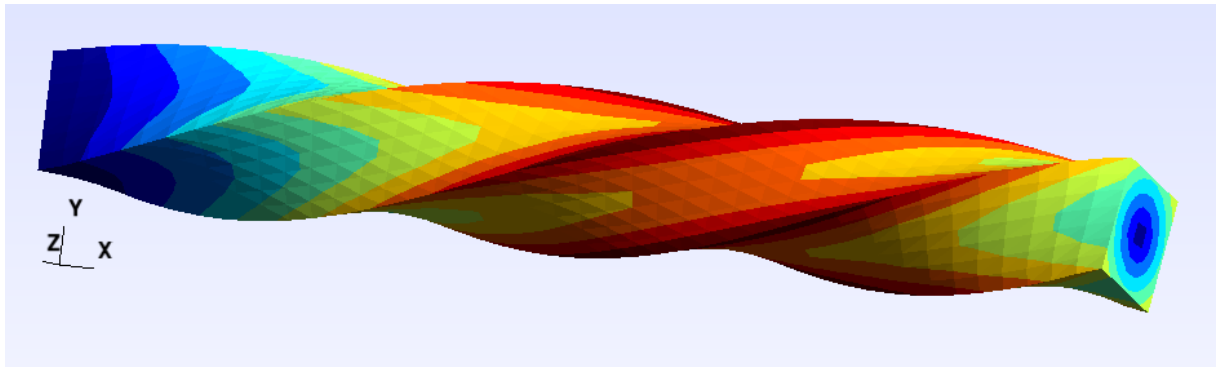


Tutoriel : Torsion cinématique

Mise en place d'une torsion cinématique sur une
poutre et affinage du maillage

Gaëtan ROMAN

29/05/2017



Sommaire

I)	Introduction :	3
II)	Torsion simple :	3
a)	Création de la poutre.....	3
b)	Calcul :.....	4
c)	Résultat :.....	6
III)	Affinage du maillage :	8
IV)	Conclusion :.....	10

I) Introduction :

Ce document à pour but d'expliquer comment introduire une torsion cinématique sur une poutre. Nous utiliserons pour cela mouvement solide en rotation d'un certain angle. D'ailleurs il sera possible d'affiner le maillage sous gmsh pour ainsi obtenir une visualisation plus précise.

Ci-joint le dossier permettant de vous aider expliqué en fin du tutoriel.

II) Torsion simple :

Dans cette partie nous allons expliquer la marche à suivre pour réaliser une torsion cinématique sur une poutre.

a) Création de la poutre

Sous stamm il est possible de créer des maillages simples comme des pièces cylindriques, des plaques etc... Nous allons voir la procédure à suivre pour créer notre poutre de dimension : 100x10x10 (mm) voir figure 1.

```

type d'elements : (1D, 2D, 3D) ? 3D
choix lu: 3D
type de decoupage :
brique (reponse b ) ? b
choix de l'interpolation :
lineaire (reponse li ) ?
quadratique complet (reponse qc ) ? qc
calcul de la position des noeuds :
de maniere exacte (reponse e) ?
de maniere aleatoire autour d'une position exacte (reponse a) ?
aleatoire uniquement sur les noeuds internes (reponse i) ? e
type de geometrie :
prisme rectangulaire (reponse pr ) ?
cylindre creux (reponse cyl ) ?
parallelogramme eleve (reponse prl ) ?
portion de cylindre (reponse pcy ) ?
plaque en helice (reponse phe ) ?
portion de dome hemispherique (reponse p_dome ) ?
anneau (reponse ann ) ?
cylindre plein (reponse cyp ) ?
dome hemispherique complet (reponse dome ) ? pr
dimension du prisme rectangulaire: longueur x ? 100
largeur ? 10
hauteur ? 10
nombre d'element(s) dans la longueur ? 5
dans la largeur ? 1
dans la hauteur ? 1
nombre de point d'interpolation standard 8 ? (rep o ou n) n
nombre de pt possible : 8 27 64 : choix ? 27
nom du fichier de sortie ? : poutre
nom lue : poutre
un autre maillage ? (rep o ou n) n
=====
| fin stamm |
=====

```

On utilise des éléments 3D

Un découpage de type brique

Une interpolation quadratique complet

Le calcul de la position des nœuds se fera de manière exacte

Notre poutre est un prisme rectangulaire

Dimension de notre poutre

Nombre d'élément de notre poutre

On utilise 27 points d'intégrations

Nom du fichier possédant le maillage en sortie

Figure 1: Création de la poutre

Sous `hz_visuMail.pl` il est alors possible de visualiser le maillage obtenu avec la commande suivante « `hz_visuMail.pl nomfichier.her` ». Nous visualisons cela (figure 2) :

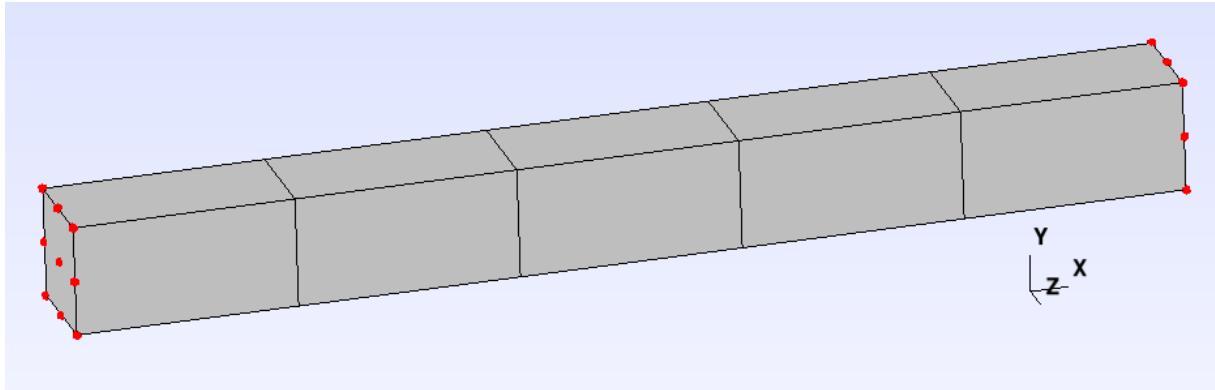


Figure 2: Maillage de la poutre

Nous observons qu'il y a bien 5 éléments selon la longueur et seulement 1 selon la hauteur et la largeur de la poutre.

b) Calcul :

C'est une poutre de dimension 100x10x10 mm possédant des éléments hexaédriques quadratiques complets à 27 points d'intégrations dont 5 éléments sur sa longueur et 1 élément sur sa largeur et sa hauteur. Le principe pour imposer les conditions limites d'une torsion est le suivant (figure 3):

- Blocage de tous les nœuds de la face droite
- Application du mouvement solide en rotation sur tous les nœuds de la face gauche

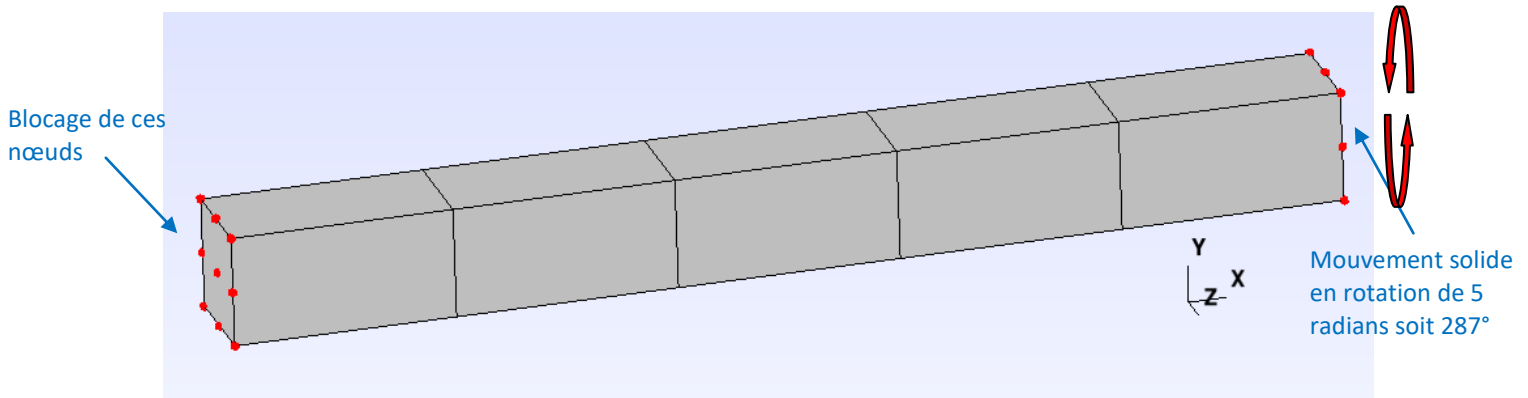


Figure 3: Schéma de principe

Explication de la mise en donnée du fichier `torsion.info` :

```
#-----
# definition de la dimension de l'espace de travail |
#-----
dimension      3                                ← Dimension de l'espace de
                                                    travail
#-----
# definition facultative du niveau d'impression (entre 0 et 10) |
#-----
niveau_commentaire  3                          ← Niveau de commentaire
                                                    variable
```

```

#-----
# definition du type de calcul |
#-----
TYPE_DE_CALCUL
non_dynamique #avec plus visualisation
#-----
#| definition du (ou des) maillage(s) |
#-----

# -- def maillage
< poutre.her

#=====
#| definition des lois de comportement|
#-----

          choix_materiaux -----
#-----
# Elements |   Nom   Materiau   |
#-----
E_tout acier
          matériaux #-----

#-----
# Nom Materiau |   Type loi   |
#-----
acier          ISOELAS
# ..... loi de comportement isoelastique 3D .....
# module d'young :   coefficient de poisson
210000 0.3

#----- fin def des lois de comportement -
#-----

# --- divers stockages (1) -----
masse_volumique #-----#
E_tout 1
          charges #-----#
          blocages #-----#

#-----
# nom du maillage | Ref noeud | Blocages
#-----
N_droit UX UY UZ
N_gauche mouvement_solide_
          centre_ = 100 5 5
          rotation_ = 287 0 0 en_degre_
          fin_mouvement_solide_

          controle #-----
#-----
# PARAMETRE | VALEUR |
#-----
SAUEGARDE 1
DELTAt 0.1
TEMPSFIN 1

```

Type de calcul ici non dynamique

avec plus visualisation :
⇒ permet de créer le fichier de sortie permettant la visualisation des résultats (fichier.Cvisu)

Insertion du maillage

Définition du nom du matériau

Utilisation d'une loi Isoélastique 3D

Ajout de la masse volumique de la poutre

Blocage des nœuds de la face droite

Mouvement solide :
⇒ Définition du centre de la rotation ici centre de la poutre de coordonnée (100 5 5)
⇒ Rotation de 287° par rapport au centre de rotation

Delta t : défini le pas de temps

Défini la fin du calcul.
NB : Il y aura 10 incréments de charge

```

DELTA $\tau$ MAXI 0.1 ← Permet de borner le pas de temps.
                    Ainsi celui-ci ne peut pas augmenter

    para_affichage #-----
#-----
# PARAMETRE      | VALEUR      |
#-----
FREQUENCE_SORTIE_FIL_DU_CALCUL 1

# -----
#-----
    resultats  pas_de_sortie_finale_
    COPIE 0

#

    _fin_point_info_

```

En sortie du fichier.info avec le paramètre « avec plus visualisation » on récupère la déformée de la poutre. Ainsi sous gmsh nous pourrions observer sa déformée.

c) Résultat :

Nous allons visualiser le résultat obtenu grâce à la mise en donnée précédente. Pour cela il suffit d'ouvrir le fichier « torsion_deplace_Gmsh.pos » avec gmsh. Puis faire quelques modification dans les options comme ci-contre (figure 4):

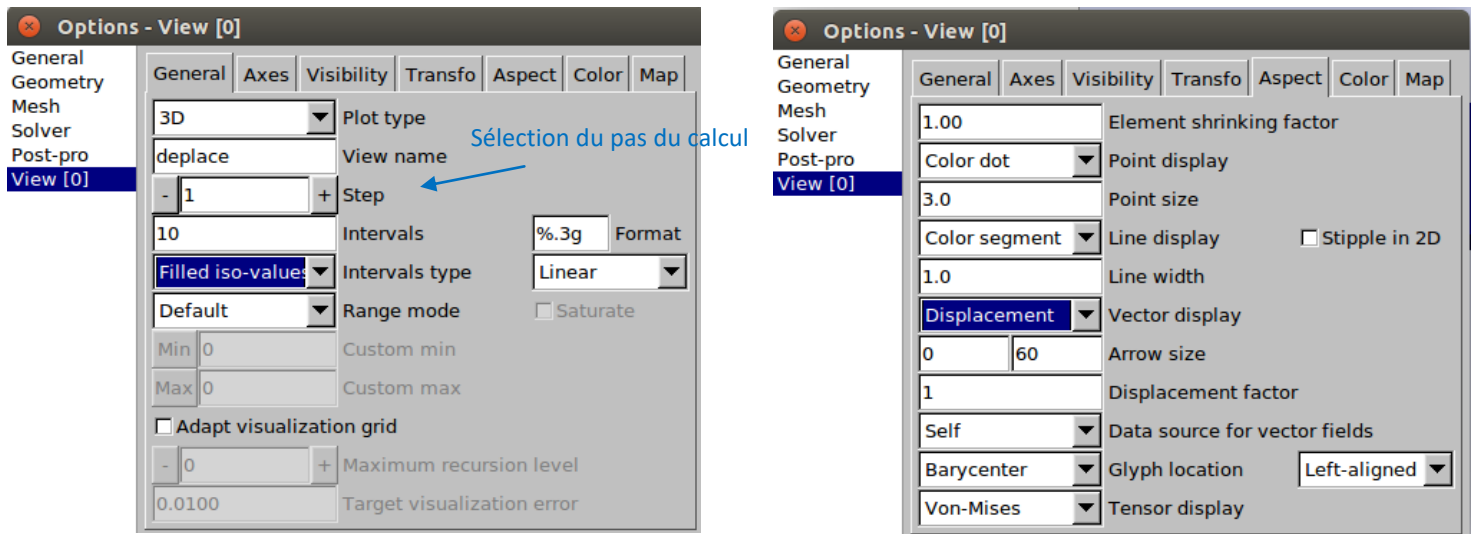


Figure 4: Réglages options

Ces modifications permet de visualiser notre déformée comme suit (figure 5):

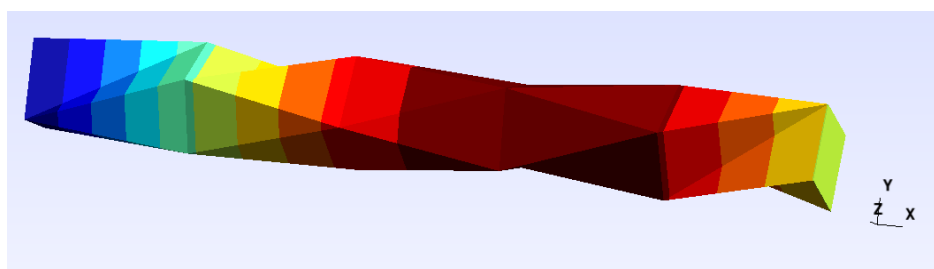


Figure 5: Poutre en torsion

Sur ce maillage il est difficile d'observer un résultat convenable. Cela provient du fait que Gmsh, par défaut, n'utilise que les nœuds sommets pour la visualisation. Alors que dans notre cas, il s'agit d'éléments quadratiques qui possèdent des nœuds intermédiaires. Pour y remédier nous allons affiner au niveau de la visualisation ce maillage puis observer le nouveau résultat comme ci-contre :

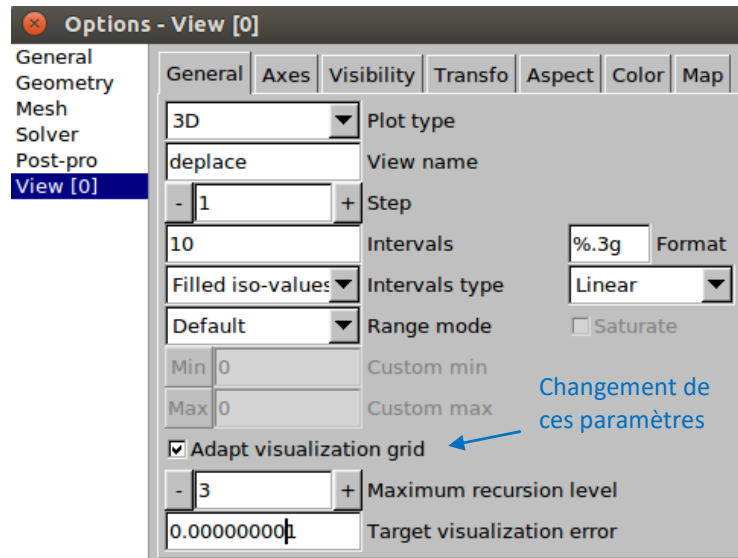


Figure 6: Réglage option

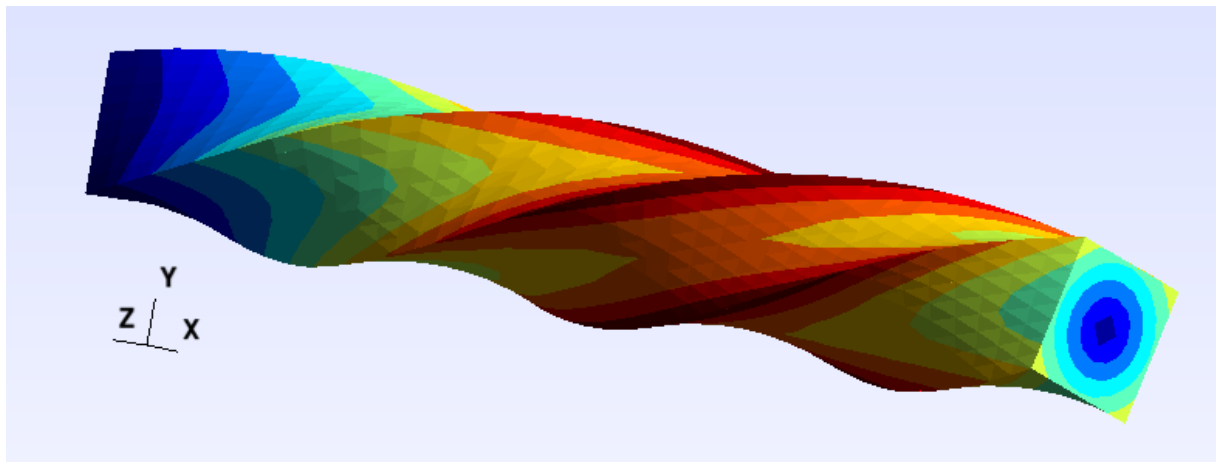


Figure 7: Poutre en torsion

Avec le maillage de la figure 7 le résultat obtenu est bien plus réaliste. On remarque que la poutre est harmonieuse est bien soumise à une torsion.

Cette option nous permet simplement de visualiser un nouveau maillage. Il faut savoir qu'il en déduit par une interpolation linéaire, quadratique ou cubique (dépend du choix de l'option) donc calcul la nouvelle position des nœuds. Ce n'est pas un calcul mécanique comme lorsque nous l'avons fait précédemment avec stamm et Herezh car le maillage reste le même.

III) Affinage du maillage :

Précédemment nous avons vu comment affiner le maillage. Nous allons alors vérifier que le résultat obtenu est bien le même qu'avait prédit gmsh. Pour cela il faut déterminer le nombre d'éléments de cette nouvelle poutre. Il faut régler l'option suivante :

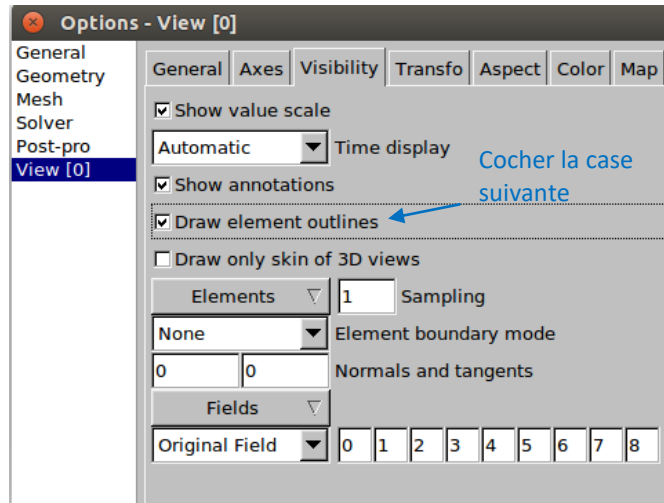


Figure 8: Réglage option

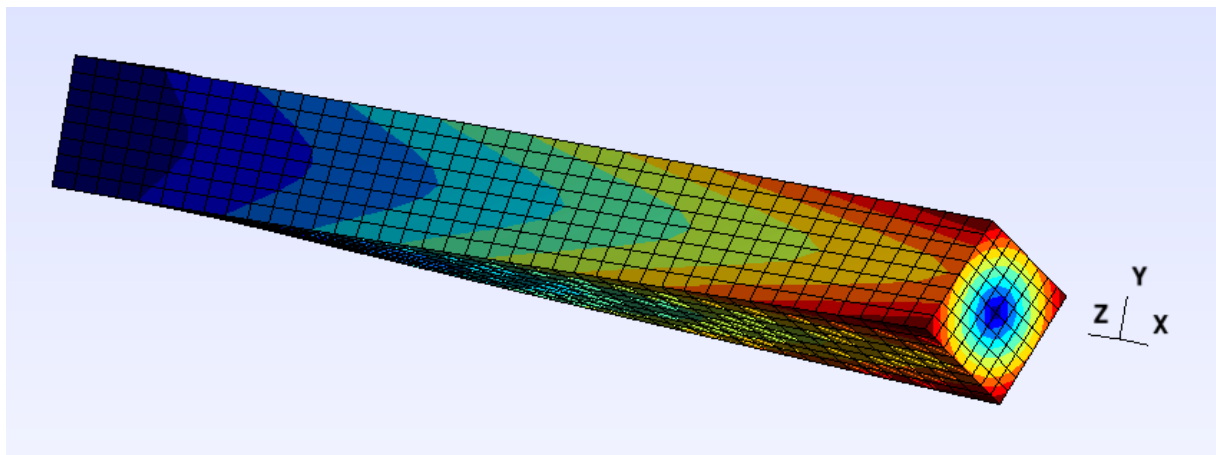


Figure 9: Visualisation du maillage

Il est dorénavant possible de déterminer le nombre d'élément de ce maillage prédit. On constate que celui-ci possède 40 éléments selon sa longueur et 7 éléments selon sa largeur et sa hauteur.

Créons ainsi ce nouveau maillage via stamm (figure 10) donc affinons ce maillage:

```
type d'elements : (1D, 2D, 3D ) ? 3D

choix lu: 3D
type de decoupage :
brique                (reponse b ) ? b

choix de l'interpolation :
lineaire              (reponse li ) ?
quadratique complet  (reponse qc ) ? li

calcul de la position des noeuds :
  de maniere exacte                        (reponse e) ?
  de maniere aleatoire autour d'une position exacte (reponse a) ?
  aleatoire uniquement sur les noeuds internes (reponse i) ? e

type de geometrie :
prisme rectangulaire          (reponse pr ) ?
cylindre creux                (reponse cyl ) ?
parallelogramme eleve        (reponse prl ) ?
portion de cylindre          (reponse pcy ) ?
plaque en helice             (reponse phe ) ?
portion de dome hemispherique (reponse p_dome ) ?
anneau                       (reponse ann ) ?
cylindre plein               (reponse cylp ) ?
dome hemispherique complet   (reponse dome ) ? pr

dimension du prisme rectangulaire: longueur x ? 100
largeur ? 10
hauteur ? 10

nombre d'element(s) dans la longueur ? 40
dans la largeur ? 7
dans la hauteur ? 7

nombre de point d'interpolation standard 8 ? (rep o ou n) n
nombre de pt possible : 8 27 64 : choix ? 27

nom du fichier de sortie ? : poutre2

nom lue : poutre2
un autre maillage ? (rep o ou n ) n

=====
|                        fin stamm                        |
=====
```

Figure 10: Création de la poutre sur stamm

C'est exactement le même principe vu à la figure 1 à part que ici nous changeons le nombre d'éléments.

Ensuite nous gardons le même fichier.info en faisant attention de bien avoir introduit ce nouveau maillage puis l'exécuter. On peut ainsi visualiser le résultat suivant sous gmsh :

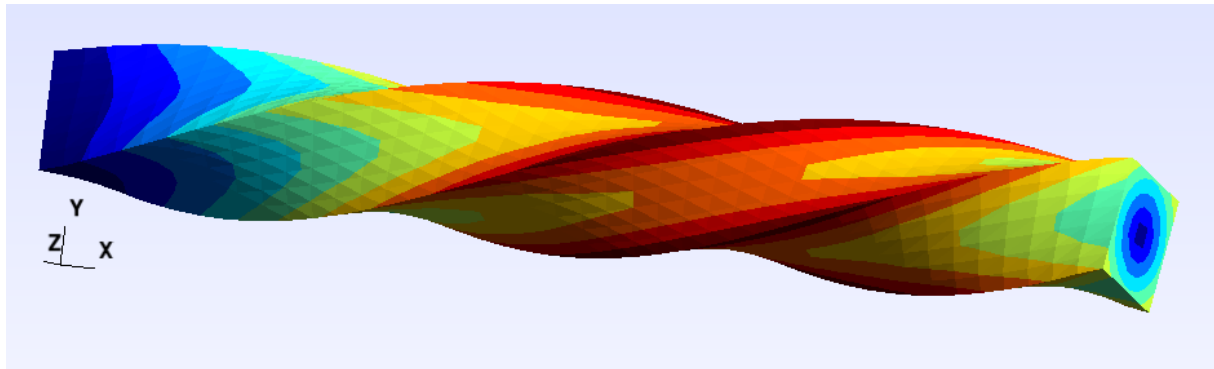


Figure 11: Poutre en torsion optimisée

On remarque que nous obtenons bien le même résultat qu'à la figure 7 sous gmsh.

IV) Conclusion :

Il vous est dorénavant possible de créer une torsion sur une pièce quelconque. Mais aussi de prédire un maillage sous gmsh pour ainsi l'affiner. L'utilité est d'avoir un maillage grossier lors d'un premier calcul pour ainsi vérifier que la mise en donnée est correcte. Puis grâce à gmsh de déterminer le nombre d'élément utile pour obtenir des résultats exploitables et cohérent. Pour finir par affinage du maillage de base puis par un calcul mécanique d'observer le résultat grâce à ce nouveau maillage optimisé.

Dossier : mise_en_donnees_1_torsion_cinematique

- Poutre.her (maillage non affiné)
- Poutre2.her (maillage affiné)
- Poutre.info (mise en donnée du calcul)
- Poutre.Cvisu (fichier permettant de visionner les résultats sous gmsh)