



METHODE PAR ELEMENTS FINIS

| Code_Aster / Plateforme Salome_Meca |

Table des matières

1	Introduction	4
1.1	Code_Aster	5
1.2	Salome_Meca	5
2	Installation du logiciel	6
2.1	Installation d'un logiciel de virtualisation	6
2.2	Installation d'un système d'exploitation Linux - Ubuntu	7
2.3	Salome_Meca & Code_Aster	8
3	Cas d'étude – Poutre en porte-à-faux	11
3.1	Présentation du cas d'étude	11
4	Calcul Analytique – Poutre en Flexion	11
4.1	Détermination du moment quadratique	11
4.2	Flèche de la poutre	12
4.3	Contrainte de la poutre au point de fixation	12
5	Modélisation avec Salome_Meca – Interface Graphique	13
5.1	Introduction à Salome_Meca	13
5.2	Module Géométrique Salome_Meca	14
5.2.1	Présentation du module géométrique	14
5.2.2	Préparation de la géométrie de la poutre	17
5.2.3	Présentation des principaux outils du module géométrique	27
5.3	Module Maillage Salome_Meca	29
5.3.1	Présentation du module de Maillage	29
5.3.2	Création du maillage de la géométrie	30
5.3.3	Affiner le maillage d'une géométrie	37
5.3.4	Outils de vérification du module Maillage	38
5.3.5	Affichage des numéros de nœuds et de l'orientation des faces et plan de coupe	39
5.3.6	Exportation d'un maillage	40
5.4	Paramétrisation de l'analyse – Module AsterStudy	41
5.4.1	Lecture du maillage :	42
5.4.2	Affectation du type de maillage	43
5.4.3	Définition du matériau	44
5.4.4	Affectation du matériau au maillage	45
5.4.5	Assignement des chargements et des conditions aux limites	47
5.4.6	Définition du type d'analyse	50
5.4.7	Déclaration des types de résultats souhaité	51

5.4.8	Extraire des résultats en format Texte	52
5.4.9	Lancement du calcul.....	53
5.5	Analyse des résultats ParaVis / ParaView.....	56
5.5.1	Afficher le déplacement résultant.....	58
5.5.2	Animation de la vue.....	62
5.5.3	Affichage des contraintes	63
5.5.4	Réalisation d'une vue en coupe sur la vue de résultats	64
6	Modélisation avec Salome_Meca – Script commande Python.....	66
6.1	Introduction & présentation du langage de programmation Python.....	66
6.2	Editeur de texte – Sublime texte.....	67
6.3	Script python pour réaliser le maillage et la géométrie	68
6.3.1	Création de la géométrie par un script Python	69
6.3.2	Création du maillage en script Python.....	72
6.3.3	Création du script de commande de l'analyse	73
6.4	Présentation de ASTK.....	75
6.4.1	Lancement de ASTK.....	75

1 Introduction

Ce document a pour but d'introduire et de présenter un logiciel de calcul par éléments finis puissant et performant intégralement gratuit et open source.

Une introduction sur les éléments finis avec tous les grands principes de base est également disponible sur le site « **Nadir Ingenia** ». Ce document part du principe que les notions de base sont déjà assimilées par le lecteur, il présente uniquement l'utilisation du logiciel.

Lien des supports de cours sur les éléments finis : <https://nadir-ingenia.fr/elements-finis/>

Il sera abordé dans ce document le code de calcul « **Code_Aster** » ainsi que la plateforme de calcul multi-physique libre « **Salome** », et plus précisément l'extension mécanique « **Salome_Meca** ».

L'outil de calcul y sera présenté au travers du calcul d'un exemple d'une poutre en porte à faux.

Le logiciel « **Salome_Meca** » intègre une interface graphique ainsi qu'une console exécutable Python permettant l'exécution de script Python capable d'automatiser et de piloter les calculs.

Dans un but pédagogique, l'exemple y sera alors présenté avec la méthodologie **utilisant l'interface graphique et en script Python**.

Toutes les étapes y seront décrites et détaillées pas à pas et le maximum d'outils seront utilisés dans le but de présenter les principaux outils disponibles.

Les avantages et les inconvénients sont principalement les suivants :

Avantages	Inconvénients
<ul style="list-style-type: none"> • Libre de droit : Gratuit • Open-source • Console Python • Interface graphique • Puissant / Polyvalent • Diversités des lois matériaux • Diversités des éléments • Mailleur intégré • Editeur de géométrie • Logiciel Français • Documentation détaillée • Communauté d'utilisateur 	<ul style="list-style-type: none"> • Complexe d'utilisation • OS Linux*

* Il est à noter que pour une pleine utilisation un système d'exploitation **Linux** sera nécessaire. Une version Windows a été développée mais pour le couplage avec le script Python il est recommandé d'utiliser le logiciel dans sa version Linux. Dans le cas d'une utilisation sur un **poste Windows** il sera recommandé de passer par une **machine virtuelle** pour **émuler un système d'exploitation Linux**.

1.1 Code_Aster



« **Code_Aster** » est un solveur (module réalisant le calcul) libre de simulation numérique en mécanique des structures, développé principalement par le service de recherche et développement « d'EDF ».

Ce code de calcul a été réalisé initialement dans le but de dimensionner les infrastructures du génie civil du réseau électrique comme les centrales électriques, les barrages hydroélectriques, les pylônes etc...

Site Web : <https://www.code-aster.org/>

Documentation : <https://www.code-aster.org/V2/spip.php?rubrique5>

Forum : <https://www.code-aster.org/forum2/>

1.2 Salome Meca



« **Salome_Meca** » est une extension mécanique du logiciel « **Salome** » servant de socle et de plateforme de calcul scientifique multi-physique développé par « EDF », le « CEA » et « Open Cascade ».

Ce logiciel est un tout en un, il comprend une interface graphique intégrant le solveur « **Code_Aster** » ainsi qu'un éditeur de géométrie, un logiciel de maillage, de pré-traitement et post-traitement.

Site Web : <https://www.salome-platform.org/>

2 Installation du logiciel

2.1 Installation d'un logiciel de virtualisation

Linux représentant une part très faible du marché (moins de 3% des postes de travail) il vous sera alors sûrement nécessaire d'utiliser une **machine virtuelle** pour émuler numériquement un poste avec un **système d'exploitation Linux**.

Remarque : Une autre possibilité est d'utiliser ce qu'on appelle du **multiboot** en installant plusieurs systèmes d'exploitation sur un même poste. Un **serveur linux** peut également être une solution très pertinente dans le cas d'utilisation par plusieurs personnes. Ces solutions sont plus complexes à mettre en œuvre et sont moins flexibles, elles ne seront pas détaillées dans ce rapport, néanmoins elles peuvent être pertinentes dans le cas d'une utilisation plus poussée du logiciel.

Il existe un ensemble de logiciel permettant la virtualisation de système d'exploitation. Ici il sera évoqué la solution gratuite proposée par la société « **Oracle** » avec le logiciel « **VirtualBox** ».



Site Web : <https://www.virtualbox.org/>

La version du logiciel de la machine virtuelle n'a pas d'importance, la dernière version du logiciel peut être installée.

Lien de téléchargement : <https://www.virtualbox.org/wiki/Downloads>



The screenshot shows the 'Download VirtualBox' page on the Oracle website. It features a sidebar with navigation links like 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. The main content area is titled 'Download VirtualBox' and includes a section for 'VirtualBox binaries' with a list of platform packages. The 'VirtualBox 6.1.18 platform packages' section is highlighted with a red box and lists: Windows hosts, OS X hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. Below this, there are instructions about the GPL license, a changelog link, and a note about checksums (SHA256 and MD5).

Sélectionner la plateforme correspondant au système d'exploitation hôte « **Windows hosts** » pour les postes Windows.

2.2 Installation d'un système d'exploitation Linux - Ubuntu

Maintenant que le logiciel de virtualisation est installé, il est nécessaire de créer une nouvelle machine virtuelle. Pour se faire il sera nécessaire en amont de **télécharger un système d'exploitation Linux**.

Il existe plusieurs systèmes d'exploitation **Linux** le plus utilisé étant « **Debian** » et notamment une de ses variantes nommé « **Ubuntu** ». C'est celui-ci que nous utiliserons dans ce document.

Ubuntu est un système d'exploitation gratuit et libre.



Site Web : <https://www.ubuntu-fr.org/>

Téléchargement du fichier ISO : <https://www.ubuntu-fr.org/download/>

A screenshot of the Ubuntu download page. The title "Téléchargements" is at the top in orange. Below it, there is a section "Choisissez la variante :" with a dropdown menu showing "Ubuntu". Underneath, there is a section "Liens de téléchargement direct :" with three options: "dernière version LTS (recommandée)" (highlighted with a red box), "dernière version intermédiaire", and "Version Bittorent" with an unchecked checkbox.

Le fichier ISO contient l'installateur du système d'exploitation, il sera nécessaire pour créer la boîte virtuelle. Le détail de l'installation de Ubuntu via la machine virtuelle n'étant pas le sujet de ce document, cette installation ne sera pas détaillée, il existe un ensemble de support et de vidéo sur internet sur ce sujet.

Lien de la documentation Ubuntu : <https://doc.ubuntu-fr.org/accueil>

Remarque : Attention un ensemble de choses et d'opérations sur Ubuntu seront à réaliser via le terminal de commande, notamment l'installation de certains logiciels. Des recherches au préalable sur la prise en main de ce système d'exploitation peut-être intéressante pour des utilisateurs novices.

2.3 Salome Meca & Code Aster

Dans ce document la version de Salome_Meca utilisé sera la suivante :

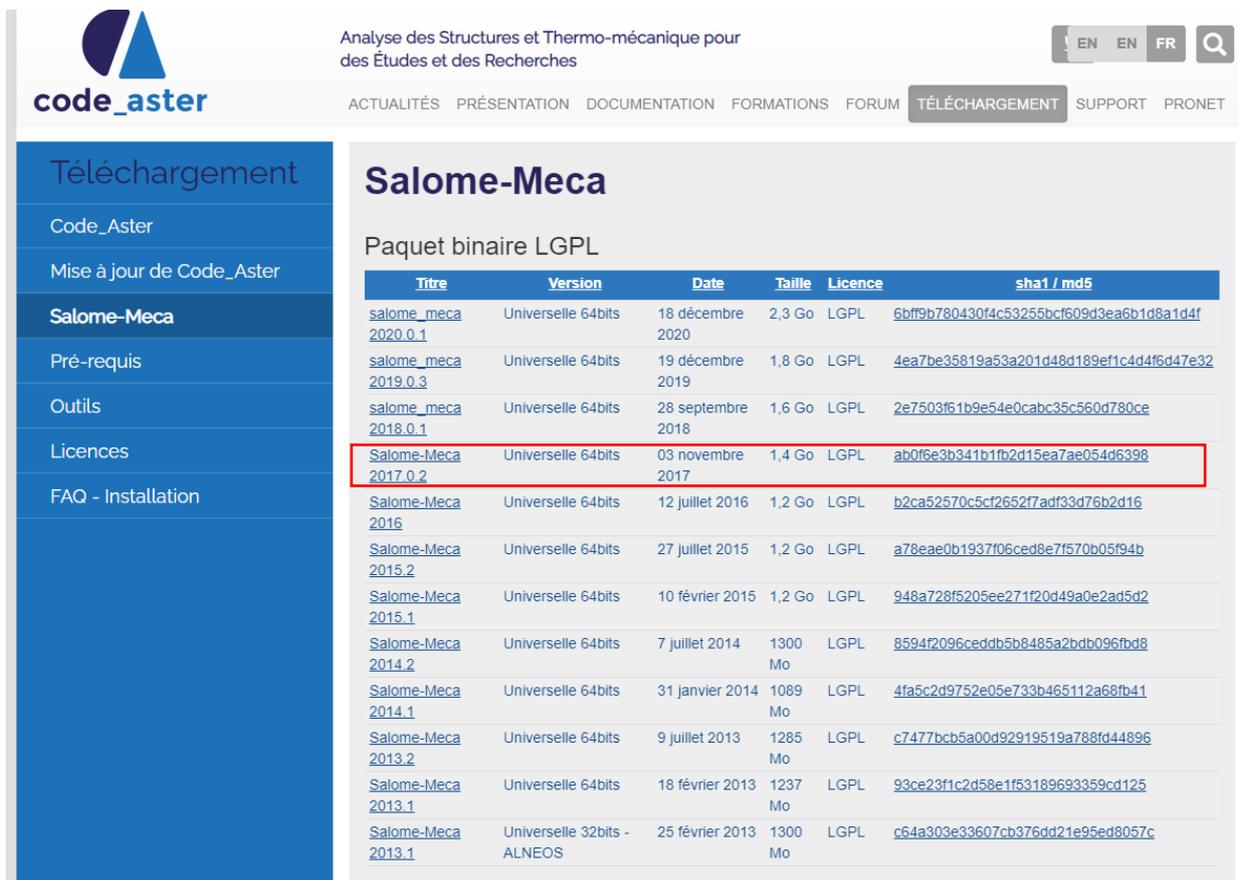
Salome_Meca 2017 0.2 (SALOME 8.3.0)

Remarque : Des versions plus récentes sont disponible, néanmoins cette version dispose d'un ensemble de documentations et de supports et a éprouvé sa stabilité. Il est à noter que les grands principes ainsi que la philosophie et les grands concepts du logiciel resteront identiques.



La version téléchargeable ci-dessous est une version **Linux**.

Lien du téléchargement : <https://www.code-aster.org/spip.php?article295>



code_aster

Analyse des Structures et Thermo-mécanique pour des Études et des Recherches

EN EN FR

ACTUALITÉS PRÉSENTATION DOCUMENTATION FORMATIONS FORUM TÉLÉCHARGEMENT SUPPORT PRONET

Téléchargement

- Code_Aster
- Mise à jour de Code_Aster
- Salome-Meca**
- Pré-requis
- Outils
- Licences
- FAQ - Installation

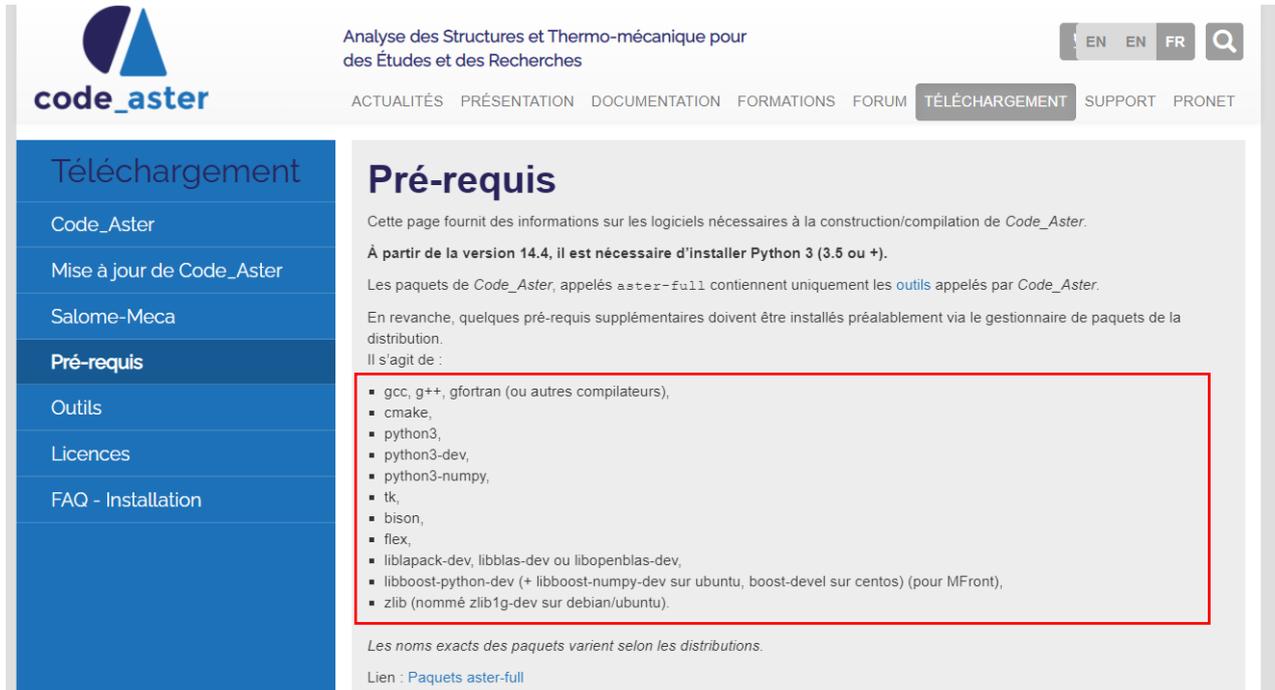
Salome-Meca

Paquet binaire LGPL

Titre	Version	Date	Taille	Licence	sha1 / md5
salome_meca 2020.0.1	Universelle 64bits	18 décembre 2020	2,3 Go	LGPL	6bff9b780430f4c53255bcf609d3ea6b1d8a1d4f
salome_meca 2019.0.3	Universelle 64bits	19 décembre 2019	1,8 Go	LGPL	4ea7be35819a53a201d48d189ef1c4d4f6d47e32
salome_meca 2018.0.1	Universelle 64bits	28 septembre 2018	1,6 Go	LGPL	2e7503f61b9e54e0cab3c35c560d780ce
Salome-Meca 2017.0.2	Universelle 64bits	03 novembre 2017	1,4 Go	LGPL	ab0f6e3b341b1fb2d15ea7ae054d6398
Salome-Meca 2016	Universelle 64bits	12 juillet 2016	1,2 Go	LGPL	b2ca52570c5cf2652f7adf33d76b2d16
Salome-Meca 2015.2	Universelle 64bits	27 juillet 2015	1,2 Go	LGPL	a78eae0b1937f06ced8e7f570b05f94b
Salome-Meca 2015.1	Universelle 64bits	10 février 2015	1,2 Go	LGPL	948a728f5205ee271f20d49a0e2ad5d2
Salome-Meca 2014.2	Universelle 64bits	7 juillet 2014	1300 Mo	LGPL	8594f2096ceddb5b8485a2bdb096fbd8
Salome-Meca 2014.1	Universelle 64bits	31 janvier 2014	1089 Mo	LGPL	4fa5c2d9752e05e733b465112a68fb41
Salome-Meca 2013.2	Universelle 64bits	9 juillet 2013	1285 Mo	LGPL	c7477bcb5a00d92919519a788fd44896
Salome-Meca 2013.1	Universelle 64bits	18 février 2013	1237 Mo	LGPL	93ce23f1c2d58e1f53189693359cd125
Salome-Meca 2013.1	Universelle 32bits - ALNEOS	25 février 2013	1300 Mo	LGPL	c64a303e33607cb376dd21e95ed8057c

Un ensemble de prérequis seront nécessaires à installer en amont pour le bon fonctionnement de Salome_Meca / Code_Aster.

Lien des prérequis : <https://www.code-aster.org/spip.php?article91>



The screenshot shows the 'Pré-requis' (Prerequisites) page on the Code_Aster website. The page title is 'Pré-requis' and it provides information on the software needed for construction/compilation. It states that from version 14.4, Python 3 (3.5 or higher) is required. A list of packages is provided, including gcc, g++, gfortran, cmake, python3, python3-dev, python3-numpy, tk, bison, flex, liblapack-dev, libblas-dev, libopenblas-dev, libboost-python-dev, and zlib. A red box highlights this list. Below the list, it notes that package names vary by distribution and provides a link to 'Paquets aster-full'.

Remarque : On appelle « **paquet** » les fichiers à télécharger pour l’installation de logiciels, pour les télécharger sur **Ubuntu** il est nécessaire de passer par le **terminal**. Le mot de passe de l’administrateur (sudo) de celui qui a installé le poste Linux sera demandé pour réaliser l’installation de ces paquets.



Attention : Pour la version 2017 de Salome_Meca et Code_Aster (version de ce rapport) la version de python 3 n’est pas encore supportée. Il faut donc remplacer ici « **python3** » par « **python** » pour l’installation des paquets. (Les deux versions de python peuvent être installées au besoin).

La commande d’installation est la commande suivante : « *sudo apt install nom_paquet* »

Une fois l’installation de tous les paquets prérequis effectuée, il sera alors ensuite possible d’installer Salome_Meca sur le poste.

Pour lancer l'installation il est nécessaire d'exécuter la commande :

```
« ./.../Salome-Meca-2017.0.2-LGPL-2.run »
```

Remarque : /.../ correspond au répertoire où se situe le fichier.

Lors de l'installation il sera demandé dans quel répertoire installer les fichiers du logiciel.

Attention : Il est primordial dans l'utilisation de Linux de faire attention au minuscules / MAJUSCULE et d'éviter l'utilisation de caractères spéciaux, d'accents ou d'espaces dans le nom des fichiers, l'utilisation de l'underscore « _ » est à favoriser pour remplacer les espaces « ».

Dans le cas de difficulté d'installation, la procédure est expliquée dans la partie téléchargement du site de Code_Aster et des explications en FAQ sont également disponibles. Un fichier « README » est aussi présent avec un ensemble d'explications.

Une fois la pré-installation terminée il est recommandé de finaliser l'installation en exécutant le script suivant afin de créer un launcher exécutable sur le bureau (clicable à la souris) :

```
« sudo ./.../Salome_meca/V2017.0.2/salome_post_install.py »
```

Il est également possible de lancer Salome via le terminal avec la commande suivante :

```
« sudo ./.../Salome_meca/appli_V2017.0.2/Salome »
```

Remarque : /.../ correspond à votre répertoire où se situe les fichiers.

En cas de difficulté ne pas hésiter à jeter un coup d'œil au forum de Code_Aster. Une communauté française et anglaise s'est créée autour de ce logiciel et de nombreux problèmes d'utilisateur y ont déjà été résolus.

Le logiciel est désormais installé et fonctionnel, il peut donc être utilisé.

3 Cas d'étude – Poutre en porte-à-faux

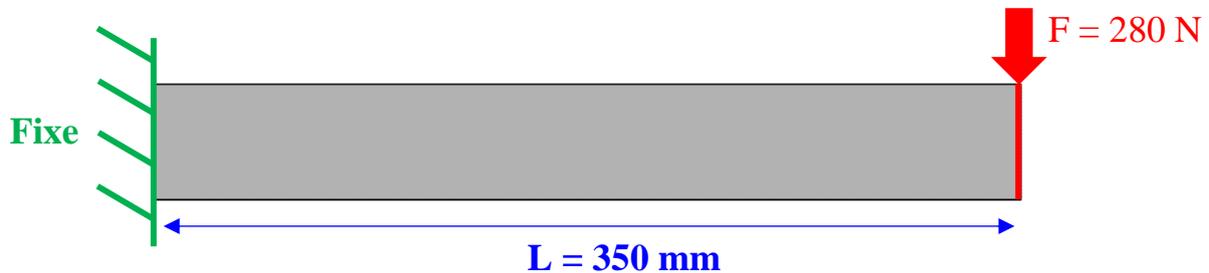
Afin de présenter le logiciel et ses fonctionnalités l'exemple d'une poutre en porte-à-faux sera étudié.

Une corrélation entre résultats par le calcul éléments finis et calcul analytique sera réalisé.

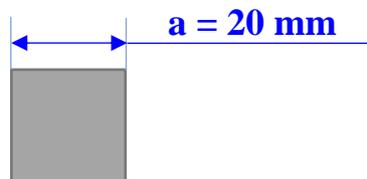
Pour la modélisation d'une poutre par la méthode des éléments finis, il existe des simplifications géométriques et des réductions de la dimension du problème possible. Néanmoins, dans le but de présenter le maximum d'outils et fonctionnalités du logiciel aucune simplification ne sera réalisée.

3.1 Présentation du cas d'étude

Le cas étudié est le suivant :



Section de la poutre carrée :



Matériau : **Acier**

Module d'Young : **210 000 MPa**

4 Calcul Analytique – Poutre en Flexion

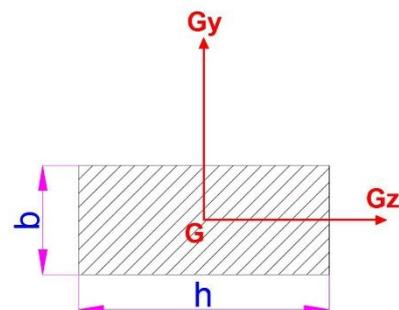
Le cas d'étude correspond à l'étude de flexion d'une poutre encastrée / libre.

4.1 Détermination du moment quadratique

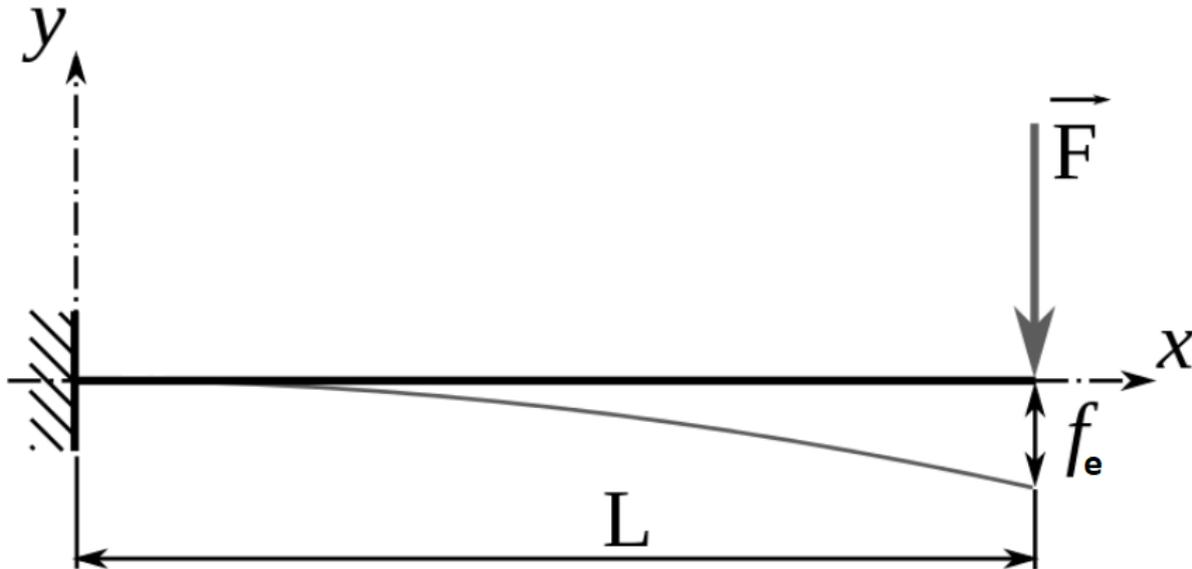
Le moment quadratique de la poutre est de :

Dans notre cas de figure : $b = h$ (Section carrée)

$$I_{GZ} = \frac{b^3 h}{12} = \frac{20^3 \times 20}{12} = 1,33 \times 10^{-4} \text{ mm}^4$$



4.2 Flèche de la poutre



La flèche de la poutre est de :

$$f_e = \frac{FL^3}{3EI} = \frac{280 \times 350^3}{3 \times 210\,000 \times 1,33 \times 10^{-4}} = \mathbf{1,43\ mm}$$

4.3 Contrainte de la poutre au point de fixation

La contrainte de la poutre au point de fixation est de :

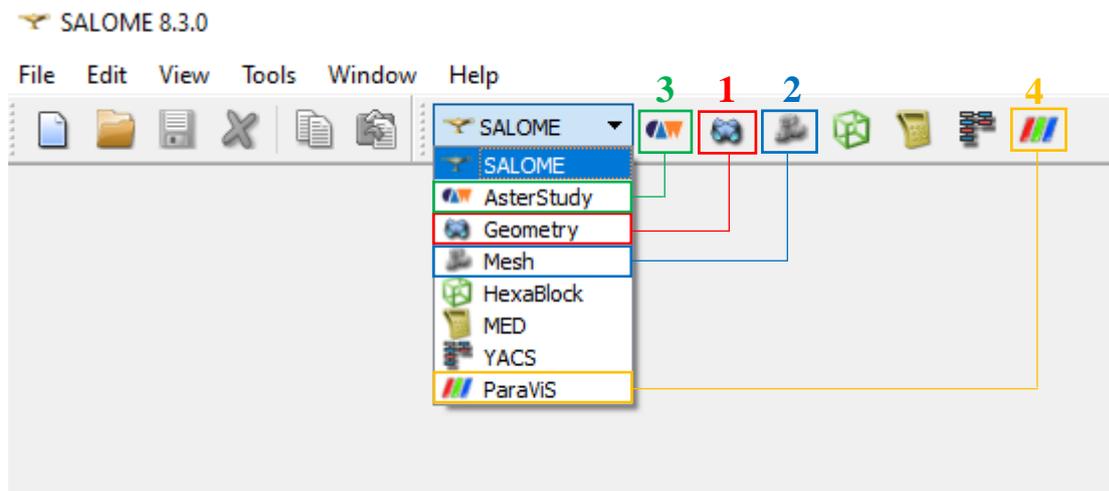
$$\sigma = \frac{FL}{\left(\frac{I}{v}\right)} = \frac{280 \times 350}{\left(\frac{1,33 \times 10^{-4}}{10}\right)} = \mathbf{73,50\ MPa}$$

5 Modélisation avec Salome Meca – Interface Graphique

Dans un but pédagogique l'intégralité de la modélisation sera réalisée avec l'interface graphique, puis dans un second temps elle sera réalisée de nouveau en script python.

5.1 Introduction à Salome Meca

Salome_Meca est une plateforme intégrant **plusieurs modules**, chaque module ayant un rôle bien spécifique. Dans ce document nous aborderons uniquement les modules « classiques » permettant de réaliser une étude mécanique par éléments finis complète du pré-traitement au post-traitement



Dans ce document nous utiliseront les modules dans l'ordre suivant :

Module « Geometry » - 1

Module géométrique permettant de réaliser et de travailler des points, des surfaces, des solides. Le logiciel permet également l'importation de géométrie via un fichier « **STEP** », « **IGES** » ou « **STL** ».

Module « Mesh » - 2

Module permettant de réaliser le maillage de la géométrie.

Module « AsterStudy » - 3

Module permettant de réaliser le paramétrage de l'analyse et de lancer le calcul.

Module « ParaVis / ParaView » - 4

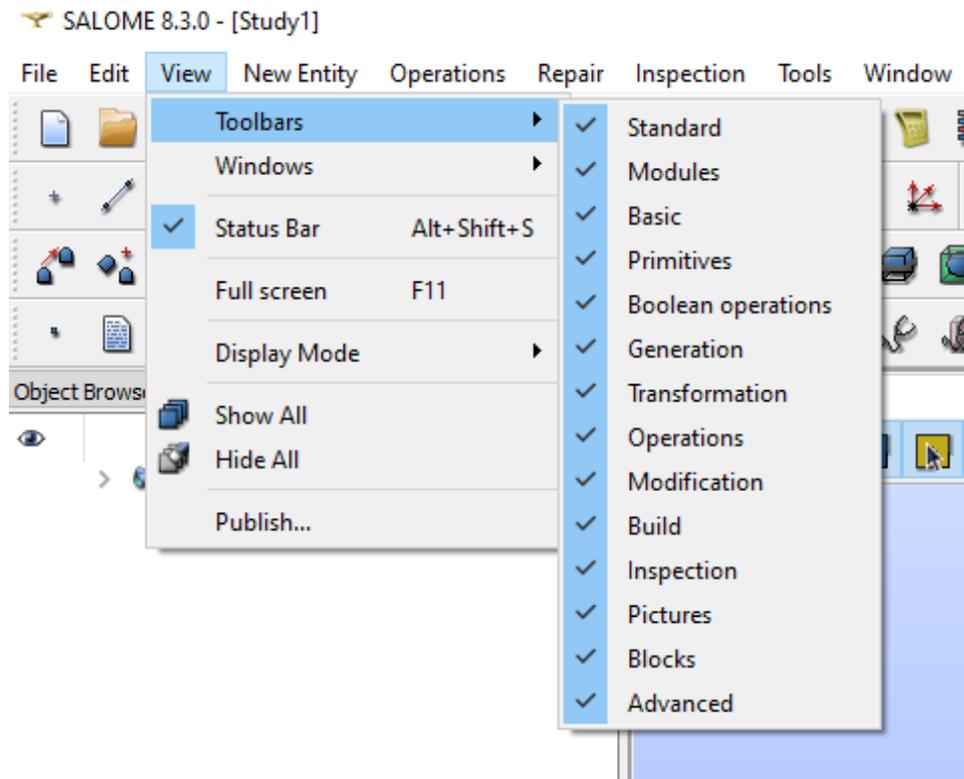
Module permettant de post-traiter les résultats de l'analyse.

5.2 Module Géométrique Salome Meca

5.2.1 Présentation du module géométrique

Conseil : Avant de commencer l'explication sur ce module, il est préférable d'afficher toutes les barres d'outils de manière à avoir un accès rapide à un maximum des outils du module.

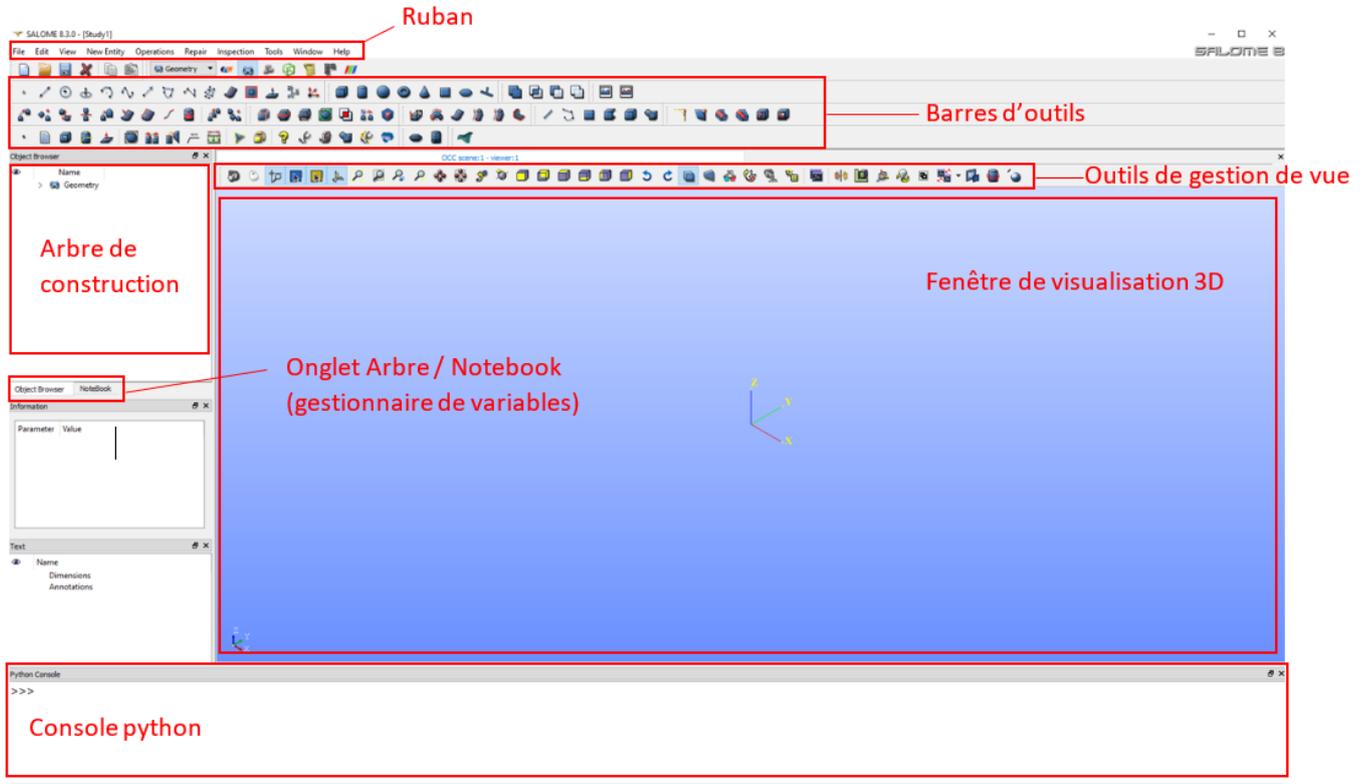
Pour ce faire il faut réaliser l'opération suivante :



Toutes les barres d'outils ne seront pas utilisées dans cette présentation, mais les afficher toutes permettent de voir toutes les fonctionnalités disponibles. Les outils principaux d'une analyse « classique » y seront utilisés au travers de l'exemple de ce cours.

Lien documentation module géométrie : <https://docs.salome-platform.org/latest/gui/GEOM/index.html>

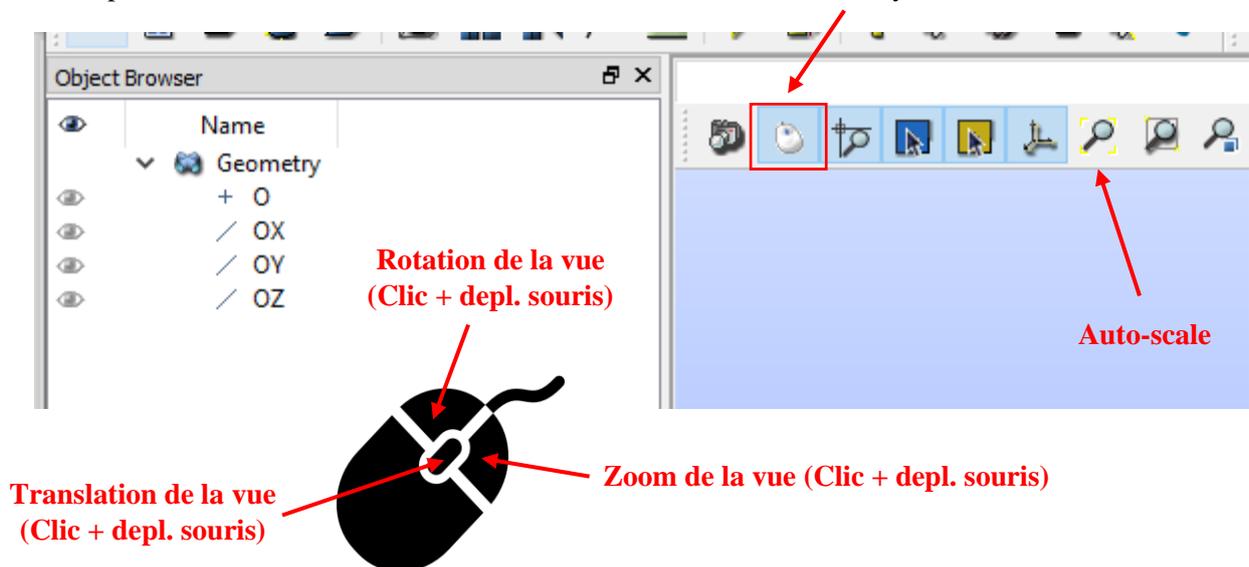
Présentation générale de l'interface :



Remarque : Il est important de comprendre qu'il n'y a pas de « précédent » (Ctrl + Z) comme dans les logiciels plus classiques qu'on a l'habitude d'utiliser. Chaque action génère une nouvelle structure, mais chaque action précédente est conservée.

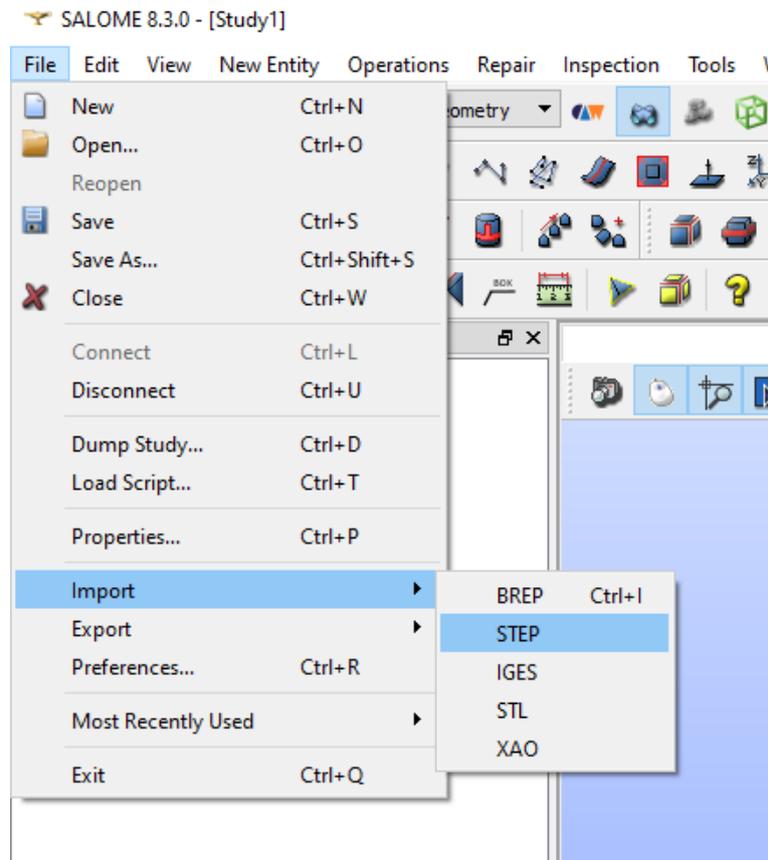
Gestion de la vue 3D :

Pour déplacer la vue 3D il est conseillé d'activer le bouton « Interaction style switch »

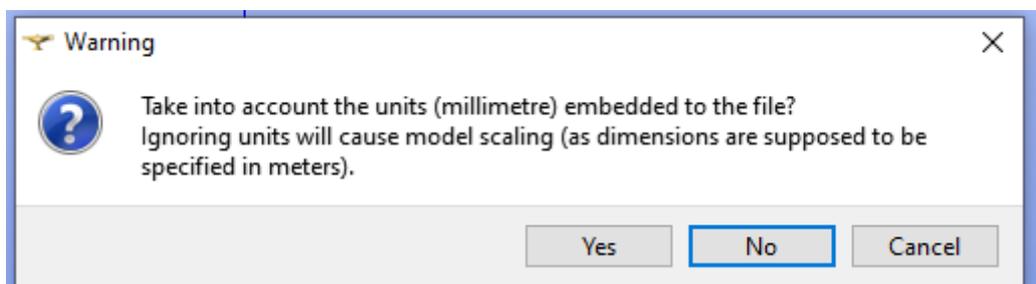


Importation de fichier STEP :

Le format « **STEP** » est à privilégier pour des échanges de fichier 3D.



Attention aux unités de dimensions lors de l'import de fichier « STEP ». Lors de l'import cette fenêtre s'ouvre, de manière générale il faut cliquer sur « No » (Dépend du logiciel de CAO utilisé).



Il faut toujours réaliser une mesure d'une distance connue une fois l'importation réalisée de manière à vérifier que les bonnes unités sont utilisées.

Remarque : Il faut faire **extrêmement attention aux unités**, de manière générale il est conseillé en éléments finis d'utiliser le système d'unités homogène « **MMNTS** » :

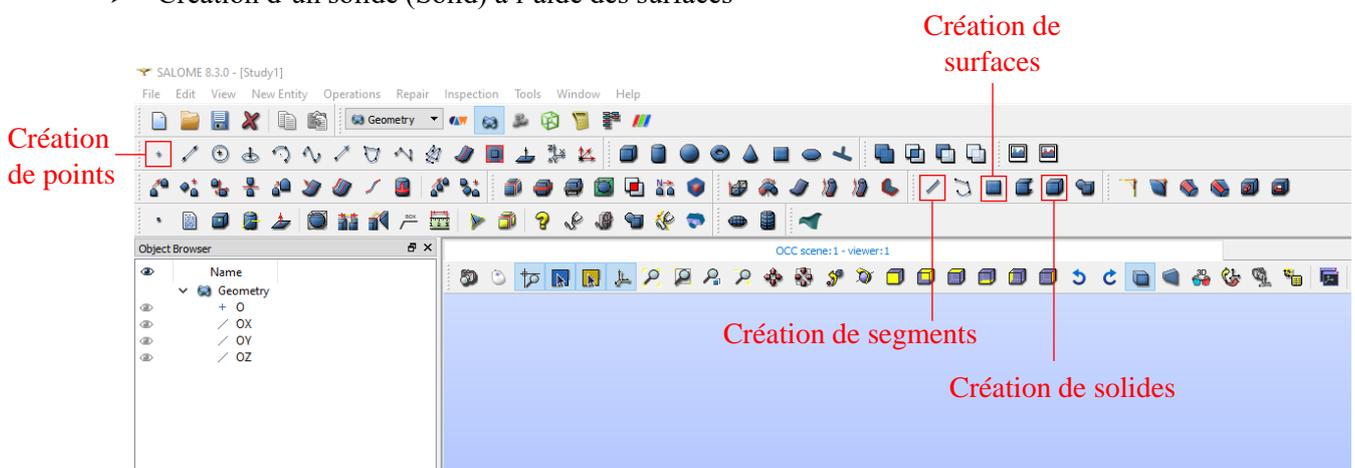
mm (millimètres) / N (Newton) / t (tonnes) / s (secondes)

5.2.2 Préparation de la géométrie de la poutre

Dans le but pédagogique, nous allons créer la géométrie de manière à utiliser et voir le plus de fonctionnalités possibles du logiciel et non de la façon la plus optimisée. Pour le cas simple d'un pavé droit comme ici une fonction existe au sein du logiciel. Cette fonction ne sera pas utilisée, le but est de comprendre le principe pour pouvoir réaliser des géométries plus complexes.

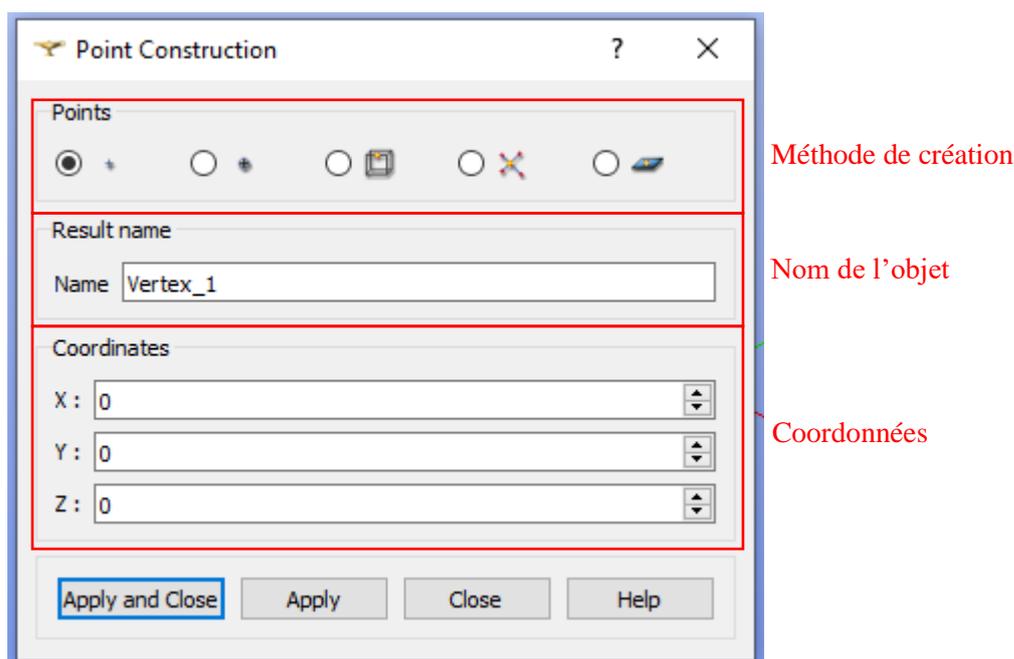
La démarche classique de la création d'un objet volumique 3D se fait de la façon suivante :

- Création de points (vertex)
- Création de segments (Edge) à partir des points
- Création de surfaces (Face) à l'aide des lignes
- Création d'un solide (Solid) à l'aide des surfaces



Création des points :

Pour information il existe plusieurs « méthode » pour la création d'objet, ici la création sera réalisée par les coordonnées cartésiennes du point souhaité. Une fois les champs renseignés il faut valider.

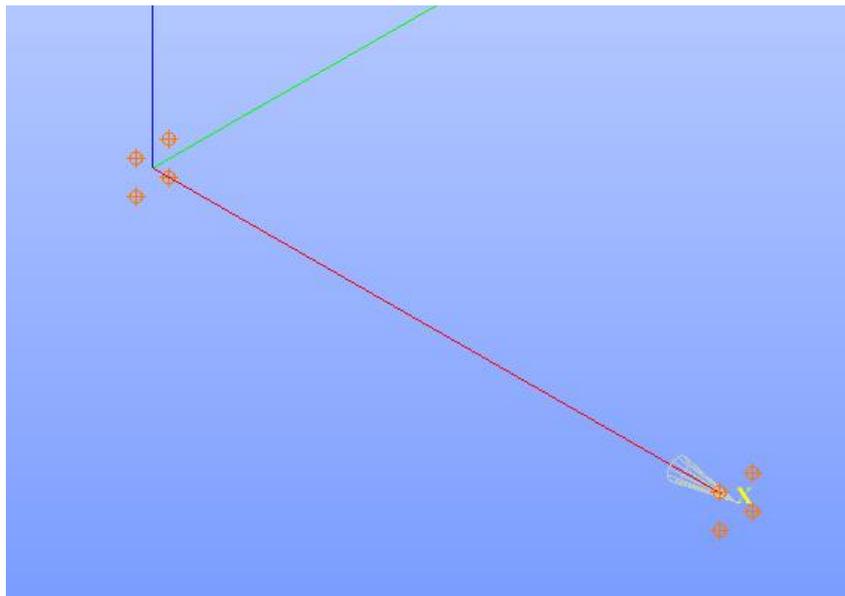


Tous les points de la poutre doivent être créé de la manière décrite ci-dessus.

Table de la coordonnée des points :

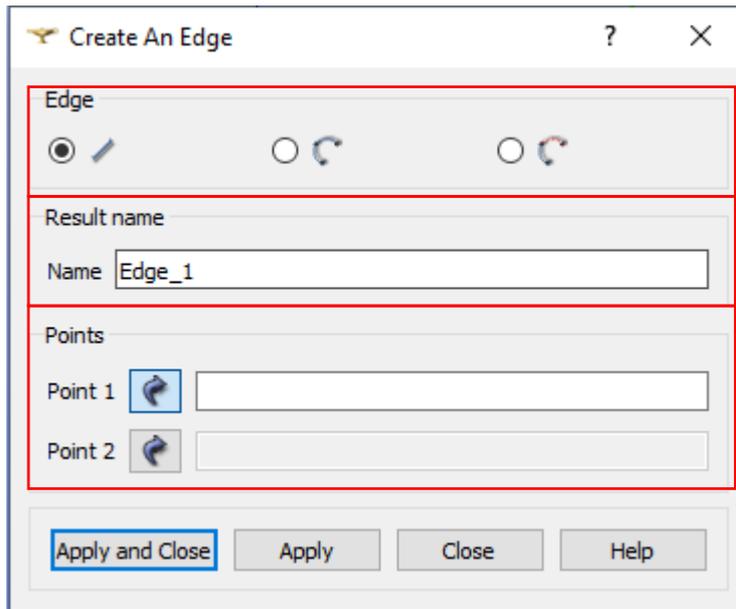
Points	X	Y	Z
P1	0	-10	-10
P2	0	-10	10
P3	0	10	10
P4	0	10	-10
P5	350	-10	-10
P6	350	-10	10
P7	350	10	10
P8	350	10	-10

Illustration de l'affichage de la fenêtre de visualisation 3D du module géométrique après la création des points :



Création des lignes :

Pour information il existe plusieurs « méthode » pour la création d'objet, ici la création sera réalisée par les points d'extrémité du segment souhaité. Les points peuvent être sélectionné via l'arbre géométrique ou directement dans la fenêtre de visualisation. Une fois les champs renseignés il faut valider.



Méthode de création

Nom de l'objet

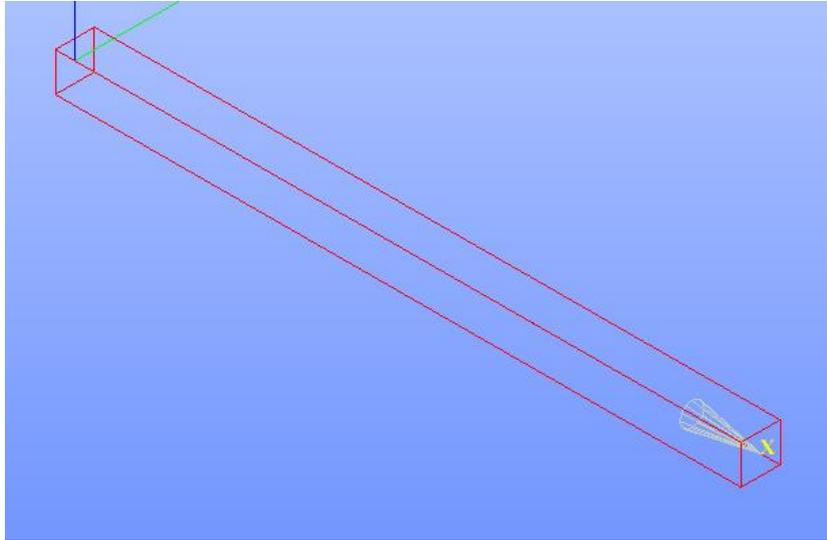
Sélection

Tous les segments de la poutre doivent être créés de la manière décrite ci-dessus.

Table de la création des segments :

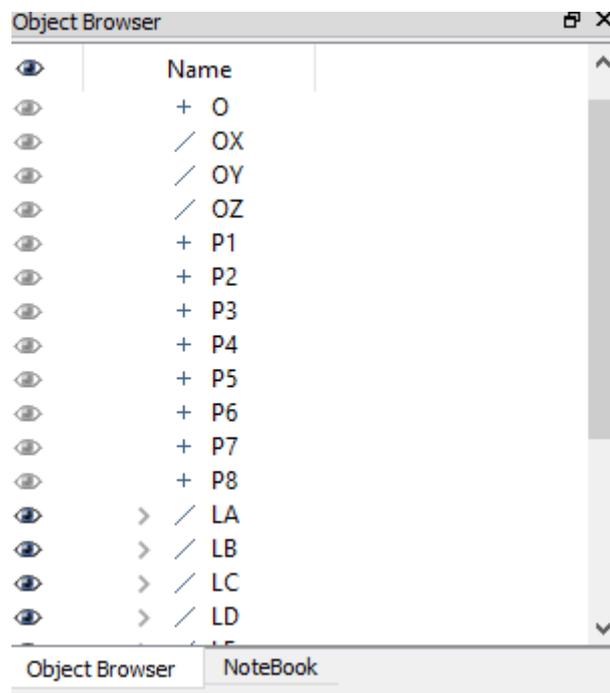
Segments	Points 1	Points 2
LA	P1	P2
LB	P2	P3
LC	P3	P4
LD	P4	P1
LE	P5	P6
LF	P6	P7
LG	P7	P8
LH	P8	P5
LI	P1	P5
LJ	P2	P6
LK	P3	P7
LL	P4	P8

Illustration de l'affichage fenêtre de visualisation après la création des segments :



Information sur l'arbre de géométrie

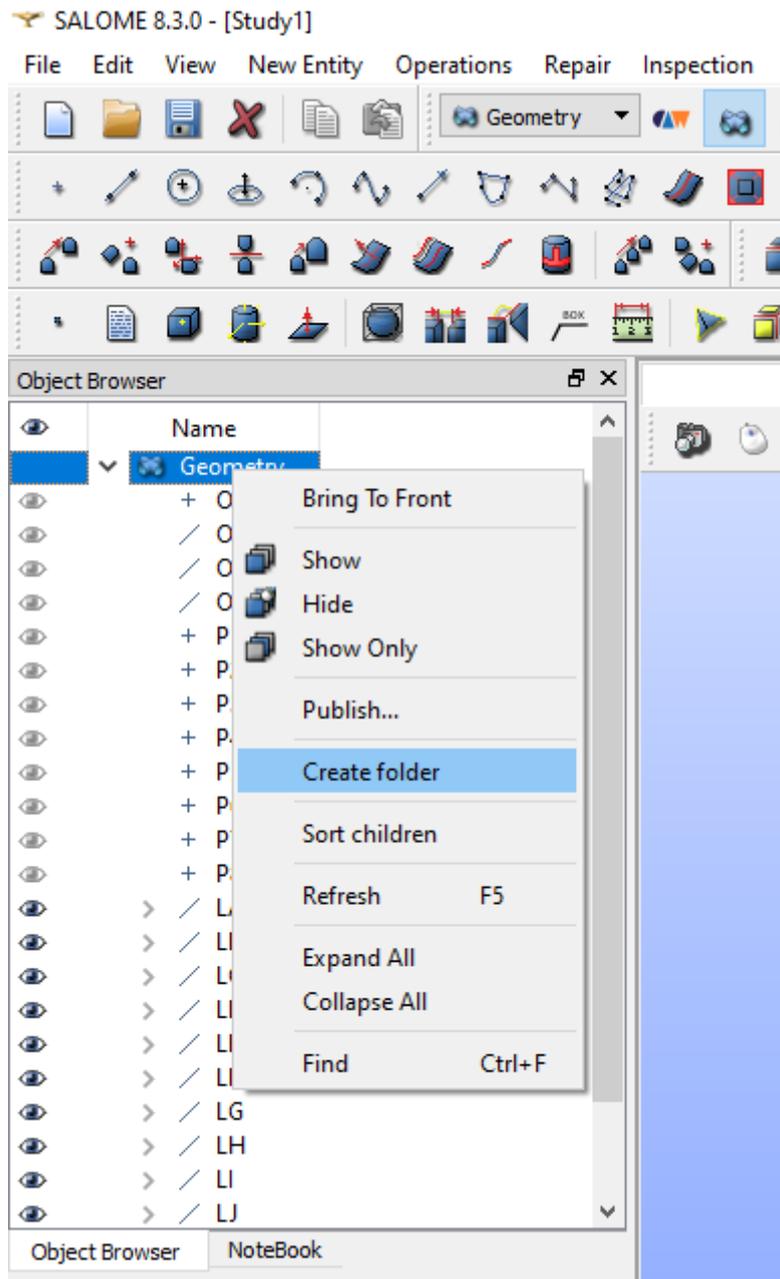
Il est à noter que chaque action sous Salome_Meca dans le module géométrie génère une nouvelle ligne dans l'arbre de géométrie, les éléments ne sont pas « consommés ». Chaque élément peut donc être réutilisé autant que souhaité. Il faut garder en tête que la touche « précédent / Ctrl + Z » n'existe pas sur Salome_Meca (ce qui peut être perturbant au début).



Création de dossier :

Il est à noter que des dossiers peuvent être créés pour classer des éléments, dans le cas de projets complexes cela devient vite indispensable. Connaître cette fonctionnalité facilite grandement l'exploitation. Les dossiers peuvent être renommés et on peut glisser/déposer avec la souris les éléments souhaités à l'intérieur.

Illustration de la méthode de création du document (clic droit) :

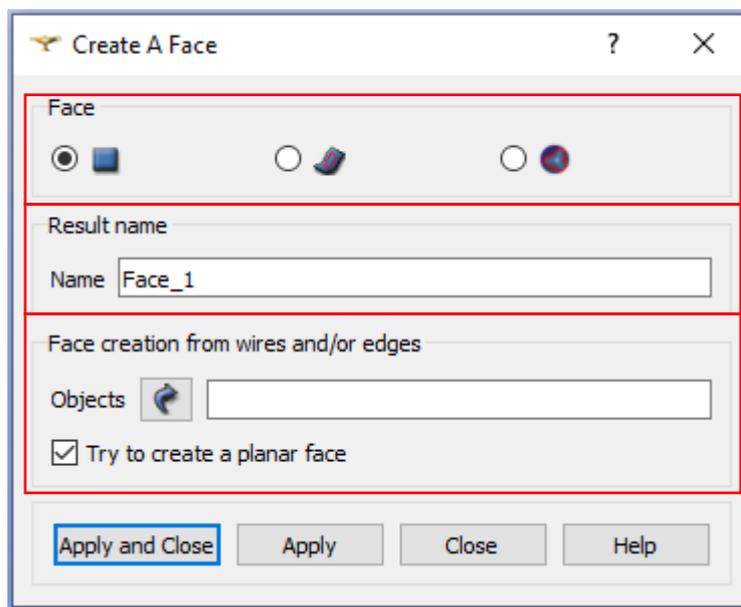


Création des surfaces :

Pour information il existe plusieurs « méthode » pour la création. Ici la création sera réalisée par les points d'extrémité du segment souhaité.



Lorsque l'on veut **sélectionner plusieurs objets** à la fois il est uniquement possible de passer par « l'arbre géométrique » en **sélectionnant les objets souhaités** et en restant appuyé sur la touche « Ctrl ».



Méthode de création

Nom de l'objet

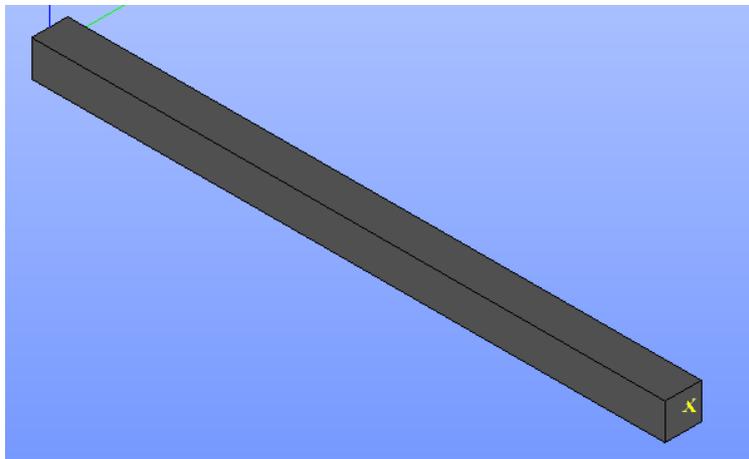
Sélection

Tous les surfaces de la poutre doivent être créées de la manière décrite ci-dessus.

Table de la création des surfaces :

Face	Segment 1	Segment 2	Segment 3	Segment 4
S1	LA	LB	LC	LD
S2	LE	LF	LG	LH
S3	LA	LJ	LE	LI
S4	LB	LK	LF	LJ
S5	LL	LC	LK	LG
S6	LI	LD	LL	LH

Illustration de l'affichage fenêtre de visualisation après la création des surfaces :



Remarque : Ici il n'y a pas de volume/solide uniquement des surfaces.

Création du solide :

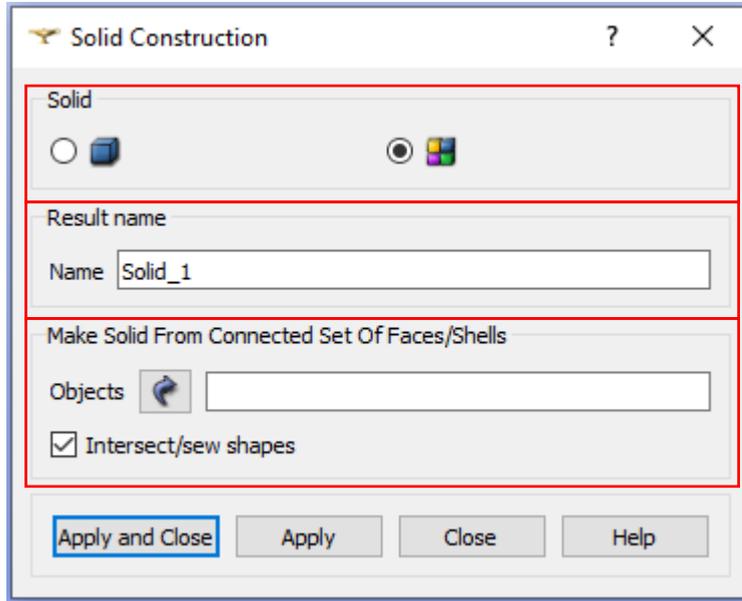
Pour information il existe plusieurs « méthode » pour la création d'objet, ici la création sera réalisée par les surfaces.



Lorsque l'on veut **sélectionner plusieurs objets** à la fois il est uniquement possible de passer par « l'arbre géométrique » en **sélectionnant les objets souhaités** et en restant appuyé sur la touche « Ctrl ».

La méthode de création à utiliser ici n'est pas l'option par défaut.

La case « *Intersect/sew shapes* » doit être cochée (Attention ce n'est pas la méthode par défaut).



Méthode de création

Nom de l'objet

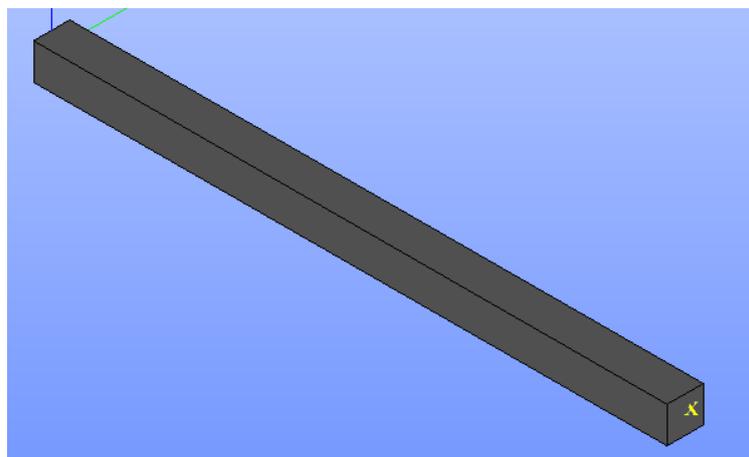
Sélection

Conseil : Donner un nom au solide pour faciliter l'exploitation de celui-ci, par exemple « **Poutre** ».

Table de la création du solide :

Solide	Face 1	Face 2	Face 3	Face 4	Face 5	Face 6
Poutre	S1	S2	S3	S4	S5	S6

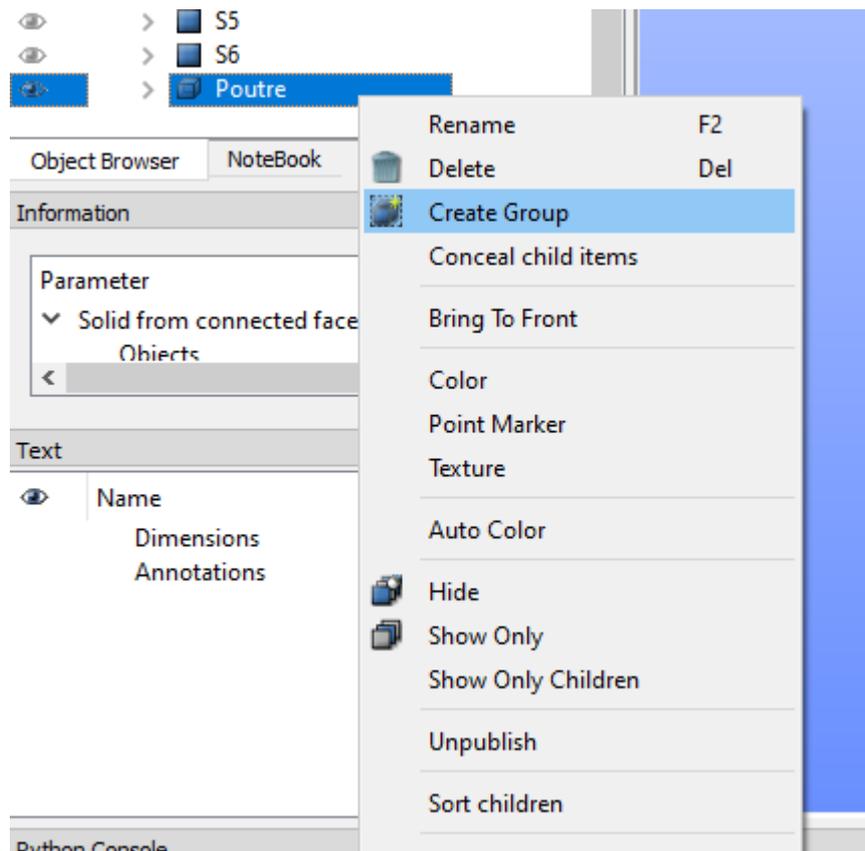
Illustration de l'affichage de la fenêtre de visualisation après la création du solide :



La géométrie est désormais créée, néanmoins afin de pouvoir réaliser le paramétrage de l'étude il est nécessaire de créer des groupes. Ces groupes permettront l'application de conditions aux limites ou de chargements.

Création du groupe de la face de fixation :

Pour créer un groupe : clic droit sur l'élément parent dont vous souhaitez créer des groupes fils, ici le solide « Poutre » :

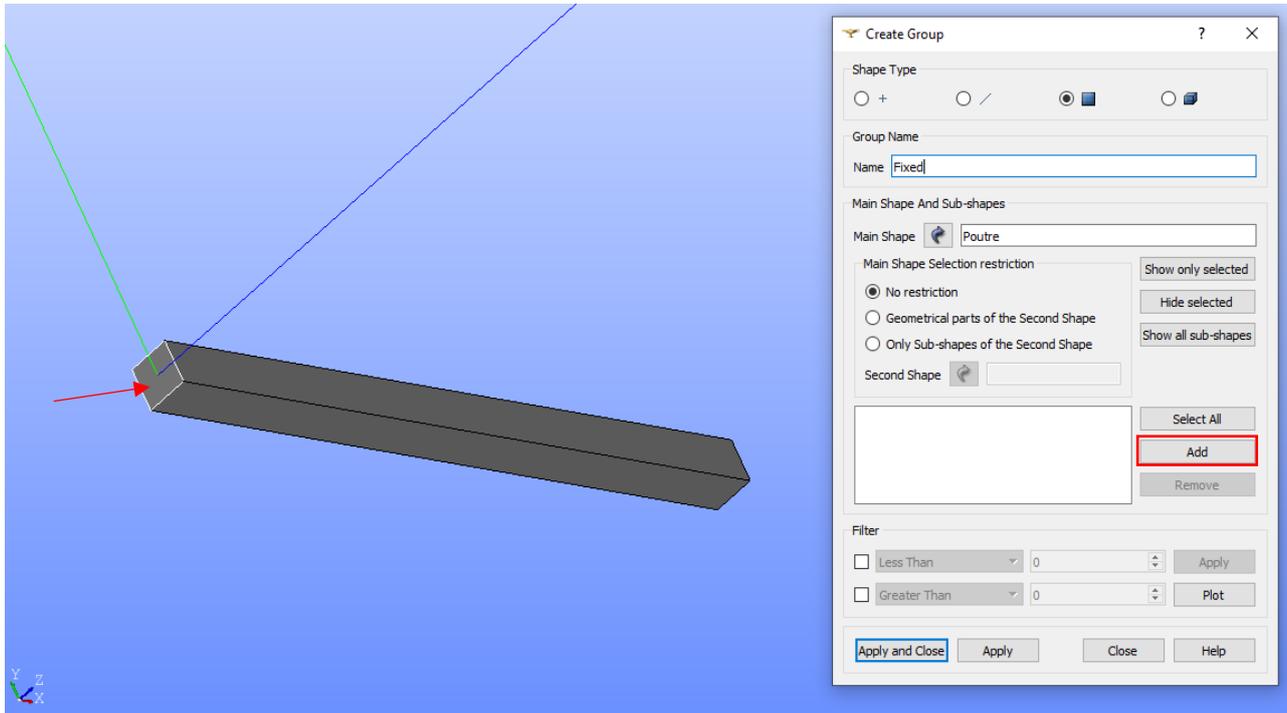


Les groupes ainsi créés appartiennent à l'objet « père ». Il est également possible de créer des groupes à un objet « fils » (sous-groupes) avec la même méthode.

Remarque : Il est indispensable de créer des objets fils appartenant à un objet père si on veut les exploiter dans même temps que celui-ci.

Sélectionner la face à fixer, dans notre cas de figure la face « S1 » puis cliquer sur « Add » afin de l'ajouter au groupe. Une fois fait, valider.

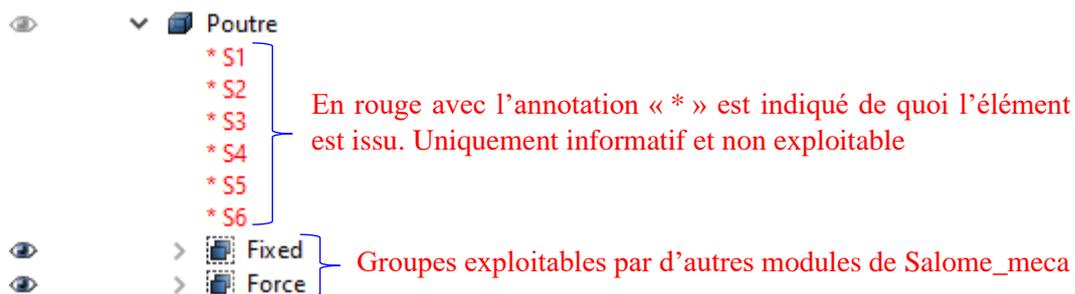
Attention : Sélection d'une **méthode par sélection** d'objet de **surface**



Création de la surface d'application de l'effort :

Avec la même méthode que précédemment il est possible de créer un groupe pour la face d'application de l'effort.

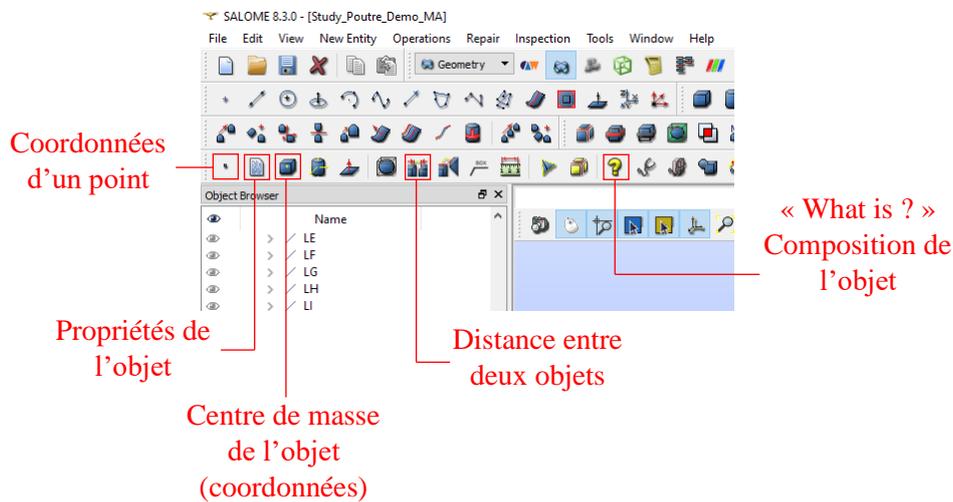
Il est à noter que les groupes ainsi créés sont rattachés au corps solide (objet « père »)



La création de la géométrie et la préparation de celle-ci pour l'analyse de la poutre est ensuite terminée.

5.2.3 Présentation des principaux outils du module géométrique

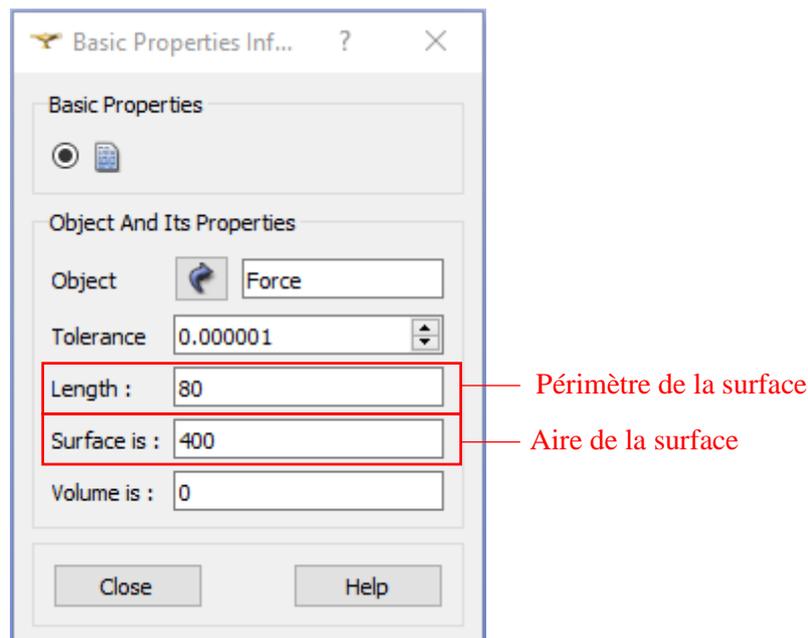
Illustration des outils principaux :



Dans ce rapport il sera présenté uniquement l'outil « **Propriétés de l'objet** » ainsi que l'outil « **centre de masse** ».

L'application d'un effort sur une face se concrétise par le paramétrage d'un effort surfacique, il est donc nécessaire pour cela de connaître l'aire de la face de l'extrémité de la poutre. Pour se faire on peut utiliser l'outil « **Propriétés de l'objet** ».

Illustration avec la surface de l'application de l'effort, ici « Force » :



Vérification **surface** : $S = a^2 = 20^2 = 400 \text{ mm}^2$

Vérification **périmètre** : $Pe = 4*a = 4*20 = 80 \text{ mm}$

Afin d'illustrer l'outil « **centre de masse** » voici un exemple sur la face d'application de l'effort. Cet outil sera **extrêmement utile et indispensable** lors de la réalisation d'un modèle complexe par un **script Python** (sans l'utilisation de l'interface graphique).

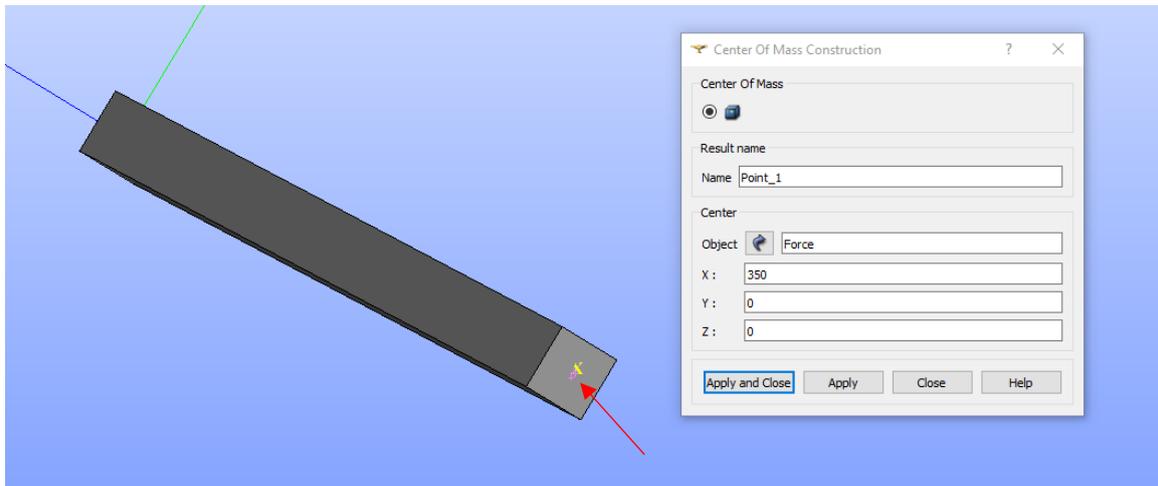
Le **centre de masse** d'une surface permettra de pouvoir réaliser un **filtre pour identifier un objet** puisqu'il est unique. En effet ces coordonnées dépendent de la position dans l'espace de celui-ci. Un filtre sur l'aire d'une surface peut être erroné puisque, deux surfaces peuvent être identiques sur un modèle, comme l'illustre parfaitement cet exemple où plusieurs surfaces ont la même aire. Dans le cas de surface concentrique avec le même centre de gravité des filtres supplémentaire devront être utilisés.

Remarque : Les coordonnées sont exprimées dans le repère général (Absolue).

Cet outil permet d'identifier (et de créer) le centre de masse d'un objet :

- Vertex (0D) : Correspond aux coordonnées de ce même point
- Edge (1D) : Correspond aux coordonnées du centre du segment
- Face (2D) : Correspond aux coordonnées du centre de la face
- Solid (3D) : Correspond aux coordonnées du centre du volume

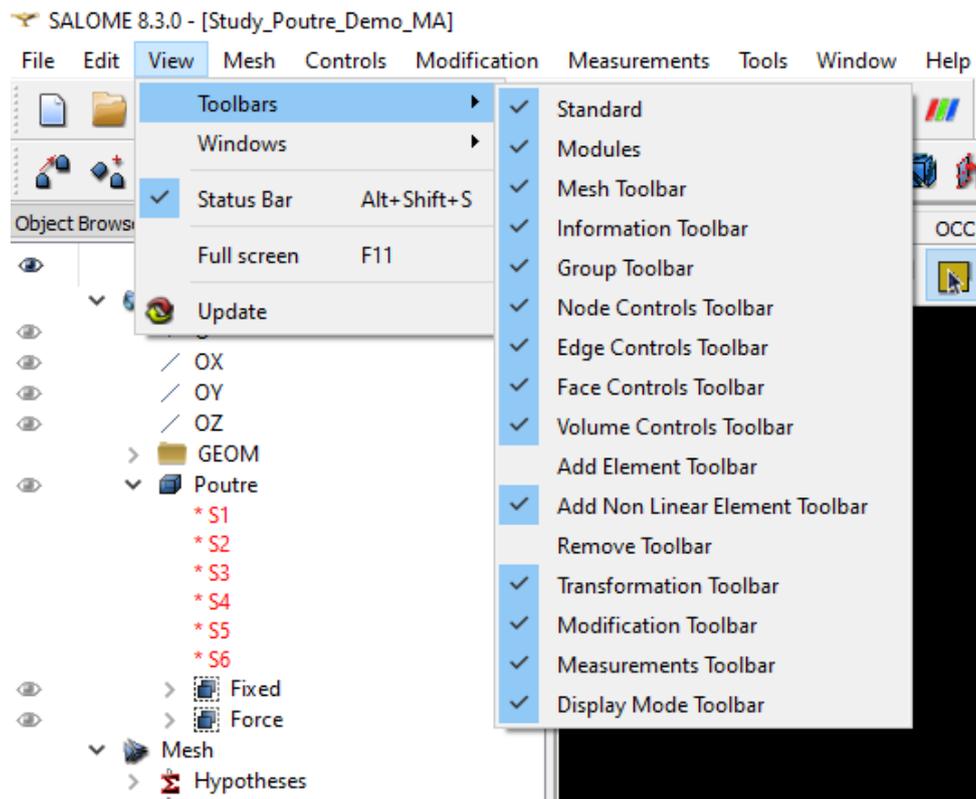
Illustration de l'outil centre de masse sur la surface d'application de l'effort :



5.3 Module Maillage Salome Meca

5.3.1 Présentation du module de Maillage

De la même manière que le module géométrique il est conseillé d'afficher un maximum de barres d'outils.

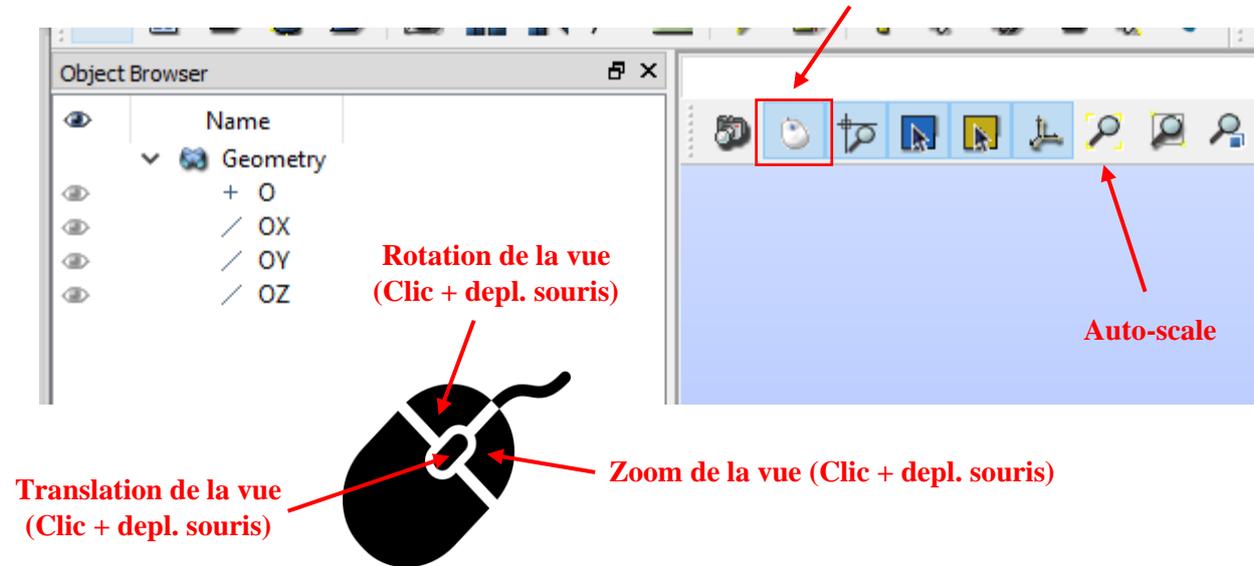


Lien documentation module maillage : <https://docs.salome-platform.org/latest/gui/SMESH/index.html>

Gestion de la vue 3D :

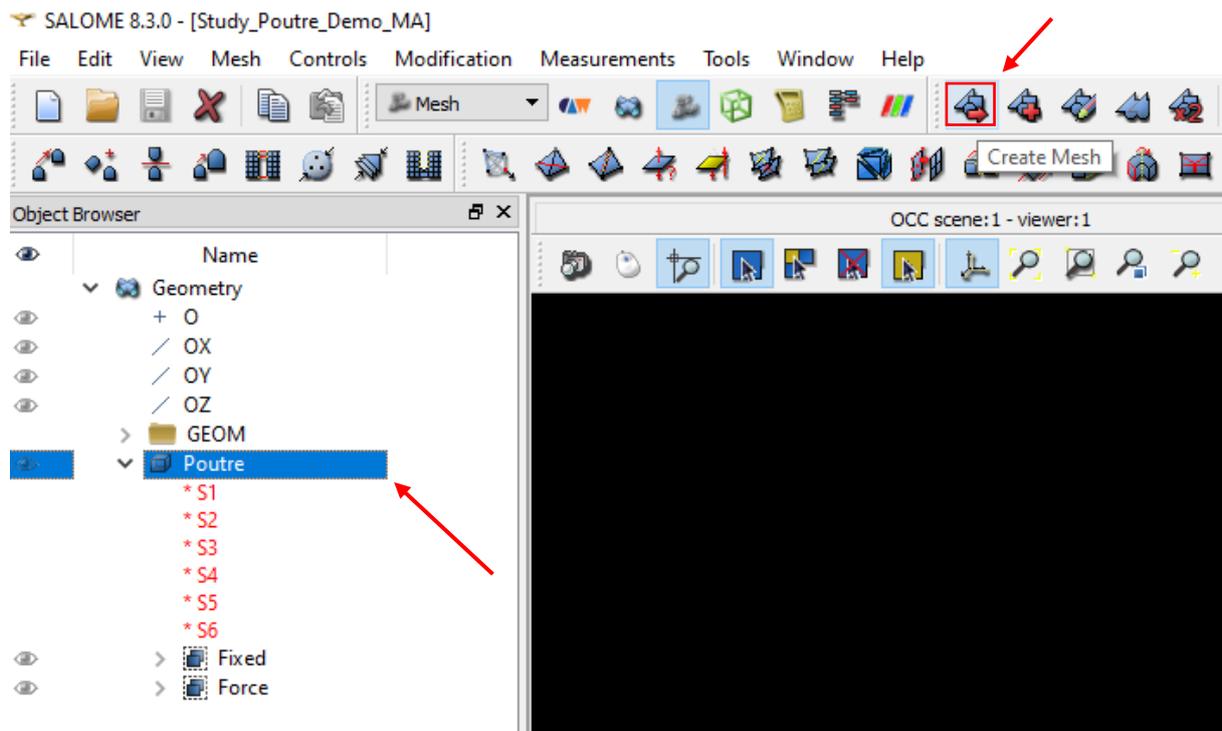
La gestion de la vue est identique au module géométrie.

Pour déplacer la vue 3D il est conseillé d'activer le bouton « Interaction style switch »

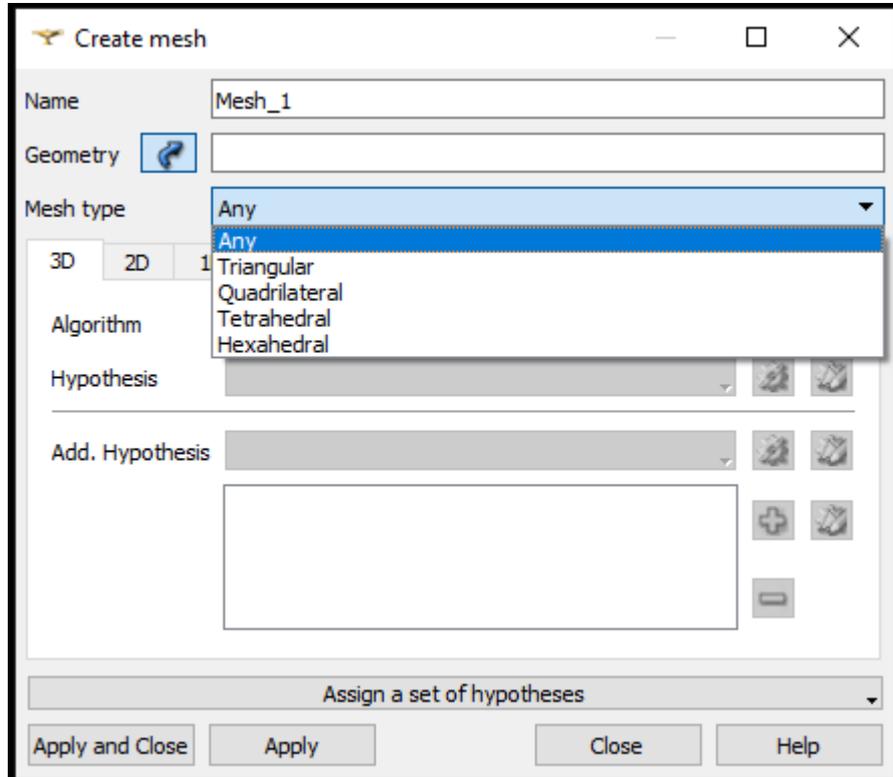


5.3.2 Création du maillage de la géométrie

Pour créer le maillage de l'élément poutre, il faut sélectionner la géométrie, puis cliquer sur « Create Mesh »



La fenêtre de paramétrisation du maillage s'ouvre :



On peut indiquer dans le « **Mesh Type** », le type de maille pour décomposer la géométrie souhaitée.

Dans notre cas on laisse le type par défaut « **Any** ».

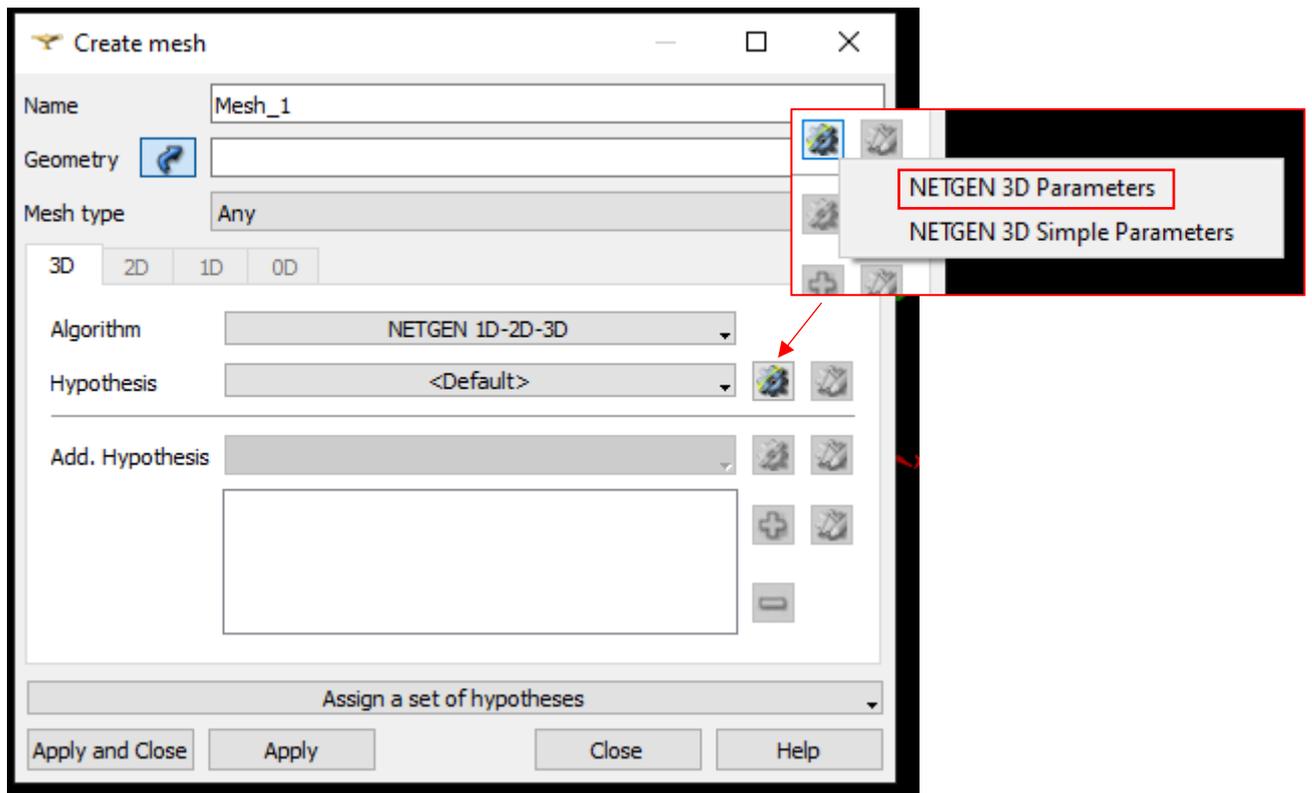
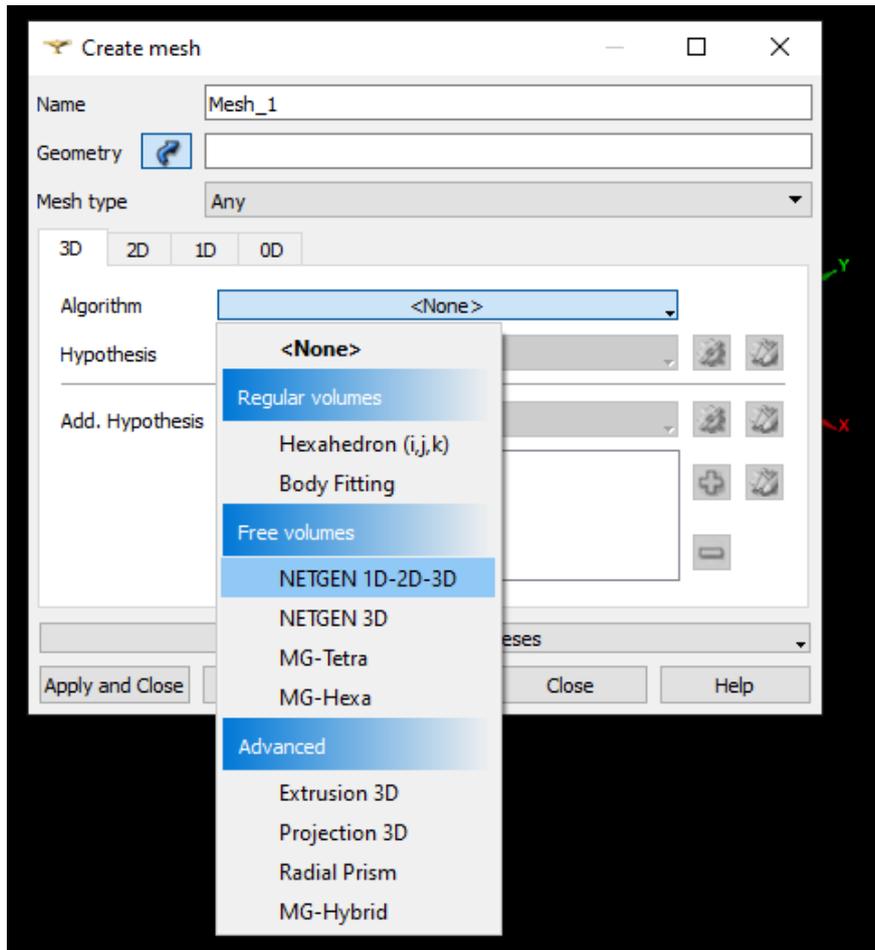
Salome_Meca propose plusieurs types d'algorithme de maillage possible.

Remarque : Il est à noter que dans Code_Aster une petite subtilité est à comprendre. Un maillage est décomposé par dimension, par exemple un maillage 3D sera composé d'éléments 3D dans son volume, d'éléments 2D sur ses faces et d'éléments 1D sur ses lignes.

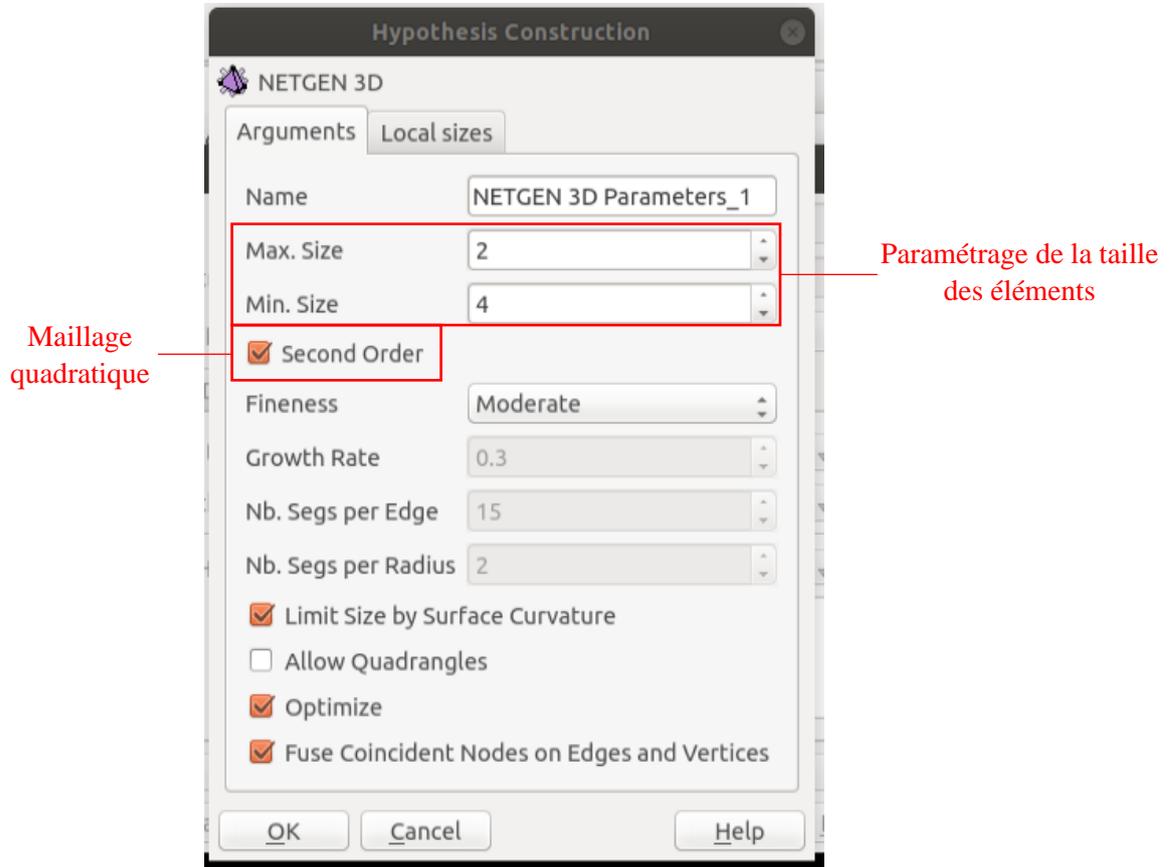
De manière général il est fortement conseillé d'utiliser l'algorithme « **NETGEN 1D-2D-3D** » qui permet de paramétrer automatiquement via des variables le maillage sur chaque dimension.

La création du paramétrage de l'hypothèse qu'utilise l'algorithme de maillage se réalise en cliquant sur l'engrenage.

(Illustration à la page suivante.)



Paramétrage des hypothèses de construction de l'algorithme de maillage :



Dans ce cas de figure pour obtenir des résultats cohérents on souhaite réaliser un **maillage quadratique** avec une taille d'éléments compris entre « **2mm** » et « **4mm** ». L'algorithme déterminera un découpage de la géométrie avec des éléments par défaut (puisque l'option « Any » a été utilisée), c'est-à-dire concrètement des éléments de forme tétraédriques, de façon à respecter les critères de dimensions d'éléments définis.

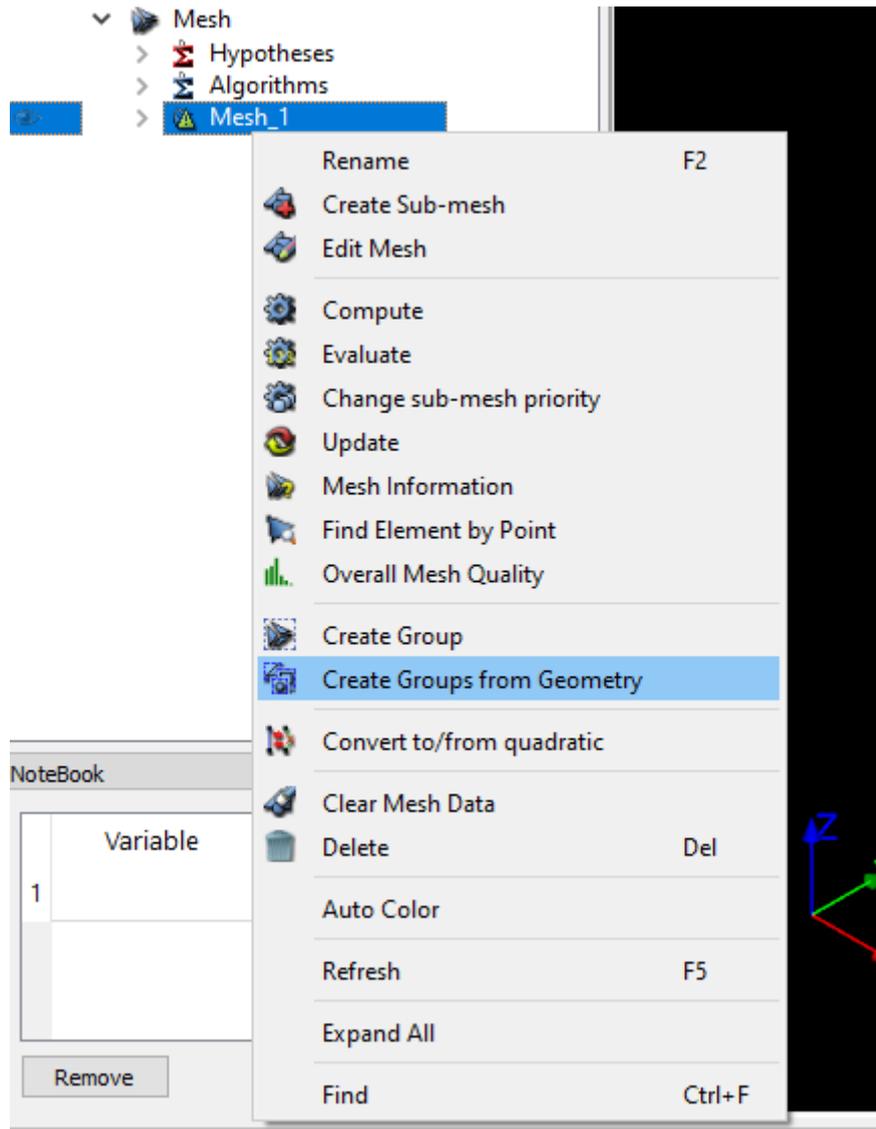
Remarque : Le maillage quadratique est un maillage de qualité supérieur avec des nœuds intermédiaires sur chaque arêtes et des éléments avec une interpolation du second ordre. Ce type d'éléments demande plus de ressources de calcul mais possède un taux de converge par nombre d'éléments plus important.

On peut noter également que dans ce cas de figure un maillage avec des éléments hexaédrique aurait été plus pertinent, mais le but recherché ici et de présenter les outils classiques du logiciel.

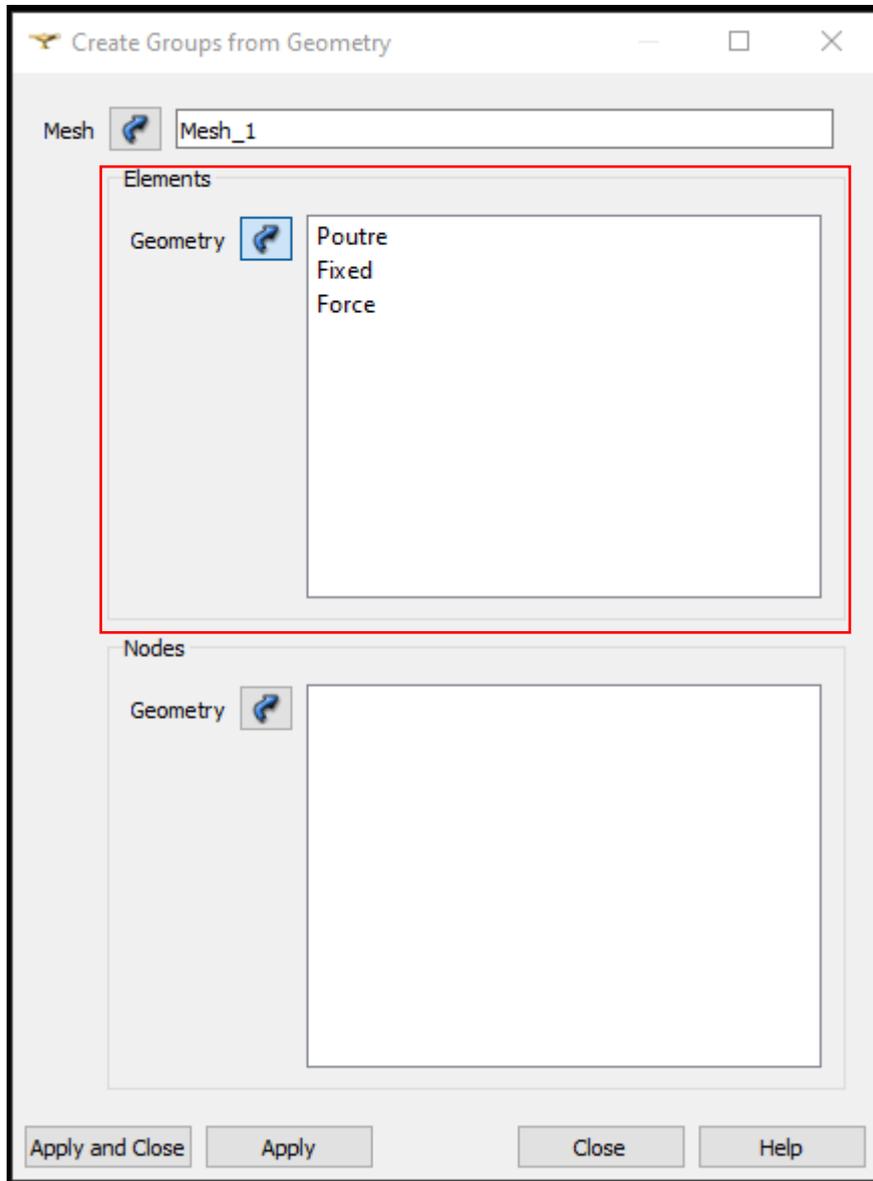
Récupération des groupes de géométrie pour le maillage :

Pour récupérer les groupes de géométrie sur le maillage il faut réaliser l'opération suivante :

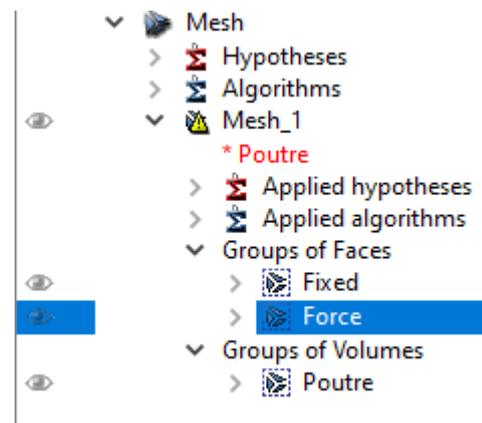
Clic droit sur le maillage, ici « Mesh_1 »



Sélectionner dans l'arbre de géométrie les objets à récupérer de la géométrie, ne pas oublier de récupérer également le solide.

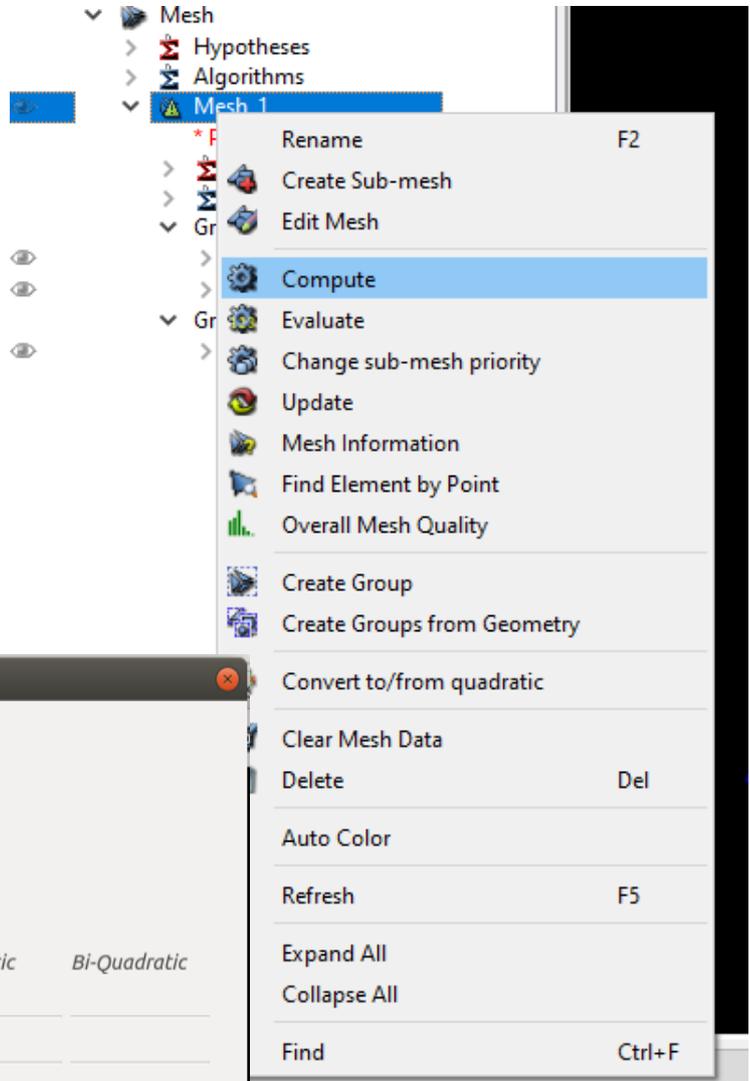
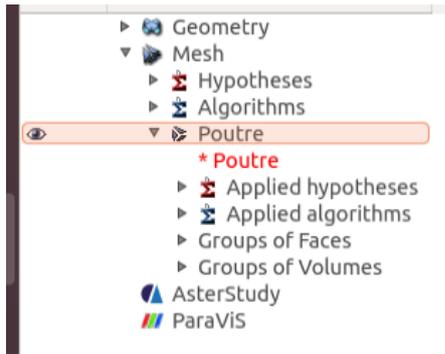


Remarque : Dans la paramétrisation du calcul, seuls les groupes appartenant au maillage pourront être sélectionnés. Une distinction entre Nœud et Élément doit être faite.



Afin de réaliser le maillage, clic droit sur le maillage à réaliser puis sélectionner « Compute » :

Il est possible de renommer le maillage en double-cliquant dessus, par exemple dans notre cas en « Poutre ».



Mesh computation succeed

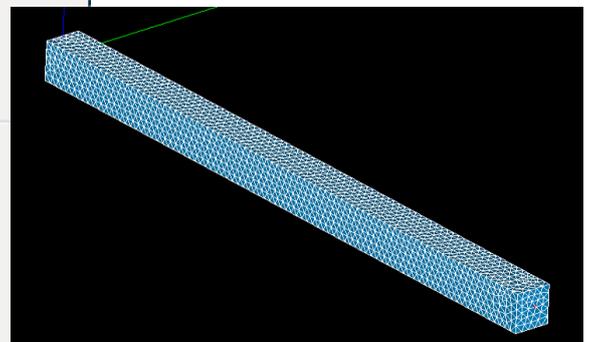
Compute mesh

Name
Poutre

Mesh Infos

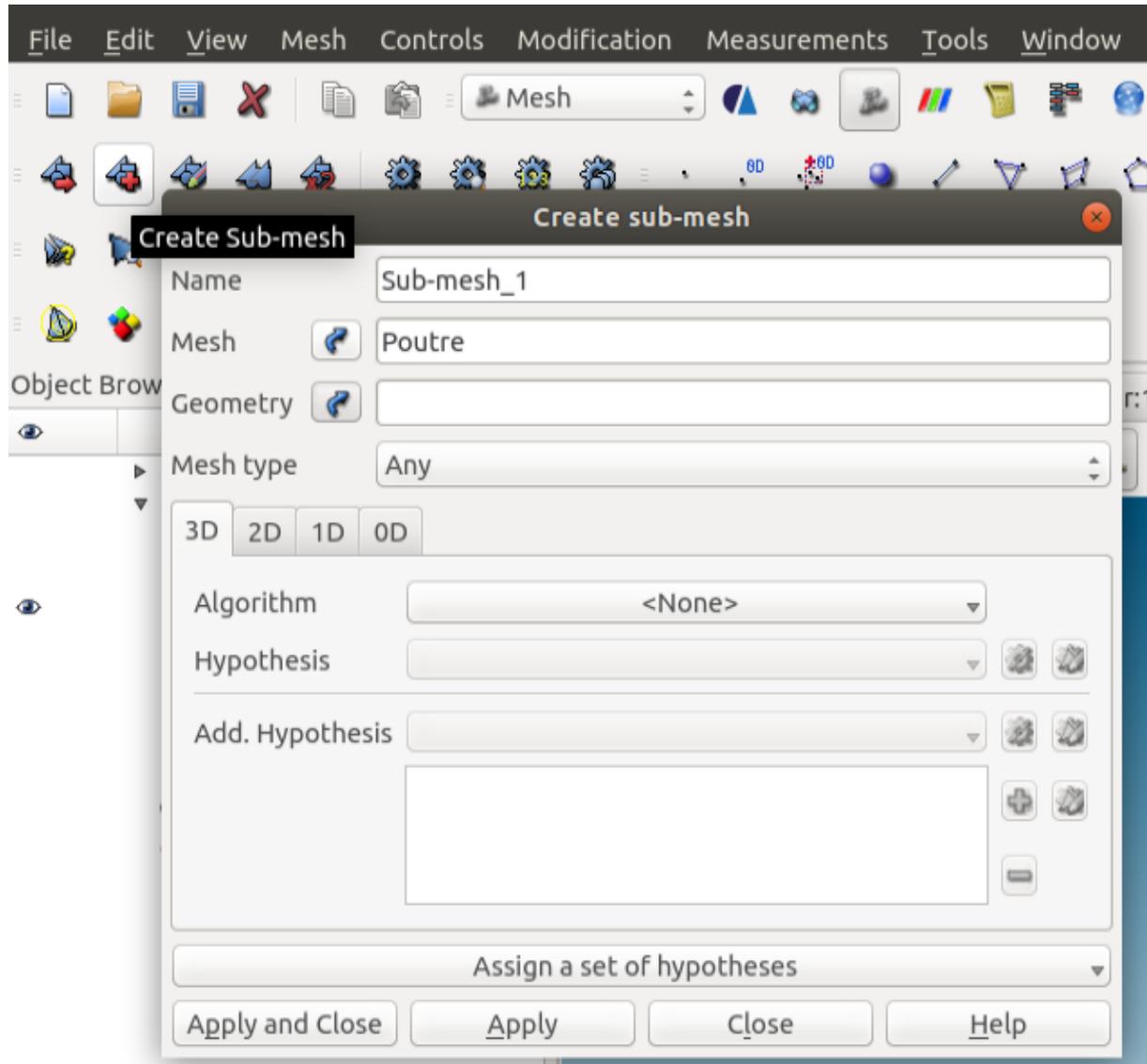
	Total	Linear	Quadratic	Bi-Quadratic
Nodes :	17748			
0D Elements :	0			
Balls :	0			
Edges :	392	0	392	
Faces :	4322	0	4322	0
Triangles :	4322	0	4322	0
Quadrangles :	0	0	0	0
Polygons :	0	0	0	
Volumes :	9894	0	9894	0
Tetrahedrons :	9894	0	9894	
Hexahedrons :	0	0	0	0
Pyramids :	0	0	0	
Prisms :	0	0	0	
Hexagonal prisms :	0			
Polyhedrons :	0			

Close Help



5.3.3 Affiner le maillage d'une géométrie

Dans le cas où l'on souhaite affiner le maillage dans une certaine zone, il faut procéder comme décrit ci-dessous :



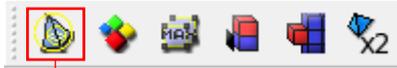
Pour affiner le maillage il faut auparavant créer une géométrie pour identifier la zone où un affinage du maillage est souhaité. Par exemple dans notre cas de figure on pourrait décider d'affiner la partie proche de la fixation en utilisant le groupe de géométrie « Fixed » et en définissant des critères de maillage plus sévères que sur le reste de la géométrie (inférieur à 2 mm).

De la même manière qu'un maillage classique, les paramètres de l'algorithme doivent être définis.

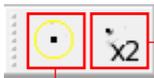
Remarque : Dans certains cas de figures il est également possible d'ajouter des arrêtes supplémentaires pour découper les surfaces pour obtenir un meilleur découpage.

5.3.4 Outils de vérification du module Maillage

Il existe plusieurs outils permettant de contrôler certains critères pour vérifier la cohérence et quantifier la qualité d'un maillage, dans ce rapport on s'intéressera uniquement à « l'aspect Ratio » :



Affiche l'aspect Ratio des éléments du maillage sélectionnés



Affiche le nombre de double nœuds (dans le cas de deux maillage superposés)

Affiche le nombre de nœuds libre (rattacher à aucune structure)

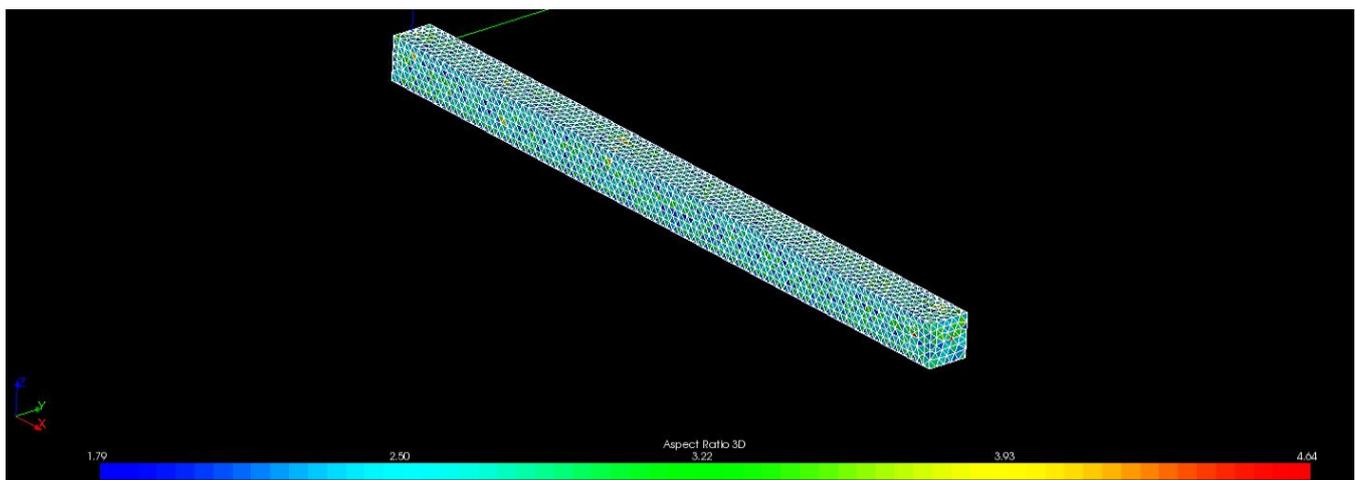
Illustration de l'outil Aspect Ratio :

Rappel de cours : L'aspect ratio d'un élément parfait est égale à 1

Ce critère mesure la difformité d'un élément. Plus la valeur numérique est importante, plus sa difformité est importante. Une difformité d'élément trop importante peut être source de résultat erroné.

Le seuil de non-validation d'un maillage dépend du phénomène qu'on souhaite étudier et du type de maillage utilisé. On retrouve de manière générale dans la littérature des seuils maximaux d'acceptabilité d'aspect ratio de « 10 » à « 20 » sur tout le modèle et un seuil maximal de « 5 » pour les zones où se porte l'étude (zone de forte contrainte).

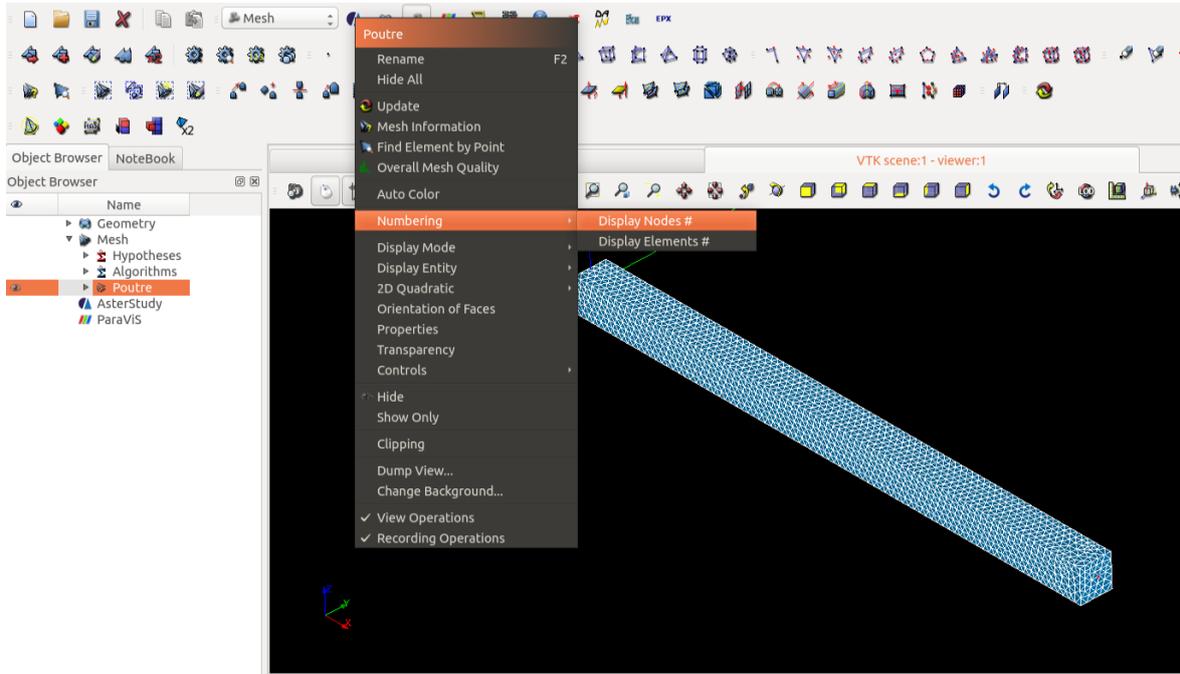
Il est **fortement conseillé** de **vérifier au minimum** le critère d'**Aspect Ratio** lors de la réalisation du maillage.



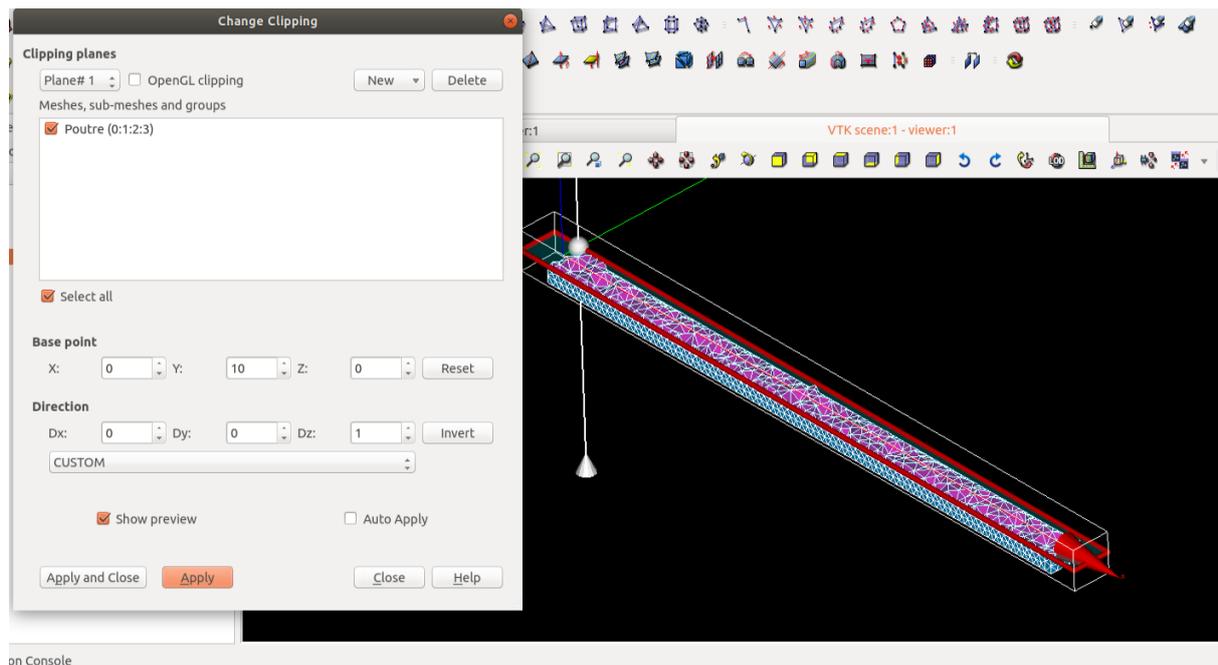
5.3.5 Affichage des numéros de nœuds et de l'orientation des faces et plan de coupe

Pour afficher le numéro des nœuds, il faut tout d'abord que le maillage soit sélectionné, clic droit dans la fenêtre de visualisations puis sélectionner « **Display Nodes #** » dans « **Numbering** ».

Pour afficher l'orientation des faces, même opération mais sélectionner « **Orientation of Faces** ».



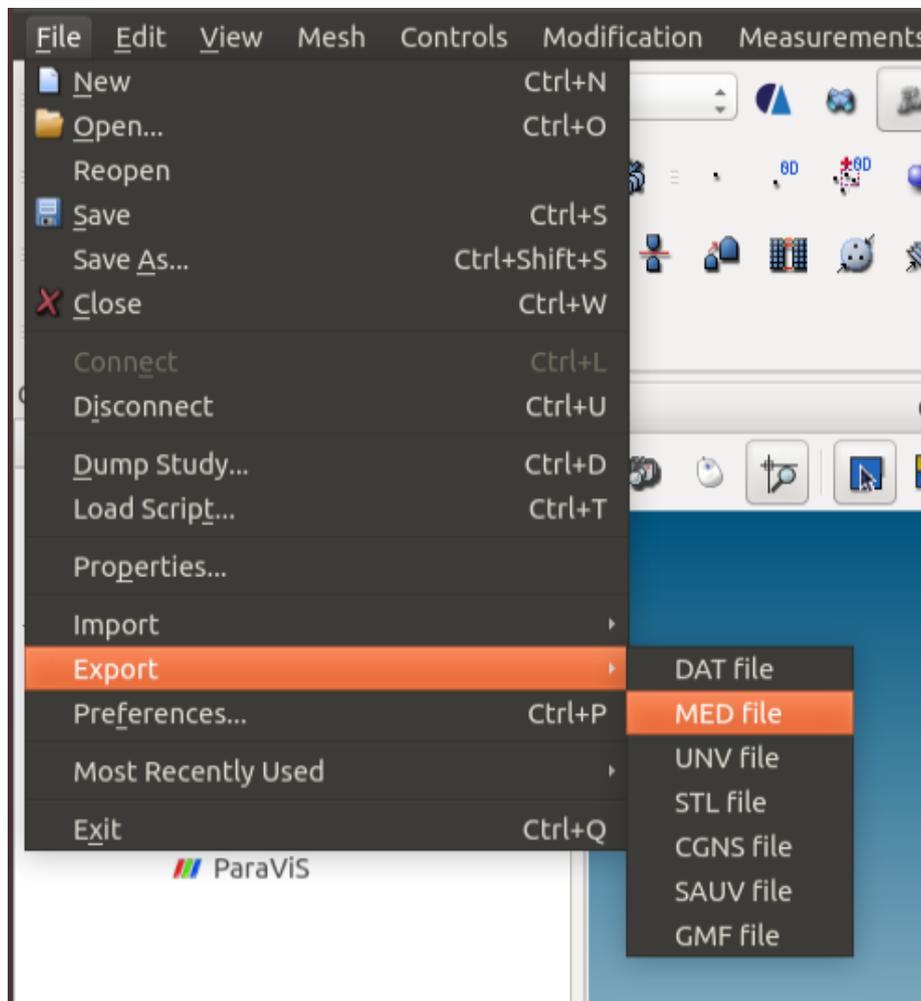
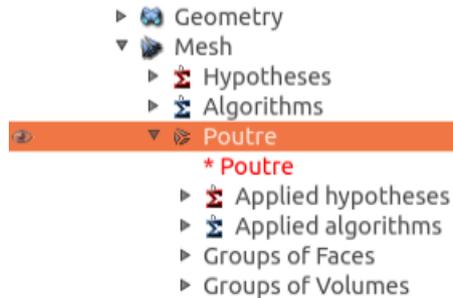
Pour réaliser un plan de coupe du maillage utiliser l'option « Clip », cet outil permet d'inspecter le maillage au sein du volume. (Cliquer sur « New » et sélectionner « Absolute », « Delete » pour le supprimer).



5.3.6 Exportation d'un maillage

La procédure pour exporter un maillage est décrite ci-dessous :

Sélectionner le maillage à exporter dans l'arborescence, puis sélectionner « Export » dans « File ».

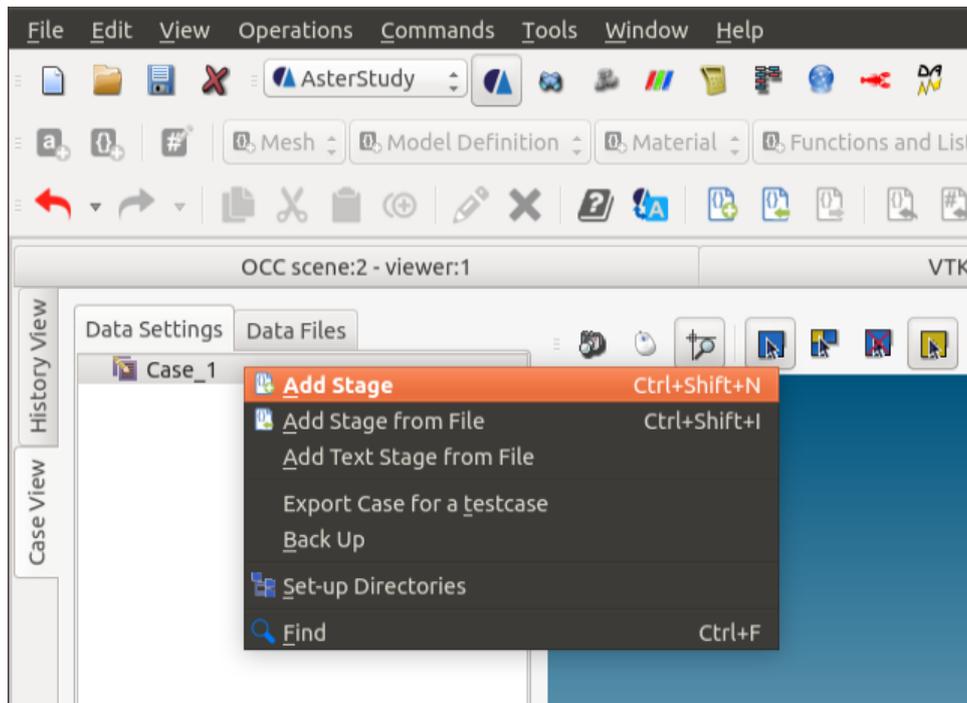


Le maillage doit être sauvegardé sous l'extension « .med ».

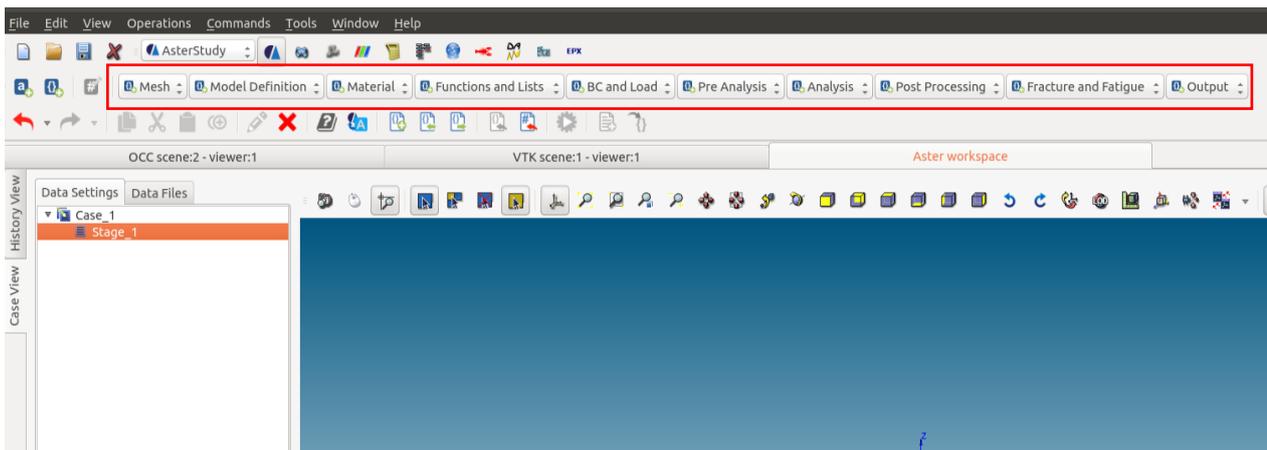
5.4 Paramétrisation de l'analyse – Module AsterStudy

Le module « AsterStudy » permet de paramétrer l'analyse par éléments finis.

La première étape consiste à la création d'un « Stage » qui correspond à un cas d'étude, pour ce faire clic droit sur « Case_1 » :



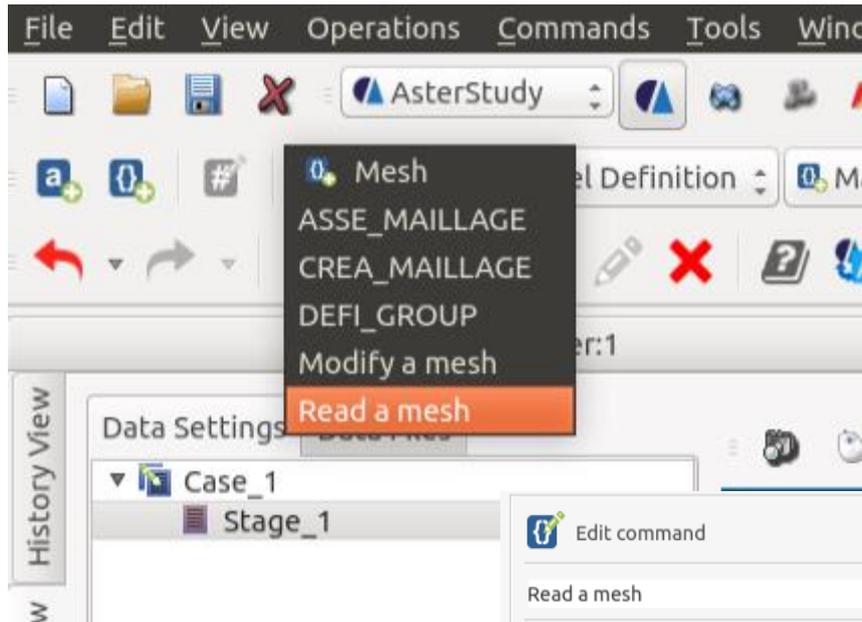
La paramétrisation du calcul sera réalisé par un assemblage de plusieurs fonctions. Les fonctions se situent par défaut en haut dans le menu des barres d'outils :



5.4.1 Lecture du maillage :

La première étape consiste à indiquer au logiciel sur quel maillage on veut réaliser l'étude, cette étape est indispensable et sera toujours la première étape à réaliser.

Cette étape se situe dans le module « Mesh » (maillage en anglais), en cliquant sur « Mesh » un menu déroulant s'ouvre, il faut ensuite sélectionner « Read a mesh », cette opération ouvre la fenêtre d'édition de commande :



Cliquer sur le menu déroulant de « Mesh file location » et sélectionner le maillage, ici « Poutre »



Cliquer sur « OK » pour valider.

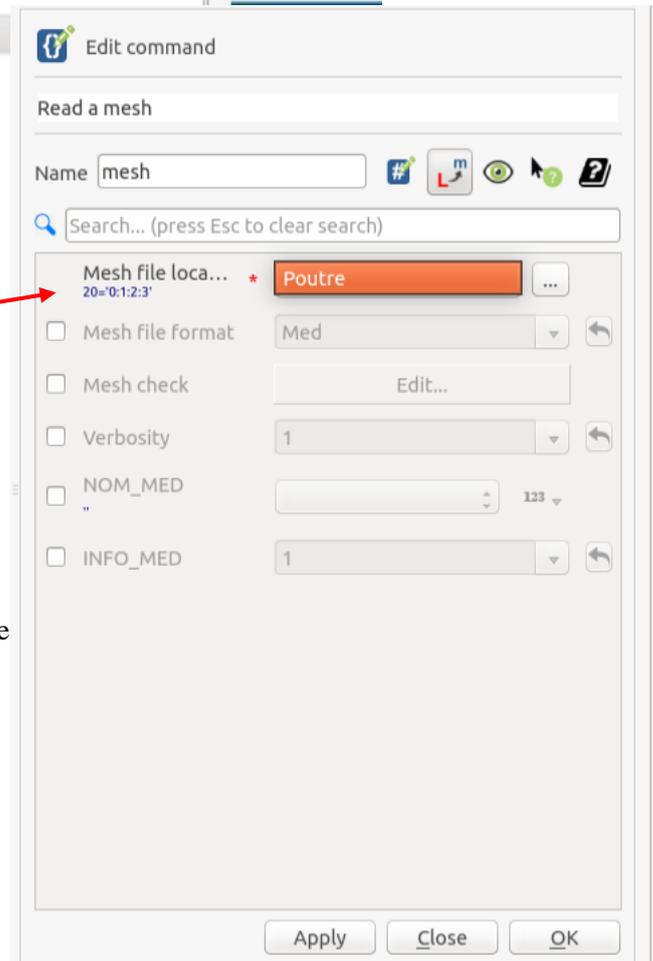
Dans « AsterStudy » il y a deux manières de valider :

- Apply : Valide l'opération effectuée,
- Ok : Valide et ferme la fenêtre.

Remarque :

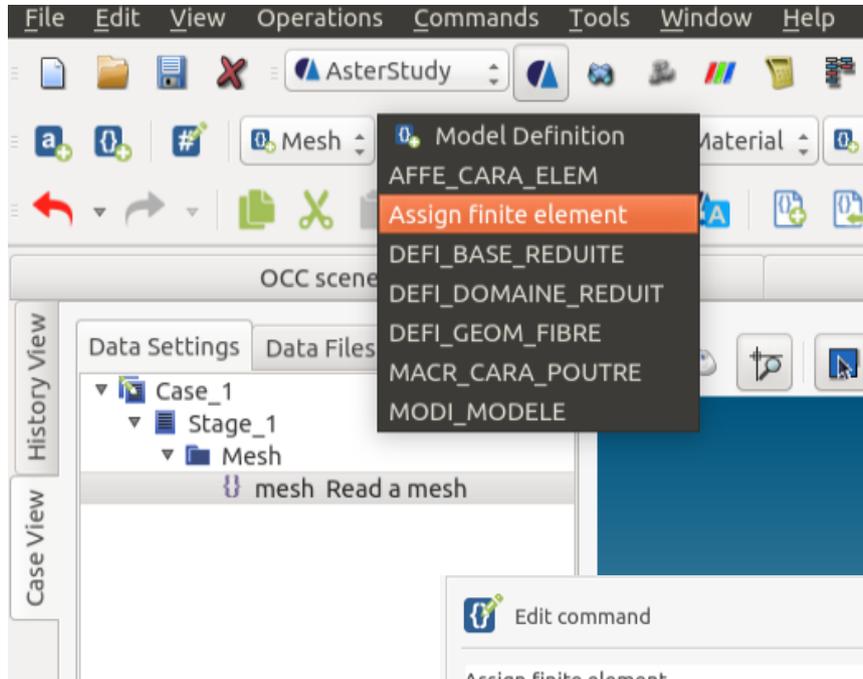
Dans l'onglet « Information » la fonction est retranscrite en langage Python. Ici l'« unité 20 » correspond à un numéro d'identification qui est attribué à l'entrée, ici le maillage.

```
Information
mesh = LIRE_MALLAGE(
  UNITE=20
)
```

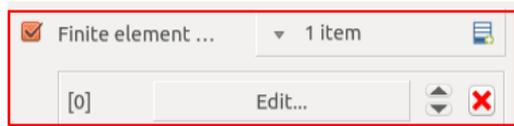


5.4.2 Affectation du type de maillage

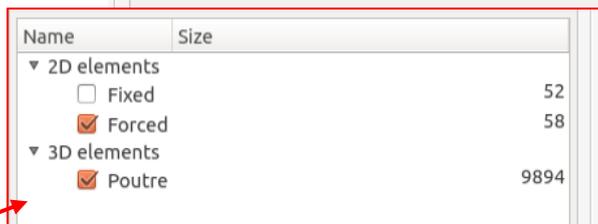
La deuxième étape consiste à la déclaration du type de maillage de la géométrie. Pour ce faire selon le même principe cliquer sur le module « Model Definition » puis sélectionner « Assign finite element » :



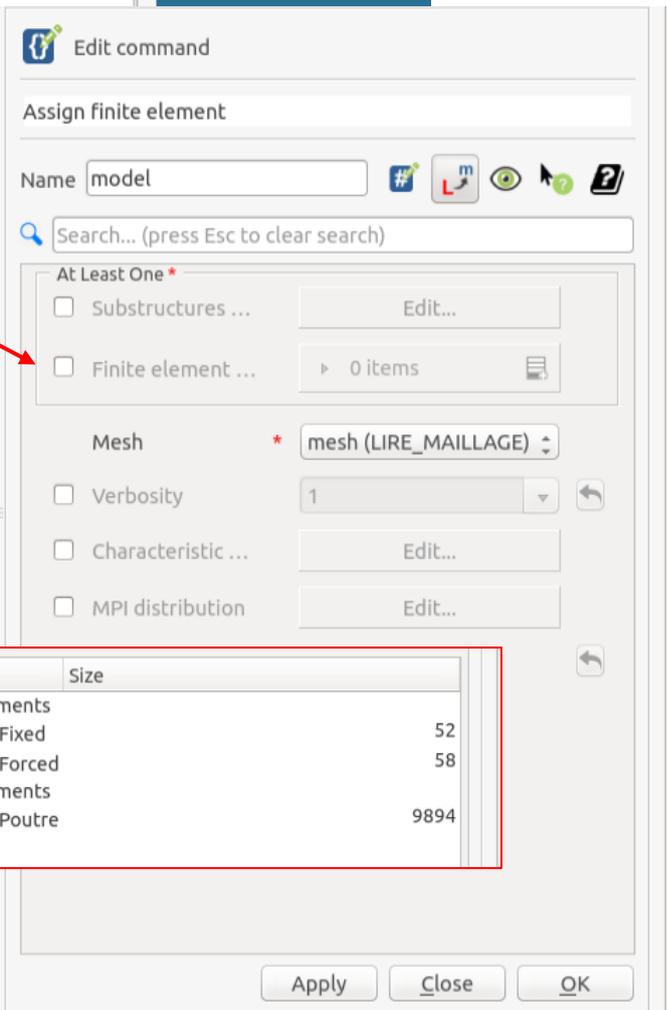
Cocher la case « Finite elment... » puis cliquer sur « Edit... ».



Sélectionner « Mechanic » dans le menu déroulant de « Phenomenom » et sélectionner « 3D » dans « Modelisation »

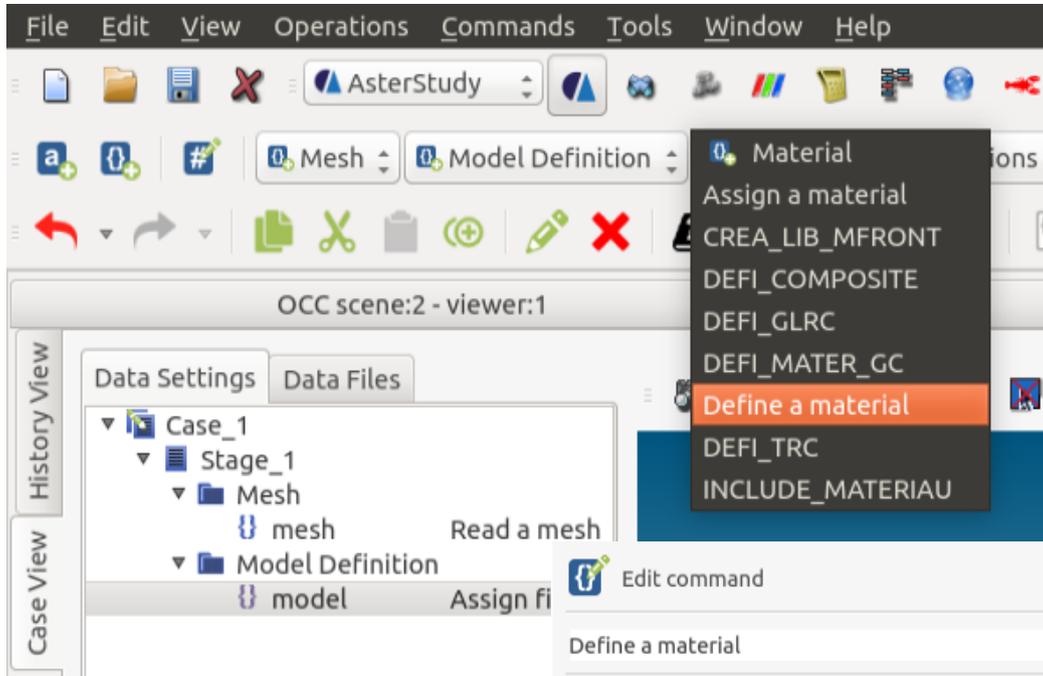


Cliquer sur « Edit... » dans « Group of Element » Sélectionner « Force » & « Poutre » et valider.



5.4.3 Définition du matériau

Pour définir le matériau cliquer sur le module « Material » et sélectionner « Define a material » :

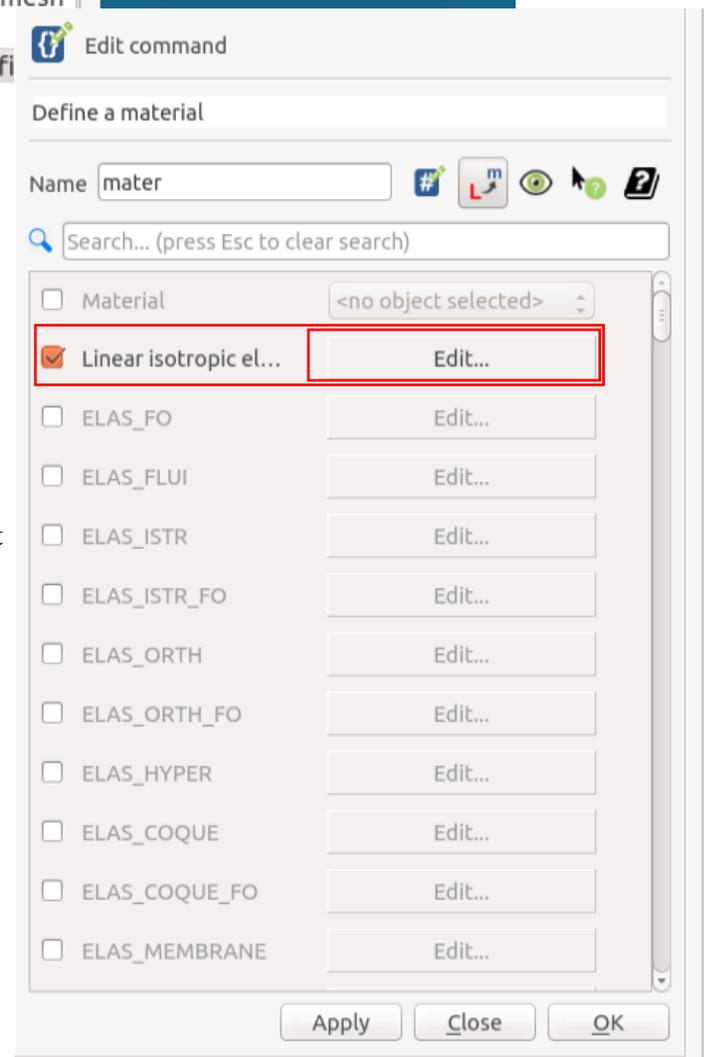


Sélectionner « Linear isotropic elasticity » puis cliquer ensuite sur « Edit... » :

Indiquer ensuite le module d'Young et le coefficient de Poisson suivant :

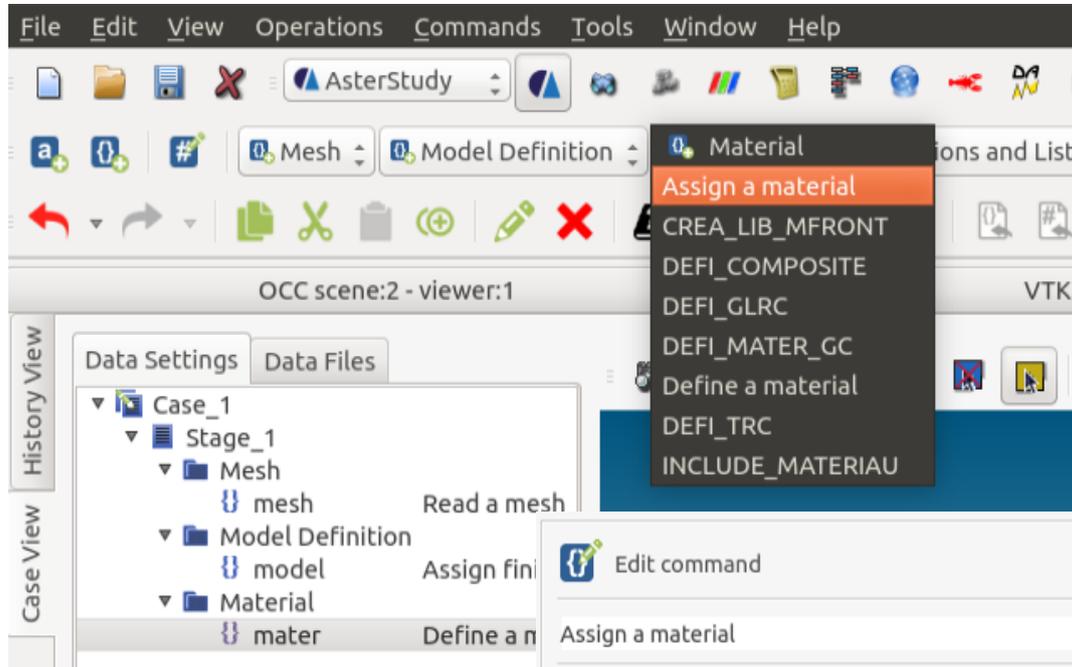


Valider en cliquant sur « OK ».



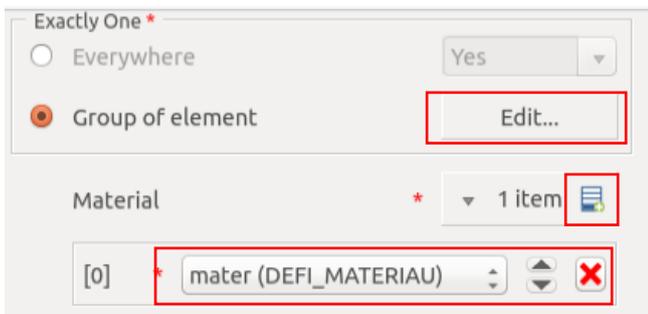
5.4.4 Affectation du matériau au maillage

De la même manière que précédemment sélectionner « Assign a material » du module « Material » :

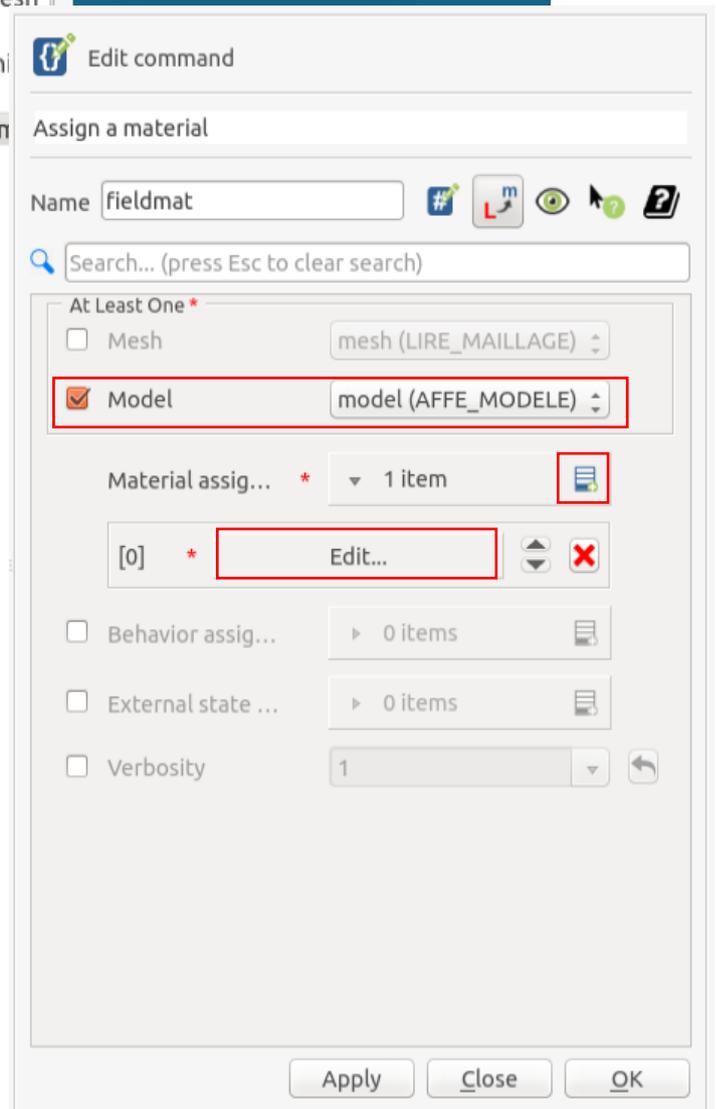


Cocher la case « Model » puis cliquer sur l'icône pour créer un nouvel item dans « Materail assignement », cliquer ensuite sur « Edit ... » :

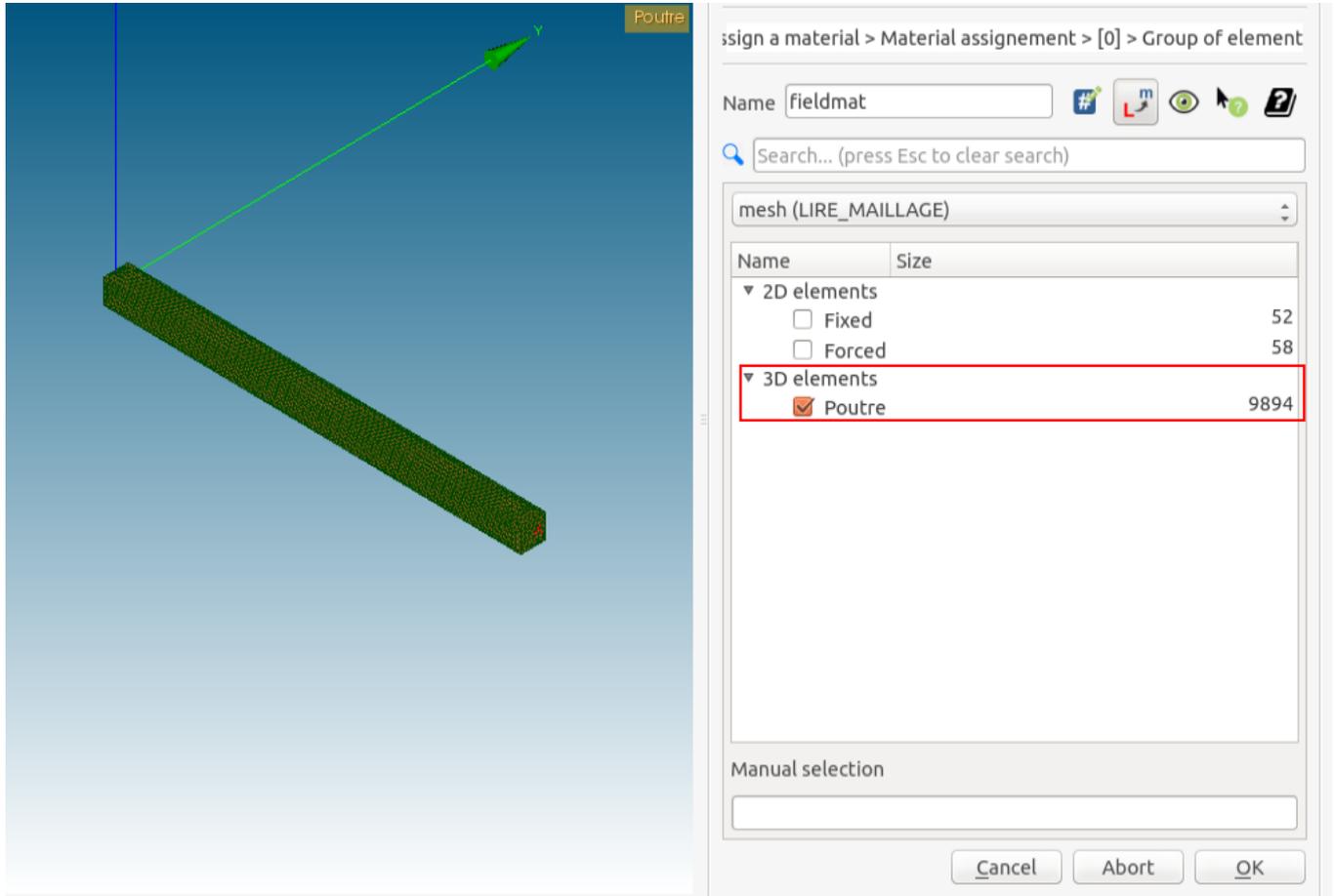
Dans la nouvelle fenêtre ajouter un nouvel item dans « Material » et sélectionner le matériau créer précédemment :



Cliquer ensuite sur « Edit... ».



Sélectionner ensuite le groupe de maillage 3d « Poutre » :

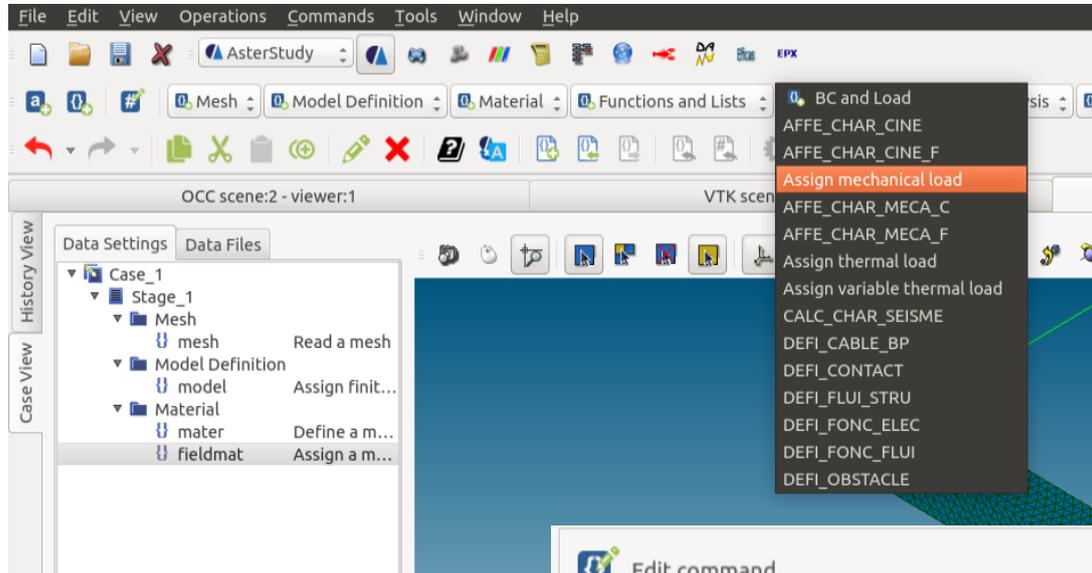


Remarque : Lors de sa sélection d'un groupe, celui sélectionné par l'utilisateur s'affiche en orange sur la vue 3D.

Finaliser en validant en cliquant sur « OK ».

5.4.5 Assignement des chargements et des conditions aux limites

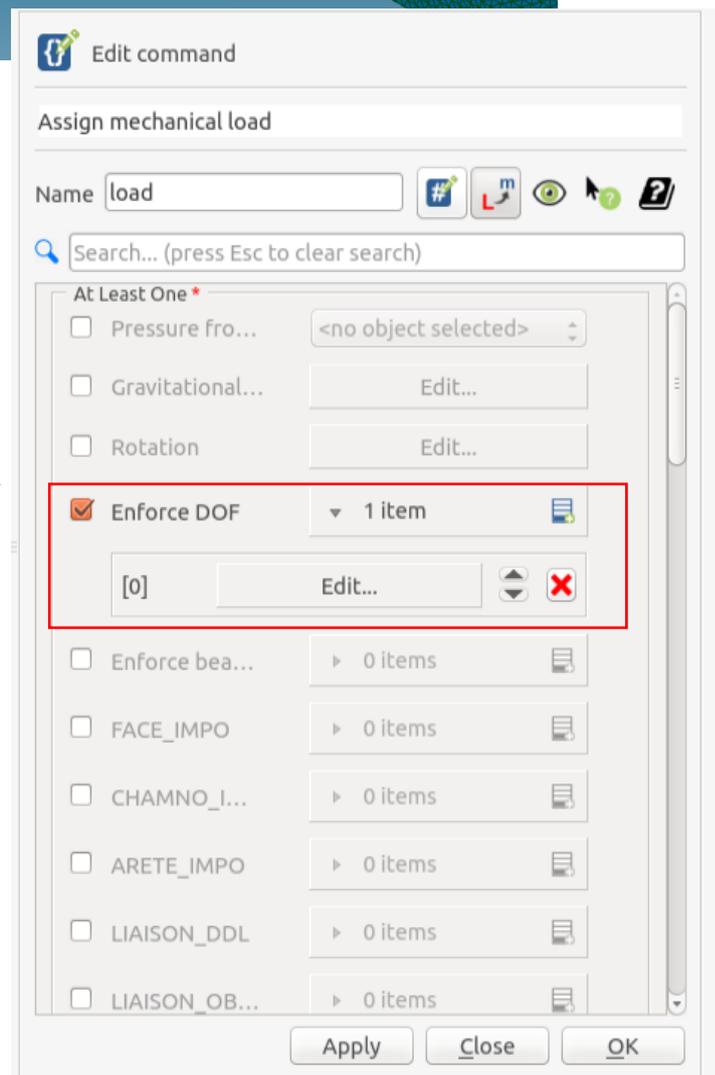
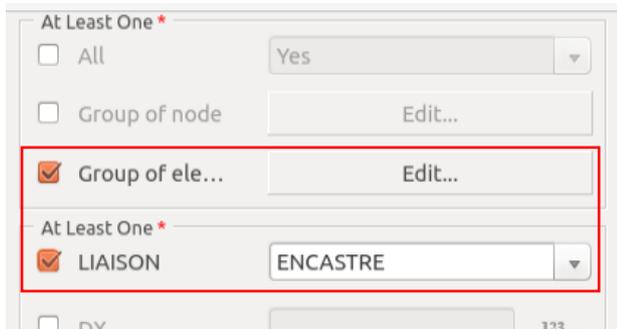
Sélectionner « Assign mechanical load » dans le menu « BC and Load » :



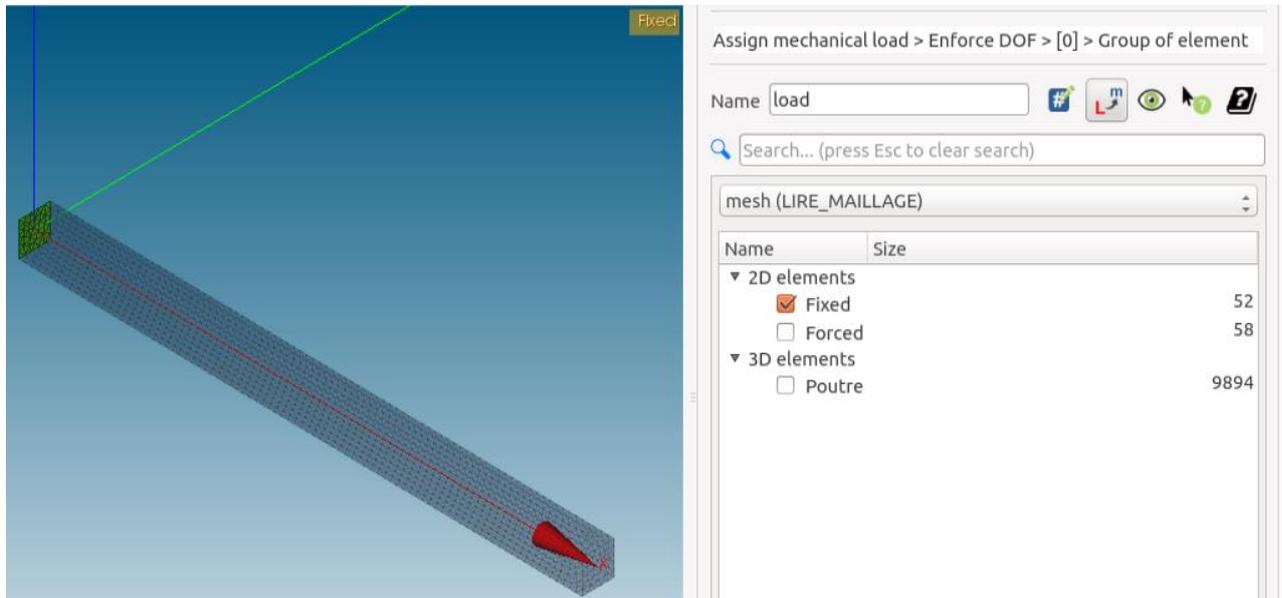
Création de la fixation de la poutre :

Pour se faire, cocher l'option « Enforce DOF » et créer un nouvel item et cliquer sur « Edit... »

Cocher la case « Liaison Encastrée » et la case « Groupe of éléments » puis cliquer sur « Edit... »

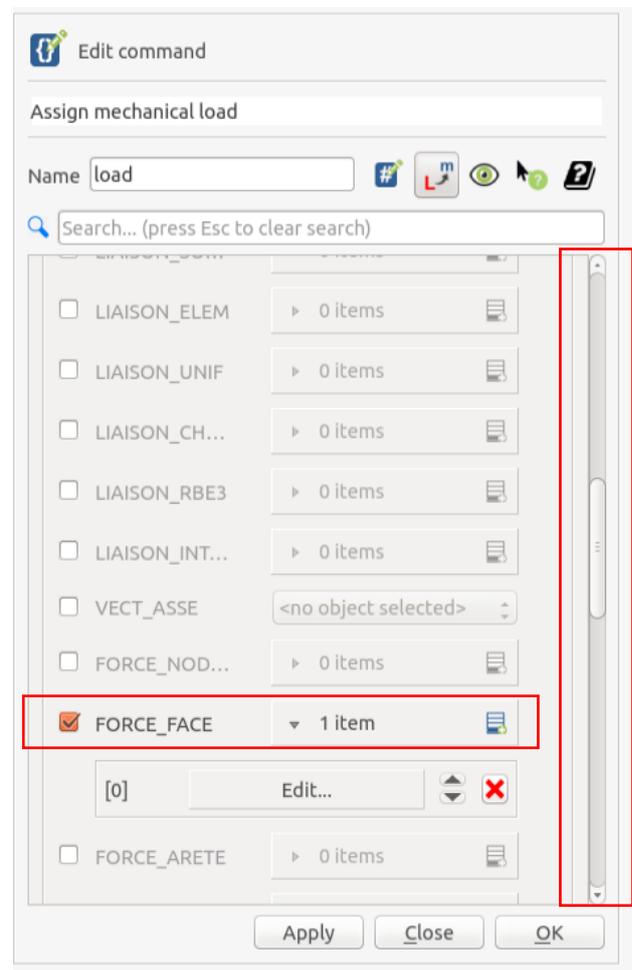


Sélectionner la face « Fixed » et valider en appuyant sur « OK » jusqu'à revenir au menu principal de la fonction « Load » (rester dans cette fonction) :



Création du chargement :

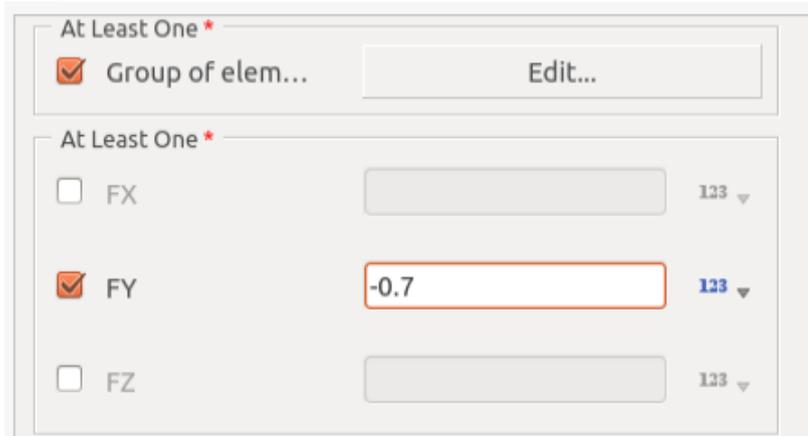
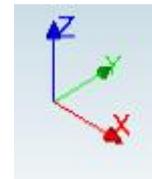
Descendre l'ascenseur jusqu'à trouver la fonction « Force_Face », ajouter un nouvel item et cliquer sur « Edit... » :



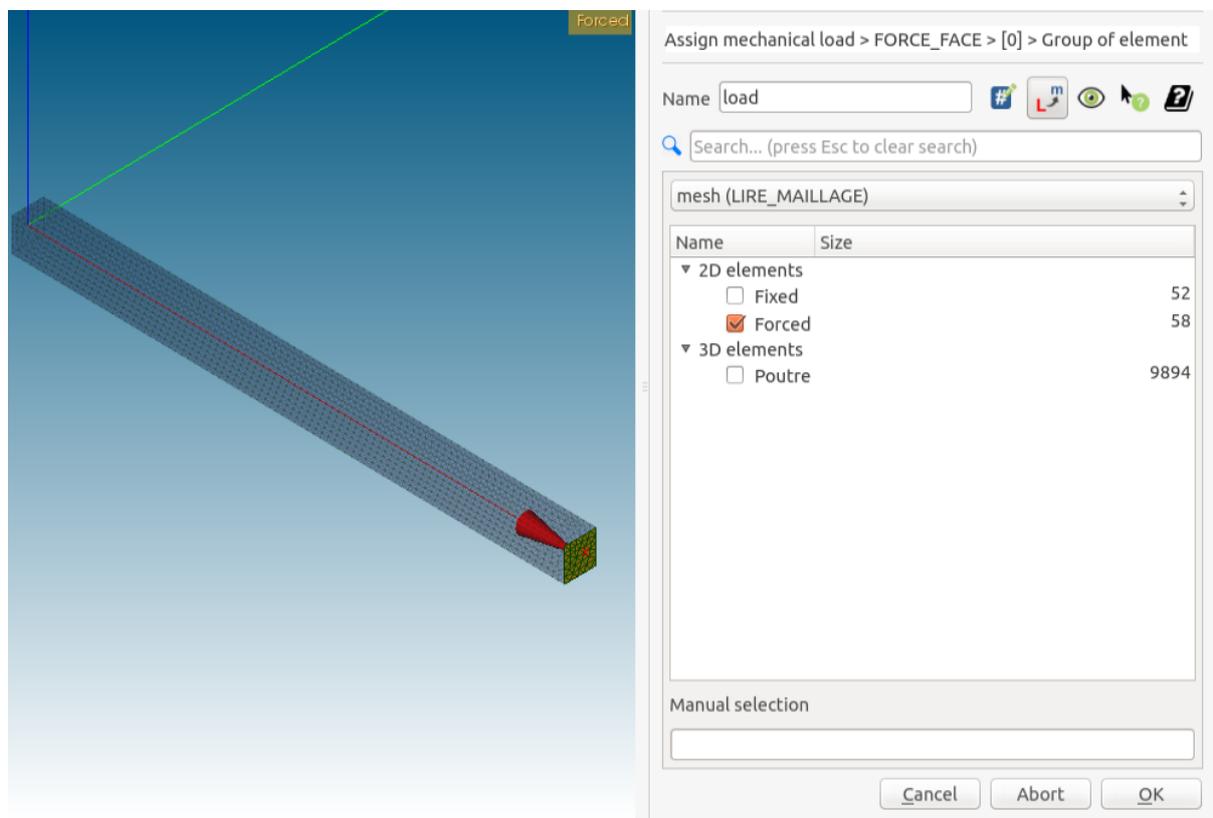
Comme évoqué précédemment la fonction « Force_Face » est un effort réparti sur une surface.

$$F_s = \frac{F}{S} = \frac{280}{400} = 0.7$$

Le signe négatif provient de l'orientation du repère, l'effort est orienté vers le bas.

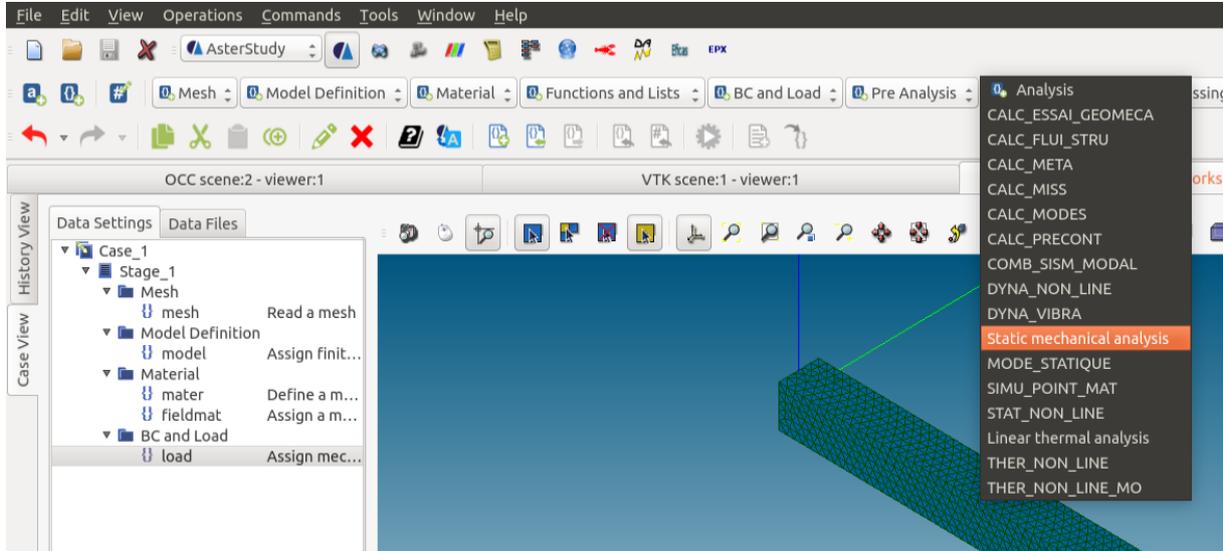
Cliquer sur « Edit... » dans « Group of éléments » et sélectionner la face « Forced » :



Finaliser en appuyant sur « Ok » jusqu'à fermeture de la fenêtre.

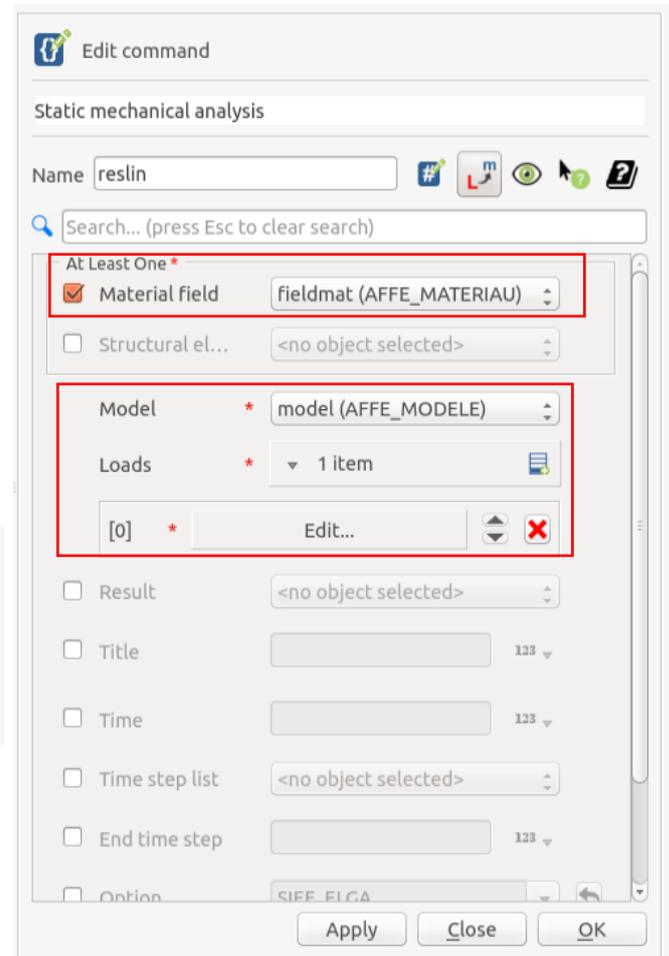
5.4.6 Définition du type d'analyse

Sélectionner « Static mechanical analysis » dans le module « Analysis » :



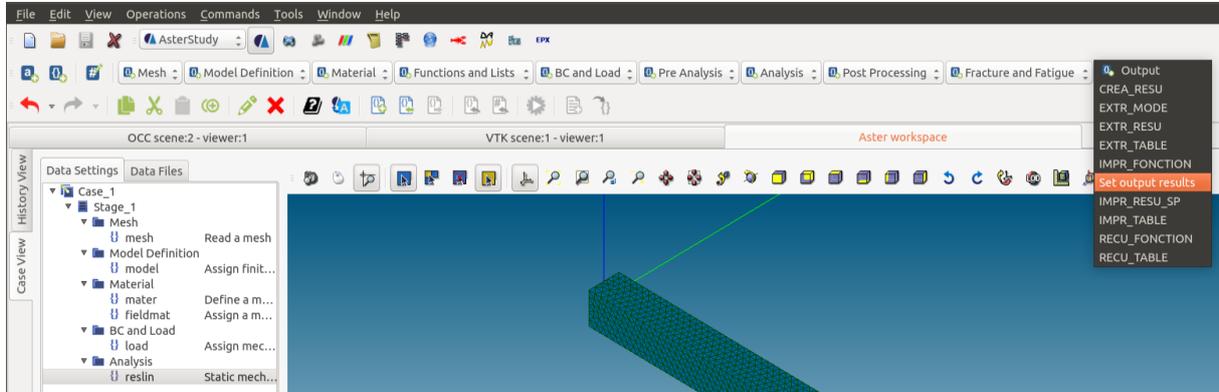
Cocher la case « Material field » et ajouter un item dans « Load », puis cliquer sur « Edit... » :

Vérifier que « load (AFFE_CHAR_Meca) » est bien sélectionné et valider en appuyant sur « OK » :



5.4.7 Déclaration des types de résultats souhaité

Sélectionner « Set output results » dans « Output » :



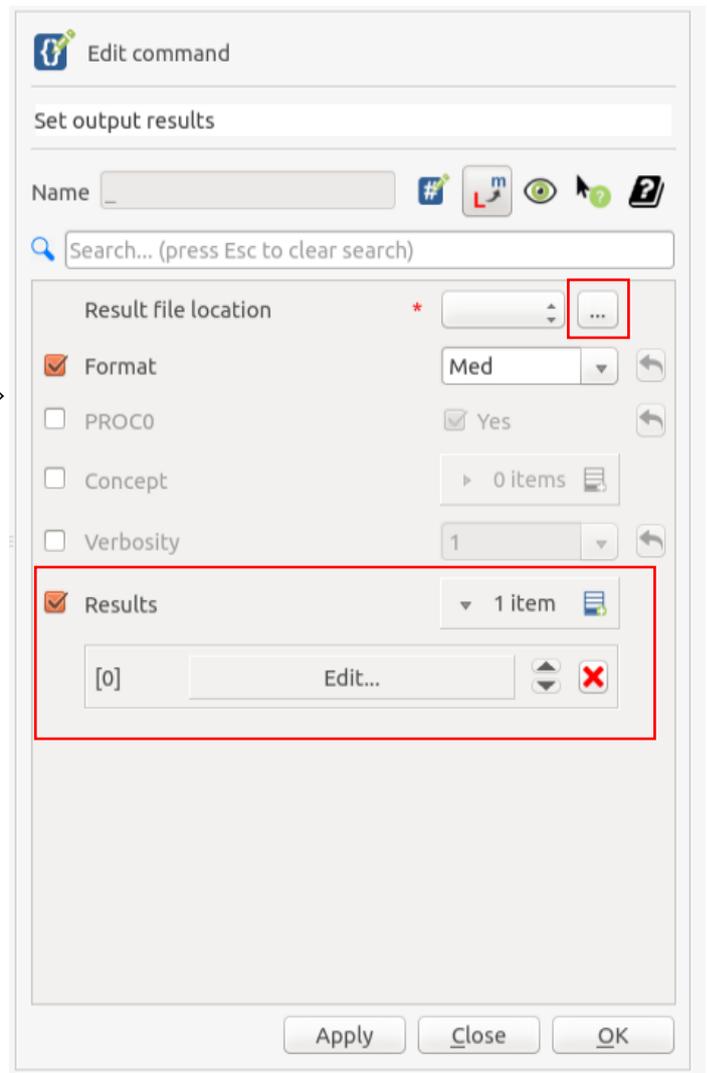
Cliquer sur « ... » de « Result file location » pour Définir un répertoire ou sauvegarder les résultats.

Vérifier que le format est de type « MED ».

Attention extension « .rmed » pour le résultats.

Ajouter un nouvel item dans « Results » puis cliquer sur « Edits... ».

Sélectionner ensuite « Reslin MECA_STATIQUE » puis valider en appuyant sur « OK » :



La paramétrisation du calcul est enfin terminée.

Enregistrer l'étude Salome.

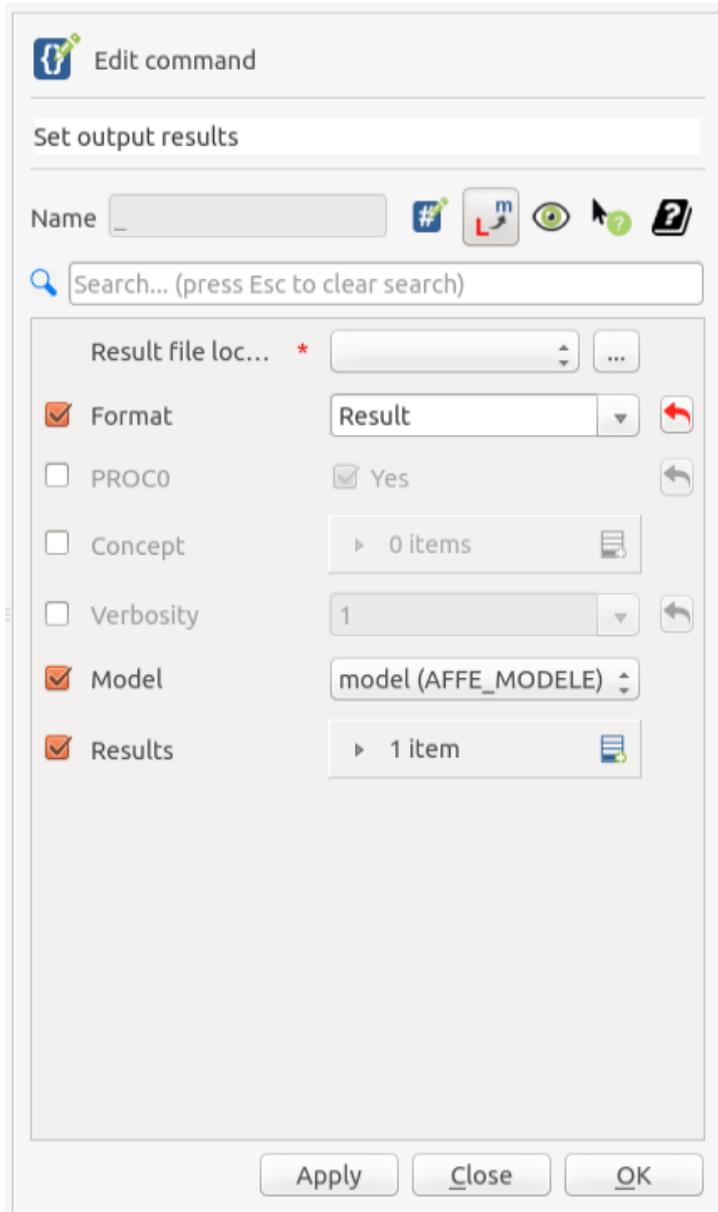
5.4.8 Extraire des résultats en format Texte

Il est possible avec Salome_Meca d'extraire des résultats sous forme de table de valeur en format texte.

Remarque : Ce format texte pourra être ouvert ultérieurement avec un tableur (avec la bonne déclaration de séparateur d'unité) dans le cas où des résultats plus précis doivent être exploiter.

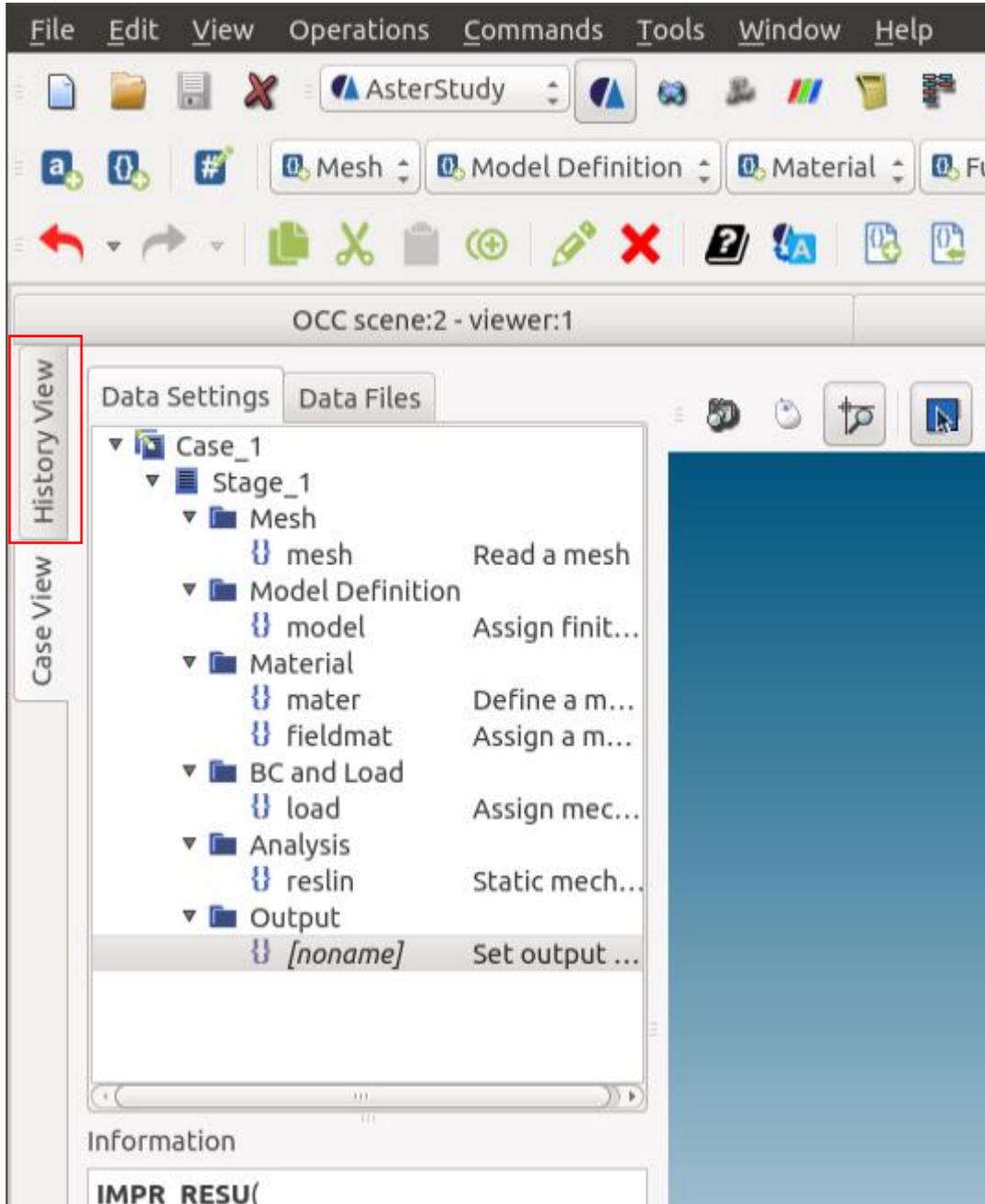
Un autre impression de résultat doit être créée, avec cette fois-ci un format « **Result** » et le fichier doit être sauvegardé sous l'extension « **.resu** », il est à noter que certains résultats doivent être calculé avec la fonction « CALC_CHAMP » du module « Post Processing », comme la réaction nodal par exemple. La création antérieure de groupes de nœuds peut aussi être nécessaire.

Le détail de la méthodologie ne sera pas présenté dans ce rapport.



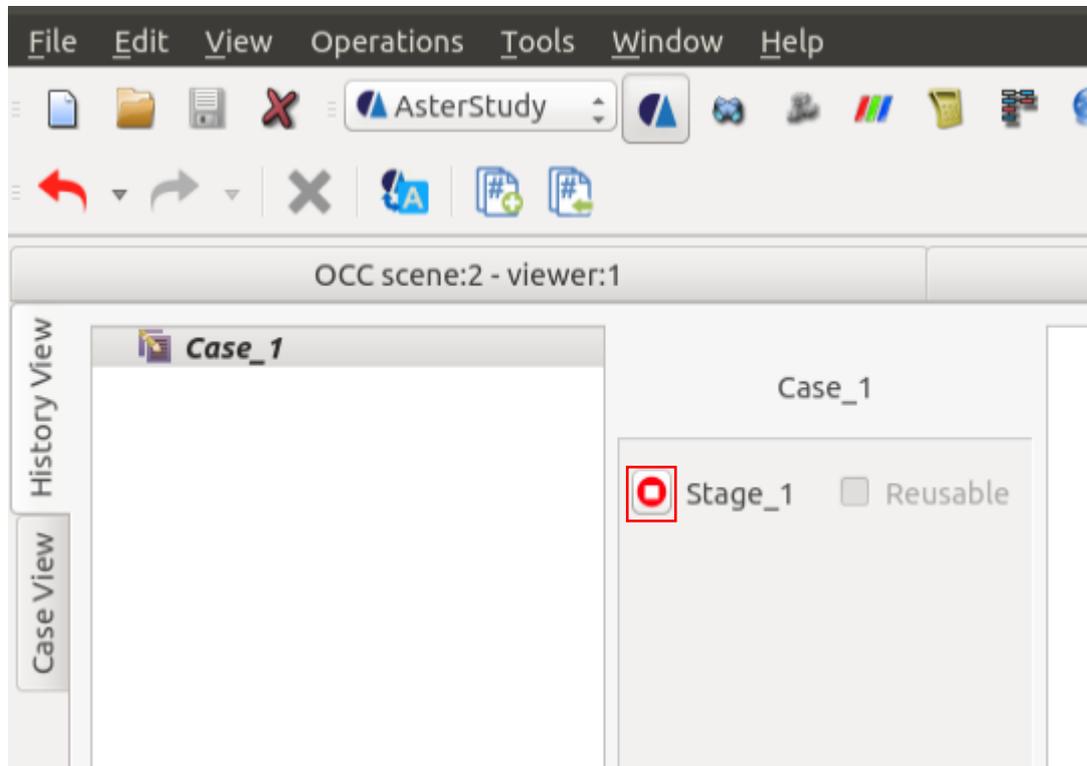
5.4.9 Lancement du calcul

Pour lancer le calcul cliquer sur l'onglet « History View » :

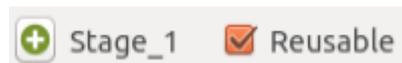


Remarque : Pour lancer un calcul il est nécessaire que l'étude Salome soit sauvegardée.

Dans cet onglet, cliquer ensuite sur le bouton rouge de « Stage_1 » :



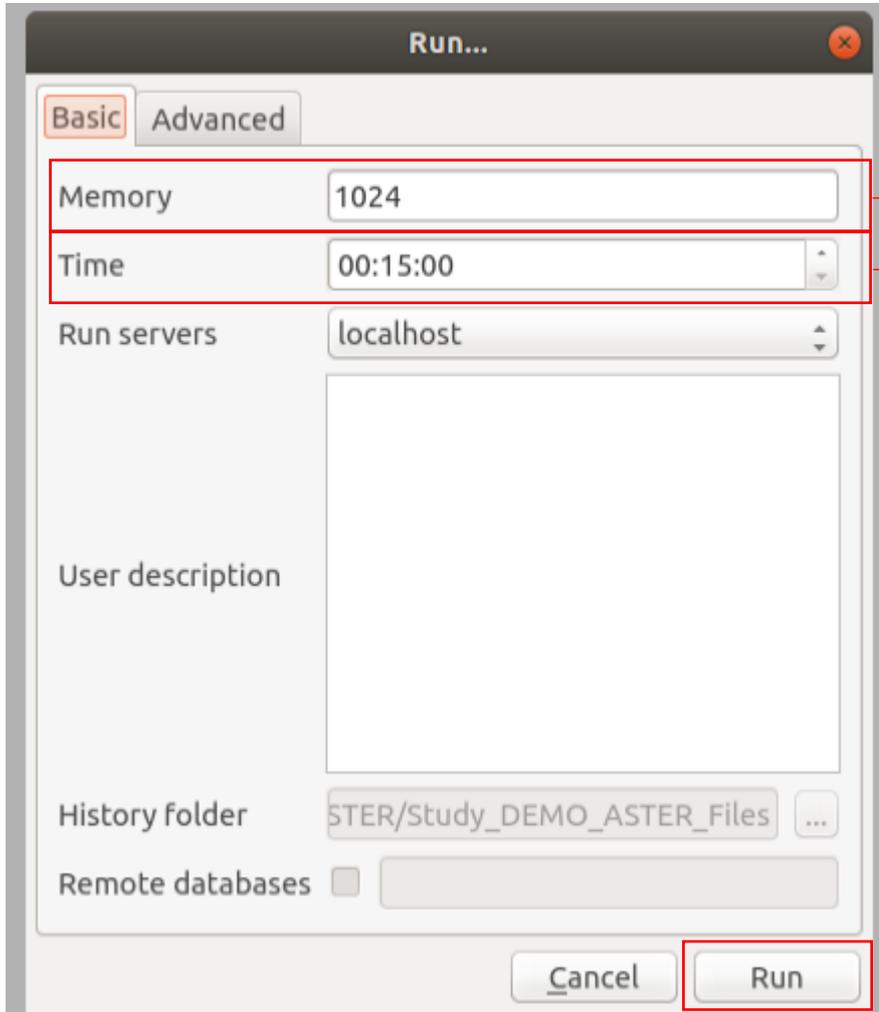
Le voyant rouge passe ensuite au vert :



Vous pouvez ensuite cliquer sur « Run » pour lancer le calcul



Une fenêtre de contrôle s'ouvre, elle permet de définir des paramètres du solveur :



Mémoire attribuée au calcul

Temps maximal alloué au calcul
(attention STOP le calcul en cas de dépassement)

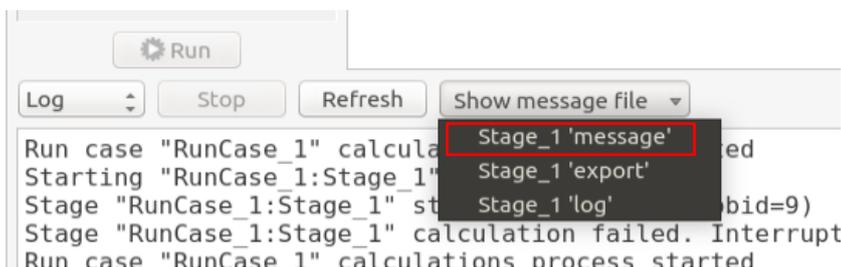
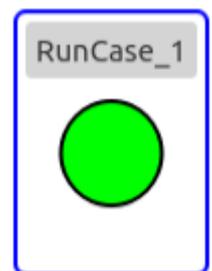
Dans notre cas de figure on peut laisser les paramètres par défaut, Cliquer ensuite sur « Run » pour lancer le calcul.

Le voyant « **Vert** » indique que l'analyse a été réalisée sans problématique.

Un voyant « **Orange** » indique une erreur de calcul.

Un voyant « **Rouge** » indique un problème de syntaxe.

En cas de problème ou d'erreur il est conseillé de regarder le fichier de message (.mess), générée automatiquement lors d'un calcul, il se trouve dans un dossier créé à la racine de l'étude. Une autre possibilité et de faire comme suis :



A la dernière ligne, Code_Aster renvoie un code (ce code donne une indication pour identifier l'erreur s'il y en a une, différents messages d'erreur au long du document sont également présents).

Illustration d'un code d'exécution, ici un code de réussite de la simulation :

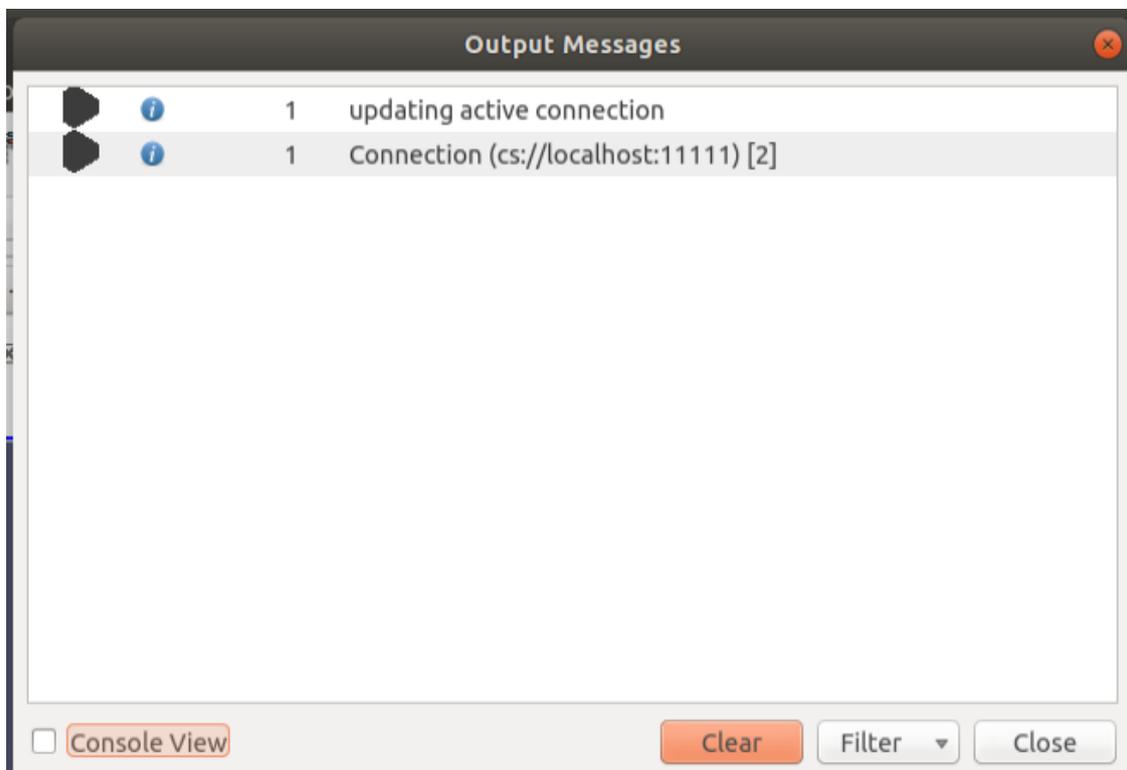
```
EXECUTION_CODE_ASTER_EXIT_5965=0
```

5.5 Analyse des résultats ParaVis / ParaView

L'analyse des résultats se réalise sur le module « ParaVis / Paraview » de Salome_Meca.

Lien du site Web : <https://www.paraview.org/>

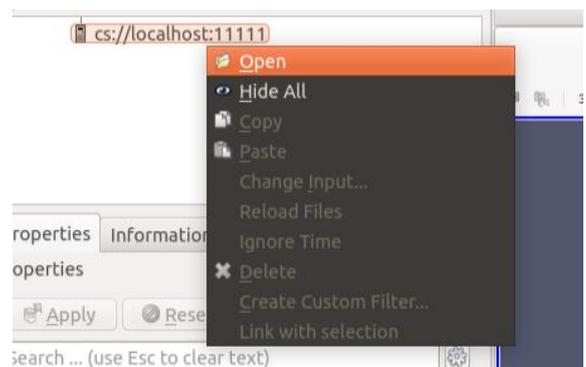
Lors de l'ouverture du module une fenêtre « Output Messages » peut éventuellement s'ouvrir.



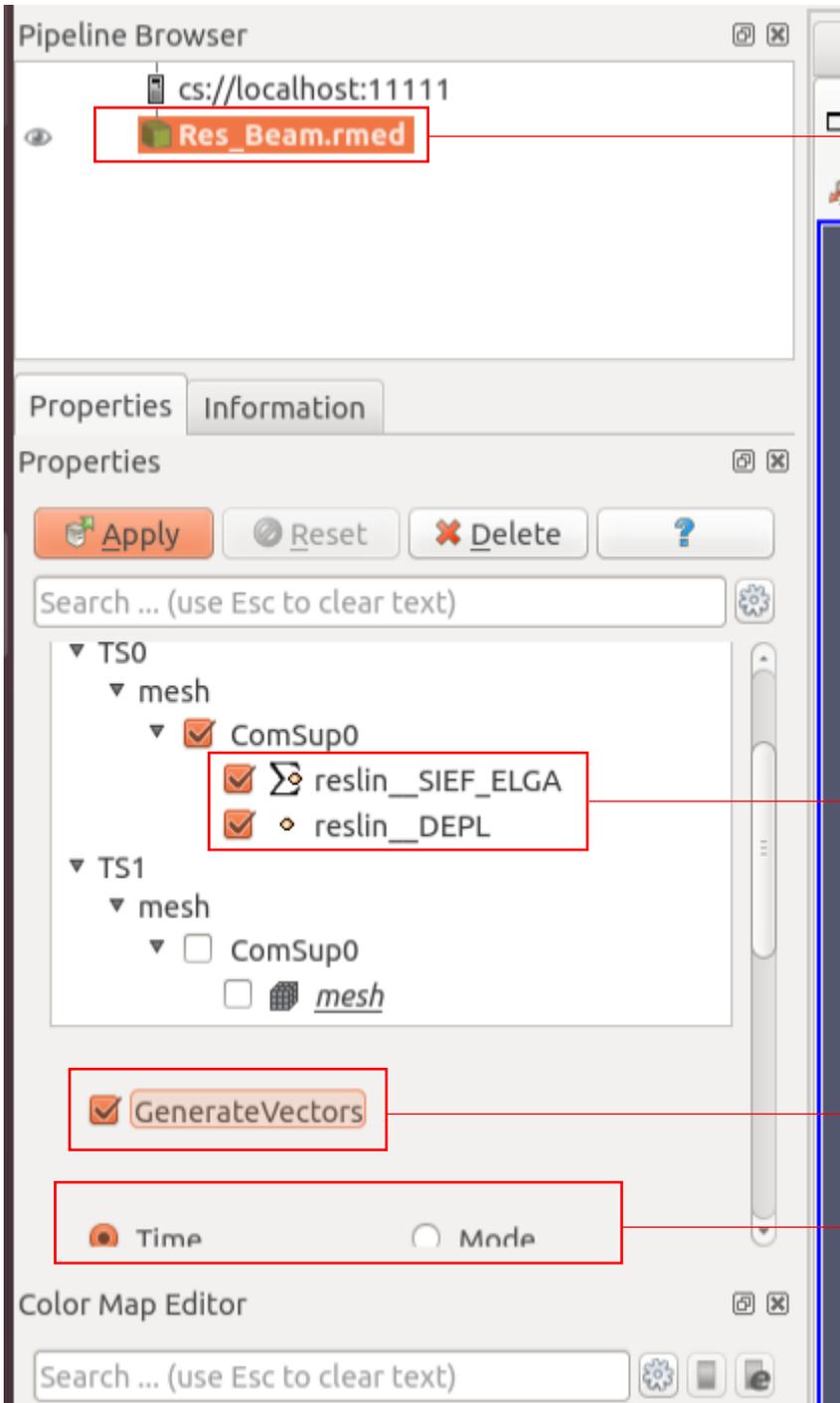
Si c'est le cas, appuyer sur le bouton « Clear » puis fermer la fenêtre.

Pour charger le résultat clic droit sur « Cs://Localhost... »

Cliquer sur « Open » et chercher le fichier résultats dans le répertoire que vous avez défini tout à l'heure.
(le fichier résultats « .rmed »)



Un onglet « Properties » s’ouvre alors, utiliser l’ascenseur et cocher les cases comme suis :



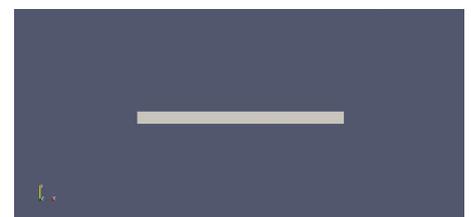
Arborescences des fichiers :
Des fichiers « fils » seront créés
en fonction des filtres ajoutés.

Contrainte & Déplacement

Générer un champ vectoriel
pour afficher une vue déformée

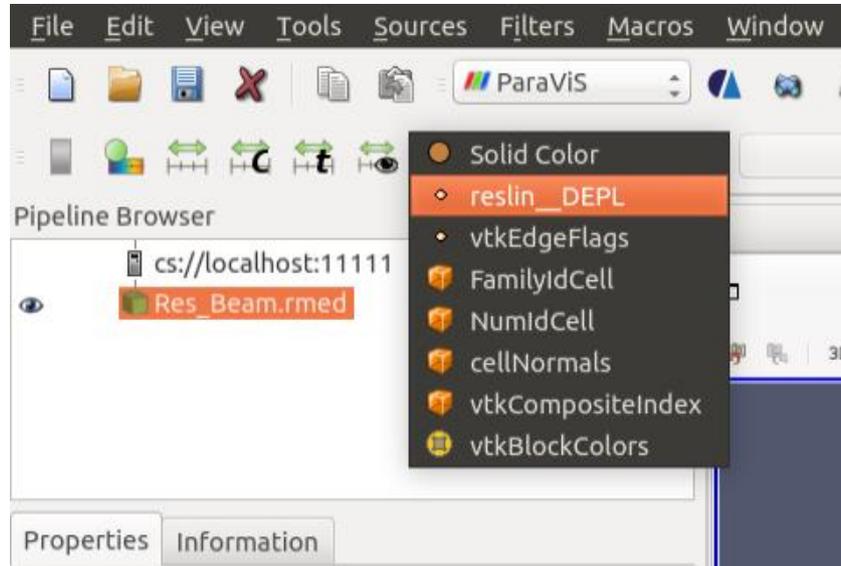
Passage temporel / Modal

Cliquer ensuite sur « Apply ». La poutre apparait sur la fenêtre de visualisation 3D :

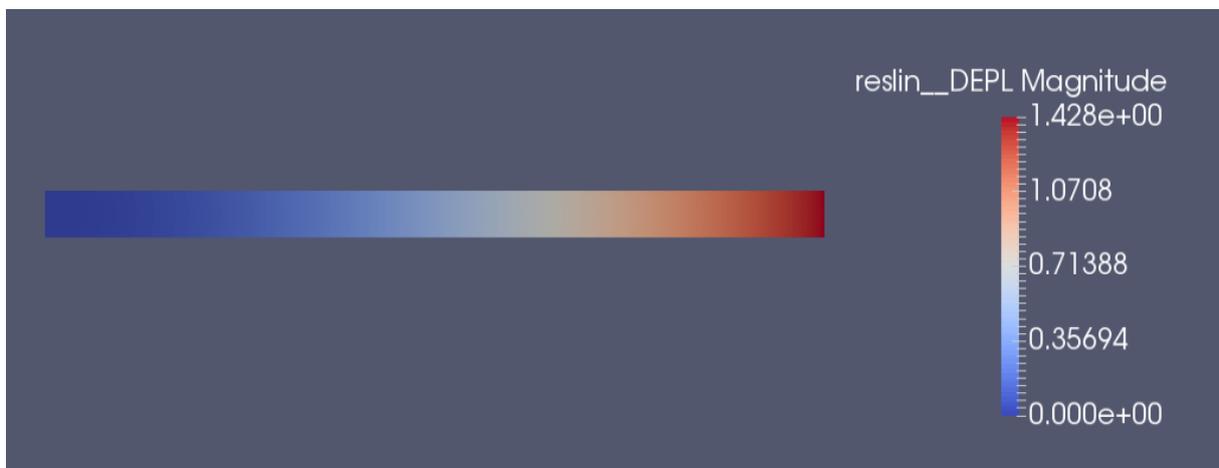


5.5.1 Afficher le déplacement résultant

Pour afficher le déplacement cliquer sur le menu déroulant « Solid Color » :



Le contour de la géométrie affiche désormais le déplacement dans la fenêtre de visualisation 3D :



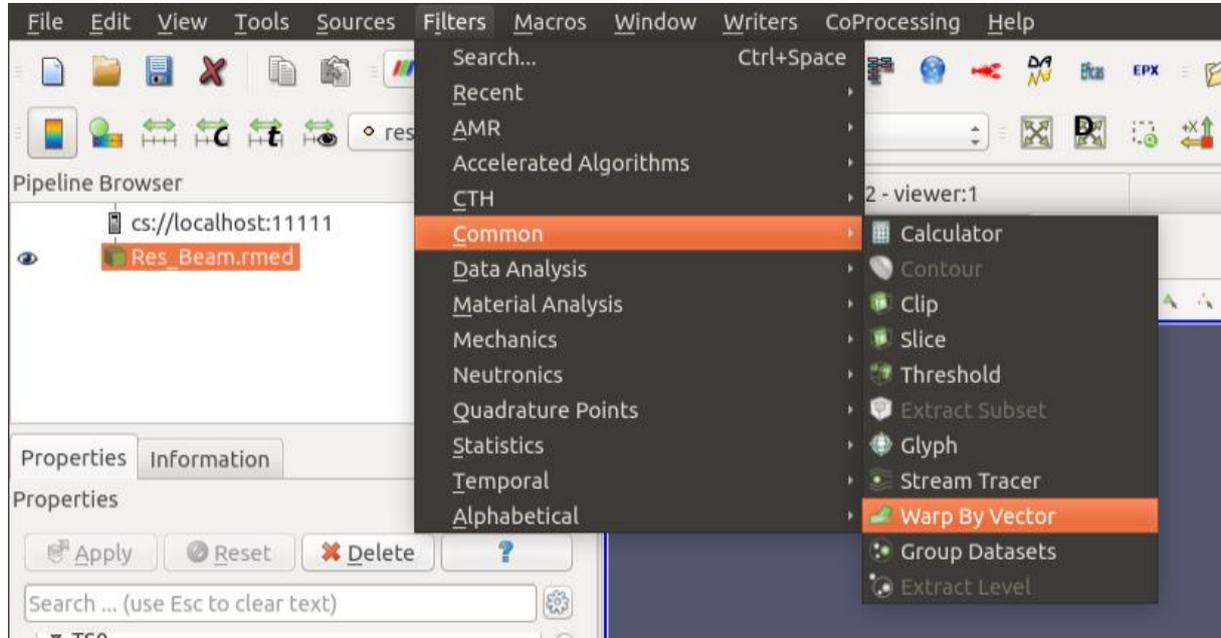
Les résultats du calcul correspondent bien à la valeur calculée de manière analytique « **1.43 mm** ».

Remarque : L'unité des résultats dépend des unités des entrées, ici le système d'unités homogène est :

« MMNTS » : **mm (millimètres) / N (Newton) / t (tonnes) / s (secondes)**

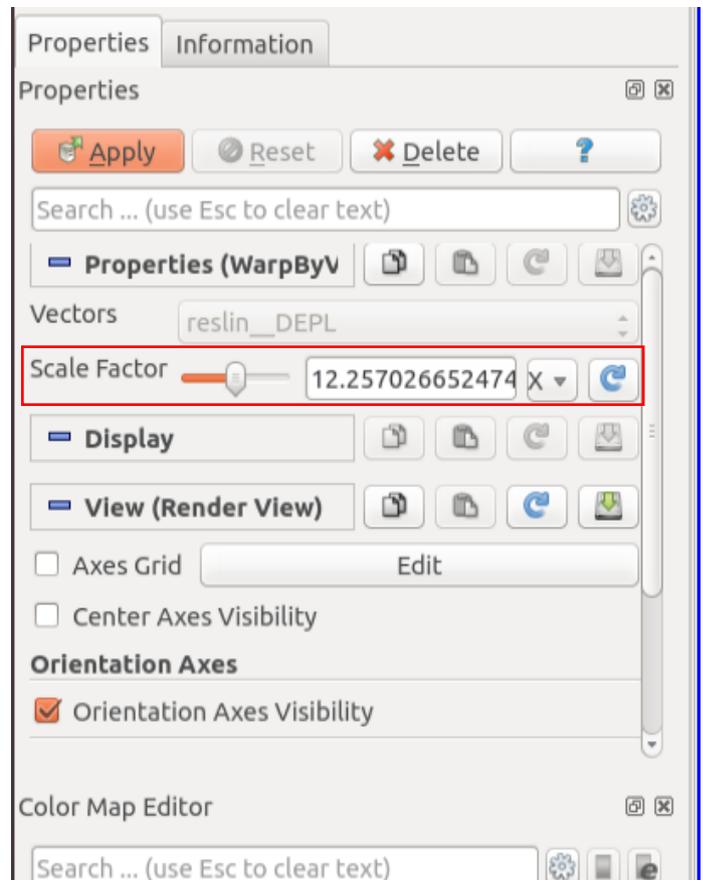
Le **déplacement** s'exprime donc en **millimètre**, la **contrainte** sera en **mégapascal** (MPa = N/mm²)

Pour afficher la vue déformée il suffit de faire comme suis :

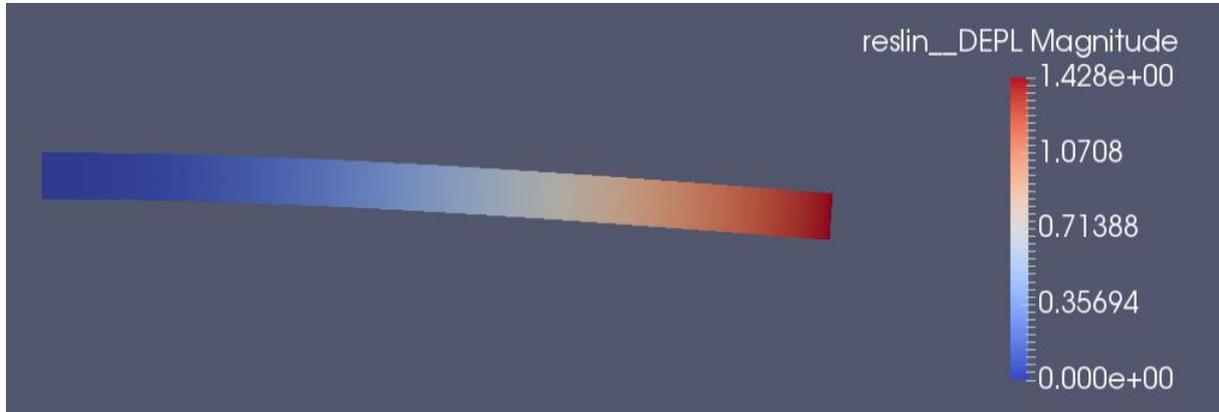


L'onglet de propriétés s'ouvre, cliquer sur « Apply ».

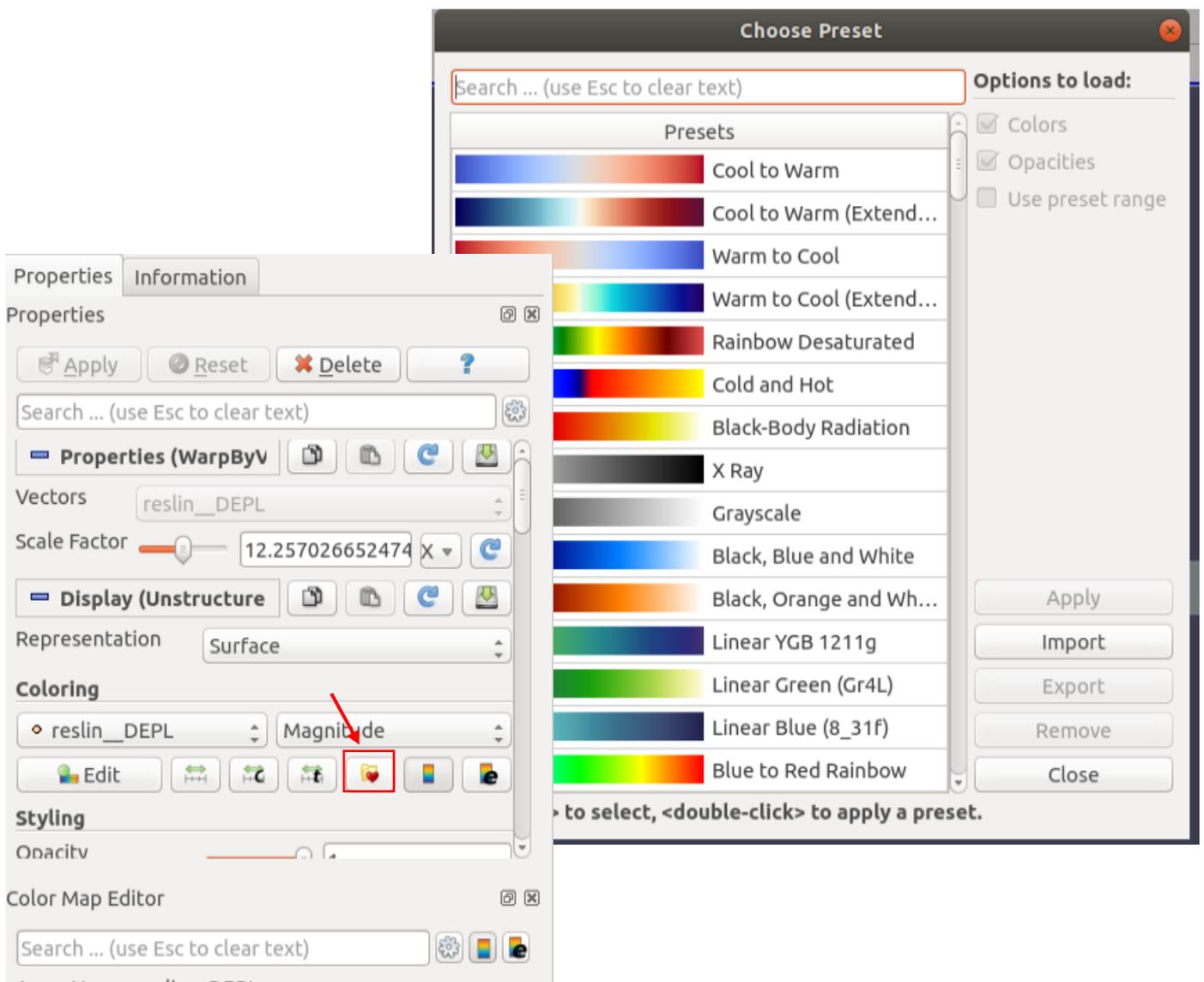
Remarque : Le facteur de déformation numérique de la vue peut être modifié avec le curseur ou la valeur numérique dans le champ « Scale Factor ».



La vue déformée s'affiche alors dans la fenêtre de visualisation 3D :



Il est possible de changer et de personnaliser la palette de couleur d'affichage si cela est souhaité :



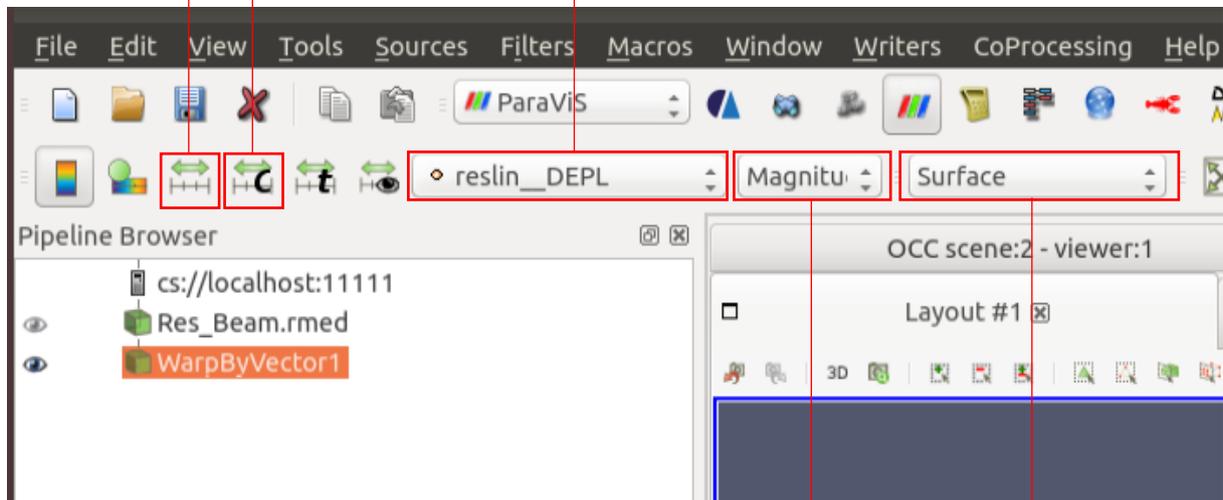
Un certain nombre d'outils et de paramètres d'affichage sont disponibles pour contrôler et gérer l'affichage des résultats sur le contour de la géométrie :

Personnaliser les bornes d'affichage de la plage du contour.
(très utile dans le cas de singularité pour abaisser la contrainte)



Reset la plage de valeur

Contour à afficher sur la fenêtre de visualisation



Modifier le critère de l'affichage
Magnitude / DX / DY / DZ

Modifier le type de l'affichage

Un ensemble d'outil de gestions de la vue sont disponibles pour contrôler la gestion de la vue :

Afficher / Cacher le repère



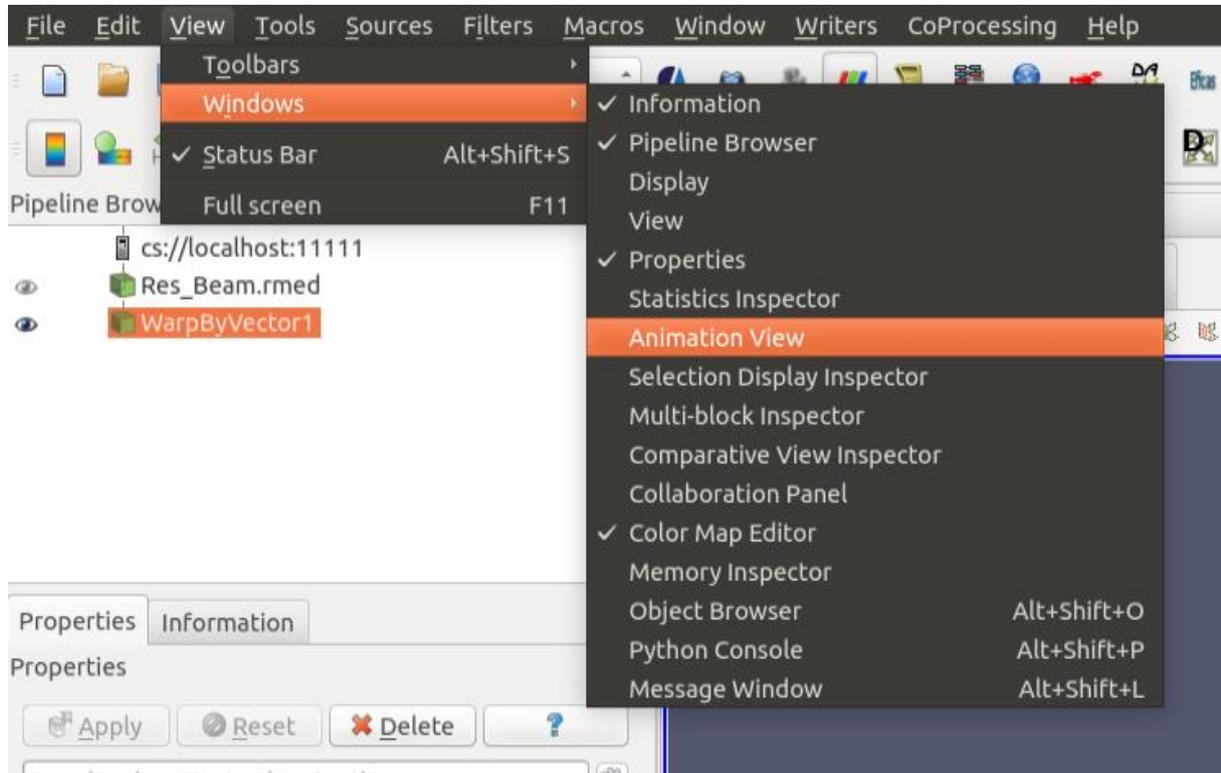
Rescale

Direction de la vue d'affichage

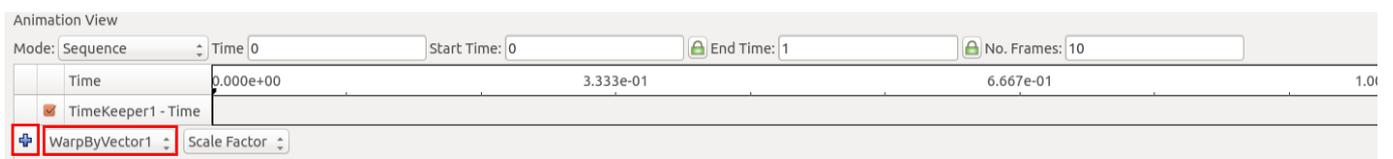
Rotation de la vue

5.5.2 Animation de la vue

Pour réaliser une animation il est tout d'abord nécessaire d'afficher le volet d'animation, comme illustré ci-dessous :



Cliquer ensuite sur le « + » pour ajouter une animation en vérifiant que la vue du déplacement correspondant à la déformée de la poutre est sélectionner, ici « WarpByVector1 ».



Les outils de lecture sont les suivants :



Play/Pause

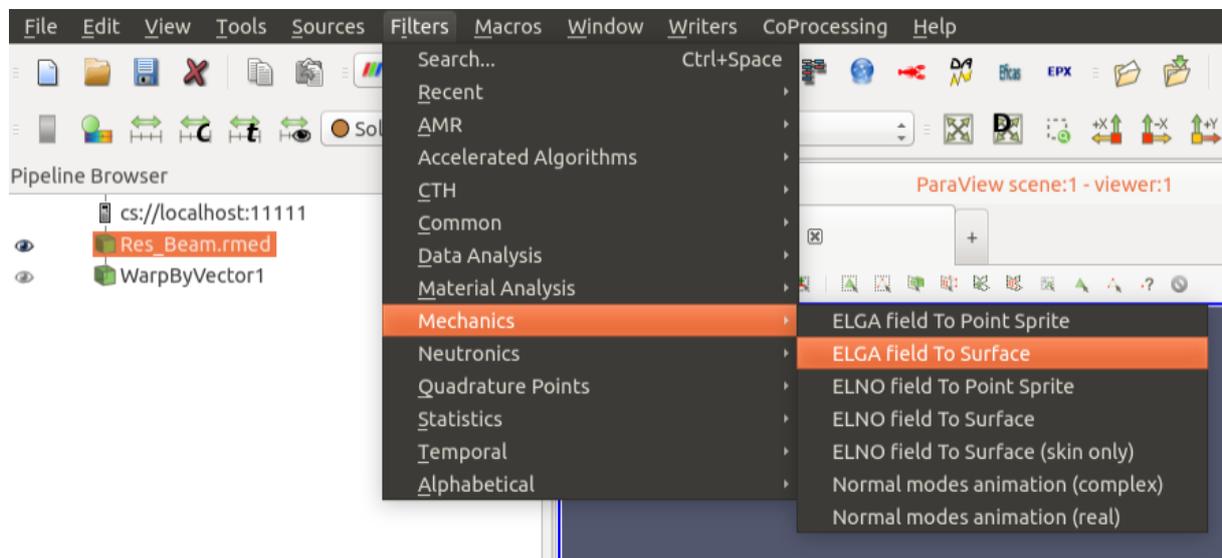
Lecture en boucle

5.5.3 Affichage des contraintes

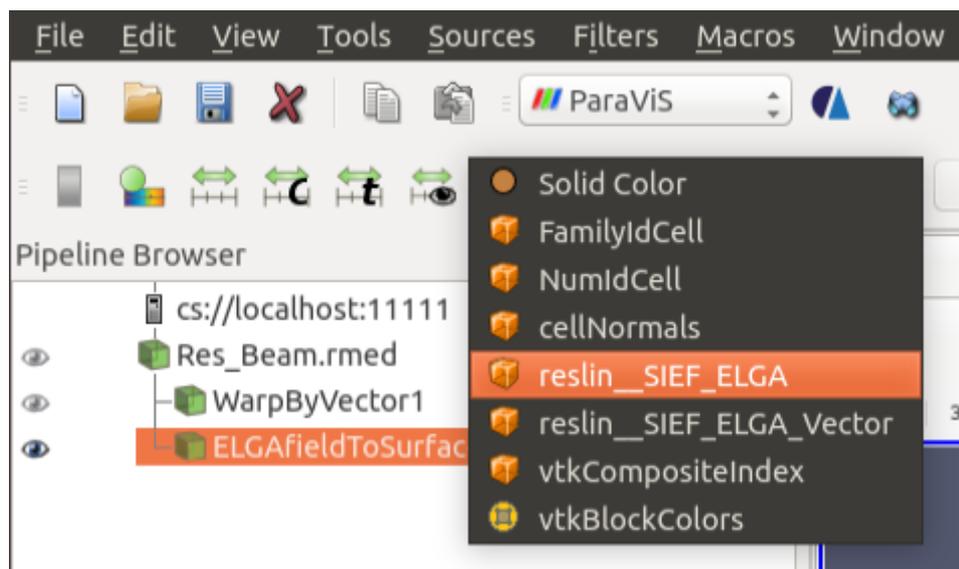
Pour afficher les contraintes il faut procéder comme décrit ci-dessous, dans un premier temps il faut se replacer sur le fichier résultat initial (Pour ne pas prendre en compte la déformée).



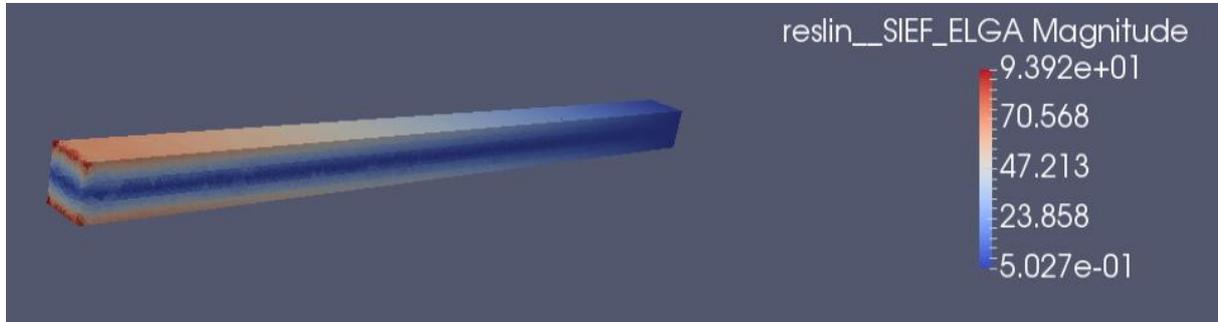
Appliquer le filtre « ELGA Field To Surface » de « Mechanics », cliquer ensuite sur « Apply » :



Sélectionner dans le menu déroulant du contour à afficher l'option « reslin__SIEF_ELGA » :



La contrainte s'affiche ensuite dans la fenêtre de visualisation 3D :



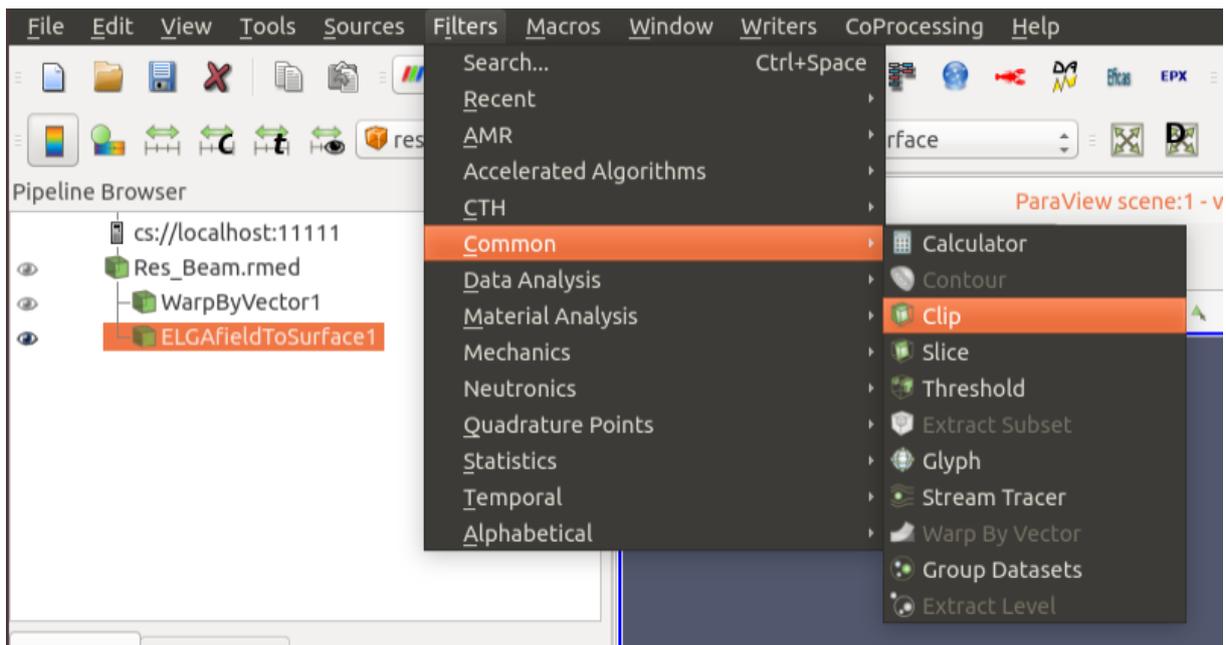
On observe une contrainte maximale au niveau de l'extrémité de la face de fixation dans le sens de flexion. On peut également noter que la zone centrale le long de la fibre neutre est peu sollicitée, ce qui correspond parfaitement aux théories de la poutre. La simulation semble donc pertinente.

La contrainte maximale est de « 94 MPa », cette contrainte est légèrement supérieure à la contrainte calculée de manière analytique « 74 MPa ». Cette différence s'explique par la présence de singularités aux extrémités de la fixation de la poutre (valeur de point singulier non physique dû à l'angle droit et à la condition d'encastrement), la valeur de contrainte en proximité de ces points est néanmoins correcte.

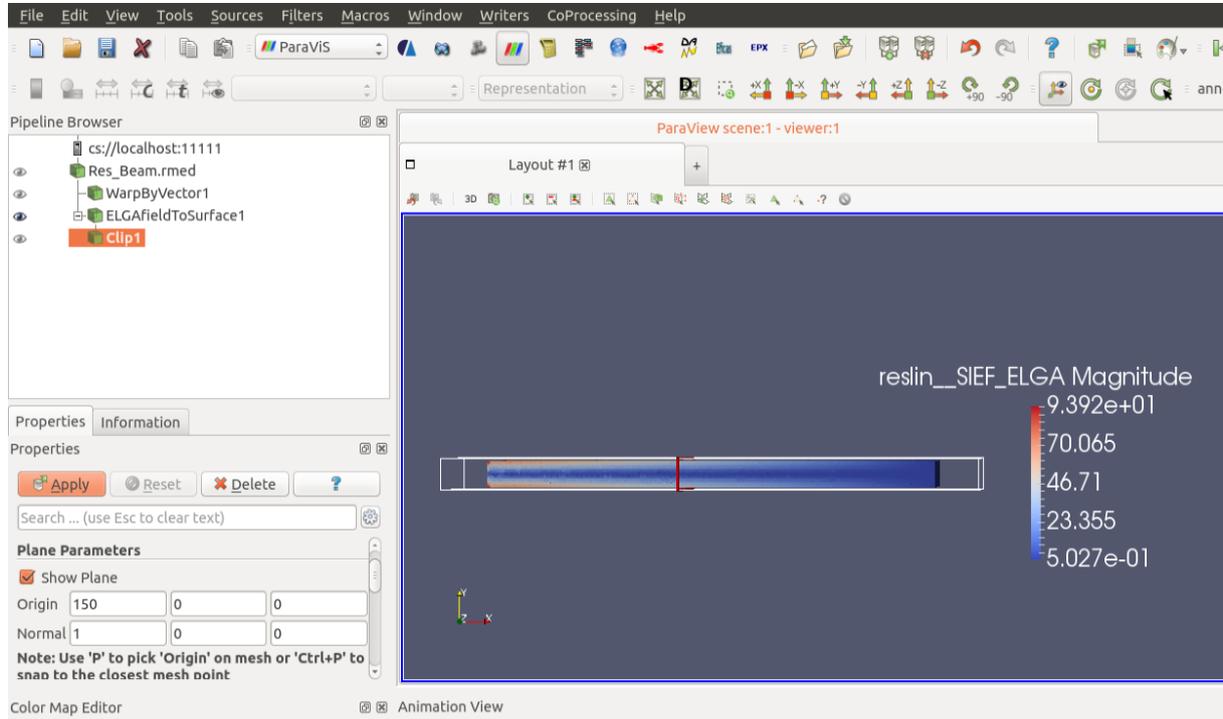
5.5.4 Réalisation d'une vue en coupe sur la vue de résultats

Il est possible de réaliser une vue en coupe de manière à chercher le résultat au sein de la pièce. Pour l'exemple, on souhaite réaliser une coupe à une longueur de « 150mm » de la poutre.

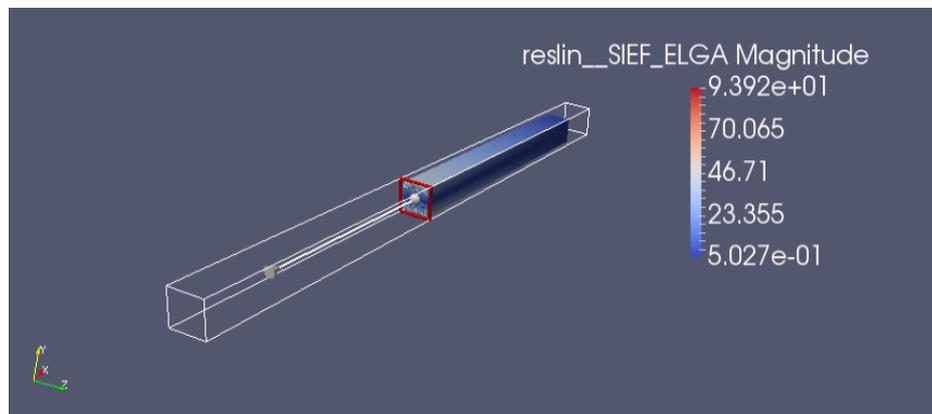
Pour se faire, ajouter un filtre « Clip » :



Paramétrer le plan de coupe comme ci-dessous, puis cliquer sur « Apply » pour valider.



On obtient alors une vue en coupe :



Remarque : Il est possible de cacher le plan de coupe en décochant la case « show Plane ». Il est également possible d'afficher uniquement la section de coupe avec le filtre « Slice ».



Attention : L'échelle du contour ne se met pas à jour automatiquement avec uniquement ce qui est affiché, il faut donc le faire manuellement avec le bouton suivant.



6 Modélisation avec Salome Meca – Script commande Python

Dans cette partie il sera détaillé comment reproduire l'étude de la poutre en flexion en script de commande Python. L'avantage de cette solution est de pouvoir automatiser la simulation, avec l'utilisation de variables de dimension de la poutre par exemple.

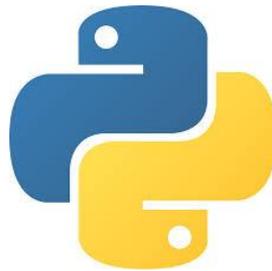
Pour la gestion d'une étude Code_Aster on distinguera 2 scripts :

- Un script Python pour réaliser la géométrie et le maillage avec l'extension « **.py** » (correspondant au module géométrique et maillage de Salome_Meca)
- Un script de commande pour la paramétrisation de l'étude avec l'extension « **.comm** » (correspondant à AsterStudy)

Le script python pourra être exécuté sur la console de Salome_Meca.

6.1 Introduction & présentation du langage de programmation Python

Python est un langage de programmation open source extrêmement utilisé dans le calcul scientifique et l'intelligence artificielle. Ce langage est à la fois simple et extrêmement puissant, s'y intéresser d'une manière ou d'une autre dans le domaine des sciences de l'ingénieur est indispensable.



Lien du site Web de la fondation : <https://www.python.org/>

Il existe une multitude de bibliothèques python qui peuvent être installées et permettent d'ajouter de nouvelles fonctions à vos scripts. On peut citer notamment des bibliothèques disposant d'outils dédiés pour le calcul scientifique ou bien des outils pour la création de courbes ou de figures.

Il existe plusieurs versions du langage python (il n'y a pas de bouleversement d'une version à une autre), dans le cadre de Salome_Meca & Code_Aster la version de python qui devra être utilisée va dépendre de la version des logiciels que vous utilisez.

La version de python nécessaire pour la version de **Salome_Meca 2017** est la suivante :

Python 2.7

Ce document n'a pas pour but d'apprendre le langage Python, de nombreux cours sur internet y sont déjà consacrés. Seules des explications sommaires servant le propos y seront présentées.

L'intégralité des scripts présentés dans ce document seront téléchargeables sur le site de Nadir Ingenia.

6.2 Editeur de texte – Sublime text

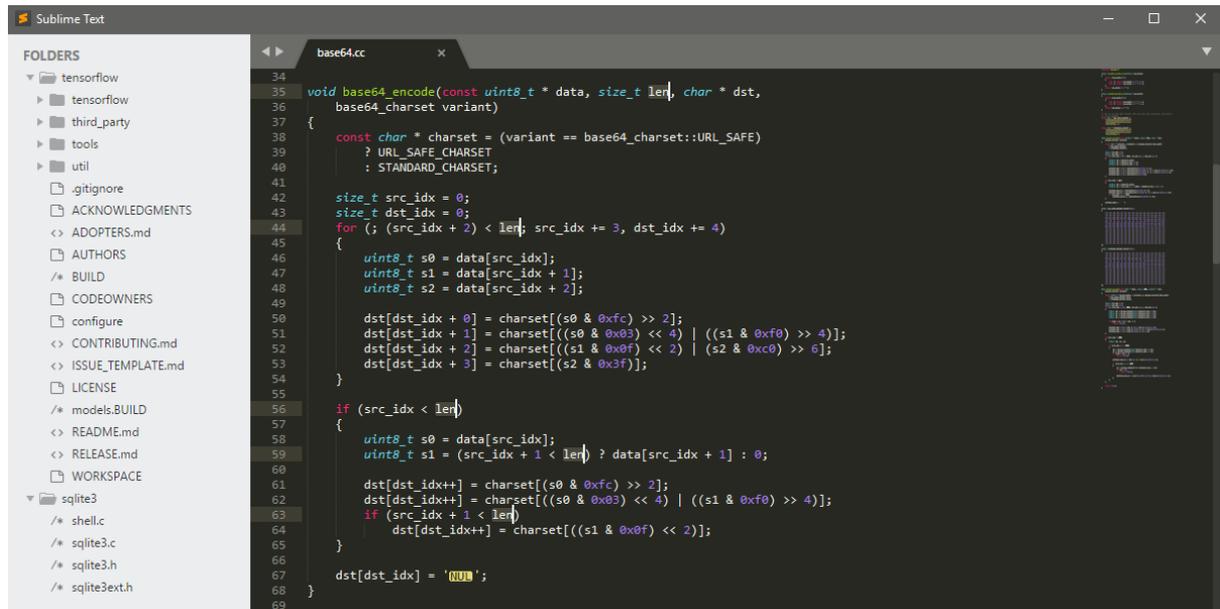
Pour la création de script Python un éditeur de texte est indispensable. Il en existe une multitude et de toute sorte. Une solution gratuite et performante est « **Sublime Text** » disponible sur Linux, Windows ou Mac OS.

L'avantage de cette solution est de pouvoir colorer les mots clés du langage Python (et une multitude d'autres langages), avoir une reconnaissance de variables, de réaliser de l'autocomplétion et disposer d'une gestion de projet. Absolument tout ce qu'on demande à un éditeur de texte en programmation !!



Lien du site : <https://www.sublimetext.com/>

Illustration de l'éditeur de texte « sublime text » :



```

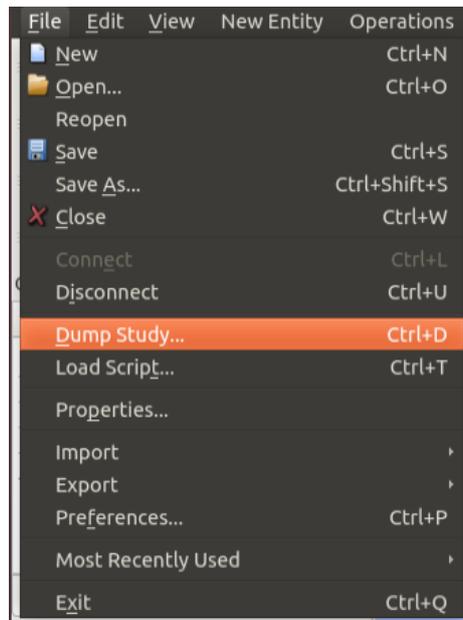
34
35 void base64_encode(const uint8_t * data, size_t len, char * dst,
36 base64_charset variant)
37 {
38     const char * charset = (variant == base64_charset::URL_SAFE
39 ? URL_SAFE_CHARSET
40 : STANDARD_CHARSET;
41
42     size_t src_idx = 0;
43     size_t dst_idx = 0;
44     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
45     {
46         uint8_t s0 = data[src_idx];
47         uint8_t s1 = data[src_idx + 1];
48         uint8_t s2 = data[src_idx + 2];
49
50         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
51         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
52         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
53         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
54     }
55
56     if (src_idx < len)
57     {
58         uint8_t s0 = data[src_idx];
59         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
60
61         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
62         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
63         if (src_idx + 1 < len)
64             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
65     }
66
67     dst[dst_idx] = '\0';
68 }
69

```

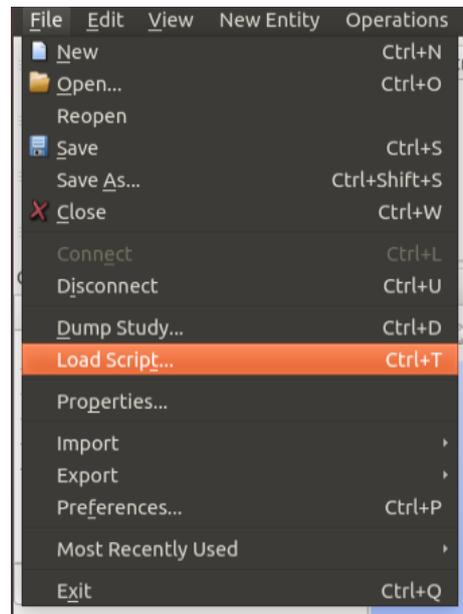
6.3 Script python pour réaliser le maillage et la géométrie

Toute action réalisée sur l'interface graphique de Salome_Meca a une équivalence en langage de programmation Python (et inversement). Il est alors possible de récupérer ces actions en script Python.

Pour ce faire, sélectionner « **Dump Study...** » dans « **File** ».



A l'inverse, pour exécuter un script Python sur Salome_Meca cliquer sur « **Load Script...** »



Remarque : Les commandes peuvent également être éditées directement dans la console Python située par défaut en bas de la fenêtre de visualisation 3D. Néanmoins il est plus pratique et flexible de passer par un script en fichier texte qui n'est pas perdu après l'exécution.

6.3.1 Création de la géométrie par un script Python

Lien du site Web des commandes Python du module géométrique : https://docs.salome-platform.org/latest/gui/GEOM/geompy_page.html

Lien du script de la création de la géométrie du maillage de la poutre : https://github.com/Nadir-Ingenia/Salome_Meca-Code_Aster/blob/main/Poutre_Aster_Script_demo

Il est conseillé de commencer son script Python avec les lignes suivantes de manière à indiquer à l'éditeur de texte le langage de programmation et le type de codage des caractères textuels utilisés.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
```

Si le type de langage de programmation n'est pas identifié automatiquement, il est possible de le sélectionner manuellement en cliquant en bas à gauche de la fenêtre de Sublime Text. (cette opération sera sûrement nécessaire pour la première ouverture d'un script).

Remarque : Les caractères de la ligne suivant « # », ou les caractères de la ligne entre « " " » ; ou les caractères de la ligne entre « ' ' », ou les caractères du paragraphe entre « "" "" » seront considérés comme des commentaires et ne seront pas pris en compte dans l'exécution du code. Il est fortement conseillé de commenter son script de manière à favoriser sa lisibilité.

La première étape est de réaliser l'import de plusieurs bibliothèques qui seront nécessaires par la suite :

```
13  from math import *
14  import sys
15  import salome
```

Il est intéressant de passer par des variables pour la construction de la géométrie plutôt que d'utiliser directement les valeurs numériques. Cela permet d'automatiser tout le processus de l'étude en fonction de ces paramètres.

```
18  #####
19  ##                               Variables                               ##
20  #####
21
22  # Longueur de la poutre - L_p (mm) :
23  L_p = 350
24
25  # largeur de la poutre - la_p (mm) :
26  la_p = 20
27
28  # Hauteur de la poutre - h_p (mm)
29  h_p = 20
30
```

Tout le script sera alors basé sur ces variables. Modifier l'une d'elles permettra de reconstruire automatiquement la géométrie de la poutre et son maillage.

Initialisation de Salome_Meca et du Notebook :

```

31 salome.salome_init()
32 theStudy = salome.myStudy
33
34 import salome_notebook
35 notebook = salome_notebook.NoteBook(theStudy)
36 sys.path.insert( 0, r'../Demo_Poutre_Aster_Script.py')

```

Le **NoteBook** est l'outil de gestion de variables de Salome_Meca. Seules les variables définies dans celui-ci seront utilisables dans Salome_Meca. Des opérations mathématiques ne peuvent être faites sur ces valeurs dès lors qu'elles sont définies. Il faudra donc les anticiper et créer des variables dédiées dans celui-ci s'il le faut.

```

39 #####
40 ##      Begin of Notebook variables section      ##
41 #####
42
43 # Parametres de construction de la géométrie
44
45 notebook.set("P1_X", 0)
46 notebook.set("P1_Y", -La_p/2)
47 notebook.set("P1_Z", -h_p/2)
48
49 notebook.set("P2_X", 0)
50 notebook.set("P2_Y", -La_p/2)
51 notebook.set("P2_Z", h_p/2)
52
53 notebook.set("P3_X", 0)
54 notebook.set("P3_Y", La_p/2)
55 notebook.set("P3_Z", h_p/2)
56
57 notebook.set("P4_X", 0)
58 notebook.set("P4_Y", La_p/2)
59 notebook.set("P4_Z", -h_p/2)
60
61 #
62
63 notebook.set("P5_X", L_p)
64 notebook.set("P5_Y", -La_p/2)
65 notebook.set("P5_Z", -h_p/2)
66
67 notebook.set("P6_X", L_p)
68 notebook.set("P6_Y", -La_p/2)
69 notebook.set("P6_Z", h_p/2)
70
71 notebook.set("P7_X", L_p)
72 notebook.set("P7_Y", La_p/2)
73 notebook.set("P7_Z", h_p/2)
74
75 notebook.set("P8_X", L_p)
76 notebook.set("P8_Y", La_p/2)
77 notebook.set("P8_Z", -h_p/2)

```

Une fois que les variables du **NoteBook** sont définies il faut charger le module géométrique de Salome_Meca :

```
81 import GEOM
82 from salome.geom import geomBuilder
83 import math
84 import SALOMEDS
85
86 geompy = geomBuilder.New(theStudy)
```

Il est conseillé de toujours créer le point de centre ainsi que les vecteurs des axes du repère, ces objets pourront être nécessaires dans la réalisation de diverses opérations.

```
O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)
```

La méthode de création de points est la ligne de commande suivante :

```
93 P1 = geompy.MakeVertex("P1_X", "P1_Y", "P1_Z")
```

La méthode de création de segments à partir de points est la ligne de commande suivante :

```
103 LA = geompy.MakeEdge(P1, P2)
```

La méthode de création de faces à partir de segments est la ligne de commande suivante :

```
118 S1 = geompy.MakeFaceWires([LA, LB, LC, LD], 1)
```

La méthode de création de solides à partir de faces est la ligne de commande suivante :

```
125 Poutre = geompy.MakeSolidFromConnectedFaces([S1, S2, S3, S4, S5, S6], True)
```

Les objets créés jusqu'à présent sont fictifs, ils ne sont pas ajoutés dans l'arbre géométrique. Pour ajouter un objet à l'arbre de géométrie l'opération suivante est à réaliser :

```
127 # Ajout de la géométrie dans l'étude
128 geompy.addToStudy( Poutre, 'Poutre' )
```

Pour créer les groupes de conditions aux limites et de chargement il est nécessaire de procéder comme suit :

```

130 # Extraction des Faces de contidions aux limites et des chargements et ajout à l'objet "père"
131 Face_All=geompy.ExtractShapes(Poutre, geompy.ShapeType["FACE"],True)
132 for i in range(len(Face_All)):
133     xyz=geompy.PointCoordinates(geompy.MakeCDG(Face_All[i]))
134     if abs(xyz[0])<0.01:
135         geompy.addToStudyInFather(Poutre, Face_All[i], "Fixed")
136         Face_Fixed=Face_All[i]
137     if abs(xyz[0])>L_p-0.01:
138         geompy.addToStudyInFather(Poutre, Face_All[i], "Forced")
139         Face_Forced=Face_All[i]
    
```

Cette opération est un peu plus complexe et mérite une explication plus détaillée.

Il faut comprendre qu'avec la méthode graphique il était possible de cliquer directement sur les faces voulues pour les sélectionner. Or avec une méthode par script de commande Python cela n'est plus possible, il est donc nécessaire de distinguer numériquement une face par rapport à une autre.

L'astuce ici est alors d'extraire toutes les faces du solide « Poutre » et pour chacune des faces y extraire son centre de gravité.

Une boucle « if » (si) est alors utilisée pour filtrer et balayer parmi toutes les faces, lesquelles correspondent à un critère. Dans notre cas de figure le critère est basé sur la valeur absolue de la coordonnée x du centre de gravité de chaque face « xyz[0] ». En effet pour identifier les faces correspondant aux extrémités de la poutre, il suffit de filtrer celles qui possèdent un centre de gravité avec pour coordonnées « x = 0 » et l'autre « x = L_p ».

Remarque : On peut noter que sur le script une tolérance de « 0.01 » a été ajoutée sur la filtration des faces. Il est conseillé de toujours prendre une tolérance pour éviter les erreurs numériques.

6.3.2 Création du maillage en script Python

Le maillage de la géométrie peut être réalisé dans le même fichier (à la suite du script présenté précédemment) que la création de la géométrie.

Lien de la documentation des commandes du module maillage : https://docs.salome-platform.org/latest/gui/SMESH/smashpy_interface.html

Lien du script de la création de la géométrie du maillage de la poutre : https://github.com/Nadir-Ingenia/Salome_Meca-Code_Aster/blob/main/Poutre_Aster_Script_demo

Pour initialiser le module du maillage la commande est la suivante :

```

145 import SMESH, SALOMEDS
146 from salome.smesh import smeshBuilder
147
148 smesh = smeshBuilder.New(theStudy)
    
```

Commande pour la paramétrisation du maillage :

```
150 Mesh_Poutre = smesh.Mesh(Poutre)
151 NETGEN_1D_2D_3D = Mesh_Poutre.Tetrahedron(algo=smeshBuilder.NETGEN_1D2D3D)
152 NETGEN_3D_Parameters_1 = NETGEN_1D_2D_3D.Parameters()
153 NETGEN_3D_Parameters_1.SetMaxSize( 4 )
154 NETGEN_3D_Parameters_1.SetSecondOrder( 1 )
155 NETGEN_3D_Parameters_1.SetMinSize( 2 )
```

Création des groupes du maillage par rapport aux groupes de la géométrie :

```
157 Poutre_1 = Mesh_Poutre.GroupOnGeom(Poutre, 'Poutre', SMESH.VOLUME)
158 Fixed_1 = Mesh_Poutre.GroupOnGeom(Face_Fixed, 'Fixed', SMESH.FACE)
159 Force_1 = Mesh_Poutre.GroupOnGeom(Face_Forced, 'Forced', SMESH.FACE)
160 [ Poutre_1, Fixed_1, Force_1 ] = Mesh_Poutre.GetGroups()
```

Réalisation du maillage de la géométrie :

```
162 isDone = Mesh_Poutre.Compute()
```

Le script de la création de la géométrie et du maillage est terminé. Il peut être exécuté sur Salome_Meca comme expliqué en début de chapitre.

6.3.3 Création du script de commande de l'analyse

Lien du script de commande Nadir Ingenia : https://github.com/Nadir-Ingenia/Salome_Meca-Code_Aster/blob/main/COM_DEMO_POUTRE_ASTER

Remarque : Une grande précaution est à prendre avec les virgules et la fermeture des parenthèses. Ces éléments sont la plus grande source d'erreurs de syntaxe.

Le script commence avec la commande :

```
13 DEBUT()
```

Lecture du maillage :

```
15 mesh = LIRE_MALLAGE(FORMAT='MED', UNITE=20)
```

Remarque : Comme expliqué lors de la méthode par interface graphique, un numéro d'unité est à déclarer pour identifier correctement le fichier d'entrée (ici le maillage).

Définition du type de maillage :

```
17 model = AFFE_MODELE(
18     AFFE=(
19         _F(GROUP_MA=('Poutre', 'Forced',), MODELISATION=('3D', ), PHENOMENE='MECANIQUE'),
20     ),
21     MAILLAGE=mesh
22 )
```

Remarque : On peut noter que le groupe d'application du chargement est également à déclarer.

Définition du matériau :

```
24 mater = DEFI_MATERIAU(ELAS=_F(E=215000, NU=0.3))
```

Affectation du matériau aux maillages :

```
26 ▼ fieldmat = AFFE_MATERIAU(  
27     AFFE=(  
28         _F(GROUP_MA=('Poutre',), MATER=(mater, )),  
29     ),  
30     MODELE=model  
31 )
```

Définition des chargements et des conditions aux limites :

```
33 load = AFFE_CHAR_MECA(  
34     DDL_IMPO=(  
35         _F(GROUP_MA=('Fixed'), LIAISON='ENCASTRE'),  
36     ),  
37     FORCE_FACE=(  
38         _F(FX=-0.7, GROUP_MA=('Forced')),  
39     ),  
40     MODELE=model  
42 )
```

Définition du type de l'analyse :

```
44 reslin = MECA_STATIQUE( CHAM_MATER=fieldmat, EXCIT=_F(CHARGE=load),  
45     MODELE=model  
46 )
```

Impression des résultats :

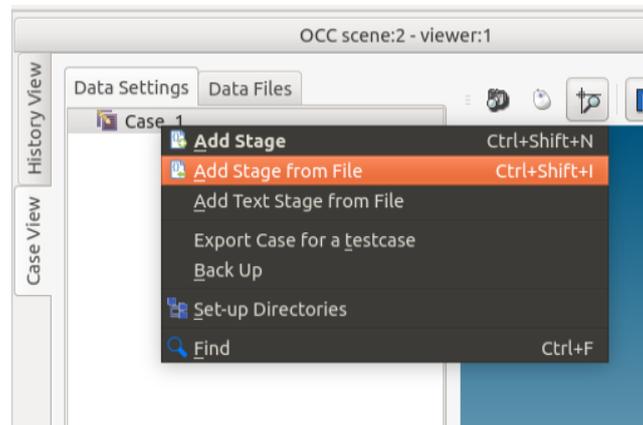
```
48 IMPR_RESU(FORMAT='MED', RESU=_F(RERESULTAT=reslin), UNITE=80)
```

Remarque : Comme expliqué lors de la méthode par interface graphique, un numéro d'unité est à déclarer pour identifier le fichier de sortie (ici le résultat de l'étude).

Commande de fin du script :

```
50 FIN()
```

Le script de commande peut ensuite être importé dans le module de « Aster_Study » de la manière suivante.



Remarque : Dans le cas d'un problème de syntaxe une fenêtre de débogage s'ouvre.

Les entrées (lecture du maillage) et les sorties (impression des résultats) doivent être impérativement récupérées en éditant les fonctions de la même manière que la méthode par l'interface graphique.

Le calcul est ensuite à lancer et à être exploité d'une façon identique à l'interface décrite plus haut dans le rapport.

6.4 Présentation de ASTK

Une solution annexe plus pertinente existe pour s'affranchir complètement de l'interface graphique. Cette solution est intégrée au pack d'installation de Salome_Meca & Code_Aster.

6.4.1 Lancement de ASTK

Lancement du module ASTK via l'interface graphique de « Salome_Meca »



Remarque : « astk » est à prononcer « asteK ».

Lancement du module ASTK en ligne de commande dans le terminal :

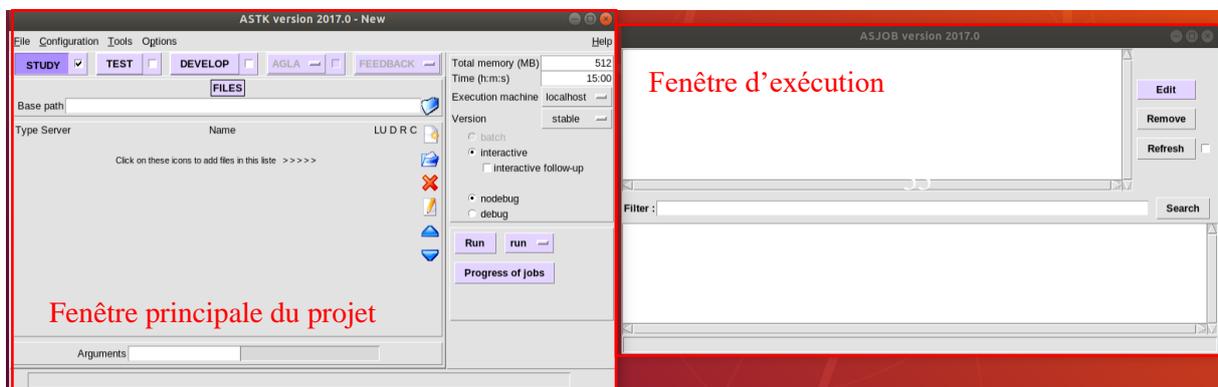
Exécuter dans le terminal les commandes ci-contre :

« *sudo su* » (super-utilisateur linux)

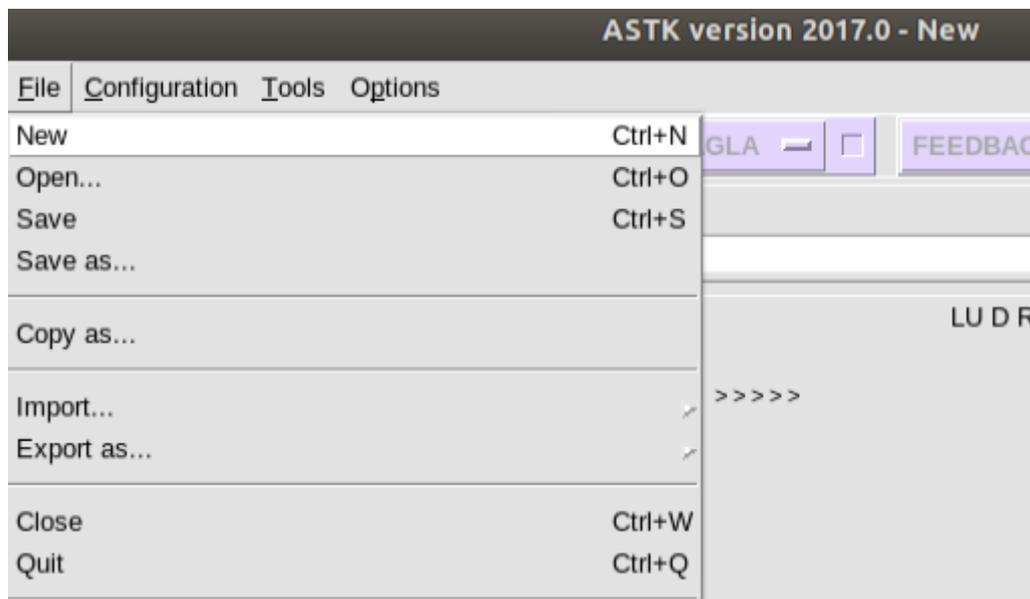
« *Salome shell* » (environnement Salome)

« *astk* » (lancer le module astk)

ASTK se compose des deux fenêtres suivantes :



Pour créer une nouvelle étude astk, cliquer sur « New » dans « File » :

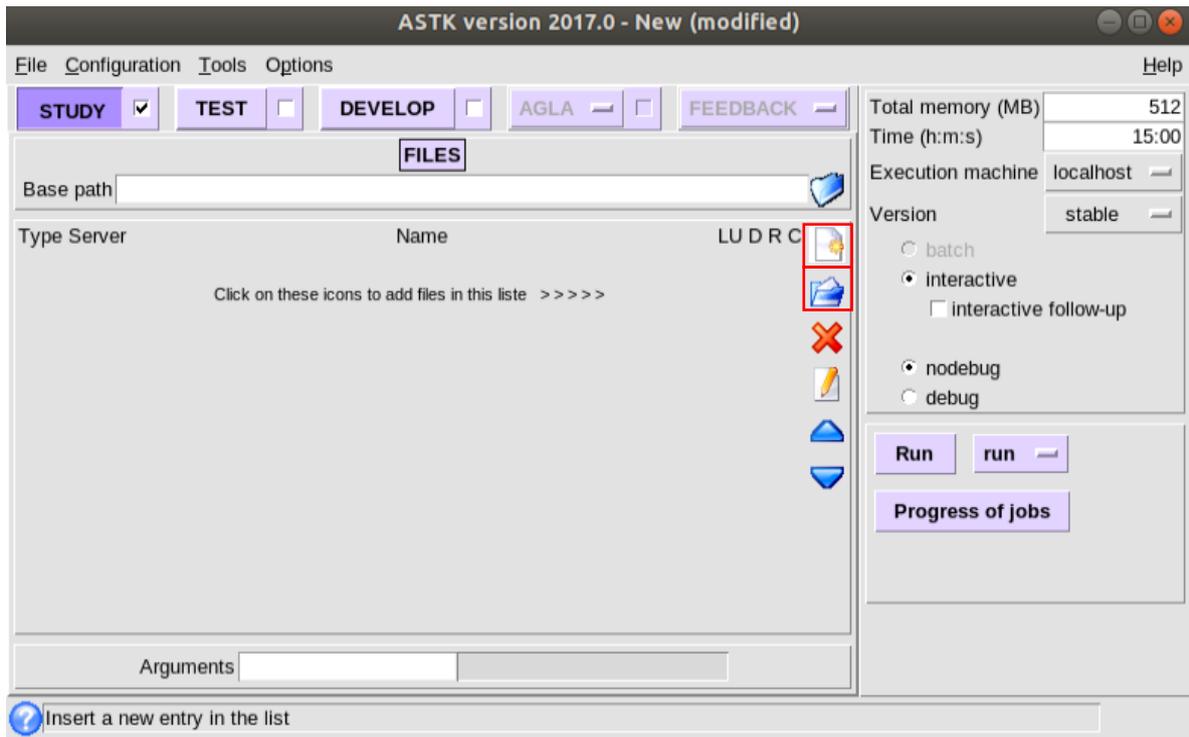


ASTK utilise les mêmes fichiers que Salome_Meca, le fichier de commande (exporter en « .comm »), le maillage (exporter en « .med ») seront nécessaires. Ces documents sont créés de la manière décrite plus haut dans ce rapport.

Les fichiers d'une même étude étant souvent localisés dans un même dossier, ASTK permet d'identifier la racine de base pour s'en affranchir :



Pour créer l'analyse il faut ajouter à ASTK les différents fichiers nécessaires à l'étude. Pour ajouter un document existant cliquer sur « **open file** » et cliquer sur le bouton « **insert** » pour la création de nouveaux fichiers.



Les fichiers devront être ouverts sur ASTK dans l'ordre suivant :

Type	Server	Name	LU	D	R	C	
comm	Local	./COM_DEMO_POUTRE_ASTER.comm	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
mmed	Local	./Poutre_mesh.med	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
mess	Local	./log.mess	6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
rmed	Local	./result.rmed	80	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Attention les unités des entrées et sorties du fichier commande doivent être identiques avec l'indication de la rubrique « LU ». Modifier au besoin.

Sauvegarder ensuite l'étude.

Pour lancer l'étude cliquer sur « RUN »



Si le calcul a été exécuté sans problématique, la fenêtre d'exécution renvoie « retourne 0 ».

Le fichier « **.mess** » renvoie les logs du calcul.

Une fois une étude créée, il est possible d'exécuter le calcul uniquement avec le terminal avec le fichier « **.export** » qui est créé l'or de la sauvegarde d'une étude.

Pour se faire exécuter la commande sous « **salome shell** » suivante :

« as_run ../calcul.export »

Remarque : Dans cet exemple « calcul.export » et le nom du fichier de l'étude.