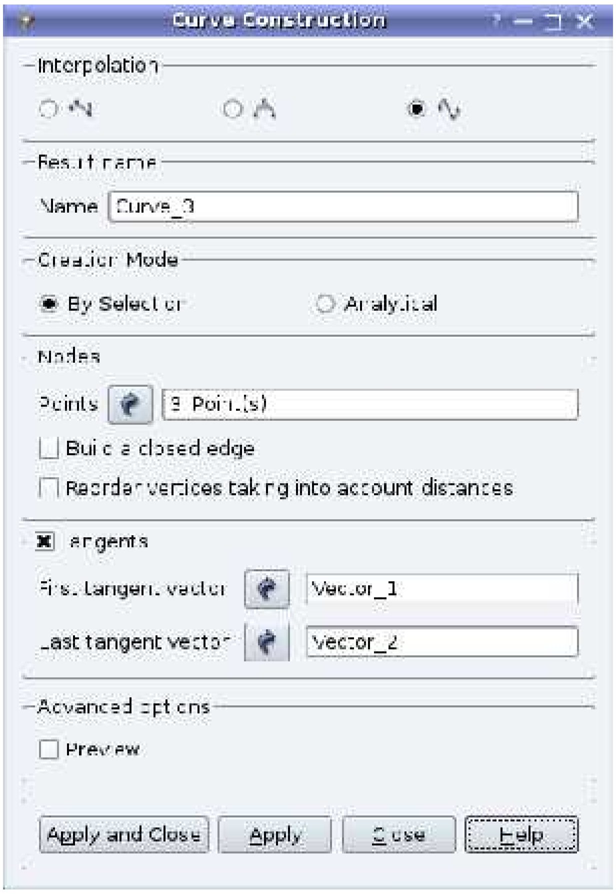
□ Build a closed edge

Advanced options

□ Preview

Polyline or polygonal chain

Bezier curve



Interpolation

○ ○ ●

Result name

Name Curve_3

Creation Mode

● By Selection ○ Analytical

Nodes

Points 3 Point(s)

□ Build a closed edge

□ Reorder vertices taking into account distances

☒ Tangents

First tangent vector Vector_1

Last tangent vector Vector_2

Advanced options

□ Preview

B-spline curve

**Line**

To create a Line in the Main Menu select New Entity - >Basic - > Line

There are 2 algorithms to create a Line in the 3D space. The Result of each operation will be a GEOM_Object (edge).

Firstly you can define a Line by Point1 and Point2, which are the points through which the Line passes.

Secondly you can define a Line as intersection of Plane1 and Plane2.

**Circle**

To create a Circle in the Main Menu select New Entity - > Basic - > Circle

There are 3 algorithms to create a Circle in the 3D space.

The Result of each operation will be a GEOM_Object (edge).

Firstly, you can define a Circle by a Center Point, a Vector giving the circle's normal and a Radius.

Secondly, you can define a Circle by three Points that lie on it.

Finally, you can define a circle by a Center Point and two Points.

**Arc**

To create an Arc in the Main Menu select New Entity - > Basic - > Arc

There are 2 algorithms to create an Arc in the 3D space.

The Result of each operation will be a GEOM_Object (edge).

Firstly, you can define an Arc by three Points that lie on it, where Point1 is the starting point, Point2 is a middle point of the arc and Point3 is the ending point of the arc.

Secondly, you can define an Arc by Center, Start and End points. The arc is built from the Start point to the End point. The radius of the arc is defined by the distance between the Center point and the Start point. The End point defines the angle of the arc. If the distance between the Center point and the End point is not equal to the radius, the arc will be built anyway.

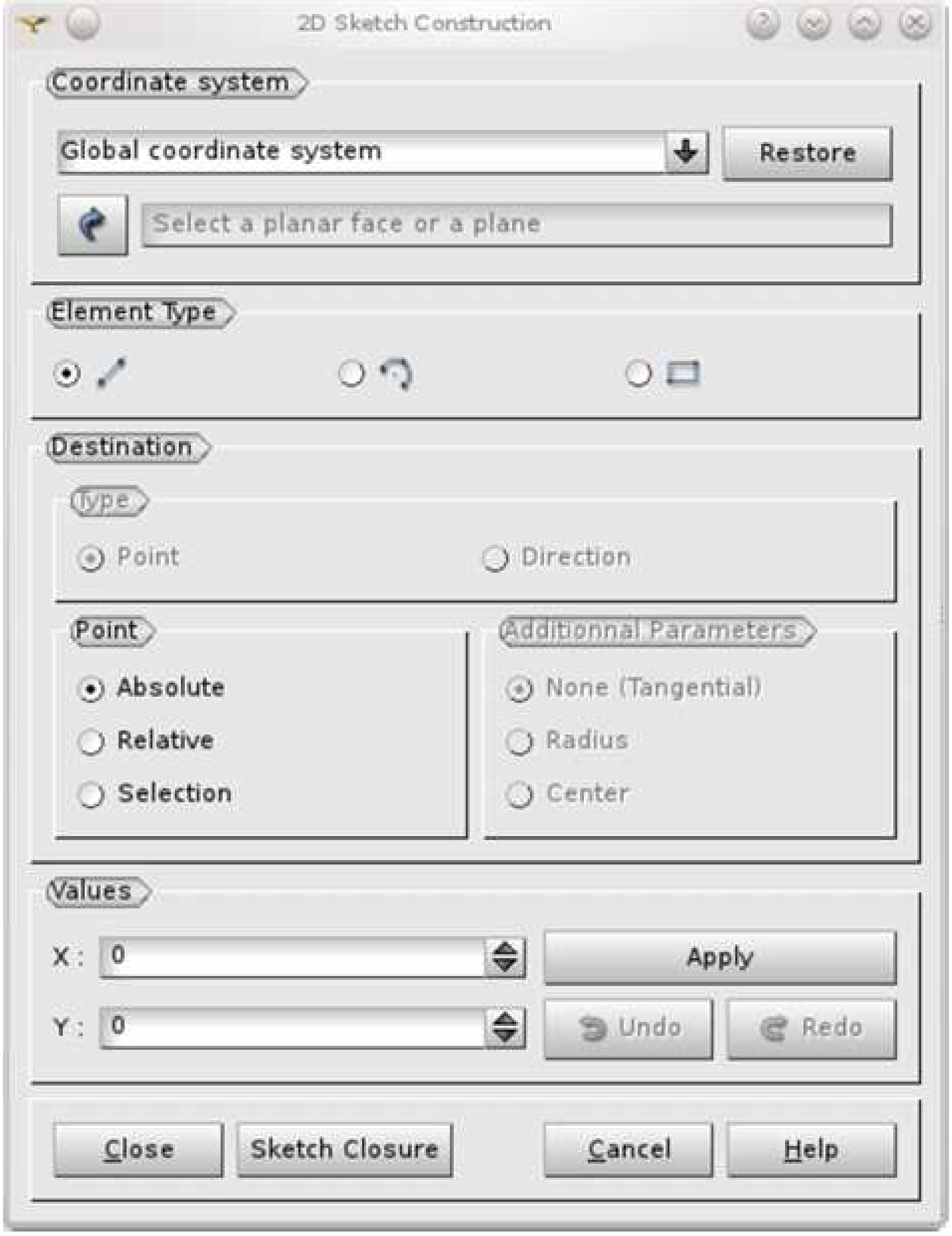
Finally, it is possible to create an Arc of Ellipse, which is also done by three points: the Arguments Point 1, Point 2 and the Center Point. The arc is built on the Ellipse that lies in the plane defined by the three points, with the center in the Center point. The major radius of the Ellipse is defined by the distance between the Center Point and the most remote Argument point. The minor radius is defined by the shortest distance between the other Argument point and the major radius.

**Curve**

To create a Curve in the Main Menu select New Entity - > Basic - > Curve

There are three Curve Construction menu choices corresponding to three possible types of curves: Polyline, Besier or B-spline (Interpolated). The curve is defined by a list of Points through which it passes.

**2D Sketcher**



2D Sketch Construction

Coordinate system

Global coordinate system

Select a planar face or a plane

Element Type

● ╱ ○ ○

Destination

Type

● Point ○ Direction

Point

● Absolute

○ Relative

○ Selection

Additionnal Parameters

● None (Tangential)

○ Radius

○ Center

Values

X : 0

Y : 0

Polyline or polygonal chain is a connected series of line segments. It can be defined by the following parameters:

- *Points* at least 2 points which will serve as nodes on the curve.
- *Build a closed wire* checkbox allows creating the curve as closed wire.

Bezier curve is a curve completely contained in a convex hull of its control points. It can be defined by the follow

- *Points* at least 2 points used to approximate the curve.
- *Build a closed edge* checkbox allows creating the curve as closed edge.ing parameters:

B-spline is a union of curve segments defined at each node span. It can be defined by the following parameters:

- *Points* at least 2 points which will serve as nodes on the curve.
- *Build a closed edge* checkbox allows creating the curve as closed edge.
- If *Reorder vertices taking into account distances* is checked, the interpolation algorithm does not follow the order of vertices but searches for the closest vertex.
- *Tangents* are two vectors, defining the direction at the ends of the curve. It is necessary to set both vectors or none. This option is available only if both above check boxes are not checked.

**2D Sketcher**

The 2D Sketcher allows drawing arbitrary 2D shapes.

To create a 2D Sketch select in the main menu New Entity -> Basic -> 2D Sketch.

At first it is necessary to define the reference coordinate system, where the sketch will be created:

- The XOY plane of the Global coordinate system is suggested by default (if no plane or planar face is selected when 2D Sketcher is activated).
- If any Local Coordinate Systems have been created in the study earlier, they can be selected as a reference coordinate system in the corresponding list box.
- The Arrow button allows selecting a plane or a planar face in the viewer or in the object browser as the reference coordinate system for the sketch. The start point will be set in the geometric center of the face.
- Restore button orientates the viewer correspondingly to the chosen working plane and fits the scene to show all objects.

It is possible to create sketches of two types:

- Profiles made of line segments and arcs, or
- Rectangles.

**Plane**

To create a Plane in the Main Menu select New Entity - > Basic - > Plane

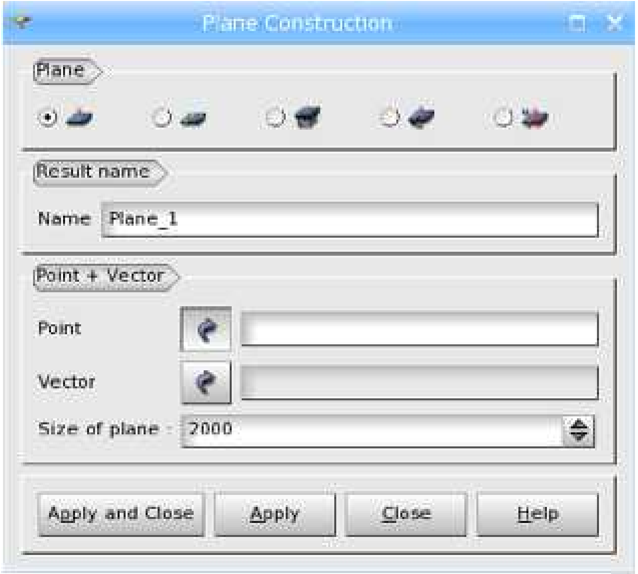
There are five algorithms to create a plane in the 3D space.

The Result of each operation will be a GEOM_Object (face).

Firstly, you can define a Plane by a Point through which the plane passes, a Vector giving a normal of the plane and a Size of the Plane (size of a side of quadrangle face, representing the plane).

Secondly, you can define a Plane by three Points through which the plane passes and a Size of the Plane.

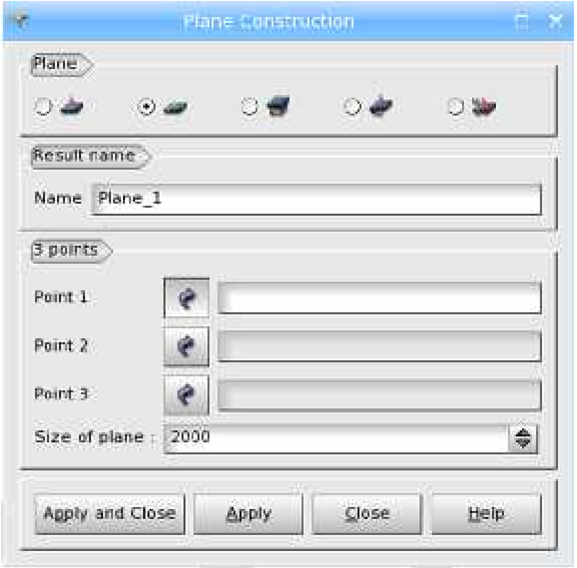
**Plane Creation:**



Name: Plane_1

Size of plane : 2000

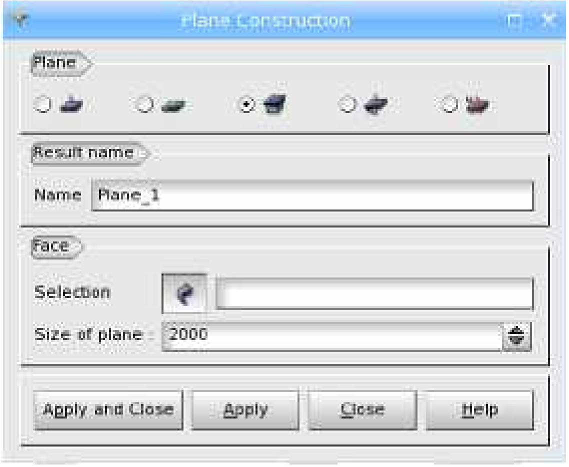
Algorithm I



Name: Plane_1

Size of plane : 2000

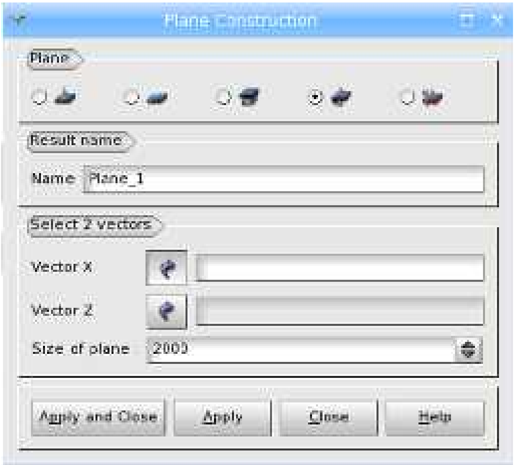
Algorithm II



Name: Plane_1

Size of plane : 2000

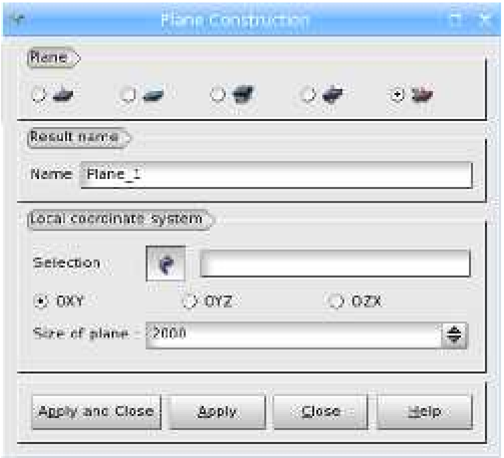
Algorithm III



Name: Plane_1

Size of plane : 2000

Algorithm IV



Name: Plane_1

● OXY ○ OYZ ○ OZX

Size of plane : 2000

Algorithm V

Thirdly, you can define a Plane by another Plane or Face and a Size of the Plane.

Fourthly, you can define a Plane by two Vectors. The first vector defines the center point and the direction, the second vector defines the normal to the Plane.

Finally, you can define a Plane by the Local Coordinate System and the orientation (OXY, OYZ, OZX).

**Creating Primitives**

New Entity -> Primitives submenu allows to create volumical geometrical objects, such as:

- Box
- Cylinder
- Sphere
- Torus
- Cone
- Rectangle
- Disk
- PipeTShape

Creating Complex Objects

New entity -> Generation sub-menu allows creating new geometric objects by modification and multiplication of the existing objects:

- Extrude a 2D object by vector, creating a 3D object.
- Revolve an object around an axis, creating a body of revolution.
- Create a surface from a set of edges.
- Extrude an object along a path, creating a more complex trajectory object.
- Restore Path of a pipe-like shape.
- Add thickness to objects.
- Generate Groups for certain generation operations.

New entity -> Advanced sub-menu allows creating new geometric objects using advanced algorithms:

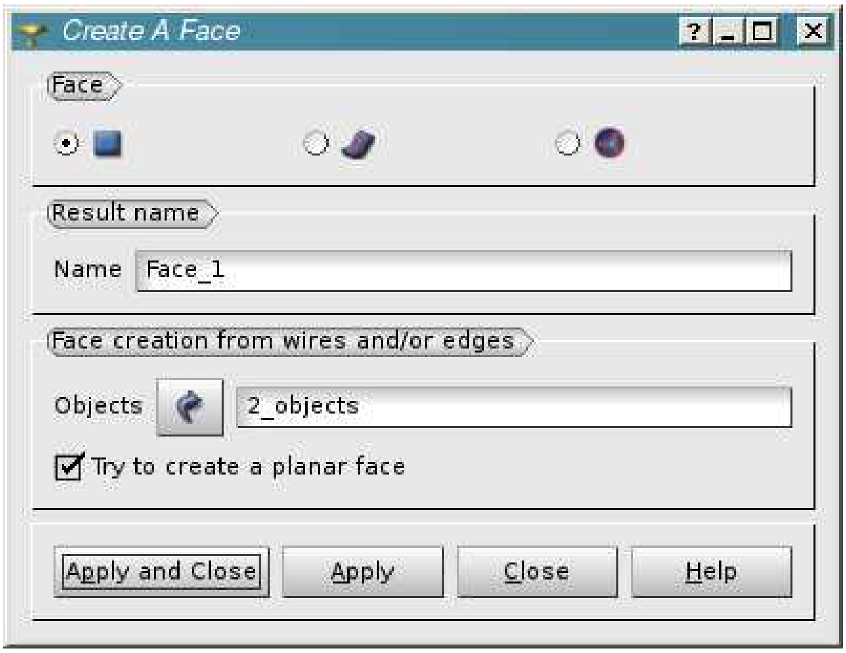
- Smoothing surface from a cloud of points.

**Creating Topological Objects**

**New Entity -> Build** submenu allows to create topological entities:

- Edge
- Wire
- Face
- Shell
- Solid
- Compound

**Build a Face**

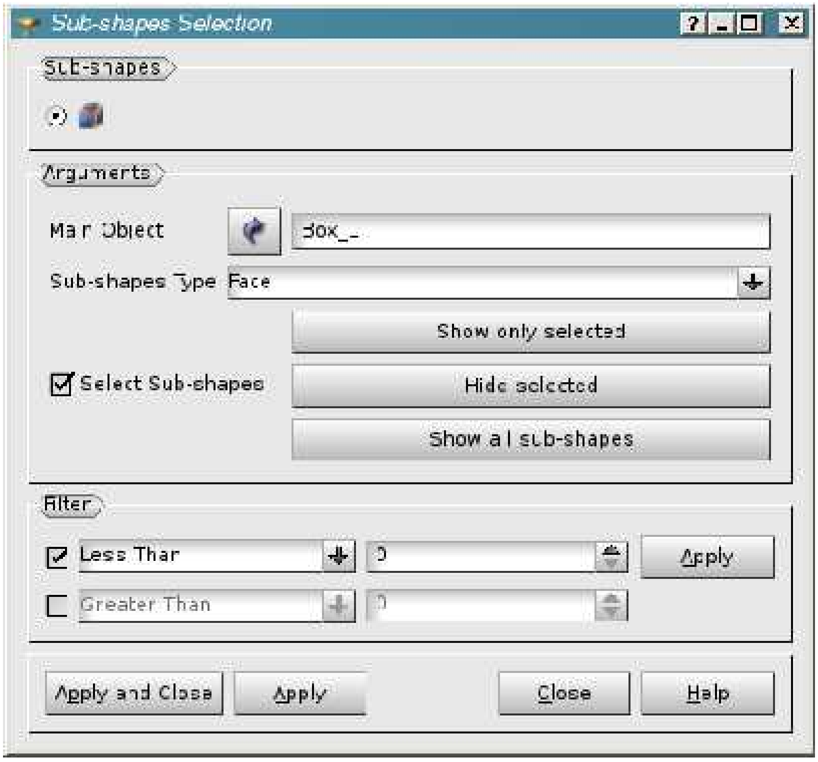


Name: Face_1

Objects: 2_objects

☑ Try to create a planar face

**Explode**



Main Object: Box_1

Sub-shapes Type: Face

☑ Select Sub-shapes

☑ Less Than 0

□ Greater Than 0

## Build a Face

To create a Face in the Main Menu select New Entity - > Build - > Face

There are three algorithms to create a Face. In all cases the Result of the operation will be a GEOM_Object (FACE).

Firstly, to create a Face you need to select input shape(s). The list of input shapes can include shapes of any type except vertices; if the shapes are neither wires nor edges, the algorithm extracts all edges from the input shapes and works on the obtained edges.

The edges and wires do not necessarily have to be closed, the algorithm automatically builds a wire of maximum length from all given edges and wires. If several closed wires are detected the algorithm tries to create a face with holes. It is possible only if there is only one wire that can be interpreted as an outer one; other wires can be considered as inner ones.

Check Try to create a planar face to create a planar face or nothing if it is impossible.

## Explode

To Explode an object into sub-shapes, in the Main Menu select New Entity > Explode. This operation opens the Sub Shapes Selection dialog box.

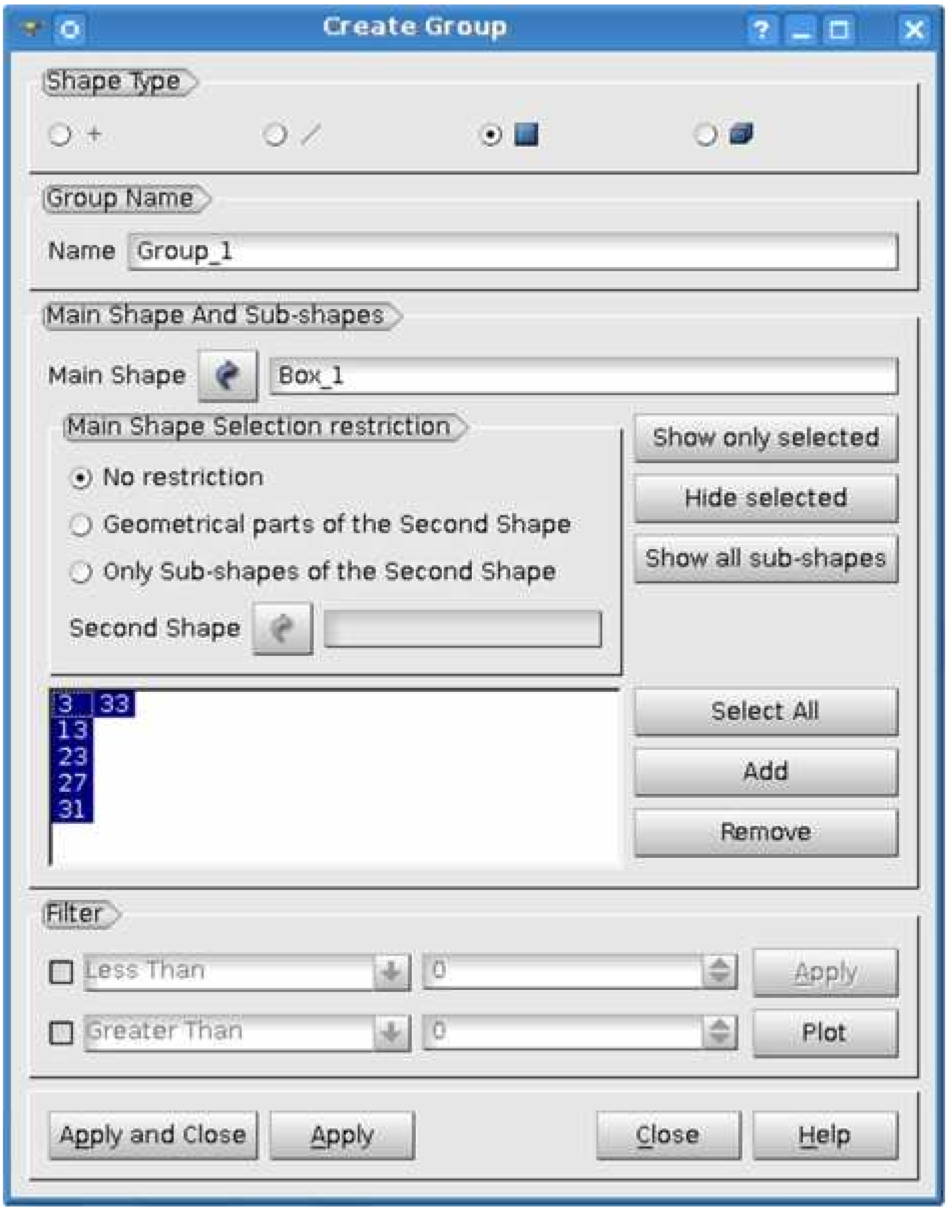
To create a list of sub-shapes (vertices, edges, wires etc.) of the given shape using the Explode operation, you need to define the Main Object, which will be exploded and the Type of Sub-shapes you wish to obtain from it.

The Result of the operation will be a List of GEOM_Objects (vertexes, edges, wires, faces, shells or solids).

The choices available in the Sub Shapes Type combo box depend on the type of selected Main Object:

- **Compound:** to extract compounds;
- **Compsolid:** to extract compsolids;
- **Solid:** to extract solids;
- **Shell:** to extract shells;
- **Face:** to extract faces;
- **Wire:** to extract wires;
- **Edge:** to extract edges;
- **Vertex:** to extract vertices;
- **Shape:** to extract top-level contents of the compound shape;
- **Flat:** to extract "flat" contents of the compound shape.

**Create a group**



Create Group

Shape Type

○ + ○ ⁄ ● ▪ ○ ▪

Group Name

Name: Group_1

Main Shape And Sub-shapes

Main Shape: Box_1

Main Shape Selection restriction

● No restriction

○ Geometrical parts of the Second Shape

○ Only Sub-shapes of the Second Shape

Second Shape:

3 33
13
23
27
31

Filter

□ Less Than 0

□ Greater Than 0

### Working with groups

Creation and editing groups of sub-shapes of a geometrical object makes handling sub-shapes much easier. Boolean operations on groups are also available.

- Create a Group
- Edit a Group
- Union Groups
- Intersect Groups
- Cut Groups

### Create a group

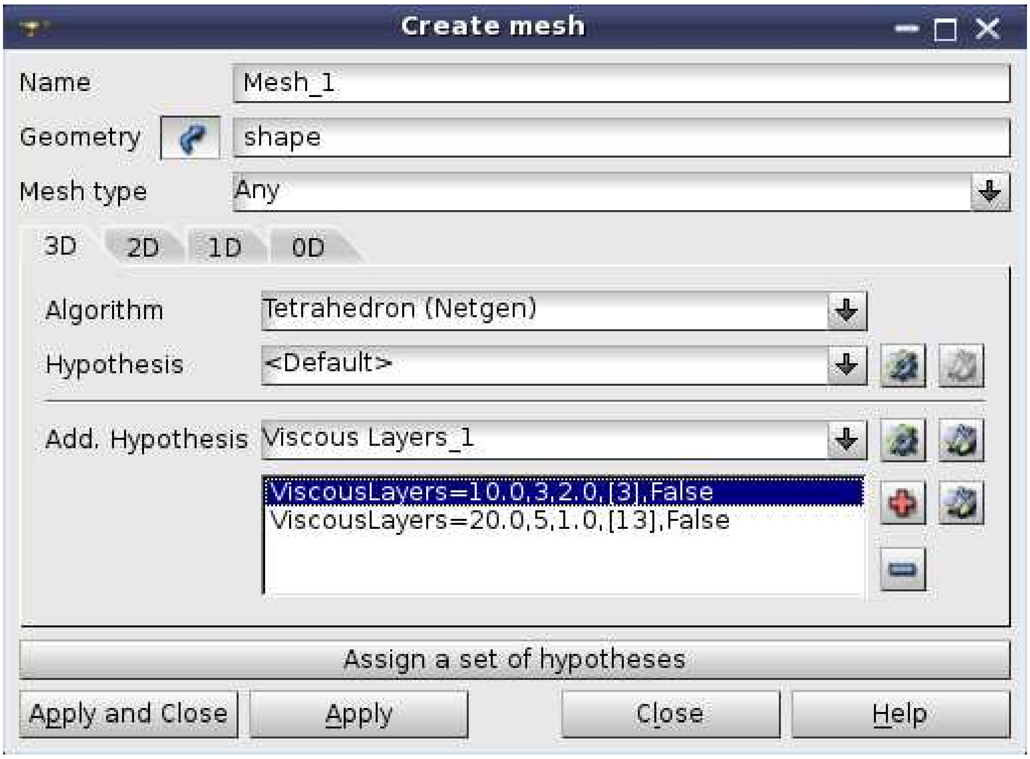
To create a group of sub-shapes of a geometrical object in the main menu select New entity > Group > Create .

The following dialog box will appear:

In this Menu:

- Shape Type radio button defines the type of elements for the group (points, wires, faces, shells or solids).
- Group Name - allows defining the name of the group, by default, it is Group_n.
- Then, using the "Select" button, select the Main Shape (a geometrical object on which the group will be created).
- Main Shape Selection restriction button group allows limiting the range of available group elements,
    - No restriction button enables all sub-shapes of the Main Shape.
    - Geometrical parts of the Second Shape restricts the range of accessible elements to the sub-shapes of the Main Shape that geometrically coincide with the Second Shape.
    - Only Sub-shapes of the Second Shape restricts the range of accessible elements to the sub-shapes of the Main Shape that are sub-shapes of the Second Shape. This is useful because sometimes purely geometrical coincidence is not enough and it is necessary to work with shapes, which actually belong both to the main and the second shape.
- You can selectively display the selected elements using the following buttons:
    - Show only selected - displays only the sub-shapes selected in the list box.
    - Hide selected - hides the sub-shapes selected in the list box.
    - Show all sub-shapes - displays only the sub-shapes of the Main Shape.
- You can select the elements of your group in several ways:
    - You can select them manually in the 3D Viewer, and add to the group by clicking the Add button (keep down the Shift button on the keyboard to select several elements and add all them together). The indexes of the selected elements will be displayed in the list. To delete elements from the list, select them and click Remove button.
    - Clicking Select All button you can add all object's elements of a certain type in the list of the elements of the group. If the Second Shape is used, the elements are added according to Main Shape Selection restriction settings. To delete elements from the list, select them and click Remove button.
    - Filtering out some entities according to the specified threshold value or values (see below).
- Finally, confirm your selection by clicking Apply and Close (also closes the Menu) or Apply (leaves the Menu open for creation of other groups), or skip it by clicking Close button.

**Create mesh dialog box**



Create mesh

Name: Mesh_1

Geometry: shape

Mesh type: Any

Algorithm: Tetrahedron (Netgen)

Hypothesis: <Default>

Add. Hypothesis: Viscous Layers_1

ViscousLayers=10.0,3,2.0,[3],False
ViscousLayers=20.0,5,1.0,[13],False

**Introduction to MESH**

Mesh module of SALOME is destined for:

- creating meshes in different ways:
  - by meshing geometrical models previously created or imported by the Geometry component;
  - bottom-up, using mesh edition operations, especially extrusion and revolution;
  - by generation of the 3D mesh from the 2D mesh (the latter can be either imported or manually created);
- importing and exporting meshes in various formats;
- modifying meshes with a vast array of dedicated operations;
- creating groups of mesh elements;
- filtering mesh entities (nodes or elements) using Filters functionality for creating groups and applying mesh modifications;
- viewing meshes in the VTK viewer and getting info on mesh and its sub-objects;
- applying to meshes Quality Controls, allowing to highlight important elements;
- Taking various measurements of the mesh objects.

**About meshes**

MESH represents a discrete approximation of a subset of the three-dimensional space by elementary geometrical elements.

Mesh module provides several ways to create the mesh:

- The main way is to construct the mesh on the basis of the geometrical shape produced in the Geometry module. This way implies selection of
  - a geometrical object ( main shape) and
  - meshing parameters (meshing algorithms and characteristics (e.g. element size) of a required mesh encapsulated in hypothesis objects).

Construction of sub-meshes allows to discretize some sub-shapes of the main shape, for example a face, using the meshing parameters that differ from those for other sub-shapes.

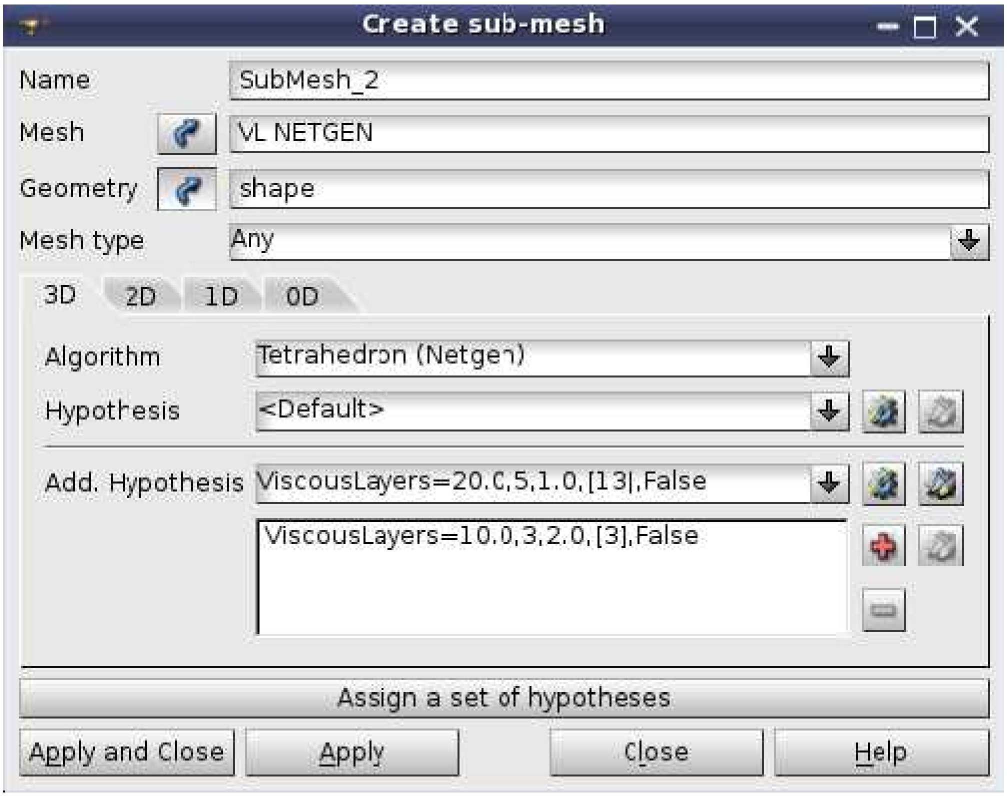
Meshing parameters of meshes and sub-meshes can be edited.

The structure of a SALOME mesh is described by nodes and elements based on these nodes.

The mesh can include the following entities:

- Node — a mesh entity defining a position in 3D space with coordinates (x, y, z).
- Edge (or segment) — 1D mesh element linking two nodes.
- Face — 2D mesh element representing a part of surface bound by links between face nodes. A face can be a triangle, quadrangle or polygon.
- Volume — 3D mesh element representing a part of 3D space bound by volume facets. Nodes of a volume describing each facet are defined by the MED connectivity convention. A volume can be a tetrahedron, hexahedron, pentahedron, pyramid, hexagonal prism or polyhedron.
- 0D element — mesh element defined by one node.
- Ball element — discrete mesh element defined by a node and a diameter.

### Create sub-mesh dialog box



Name: SubMesh_2

Mesh: VL NETGEN

Geometry: shape

Mesh type: Any

Algorithm: Tetrahedron (Netgen)

Hypothesis: <Default>

Add. Hypothesis: ViscousLayers=20.0,5,1.0,[13],False

ViscousLayers=10.0,3,2.0,[3],False

### Constructing meshes

To create a mesh on geometry, it is necessary to create a mesh object by choosing

- a geometrical shape produced in the Geometry module (main shape);
- meshing parameters, including
  - meshing algorithms and
  - hypotheses specifying constraints to be taken into account by the chosen meshing algorithms.

Then you can launch mesh generation by invoking Compute command.

Construction of a mesh on a geometry includes at least two (mesh creation and computing) of the following steps:

- Creation of a mesh object, where you can specify meshing parameters to apply to all sub-shapes of the main shape.
- Creation of sub-meshes, (optional) where you can specify meshing parameters to apply to the selected sub-shapes.
- Evaluating mesh size (optional) can be used to know an approximate number of elements before their actual generation.
- Previewing the mesh (optional) can be used to generate mesh of only lower dimension(s) in order to visually estimate it before full mesh generation, which can be much longer.
- Changing sub-mesh priority (optional) can be useful if there are concurrent sub-meshes defined.
- Computing the mesh uses defined meshing parameters to generate mesh elements.
- Editing the mesh (optional) can be used to modify the mesh of a lower dimension before computing elements of an upper dimension.

### Creation of a mesh object

To construct a mesh:

(1) Select a geometrical object for meshing.

(2) In the Mesh menu select Create Mesh or click "Create Mesh" button in the toolbar.

The Create mesh dialog box will appear.

(3) To filter off irrelevant meshing algorithms, you can select Mesh Type in the corresponding list from Any, Hexahedral, Tetrahedral, Triangular and Quadrilateral (there can be less items for the geometry of lower dimensions).

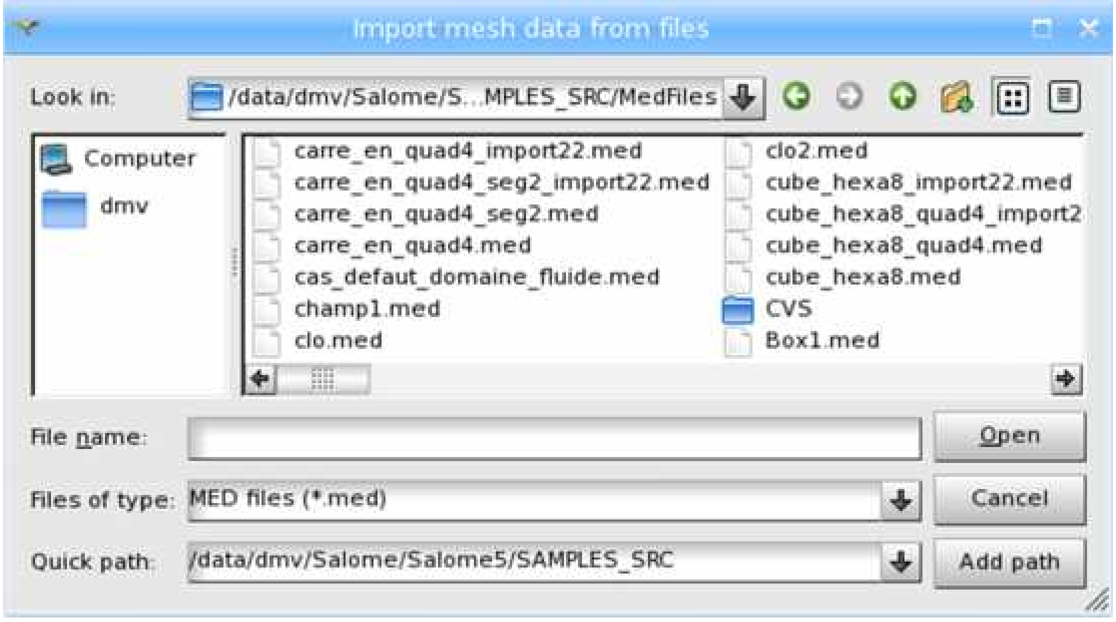
Selection of a mesh type hides all meshing algorithms that cannot generate elements of this type.

(4) Apply **meshing algorithms** and **hypotheses** which will be used to compute this mesh.

"Create mesh" dialog box contains several tab pages titled **3D, 2D, 1D and 0D**. The title of each page reflects the dimension of the sub-shapes the algorithms listed on this page affect and the maximal dimension of elements the algorithms generate. For example, 3D page lists the algorithms that affect 3D sub-shapes (solids) and generate 3D mesh elements (tetrahedra, hexahedra etc.)

As soon as you have selected an algorithm, you can create a hypothesis (or select an already created one). A set of accessible hypotheses includes only the hypotheses that can be used by the selected algorithm.

**Importing mesh**



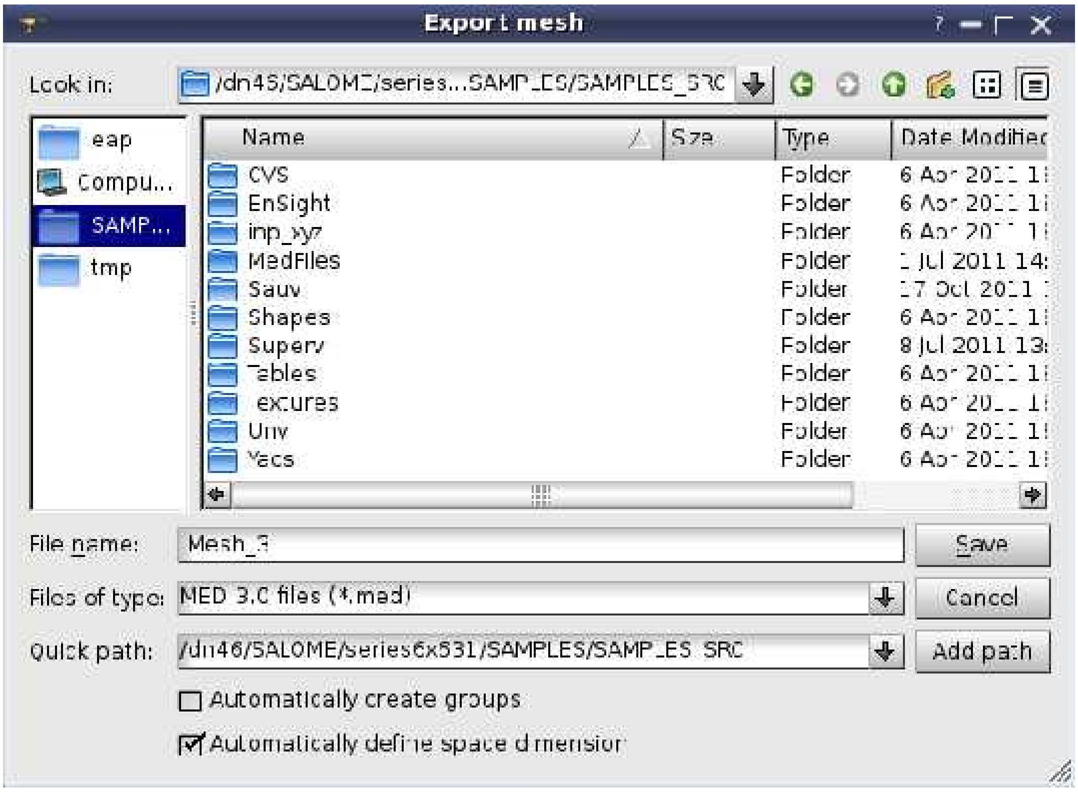
Look in: /data/dmv/Salome/S...MPLES_SRC/MedFiles

carre_en_quad4_import22.med
carre_en_quad4_seg2_import22.med
carre_en_quad4_seg2.med
carre_en_quad4.med
cas_defaut_domaine_fluide.med
champ1.med
clo.med
clo2.med
cube_hexa8_import22.med
cube_hexa8_quad4_import2
cube_hexa8_quad4.med
cube_hexa8.med
CVS
Box1.med

Files of type: MED files (*.med)

Quick path: /data/dmv/Salome/Salome5/SAMPLES_SRC

**Exporting mesh**



Look in: /dn46/SALOME/series...SAMPLES/SAMPLES_SRC

| Name | Size | Type | Date Modified |
|---|---|---|---|
| CVS | | Folder | 6 Apr 2011 1 |
| EnSight | | Folder | 6 Apr 2011 1 |
| inp_xyz | | Folder | 6 Apr 2011 1 |
| MedFiles | | Folder | 1 Jul 2011 14: |
| Sauv | | Folder | 17 Oct 2011 1 |
| Shapes | | Folder | 6 Apr 2011 1 |
| Superv | | Folder | 8 Jul 2011 13: |
| Tables | | Folder | 6 Apr 2011 1 |
| Textures | | Folder | 6 Apr 2011 1 |
| Unv | | Folder | 6 Apr 2011 1 |
| Xacs | | Folder | 6 Apr 2011 1 |

File name: Mesh_3

Files of type: MED 3.0 files (*.med)

Quick path: /dn46/SALOME/series6x631/SAMPLES/SAMPLES_SRC

☐ Automatically create groups

☑ Automatically define space dimension

### Constructing sub-meshes

By purpose, the sub-mesh is an object used to assign to a sub-shape different meshing parameters than those assigned to the main shape.

Structurally, the sub-mesh is a mesh on a certain sub-shape, or a group of sub-shapes, possibly generated using different meshing algorithms and/or hypotheses than those used to generate the mesh on other sub-shapes.

Creation of a sub-mesh allows to control individually meshing of a certain sub-shape, thus to get a locally coarser or finer mesh, to get elements of different types in the same mesh, etc.

A sub-shape to create a sub-mesh on should be retrieved from the main shape in one of the following ways:

- In Geometry module, via New Entity > Explode menu.
- In Geometry module, by creation of a group (New Entity > Group > Create Group menu).
- In Mesh module, by selecting a mesh element generated on a sub-shape of interest. This way is accessible if the mesh is already computed.
- In Mesh module, by clicking Publish Sub-shape button in a dialog showing meshing errors.

Construction of a sub-mesh consists of:

- Selecting a mesh which will encapsulate the sub-mesh
- Selecting a sub-shape for meshing
- Applying one or several hypotheses and meshing algorithms which will be used for discretization of this sub-shape.

To construct a sub-mesh:

From the Mesh menu select Create Sub-mesh or click "Create Sum-mesh" button in the toolbar.

The sub-mesh dialog box will appear:

It allows to define the Name, the parent Mesh and the Geometry (e.g. a face if the parent mesh has been built on box) of the sub-mesh. You can define meshing algorithms and hypotheses in the same way as in Create mesh dialog.

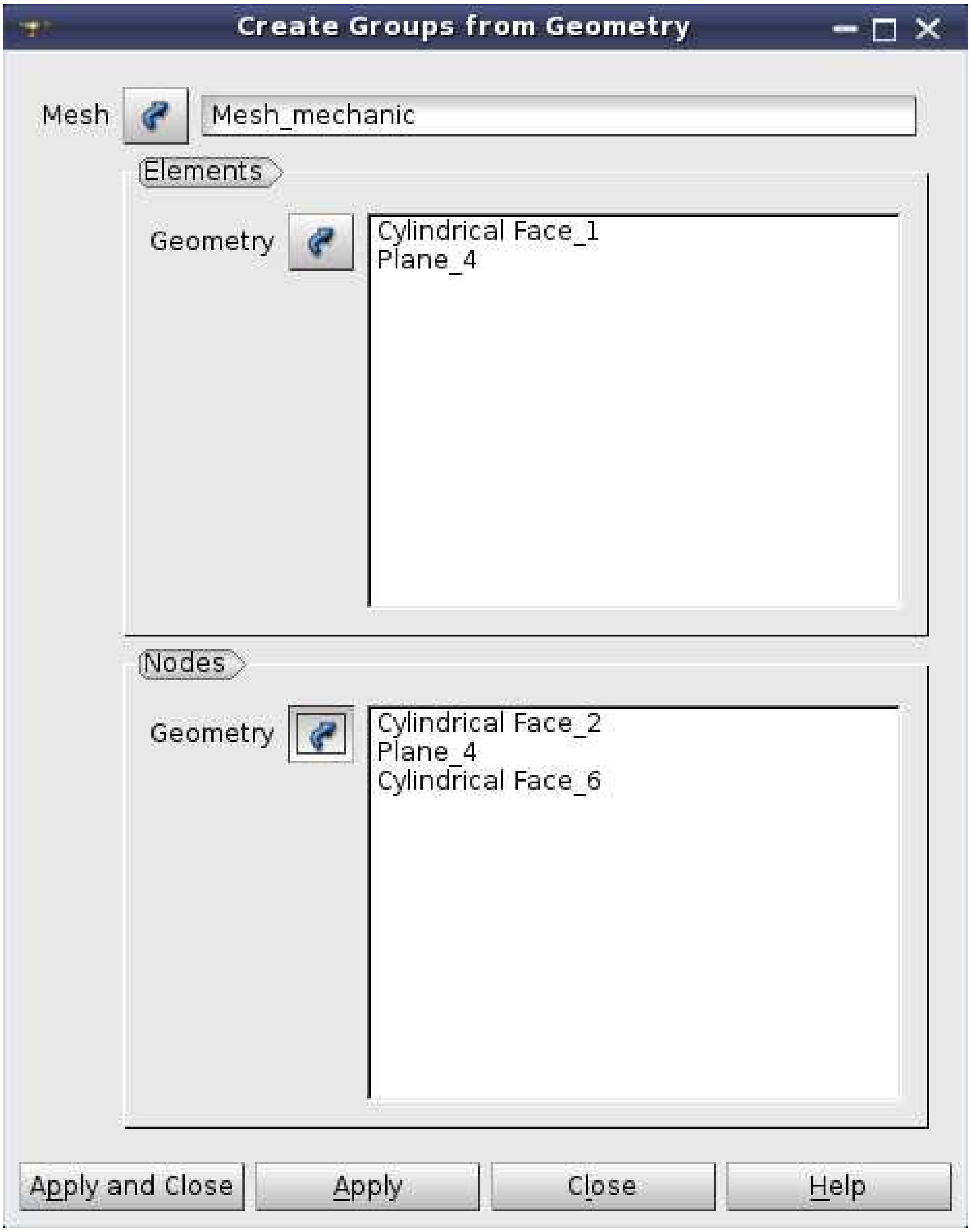
### Importing and exporting meshes

In MESH there is a functionality allowing import/export of meshes from/to MED, UNV (I-DEAS 10), DAT (simple ascii format), STL, GMF (internal format of DISTENE products, namely MG-CADSurf, MG-Tetra and MG-Hexa algorithms) and CGNS format files. You can also export a group as a whole mesh.

To import a mesh:

(1) From the File menu choose the Import item, from its sub-menu select the corresponding format (MED, UNV, STL, GMF and CGNS) of the file containing your mesh.
(2) In the standard Search File dialog box find the file for import. It is possible to select multiple files to be imported all at once.
(3) Click the OK button.

**Create Groups from Geometry**



Mesh: Mesh_mechanic

Elements – Geometry:
Cylindrical Face_1
Plane_4

Nodes – Geometry:
Cylindrical Face_2
Plane_4
Cylindrical Face_6

To export a mesh or a group:

(1) Select the object you wish to export.
(2) From the File menu choose the Export item, from its sub-menu select the format (MED, UNV, DAT, STL, GMF and CGNS) of the file which will contain your exported mesh.
(3) In the standard Search File select a location for the exported file and enter its name.
(4) Click the OK button.

**Create Groups from Geometry**

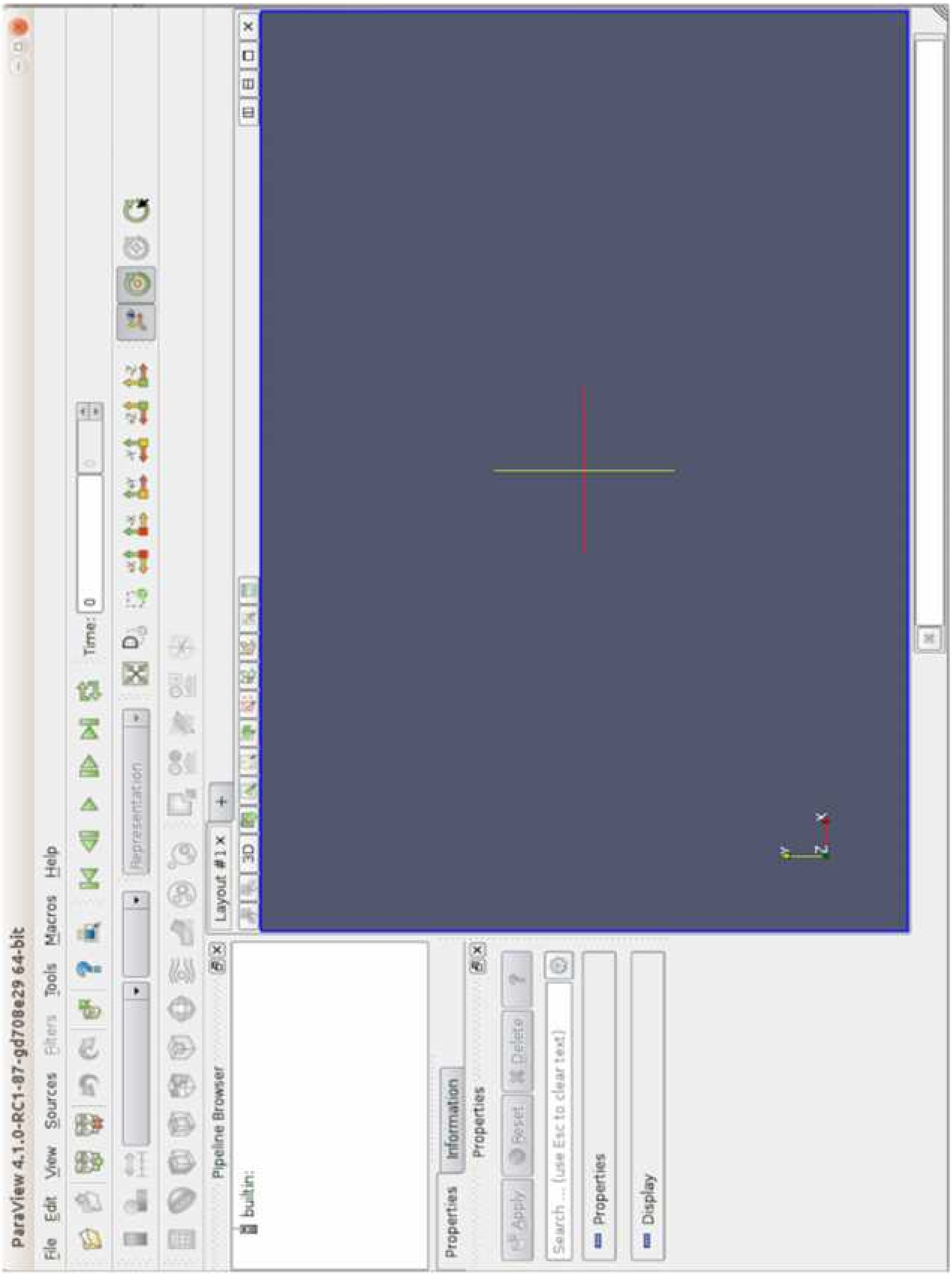
This operation allows creating groups on geometry on all selected shapes. Only the main shape of the mesh and its sub-shapes can be selected.

The type of each new group is defined automatically by the nature of the **Geometry**. The group names will be the same as the names of geometrical objects.

To use this operation, select in the **Mesh** menu or in the contextual menu in the Object browser **Create Groups from Geometry** item.

In this dialog Elements group contains a list of shapes, on which groups of elements will be created; Nodes group contains a list of shapes, on which groups of nodes will be created.

## PARAVIEW GRAPHICAL USER INTERFACE



# PARAVIEW

## Introduction

ParaView is an open-source, multi-platform scientific data analysis and visualization tool that enables analysis and visualization of extremely large datasets. ParaView leverages parallel data processing and rendering to enable interactive visualization for extremely large datasets. It also includes support for large displays including tiled displays and immersive 3D displays with head tracking and wand control capabilities.

ParaView is open-source (BSD licensed, commercial software friendly). As with any successful open-source project, ParaView is supported by an active user and developer community.

## PARAVIEW GRAPHICAL USER INTERFACE

ParaView is the graphical front-end to the ParaView application. The UI is designed to allow you to easily create pipelines for data processing with arbitrary complexity. The UI provides panels for you to inspect and modify the pipelines, to change parameters that in turn affect the processing pipelines, to perform various data selection and inspection actions to introspect the data, and to generate renderings.

The UI is comprised of menus, dockable panels, toolbars, and the viewport – the central portion of the application window.

Menus provide the standard set of options typical with a desktop application including options for opening/saving files (File menu), for undo/redo (Edit menu), for the toggle panel, and for toolbar visibilities (View menu). Additionally, the menus provide ways to create sources that generate test datasets of various types (Sources menu), as well new filters for processing data (Filters menu). The Tools menu provides access to some of the advanced features in ParaView such as managing plugins and accessing the embedded Python shell.

Panels provide you with the ability to peek into the application’s state. For example, you can inspect the visualization pipeline that has been set up (Pipeline Browser), as well as the memory that is being used (Memory Inspector) and the parameters or properties for a processing module (Properties panel). Several of the panels also allow you to change the values that are displayed, e.g., the Properties panel not only shows the processing module parameters, but it also allows you to change them. Several of the panels are context sensitive. For example, the Properties panel changes to show the parameters from the selected module as you change the active module in the Pipeline Browser.

Toolbars are designed to provide quick access to common functionality. Several of the actions in the toolbar are accessible from other locations, including menus or panels. Similar to panels, some of the toolbar buttons are context sensitive and will become enabled or disabled based on the selected module or view.

## Dealing with time

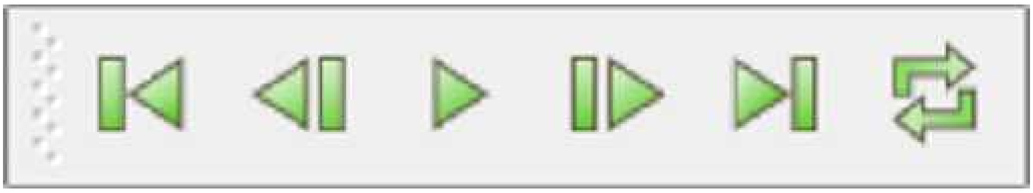
When you open a dataset with time, either as a file series or in a file format that natively supports time, paraview will automatically setup an animation for you so that you can play through each of the time steps in the dataset

by using the ▶ button on the VCR Controls toolbar (Figure). You can change or modify this animation and further customize it.

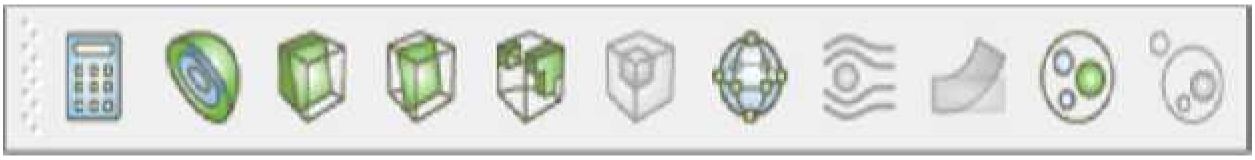
## Filters for sub-setting data

These filters are used for extracting subsets from an input dataset. How this subset is defined and how it is extracted depends on the type of the filter.

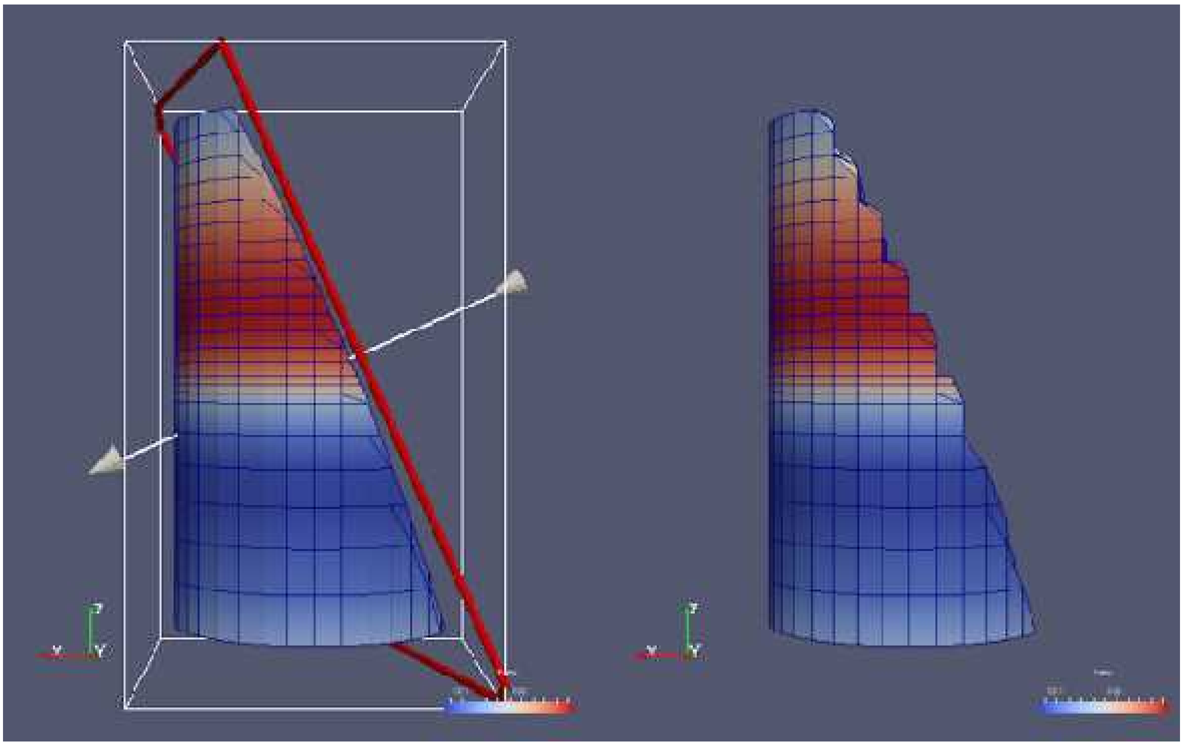
**VCR Controls toolbar for interacting with an animation**



**The Common filters toolbar in paraview for quick access to the commonly used filters**



**Clip**



**Clip**

Clip is used to clip any dataset using either an implicit function (such as a plane, sphere, or a box) or using values of a scalar data array in the input dataset. A scalar array is a point or cell attribute array with a single component. Clipping involves iterating over all cells in the input dataset and then removing those cells that are considered outside of the space defined by the implicit function or that have an attribute values less than the selected value.

The clip image shown is a Comparison between results produced by the Clip filter with Crinkle Clip unchecked (left) and checked (right) when clipping with an implicit plane. The image on the left also shows the 3D widget used to interactively place the implicit plane for the clipping operation.

To create the Clip filter, you can use the Filters > Common or the Filters >Alphabetical menu. This filter is also accessible from the Common filters toolbar. You can click the button to create this filter.

On the Properties panel, you will see the available properties for this filter. One of the first things that you should select is the Clip Type. Clip Type is used to specify the type of implicit function to use for the clipping operations.

**Slice**

The Slice filter slices through the input dataset with an implicit function such as a plane, a sphere, or a box. Since this filter returns data elements along the implicit function boundary, this is a dimensionality reducing filter (except when crinkle slicing is enabled), i.e., if the input dataset has 3D elements like tetrahedrons or hexahedrons, the output will have 2D elements, line triangles, and quads, if any. While slicing through a dataset with 2D elements, the result will be lines.

In paraview, this filter can be created using the button on the Common filters toolbar, besides the Filters menu.

The slice image shown is a Comparison between results produced by the Slice filter when slicing image data with an implicit plane with different options. The lower-left image shows the output produced by the Clip filter when clipping with the same implicit function, for contrast.

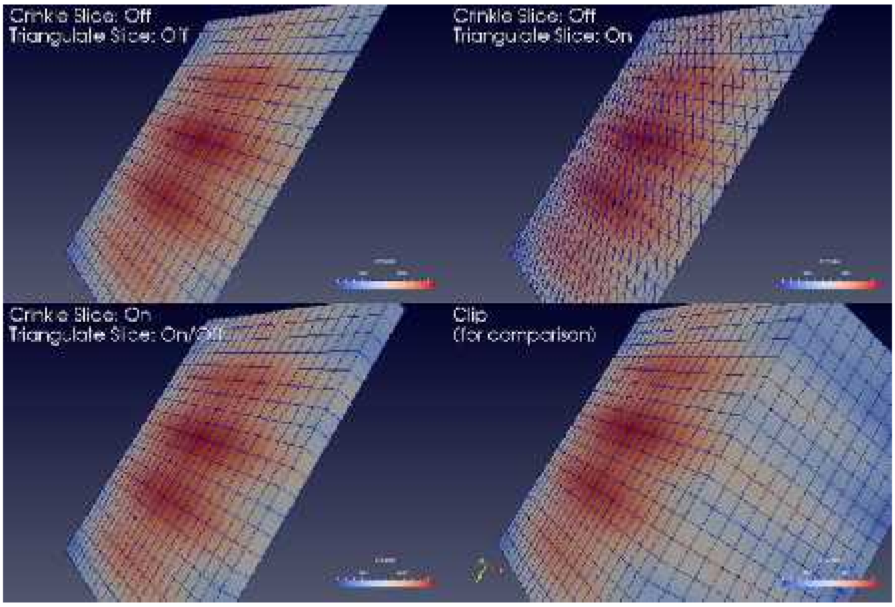
**Glyph**

Glyph is used to place markers or glyphs at point locations in the input dataset. The glyphs can be oriented or scaled based on vector and scalar attributes on those points.

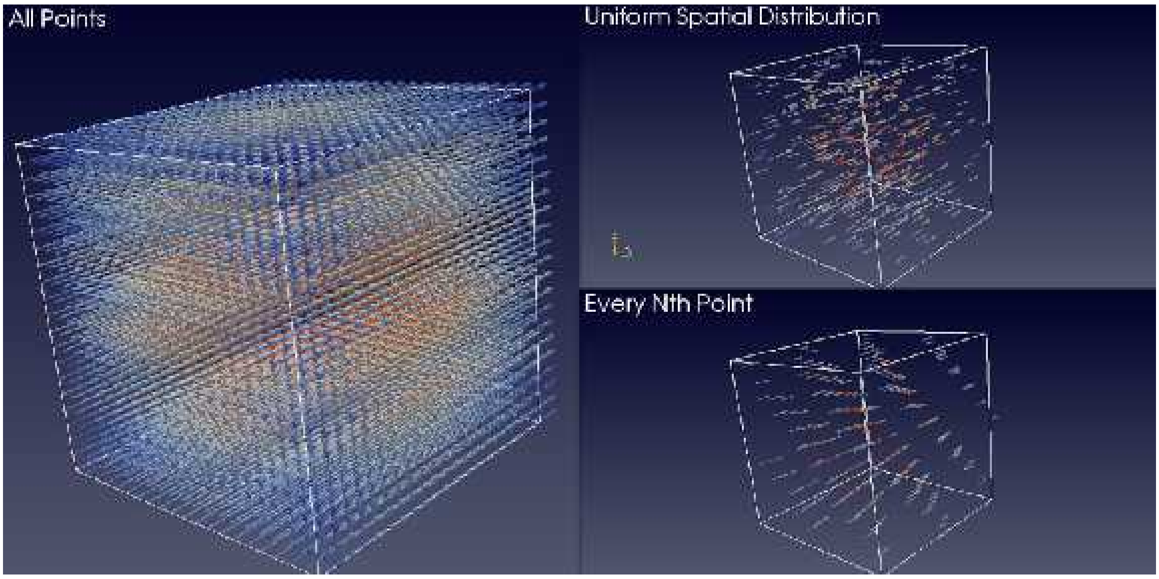
To create this filter in paraview, you can use the Filters menu, as well as the button on the Common filters toolbar. You first select the type for the glyph using one of the options in Glyph Type. The choices include Arrow, Sphere, Cylinder, etc.

The Glyph image shown is a Comparison between various Glyph Modes when applied to the same dataset generated by the Wavelet source.

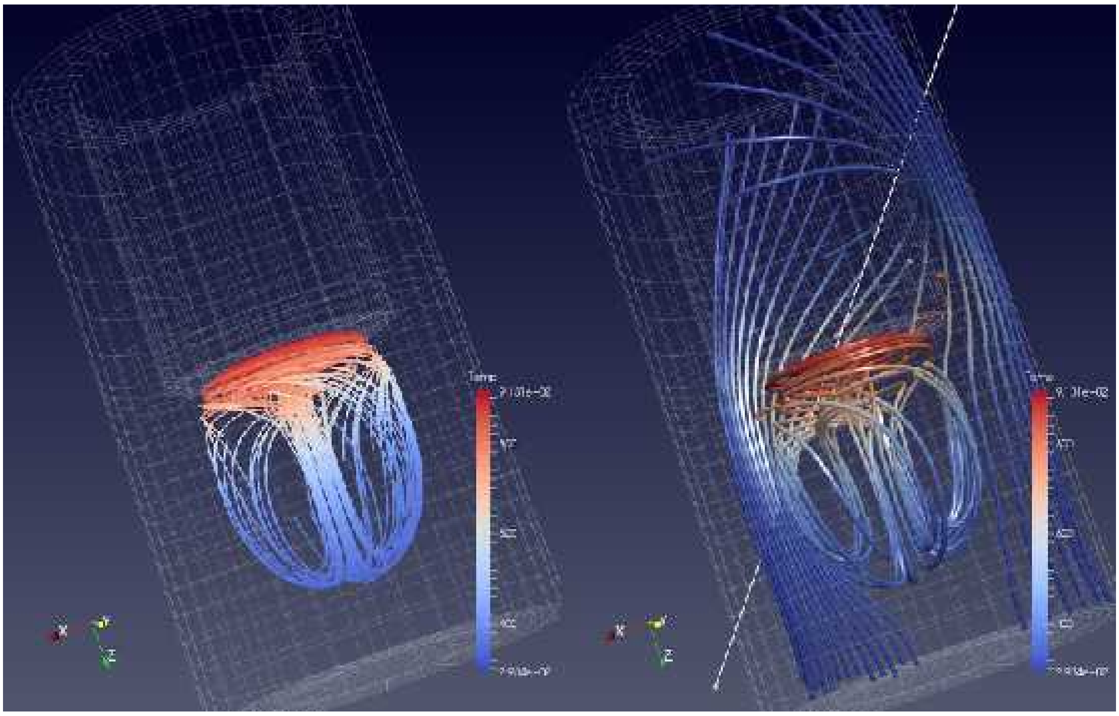
## Slice


Crinkle Slice: Off
Triangulate Slice: Off
Crinkle Slice: Off
Triangulate Slice: On
Crinkle Slice: On
Triangulate Slice: On/Off
Clip
(for comparison)

## Glyph


All Points
Uniform Spatial Distribution
Every Nth Point

## Stream Tracer



## Stream Tracer

The Stream Tracer filter is used to generate streamlines for vector fields. In visualization, streamlines refer to curves that are instanteneously tangential to the the vector field in the dataset. They provide an indication of the direction in which the particles in the dataset would travel at that instant in time. The algorihm works by taking a set of points, known as seed points, in the dataset and then inegrating the streamlines starting at these seed points.

In paraview, you can create this filter using the Filters menu, as well as the button on the Common filters toolbar.

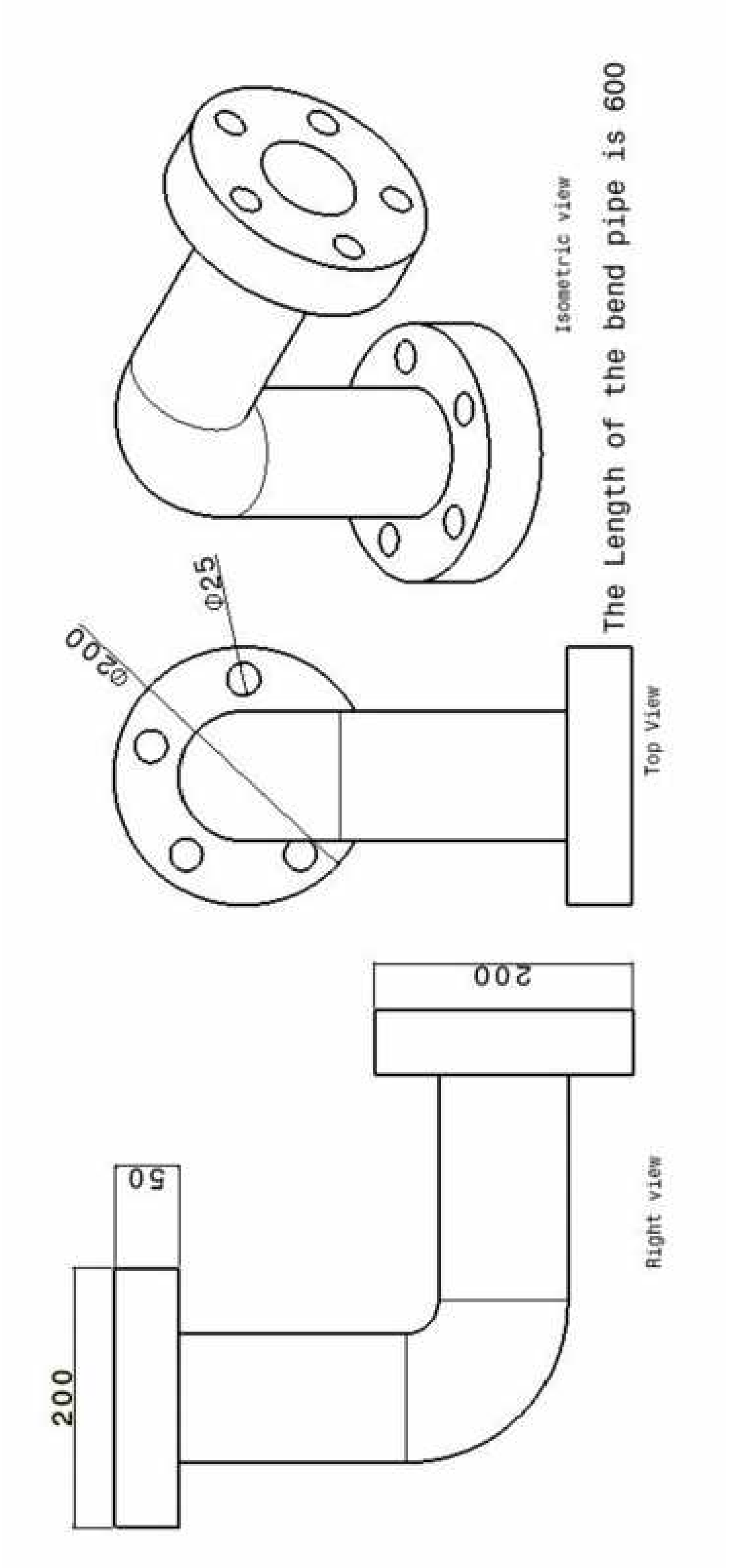
The Stream Tracer image shows the Streamlines generated from the disk out ref.ex2 dataset using the Point Source (left) and the High Resolution Line Source (right).

# LIST OF EXPERIMENTS

1. Modeling of a 3D Pipe with Flange
2. Structured grid generation of a 2D Plate
3. Unstructured grid generation of a 2D Plate
4. Structured grid generation of a 3D Pipe
5. Unstructured grid generation of a 3D Pipe
6. Incompressible internal laminar flow in a 3D Pipe
7. Incompressible internal turbulent flow in a 3D Pipe
8. Incompressible external laminar flow over an Aerofoil
9. Incompressible external turbulent flow over an Aerofoil

## Modeling of a 3D Pipe with Flange

Given: Draw the given 3D Model of the given pipe with Flange using Salome V6.



# Modeling of a 3D Pipe with Flange

**Ex.No. _______** **Date:**

**Aim**

To create a model of the given 3D Pipe with Flange using the Salome V6 package in CAE Linux.

**Software Required**

CAE Linux 2013 Operating System - Salome V6 Package

**Tools used**

| | |
|---|---|
| New Entity Tools | - Basic, Generation, Group, Blocks, Explode & Build |
| Operation Tools | - Boolean, Transformation & Fillets |

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required basic object.
5) The Build > Face tool is used to build a face of the object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required path to be followed by the face to create solid.
8) Create fillet for the sketch created using Operation>1D Fillet Tool.
9) Use the New Entity> Generation> Extrusion along the path tool to create the 3D solid from the face and path.
10) Create the flanges using the following tools
    - New Entity> Generation> Extrusion
    - New Entity> Basic>Circle
    - New Entity> Build> Face
    - Operation> Boolean> Cut
    - Operation > Transformation
11) Using Operation> Boolean> Add Tool finish the assembly of the created pipe with two flanges at both ends.
12) Save the output as JPEG image by using “Dump View” Tool in the View toolbar.

**Result**

Thus the required 3D model of the given Pipe with Flange is created using the Salome V6 package in CAE Linux.

# Structured grid generation of a 2D Plate

**Ex.No. _______** **Date:**

**Aim**

To generate structured grid for 2D Flat Plate using the Salome V6 package in CAE Linux.

**Software Required**

CAE Linux 2013 Operating System - Salome V6 Package

**Tools used**

Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode |

Meshing Module:

| | |
|---|---|
| Mesh Tools | - Create Mesh, Edit Mesh/Sub-Mesh, and Compute. |

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw a rectangular 2D Plate using 2D Sketch & Explode Tools.
4) Create a Block of Quadrangular Face using 4Edges method.
5) Go to the Mesh Module using icon or drop down option in the toolbar.
6) Mesh is created using the Mesh> Create Mesh tool with Quadrangular face created as the main geometry.
7) The algorithm of 2D: Automatic Quadrangulation is used.
8) The type of distribution is selected as “Equidistant distribution” is selected with no. of segments as 20.
9) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
10) The output of the generated structured grid is saved as JPEG using “Dump View” Tool in the View toolbar.

**Result**

Thus the 2D Structured Grid is generated for the flat plate using the Salome V6 package in CAE Linux.

## Unstructured grid generation of a 2D Plate

**Ex.No. _______** **Date:**

**Aim**

To generate unstructured grid for 2D Flat Plate using the Salome V6 package in CAE Linux.

**Software Required**

CAE Linux 2013 Operating System - Salome V6 Package

**Tools used**

Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode |

Meshing Module:

| | |
|---|---|
| Mesh Tools | - Create Mesh, Create Sub-Mesh, Edit Mesh/Sub-Mesh, Compute. |

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw a rectangular 2D Plate using 2D Sketch & Explode Tools.
4) Create a Block of Quadrangular Face using 4Edges method.
5) Go to the Mesh Module using icon or drop down option in the toolbar.
6) Mesh is created using the Mesh> Create Mesh tool with Quadrangular face created as the main geometry.
7) The algorithm of “2D: Automatic Quadrangulation” is used.
8) The type of distribution is selected as “Scaled Distribution” with a scale Factor of 0.25 is selected with no. of segments as 20.
9) The unstructured grid generation can also be done by creating a Sub-mesh using Vertex or Edge of the Quadrangle Face.
10) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
11) The output of the generated unstructured grid is saved as JPEG using “Dump View” Tool in the View toolbar.

**Result**

Thus the 2D Unstructured Grid is generated for the flat Plate using the Salome V6 package in CAE Linux.

# Structured grid generation of a 3D Pipe

**Ex.No. _______** **Date:**

**Aim**

To generate structured grid for 3D Pipe using the Salome V6 package in CAE Linux.

**Software Required**

CAE Linux 2013 Operating System - Salome V6 Package

**Tools used**

Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode, Extrusion along path |
| Operation Tools | - Boolean and 1D Fillet |

Meshing Module:

| | |
|---|---|
| Mesh Tools | - Create Mesh, Edit Mesh/Sub-Mesh, and Compute. |

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required basic object.
5) The Build > Face tool is used to build a face of the object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required path to be followed by the face to create solid.
8) Create fillet for the sketch created using Operation>1D Fillet Tool.
9) Use the New Entity> Generation> Extrusion along the path tool to create the 3D solid from the face and path.
10) Go to the Mesh Module using icon or drop down option in the toolbar.
11) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry.
12) The algorithm of “Netgen-1D-2D-3D” is used.
13) The hypothesis parameters are also applied with same minimum & maximum size to get a structured grid.
14) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
15) The output of the generated structured grid is saved as JPEG using “Dump View” Tool in the View toolbar.

**Result**

Thus the 3D Structured Grid is generated for the 3D Pipe using the Salome V6 package in CAE Linux.

# Unstructured grid generation of a 3D Pipe

**Ex.No. _______** **Date:**

**Aim**

To generate unstructured grid for 3D Pipe using the Salome V6 package in CAE Linux.

**Software Required**

CAE Linux 2013 Operating System - Salome V6 Package

**Tools used**

Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode, Extrusion along path |
| Operation Tools | - Boolean and 1D Fillet |

Meshing Module:

Mesh Tools - Create Mesh, Create Sub-Mesh, Edit Mesh/Sub-Mesh, and Compute.

**Procedure**

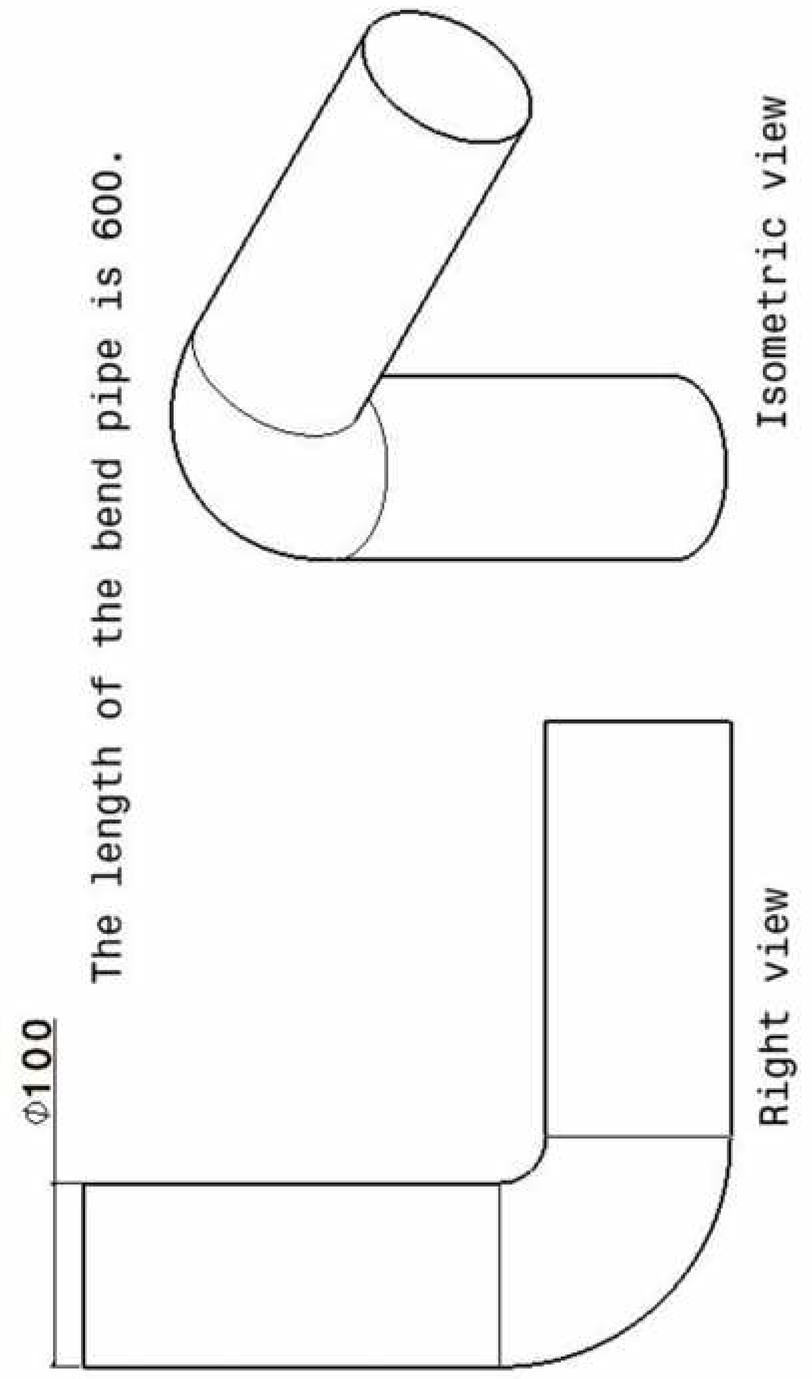
1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required basic object.
5) The Build > Face tool is used to build a face of the object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required path to be followed by the face to create solid.
8) Create fillet for the sketch created using Operation>1D Fillet Tool.
9) Use the New Entity> Generation> Extrusion along the path tool to create the 3D solid from the face and path.
10) Go to the Mesh Module using icon or drop down option in the toolbar.
11) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry. The algorithm of “Netgen-1D-2D-3D” is used.
12) The hypothesis parameters are also applied with varying minimum & maximum size to get an unstructured grid.
13) The unstructured grid generation can also be done by creating a Sub-mesh using Vertex or Edge or Face of the 3D Pipe.
14) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
15) The output of the generated structured grid is saved as JPEG using “Dump View” Tool in the View toolbar.

**Result**

Thus the 3D Unstructured Grid is generated for the 3D Pipe using the Salome V6 package in CAE Linux.

## Incompressible internal laminar flow in a 3D Pipe

**Given:** Analyze and study the Incompressible internal laminar flow of a fluid in a 3D Pipe using the CAE Linux Packages.



# Incompressible internal laminar flow in a 3D Pipe

**Ex.No. _______** **Date:**

**Aim**

To analyze and study the Incompressible internal laminar flow of a fluid in a 3D Pipe using the CAE Linux Packages.

**Software Required**

CAE Linux 2013 Operating System. Salome V6 Package, IcoFoam Solver.

**Tools used**

Salome- Geometry Module:

New Entity Tools - Basic, Blocks, Explode, Extrusion along path

Operation Tools - Boolean and 1D Fillet

Salome- Meshing Module:

Mesh Tools - Create Mesh, Edit Mesh/Sub-Mesh, and Compute.

IcoFoam Commands

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required basic object.
5) The Build > Face tool is used to build a face of the object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required path to be followed by the face to create solid.
8) Create fillet for the sketch created using Operation>1D Fillet Tool.
9) Use the New Entity> Generation> Extrusion along the path tool to create the 3D solid from the face and path.
10) The groups of “inlet, outlet & wall” are created.
11) Go to the Mesh Module using icon or drop down option in the toolbar.
12) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry.
13) The algorithm of “Netgen-1D-2D-3D” is used.
14) The hypothesis parameters are also applied with required minimum & maximum size.
15) The groups created are imported in meshing.
16) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
17) The generated mesh is exported as .unv mesh format and saved.
18) The study folder for “Icofoam- Incompressible-Cavity” is used for studying the given pipe flow.
19) The ideas-unv mesh file of the given pipe is converted to icofoam using the following command “ideasUnvToFoam Mesh_1.unv”.
20) The following changes are made as initial condition for the files in “Folder 0”:

Pressure P file:

- Group names are changed as inlet, wall, outlet
- Inlet & wall – no changes for commands
- Outlet – “*type fixedValue;*” & “*value uniform 0;*” commands are changed

Velocity U file:

- Group names are changed as inlet, wall, outlet
- Inlet & wall – no changes for commands
- Outlet – “*type zeroGradient;*” command is changed

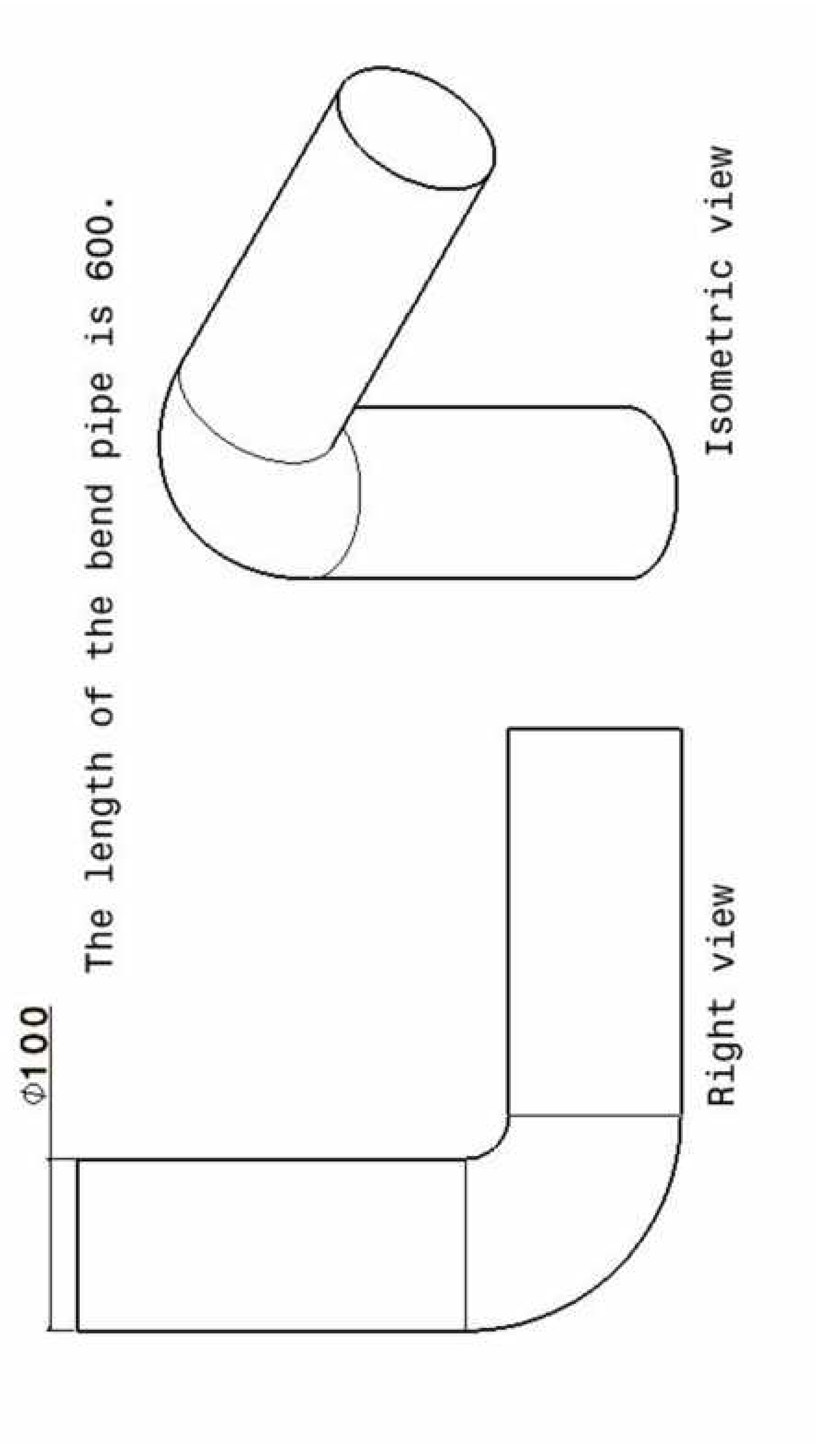
21) Then the study is solved using “icoFoam” command.
22) The Result is viewed using Paraview Viewer by executing “paraFoam” command.
23) The various results like Pressure & Velocity distribution, Vector Representation and Stream flow trace are done by “Slice, Glyph & Stream Tracer” Tools in the Paraview Package.
24) The outputs are saved as JPEG file using “Save as Screenshot” in the file menu.

**Result**

Thus the Incompressible internal laminar flow in a 3D Pipe study is done using the Salome V6 package & icoFoam Solver in CAE Linux.

### Incompressible internal turbulent flow in a 3D Pipe

**Given:** Analyze and study the Incompressible internal turbulent flow of a fluid in a 3D Pipe using the CAE Linux Packages



# Incompressible internal turbulent flow in a 3D Pipe

**Ex.No. _______** **Date:**

**Aim**

To analyze and study the Incompressible internal turbulent flow of a fluid in a 3D Pipe using the CAE Linux Packages.

**Software Required**

CAE Linux 2013 Operating System- Salome V6 Package, IcoFoam Solver.

**Tools used**

Salome- Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode, Extrusion along path |
| Operation Tools | - Boolean and 1D Fillet |

Salome- Meshing Module:

| | |
|---|---|
| Mesh Tools | - Create Mesh, Edit Mesh/Sub-Mesh, and Compute. |

Openfoam Commands

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required basic object.
5) The Build > Face tool is used to build a face of the object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required path to be followed by the face to create solid.
8) Create fillet for the sketch created using Operation>1D Fillet Tool.
9) Use the New Entity> Generation> Extrusion along the path tool to create the 3D solid pipe from the face and path.
10) The groups of “inlet, outlet & wall” are created.
11) Go to the Mesh Module using icon or drop down option in the toolbar.
12) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry.
13) The algorithm of “Netgen-1D-2D-3D” is used.
14) The hypothesis parameters are also applied with required minimum & maximum size.
15) The groups created are imported in meshing.
16) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
17) The generated mesh is exported as .unv mesh format and saved.
18) The openfoam Wizard mode is accessed and the following values are entered
    a. Mesh File is selected.
    b. Group names =inlet, outlet & wall
    c. Inlet velocity = For required direction- x/y/z = 1
    d. Inlet K = 0.00375

e. Inlet Epsilon = 0.000765
f. Nu = 0.001

19) Click OK to create the study files.
20) The terminal is opened in the study folder.
21) Then the study is solved using "./SimpleRun.sh" command.
22) The Result is viewed using Paraview Viewer by executing "paraFoam" command.
23) The various results like Pressure & Velocity distribution, Vector Representation and Stream flow trace are done by "Slice, Glyph & Stream Tracer" Tools in the Paraview Package.
24) The outputs are saved as JPEG file using "Save as Screenshot" in the file menu.

**Result**

Thus the Incompressible internal turbulent flow in a 3D Pipe study is done using the Salome V6 package & openFoam wizard Solver in CAE Linux.

## Incompressible external laminar flow over an Aerofoil

**Ex.No. _______** **Date:**

**Aim**

To analyze and study the Incompressible external laminar flow of a fluid flow over an aerofoil using the CAE Linux Packages.

**Software Required**

CAE Linux 2013 Operating System- Salome V6 Package, IcoFoam Solver.

**Tools used**

Salome- Geometry Module:

New Entity Tools - Basic, Blocks, Explode, Extrusion along path
Operation Tools - Boolean and 1D Fillet

Salome- Meshing Module:

Mesh Tools - Create Mesh, Edit Mesh/Sub-Mesh, and Compute.

IcoFoam Commands

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required aerofoil.
5) The Build > Face tool is used to build a face of the aerofoil object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required wireframe rectanglular boundary.
8) The Build > Face tool is used and the rectangle object is extruded surrounding the aerofoil.
9) The aerofoil is subtracted from the rectangular solid by using Boolean>Cut tool.
10) Then explode the cut object to form Faces using Explode>Face option.
11) The groups of “inlet, outlet & wall” are created.
12) Go to the Mesh Module using icon or drop down option in the toolbar.
13) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry.
14) The algorithm of “Netgen-1D-2D-3D” is used.
15) The hypothesis parameters are also applied with required minimum & maximum size.
16) The groups created are imported in meshing.
17) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
18) The generated mesh is exported as .unv mesh format and saved.
19) The study folder for “Icofoam- Incompressible-Cavity” is used for studying the given pipe flow.
20) The ideas-unv mesh file of the given pipe is converted to icofoam using the following command “ideasUnvToFoam Mesh_1.unv”.
25) The following changes are made as initial condition for the files in “Folder 0”:

Pressure P file:

- Group names are changed as inlet, wall, outlet
- Inlet & wall – no changes for commands
- Outlet – “*type fixedValue;*” & “*value uniform 0;*” commands are changed

Velocity U file:

- Group names are changed as inlet, wall, outlet
- Inlet & wall – no changes for commands
- Outlet – “*type zeroGradient;*” command is changed

21) Then the study is solved using “icoFoam” command.
22) The Result is viewed using Paraview Viewer by executing “paraFoam” command.
23) The various results like Pressure & Velocity distribution, Vector Representation and Stream flow trace are done by “Slice, Glyph & Stream Tracer” Tools in the Paraview Package.
24) The outputs are saved as JPEG file using “Save as Screenshot” in the file menu.

**Result**

Thus the the Incompressible external laminar flow of a fluid flow over an aerofoil study is done using the Salome V6 package & icoFoam Solver in CAE Linux.

# Incompressible external turbulent flow over an Aerofoil

**Ex.No. _______** **Date:**

**Aim**

To analyze and study Incompressible external turbulent flow of a fluid over an Aerofoil using the CAE Linux Packages.

**Software Required**

CAE Linux 2013 Operating System- Salome V6 Package, Openfoam Wizard Solver.

**Tools used**

Salome- Geometry Module:

| | |
|---|---|
| New Entity Tools | - Basic, Blocks, Explode, Extrusion along path |
| Operation Tools | - Boolean and 1D Fillet |

Salome- Meshing Module:

| | |
|---|---|
| Mesh Tools | - Create Mesh, Edit Mesh/Sub-Mesh, and Compute. |

Openfoam Commands

**Procedure**

1) Open the software Salome V6 package in CAE Linux.
2) Go to Geometry Module using icon or drop down option in the toolbar.
3) Use the New Entity Tools to draw the objects.
4) Use Basic > Circle, line, point, arc, ellipse, etc to draw the required aerofoil.
5) The Build > Face tool is used to build a face of the aerofoil object to be extruded.
6) The required Plane is created using Basic>Plane Tool in the new entity toolbar.
7) Using 2D sketch in Basic tool draw the required wireframe rectanglular boundary.
8) The Build > Face tool is used and the rectangle object is extruded surrounding the aerofoil.
9) The aerofoil is subtracted from the rectangular solid by using Boolean>Cut tool.
10) Then explode the cut object to form Faces using Explode>Face option.
11) The groups of “inlet, outlet & wall” are created.
12) Go to the Mesh Module using icon or drop down option in the toolbar.
13) Mesh is created using the Mesh> Create Mesh tool with Pipe1 created as the main geometry.
14) The algorithm of “Netgen-1D-2D-3D” is used.
15) The hypothesis parameters are also applied with required minimum & maximum size.
16) The groups created are imported in meshing.
17) Then the mesh is computed using Mesh>Compute Tool to generate the mesh.
18) The generated mesh is exported as .unv mesh format and saved.
19) The openfoam Wizard mode is accessed and the following values are entered
    a. Mesh File is selected.
    b. Group names =inlet, outlet & wall
    c. Inlet velocity = For required direction- x/y/z = 30
    d. Inlet K = 0.00375

   e. Inlet Epsilon = 0.000765
   f. Nu = 0.001

20) Click OK to create the study files.
21) The terminal is opened in the study folder.
22) Then the study is solved using "./SimpleRun.sh" command.
23) The Result is viewed using Paraview Viewer by executing "paraFoam" command.
24) The various results like Pressure & Velocity distribution, Vector Representation and Stream flow trace are done by "Slice, Glyph & Stream Tracer" Tools in the Paraview Package.
25) The outputs are saved as JPEG file using "Save as Screenshot" in the file menu.

**Result**

Thus the Incompressible external turbulent flow of a fluid over an Aerofoil study is done using the Salome V6 package & openFoam wizard Solver in CAE Linux.