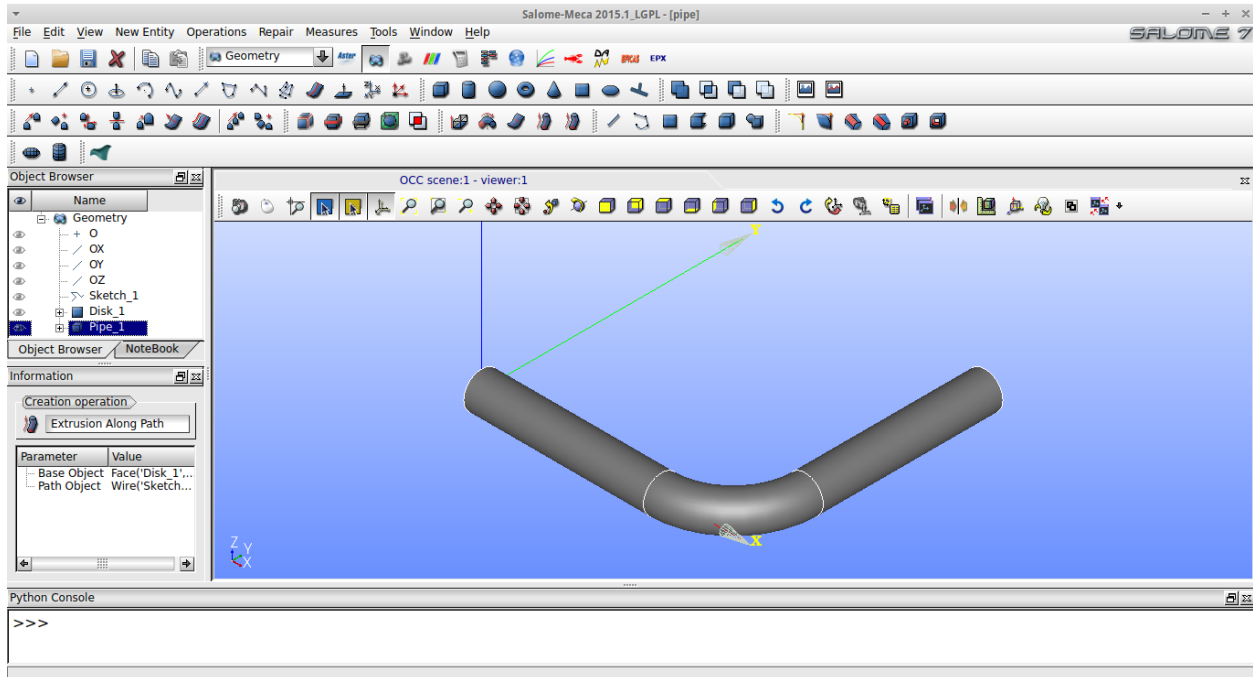


SnappyHexMesh es una herramienta excelente para hacer mallas sobre geometrias complicadas. No obstante también se puede hacer una malla desde el propio Salome para, posteriormente, ser usada con simpleFoam, o cualquier otro solver de OF. Veamos como se hace.

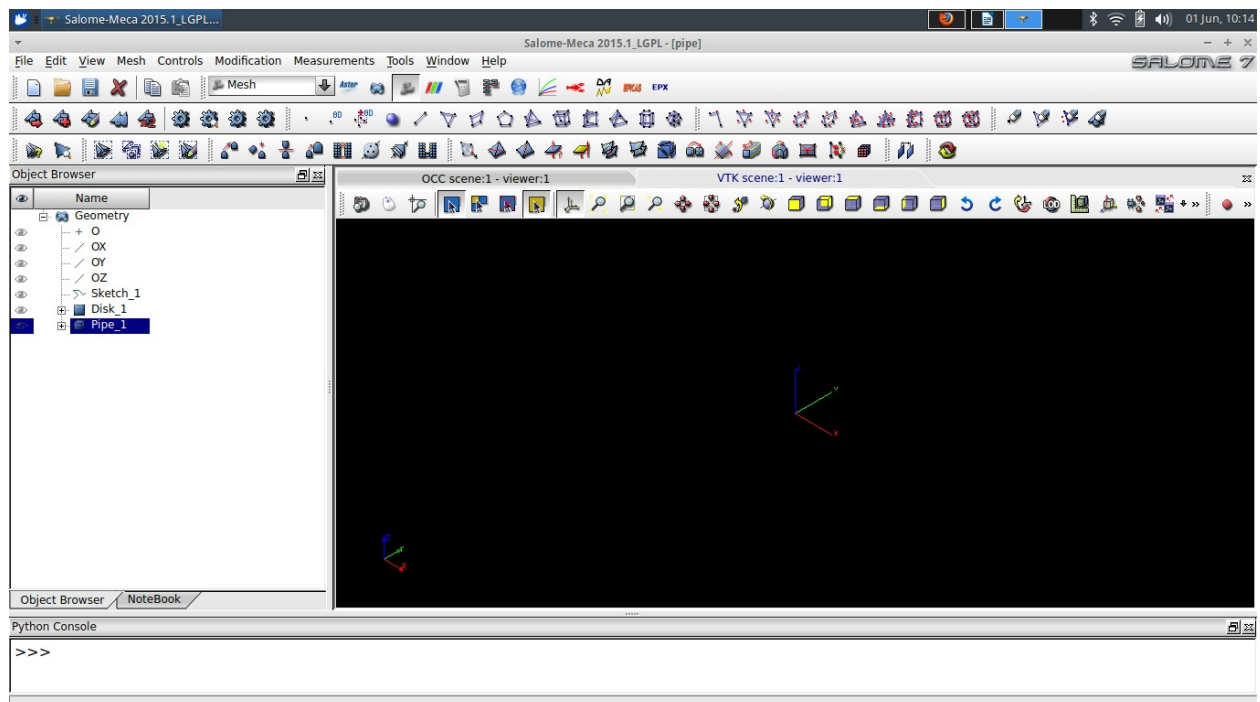
Partimos del estudio que hemos grabado de la tueria hecha con Salome, que, posiblemente, hemos grabado con el nombre “pipe.hdf” o similar. Si no lo tenemos, tendremos que volver a generar la geometría, como se explica en el documento anterior.

Nuestra geometria en Salome debe verse asi:

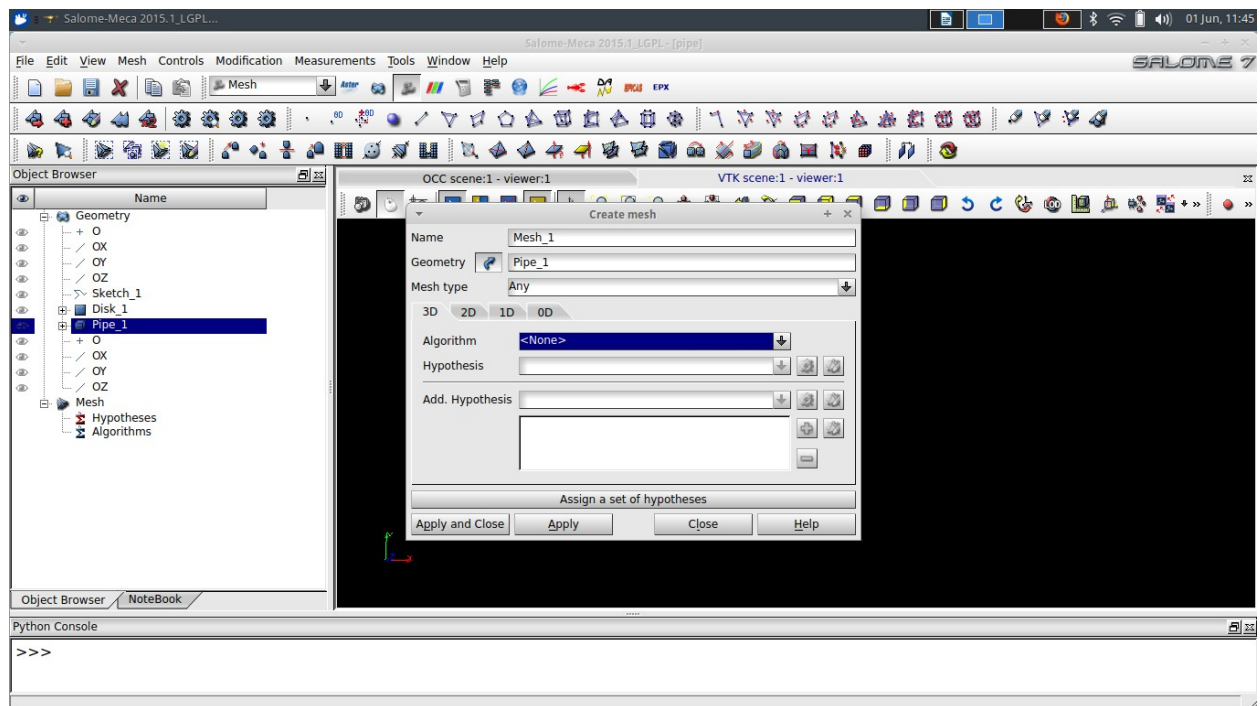


Para crear una malla, activamos el modulo SMESH, o bien con el icono de la primera barra de herremientas, o bien con el menu “Modules”. Salome cambia ahora de formato.

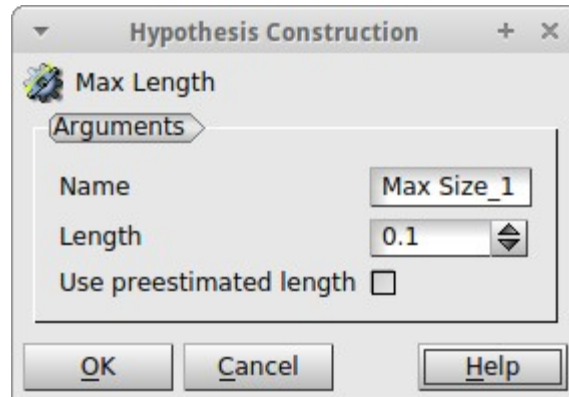
Salome no es exactamente un mallador, si no, más bien, un “Front End” para diferentes malladores que debemos tener instalados en nuestro ordenador. Estos malladores son los “Algoritmos” que usa salome, y, para crear las mallas, necesita una serie de hipótesis. En ocasiones, si la malla es 3D, necesitamos antes generar las hipótesis para las mallas 2D y 1D que ha de generar en los niveles inferiores.



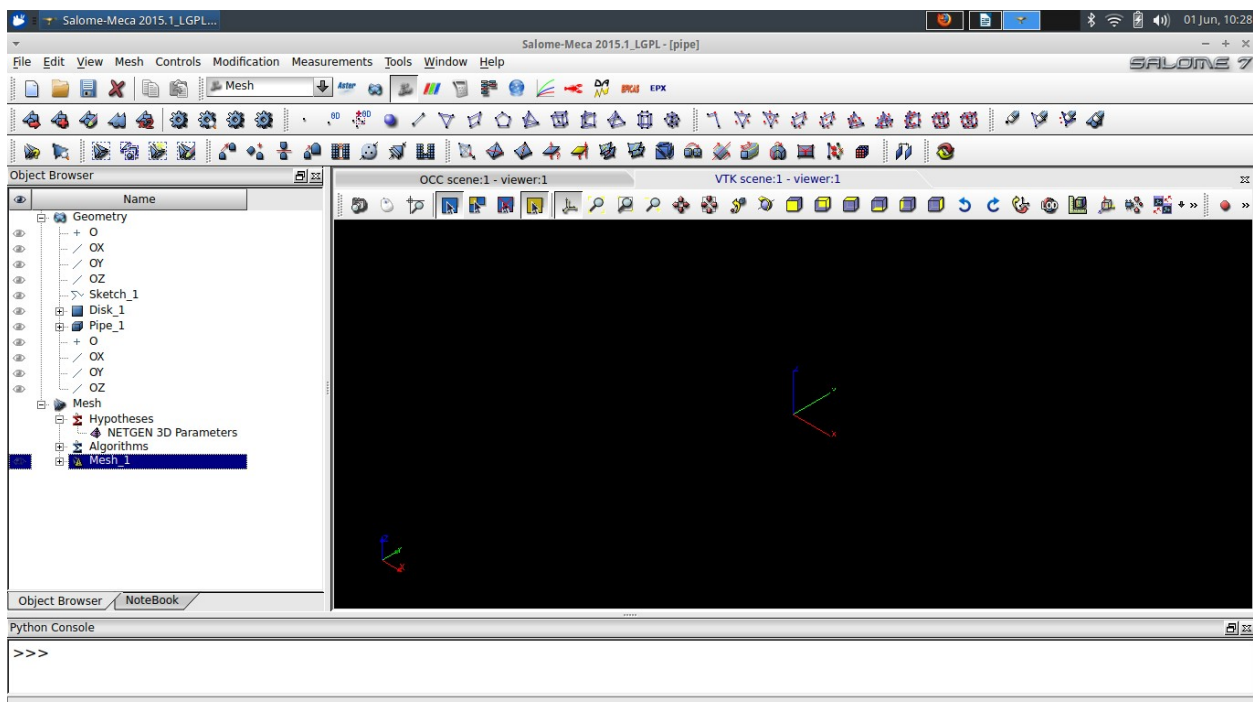
Creamos una nueva malla, Mesh → Create Mesh, escogemos un nombre para la malla (podemos dejar el que hay por defecto, Mesh_1), una geometría (nuestra tubería). Una forma rápida y sencilla de crear una malla de tetrahedros es con “Assign a set of hypothesis”.



Escogemos “3D: Automatic tetrahidralization” y, como nuestro diámetro es de 1 metro, definimos la longitud máxima de discretización como 0.1 m



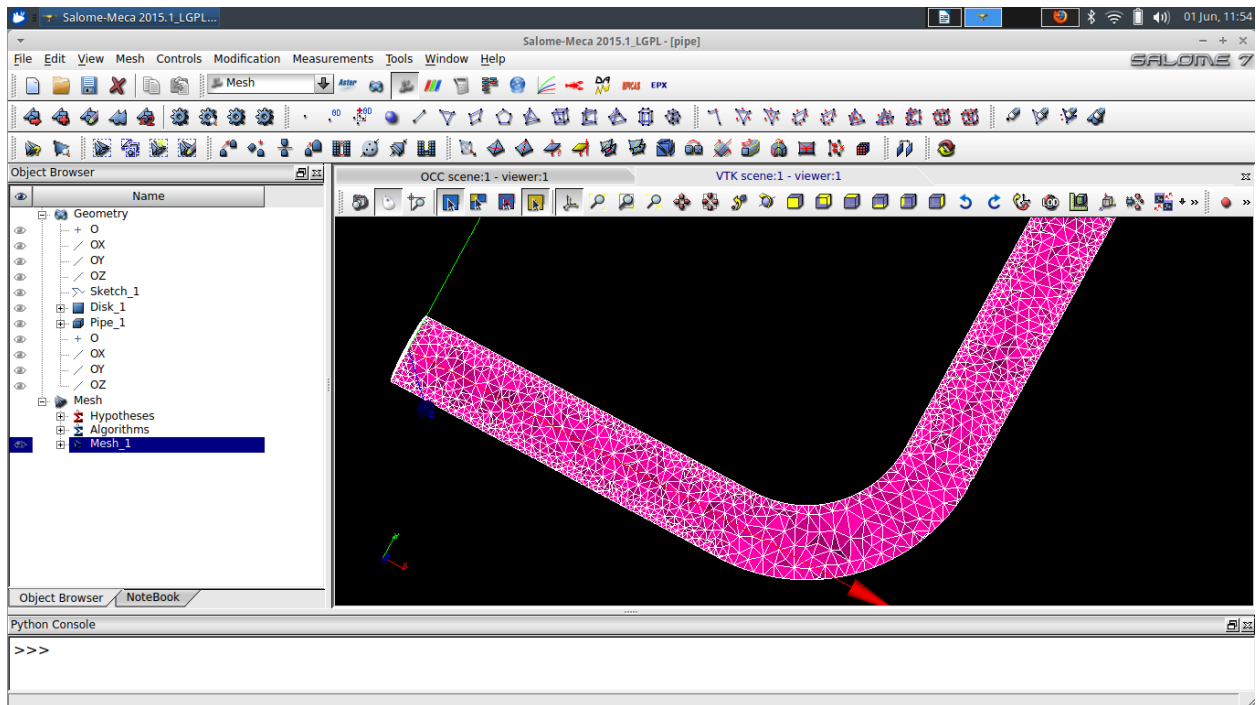
Clicamos en “OK” y en “Apply and close”
La malla todavía no ha sido creada, solo definida.



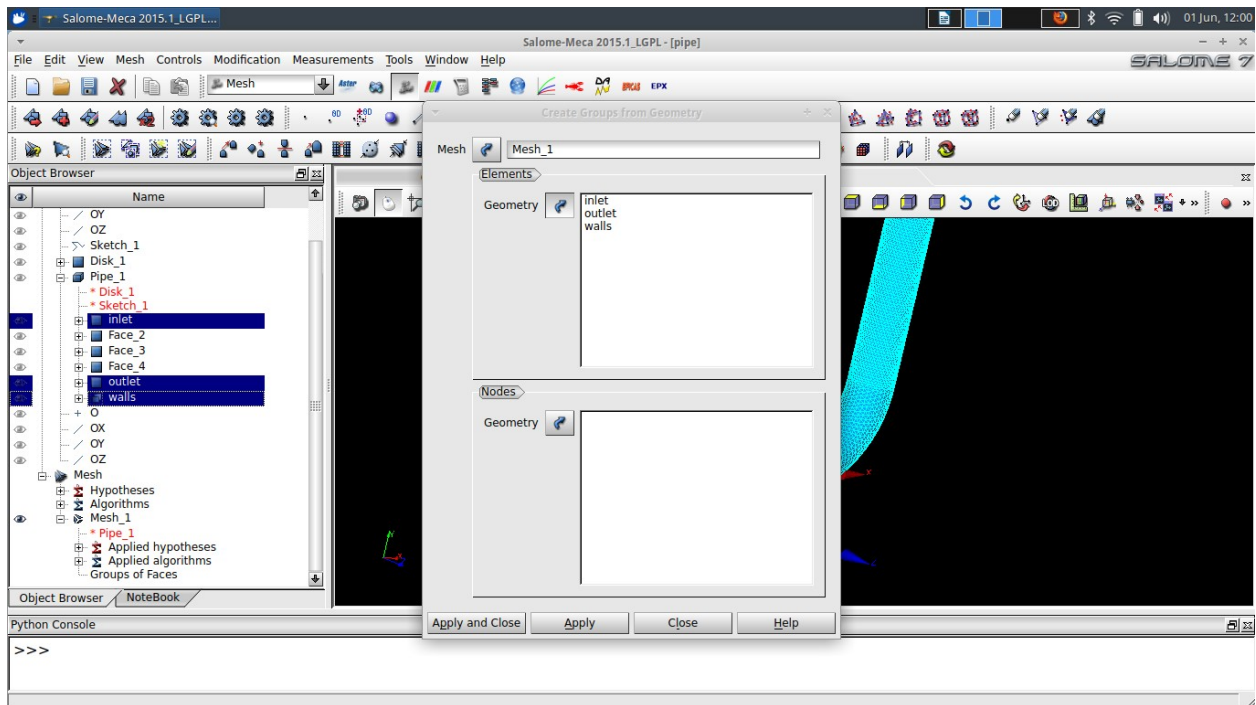
Para generarla, hacemos Mesh → Compute, y nos crea una malla de tetrahedros de unos 57000 volúmenes.

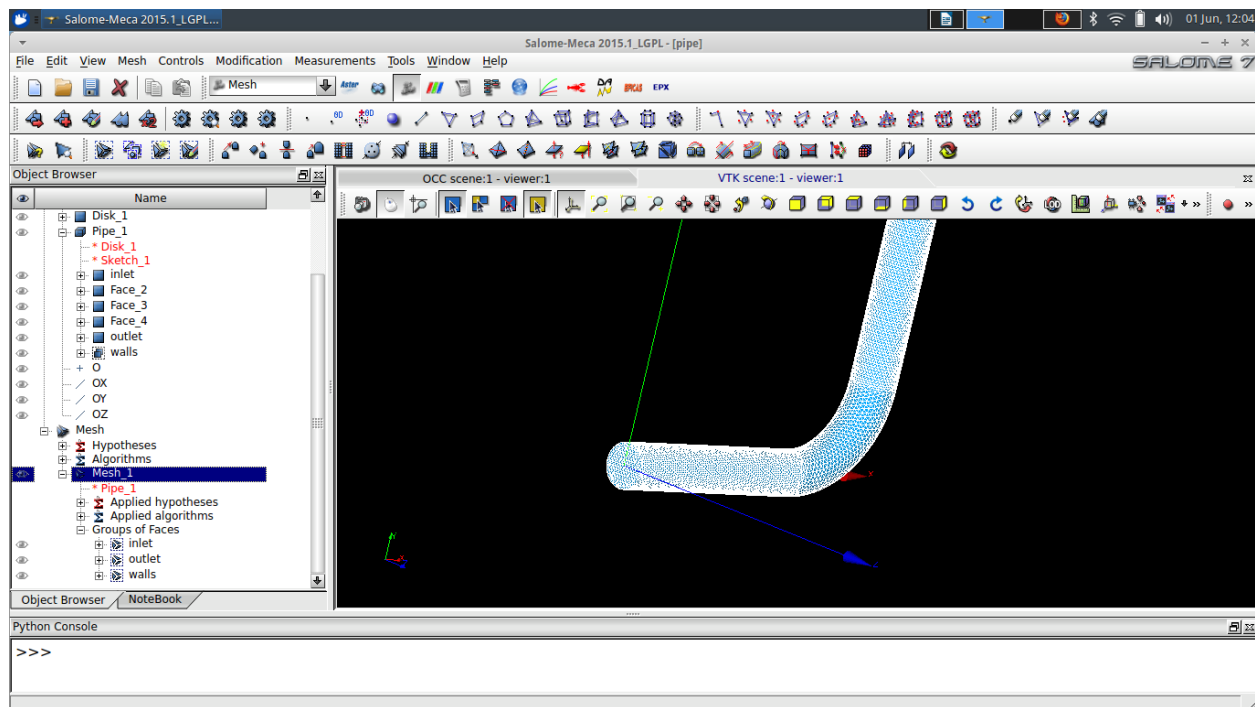
Si lo deseamos, podemos generar una malla de capa límite en las hipótesis, antes de calcular la malla.

Para visualizar el interior de la malla, en el menú contextual de la misma, podemos escoger “Clipping” que nos permite definir un plano de corte y ver la malla en ese plano. Si definimos el plano X=1 (el plano XY), veremos algo parecido a esto:



Definimos ahora las condiciones de contorno (inlet, outlet y walls). Lo hacemos como grupos de elementos de la malla. Mesh → Create Groups from Geometry. Podemos escoger varias a la vez con la tecla “Control”.





Y ya esta creada la malla! Ahora solo falta exportarla a un formato que se pueda usar con OF. Usamos el formato UNV de IDEAS. File → Export → UNV File. La llamamos, por ejemplor, pipe.unv

Ya podemos grabar la sesión de Salome y cerrarla.

Creamos una carpeta “pipeSalome” y copiamos en ella las carpetas “0.org”, “system” y “constant” del caso que teniamos hecho con snappyHexMesh. Copiamos también la malla pipe.unv

En “pipeSalome” ejecutamos

```
foamClearPolyMesh
ideasUnvToFoam pipe.unv
cp -r 0.org 0
```

Podemos visualizar la malla con paraFoam.

La simulación es igual que con snappyHexMesh

```
foamJob -s potentialFoam ; mv log lo.potentialFoam
```

Antes de hacer simpleFoam, debemos recordar de cambiar en “constant/polyMesh/boundary” el tipo de patch de “walls”, de “patch” a “wall”

```
boundary - Mousepad
File Edit View Text Document Navigation Help
format ascii,
class polyBoundaryMesh;
location "constant/polyMesh";
object boundary;
}
// *****

3
(
    inlet
    {
        type patch;
        nFaces 322;
        startFace 104703;
    }
    outlet
    {
        type patch;
        nFaces 322;
        startFace 105025;
    }
    walls
    {
        type wall;
        nFaces 16218;
        startFace 105347;
    }
)
// *****

Filetype: None Line: 30 Column: 31 OVR
```

Ya podemos hacer la simulación

```
foamJob -s simpleFoam; mv log log.simpleFoam
```

Notamos que necesita casi el doble de iteraciones que con sHM. La calidad de la malla (ángulo de no-Ortogonalidad) es inferior, y de aquí, posiblemente, la mayor cantidad de iteraciones para converger.