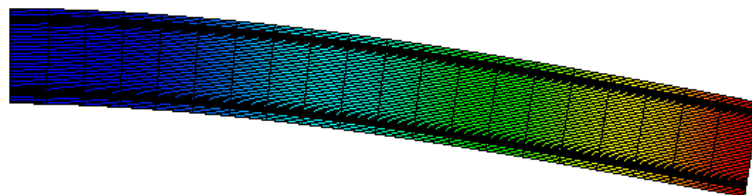
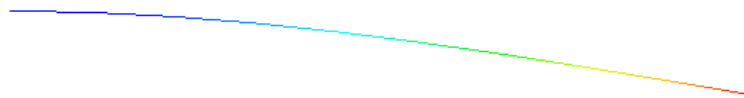


Applying simple beam theory with Code Aster®

1D and 3D

For CAELinux.com, May 2011 - Claus Andersen
Rev. 1.0

$$\delta = \frac{P \cdot x^2 \cdot (3 \cdot L - x)}{6 \cdot E \cdot I_x}$$





• Table of Contents

Rev. 1.0.....	1
1 Introduction.....	4
2 Analytical solution.....	5
2.1 Analytical solution.....	5
3 Using MACR_CARA_POUTRE to calculate a cross section.....	8
3.1.1 MACR_CARA_POUTRE values.....	9
3.1.2 Analytical values.....	9
3.2 Difference between analytical and MACR_CARA_POUTRE values.....	10
3.3 Note on MACR_CARA_POUTRE.....	10
3.4 ASTK set-up for MACR_CARA_POUTRE.....	11
4 1D Beam calculation with Code Aster®.....	12
4.1 Mesh.....	12
4.2 Command file.....	13
4.2.1 Reading the mesh and assigning a finite element model.....	13
4.2.2 Define and assign the material.....	13
4.2.3 Characteristics of the beam.....	13
4.2.4 Boundary conditions and load.....	14
4.2.5 Linear elastic solution.....	14
4.2.6 Calculating elements and nodes.....	15
4.2.7 Writing results to .MED file and text file.....	15
4.2.8 Comparing results.....	15
4.3 Comparing analytical results to Code Aster® (1D):.....	16
4.4 Post-processing with Salomé.....	17
4.5 ASTK set-up for 1D beam.....	18
5 3D beam.....	19
5.1 Preparing and meshing the 3D beam.....	19
5.1.1 Preparing the geometry.....	19
5.1.2 Creating the mesh.....	20
5.1.3 Creating the mesh – creating groups and extrusion.....	21
5.2 Command file.....	24
5.2.1 Load mesh, assign model and material.....	24
5.2.2 Boundary condition and load.....	25



5.2.3 Calculating a solution and writing the result.....	26
5.2.4 Post-processing options in the command file.....	27
5.3 Post-processing.....	29
5.3.1 Reviewing the textual output.....	29
5.3.2 Post-processing in Salomé.....	29
5.4 Comparing results – analytical to Code Aster® (3D).....	32
5.5 ASTK setup for 3D calculation.....	32
6 Conclusion, remarks and author(s).....	33
7 Links.....	34



1 Introduction

For this exercise, we'll analyze a cantilever beam in three different ways: analytical approach and 1D modelization / 3D modelization with Code Aster®

Simple beam theory will be kept at a minimum and emphasis will be on how to utilize Code Aster® to calculate the results, and how to extract and view said results.

A standard steel H profile beam is used in this exercise, as tables with values for these profiles are abundant. This saves us some time doing trivial calculations on the specific profile, but should we want to use a non-standard profile or just a profile not found in a table, a short introduction to Code Aster's® integrated profile calculator (**MACR_CARA_POUTRE**) is included.

2 Analytical solution

2.1 Analytical solution

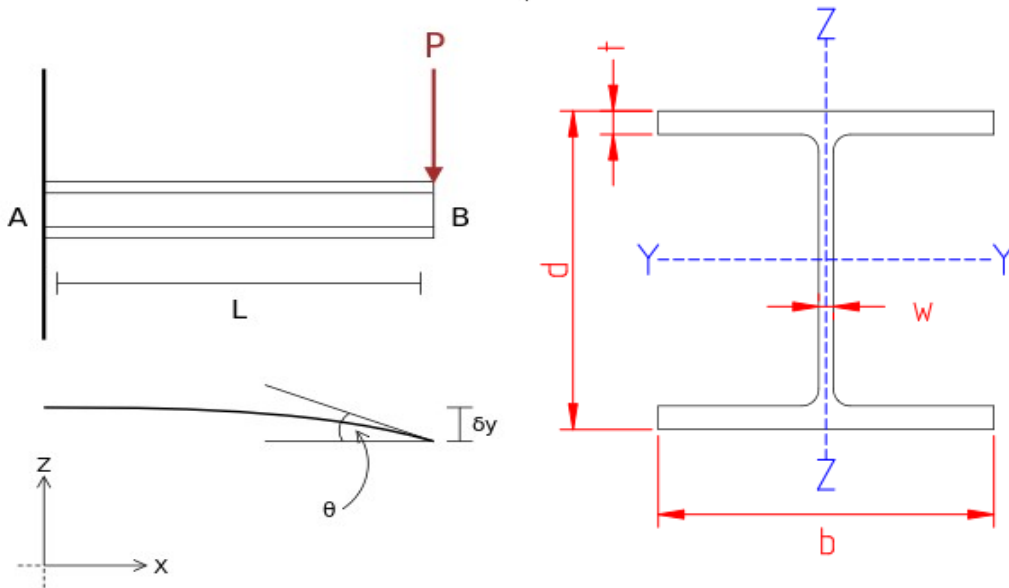


Figure 2.1: CCW

t:=27.7mm	IY:=712·10 ⁶ mm ⁴	E:=66·10 ³ MPa	
w:=17.3mm	IZ:=283·10 ⁶ mm ⁴	P:=-10000N	
b:=394mm	L:=3000mm	x:=L	
d:=375mm	d ₁ :=d-t	c:=d/2	

Area:

$$A := 2 \cdot t \cdot b + (d - 2 \cdot t) \cdot w = 2.74 \cdot 10^4 \text{ mm}^2 \quad (1)$$

Displacement at free end:

$$\delta = \frac{P \cdot x^2 \cdot (3 \cdot L - x)}{6 \cdot E \cdot I_x} = -1.92 \text{ mm} \quad (2)$$

Angle of slope at free end:

$$\phi = -\frac{P \cdot (2 \cdot L - x)}{2 \cdot E \cdot I_x} = 5.49 \cdot 10^{(-2)} \text{ deg} \vee 9.98 \cdot 10^{(-4)} \text{ rad} \quad (3)$$

Bending moment at fixed end:

$$M_A = P \cdot L = -3 \cdot 10^7 \text{ N} \cdot \text{mm} \quad (4)$$

Shear force:

$$V = P \quad (5)$$

Maximum normal stress:

$$\sigma_{max} = |M_A| \cdot \frac{c}{I_y} = 7.9 \text{ MPa} \quad (6)$$

Average shearing stress:

$$\tau_{avg} = \frac{V}{A} = -0.37 \text{ MPa} \quad (7)$$

Max. shearing stress (in the web of profile, neglecting shear in flanges)

$$\tau_{max.web} = \frac{V}{A_{web}} \rightarrow \frac{V}{(d - 2 \cdot t) \cdot w} = -1.81 \text{ MPa} \quad (8)$$

Section warping constant:

$$J_G = \frac{d_1^2 \cdot b^3 \cdot t}{24} = 8.51 \cdot 10^{12} \text{ mm}^6 \quad (9)$$

Torsion (stiffness) constant :

$$C_T = \frac{2 \cdot b \cdot t^3 + d_1 \cdot w^3}{3} = 6.18 \cdot 10^6 \text{ mm}^4 \quad (10)$$

Note on shearing stress: Using the web area to determine maximum shearing stress is a bit crude and gives a conservative (greater than actual stress, not by much though) result.

An alternative approach, is to determine maximum shearing stress by means of area coefficients.

We are working in the OYZ coordinate system and want to determine maximum shearing stress τ_{xz} , and thus need the **AZ** area coefficient This can be manually calculated, or obtained from **MACR_CARA_POUTRE** (see section 3).

ATTENTION(!): pay special attention to section 3.3 when extracting **AY/AZ** from **MACR_CARA_POUTRE**.

The following uses the concept of reduced area: $\text{Reduced area} = \frac{\text{Actual area}}{\text{Area coefficient}}$

Thus we have (using values from **MACR_CARA_POUTRE**):

$$A_{reduced} = \frac{A}{AZ} \rightarrow \frac{2.77143E+04 \text{ mm}^2}{4.64168E+00} = 5970,7477 \text{ mm}^2 \quad (11)$$

Which gives the following maximum shearing stress τ_{xz} :

$$\tau_{xz} = \frac{V}{A_{reduced}} \rightarrow \frac{-10 \text{ kN}}{5970.7477 \text{ mm}^2} = -1.675 \text{ MPa} \quad (12)$$

3 Using MACR_CARA_POUTRE to calculate a cross section

There exists many, many tables filled with profile characteristics, readily available for all your beam calculation needs. However, you could happen to have an irregular profile not found in any table, or want to skip calculating the values by hand. In either case, **MACR_CARA_POUTRE** can do this for you.

To have Code Aster® calculate the values for you, you must provide it with a 2D mesh of the profile in question, laying on the OXY plane (Fig. 3.1)

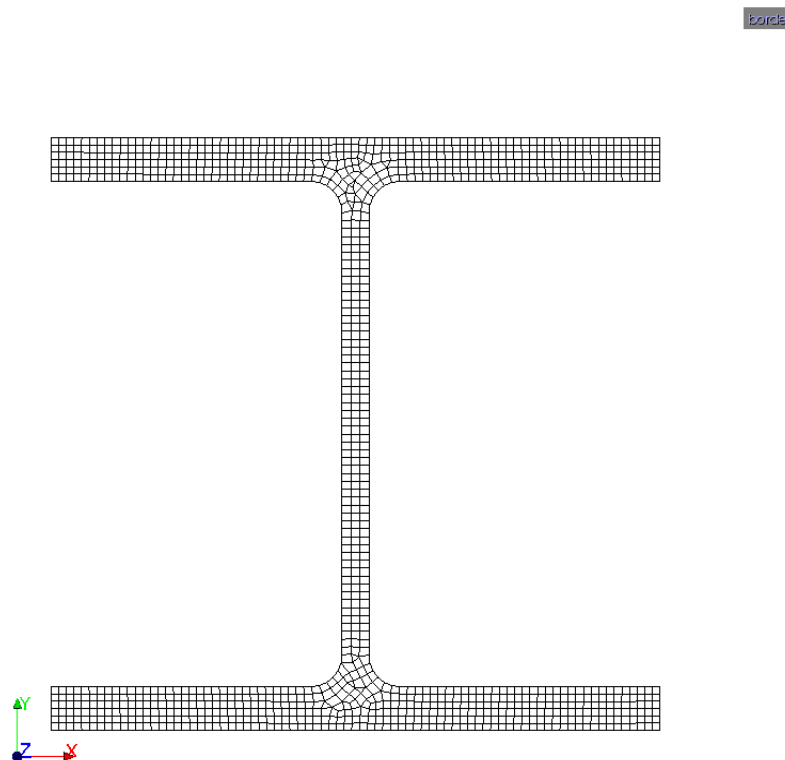


Figure 3.1: W360x216 profile

To get **MACR_CARA_POUTRE** calculate the sectional warping constant J_G and torsional inertia constant C_T , the mesh must have an element group of the entire border/edge of the profile - this group is used with the keyword **GROUP_MA_BORD**.

Once the values have been calculated, they are written into the *.resu* file with the **IMPR_TABLE** keyword. A list of parameters is used as not to clutter the *.resu* file. Each entry is separated by a comma (,).

```
DEBUT ( ) ;
```


3 Using MACR_CAR_POUTRE to calculate a cross section

```

mesh=LIRE_MALLAGE (FORMAT='MED',);

Xsection=MACR_CAR_POUTRE (MAILLAGE=mesh,
                          ORIG_INER=(0.0,0.0,),
                          GROUP_MA_BORD='border',
                          NOEUD='N421',);

IMPR_TABLE (TABLE=Xsection,
            FORMAT='TABLEAU',
            NOM_PARA=('Y_MAX','Z_MAX','Y_MIN','Z_MIN','R_MAX','
AIRE','CDG_X','CDG_Y','IX_G','IY_G','IXY_G','CT','JG','AY','AZ'),
            SEPARATEUR=' ',);

FIN();
    
```

3.1.1 MACR_CAR_POUTRE values

2 nd MOI	2 nd MOI	Area	Torsional inertia constant	Sectional warping constant
[mm ⁴]	[mm ⁴]	[mm ²]	[mm ⁴]	[mm ⁶]
IX_G	IY_G	AIRE	CT	JG
7.15279e8	2.82575e8	2.77143e4	6.31927e6	8.42738e12

- AY: 4.64168E+00
- AZ: 1.46472E+00

3.1.2 Analytical values

(Or values from tables)

2 nd MOI	2 nd MOI	Area	Torsional inertia constant	Sectional warping constant
[mm ⁴]	[mm ⁴]	[mm ²]	[mm ⁴]	[mm ⁶]
I _y	I _z	A	C _T	J _G
7.12e8	2.83e8	2.74e4	6.18e6	8.51e12



3.2 Difference between analytical and MACR_CARA_POUTRE values

Most of the difference can be attributed to omitting fillets in the analytical approach.

2 nd MOI	2 nd MOI	Area	Torsional inertia constant	Sectional warping constant
0.46%	0.15%	1.13%	2.20%	0.97%

3.3 Note on MACR_CARA_POUTRE

Values cannot be chained directly from the MACR_ concept to a beam calculation, not without involving Python scripting. As such, values must be written to a file and entered manually in the beam calculation.

Much confusion can come from using **MACR_CARA_POUTRE (MCP)** if one is not paying attention to what coordinate system (CSYS) is the reference frame.

MCP outputs values in two reference coordinate systems: OXY, the CSYS used to create and mesh the 2D profile and OZY – principal axes

This is very important when importing the calculated values into **AFFE_CARA_ELEM** since this keyword used a third (!) CSYS: OYZ.

Below is an attempt to clarify the different reference CSYS' :

Mesh CSYS input for **MCP**. Outputs AIRE_M, CDG_X_M, CDG_Y_M, IX_G_M, IY_G_M, IXY_G_M and IX_G, IY_G in this CSYS:

Y
↑
z → x

Values Y_MAX, Z_MAX, Y_MIN, Z_MIN, IY_PRIN_G, IZ_PRIN_G, AY, AZ in this CSYS:

Y
↑
z ← x

Assuming the beam lies on the (global) X-axis, **AFFE_CARA_ELEM (A_C_E)** uses this CSYS:

z

↑
 x→y

This effectively means that when using values from **MCP** in **A_C_E**, the following conversion is necessary (only relevant values for our analysis is shown here):

	MACR_CARA_POUTRE		AFFE_CARA_ELEM
2 nd MOI, I _y	IX_G_M, IX_G	→	I _y
2 nd MOI, I _z	IY_G_M, IY_G	→	I _z
Extremity, width (Y)	Z_MAX	→	RY
Extremity, height (Z)	Y_MAX	→	RZ
Area coefficient	AZ	→	AY
Area coefficient	AY	→	AZ

Kees Wouters have written a more comprehensible tutorial for **MACR_CARA_POUTRE** on the CAELinux.com wiki.

3.4 ASTK set-up for MACR_CARA_POUTRE

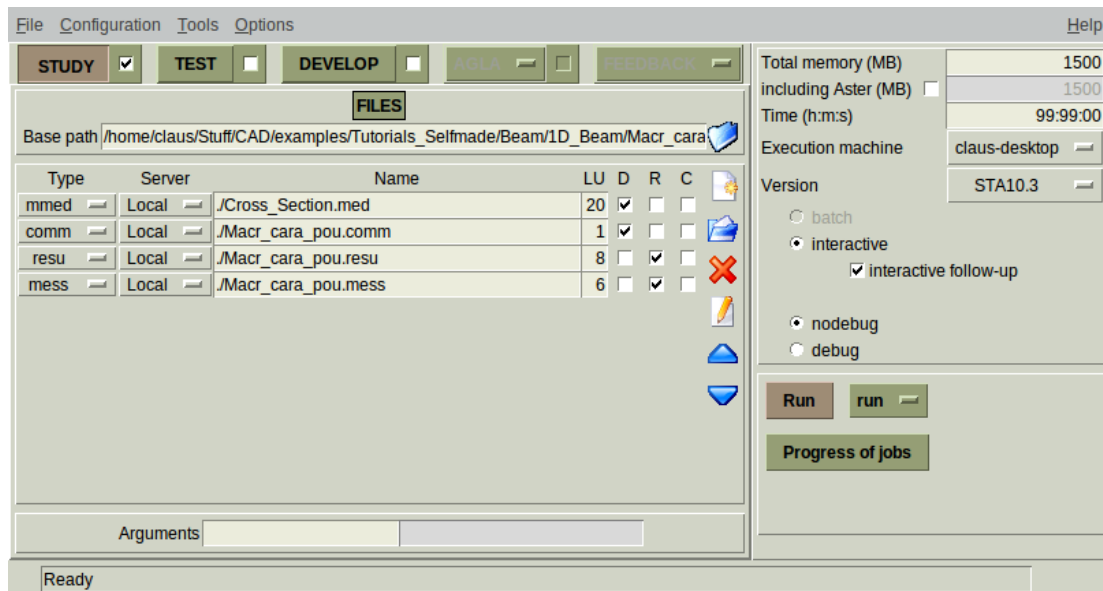


Figure 3.2: ASTK set-up

4 1D Beam calculation with Code Aster®

4.1 Mesh

Creating the geometry and mesh in Salomé® is quite straight-forward:

(To aid post-processing we off-set the beam in the Z-axis)

- GEOM: Create two points at (0,0,500) and (3000,0,500), and create a line using these two points.
- SMESH: Create a new mesh using the line as a Geometry input, under 1D select 'Wire discretization' as algorithm, and 'Nb. of segments' as hypothesis – set number of segments to 9.
- SMESH: Create two node groups: One called 'Fix' at node (0,0,500) and one called 'Load' at (3000,0,500). Export the mesh as a *.med* file. The result should look like figure 4.1

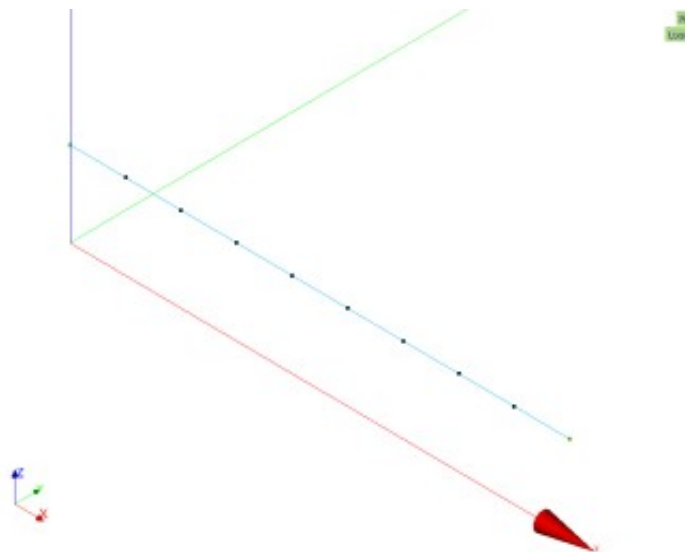


Figure 4.1: 1D mesh

Note: Placing the beam as pictured will make sure the local coordinate system (csys) of the beam matches the global csys. For orientations that differ from the global csys, see **ORIENTATION** keyword in [U4.42.01] **Opérateur AFFE_CARA_ELEM**.

4.2 Command file

4.2.1 Reading the mesh and assigning a finite element model

Load the mesh and name the concept ('Mesh') as per usual – format MED

Create an element group called 'TOUT' for all elements in the mesh.

Assign a mechanical phenomenon and a modelization of **POU_D_E** to everything (**TOUT=OUI**).

Modelization **POU_D_E** is the Euler-Bernoulli hypothesis that assumes that the sections remain straight and perpendicular to the fiber and assumes a long slender profile.

POU_D_E has the following degrees of freedom: DX, DY, DZ, DRX, DRY, DRZ.

(See [U3.11.01] Modélisations **POU_D_T**, **POU_D_E**, **POU_C_T**, **BARRE**)

```
DEBUT ();

Mesh=LIRE_MAILLAGE (FORMAT='MED', );

Mesh=DEFI_GROUP (reuse =Mesh,
                MAILLAGE=Mesh,
                CREA_GROUP_MA=_F (NOM='TOUT',
                                   TOUT='OUI', ), );

Model=AFFE_MODELE (MAILLAGE=Mesh,
                  AFFE=_F (TOUT='OUI',
                           PHENOMENE='MECANIQUE',
                           MODELISATION='POU_D_E', ), );
```

4.2.2 Define and assign the material

Young's module of $66 \cdot 10^3$ MPa (aluminum) and a Poisson's ratio of 0.3 – assign to everything.

```
Material=DEFI_MATERIAU (ELAS=_F (E=66000,
                                 NU=0.3, ), );

MatField=AFFE_MATERIAU (MAILLAGE=Mesh,
                       AFFE=_F (TOUT='OUI',
                                 MATER=Material, ), );
```

4.2.3 Characteristics of the beam

In order to define our non-standard cross-section (i.e. not a square, circle etc.), the keyword **GENERALE** must be used. For a Euler-Bernoulli beam, six parameters are needed: Area (A), 2nd moment of inertia around the Y- and Z-axis (IY and IZ), a torsion



constant and RY/RZ. RY and RZ denotes maximum distance from neutral axis to extremity of either axis. The torsion constant C_T is now referred to as 'J'. (Section warping constant is still called J_G , but not needed here).

These values are entered as parameters and referenced under **AFFE_CARA_ELEM** -> **VALE**. (See doc. [U4.42.01] section 9.4.3)

```
A = 27400.0;

I_y = 712000000.0;

I_z = 283000000.0;

J = 6180000.0;

RY = 394 / 2;

RZ = 375 / 2;

CARA_POU=AFFE_CARA_ELEM (MODELE=Model,
                          POUTRE=_F (GROUP_MA='TOUT',
                                       SECTION='GENERALE',
                                       CARA=('A', 'IY', 'IZ', 'JX', RY, RZ, ),
                                       VALE=(A, I_y, I_z, J, RY, RZ, ), ), );
```

4.2.4 Boundary conditions and load

Fix left extremity – **ENCASTRE** is equal to imposing $DX=0$, $DY=0$... etc.

Apply a point load at right extremity (ignoring gravity).

```
Hold=AFFE_CHAR_MECA (MODELE=Model,
                    DDL_IMPO=_F (GROUP_NO='Fix',
                                  LIAISON='ENCASTRE', ), ), );

Load=AFFE_CHAR_MECA (MODELE=Model,
                    FORCE_NODALE=_F (GROUP_NO='Load',
                                      FZ=-10000, ), ), );
```

4.2.5 Linear elastic solution

Calculate a linear solution using the material field, characteristics of the beam and loads.

```
RESU1=MECA_STATIQUE (MODELE=Model,
                    CHAM_MATER=MatField,
                    CARA_ELEM=CARA_POU,
                    EXCIT=( _F (CHARGE=Hold, ),
                           _F (CHARGE=Load, ), ), );
```



4.2.6 Calculating elements and nodes

EFGE: Calculate generalized forces (N,Vx,Vy,Mx,My,Mz) on elements and nodes, according to the *global* csys. For forces in the local csys of beam, you must provide the orientation in **AFGE_CARA_ELEM**.

SIPO: Calculate stress based on the properties entered in **AFGE_CARA_ELEM**.

REAC_NODA: Reactions on nodes

```
RESU1=CALC_ELEM(reuse =RESU1,  
                RESULTAT=RESU1,  
                OPTION=('EFGE_ELNO_DEPL','SIPO_ELNO_DEPL',),),);
```

```
RESU1=CALC_NO(reuse =RESU1,  
              RESULTAT=RESU1,  
              OPTION=('REAC_NODA','FORC_NODA','EFGE_NOEU_DEPL','SIPO_NOEU  
_DEPL',),),);
```

4.2.7 Writing results to .MED file and text file

Write the calculated results to *.med* and *.resu* file, respectably.

```
IMPR_RESU(MODELE=Model,  
          FORMAT='MED',  
          RESU=_F(MAILLAGE=Mesh,  
                 RESULTAT=RESU1,  
                 NOM_CHAM=('DEPL','EFGE_NOEU_DEPL','REAC_NODA','FORC_NOD  
A','SIPO_NOEU_DEPL',),),),);
```

```
IMPR_RESU(MODELE=Model,  
          FORMAT='RESULTAT',  
          RESU=_F(RESULTAT=RESU1,  
                 NOM_CHAM=('DEPL','EFGE_NOEU_DEPL','SIPO_NOEU_DEPL','REA  
C_NODA','FORC_NODA',),  
                 VALE_MAX='OUI',  
                 VALE_MIN='OUI',),),);
```

4.2.8 Comparing results

Values are found in the *.resu* file under their respective fields. Component=cmp

- Displacement at free end: DEPL field, DZ cmp: **-1.915[mm]**
- Angle of slope at free end: DEPL field, DRY cmp: **9.57610E-04** radians, equivalent to **5.487E-2 degrees**
- Bending (force) moment at fixed end: EFGE_ELNO_DEPL field, DRY cmp: **3E+7[Nmm]**
- Bending (reaction) moment at fixed end: REAC_NODA field, DRY cmp:



-3E+7[Nmm]

- Shearing force: EFGE_ELNO_DEPL field, cmp VZ: **-1.00000E+04[N]**
- Normal stress due to bending: SIPO_ELNO_DEPL, SMFY cmp: **7.879[MPa]**
- Shearing stress: SIPO_ELNO_DEPL field, SVZ cmp: **-3.65E-01[MPa]**

4.3 Comparing analytical results to Code Aster® (1D):

	Analytical	Code Aster®	Percentage
Displacement, free end	-1.92 mm	-1.915 mm	0.26 %
Angle of slope	9.58E-4 radians	9.576E-4 radians	4.17E-2 %
Normal stress, bending	7.9 MPa	7.879 MPa	0.27 %
Shearing stress	-3.7E-1 MPa	-3.65E-1 MPa	1.35 %

*Difference mostly due to rounding off.

4.4 Post-processing with Salomé

Salomé® has *some* capabilities for post-processing beams elements, and as such, here's an attempt to visualize the values obtained – Figure 4.2.

From bottom up:

- Shear force diagram: Plot3D with VZ cmp. Default scaling.
- Node forces: Vectors, cones of 2nd type, no shading. Default scaling.
- Moment diagram: Plot3D, MFY cmp. Scaled to 5e-05.

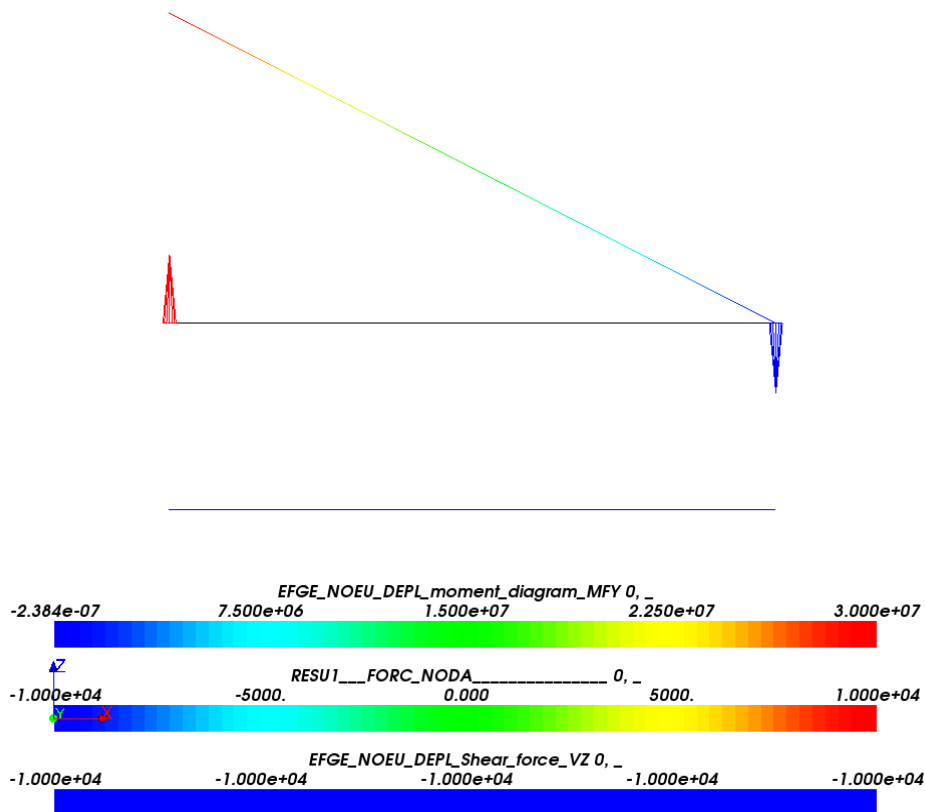


Figure 4.2: Post-processing with Salomé

4.5 ASTK set-up for 1D beam

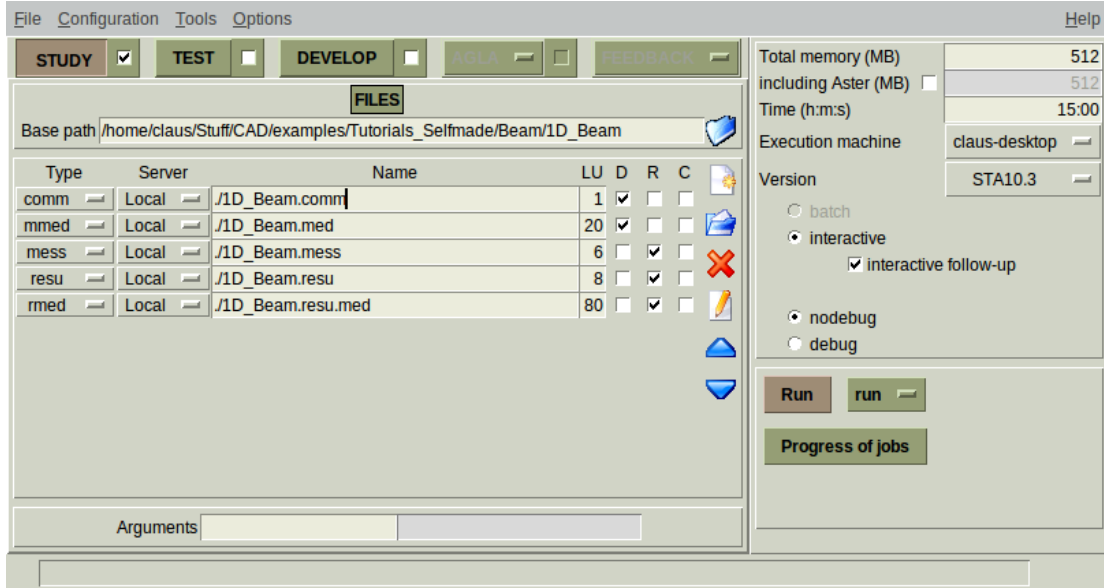


Figure 4.3: ASTK set-up

5 3D beam

5.1 Preparing and meshing the 3D beam

To achieve accurate results from the calculation, the beam should be meshed with hexahedral elements. Since most geometries with Salomé's SMESH module requires vigorous partitioning to satisfy the '4 sides only' rule for the quadrangle algorithm, we exploit the geometry of the beam to extrude quads into hexas.

The methodology is as follows:


- Import a 3D version of the beam in STEP format
- 'Explode' (extract) a face from the geometry
- Create groups on the resulting 2D shape
- Mesh 2D shape with quads, import groups
- Extrude 2D shape, create groups for boundaries and loads



5.1.1 Preparing the geometry

Start a new session in Salomé(MECA) and open the GEOM module. Select *File* → *Import...* and select the beam STEP file.

Note: When importing STEP files in Salomé(MECA), Salomé automatically scales the model down by a factor of 1000 in preparation of using the meter/kilogram/second system.

Select the model and scale the model up by a factor of 1000. *Operations* →

Transformation → *Scale transformation*, or click the appropriate icon in the toolbar 

If needed, rotate the model so it aligns with the X-axis: Create an origin and base vectors by clicking the icon  - rotate the model -90 degrees around the Z-axis 

Rotate the view so you are able to select the beam profile face closest to the origin, and select *New Entity* → *Explode* and extract the face such as shown in figure 5.7

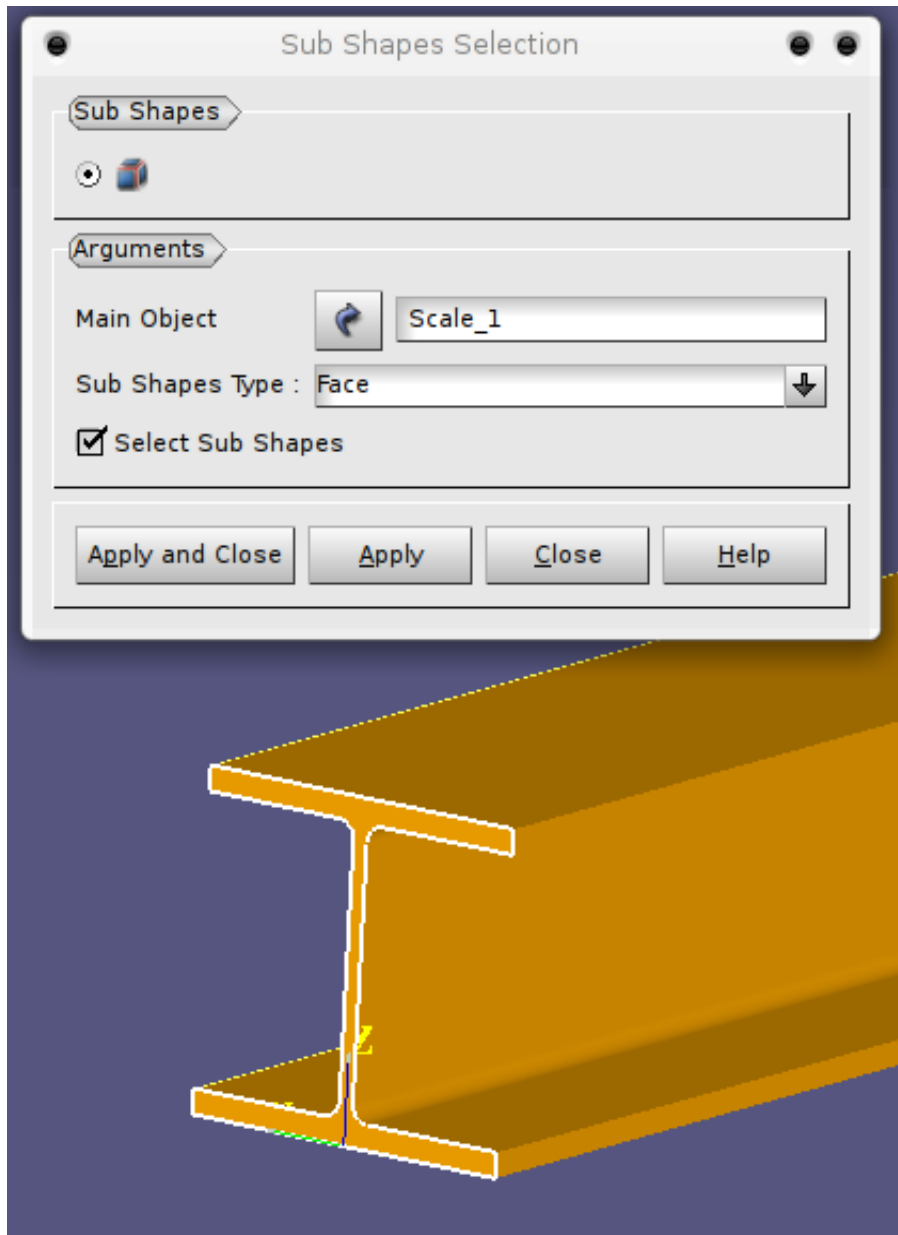



Figure 5.1: Extracting the profile

To save some time creating groups in the mesh module, create a face group on the extracted profile – 'Face_1' (expand the 'Scale_1' geometry) - and call it 'Fix'. Also create an edge group of the top edge of the profile – call it 'Press'.

5.1.2 Creating the mesh

Switch to the mesh module, and while the 'Face_1' geometry is selected, click *Create mesh* , enter the values shown in figure 5.2 and compute the mesh.

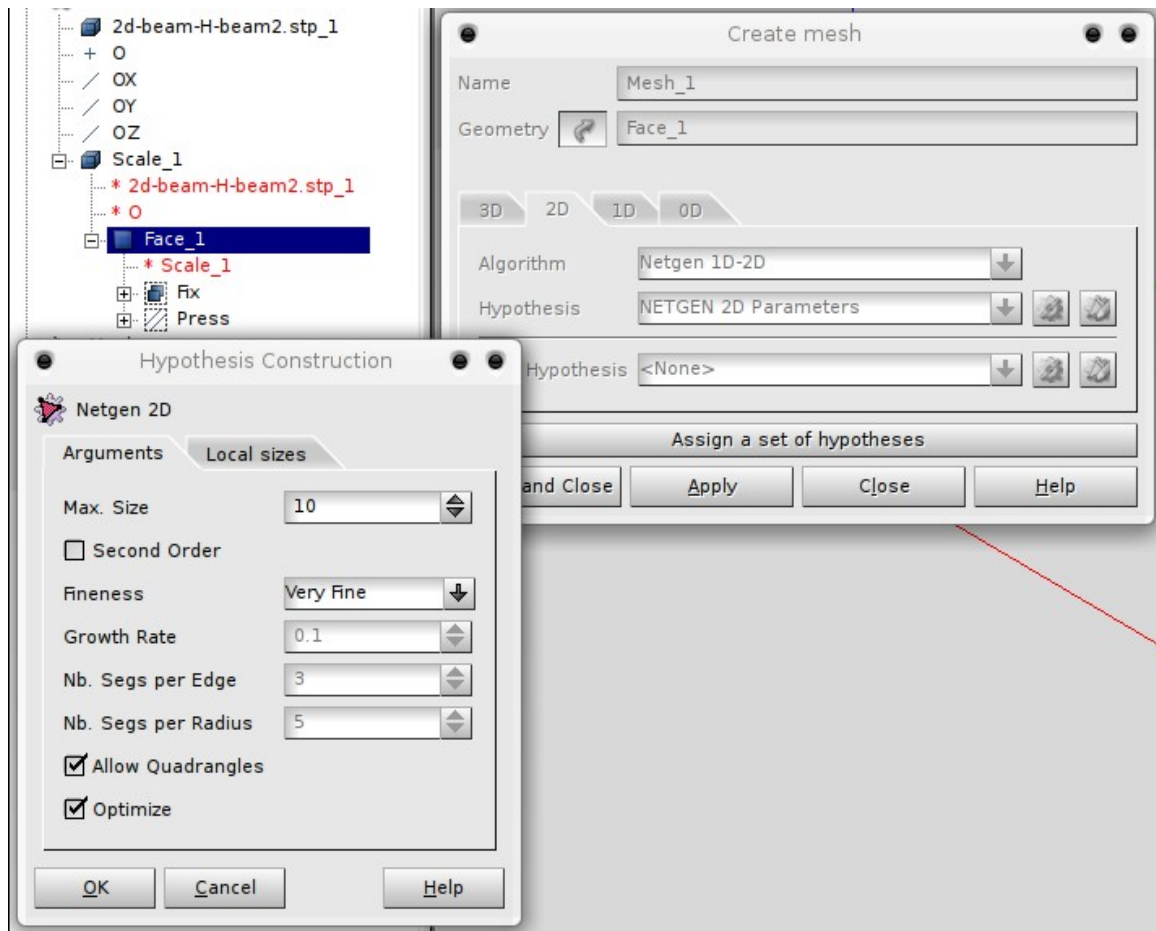



Figure 5.2: Mesh generation settings

5.1.3 Creating the mesh – creating groups and extrusion

Import the groups from the geometry by right-clicking the mesh and selecting *Create group*, switch to *Group on geometry* and select *Direct geometry selection* (press the blue arrow). Now create a *face* group from the 'Fix' group, and create an *edge* group from the 'Press' group.

To extrude the 2D mesh into a 3D mesh, make sure the 2D mesh is selected and press the extrude button 

To create a 3D mesh 3 meters long with reasonable hexahedral distribution, set X-distance to 150 and number of steps to 20 (distance of 150 X 20 steps = 3000mm)

See remaining settings in figure 5.3

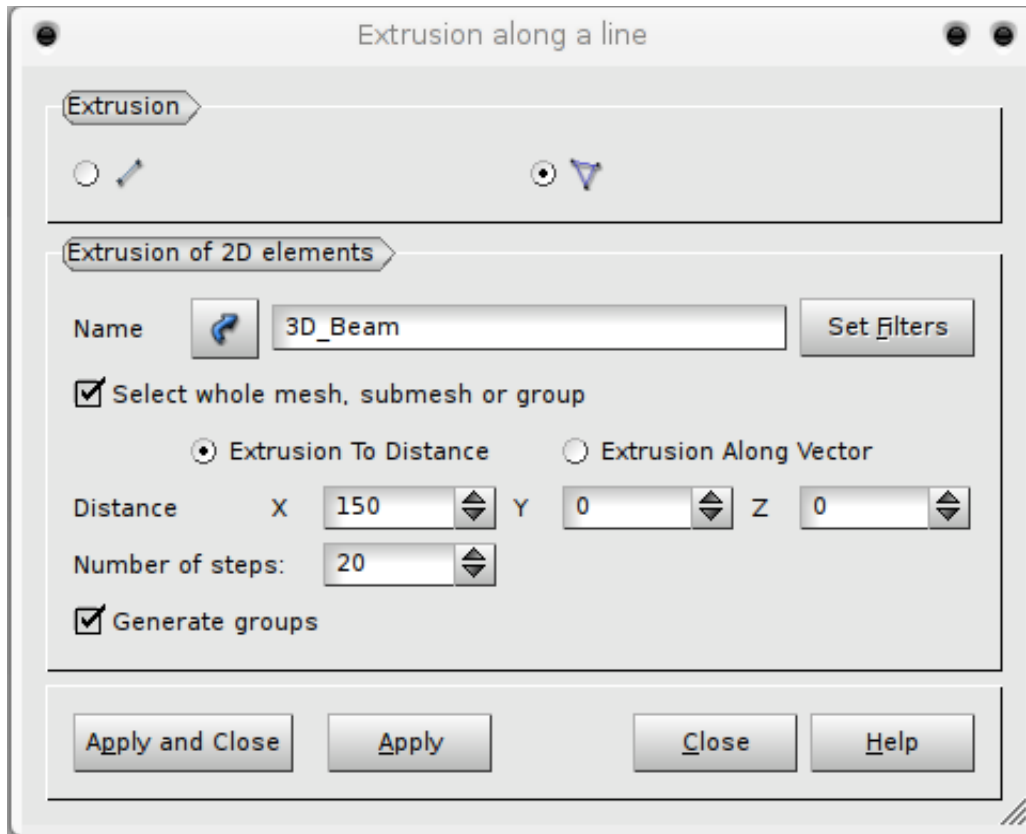


Figure 5.3: Settings for 3D extrusion

When *Generate groups* is checked, the mesh module creates volume groups from face groups and face groups from edge groups. For this study we don't need the newly created volume group (we'll retain the 'Press' group in case it's needed for another load case), so delete the 'Fix_extruded' volume group, delete the 'Press' *edge* group and finally rename the *face* group 'Press_extruded' → 'Press'.

The last group we need, is an edge group at the free end of the beam so we can apply the concentrated load.

To easily select the elements for the group, we can use 'clipping' to isolate the elements: Select the mesh and in the view port right-click and select 'clipping'

Click *New* and select Y-Z for orientation. To isolate the elements of interest, the plane needs to be rotated 180 degrees around the Y-axis (or Z). Set the distance to 0.99 and click *Apply and close*.

Note: The clipping plane has two colors: Dark blue and pale blue; everything on the dark

blue side of the plane will remain visible while everything else will be clipped away.

When the 3D mesh is clipped, we are left with a 2D mesh of the free end. With a the view port set to Front View, an edge group can be create easily. Name it 'Load' – see figure 5.4

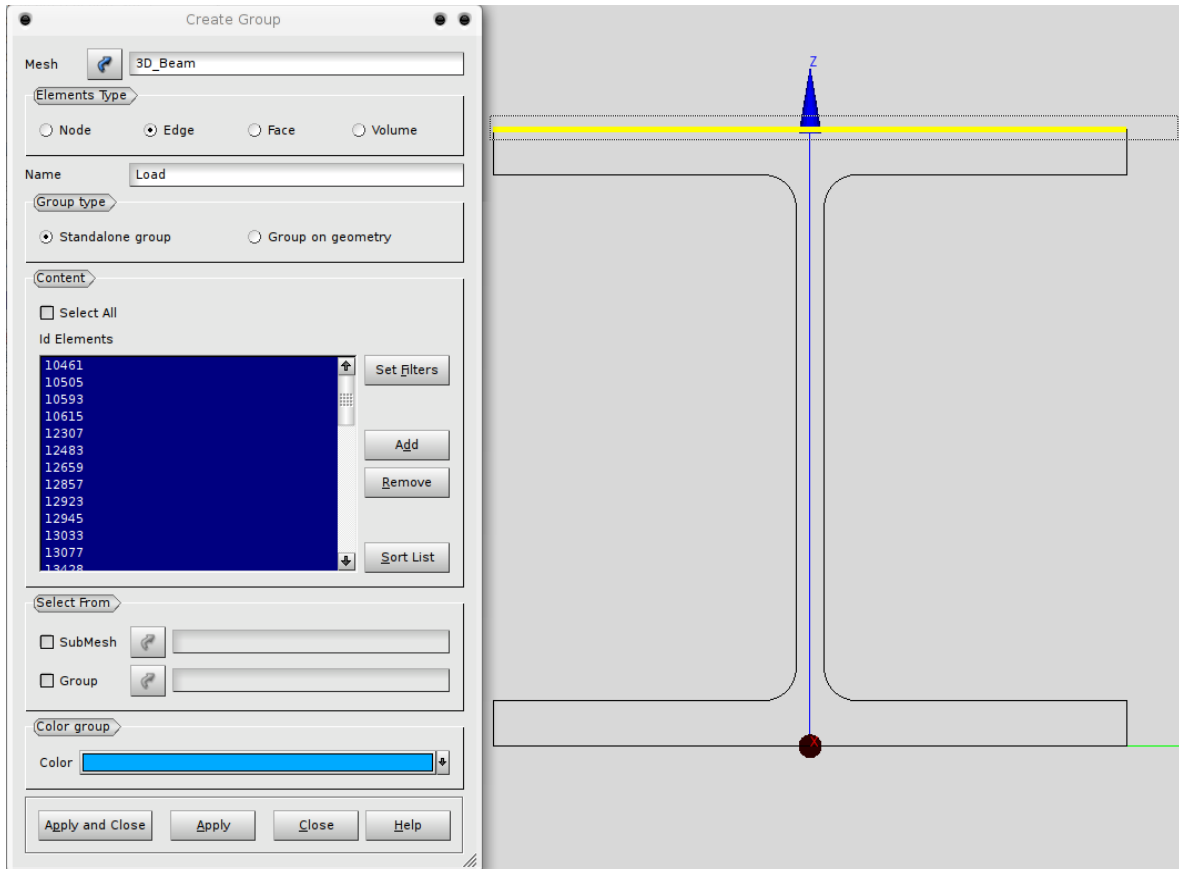


Figure 5.4: Creating the edge group 'Load'

We are now left with a mesh comprised of hexahedral elements containing the face group 'Fix' and the edge group 'Load'. See figure 5.5

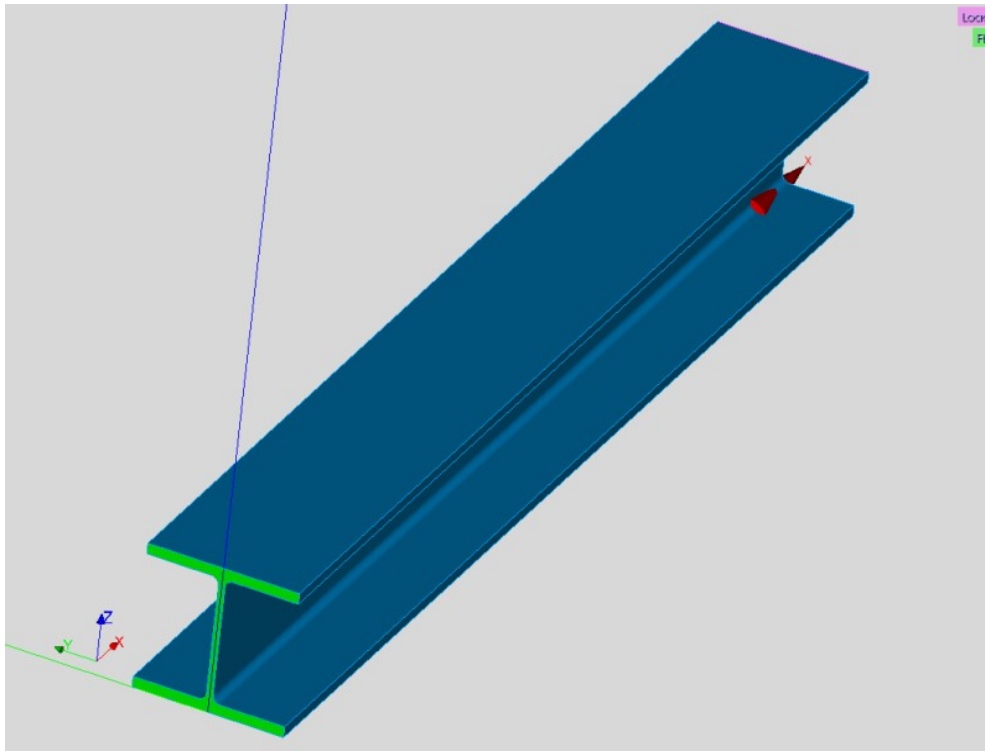


Figure 5.5: Green: Face group 'Fix' – Pink: Edge group 'Load'

Right-click the name of the mesh and convert it to a quadratic mesh by selecting 'Convert to/from quadratic' – tick 'Medium nodes on geometry'

Export the mesh to your working folder – right-click the mesh and select *Export* → *MED file*

5.2 Command file

The command file is a straight forward linear elastic study, and with exception of the application of load and post-processing commands, it won't be commented further upon, except directly in the *.comm* file.

5.2.1 Load mesh, assign model and material

```
DEBUT ();
```

```
#Defining the material.
```

```
#
```

```
#Linear elastic material - E-module = 66000MPa for Aluminum and a  
Poisson's ratio of 0.35.
```

```
Alu=DEFI_MATERIAU (ELAS=_F (E=66000.0,
```




```

                                NU=0.35, ), );

#'Read the mesh' - we use the 'med' file format here.

mesh=LIRE_MAILLAGE (UNITE=20,
                   FORMAT='MED',
                   INFO_MED=2, );

#Assigning the model for which CA will calculate the results: 'Mecanique'
- since we are dealing with a linear elastic beam and '3D' since it's a
3D model.

Meca=AFFE_MODELE (MAILLAGE=mesh,
                 AFFE=_F (TOUT='OUI',
                         PHENOMENE='MECANIQUE',
                         MODELISATION='3D', ), );

#Assigning the material of the beam using previously defined material.
'Tout = oui' means the whole model will have this material assigned.

Mat=AFFE_MATERIAU (MAILLAGE=mesh,
                  AFFE=_F (TOUT='OUI',
                          MATER=Alu, ), );
```

5.2.2 Boundary condition and load

The fixed end of the beam is imposed with zero displacement. In order to apply an equivalent to a point force, the keyword **FORCE_ARETE** is used to impose a linear force over the edges of the 'Load' edge group. A parameter ('Force') is used to impose this force.

Further more, the keyword **LIAISON_UNIF** with option=DZ, ensures the edge displaces uniformly in the Z direction.

```
#Boundary conditions
#'Fix' is blocked in all directions

BCs=AFFE_CHAR_MECA (MODELE=Meca,
                   FACE_IMPO=_F (GROUP_MA='Fix',
                                   DX=0.0,
                                   DY=0.0,
                                   DZ=0.0, ), );

#Load parameter
#The load is distributed linearly over the extreme end of the beam.
#-10000N / 394mm = -25.4N/mm
Force = -10000 / 394;

#'Load' element group is loaded with the parameter 'Force' in the Z
direction
#'Load' elements are constrained to move uniformly in the DZ direction
```



```
Loads=AFFE_CHAR_MECA (MODELE=Meca,  
                      LIAISON_UNIF=_F (GROUP_MA='Load',  
                                       DDL='DZ',),  
                      FORCE_ARETE=_F (GROUP_MA='Load',  
                                       FZ=Force,)),);
```

5.2.3 Calculating a solution and writing the result

```
#Defining the calculation model using previously defined model and loads  
( 'Meca', 'BCs' and 'Loads' )  
  
RESU=MECA_STATIQUE (MODELE=Meca,  
                   CHAM_MATER=Mat,  
                   EXCIT=( _F (CHARGE=BCs, ),  
                           _F (CHARGE=Loads, ), ), );  
  
#Calculate the elements.  
#  
#Using the model 'Meca', the material 'Mat' and entering the results into  
'RESU'  
#  
# 'b_lineaire' contains the things we want C_A to calculate.  
#  
#Normal stress and displacement on the elements.  
#  
#SIGM_ELNO_DEPL = Principal stress based on the 'displacement'.  
#  
#EQUI_ELNO_SIGM = Equivalent stress on the elements. ELGA: Elements Gauss  
points  
  
RESU=CALC_ELEM(reuse =RESU,  
              MODELE=Meca,  
              CHAM_MATER=Mat,  
              RESULTAT=RESU,  
              OPTION=('EQUI_ELNO_SIGM', 'EQUI_ELGA_SIGM', 'SIGM_ELNO_DEPL'  
), );  
  
#Calculate the nodes..  
#  
#and enter it to 'RESU'  
#  
#OPTION = SIGM_NOUD_DEPL, EQUIV_NOEU_SIGM corresponds to the calculation  
of the elements. FORC_NODA = Forces on nodes  
  
RESU=CALC_NO(reuse =RESU,  
            RESULTAT=RESU,  
            OPTION=('EQUI_NOEU_SIGM', 'SIGM_NOEU_DEPL', 'FORC_NODA', ), );  
  
#Write the results from the calculation.  
#  
# 'med' file format is our weapon of choice.
```



5 3D beam

```
#
# 'RESU' - b_extrac = what do we want to extract and write from the
# calculation? = SIGM_NOEU_DEPL,EQUI_NOEU_SIGM,DEPL
#
# Stress at the nodes from the displacement (what, where, how)
#
# EQUIvalent Nodal Stress
#
# and Displacement.

IMPR_RESU (FORMAT='MED',
           RESU=_F (RESULTAT=RESU,
                   NOM_CHAM=
                   ('EQUI_NOEU_SIGM','DEPL','EQUI_ELGA_SIGM','SIEF_ELGA_DE
PL',
                   'FORC_NODA','SIGM_NOEU_DEPL',),),),);
```

5.2.4 Post-processing options in the command file

Creating node groups from element groups is necessary in order to perform certain actions with the keyword **POST_RELEV_T**

#Create / define a new node group based on an element group

```
mesh=DEFI_GROUP (reuse =mesh,
                MAILLAGE=mesh,
                CREA_GROUP_NO=_F (GROUP_MA='Load',
                                   NOM='Load_no',),),);
```

#Extract reactions on nodes.

#Operation of 'actions' is to extract forces on nodes from the FORC_NODA field calculated by CALC_NO.

#Use the node groups previously defined (element groups obviously can't be used) for each 'action' respectively.

#Extract the 'resultant force' in the DZ direction on the nodes.

#Note: Probably self-evident, but using RESULTANT calculates the resulting vector of each individual vector on each node. Using the keyword NOM_CMP=DZ (name_of_component) will print a list of nodes in the group and their corresponding vector.

#Third actions extracts the displacement of a single node at the free end of the beam (node id. found i Salomé)

#Forth action computes the resulting moment from the forces in Fix group at a specific POINT (0,0,0)

```
Table=POST_RELEV_T (ACTION=( _F (OPERATION='EXTRACTION',
                                  INTITULE='Force_Fix',
                                  RESULTAT=RESU,
                                  NOM_CHAM='FORC_NODA',
                                  GROUP_NO='Fix_no',
```



5 3D beam

```
RESULTANTE='DZ',),
_F (OPERATION='EXTRACTION',
    INTITULE='Force_Load',
    RESULTAT=RESU,
    NOM_CHAM='FORC_NODA',
    GROUP_NO='Load_no',
    RESULTANTE='DZ',),
_F (OPERATION='EXTRACTION',
    INTITULE='Depl_Load',
    RESULTAT=RESU,
    NOM_CHAM='DEPL',
    NOEUD='N1458',
    NOM_CMP='DZ',),
_F (OPERATION='EXTRACTION',
    INTITULE='Moment_Fix',
    RESULTAT=RESU,
    NOM_CHAM='FORC_NODA',
    GROUP_NO='Fix_no',
    RESULTANTE=('DX','DY','DZ',),
    MOMENT=('DRX','DRY','DRZ',),
    POINT=(0,0,0,),),),);

#Print maximum and minimum values from the SIEF_ELGA_DEPL field to a text
file with the unit number 9

IMPR_RESU (MODELE=Meca,
    FORMAT='RESULTAT',
    UNITE=9,
    RESU=_F (RESULTAT=RESU,
        NOM_CHAM='SIEF_ELGA_DEPL',
        FORM_TABL='OUI',
        VALE_MAX='OUI',
        VALE_MIN='OUI',
        IMPR_COOR='NON',),),);

#Print the table with extracted forces, displacement and moment, to a
text file with the unit number 9 - only show parameters DZ and MOMENT_Y

IMPR_TABLE (TABLE=Table,
    UNITE=9,
    NOM_PARA=('DZ','MOMENT_Y',),),);

FIN();
```

5.3 Post-processing

5.3.1 Reviewing the textual output

Since we specifically asked Code Aster® to output values to the text file with unit number 9, this is the file we have to review, and not the default *.resu* file.

Starting with forces, displacement and moment, the (primitive) filtration of parameters has cut off the names of the rows, so the list corresponds like this:

- Resulting force on fixed end ('fix_no')
- Resulting force on free end ('load_no')
- Displacement of node
- Moment resulting from forces from group 'fix_no' at point [0,0,0]

```
Table CALCULE LE 19/04/2011 A 11:56:52 DE TYPE
#TABLE_SDASTER
DZ          MOMENT_Y
 1.02440E+04 -
-1.02440E+04 -
-2.07089E+00 -
-          -3.07320E+07
```

From the **SIEF_ELGA_DEPL** field, the normal stress of the beam is the **SIXX** component:

```
CHAMP PAR ELEMENT AUX POINTS DE GAUSS DE NOM SYMBOLIQUE  SIEF_ELGA_DEPL
NUMERO D'ORDRE: 1 INST:  0.00000E+00

LA VALEUR MAXIMALE DE SIXX          EST  9.80867E+00 EN  1
MAILLE(S) : M10581
```

5.3.2 Post-processing in Salomé

Field and component is noted in each image (figures 5.6, 5.7 and 5.8)

5 3D beam

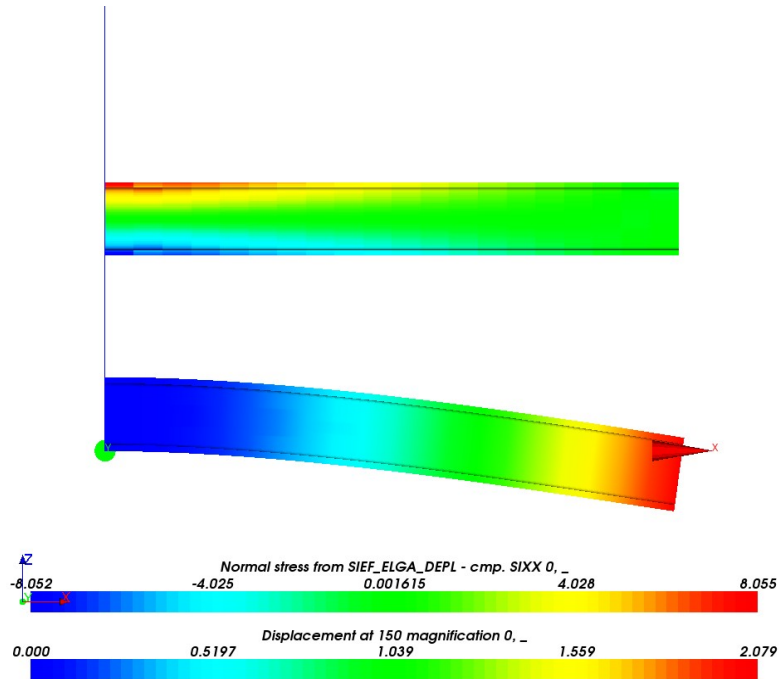


Figure 5.6: Normal stress and displacement – side view

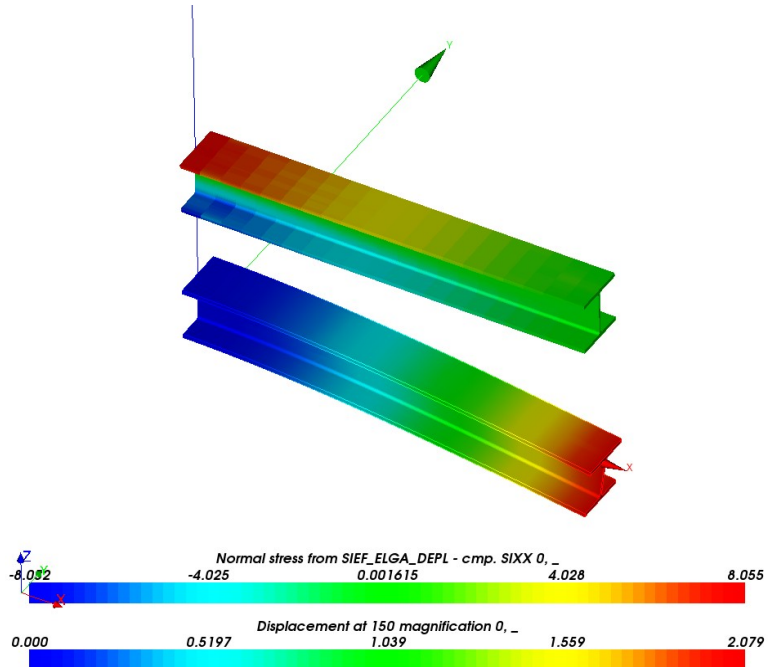


Figure 5.7: Normal stress and displacement – arbitrary view

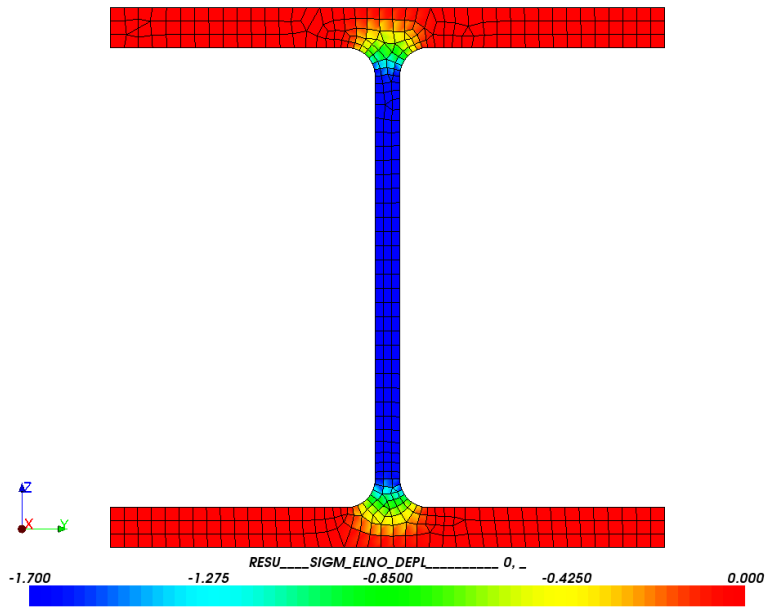


Figure 5.8: Shearing stress: SIXZ cmp – Cut plane at ~1.5m
(or use clipping)

5.4 Comparing results – analytical to Code Aster® (3D)

	Analytical	Code Aster®	Percentage
Displacement, free end	-1.92 mm	-2.07089E+00 mm	7.29 %
Angle of slope	9.58E-4 radians	N/A	N/A
Normal stress, bending	7.9 MPa	8.055 MPa	1.9 %
Shearing stress (average)	-3.7E-1 MPa	N/A	N/A
Shearing stress (web)	-1.81 MPa	-1.77 MPa	2.2 %
Shearing stress (AZ)	-1.675 MPa	-1.77 MPa	5.4 %

5.5 ASTK setup for 3D calculation

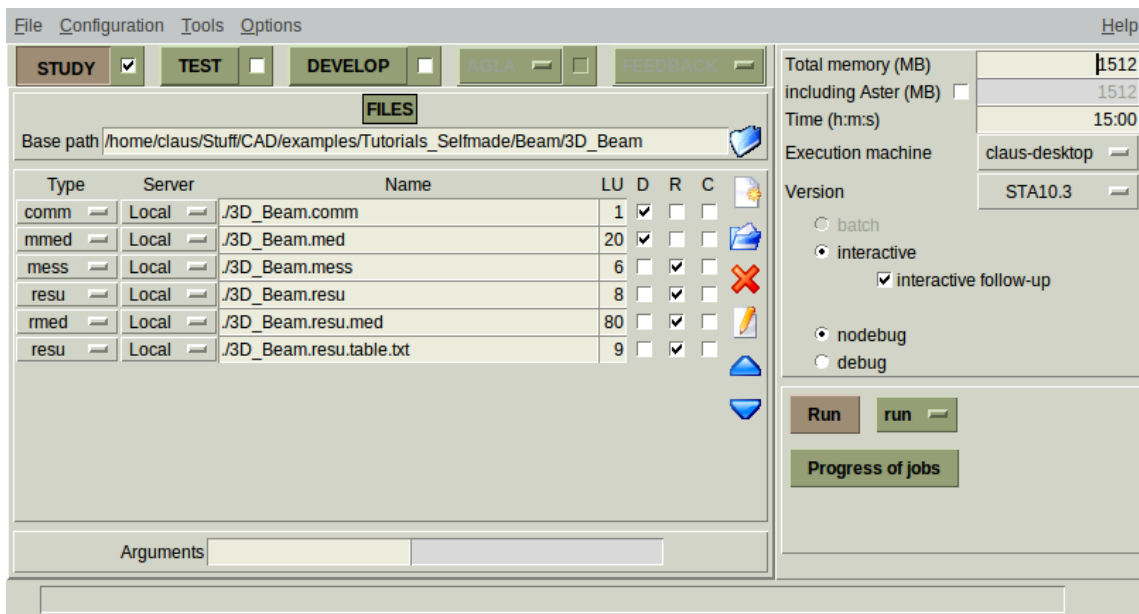


Figure 5.9: ASTK setup – note last entry (unit 9)



6 Conclusion, remarks and author(s)

That's it for this tutorial. Much more information can be found in the user documents on the Code Aster® website, its forum and on the CAELinux website.

Remark:

Any and all information and content in this document is published under the GPL license and can as such be used or reproduced in any way. The author(s) only ask for acknowledgment in such an event.

Acknowledgment goes out to EDF for releasing Code Aster® as free software and to all those who help out by answering questions in the forum and writing documentation / tutorials.

Contributions and/or corrections to this tutorial are always welcomed.

Author(s):

Claus Andersen – ClausAndersen81_[at]_gmail.com

ENDED OK



7 Links

[CAELinux Website] www.caelinux.com

[Code Aster® Website] www.code-aster.org

[GMSH® Website] www.geuz.org/GMSH

[XMGrace® Website] <http://plasma-gate.weizmann.ac.il/Grace/>