

Tutoriel Herezh++ : Simulation d'une poutre en flexion

Objectif :

Dans ce tutoriel, on décrit pas à pas la simulation du comportement mécanique statique d'une poutre de section rectangulaire constante, sur deux appuis, soumise à une charge verticale répartie.

On commence par construire un maillage adéquate. On se place dans une fenêtre de terminal et on invoque le programme stamm

NB : sur linux le nom du programme est « stamm », sur osX le nom est « stammD »

```
#####  
#                                                                 #  
#                               STAMM                             #  
#           (maillage automatique de pieces simples)           #  
#####  
# Copyright © 1997-2013 Gerard Rio,   gerard.rio@univ-ubs.fr   #  
# All rights reserved.                http://www-lg2m.univ-ubs.fr/~rio #  
#                                                                 #  
#####  
version 03.29 , 22 janvier 2013, compatible avec la version Herezh++6.576
```

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

Ensuite on répond pas à pas aux questions posées

on veut des volumiques donc de dimension 3

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

on veut des éléments hexaédriques (des briques)

choix lu: 3D

type de decoupage :

brique (reponse b) ? b

on choisit une interpolation quadratique pour obtenir une bonne précision d'interpolation

choix de l'interpolation :

lineaire (reponse li) ?

quadratique complet (reponse qc) ? qc

on veut une description exacte de la géométrie, c'est-à-dire que les 3 dimensions de la poutre: longueur . largeur . hauteur, seront exactement représentées par les maillages

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

on veut une répartition régulière des noeuds sur la géométrie

```
maillage regulier (rep = 1) ou non (rep = 0) 1
```

La géométrie retenue est de type prismatique

```
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
```

On retient les dimensions : 200 mm, 20mm, 30mm

NB : en fait il n'y a pas d'unité, c'est à l'utilisateur de gérer la cohérence des différentes unités (mm, Mpa, N etc.) Ce point est à retenir pour la mise en données du calcul.

```
dimension du prisme rectangulaire: longueur x ? 200  
largeur ? 20  
hauteur ? 30
```

On choisit un découpage : 10 x 2 x 2

```
nombre d'element(s) dans la longueur ? 10  
dans la largeur ? 2  
dans la hauteur ? 2
```

Par défaut dans Herezh, les éléments hexaédriques quadratiques utilisent 8 points d'intégration. Pour notre calcul il est nécessaire d'utiliser une intégration à 27 points d'intégration.

NB : Il est possible d'utiliser 8 points d'intégration avec une technique particulière : le blocage de mode d'hourglass, cette technique ne sera pas utilisée dans ce tutoriel.

```
nombre de point d'interpolation standard 8 ? (rep o ou n) n
```

```
nombre de pt possible : 8 27 64 : choix ? 27
```

Et pour finir, on indique le nom de sauvegarde du fichier texte qui contiendra le maillage généré

```
nom du fichier de sortie ? : poutre
```

```
nom lue : poutre
```

```
un autre maillage ? (rep o ou n ) n
```

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

Le logiciel génère alors le fichier : poutre.her

On peut éditer le fichier poutre.her (avec n'importe quel type d'éditeur : par exemple gedit ou nedit sur un système linux).

Le fichier poutre.her, contient :

- une entête qui rappelle les données géométriques, et les choix de construction du maillage

```
#####  
#                                                                 #  
#                               STAMM                             #  
#          (maillage automatique de pieces simples)             #  
#####  
# Copyright © 1997-2013 Gerard Rio,   gerard.rio@univ-ubs.fr    #  
# All rights reserved.             http://www-lg2m.univ-ubs.fr/~rio #  
#                                                                 #  
#####  
# version 03.29 , compatible avec la version Herezh++6.576  
  
# -----  
# | construction automatique d'un maillage    3D                |  
# | pour une utilisation par herezh++        |  
# -----  
# prisme de dimension : 200 x 20 x 30  ( 10 x 2 x 2 elements )  
# geometrie prismatique,   decoupage en hexaedre(s),   interpolation quadratique complete.
```

- la définition d'un nom de maillage, ici « poutre » :

```
nom_maillage  poutre
```

- la liste des coordonnées des noeuds

```

noeuds -----
525  NOEUDS

#-----#
#|NO DU|      X      |      Y      |      Z      |
#|NOEUD|      |      |      |
#-----#
1          0          0          0
2          0          0          7.5
3          0          0          15
4          0          0          22.5

```

- puis des noms de référence (groupes) de nœuds, par exemple ici le premier est « N_avant », chaque nom doit-être préfixé par « N_ » pour indiquer que c'est une référence de noeuds

references des noeuds

N_avant	21	22	23	24	25	46	47	48	49	50	71	72	73	74
97	98	99	100	121	122	123	124	125	146	147	148	149	150	171
173	174	175	196	197	198	199	200	221	222	223	224	225	246	247
249	250	271	272	273	274	275	296	297	298	299	300	321	322	323
325	346	347	348	349	350	371	372	373	374	375	396	397	398	399
421	422	423	424	425	446	447	448	449	450	471	472	473	474	475
497	498	499	500	521	522	523	524	525						
N_arriere	1	2	3	4	5	26	27	28	29	30	51	52	53	54
77	78	79	80	101	102	103	104	105	126	127	128	129	130	151
153	154	155	176	177	178	179	180	201	202	203	204	205	226	227
229	230	251	252	253	254	255	276	277	278	279	280	301	302	303
305	326	327	328	329	330	351	352	353	354	355	376	377	378	379
401	402	403	404	405	426	427	428	429	430	451	452	453	454	455
477	478	479	480	501	502	503	504	505						
N_haut	5	10	15	20	25	30	35	40	45	50	55	60	65	70
85	90	95	100	105	110	115	120	125	130	135	140	145	150	155
165	170	175	180	185	190	195	200	205	210	215	220	225	230	235
245	250	255	260	265	270	275	280	285	290	295	300	305	310	315
325	330	335	340	345	350	355	360	365	370	375	380	385	390	395
405	410	415	420	425	430	435	440	445	450	455	460	465	470	475

- la liste des éléments. Chaque élément est défini par un type, ici « HEXAEDRE » QUADRACOMPL » suivant optionnellement du nombre de point d'intégration, puis on trouve la table de connexion, c'est-à-dire la liste des numéros de nœuds qui définissent l'élément

```
# les elements
elements -----
 40 ELEMENTS
#-----
#| NO  |          |          |          |
#|ELTS |  type element |          |          |
#-----
 1  HEXAEDRE QUADRACOMPL _cm27pti  51  61  11  1  53  63  13
 2  HEXAEDRE QUADRACOMPL _cm27pti  53  63  13  3  55  65  15
 3  HEXAEDRE QUADRACOMPL _cm27pti  61  71  21  11  63  73  23
 4  HEXAEDRE QUADRACOMPL _cm27pti  63  73  23  13  65  75  25
 5  HEXAEDRE QUADRACOMPL _cm27pti  101 111  61  51  103 113  63
 6  HEXAEDRE QUADRACOMPL _cm27pti  103 113  63  53  105 115  65
```

- puis des noms de référence (groupes) relatives aux éléments, par exemple ici le premier est « E_avant », chaque nom doit-être préfixé par « E_ » pour indiquer que c'est une référence d'éléments

```
# references des elements
E_avant      3      4      7      8      11      12      15      16      19      20      23
 35      36      39      40
E_derriere   1      2      5      6      9      10      13      14      17      18      21
 33      34      37      38
```

- on peut trouver également des références
 - . de faces
 - . d'arêtes
 - . de points de Gauss

```
F_avant      3 5      4 5      7 5      8 5
A_haut_droit 2 11     4 11
A_haut_gauche 38 9      40 9
```

NB : ici les listes sont toujours composées de deux chiffres : le premier donne le numéro de l'élément, le second donne le numéro local dans l'élément, de la face ou de l'arête ou du point de Gauss.

Maintenant nous allons faire une mise en donnée du problème mécanique. Pour cela, il nous faut construire un fichier qui contiendra les données nécessaires au calcul.

Ce fichier devra avoir l'extension « .info »

Il contient essentiellement une série de mots clés, associés éventuellement à des valeurs numériques. La description exhaustive de ces mots clés se trouvent dans la documentation d'Herezh.

Ici, pour éviter de se tromper dans la syntaxe des mots clés et dans l'organisation du fichier, nous allons utiliser directement Herezh pour générer un fichier .info correct.

Dans une fenêtre terminal on écrit :

Sur macos

```
HZpp_Vn-1 -n flexion_poutre
```

NB : l'appel du programme Herezh++ diffère suivant le type de machine et d'os :

- Pour macos : deux programmes : HZpp_Vn-1 et HZppfast_Vn-1
- Pour linux 32bits : deux programmes : HZpp et HZppfast
- Pour linux 64bits : deux programmes : HZpp64 et HZppfast64

Sur linux 64 bits : HZpp64 -n flexion_poutre

« -n flexion_poutre » indique au programme que l'on veut créer un nouveau fichier
« flexion_poutre.info »

- on donne la dimension de l'espace de travail, ici 3

```
#####  
#  
# | | == = == = | | | | | #  
# | == = == = / | | | | | #  
# | == = == = / | | | | | #  
# | | == = == = | | | | | #  
# | | == = == = | | | | | #  
#####  
# Copyright (c) 1997-2014 Gerard Rio, gerard.rio@univ-ubs.fr #  
# All rights reserved. http://http://kleger.univ-ubs.fr/Herezh/ #  
# Certification IDDN.FR.010.0106078.000.R.P.2006.035.20600 #  
# #  
#####  
# (version avec le plus de verifications pendant le calcul et les I/O )  
version 6.648  
-- initialisation de l'entree des donnees racine du nom de fichier princi  
pal = flexion_poutre  
  
Donner la dimension de l'espace geometrique de travail : 1 2 ou 3 ? 3
```

- on donne le niveau de commentaire qu'utilisera Herezh++ lors de l'affichage de commentaires, pendant le calcul

```
Donner le niveau de commentaire que vous souhaitez : 0 a 10 ? 4
```

- on choisit un type de calcul : dans notre cas on retient un calcul non dynamique (donc qui peut représenté un cas statique ou transitoire)

```
Donner le type principal de calcul que vous souhaitez : ?  
(1) dynamique_implicit (2) dynamique_explicite_zhai  
(3) dynamique_explicite_chung_lee (4) dynamique_explicite_tchamwa  
(5) dynamique_explicite (6) dynamique_Runge_Kutta  
(7) non_dynamique (8) non_dyna_contact  
(9) flamb_lineaire (10) informations  
(11) utilitaires (12) umat_abaqus  
(13) dynamique_explicite_bonelli (14) dynamique_relaxation_dynam  
(15) mixte_statique_dynamique_explicite
```

- on indique qu'à la suite du calcul on désire avoir la possibilité de choisir un type de posttraitement

```
Donner le sous type de calcul que vous souhaitez : ?
(0) (pas de sous type)
(1) avec_remonte (2) remonte
(3) avec_remonte_erreur (4) remonte_erreur
(5) avec plus visualisation (6) visualisation
(7) frontieres (8) lineaire -> quadra incomplet
(9) quadra incomplet -> quadra complet
(10) reloc pt milieu Quadra (11) sauveCommandesVisu
(12) lectureCommandesVisu (13) prevision_visu_sigma
(14) prevision_visu_epsilon (15) prevision_visu_erreur
(16) commandeInteractive (17) sauveMaillagesEnCours
(18) extrusion2D3D (19) creation_de_references
(20) modif_orientation_element (21) creationMaillageSFE
(22) suppression_noeud_non_references (23) renumerotation_des_noeuds
(24) fusion_de_noeuds (25) fusion_elements
(26) fusion_maillages
5
```

- on ne veut pas de paramètre facultatif pour l'algorithme non dynamique

```
# voulez-vous un exemple des para facultatifs l'algo AlgoriNonDyna (rep o ou n) ? n
```

- on obtient alors un menu, qui permet l'accès à l'ensemble des possibilités de mise en données

```
[c] -> para tres classique [-] para assez courants
(0 ou f) (fin)
(1)maillage(s).....[c] (2)courbes .....[-]
(3)loi de comportement..[c] (4)divers stockages 1.....[c]
(5)auto_contact
(6)efforts.....[c] (7)conditions limites cinematiques..[c]
(8)divers stockages 2 (9)initialisation .....[-]
(10)type de chargement (11)parametres de controle.....[c]
(12)fin fichier.....[c] (13 ou ? ) informations
```

A ce stade, on peut remarquer qu'un nouveau fichier « **flexion_poutre.info** » est apparu dans le répertoire de travail, et que ce fichier se remplit, au fur et à mesure des réponses fournies aux questions posées par Herezh++ . ,

- on va indiquer que le maillage est contenu dans le fichier **poutre.her**

Pour cela on choisit l'option (1) du menu principal puis :

- on définit le nom du fichier de maillage

- on ne définit pas de fichier .lis (concerne des définitions supplémentaires de références)

- on ne veut aucun traitement particulier sur « poutre.her »

- aucun autre fichier de maillage

- pas de traitement entre plusieurs maillages

- pas de domaine esclave (concerne le contact)

```
# ----- definition des maillages -----
# def des fichiers de maillage
dans le fichier de commande final, on utilisera de preference une inclusion
d'un fichier qui contiendra le maillage (ou plusieurs fichiers pour plusieurs
s maillages

(0 ou f) (fin)
(1) inclusion de fichier et affinage
(2) fusion des deux precedents maillages
(3) renumerotation de tous les maillages
(4 ou ? ) exemple - informations sur fichier !

1
non du fichier .her (sans l'extension .her) a inclure ? poutre

un fichier .lis a inclure ? (o ou n ) n

(0 ou f) (fin)
(1) mouvement solide
(2) suppression des noeuds non reference
(3) collapse de noeuds proches
(4) collapse d'elements superposes
(5) creation de references frontieres
(6) renumerotation
(7 ou ? ) informations

f

autre fichier ou fin (rep o ou f) f

(0 ou f) (fin)
(1) inclusion de fichier et affinage
(2) fusion des deux precedents maillages
(3) renumerotation de tous les maillages
(4 ou ? ) exemple - informations sur fichier !

f

def d'un nombre de domaine(s) esclave(s) rep: o ou n ? f
```

La partie suivante concerne la définition de la loi de comportement, qui se déroule en deux phases .

1) On associe une référence d'éléments à un nom de loi de matériau (appelé ici nom de loi).

```
# ----- definition des lois de comportements -----
cas d'un seul maillage (par default) : rep o
cas de plusieurs maillages           : rep n o

-- definition des ref d'elements associes a des noms de lois --
(0 ou f) (fin)
(1) def d'une ref d'elements associe a un nom de loi
(2 ou ? ) exemple - informations !
1

nom de la reference d'element ? E_tout
ref lu =E_tout
nom de la loi associee a E_tout ? acier
nom loi lu =acier

-- definition des ref d'elements associes a des noms de lois --
(0 ou f) (fin)
(1) def d'une ref d'elements associe a un nom de loi
(2 ou ? ) exemple - informations !
f
```

- on a un seul maillage

- def d'une association (il peut y en avoir plusieurs)

- nom de la référence (doit commencer par « E_ »)

- nom de la loi de matériau (un nom arbitraire)

- et c'est tout

2) On définit la loi en question (ou les lois s'il y en a plusieurs) .

```

.....
definition des lois de comportement associee au(x) materiau(x) precedemment
definit
(0) fin
(1)          ISOELAS      (2)          ISOELAS1D
(3)          ISOELAS2D_D (4)          ISOELAS2D_C
(5)          ISO_ELAS_ESP01D (6)          ISO_ELAS_ESP03D
(7)          ISO_ELAS_SE1D (8)          ISOHYPER3DFAVIER3
(9)          ISOHYPER3DORGEAS1 (10)          ISOHYPER3DORGEAS2
(11)         ISOHYPERBULK3 (12)          PRANDTL_REUSS
(13)         PRANDTL_REUSS2D_D (14)          PRANDTL_REUSS1D
(15)         TRELOAR      (16)          NEWTON1D
(17)         NEWTON2D_D   (18)          NEWTON3D
(19)         MAXWELL1D    (20)          MAXWELL2D_D
(21)         MAXWELL2D_C (22)          MAXWELL3D
(23)         LOI_ADDITIVE_EN_SIGMA (24)          LOI_DES_MELANGES_EN_SIGMA
(25)         LOI_CONTRAINTES_PLANES (26)          LOI_DEFORMATIONS_PLANES
(27)         HYSTERESIS_1D (28)          HYSTERESIS_3D
(29)         LOI_ISO_THERMO (30)          LOI_DE_TAIT
(31)         MOONEY_RIVLIN_1D (32)          MOONEY_RIVLIN_3D
(33)         HART_SMITH3D (34)          POLY_HYPER3D
(35)         HYPO_ELAS3D (36)          HYPO_ELAS2D_C
(37)         LOI_VIA_UMAT (38)          LOI_COULOMB
(39)         LOI_RIEN1D (40)          LOI_RIEN2D_C
(41)         LOI_RIEN2D_D (42)          LOI_RIEN3D

donnez la loi de comportement pour le materiau: acier (un numero): 1
definition standart (rep o) ou exemples exhaustifs (rep n'importe quoi) ? o
.....

```

- définition de la loi, c'est-à-dire choix du type de modélisation : ici isotrope élastique
- puis on signale que c'est une définition standart

Après la loi de comportement, on doit définir quelques grandeurs qui dépendent du maillage et des informations déjà lues : on choisit (4) dans le menu principal (divers stockages 1)

```
(0 ou f) (fin)
(1)maillage(s).....[c]    (2)courbes .....[-]
(3)loi de comportement..[c] (4)divers stockages 1.....[c]
(5)auto_contact
(6)efforts.....[c]        (7)conditions limites cinematiques..[c]
(8)divers stockages 2      (9)initialisation .....[-]
(10)type de chargement    (11)parametres de controle.....[c]
(12)fin fichier.....[c]   (13 ou ? ) informations
4
```

```
cas d'un seul maillage (par default) : rep o
cas de plusieurs maillages           : rep n o
```

-- definition de : ---

```
(0 ou f) (fin)
(1) epaisseurs
(2) largeurs
(3) sections
(4) masse volumique (obligatoire)
(5) dilatation_thermique
(6) gestion hourglass
(7 ou ? ) informations
4
```

def de la ref associee :

```
nom de la reference ? E_tout
nom lu =E_tout
valeur de: masse_volumique (un reel) ? 1.
valeur lu =1
```

-- definition de : ---

```
(0 ou f) (fin)
(1) epaisseurs
(2) largeurs
(3) sections
(4) masse volumique (obligatoire)
(5) dilatation_thermique
(6) gestion hourglass
(7 ou ? ) informations
f
```

- on indique qu'il n'y a qu'un seul maillage

- dans notre cas, seule la masse volumique doit-être définie. Mais en fait on ne s'en servira pas, on met donc une valeur arbitraire quelconque.

- et c'est tout

```
(0 ou f) (fin)
(1)maillage(s).....[c]      (2)courbes .....[-]
(3)loi de comportement..[c]  (4)divers stockages 1.....[c]
(5)auto_contact
(6)efforts.....[c]          (7)conditions limites cinematiques..[c]
(8)divers stockages 2        (9)initialisation .....[-]
(10)type de chargement       (11)parametres de controle.....[c]
(12)fin fichier.....[c]      (13 ou ? ) informations
```

On continue en définissant les efforts (le chargement)

```
6 ←
cas d'un seul maillage (par default) : rep o
cas de plusieurs maillages           : rep n o ←
```

- un seul maillage

```
(0 ou f) (fin)
(1) ponctuelle                (PONCTUELLE)
(2) volumique                 (VOLUMIQUE)
(3) lineique de direction fixe (LINEIQUE)
(4) lineique suiveuse         (LINEIC_SUIVEUSE)
(5) surfacique de direction fixe (UNIFORME)
(6) pression                  (PRESSION)
(7) surfacique suiveuse       (PRESSDIR)
(8) hydrostatique             (PHYDRO)
(9) aero - hydrodynamique     (P_HYDRODYNA)
(10 ou ? ) informations
```

- on souhaite imposer une charge surfacique, uniforme, de module 100 en négatif sur l'axe 3, sur tout le haut de la poutre. La référence des faces supérieures de la poutre (F_haut) a été généré automatiquement par stamm lors de la création du fichier : poutre.her

```
5 ←
nom de la reference surfacique ? F_haut ←
nom lu =F_haut
donner la force surfacique (3 reels) ? 0. 0. -100. ←
```

```
(0 ou f) (fin)
(1) utilisation d'une courbe de charge
(2) echelle
(3) temps mini
(4) temps maxi
f
```

```
(0 ou f) (fin)
(1) ponctuelle                (PONCTUELLE)
(2) volumique                 (VOLUMIQUE)
(3) lineique de direction fixe (LINEIQUE)
(4) lineique suiveuse         (LINEIC_SUIVEUSE)
(5) surfacique de direction fixe (UNIFORME)
(6) pression                  (PRESSION)
(7) surfacique suiveuse       (PRESSDIR)
(8) hydrostatique             (PHYDRO)
(9) aero - hydrodynamique     (P_HYDRODYNA)
(10 ou ? ) informations
f ←
```

- et c'est tout

On définit maintenant les conditions limites cinématiques, il y en a plusieurs. On souhaite avoir des conditions quasi-isostatiques.

```
# ----- definition des conditions limites cinematiques -----  
cas d'un seul maillage (par default) : rep o  
cas de plusieurs maillages          : rep n o
```

- on a un seul maillage

```
(0 ou f) (fin)  
(1) def d'une nouvelle condition  
1
```

- on définit une première condition

```
(0 ou f) (fin)  
(1) choix d'un déplacement impose  
(2) ou autre ddl impose  
(9) condition de tangence  
(10) mouvement solide impose  
(11) champ de valeurs  
(12) liste de ddl existant  
(13 ou ? ) information  
(14) plusieurs ddl de déplacement, nuls
```

- elle sera de type : déplacement bloqué

```
rep : 14  
grandeur lue 14  
-- choix de ddl --  
(0 ou f) (fin)  
(1) UX  
(2) UY  
(3) UZ
```

```
3  
valeur lue 3  
-- choix de ddl --  
(0 ou f) (fin)  
(1) UX  
(2) UY  
(3) UZ
```

- suivant z
uniquement

```
f  
valeur lue f  
nom de la référence de nœuds associée? N_bas_gauche  
nom lu =N_bas_gauche
```

- il s'agit des nœuds qui se trouvent
en bas de la face gauche de la
poutre (cette référence a été
générée automatiquement par
stamm)

```
(0 ou f) (fin)  
(4) ou utilisation d'une courbe de charge  
(5) échelle  
(6) temps mini  
(7) temps maxi  
(8) blocage relatif  
(13 ou ? ) information  
rep : f
```

- et c'est tout pour cette condition

```
grandeur lue f  
(0 ou f) (fin)  
(1) def d'une nouvelle condition
```

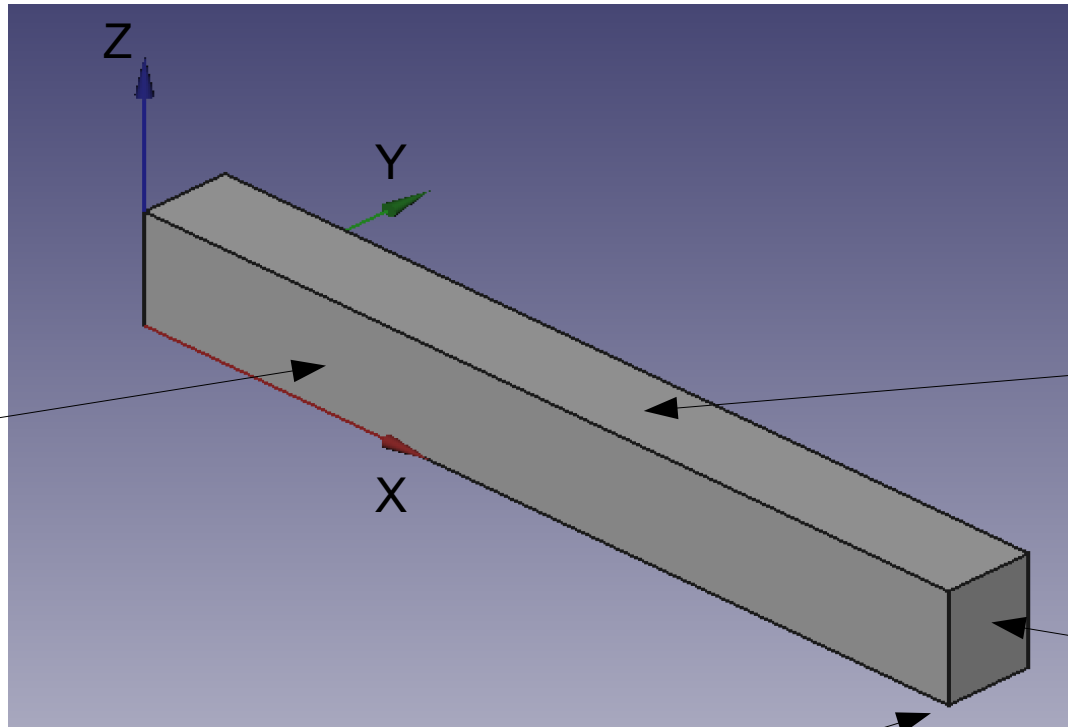

D'une manière analogue, en suivant le même cheminement on définit :

- une condition de blocage selon la direction y pour tous les nœuds de la face avant de la poutre dont la référence est « N_avant ».

Cette condition revient à imposer une condition de symétrie suivant le plan xoz (1 et 3), qui passe par la face avant de la poutre.

- une condition de blocage selon la direction x pour le nœud qui est à l'intersection de la face inférieure de la poutre, de la face arrière et de la face gauche . Sa référence est « N_bas_arriere_gauche ».

Cette condition permet de bloquer le mouvement solide suivant x (direction 1)



L'observateur est sur le plan xoy , et sur les y positifs, il voit donc la face « avant » devant lui !

« arriere »
(la face opposé est : « devant »)

« dessus »
(la face opposée est : « dessous »)

« gauche »
(la face opposée est : « droit »)

« N_bas_arriere_gauche »

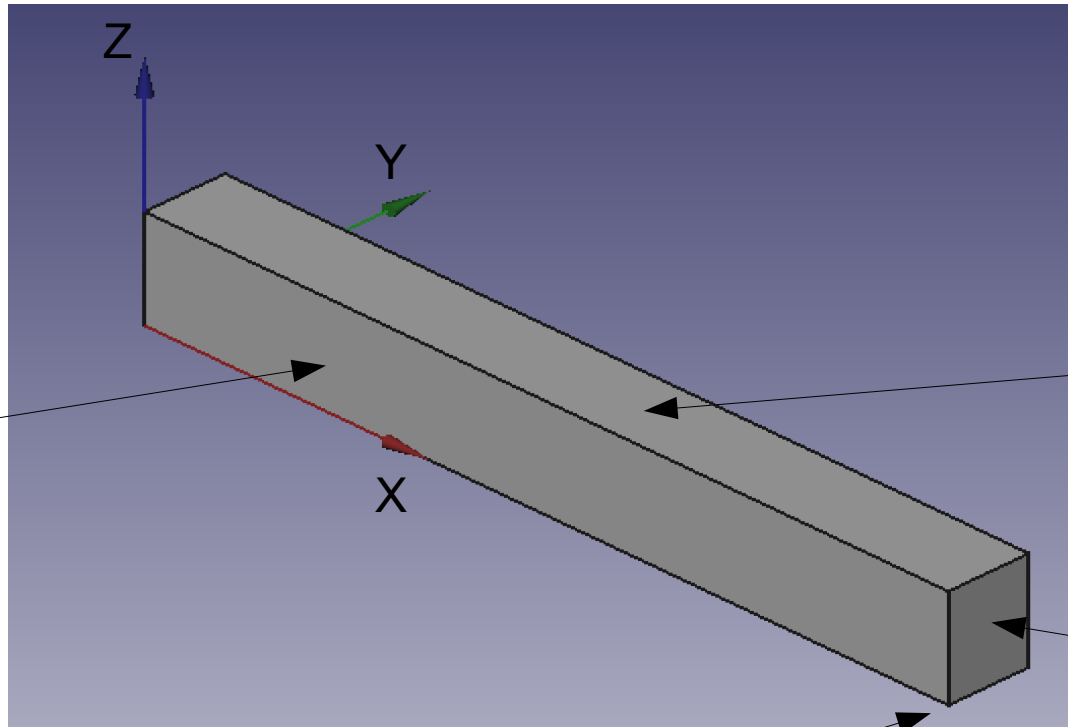
D'une manière analogue, en suivant le même cheminement on définit :

- une condition de blocage selon la direction y pour tous les nœuds de la face avant de la poutre dont la référence est « N_avant ».

Cette condition revient à imposer une condition de symétrie suivant le plan xoz (1 et 3), qui passe par la face avant de la poutre.

- une condition de blocage selon la direction x pour le nœud qui est à l'intersection de la face inférieure de la poutre, de la face arrière et de la face gauche . Sa référence est « N_bas_arriere_gauche ».

Cette condition permet de bloquer le mouvement solide suivant x (direction 1)



L'observateur est sur le plan xoy , et sur les y positifs, il voit donc la face « avant » devant lui !

« dessus »
(la face opposée est :
(dessous »)

« gauche »
(la face opposée est : « droit »)

« arriere »
(la face opposée est :
« devant »)

« N_bas_arriere_gauche »

Nous allons maintenant définir un paramètre de pilotage

```
(12)fin fichier.....[c]      (13 ou ? ) informations
11 ←
    --- definition des parametres de controle -----
choix du groupe de controle :
[c] -> para tres classique [-] para assez courants
(0 ou f) (fin)
(1) controles generaux pour la resolution de l'equilibre generale
(2) parametres lies au systeme d'equation lineaire
(3) parametres lies à la dynamique
(4) parametres lies à l'affichage des resultats [c]
(5) parametres lies au contact
(6) parametres lies aux energies
(7) parametres lies aux calculs geometriques sur les elements
4 ←
    --- affichage -----
(0 ou f) (fin)
    --- parametres de controle de l'algorithmes et para par default
    [c] -> para tres classique [-] para assez courants
(1) frequence d'affichage sur les increment = 1          [-]
    (FREQUENCE_AFFICHAGE_INCREMENT)
(2) frequence d'affichage sur les iteration = 1          [-]
    (FREQUENCE_AFFICHAGE_ITERATION)
(3) frequence de la sortie au fil du calcul = -10000000 [c]
    (FREQUENCE_SORTIE_FIL_DU_CALCUL)
(4) cas de sortie fil calcul
    (FREQUENCE_AFFICHAGE_INCREMENT)
(5) NB chiffre significatifs pour archive et ecran = 17
    (NB_CHIFFRE_POUR_DOUBLE_CALCUL)
(6) NB chiffre significatifs pour graphique = 12
    (NB_CHIFFRE_POUR_DOUBLE_GRAPHIQUE)
3 ←
nouvelle valeur (un reel) ? 1 ←
valeur lue =1
```

- choix de la définition des paramètres de pilotage

- choix des paramètre liés à l'affichage

- choix de la fréquence de sortie au fil du calcul. Ici en indiquant « 1 », on souhaite une sortie des résultats après chaque incrément de chargement. Dans notre cas il y aura un seul incrément, on obtiendra donc un seul résultat, celui final.

Puis deux fois « f »
pour revenir au
menu principal

Première phase de calcul :

On relance Herezh++, avec l'ordre :

```
HZppD_Vn-1 -f flexion_poutre
```

C'est le même ordre que précédemment sauf que «-n » est remplacé par «-f »
Après validation (return), le calcul tourne et à la fin on obtient le menu texte suivant :

```
.....max du Residu total.....0.00050148
**** Norme Residu/Reaction --> 0.00050148

erreur en ouverture pour la lecture du fichier flexion_poutre.CVisu
UtilLecture::Ouverture_CommandeVisu()

**** warning: erreur en ecriture pour la visualisation au fil du calcul
Algori::VisuAuFilDuCalcul(..

>>>> temps fin atteint <<<<<
```

Parmi toutes les informations qui sont écrites, on remarque
1) une erreur
2) et un warning
Cela provient du fait qu'aucune sortie de résultat n'est pour l'instant définie.

L'étape qui suit est donc consacrée a préciser les types de sortie que l'on souhaite

```
>>>> temps fin atteint <<<<<
===== choix du module de visualisation interactive =====
sauvegarde des commandes de visualisation          ? (rep 1)
visualisation automatique                          ? (rep 2)
visualisation au format vrml ?                    (rep 3)
visualisation par fichier de points, format maple ? (rep 4)
visualisation au format geomview                  ? (rep 5)
visualisation au format Gid                        ? (rep 6)
changement de fichier de commande .CVisu         ? (rep 7)
visualisation au format Gmsh                      ? (rep 8)
fin                                                (rep 0 ou f)
reponse ? █
```

On choisit 8, pour une sortie graphique au format gmsh

L'objectif ici est de définir les informations que l'on veut visualiser.
Aussi successivement on fait :

```
===== module de visualisation format Gmsh =====
```

```
=== choix des increments utilises pour l'initialisation de la visualisation  
option par default : tous les increments                (rep 1)  
choix d'un nombre plus petit d'increment                (rep 2)  
reponse ? 1
```

Choix 1 :
Le logiciel utilise tous les
incrémentes pour proposer
les informations
disponibles

```
.....maillage initiale:                mi  
.....isovaleurs:                       iso  
.....deformee:                         de  
.....choix numeros d'increment:         cni  
.....choix du ou des maillages a visualiser: cmv  
.....visualisation :                    visu  
.....arret de la visualisation interactive: f  
reponse ? mi  
  
----> preparation de la visualisation des coordonnees initiales  
parametre par default ? : resultat brut du calcul elements finis ,  
pas d'homotheties sur les coordonnees initiales ,  
sortie des references dans un seul fichier,  
---> (rep 'o') pour accepter ces parametres sinon autre  
reponse ? o
```

On veut le maillage
initial
On choisit les
paramètres par
défaut

Ensuite on choisit
les isovaleurs

```
reponse ? iso
```

```
---- isovaleurs ----
```

```
(0 ou f ou fin) fin modif
```

```
(1) ou (de) parametres par default:
```

```
(2) ou (to) toutes les isovaleurs
```

```
(3) parametres generaux pour la sortie
```

```
(4) ddl naturellement defini aux noeuds
```

```
(5) grandeurs scalaires venant des pts integ
```

```
(6) grandeurs tensoriel venant des pts integ
```

```
(7) grandeurs particulieres venant des pts integ
```

```
(8) choix de l'ancien format gms (faux par default)
```

```
(9) ddl etendu aux noeuds
```

```
(10) grandeurs evoluees aux noeuds
```

```
reponse ? 5
```

```
garithmique11 logarithmique22 logarithmique33 loge  
1 Cauchy_local22 Cauchy_local33 Cauchy_local12 C  
22 Almansi_local33 Almansi_local12 Almansi_local23  
incipaleIII Sigma_principaleI Sigma_principaleII S  
rincipaleIII Delta_def11 Delta_def22 Delta_def33  
Cos3phi_eps Spherique_sig Q_sig Cos3phi_sig cont  
lente def_duale_mises_maxi vitesse_def_equivalente  
euse
```

```
Maillage nb: 1 liste des types de ddl enregistres
```

```
debut_List_IO= (taille= 1 ) contrainte_mises
```

```
donner le ddl a visulaliser
```

```
(to) tous les ddl
```

```
(une liste de ddl)
```

```
(ef) pour effacer la liste
```

```
(ef1) effacer un ddl de la liste
```

```
(0 ou f ou fin) fin choix ddl
```

```
reponse ? f
```

On choisit ensuite 5
Correspond par exemple aux
contraintes et déformations qui sont
calculées aux points d'intégration

```
.....deformee: de  
.....choix numeros d'increment: cni  
.....choix du ou des maillages a visualiser: cmv  
.....visualisation: visu  
.....arret de la visualisation interactive: f  
reponse ? de
```

Dans le menu principal on
choisit de sortir la déformée

```
----> preparation de la visualisation des deforme  
parametre par default ? : pas de limites d'alerte sur les depl  
s sinon autre  
reponse ? o
```

Avec les paramètres par défaut

```

.....choix du ou des maillages a visualiser:          cmv
.....visualisation :                                  visu
.....arret de la visualisation interactive:           f
reponse ? f

```

On sort du menu relatif à gmsh

```

===== fin du module de visualisation format Gmsh =====

===== choix du module de visualisation interactive =====
sauvegarde des commandes de visualisation             ? (rep 1)
visualisation automatique                             ? (rep 2)
visualisation au format vrml ?                       (rep 3)
visualisation par fichier de points, format maple ?  (rep 4)
visualisation au format geomview                     ? (rep 5)
visualisation au format Gid                           ? (rep 6)
changement de fichier de commande .CVisu            ? (rep 7)
visualisation au format Gmsh                          ? (rep 8)
fin                                                    (rep 0 ou f)
reponse ? 1

===== choix du module de visualisation interactive =====
sauvegarde des commandes de visualisation             ? (rep 1)
visualisation automatique                             ? (rep 2)
visualisation au format vrml ?                       (rep 3)
visualisation par fichier de points, format maple ?  (rep 4)
visualisation au format geomview                     ? (rep 5)
visualisation au format Gid                           ? (rep 6)
changement de fichier de commande .CVisu            ? (rep 7)
visualisation au format Gmsh                          ? (rep 8)
fin                                                    (rep 0 ou f)
reponse ? f

=====
|                               fin HEREZH++                               |
=====

```

On n'oublie pas de sauvegarder les choix que l'on vient d'effectuer

Et on stop Herezh

On remarque que dans le répertoire un nouveau fichier a été créé, il contient « flexion_poutre.CVisu » il contient tous les choix de visualisation.

On relance Herezh++ :

```
HZppD_Vn-1 -f flexion_poutre
```

On remarque qu'il n'y a plus de message d'erreur ni de warning,

```
**** Norme Residu/Reaction --> 0.00050148

===== lecture du fichier de commande de visualisation =====
===== fin de la lecture du fichier de commande de visualisation =====

    >>>> temps fin atteind <<<<<<
===== choix du module de visualisation interactive =====
sauvegarde des commandes de visualisation      ? (rep 1)
visualisation automatique                      ? (rep 2)
visualisation au format vrml ?                 (rep 3)
visualisation par fichier de points, format maple ? (rep 4)
visualisation au format geomview              ? (rep 5)
visualisation au format Gid                   ? (rep 6)
changement de fichier de commande .CVisu     ? (rep 7)
visualisation au format Gmsh                  ? (rep 8)
fin                                             (rep 0 ou f)
reponse ? f ←
```

=====
| fin HEREZH++ |
=====

au contraire on a eu :
la lecture du fichier .Cvisu
avec l'exécution automatique
des sorties demandées

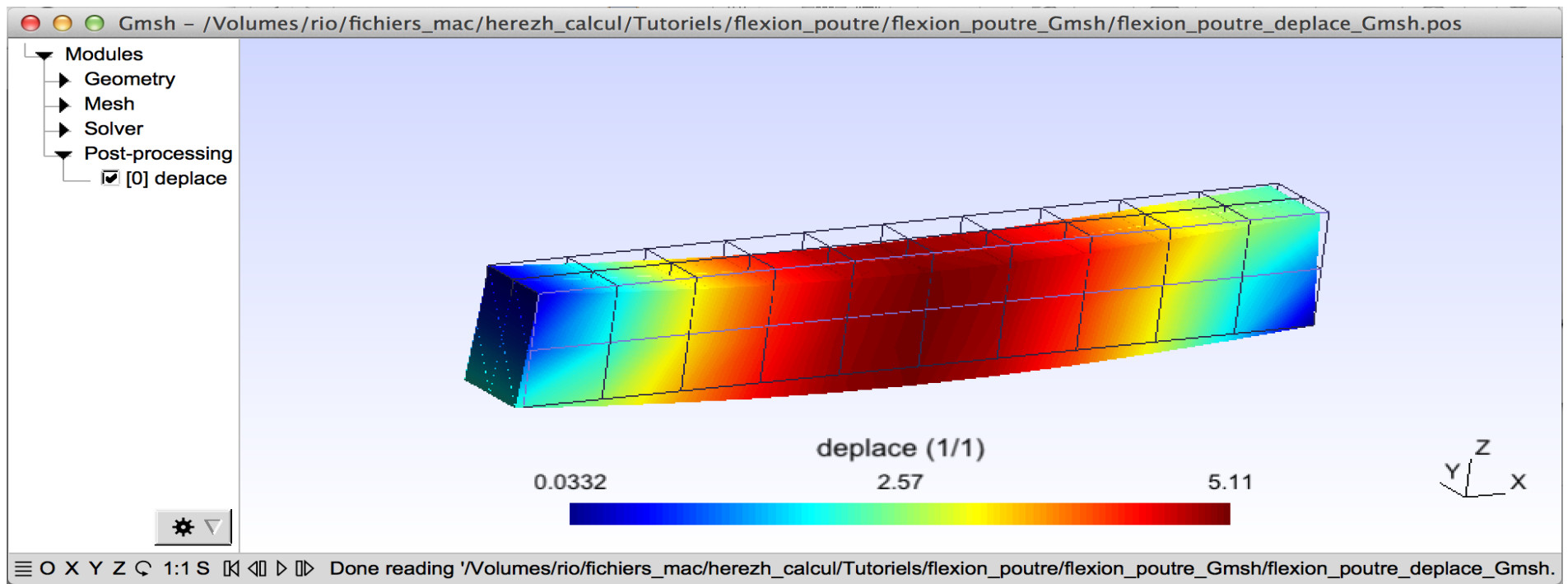
On stop Herezh

Après avoir tapé « **ls** » on remarque un nouveau fichier **flexion_poutre_Gmsh.msh** qui contient le maillage initial avec les références et un nouveau répertoire **flexion_poutre_Gmsh** : qui contient les fichiers de résultats c-a-d ici :

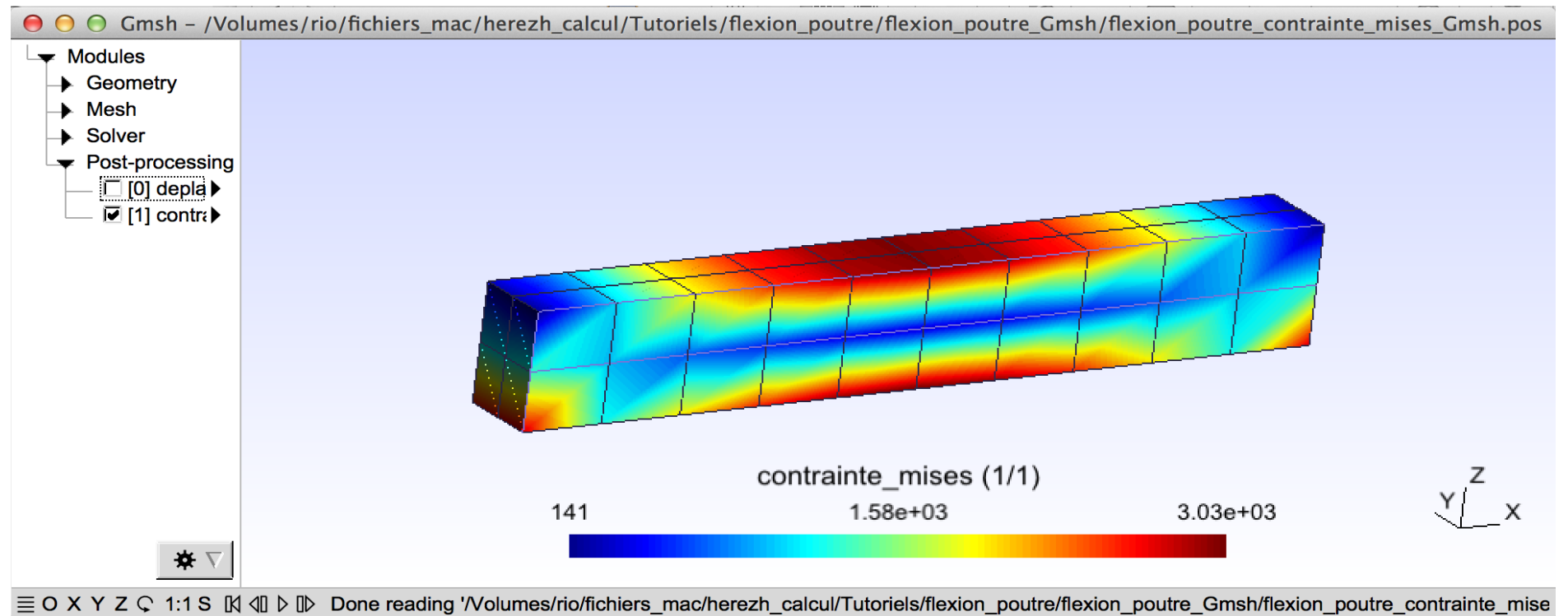
- flexion_poutre_constrainte_mises_Gmsh.pos** : isovaleurs de Mises
- flexion_poutre_deplace_Gmsh.pos** : les déplacements

➡ Il faut utiliser le logiciel **gmsk** pour visualiser ces résultats

La déformée



Les isovaleurs de Mises



Il existe de Nombreux tutoriels pour explorer les différentes possibilités de gmesh
En particulier il est possible de visualiser les contraintes de Mises (ou autres) sur le maillage déformée, ici :

