

ASSÉMAT Jean-Pascal, LETEMPLIER Warren

Affiliation of Authors

Abstract: This paper details a GUI based application for performing automated design and analysis of piping parts at Technip. Technip is a world leader in the fields of project management, engineering and construction for the oil & gas industry.

Strategy adopted for this application is to provide designers with a plugin to update CAE parametric model build by experts. The application can be customized via xml configuration files to add other models.

Keywords: pipe, automation, Abaqus/CAE

1. Motivations

This application goal is to automate the design of tee used in sub-sea deepwater installation.

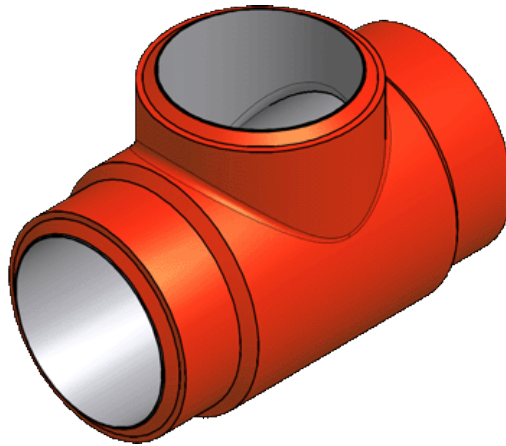


Figure 1. a tee

Main motivations for this development are:

- Contributing to higher productivity and reduced analysis time,
- Enabling complex Abaqus/CAE model to be easily manipulated by non Abaqus experts,

- Offering better traceability through the use of a model library

2. Development

2.1 Structure of the development

This development is based on python scripts, Abaqus/CAE models (constituting the model library) and configuration files.

The user will be prompted to choose one configuration of tee and to define some dimensional and material information. He will be able to apply specific load cases to the tee.

Tee dimensions are covered by standards for onshore installation. But when one uses these standards for a sub-sea deepwater tee, those which cover operation in service apply, whereas none covers the constraints generated during the lay of this piping.

The purpose of configuration files is to provide an application which can be easily extended or modified.

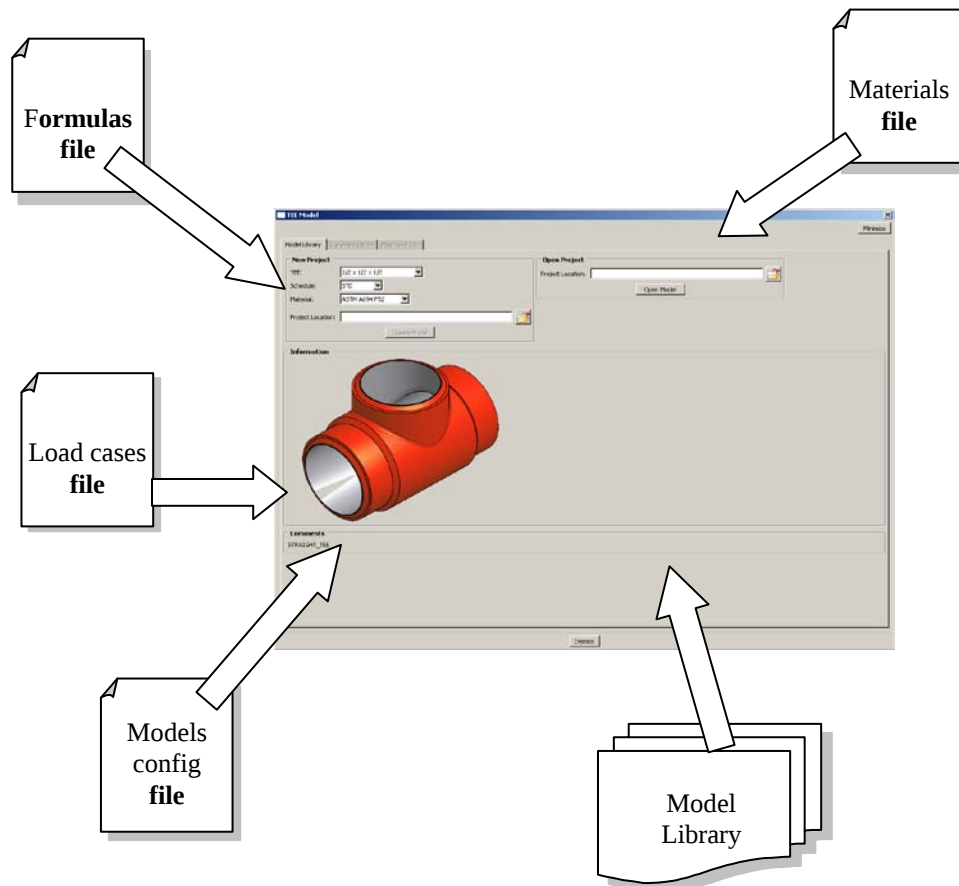


Figure 2. Files used to configure the application

2.2 Models configuration file

This file will list all available tee configurations.

An Abaqus/CAE model can be used for one or several configuration depending on geometric and meshing specificities.

For each Abaqus/CAE model, the file will describe all the parameters that can be modified by the application.

A parameter is defined by:

- A name (used to retrieve the parameter in the Abaqus/CAE model),
- An alias (the one which will be shown to the end user),
- A description of the parameter,
- A visibility to the end-user,
- An editability,

- Name of formula in the formula file which will be used to compute parameter value (optional),
- Locations of the parameter in the Abaqus/CAE model (the application can drive several parameters, in several sketches for instance)

```
<Parameter name="Dr0" alias="Dr0">
  <Description>outside diameter of run</Description>
  <Interface visible="False" editable="False" />
  <Formula>fDr0</Formula>
  <Reference partName="TEE_STRAIGHT"
    featureName="Run_Outside_Contour" attributeName="Dr"
    type="SKT_PARAM" />
  <Reference partName="TEE_STRAIGHT"
    featureName="Run_Inside_Contour" attributeName="Dr"
    type="SKT_PARAM" />
</Parameter>
```

Figure 3. Models Configuration file: Parameter section

This file will define mesh specification too. Specifications are set on edges groups and can be a number of element or an element size.

```
<Region name="Width_Big" alias="Width_Big">
  <Value>2</Value>
  <Reference instanceName="TEE_INSTANCE" setName="Edges_Width_Big"
    type="NUMBER" />
</Region>
```

Figure 4. Models Configuration file: mesh specification section

2.3 Formulas file

The formulas file is a plain text file. Formulas are written in Python language. Function names are used in the models configuration file to compute parameter value.

Variables used in formula can be parameters defined in the configuration file and are identified directly with their name.

```

import math

def fDr0():
    """
    Returns the value of Dr
    @type: float
    """
    if math.fabs(Odh-Dr)<0.001:
        return Odh+0.001
    else:
        return Dr

```

Figure 5. Formulas file: example of function

2.4 Materials file

This file is dedicated to store all available materials which can be applied to the tee.

```

<Material>
  <Name>ASTM A694 F52</Name>
  <Comment>Matériau 01</Comment>
  <Parameter name="mass density" alias="Rho"
    unit="kg / mm ^3">0.00000785</Parameter>
  <Parameter name="young's modulus" alias="E" unit="MPa">210000</Parameter>
  <Parameter name="poisson ratio" alias="Nu" unit="">0.3</Parameter>
  <Parameter name="yield strength" alias="SY" unit="MPa">360</Parameter>
  <Parameter name="tensile strength" alias="ST" unit="MPa">455</Parameter>
</Material>

```

Figure 6. Material file: example of material

2.5 Load cases file

This file is composed of 4 sections :

- Outputs (outputs in the odb file),

```

<Output name="Field_Static">
  <Type>FIELD</Type>
  <Variables>CDISP,CF,CSTRESS,LE,PE,PEEQ,PEMAG,RF,S,U</Variables>
</Output>

```

Figure 7. Load cases file : Output section

- Loads (name, application point name, type)

```

<Load name="PRESSURE_I">
<Description>pressure inside tee</Description>
<Reference instanceName="TEE_INSTANCE" surfaceName="SURF_INSIDE" type="PRESSURE" />
</Load>

```

Figure 8. Load cases file : Loads section

- BoundaryConditions (name, support, values)

```

<BoundaryCondition name="PT_C_U2_U3">
<Description>u2 and u3 clamped on point C</Description>
<Reference instanceName="TEE_INSTANCE" setName="Surface_C" csysName="CSYS_CYLINDRICAL"
type="DISPLACEMENT" />
<Displacement u2="0.0" u3="0.0" />
</BoundaryCondition>

```

Figure 9. Load cases file : Bounday conditions section

- LoadCases (name, outputs/loads/boundary conditions to be used)

```

<LoadCase>
<Type>Factory</Type>
<Name>Onshore Hydro Test</Name>
<Comment>Factory Acceptance Test</Comment>
<Interface default="True" noFEM="True" />
<Steps>
<Step name="Onshore_Hydro_Test" type="STATIC">
<Loads>* </Loads>
<BoundaryConditions>PT_C_U2_U3</BoundaryConditions>
<Outputs>Field_Static,History_Static</Outputs>
</Step>
</Steps>
</LoadCase>

```

Figure 10. Load cases file : Load cases section

The user will be prompted to choose load cases that must be applied to the model. .

2.6 Model Library

The Abaqus/CAE template model is a parametric model (sketches, features and partition) and is built by an Abaqus expert. The main issue is to build the most generic model.

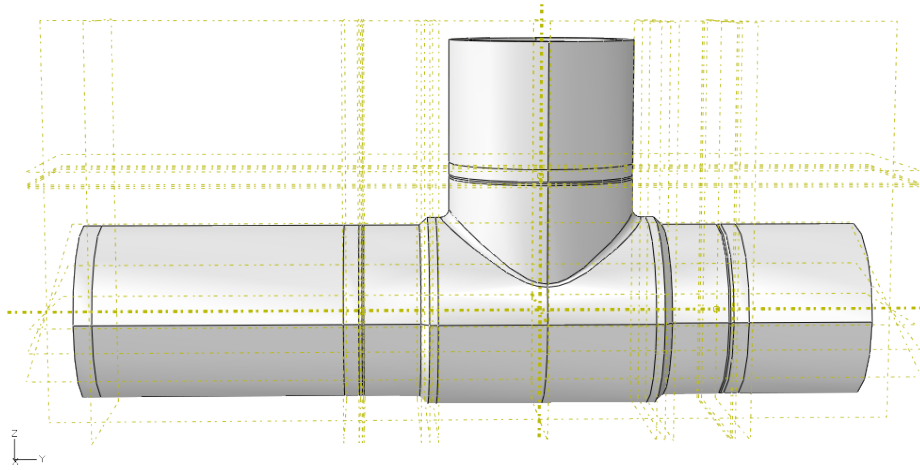


Figure 11. A parametric tee model

2.6.1 Mesh

Models are meshed using hexahedrons elements (C3D8R) or tetrahedral elements (C3D10I).

Partitions cells are driven by planes. Positions of these planes are determined through geometrical relations according to the dimension of the tee.

Meshing specifications are defined in the model configuration file. Elements size or number are set using edges groups. The specification value can be linked to a formula in the formula file.

In order to update the mesh after geometry modification, we have to keep the same topology.

Since the size of the tee and the pipe connected to the tee could be different, the topology changes. In figure 11, the mesh regeneration will fail if the angle α is modified from 0° to any other value, since in one case we have 1 edge and the other 3 edges.

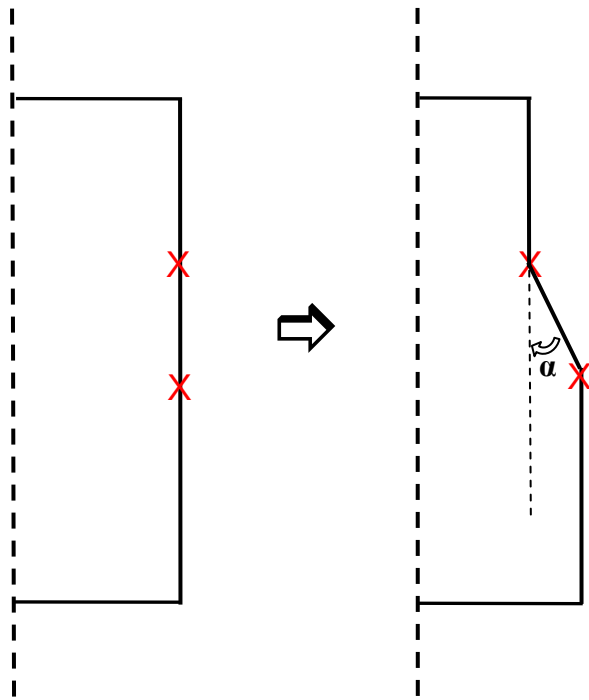


Figure 12. Topology changes

By-pass used is to introduce a permanent small angle of 1 μ rad when updating the value.

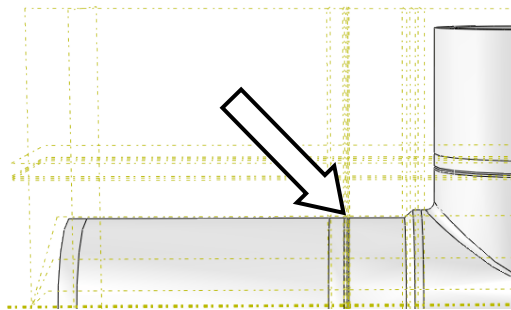


Figure 13. Pipe / Tee Connection

Another by-pass is to create via Python cut planes, loads and boundary conditions. The main issue is to retrieve cells on which the application will apply mesh, loads and boundary specifications.

2.6.2 Steps

Abaqus/CAE templates models don't contain steps or loads: they will be added by the application.

3. Use case

This section describes a basic use case beginning from the instantiation of a template model in the model library to the computation.

3.1 Model instantiation

The user will first define the configuration of the model which will be copied in the working directory. The instanced model will be configured with standard geometrical values.

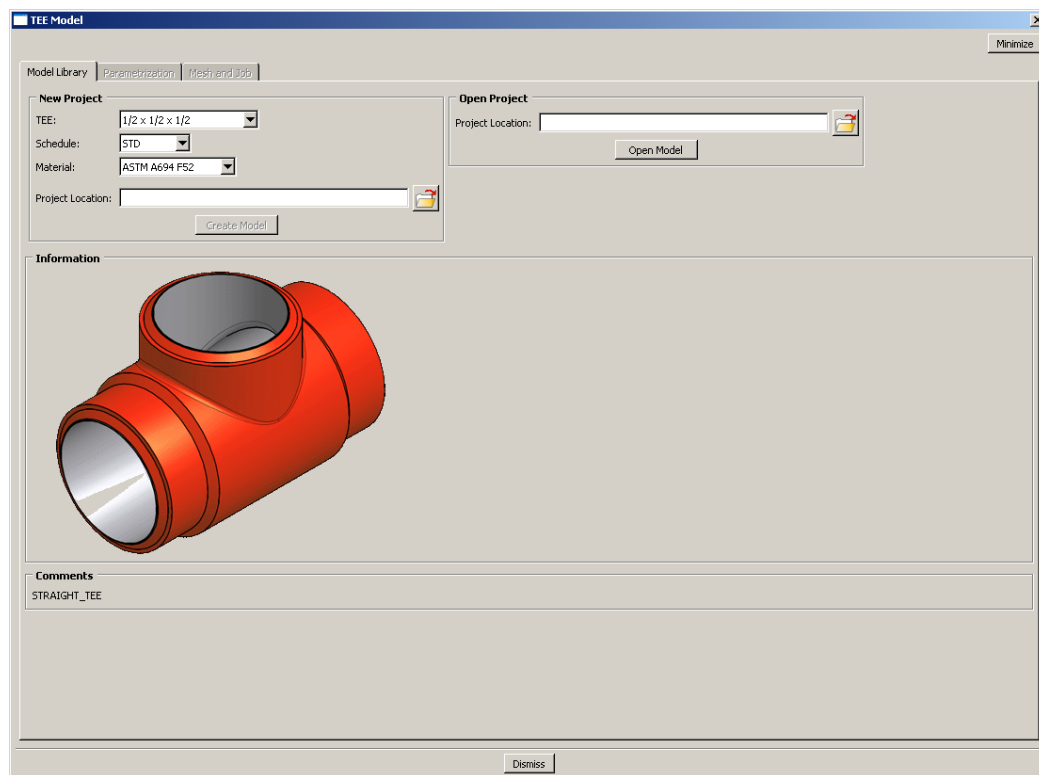


Figure 14. Models library

3.2 Parameters modification

After model creation, the user will modify geometrical values via a table as shown in figure 13. Each parameter is associated with a picture to help the designer to configure its model.

Parameters values are updated via the application. These values are either directly entered by the user or computed with formulas. These formulas can refer to other parameters values, which can be located anywhere in the model.

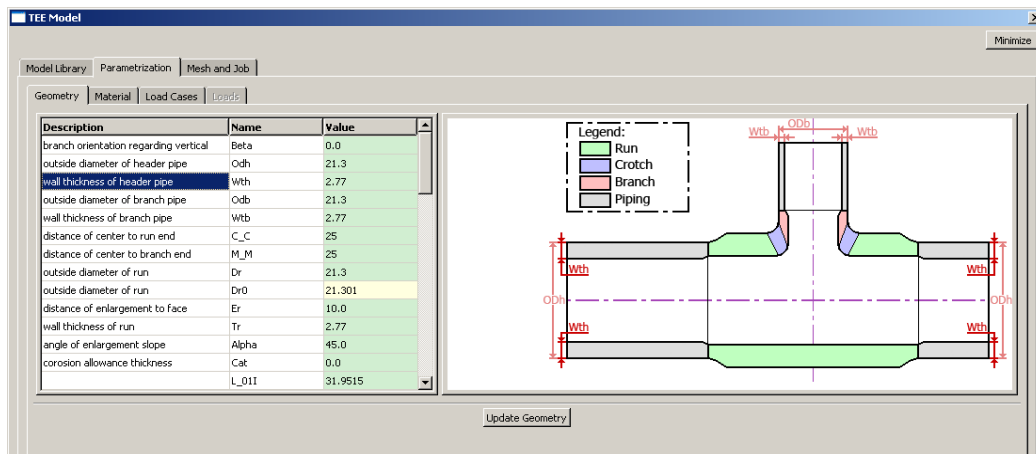


Figure 15. Parameters modification interface

3.3 Materials

The user is able to edit material used in the model and to add a plastic behaviour.

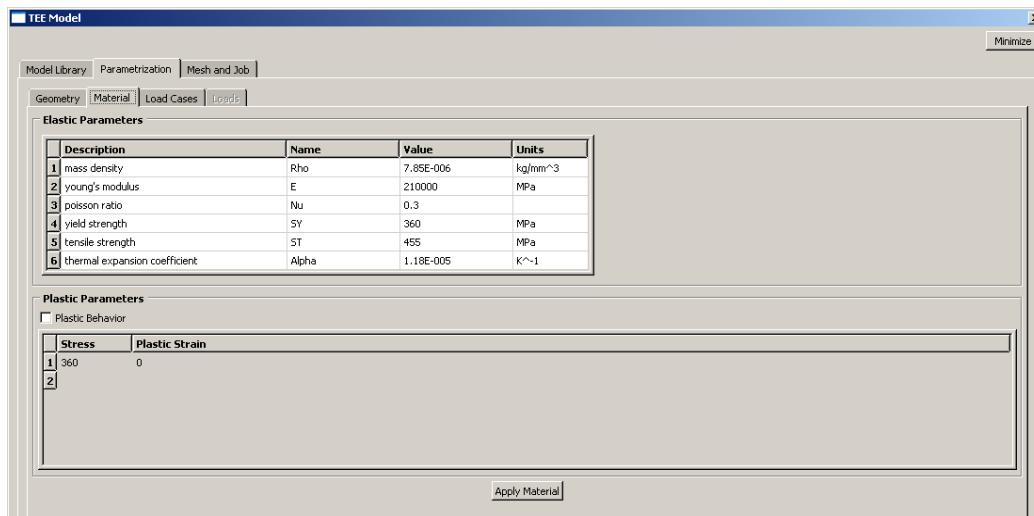


Figure 16. Material definition interface

3.4 Boundary conditions and load cases

The user will then choose loads cases defined in the load cases file. At this step, steps, loads and restraints are created in the model.

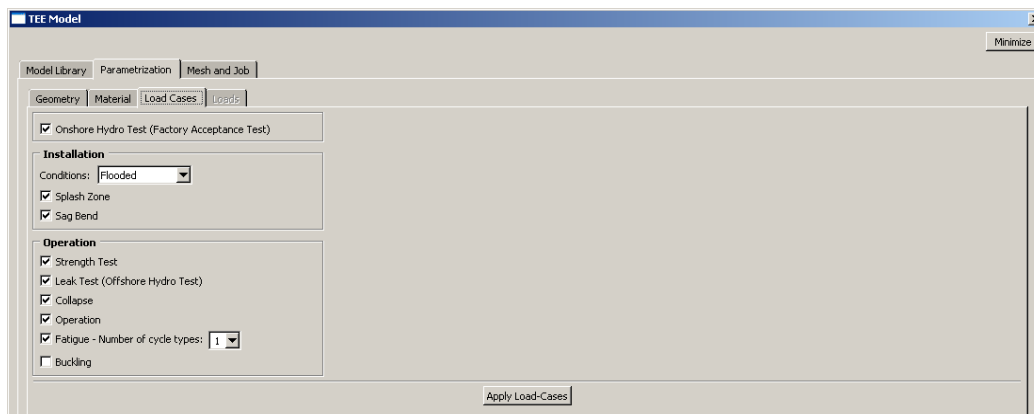


Figure 17 Load cases definition interface

The application will provide an interface to set loads values.

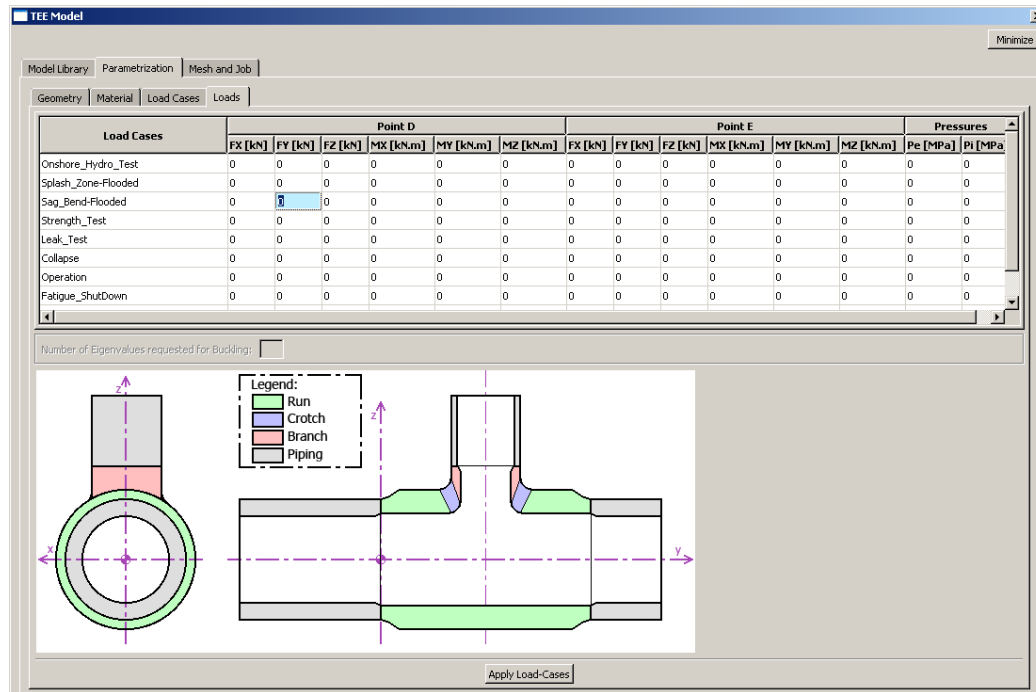


Figure 18. Loads definition interface

3.5 Model update and computation

Finally, the user will generate the inp file and compute the job locally or on a remote server using the application.

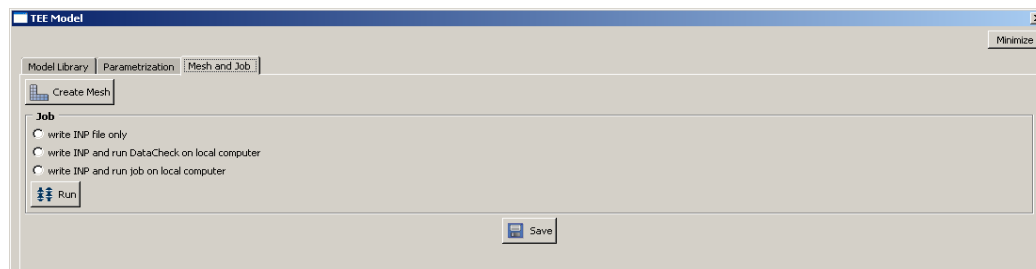


Figure 19. Computation interface

4. Tee from the application versus model by hand

Since the model is well defined (geometry and loads), it can easily be automated. The main issue is to provide a generic methodology for the meshing.

On the one hand, by using the application the designer is able to iterate on the design of the tee rapidly, but on the other hand, the automation of the process cannot provide expert meshing. More over in some configuration, the model can have some wrong elements. In all cases, designers can save a lot of time on the update of the geometry when changing parameters values.

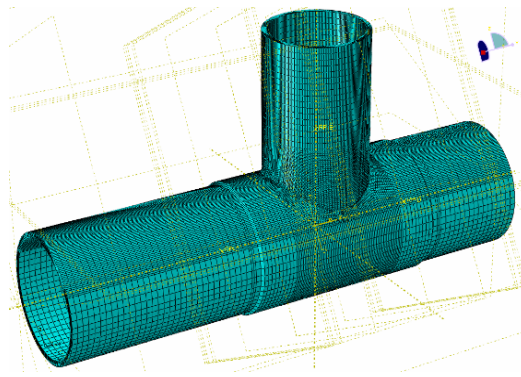


Figure 19. Mesh of the tee

5. Conclusion

This application helps designers who are not Abaqus specialists to have a high productivity by reducing analysis time. It uses the scriptability of Abaqus/CAE to provide to the user a nearly full tee parametric model.

The Python code could be used, with little modifications, for driving any kind of parametric model. The main work would be on Abaqus/CAE models and configuration files.

The development of the application took approximately 20 man/weeks which include Python coding and Abaqus/CAE Tee modeling. On one hand, the development of the application is ease by the ability to record macro in Abaqus/CAE, but in the other hand, the development environment (PDE) is quite difficult to use for debugging and managing Python project.

The automation of parts design can be made when these parts can be parameterized. To offer better models, Python scripts can be used with parametric model to avoid topology problems.

Automation can help designers to save time on basic and repetitive tasks and let them focus on more advanced issues.