YAFEMS v.0.4

(Yet Another Finite Element Method Solver)

USER MANUAL

Javier Marcelo Mora

<javiermarcelomora@gmail.com>

CONTENTS

PREFACE

YAFEMS is born as a personal project after concluding my master studies of the Finite Element Method.

The goal of the project is implementing a Finite Element Method software capable of solving plane stress and plane strain 2D cases and 3D general cases, and allowing the use of already available software such as Salome for modeling, meshing and post processing tasks.

This is the main differential feature between this solver and most small academic solvers out there. YAFEMS is designed to work with Salome's mesh output and it generates a result file compatible with ParaView, which is included in Salome.

Another feature I wanted in the program was an easy to use input file format. YAF file format is a plain text file that is formed by very few lines describing the materials, load groups and boundary conditions groups. I wanted it to be as simple as possible, because using all the existing great free FEM software proved to be a major hassle editing and understanding every parameter that could go in the input file. Open FOAM and Code Aster come to mind here. They are extremely powerful, but creating an input file needs extended study of the program input syntax.

YAFEMS is programmed in Fortran. The standard I use is Fortran 95 but the Fortran version I learned was Fortran 77 so many of the features of Fortran 95 that could have made my program better aren't implemented. I'm still learning the “new” features of Fortran 95 as it's been a long time since I last programmed using Fortran.

WHAT'S NEW IN v 0.4 AND FUTURE IMPROVEMENTS

Version 0.4 new features:

- More performance boost due to the implementation of the Cuthill-McKee algorithm. This algorithm reorders the stiffness matrix matrix and severely reduces the bandwith. This leads to a huge decrease in time on the solving stage. If the number of elements is not too big, the solving time can actually increase a little, but those are the cases that solves in under a second, it shouldn't be a problem. The piston case sees a reduction in solving time from almost a minute to 9 seconds.

- Fixed some minor typos here and there.

- I've been trying to implement parallellization using OpenMP, but it seems that in the loops I've tried it actually takes more time to create the threads, execute them and then syncing then than the time it takes to do it with a single thread. This has lead to postpone parallellization for now. Some commented out code has been left in case I decide to give it a try in the future.

- Windows version released between 0.3 and 0.4.

FUTURE IMPROVEMENTS:

- Quadratic elements: This version of YAFEMS deals with linear triangles and linear tetrahedrons only. I'm planning to implement second order triangles and tetrahedrons. I'm not sure about implementing hexahedrons, as there aren't any free meshers out there that outputs hexahedron meshes. Netgen only deals with tetrahedrons, so I think my efforts will be focused on quadratic triangle and tetrahedron elements.

- Axisymmetrical, thin shell analysis: It'd be nice to be able to implement this kind of analysis.

- Some other type of analysis: Heat, large displacements, dynamic analysis...

- Trying to optimize for parallelization. See the changes in 0.4 to read about multi-core support.

1. COMPILING THE PROGRAM AND RUNNING NOTES

LINUX: The program should be fairly simple to compile. There are some prerequisites though:

- A Fortran compiler. I use gfortran, but it should compile with any standard Fortran compiler as the code doesn't use any non-standard features.

- libmed: This library is mandatory. It's used for reading and writing MED files. In Debian based distros, the lib is in the universe repo, so it shouldn't be a problem to install. The Fortran version and dev/header files are needed.

IMPORTANT NOTE: In my distribution (Linux Mint 17.1), I needed to copy file med_parameter.hf to /usr/include, as my source code includes med.hf that is installed by libmed library. This med.hf itself includes the file med_parameter.hf that my libmed package didn't install and compiling gives an error. The solution for me was to download the libmed source code from the Salome download page and copying med_parameter.hf to the /usr/include directory. I've include the med_parameter.hf in the /doc directory of YAFEMS so you don't have to download the whole libmed source code.

If compiling gives an error about missing med_parameter.hf, then just copy the file to /usr/include.

Compiling the code can be performed by executing the ./compile script, or directly by going to the /src directory and typing:

```
gfortran -O2 -o yafems_2d yafems_2d.f90 -I/usr/include -lmed
```

```
gfortran -O2 -o yafems_3d yafems_3d.f90 -I/usr/include -lmed
```

No special installation is needed. Just copy the yafems_2d and yafems_3d executable files to your preferred location.

WINDOWS: Since version 0.3 I include a Windows version compiled under Cygwin. All needed files are in the bin folder. I haven't include an installer, so just unzip the program in your desired location.

If you want to compile it yourself, a Cygwin installation can be used. You'll need to download MED source code from Salome website. Cygwin packages you'll need to install includes the gcc compiler, c++ compiler and gfortran and the libhdf5 library and dev packages as libMED depends on it. Then it's a matter of building MED package with the usual ./configure, make, make install procedure. NOTE: I needed to run ./configure --disable-python to be able to compile libMED.

There are three things that you need to take into consideration with the Windows version:

- YAFEMS executables needs to reside in the same directory as the included DLL. I've preferred not littering your system installing the DLLs, but then you need to have them in the same directory as yafems_2d.exe and yafems_3d.exe for them to run.

- YAFEMS is a command line application. That means that it doesn't have a GUI of any kind. This is true for almost all of the FEM solvers. You can execute the program opening a command prompt. If you don't want to do that, a very fast way of using YAFEMS is just dragging and dropping the yaf file on the YAFEMS exe file. This will execute the program with that yaf file as an argument.

- Due to the use of Cygwin, the output of the log and text result file is in the UNIX format (linefeed only). This means that if you open them with Notepad, the file will appear to be badly formatted. This a a Cygwin problem. There is a solution to this: Use a better text editor. Any decent text editor will recognize this and format the file correctly. I highly recommend Notepad++.

2. DESIGN STEPS IN SALOME

This section describes the way the geometry and mesh should be designed in Salome to obtain a good MED file that YAFEMS can read. Following these guidelines will ensure good results consistently. I've included some tutorials that will illustrate this method. See the tutorial section for detailed and graphical instructions.

ABOUT UNITS: Before going on, an important remark must be made about unit treatment in YAFEMS. YAFEMS doesn't convert units, just as almost all FEM solvers. This means that you should be consistent in your units. The easiest way is using the SI standard units: meters, Newton, Pascal, etc. Using another unit will force you to convert every other unit to be consistent between them.

The design process includes two main steps: Geometry and Meshing:

-Geometry: Using the Geometry module in Salome, we start our design creating the points that outlines our figure. For yafems_2d, the points MUST be on the X-Y plane, as YAFEMS will expect the coordinates to be X and Y, so using the TOP view will prove to be very useful. For yafems_3d, it doesn't matter how the geometry is defined. Just note the chosen axis to correctly apply loads.

After defining the points, we need to draw the lines/curves that join the points. We should have a closed geometry by now.

Next step is defining a surface using the Build Face command. A planar face will be created after selecting the lines we built before. For surfaces with holes, a trick can be used: instead of performing a boolean operation after the face is created to subtract the hole from the geometry, we can begin defining the face by selecting the hole lines first, and then selecting the outer lines. When the face is created, the holes will be subtracted automatically.

One distinct face should be created for each face that has a different material.

After all faces are created, we need to join them into one object using the Build Compound command (if more than one face exists). This will give us a compound object that include every face.

Last step is creating groups. Groups are needed as they will be where loads and boundary conditions will be applied. In every case, we use the Create Group command, select the type of entity, click the corresponding object and click ADD to create the group. It's important to notice that geometry elements must be explicitly defined in order to be able to select them as groups. We can create three kind of groups:

- Node groups: We use them to assign punctual loads and BCs to our geometry. Points defined in the first step while creating the geometry MUST be used. This is important, as the Node Groups will need to be part of the mesh. If the point is not part of the mesh, YAFEMS will fail to give a correct solution. See Example 3 in the tutorial section.

- Line groups: Distributed loads and fixed lines as boundary conditions can be defined selecting complete lines in the Create Group command.

- Face groups: A group must be created for each face in order to assign materials later for yafems_2d or boundary conditions and loads for yafems_3d.

- Volume groups: A group must be created for each volume in order to assign materials later for yafems_3d. Don't create volumes if using the 2d solver.

This ends the geometry step. We should have one and only one Face or Compound object with node groups, line groups and face groups where we want to apply our loads and BCs later.

-Meshing: Now we enter the mesh module of Salome. The goal here is creating the mesh and replicate the groups created before checking that they are indeed nodes of the resulting mesh.

First step is creating the mesh using the Create Mesh command. The Face for 2d solver or Volume for 3d solver or Compound object needs to be selected. The recommended mesh algorithm is Netgen 1D-2D under the 2D tab for 2D models or Netgen 1D-2D-3D for 3d objects. IMPORTANT NOTE: Only triangles and tetrahedrons are processed in this version of YAFEMS, so be sure that Allow Quadrangles is NOT selected under the hypothesis configuration menu.

After the mesh is created, we need to recreate the groups defined earlier in the geometry step. Using the Create Groups From Geometry, we can just reselect the existent groups and reassign them. Select the line, face groups and volumes in the element box and the nodes group in the node box.

Volume, face and line groups will be correctly defined, but node groups can sometimes be problematic, so I recommend checking the mesh and seeing if the node groups is part of the actual mesh. If it is not, a node group can be created with the Create Node Group and a node on the mesh can be directly selected. See Example 3 in the tutorial section for a detailed example.

After all groups are created, we need to export the mesh. Right click on the mesh name and export it to MED format.

3. YAFEMS INPUT FILE AND EXECUTION

YAFEMS input file format is designed to be very easy to understand and use. It consists of a series of lines describing the mesh file, analysis basic data, material data, loads and boundary conditions.

The format is as follows:

- First line: Name of the mesh file used in the analysis.
- Second line: Five integers that represent for yafems_2d solver or four for yafems_3d solver:
  - 1 for plane stress, 2 for plane strain (do not include if using yafems_3d).
  - Number of different materials in model.
  - Number of groups in compound. This number should be equal to the number of faces that a compound object is formed by if a compound exists or 1 there's no compound object and only one face exist.
  - Number of boundary conditions. Number of groups in mesh where a BC will be applied.
  - Number of loads: Number of groups in mesh where a load will be applied.
- One line per material: Three real numbers per line for yafems_2d or two for yafems_3d. For each different material, a line should be added that includes:
  - Element thickness (do not include if using yafems_3d).
  - Modulus of elasticity.
  - Poisson coefficient.
- Two lines per number of groups in compound.
  - Line one: Name of the face group (yafems_2d) or volume group (yafems_3d) in mesh.
  - Line two: Number of material assigned to the face or volume group. Numbering corresponds to the order in which materials are defined above.
- Two lines per number of Boundary Conditions.
  - Line one: Name of BC group in mesh.
  - Line two: Restrictions in X and Y (and Z for yafems_3d) global directions. 1=Restricted, 0=Free.
- Two lines per number of loads.
  - Line one: Name of load group in mesh.
  - Line two: Two for yafems_2d or three for yafems_3d real numbers and one integer number. Load component in X and Y (and Z) global directions. If the load is punctual, force should be in force units, while if the load is distributed in a line in 2d cases or face in 3d cases, force/surface unit should be used. If load is defined in a line in a 3d case, force/length shpuld be use. NOTE for 2d cases: if you want to use a pure linear load, you'll have to divide the linear load value by the thickness of the element. The last number is used to define the material the load is acting upon. This number only applies to distributed loads and is needed in the cases when the line where the load is applied is shared between several elements that have different materials.
- END. Every YAF file ends with an END statement. Everything after it will be ignored, so

comments can be put after the END statement.

A yaf file example for yafems_2d follows. Comments have been added to the right for clarity's sake. :

```
test.med                ; MESH file name
1 2 3 3 2               ; Plane stress, 2 materials, 3 Faces, 3 Bcs, 2 loads.
0.1 210 0.3             ; Material 1: 0.1 thickness, E=210, Poisson=0.3
0.4 150 0.3             ; Material 2: 0.4 thickness, E=150, Poisson=0.3
Face_1                  ; Face_1 is the name of the face group in mesh
1                       ; Material 1 is assigned to it
Face_2                  ; Face_2 is the name of the face group in mesh
2                       ; Material 2 is assigned to it
Face_3                  ; Face_3 is the name of the face group in mesh
2                       ; Material 2 is assigned to it
BC_1                    ; BC_1 is the name of the bc line/node group in mesh
1 1                     ; Restriction in X and Y
BC_2                    ; BC_2 is the name of the bc line/node group in mesh
1 0                     ; Restriction in X but Y is free
BC_3                    ; BC_3 is the name of the bc line/node group in mesh
0 1                     ; Restriction in Y but X is free
Load_1                  ; Load_1 is the name of the load group in mesh
100 0 1                 ; Load of 100 in X, applied to material 1
Load_2                  ; Load_1 is the name of the load group in mesh
0 -500 2                ; Load of -500 in Y, applied to material 2
END                     ; END of yaf file
```

Files are usually even simpler, as only one material is defined most of the time.

```
test.med                ; MESH file name
1 1 1 2 1               ; Plane stress, 1 material, 1 Face, 2 Bcs, 1 loads.
0.1 210 0.3             ; Material 1: 0.1 thickness, E=210, Poisson=0.3
Face_1                  ; Face_1 is the name of the face group in mesh
1                       ; Material 1 is assigned to it
BC_1                    ; BC_1 is the name of the bc line/node group in mesh
1 1                     ; Restriction in X and Y
BC_2                    ; BC_2 is the name of the bc line/node group in mesh
1 0                     ; Restriction in X but Y is free
Load_1                  ; Load_1 is the name of the load group in mesh
100 0 1                 ; Load of 100 in X, applied to material 1
END                     ; END of yaf file
```

A yaf file example for yafems_3d follows. Comments have been added to the right for clarity's sake. :

```
test.med                ; MESH file name
2 3 3 2                 ; 2 materials, 3 Volumes, 3 Bcs, 2 loads.
210 0.3                 ; Material 1: E=210, Poisson=0.3
150 0.3                 ; Material 2: E=150, Poisson=0.3
Volume_1                ; Volume_1 is the name of the volume group in mesh
1                       ; Material 1 is assigned to it
Volume_2                ; Volume_2 is the name of the volume group in mesh
2                       ; Material 2 is assigned to it
Volume_3                ; Volume_3 is the name of the volume group in mesh
2                       ; Material 2 is assigned to it
BC_1                    ; BC_1 is the name of the bc face/line/node group
1 1 0                   ; Restriction in X and Y and Z is free
BC_2                    ; BC_2 is the name of the bc face/line/node group
1 0 1                   ; Restriction in X and Z but Y is free
BC_3                    ; BC_3 is the name of the bc face/line/node group
0 1 0                   ; Restriction in Y but X and Z are free
Load_1                  ; Load_1 is the name of the load group in mesh
100 0 0 1               ; Load of 100 in X, applied to material 1
Load_2                  ; Load_1 is the name of the load group in mesh
0 0 -500 2              ; Load of -500 in Z, applied to material 2
END                     ; END of yaf file
```

Files are usually even simpler, as only one material is defined most of the time.

```
test.med                ; MESH file name
1 1 2 1                 ; Plane stress, 1 material, 1 Face, 2 Bcs, 1 loads.
210 0.3                 ; Material 1: E=210, Poisson=0.3
Volume_1                ; Volume_1 is the name of the volume group in mesh
1                       ; Material 1 is assigned to it
BC_1                    ; BC_1 is the name of the bc line/node group in mesh
1 1 1                   ; Restriction in X, Y and Z
BC_2                    ; BC_2 is the name of the bc line/node group in mesh
1 0 1                   ; Restriction in X and Z but Y is free
Load_1                  ; Load_1 is the name of the load group in mesh
100 0 0 1               ; Load of 100 in X, applied to material 1
END                     ; END of yaf file
```

Execution of YAFEMS is very simple. For 2D case:

```
yafems_2d input.yaf
```

Or for 3D case:

```
yafems_3d input.yaf
```

If everything goes well, no message will be displayed and three files will be created: input.log (log file), input.result.txt (result text file) and input.rmed (rmed result file for opening in ParaView).

Postprocessing in ParaView is simple. Just open the rmed file.

4. TUTORIAL EXAMPLES

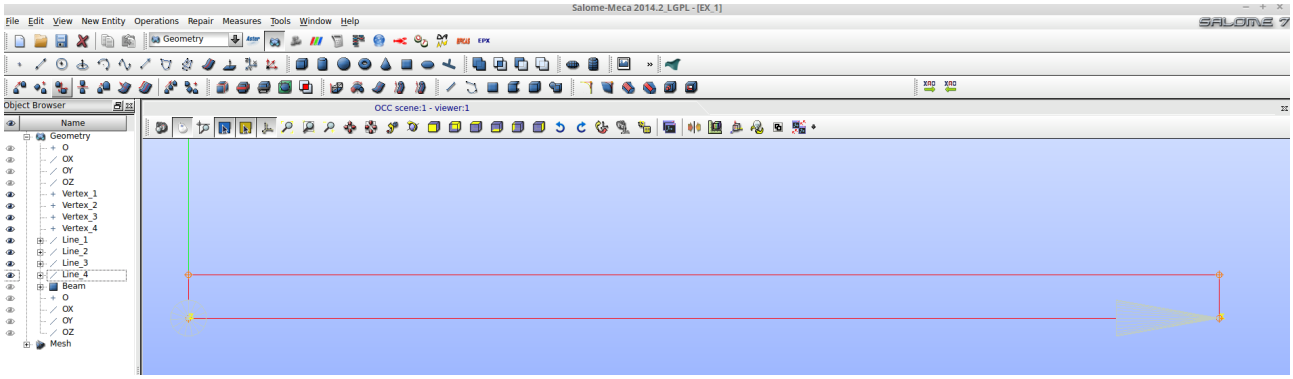
I've included 6 examples to provide an easy way of learning the way all parts come together. In the Examples folder there are three files for each example. The .hdf file is the Salome file, the .med file is the mesh exported from Salome and the .yaf file is the input for YAFEMS.

4.1 EX_1: 2D CANTILEVER BEAM

The first example is a simple cantilever beam, but it will serve us to illustrate the design steps in Salome that will lead to an appropriate mesh.

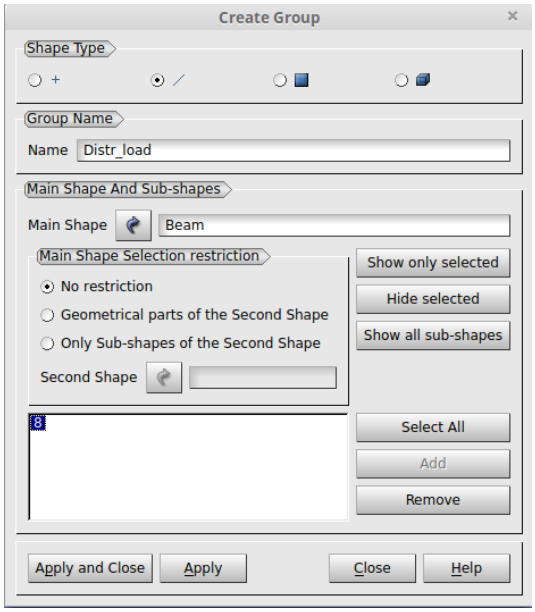
The beam has a rectangular cross section with a height of 0.5 m and a width (thickness) of 0.32 m. Length is 12 meters. We'll consider a punctual load and a distributed load and study two cases using the same mesh.

The first step is defining the four vertices of the beam, and the four lines that join them in the geometry mode. This will form a rectangle of 12x0.5m:



Next step is creating the face using the build face command and selecting all four lines. We name the face Beam.

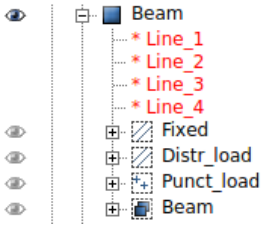
After creating the face, we need to define the groups. Using the Create Group command, we select the face option, select the surface we created, name the group as we want (Beam in the EX_1.hdf file provided) and click on Add, then click on Apply. Without closing the Create Group window, we select the line option, name the group Distr_load, click on the top line of the rectangle, click Add (a number appears in the window) and then Apply.



Same procedure is used for the boundary condition. With the line shape type selected, we name the group as Fixed, select the leftmost line and click Add.

Last group is the punctual load group. We select the point shape type, click on the top right vertex of the rectangle, click Add, name the group as Punct_load and click Apply and Close as no more groups need to be defined.

The groups should be look as follows:

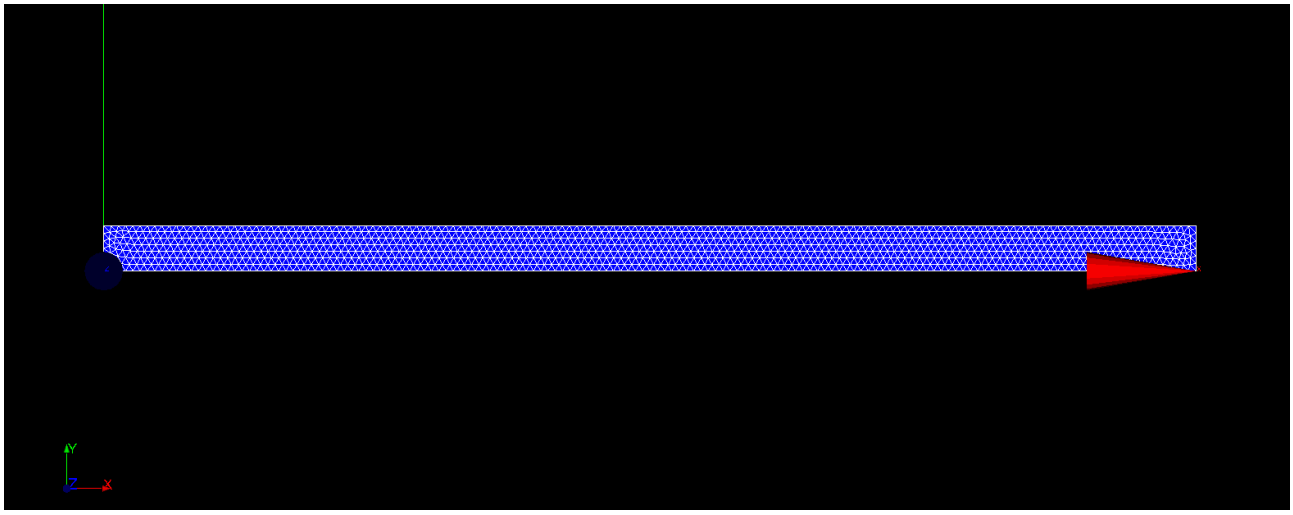


We can see all the groups defined. This groups will be used in the meshing process that follows.

Now we go to the mesh module and we create a mesh with the Create Mesh. It is very important that the geometry selected is the Beam face that includes all the groups and geometry. If we had several faces, we should join them in the geometry module using a compound object. The goal here is having a single element to mesh that includes all sub elements. Here we have one face only.

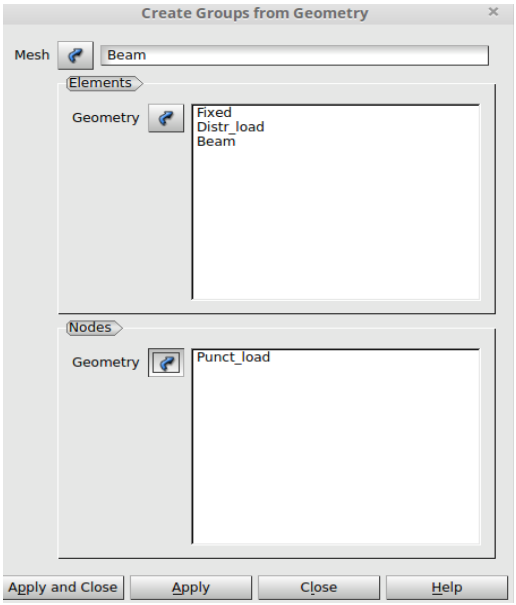
In the create mesh window, we select the Beam face element as Geometry and use Beam as name. 2D tab should be selected by default. It's highly recommended to use the Netgen 1D-2D algorithm as it's the one used for all the tests and works perfectly fine. Select the dented wheel in the Hypothesis section, and put the values you prefer for meshing. DON'T click on Allow Quadrangles and Second Order as only linear triangles are processed by YAFEMS.

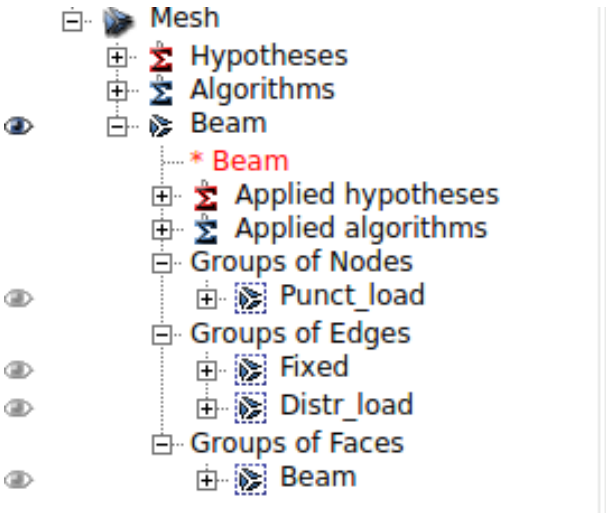
The mesh is ready to be created. Right click on the Beam mesh object with the exclamation sign and click Compute. You should get a meshed rectangle:



Last step is importing the geometry groups defined earlier in the geometry module. Use the Create Groups from Geometry command and control-select the line and face groups while the element selection is active and the node group while the nodes element selection is active.

You should get something similar to the following picture, along the mesh groups once you click Apply and Close:





Last step is right-clicking on mesh Beam and Export it to MED format.

After all the design steps, the YAF input file should be created.

Two YAF files are included, one for the punctual load case (EX_1a.yaf) and one for the distributed load case (EX_1b.yaf).

EX_1a.yaf:

```
EX_1.med
1 1 1 1 1
0.5 210000000000 0.3
Beam
1
Fixed
1 1
Punct_load
0 -150000 1
END
```

The first line tells the mesh filename (EX_1.med), the second line says that it's a plane stress case, with one material and one face, one boundary condition and one load. Third line says the element has a thickness of 0.32 m, E=2.1E11 Pa, ν=0.3. Fourth line says the Face group is called Beam, and fifth line says that the face Beam is assigned material 1. Sixth and seventh lines define the a fixed in X and Y boundary condition on the group Fixed and a load in the negative Y direction of the group Punct_load of -150000 N value. It'll be a punctual load as the group was a Node group as we defined it earlier in Salome. The number 1 only serves in the case of a distributed load that shares the line where it's applied so it's of no use here, but must be entered. EX_1b.yaf only differs in the load definition and has these two lines instead:

```
Distr_load
0 -468750 1
```

This means a load is going to be applied in the Distr_load group. As the group is a line, YAFEMS will take the load value and distribute it. The value is 468750 N/m^2. As the thickness is 0.32 m, the linear load is equivalent to 468750*0.32=150.000 N/m or 150 kN/m.

Now we can run YAFEMS (we are going to use the distributed load file in this tutorial. You can check the punctual load for yourself):

```
yafems_2d EX_1b.yaf
```

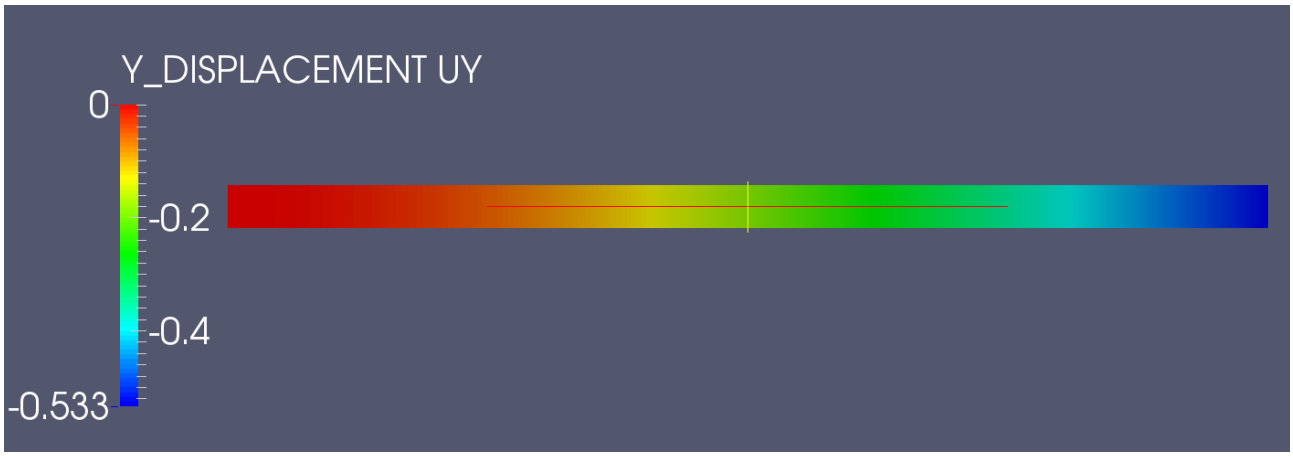
We should get no errors and three new more files:

- EX_1b.log: Log file with information about the geometry, nodes, segments and triangles in MED file. It also includes the boundary conditions and loads applied and distributed in nodes even if we defined the groups to be lines.

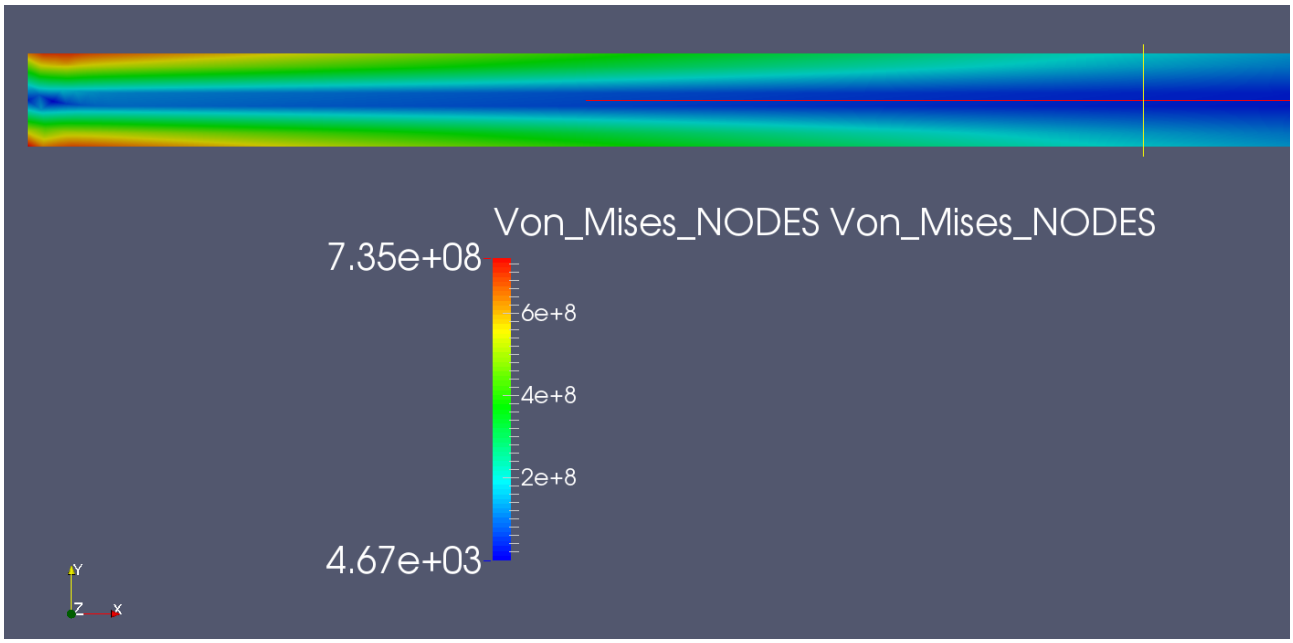
- EX_1b.result.txt: This file includes every calculation in every node and element. Results are: Displacements, stresses (SX, SY, TXY), Von Mises equivalent stress, and reaction forces. Stresses are calculated in elements and nodes. The nodes stresses are obtained using the mean of the stresses in elements to which the node belongs to.

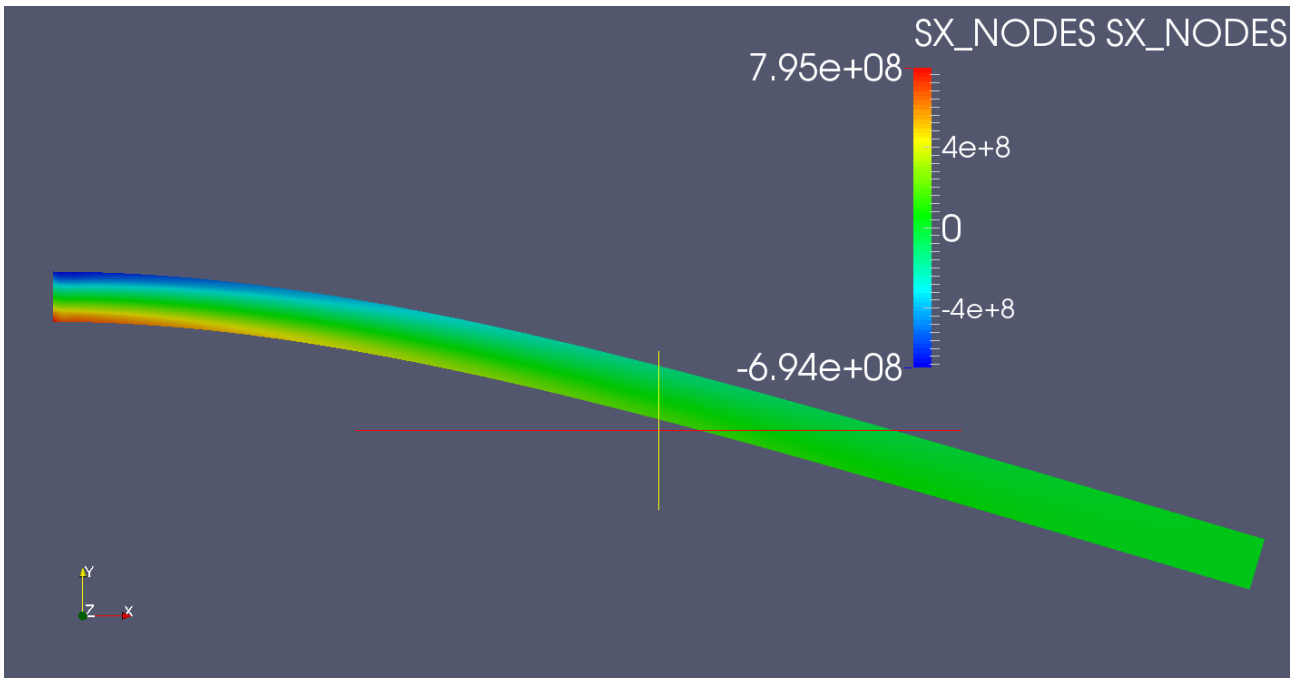
- EX_1b.rmed: Result file in MED format for using in ParaView inside Salome.

Now we proceed to open the rmed file in ParaView. All calculated parameters can be shown:



We see that the maximum displacement is -0.533 m at the end of the beam. Basic beam theory for a cantilever beam with a distributed load says the Y deflection is -0.55 m.





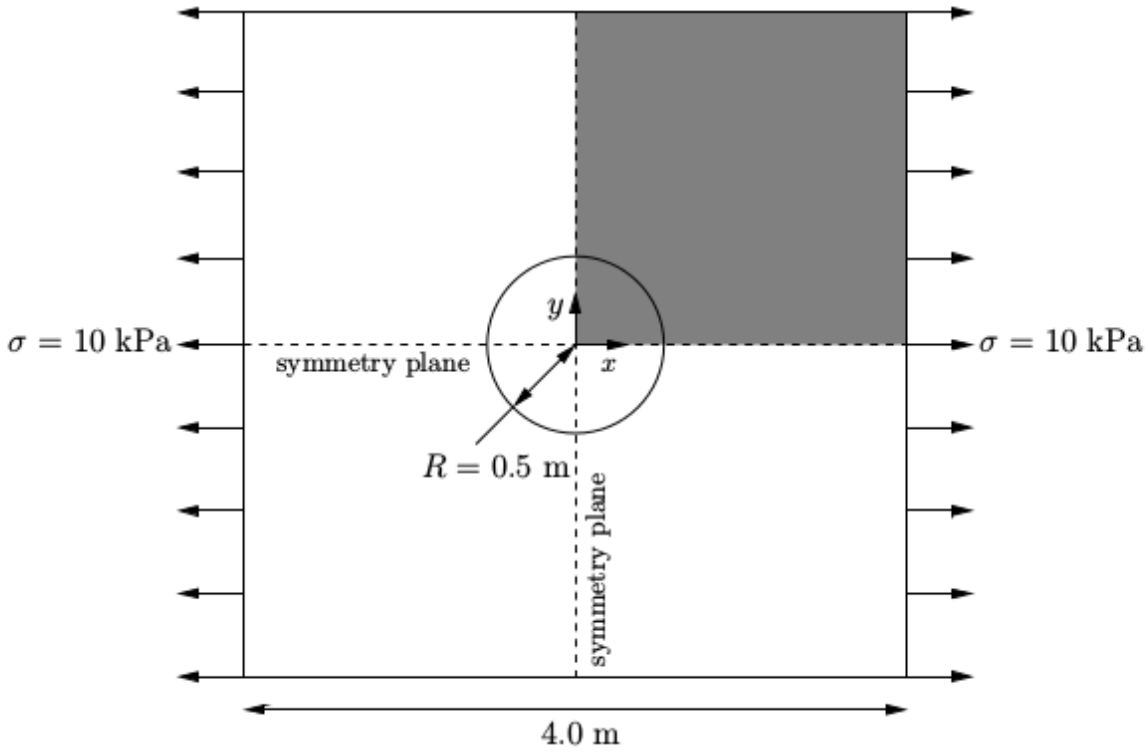
Theory says that $\sigma = \frac{M \cdot y}{I} = \frac{10800000 \cdot 0.25}{3.333\text{E-}3} = 8.1\text{E}8\,Pa$

The approximation is quite good. 10.800.000 N*m is the reaction moment at the fixed end calculated using beam theory.

4.2 EX_2: 2D PLATE WITH HOLE

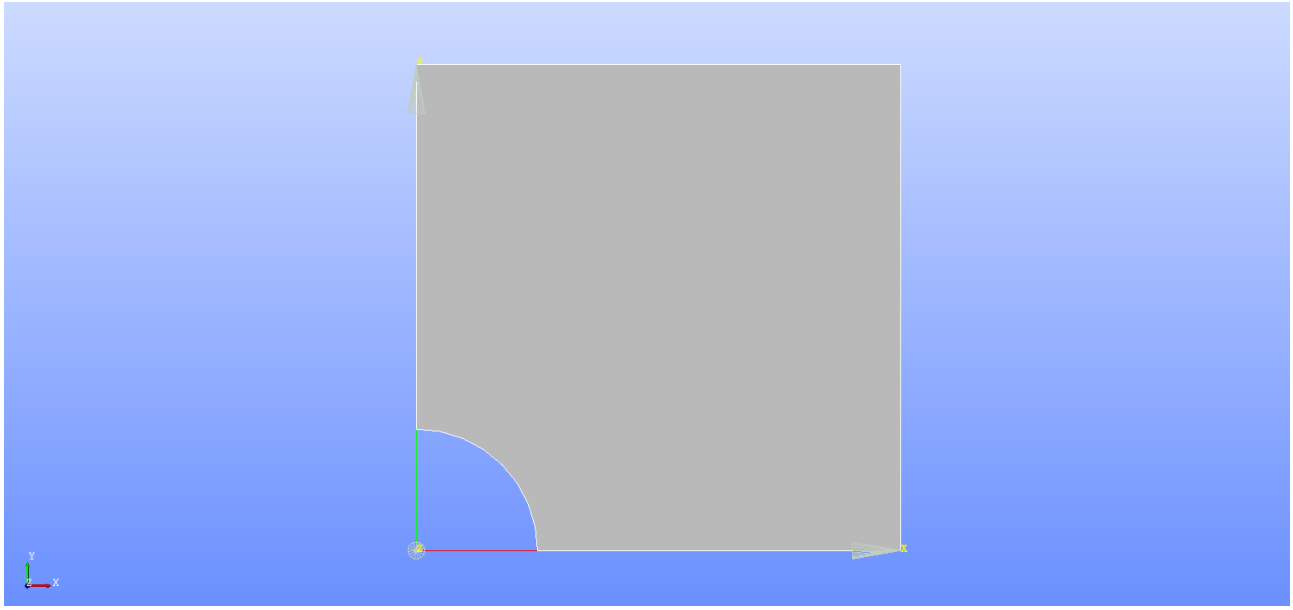
Files for this case are EX_2.yaf and EX_2.med. The Salome file is EX_2.hdf.

The next example is taken from the OpenFOAM documentation. It's a standard plane stress problem that appears in almost all elasticity manuals. The picture of the problem is:



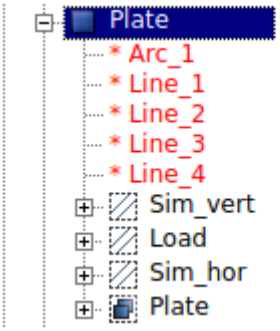
The problem can be modeled using a quarter of the plate, as it has two simmetry axis.

The same exact procedure as before is followed. We define the points first, using three points to define the center and limits of the quarter of circumference of the hole. Then we define the lines and arc and we create a face.

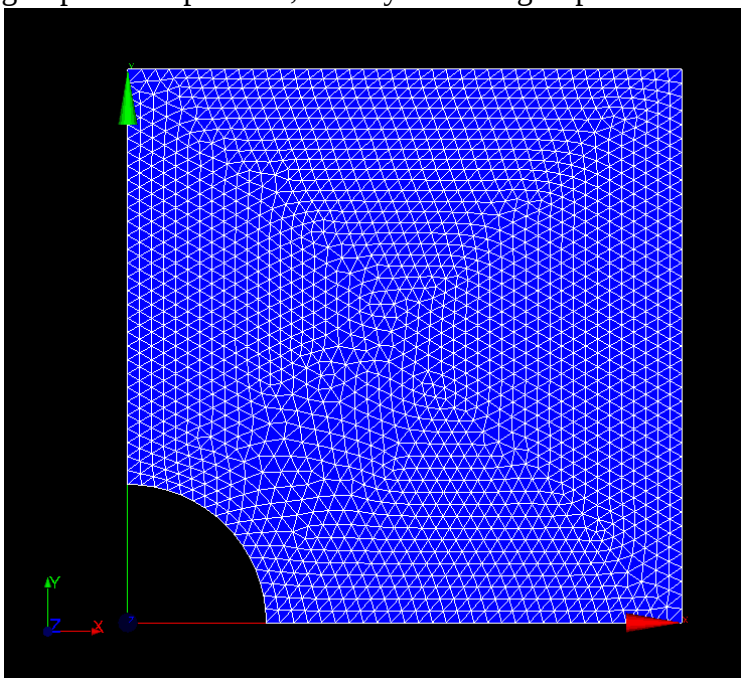


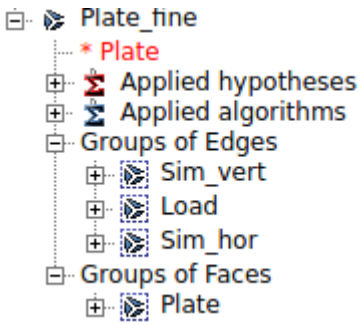
We define the groups we need that are two boundary condition lines (left line and bottom line), load line (right line) and face group using the whole plate.

Groups should look as follows:



We mesh the Plate face using
Netgen 1D-2D without quadrangles and Create Groups from Geometry. We don't have any Node groups in this problem, so only element groups need to be created.





Last step is exporting the mesh to med file. Note that in the included EX_2.hdf file there are two meshes. One is called Plate_coarse and the other is called Plate_fine. You can create several versions of the mesh and export the one you want.

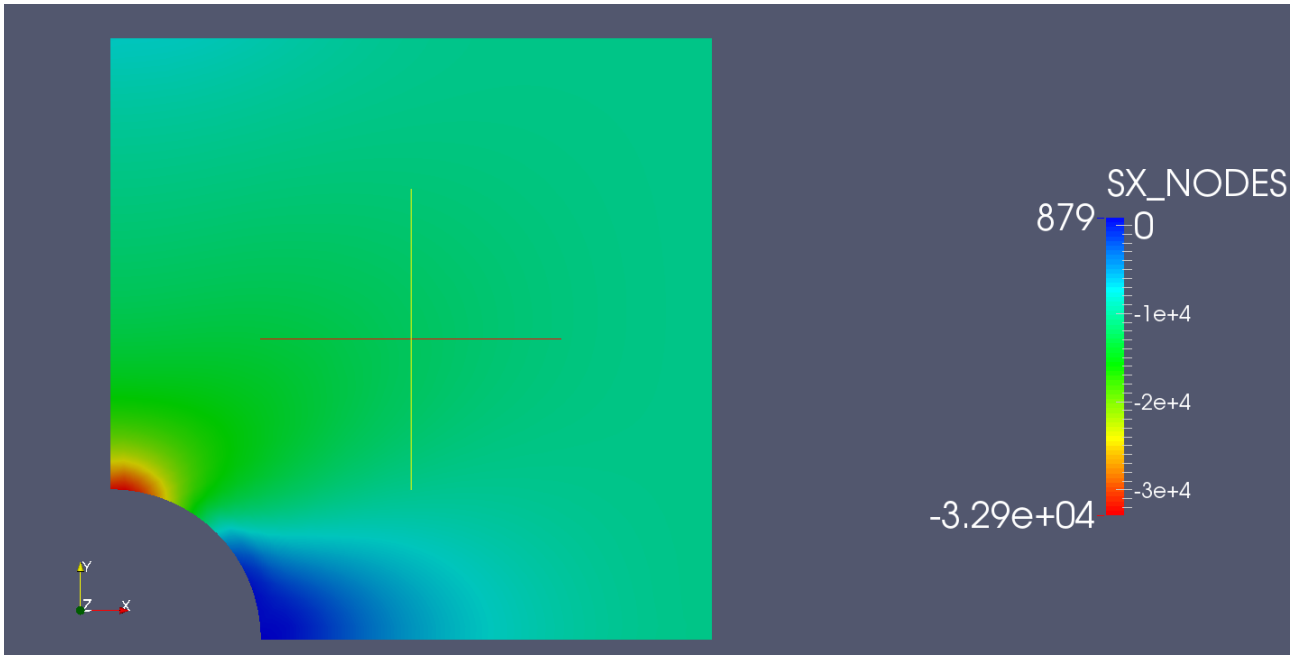
The yaf input file is quite simple:

```
EX_2.med
1 1 1 2 1
0.5 200000000000 0.3
Plate
1
Sim_vert
1 0
Sim_hor
0 1
Load
10000 0 1
END
```

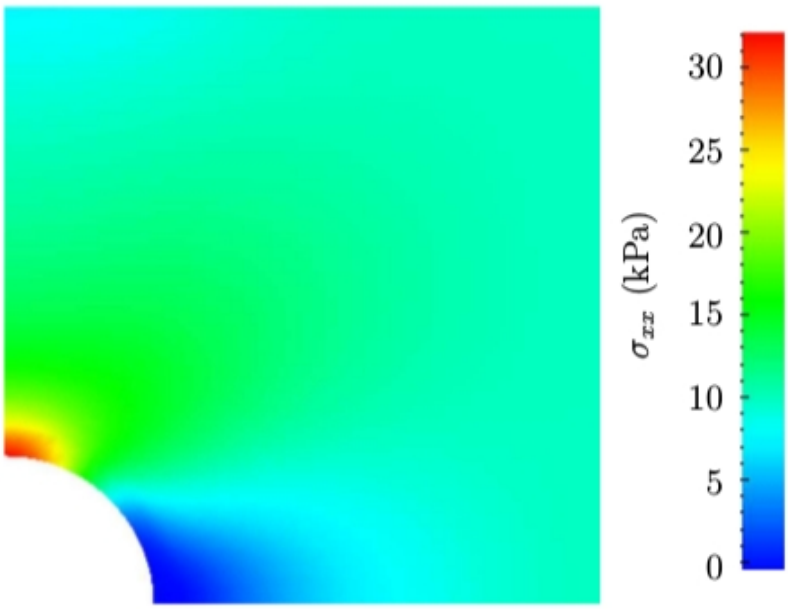
We execute YAFEMS:

```
yafems_2d EX_2.yaf
```

Now we can open the EX_2.rmed file in ParaView and check the results:

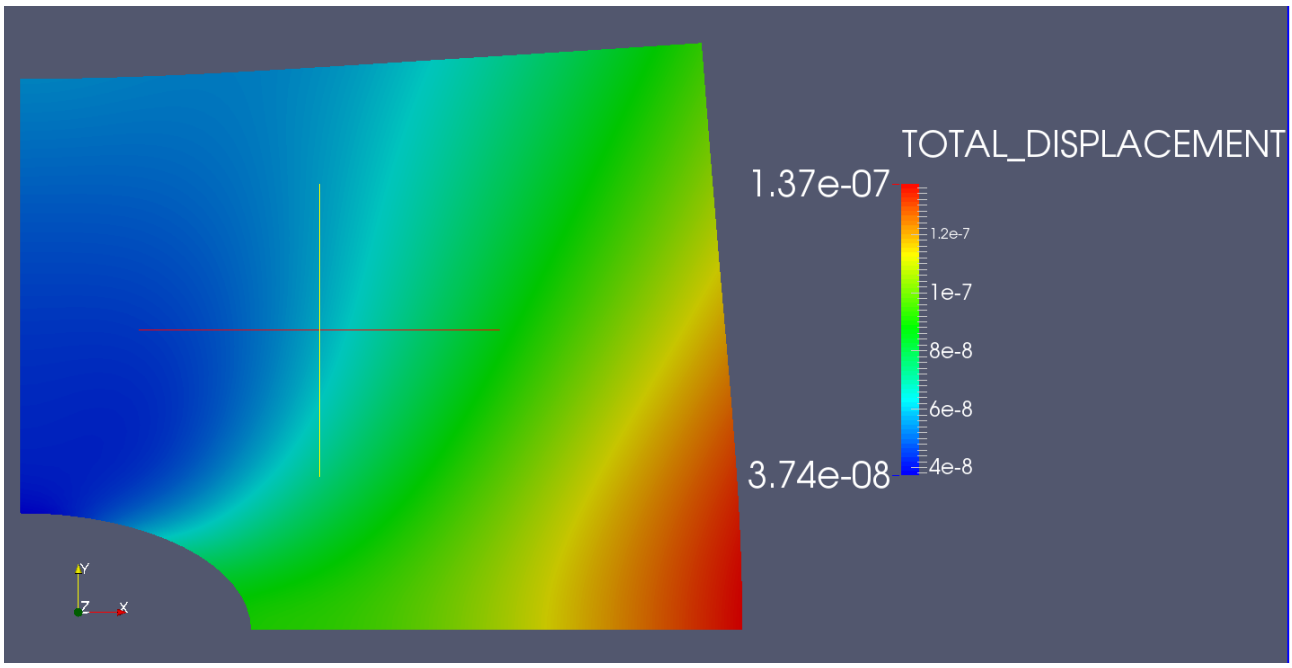


The SX stress is 3.24E4 N/m². According to OpenFOAM calculations:



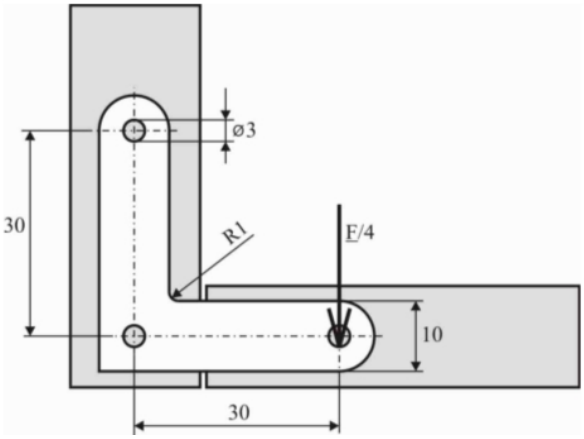
As we can see, the results are very close, as OpenFOAM gives a maximum SX of 32 kPa.

Displacement field is as follows:


TOTAL_DISPLACEMENT
1.37e-07
1.2e-7
1e-7
8e-8
6e-8
4e-8
3.74e-08
Y
Z
X

4.3 EX_3: 2D STRAP PLATE

This example is a recreation of an ANSYS plane stress tutorial by István Oldal. The geometry is as follows:

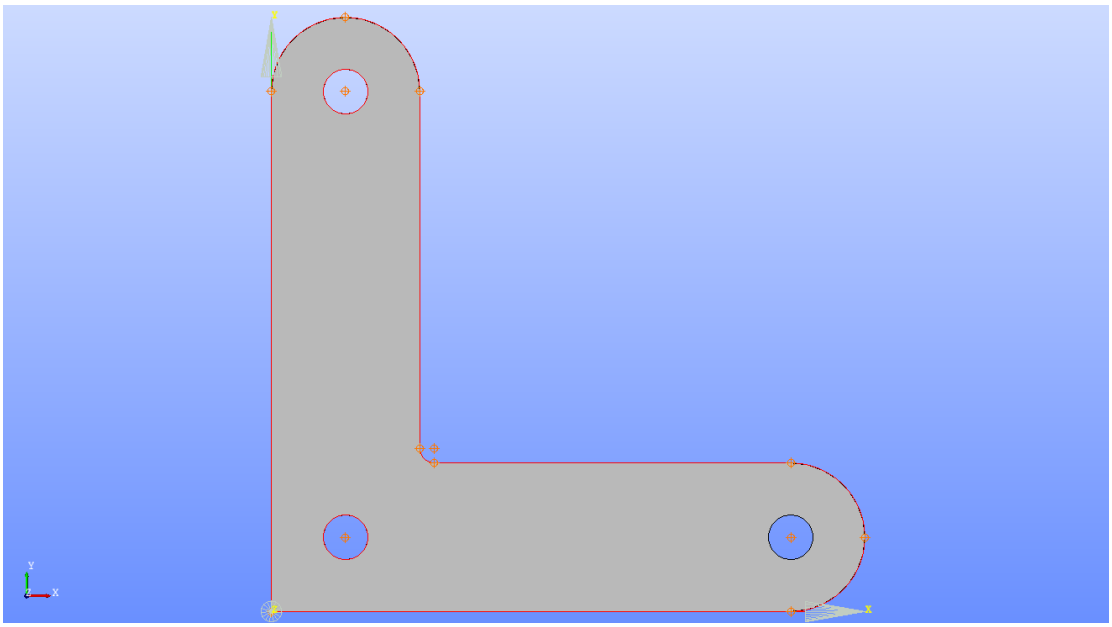


I've modeled this problem using millimeters, so every other compound unit (N, Pa) should be adapted to milliliters too.

This exercise introduces one interesting problem in the modeling process.

In the geometry module we begin defining points as usual. We create points for the center of the circles too. According to the guidelines in section 2, we should need to create a point in the lower part of the rightmost circumference where the load is applied. This was my first idea when I was modeling the geometry. Defining the center, the lower point and another point in the circumference should be enough to be able to use the lower point later when creating the Node group to apply the load and creating the circumference. THIS LEADS TO BAD RESULTS. The error occurs because when meshing, Netgen doesn't use the node as a vertex of a triangle. In order to avoid the problem, we won't create a Node group now. We'll talk about the solution in the meshing step.

After defining all points, lines and face, we should have the following geometry:



TIP: The best way for defining the face in one single step is:

- Select the Create Face command.

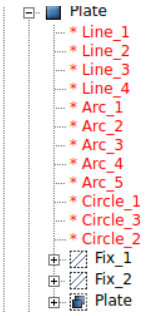
- Start by selecting the three circumferences FIRST (of course, use and hold Shift for multi-selecting)

- After all three circumferences are selected, keep selecting the outer lines and arcs.

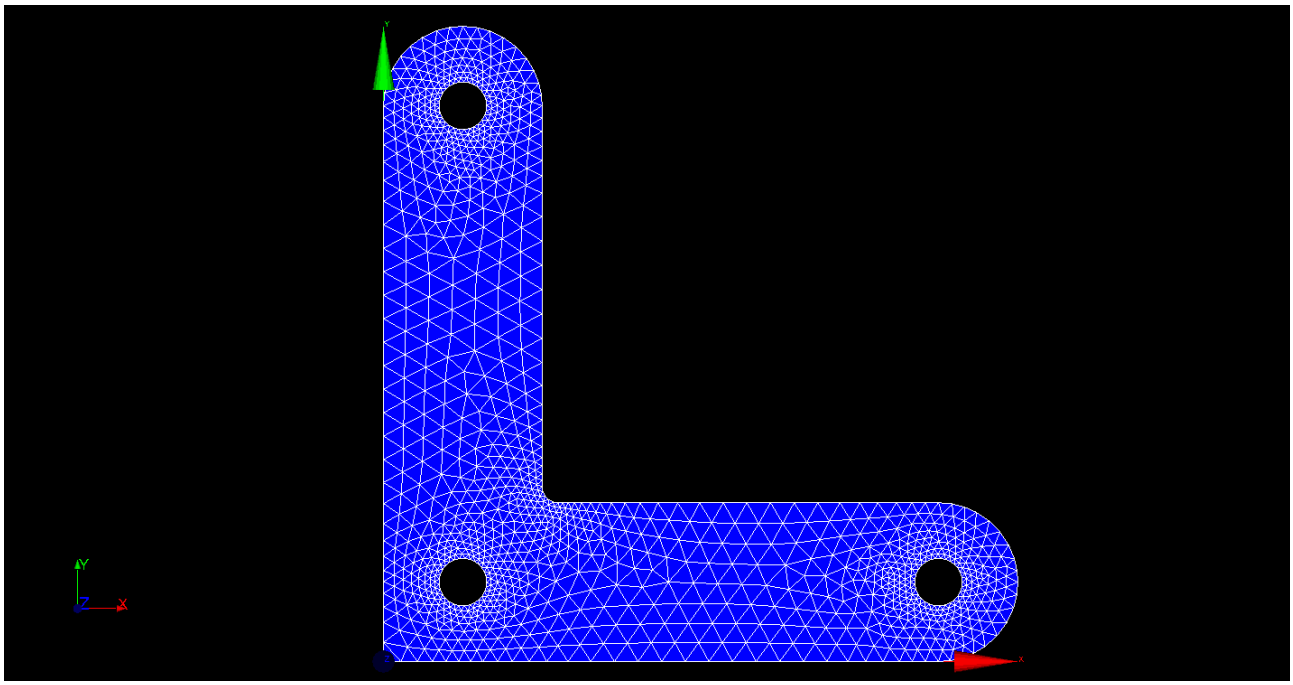
- Click Apply

Face should be correctly created, with holes in the circumferences, and not the opposite.

Next step is defining the groups. As was pointed before, we are not going to define the Node group where the force will be applied now, but later in the meshing step. We do create two line groups (Fix_1 and Fix_2 in using the two leftmost circumferences and one face group usin the plate).



Going into the meshing step, we mesh the plate with Netgen 1D_2D as usual:



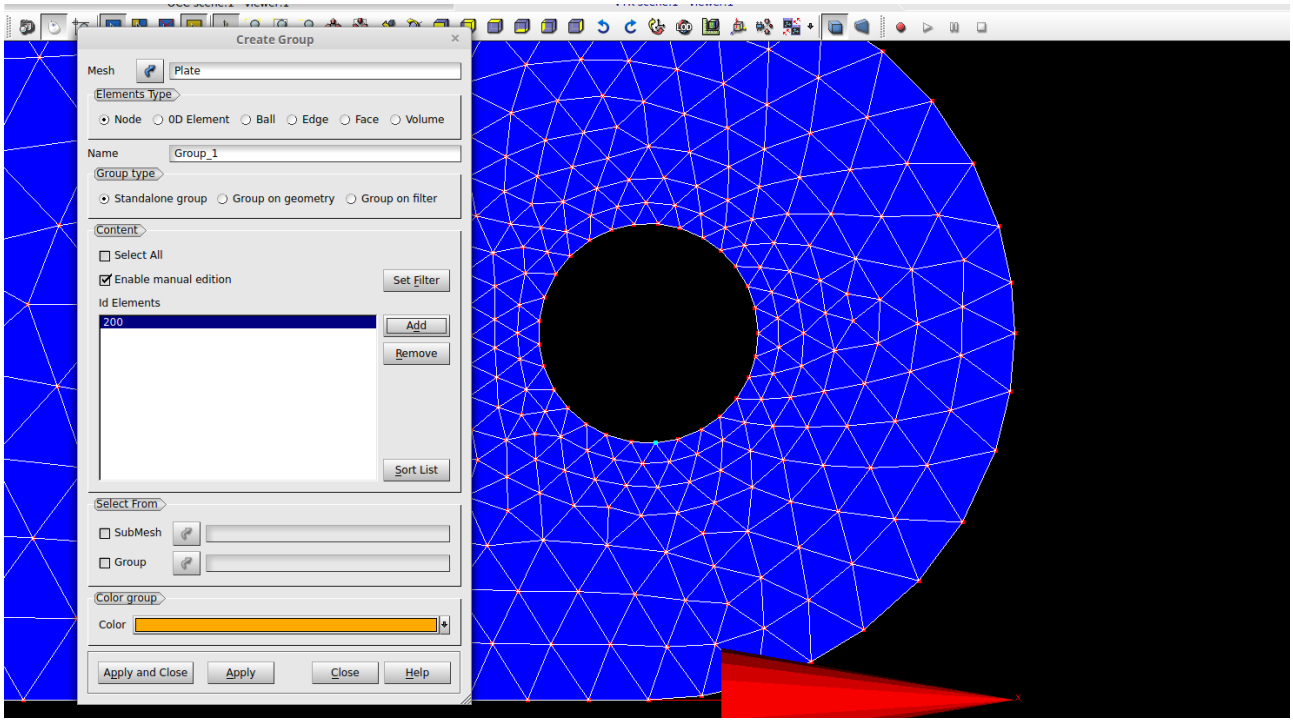
Now we Crete Groups from Geometry importing all three groups (Fix_1, Fix_2 and Plate). As we pointed out, we still need to create the Load Group as we didn't define it before.

Still in the Mesh module, we go to Create Group, select Node in element type, enable Manual selection and manually click the node where the load will be applied. Then we click Add and Apply.

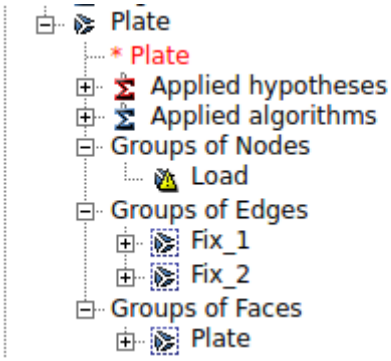
The node group should be created. An exclamation mark will appear in this group, but it is completely normal.

Using this procedure we guarantee that the node groups will really be part of the mesh.

The following picture illustrate the process (the selected node is in blue in the picture):



The mesh groups should be similar to this:



After exporting the Plate mesh, we proceed to create the yaf input file. We have two boundary conditions in the two leftmost circumferences, the punctual load and the Plate face group. Material used is steel, and the element thickness is 1.5 mm.

YAFEMS input file is:

```
EX_3.med
1 1 1 2 1
1.5 210000000 0.3
Plate
1
Fix_1
1 1
Fix_2
1 1
Load
0 -50000 1
END
```

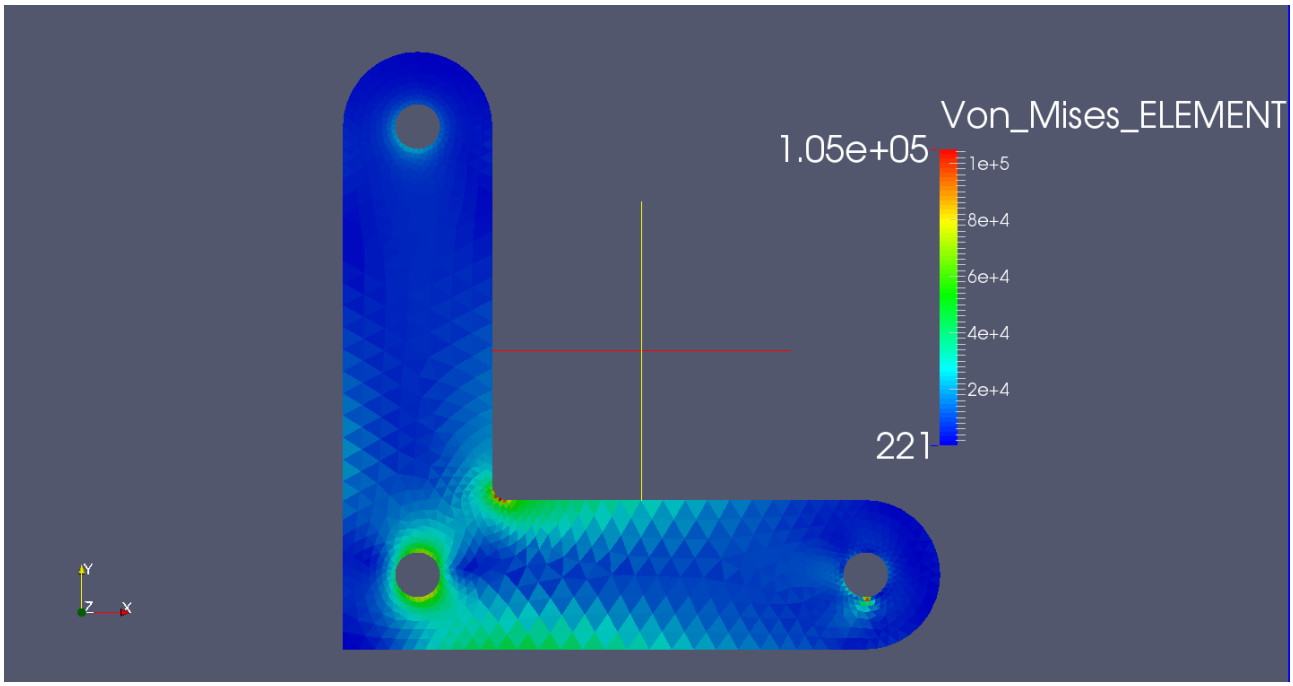
We run YAFEMS:

```
yafems_2d EX_3.yaf
```

Now the postprocessing can be done in ParaView.

In this example, values vary from the original analysis made by ANSYS. I suspect that something is different between the two models, as all other examples give very similar results to the original calculations. Von Mises stress in Elements, is nearer to the ANSYS value than Von Mises in Nodes. Von Mises in Nodes is calculated using the mean between all surrounding elements for every node, so I think that having extreme values in a very localized location can lead to this differences.

Anyway the solution is not too far away, but still it's not as close as the other examples.



ANSYS results:

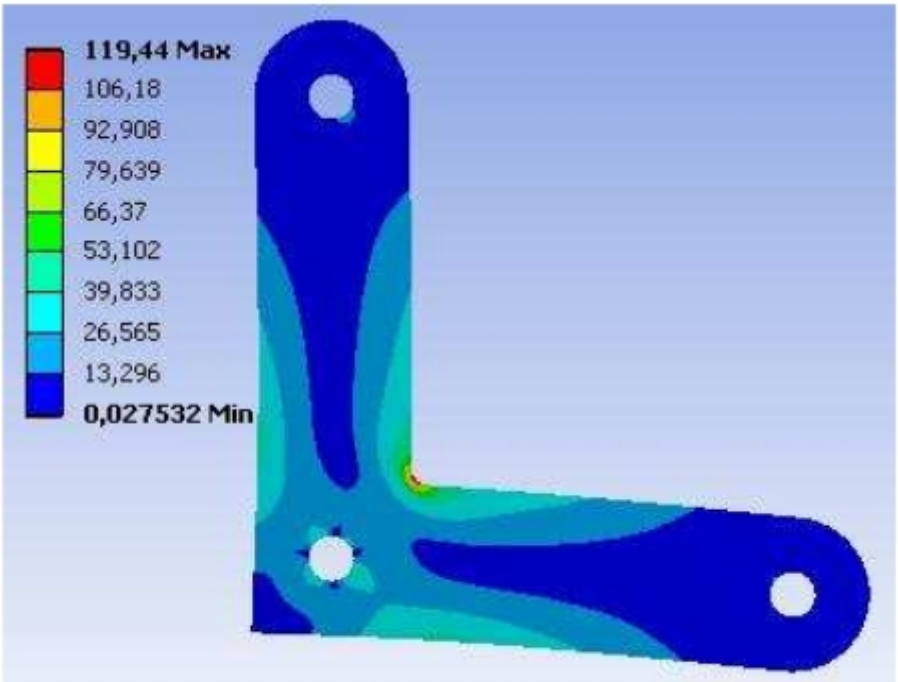
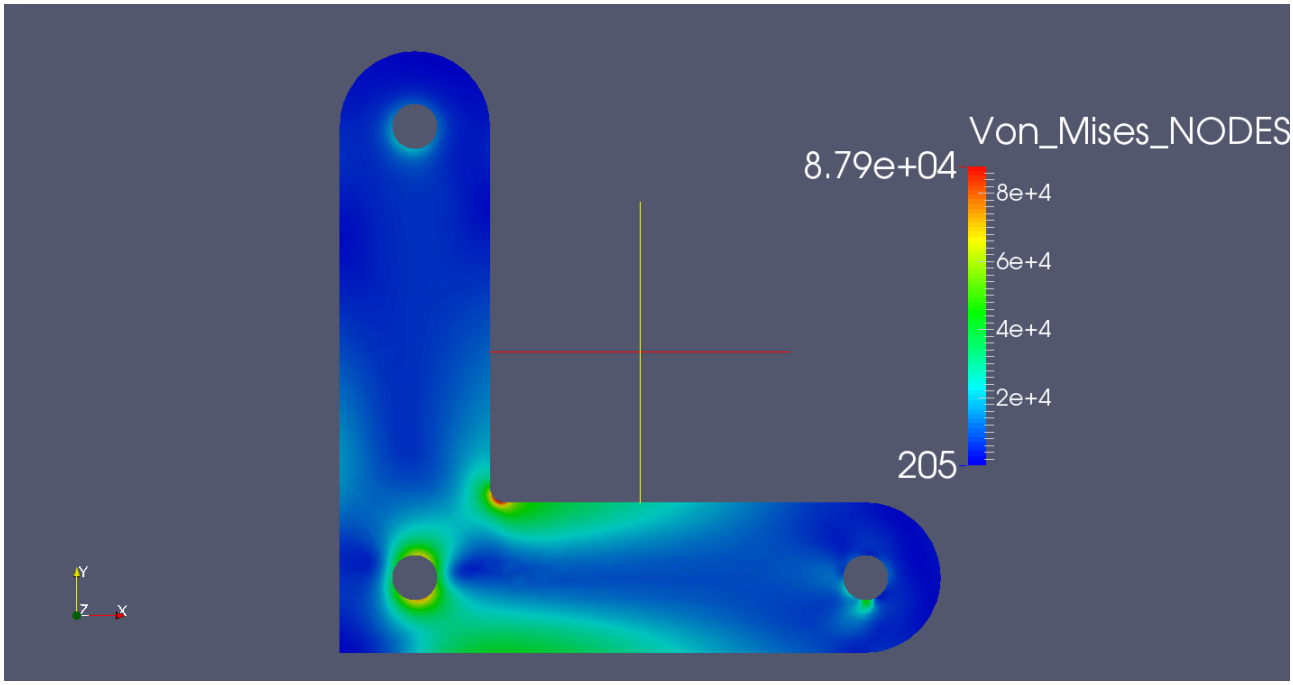
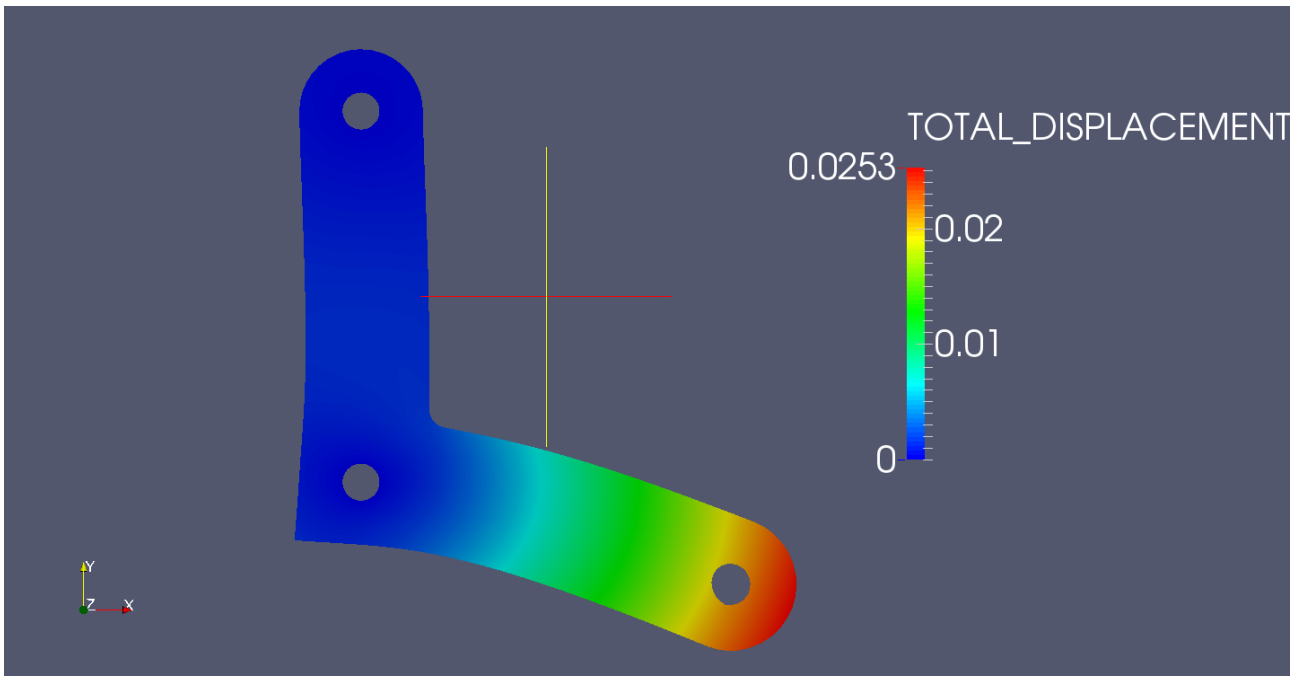


Figure 7.29: Calculated stresses [MPa]

Von Mises in nodes:

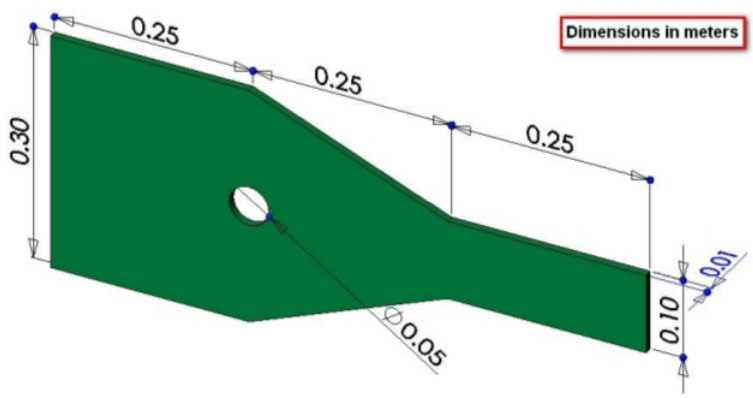


Displacements:



4.4 EX_4: 2D STEEL BRACKET

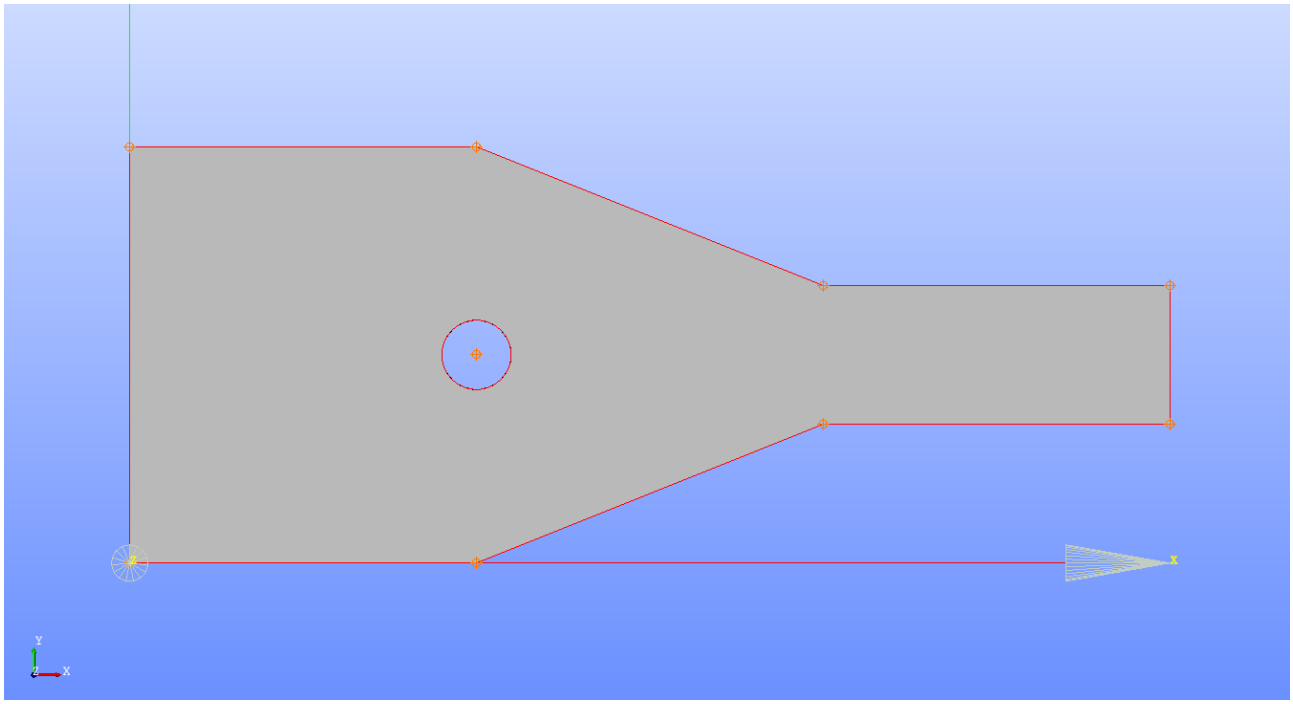
Last included example is a recreation of an Abaqus tutorial by Hormoz Zareh and Jayson Martinez. The plate geometry is:

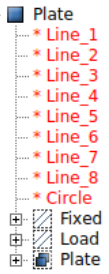


5000 N are applied to the right side of the bracket.

Although symmetry can used to define the geometry as we did in example 2, I decided to model all the plate.

The modeling process should be fairly familiar by now: Points, lines and circle, face. As pointed earlier, face can be modeled in one step selecting the circumference first and then all other line.



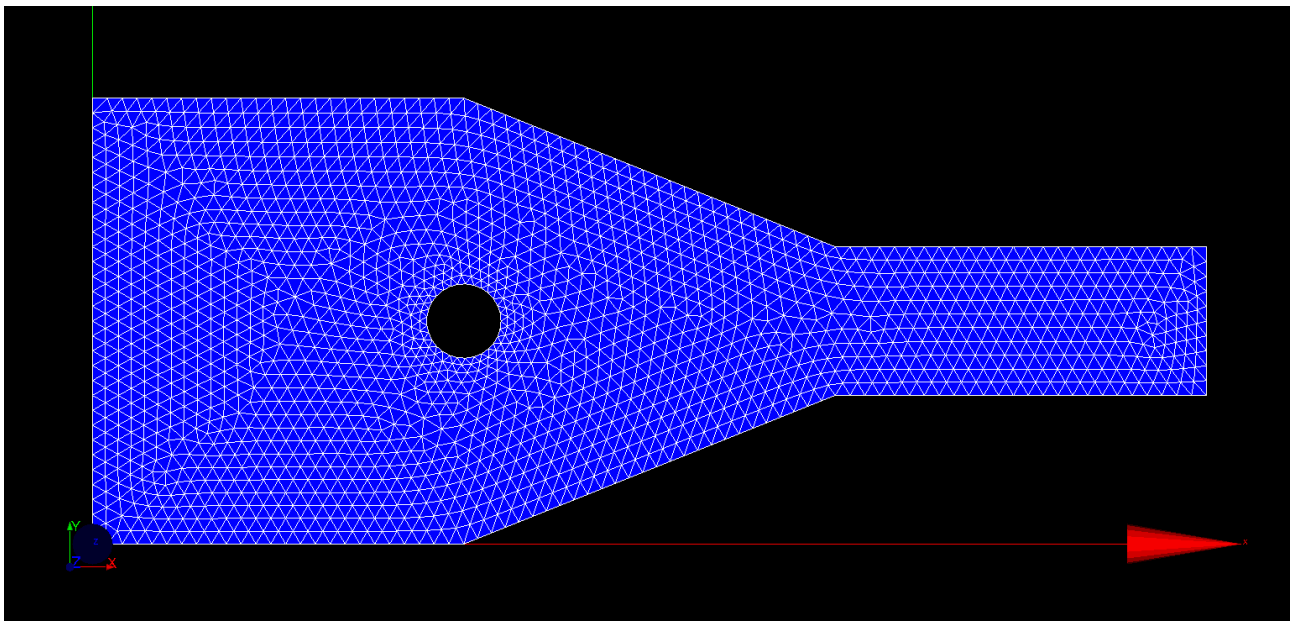


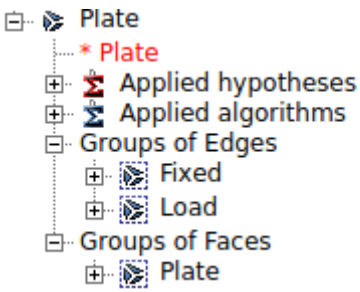
Two line groups need to be defined, one on the leftmost line that will be the fixed boundary condition and one line on the right that will be the load location.

The whole plate will be assigned to a face group as usual.

Groups are shown at the left.

Meshing and importing groups from geometry follows:





Yaf input file is:

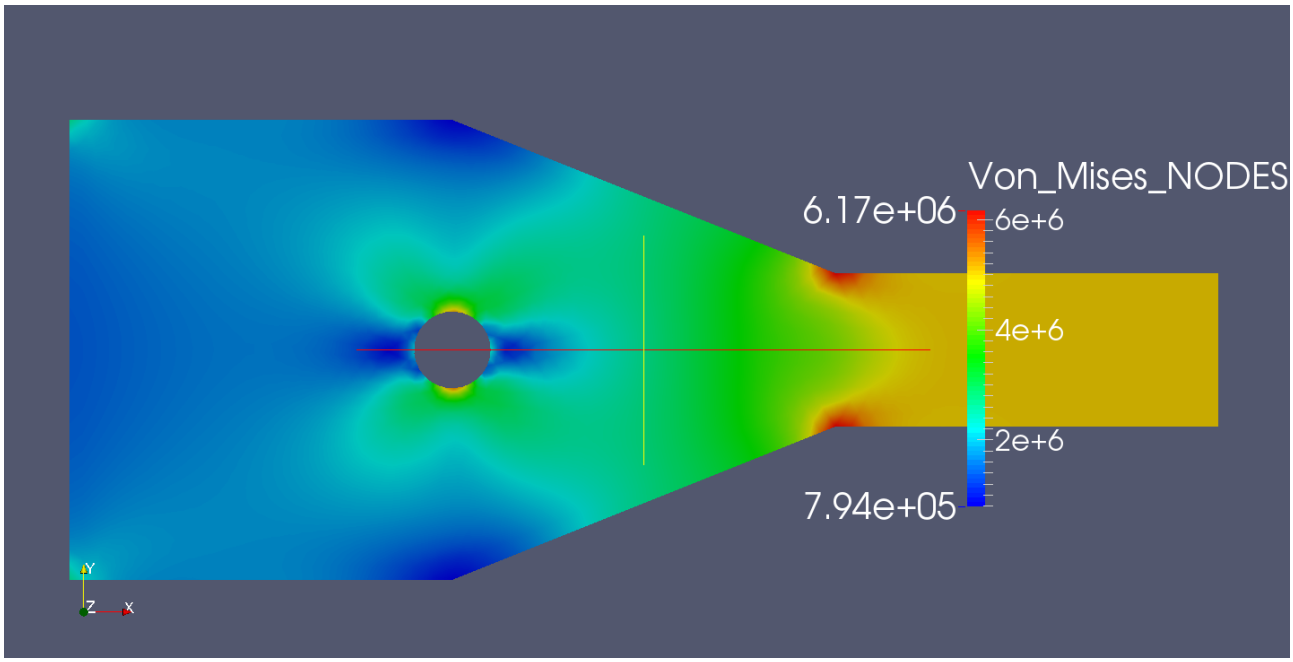
```
EX_4.med
1 1 1 1 1
0.01 210000000000 0.3
Plate
1
Fixed
1 1
Load
-5000000 0 1
END
```

Running YAFEMS:

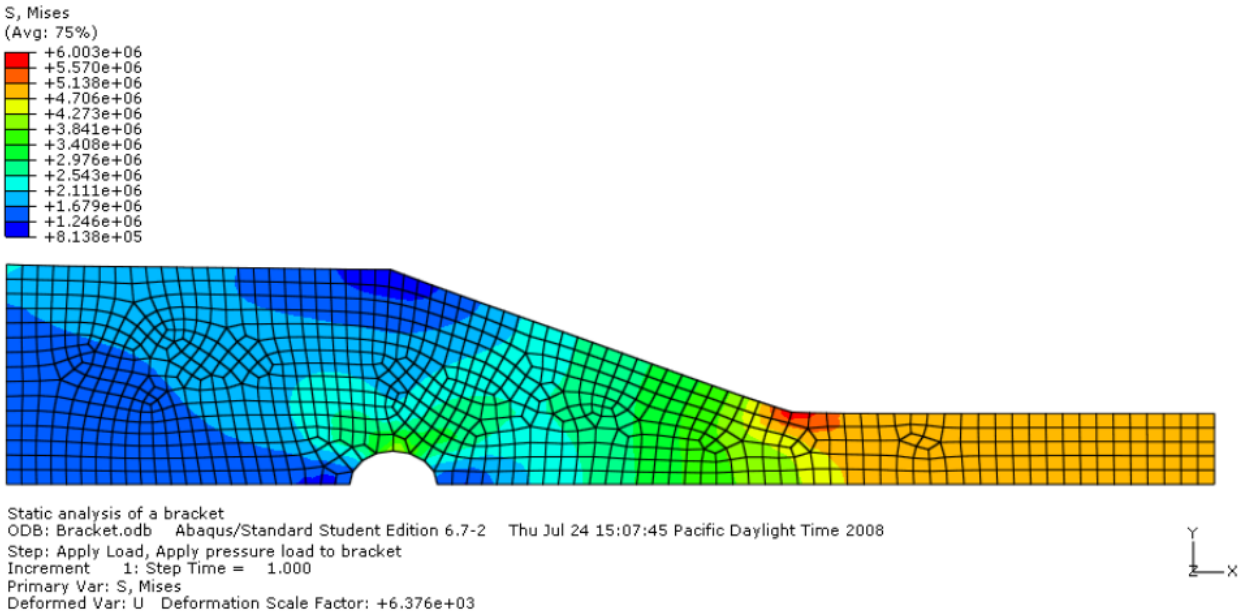
```
yafems_2d EX_4.yaf
```

Now we can open ParaView using the generated rmed file and compare the results to the Abaqus solution.

YAFEMS solution:



Abaqus solution:



We can see that the solution is quite similar.

4.5 EX_5: 3D CANTILEVER BEAM

This is a remake of the EX_1 where a cantilever beam was modeled using plane stress analysis. We saw that the results were quite accurate compared to classic beam theory. We are going to model the beam as a general 3D solid now and compare the results.

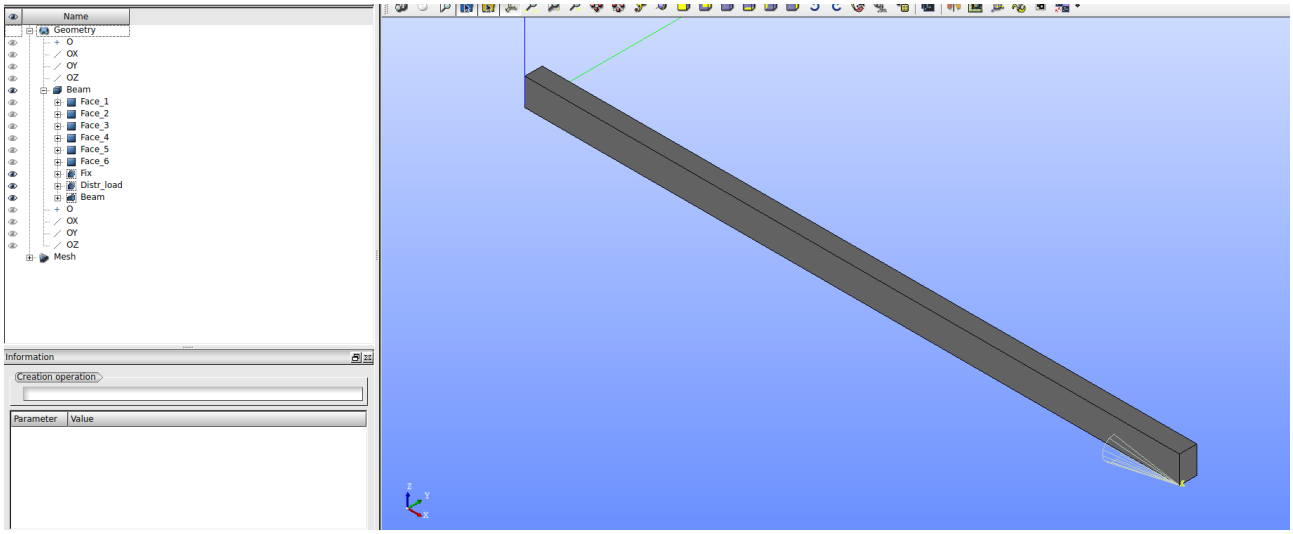
As we're creating a 3D model, concepts are a bit different from previous 2D cases. The beam is quite simple to obtain. We define a box with the correct dimensions of 12, 0.32 and 0.5, all meters.

This volume is a solid with no defined lines and vertices, so we need to “Explode” it to be able to define our geometry later. We explode for faces and vertices. As we are comparing to Example 1b, we only need to create three gorups:

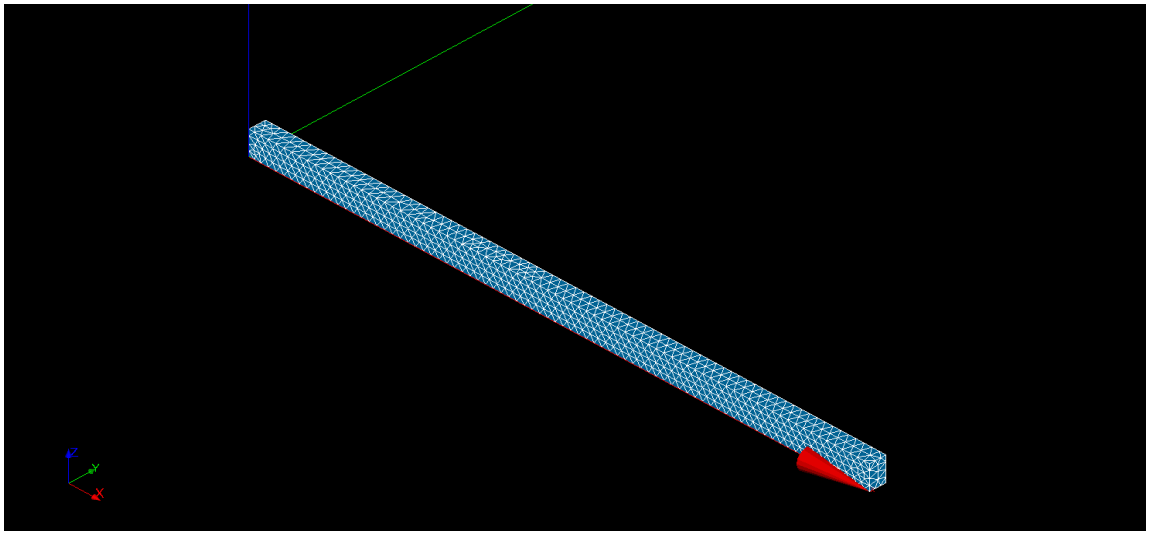
- A face group for the Fix boundary condition, selecting the leftmost face.

- A face group for the distributed load, called Distr_load selecting the upper face.

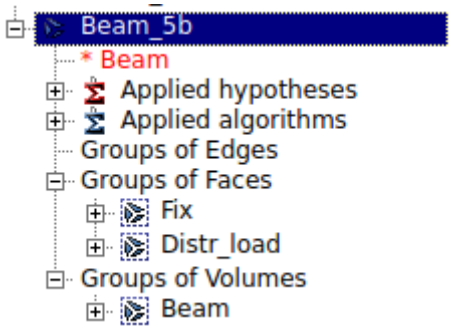
- A volume group of the whole beam to apply the material, selecting the complete beam.



We go to the Mesh module, and use the Netgen 1D-2D-3D making sure that the Allow Quadrangles and Second Order options are NOT checked.



We need to import the geometry groups now:



We now right click on the Beam_5b mesh name (in blue above) and export it to MED format.

Last step is defining the .yaf file that is quite easy and simple:

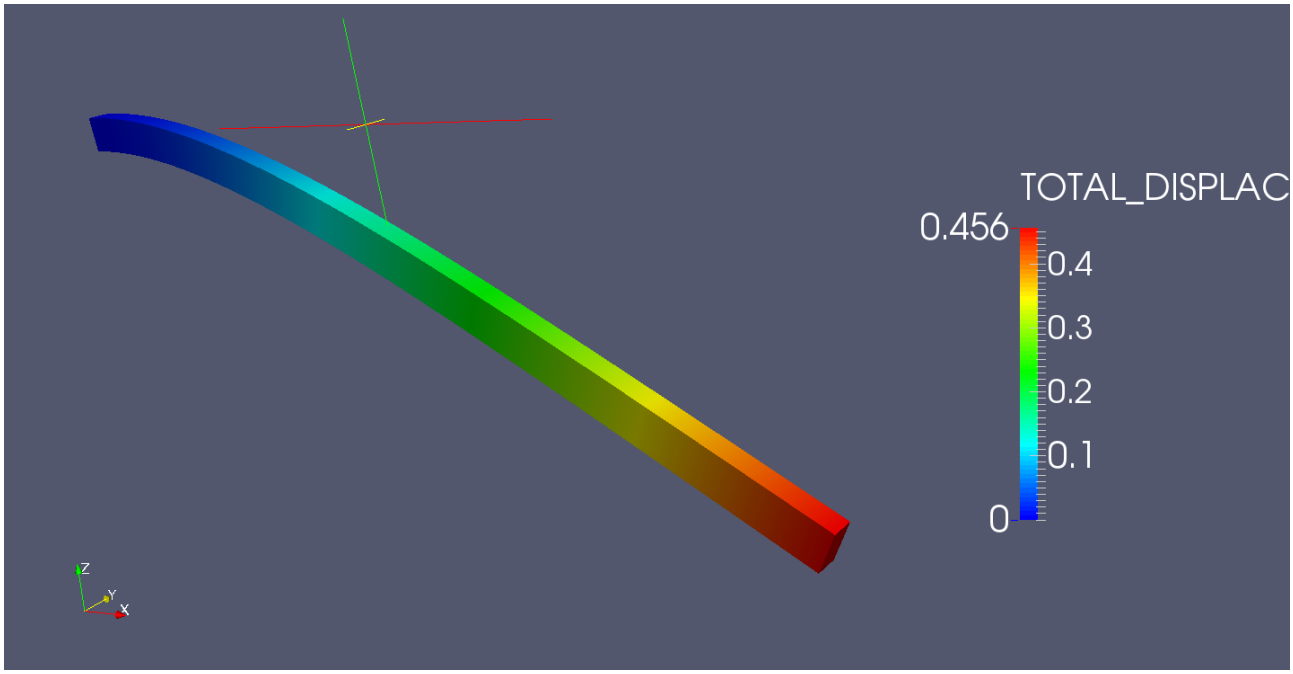
```
EX_5b.med
1 1 1 1
210000000000 0.3
Beam
1
Fix
1 1 1
Distr_load
0 0 -468750 1
END
```

Distr_load is -468750 N/m² that is -150 kN/m, the same as Example 1b.

Running YAFEMS_

```
yafems_3d EX_5b.yaf
```

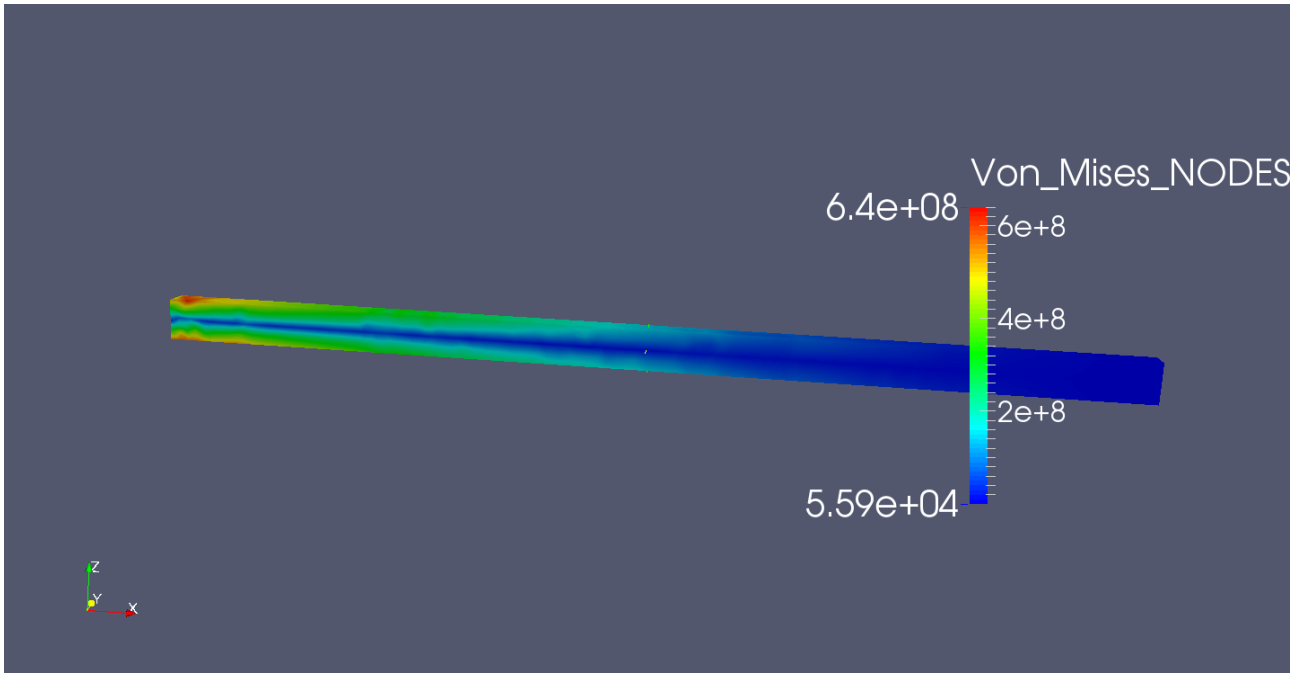
Now ParaView can be launchedto review our results. We start with the displacements.



We can clearly see that the maximum displacement is 0.456 m, that is far from the theoretical value of -0.55 m or the result from the 2d model of Example 1b that gave -0.533.

This is due to the low accuracy of the element used. Linear tetrahedron are quite inaccurate and should not be used if the actual value is wanted, but it serves the purpose of knowing the behaviour of the model.

Von Mises stress differs too:



In the following example we will prove that the inaccurate results of the program are due to the chosen meshing element (linear tetrahedron).

4.6 EX_6: 3D PISTON

This case is a the recreation of an excellent Salome-Code Aster tutorial by J. Cugnoni and C. Andersen, included in every CAElinux distribution.

The original tutorial files can be obtained in the CAElinux wiki webpage:

PDF of the tutorial:

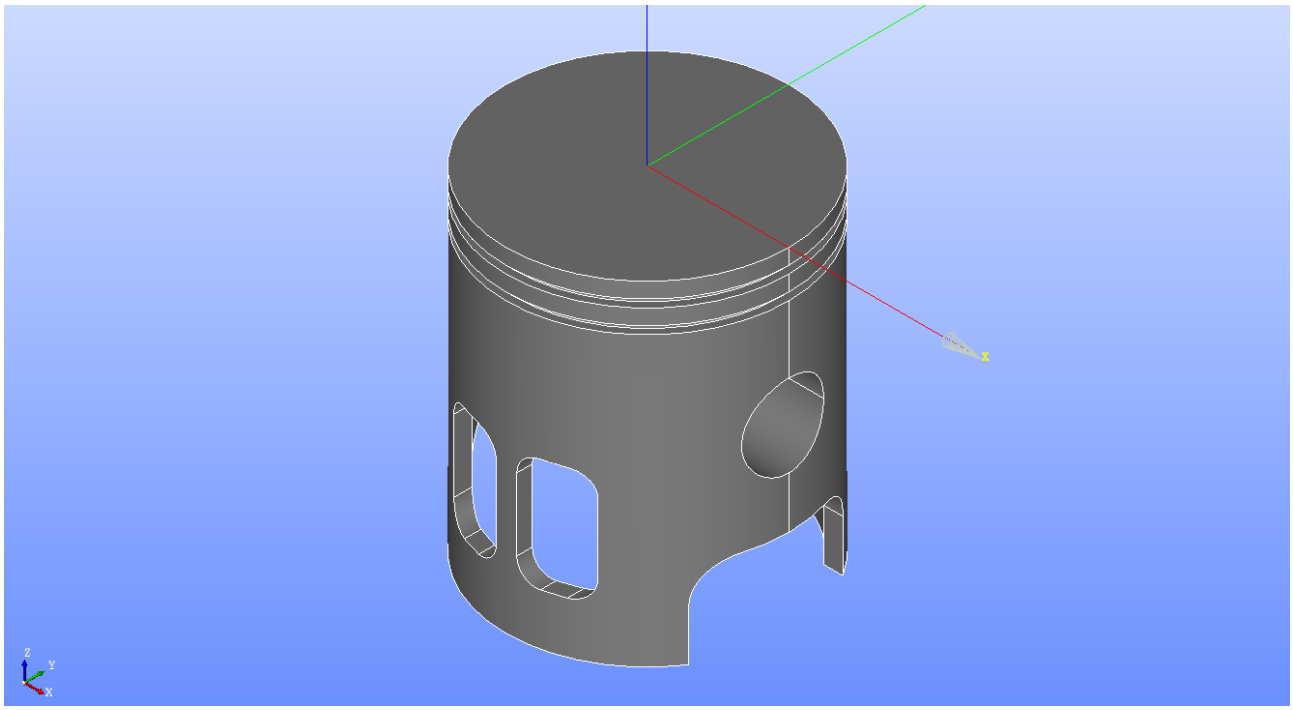
http://www.caelinux.org/wiki/images/7/74/Piston_tutorial.pdf

Geometry file:

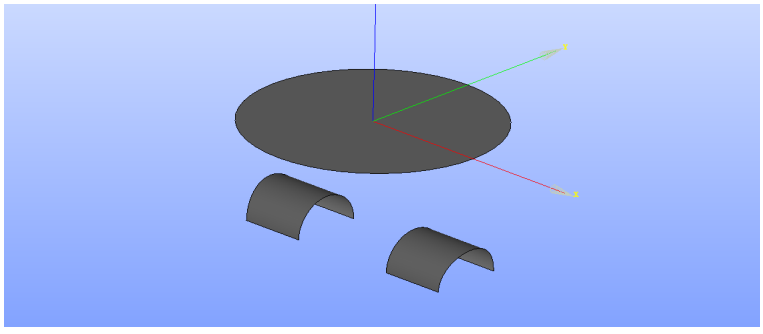
http://www.caelinux.org/wiki/images/e/e0/Piston.zip

The tutorial shows how to analyze a piston with a pressure of 300000 Pa on top, supported in the upper circular faces of two holes.

As the geometry is in STEP format, we need to import it into Salome in the Geometry module. Be sure to answer Yes when Salome asks about the units. After importing, go to Measures, dimensions, Bounding Box. Measures should be X and Y: -0.0275 to 0.0275. Z: -0.065 to 0.

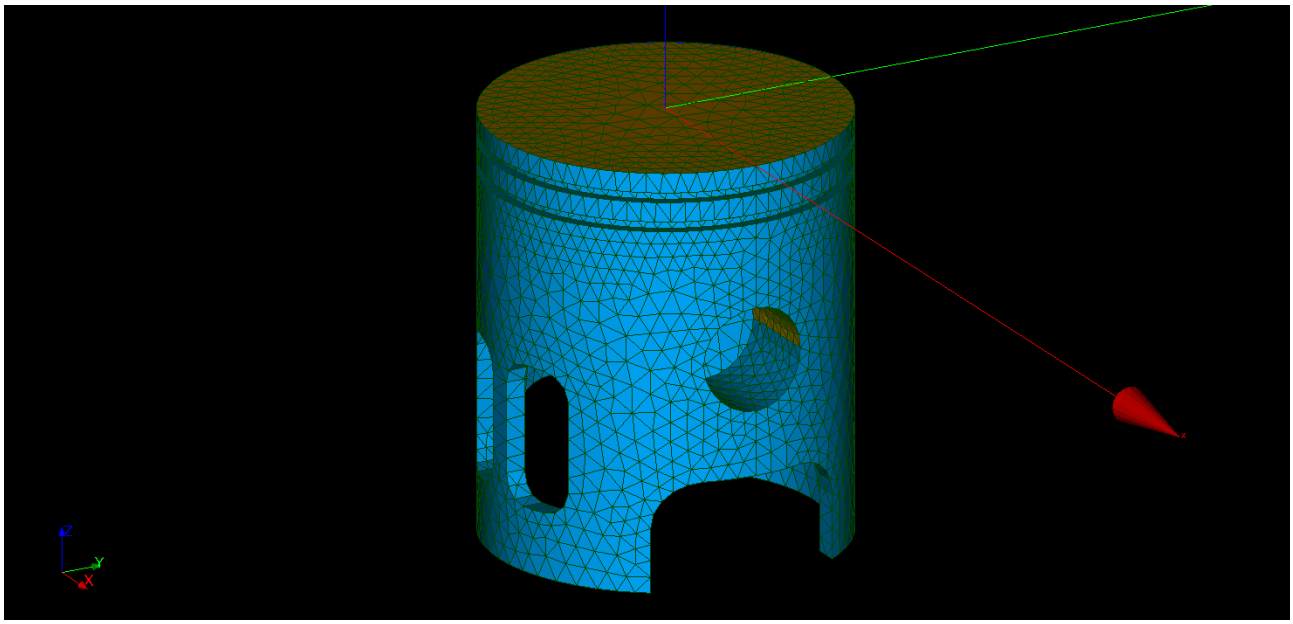


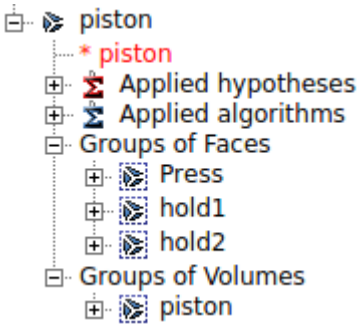
We need to set the groups as usual. We need 4 groups, 3 face groups: The top of the piston where the load will be applied, the upper two halves of the circular holes at each side of the piston and one volume group that includes the whole piston. The load and BC are shown below.



We go to the Mesh module and selecting the piston we create a mesh using Netgen 1D-2D-3D, making sure of NOT checking the Allow Quadrangles and Second Order options.

Next step is Creating Groups from Geometry. We import all groups:





We can see in the above picture the Press face and one of the two hole boundary conditions, depicted in brown color.

We export the mesh to MED format (EX_6.med) and create the .yaf file:

```
EX_6.med
1 1 2 1
70000000000 0.3
piston
1
hold1
1 1 1
hold2
1 1 1
Press
0 0 -300000 1
END
```

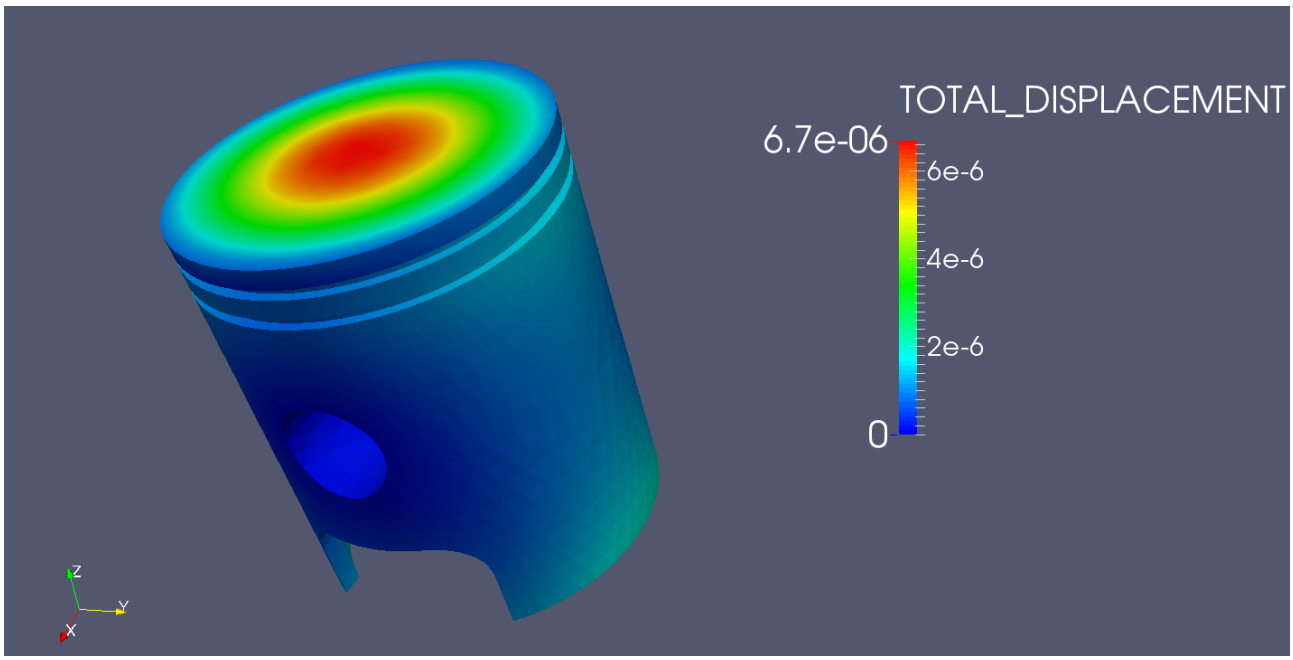
We execute YAFEMS:

```
yafems_3d EX_6.yaf
```

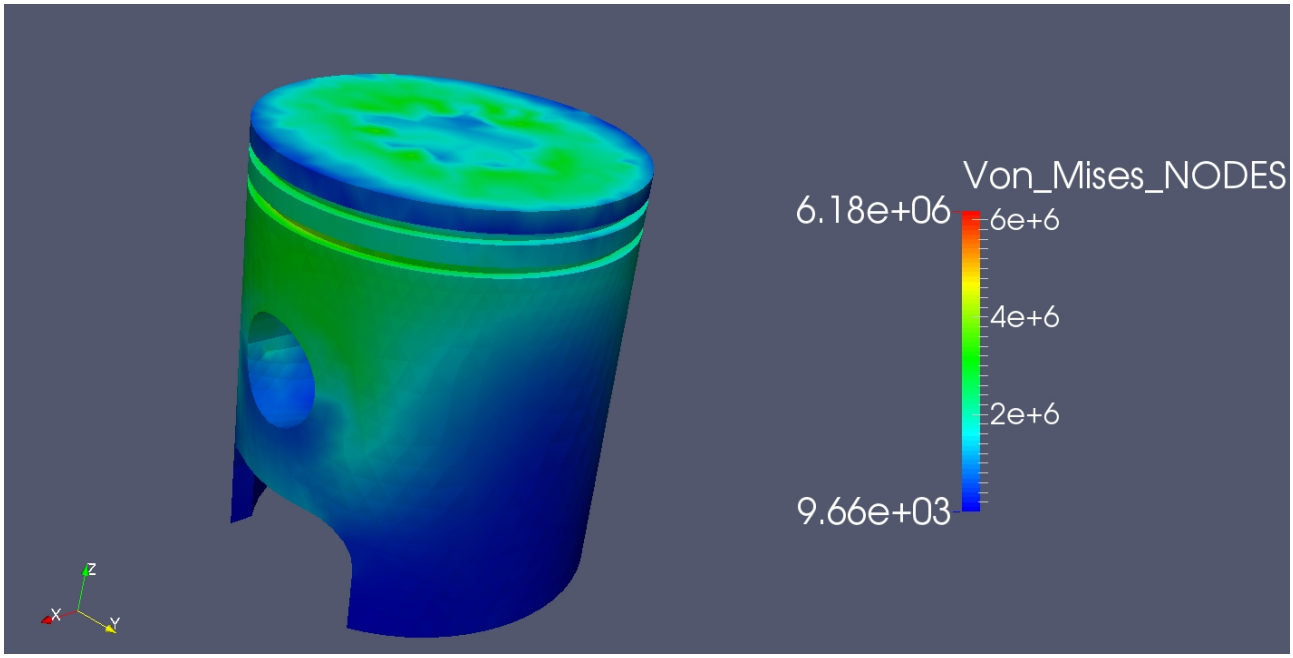
After the results are obtained we can open the EX_6.rmed file in ParaView.

I'm going to make two comparisons, showing the results obtained from YAFEMS first:

The displacements obtained by YAFEMS are:

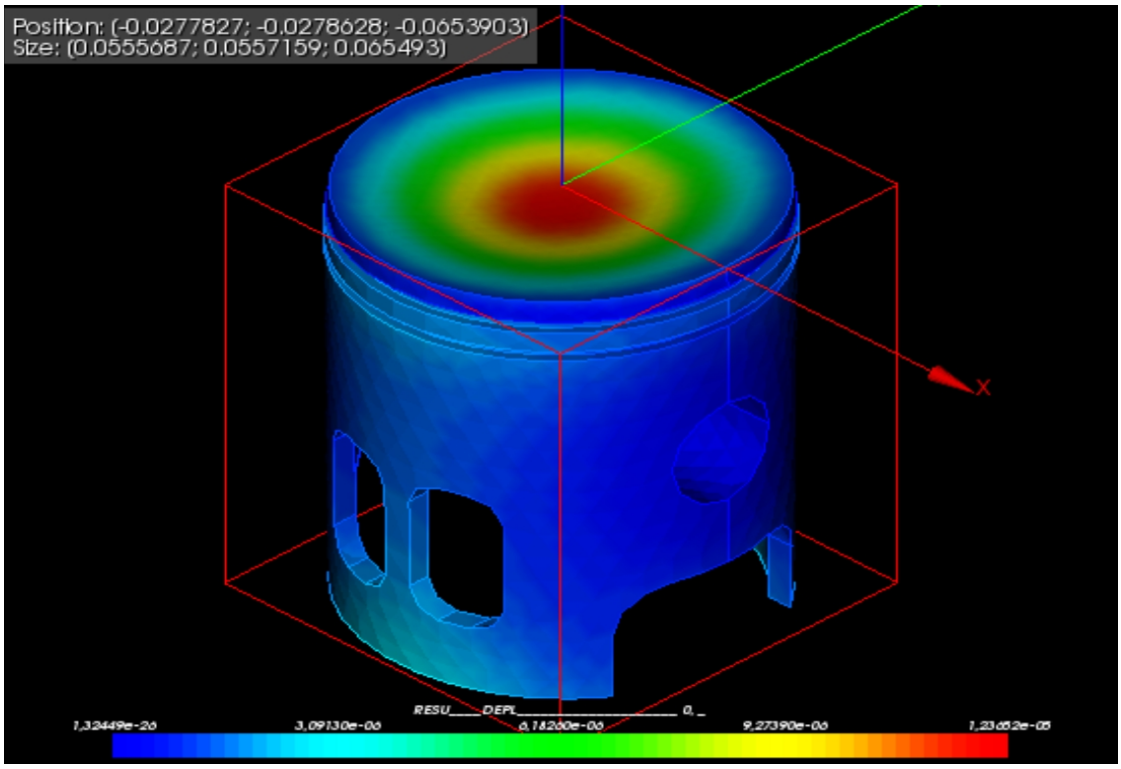


And the Von Mises stress:

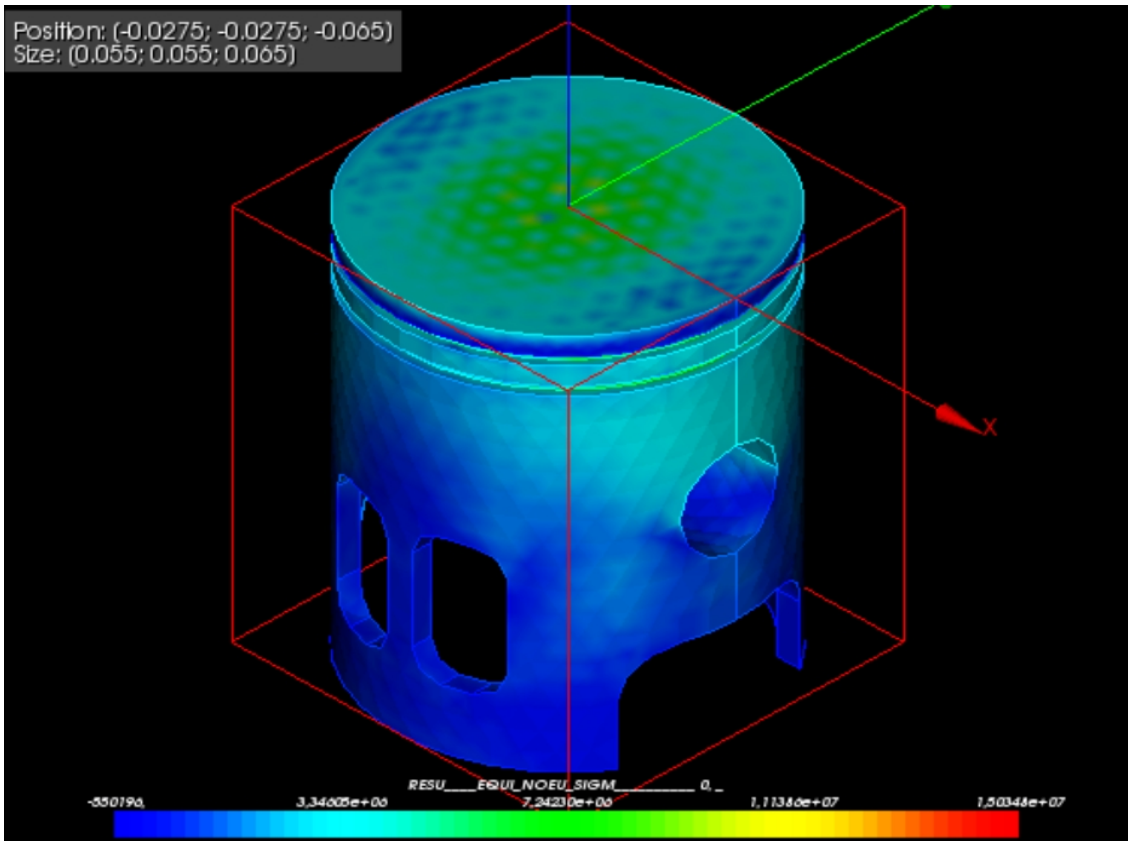


Now the solution obtained by Code Aster using **QUADRATIC TETRAHEDRONS** as the tutorial piston suggests will be shown.

Displacements by Code Aster, quadratic tetrahedrons:



Von Mises Stress, quadratic tetrahedrons:

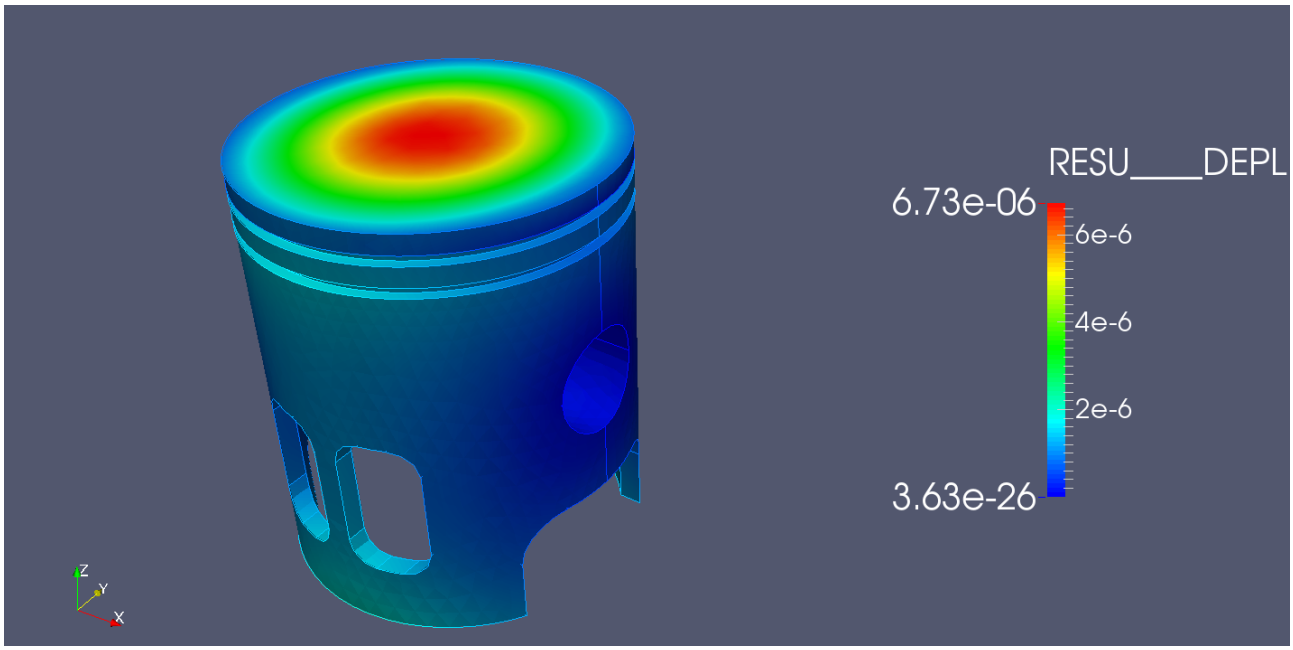


The maximum displacement is 1.23662E-05, instead of the 6.7E-6 obtained by YAFEMS. The results are quite different, as commented earlier. The same happens with the Von Mises stress,

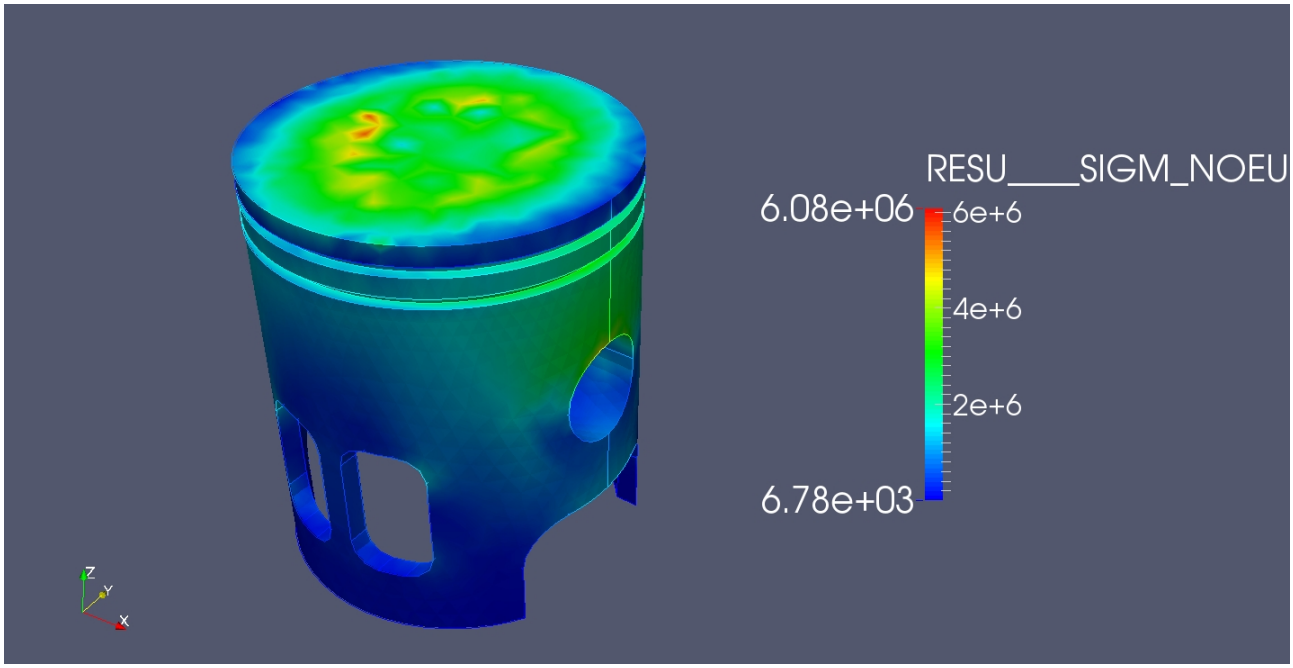
which is 6.18E6 for YAFEMS and 1.5E7 for Code Aster.

At this point, I thought that the YAFEMS implementation could be wrong, but then decided to remesh the model using the same parameters as used with YAFEMS and linear tetrahedrons and solving it with Code Aster. These are the results:

Displacements, Code Aster, linear tetrahedrons:



Von Mises Stress, Code Aster, linear tetrahedrons:



Now this results are almost identical as the results obtained by YAFEMS. The inaccuracies lie in the element used, and not in YAFEMS itself.

5. ABOUT THE AUTHOR AND CONTACT

My name is Javier Marcelo Mora, and I'm a spanish Civil Engineer with a master's degree in city planning and territory regulation (Ingeniero de Caminos, Canales y Puertos, Universidad Politécnica de Madrid) and another master's degree Theoretical & Practical Application of Finite Element Method and CAE Simulation by UNED.

My main work has developed as a technical manager in project evaluation, executing planning and cost estimation for bidding of public contracts. After the spanish economic crisis where the company I worked for had to restructure, I decided to turn my attention to other engineering field that I always liked during my undergraduate years, that is structural analysis and decided to take part in a two year master's degree in FEM. After finishing the master, I wanted to implement what I learned into YAFEMS, so this little project was born.

If you have any questions, suggestions, improvements, job offers ;-) , or just want to talk to me about YAFEMS or any other subject, please feel free to contact me at:

javiermarcelomora@gmail.com

Thank you very much, and I hope that YAFEMS will be useful to you.