Manuel d'utilisation du script hz_gmsh2maple.pl

Julien Troufflard

julien.troufflard@free.fr

2025-10-31

Table des matières

1 Résumé

hz_gmsh2maple.pl est un script Perl permettant de convertir les résultats Gmsh en fichier colonne de type .maple. Il exploite un fichier de commande définissant les maillages et les sorties à réaliser à partir d'un ensemble de fichiers Gmsh .pos contenus dans un répertoire Gmsh créé par Herezh++.

Cette notice fournit un aperçu de l'utilisation du script (dont la syntaxe du fichier de commande) ainsi que des exemples.

2 Utilisation du script

hz_gmsh2maple.pl exploite les fichiers .pos contenus dans un répertoire Gmsh créé par Herezh++.

Supposons les éléments suivants :

- un calcul Herezh++ a été réalisé via un fichier .info de nom `mon_calcul.info` et a conduit à créer un répertoire résultat de nom `mon_calcul_Gmsh/`
  IMPORTANT : si le temps n'est pas cohérent dans les `$NodeData` entre tous les fichiers .pos du répertoire Gmsh, le script ne fonctionnera pas normalement.
- un fichier de commande de nom `fic_commandes.cmd` a été écrit (voir syntaxe en section 3)
- on souhaite créer un fichier résultat de nom `fic_resu.maple`

Le script **hz_gmsh2maple.pl** se lance alors dans un terminal de la manière suivante :
> **hz_gmsh2maple.pl** `mon_calcul_Gmsh/ fic_commandes.cmd fic_resu.maple`

A l'issue de l'exécution, le fichier `fic_resu.maple` sera créé. Il contiendra les données sous forme de colonnes sachant que la première colonne sera le temps. L'en-tête fournit les informations nécessaires (rappel des fichiers exploités, des sorties demandées et lien entre numéros de colonne et grandeur pour chaque sortie).

Les options sont listées dans le tableau 1.

option	**description**
-h / -help	affichage de l'aide ex : > **hz_gmsh2maple.pl** -h
-v	affichage du numéro de version du script ex : > **hz_gmsh2maple.pl** -v
-l	affichage des grandeurs disponibles dans le répertoire Gmsh ex : > **hz_gmsh2maple.pl** -l `mon_calcul_Gmsh/`
-trame `nom_fic`	créer un exemple de fichier de commandes de nom `nom_fic` ex : > **hz_gmsh2maple.pl** -trame `fic_commandes.cmd`

TABLEAU 1 – options du script

3 Syntaxe du fichier de commandes

Le fichier de commande est un fichier texte. Il n'y a aucune règle particulière pour nommer ce fichier.

Le fichier est constitué de 2 mots-clés principaux permettant de définir les maillages (`MAILLAGES`) et les sorties souhaitées (`SORTIES`). Il y a également actuellement la liste des fonctionnalités particulières suivantes :

- mot-clé `STAT` : permet de sortir une statistique sur un groupe de noeuds au lieu de sortir les valeurs pour chacun des noeuds du groupe (données actuellement fournies par `STAT` : somme, moyenne, min, max)

La structure générale du fichier de commande est la suivante :

```
MAILLAGES #(mot-cle de declaration des maillages dans l ordre du .CVisu)
chemin/nom_fichier #(fichier .her maillage 1)
chemin/nom_fichier #(fichier .her maillage 2)
...
chemin/nom_fichier #(fichier .her maillage N)


SORTIES #(mot-cle de declaration des grandeurs a sortir)
n°mail_1  ref_noeud_1   n°mail_2  ref_noeud_2 ... #noeuds sortie 1
[STAT ] var_1 var_2 ... #grandeurs sortie 1
n°mail_1  ref_noeud_1   n°mail_2  ref_noeud_2 ... #noeuds sortie 2
[STAT ] var_1 var_2 ... #grandeurs sortie 2
...
n°mail_1  ref_noeud_1   n°mail_2  ref_noeud_2 ... #noeuds sortie N
[STAT ] var_1 var_2 ... #grandeurs sortie N
```

Le symbole # permet d'écrire des commentaires.

Comme indiqué, sous le mot-clé `MAILLAGES`, on énumère les noms de fichier de maillage .her. On entend par "nom de fichier de maillage" le chemin absolu ou relatif pour trouver le fichier .her depuis le répertoire où ce script est lancé.

Pour des raisons de numérotation des noeuds, il est obligatoire de les déclarer dans l'ordre défini dans le fichier .CVisu (en général, c'est le même ordre que dans le fichier .info, mais c'est bien l'ordre du .CVisu qui gouverne). Si cela n'est pas respecté, les sorties pourront être erronées dans le cas à plusieurs maillages.

Sous le mot-clé `SORTIES`, on peut enchainer autant de fois que nécessaire les deux lignes obligatoires pour définir une sortie :

ligne 1 : liste de paires ["numéro du maillage" "référence de noeuds"]

Le "numéro de maillage" correspond à l'ordre d'apparition sous le mot-clé `MAILLAGES`. La "référence de noeuds" est soit un numéro de noeud (entier), soit une liste de référence de noeuds définie dans le fichier .her ou .lis (NB : si la liste N_tout n'existe

pas dans le maillage, une liste N_tout est automatiquement créée et contient tous les noeuds du maillage)

ligne 2 : liste des grandeurs à sortir.

On entend par "grandeur", toute chaine de caractère fournie par l'option -l (voir tableau 1). Le script fournira alors une sortie de chaque grandeur pour chaque noeud.

En début de ligne, on peut indiquer le mot-clé `STAT` (sortie d'une statistique)

Voici l'exemple de fichier de commandes fourni par l'option -trame :

```
MAILLAGES #mot-cle de declaration des fichiers maillage
nom_fichier_1.her
nom_fichier_2.her


SORTIES #mot-cle de declaration des sorties de grandeurs

#sortie des grandeurs EPS11 et SIG11
# pour les noeuds de la liste N_tout du maillage 1
1 N_tout
EPS11 SIG11

#sortie des grandeurs Def_principaleI et Sigma_principaleI
# pour le noeud 129 du maillage 2
2 129
Def_principaleI Sigma_principaleI

#sortie des grandeurs def_duale_mises et contrainte_mises
# pour les noeuds de la liste N_S du maillage 1
1 N_S
def_duale_mises contrainte_mises

#sortie des grandeurs def_duale_mises et contrainte_mises
# statistiques sur les valeurs aux noeuds de la liste N_S du maillage 1
1 N_S
STAT  def_duale_mises contrainte_mises

#sortie de la grandeur deplace
# statistiques sur les valeurs aux noeuds de la liste N_E du maillage 1
# et de la liste N_O du maillage 2
1 N_E  2 N_O
STAT   deplace
```

4 Suivi des versions

- **version 1.00 (2023/02/01)** : version initiale (testée sur MacOSX)
- **version 1.01 (2023/02/02)** : modif majeure de la lecture des "no_maillage ref_noeud" pour chaque sortie. On fait plutôt le choix de lire un ensemble de paires "no_maillage ref_noeud" sur la première ligne, puis la liste des grandeurs (avec ou sans STAT) sur la seconde ligne. Cela permettra de gérer le cas où on veut une sortie sur une zone qui concerne plusieurs maillages a la fois (remaniement du code un peu partout, modification en conséquence de l'aide -h et du fichier exemple option -trame dans sub print_fic_cmd())
- **version 1.02 (2023/02/06)** : modif de la subroutine make_table_grandeurs_Gmsh() pour généraliser la reconnaissance des grandeurs disponibles et de leur lien vers leur fichier .pos. La subroutine get_nom_grandeur_fpos() a été ajoutée pour cela. En gros, désormais, le nom de grandeur associé à un fichier .pos est repéré en lisant son nom dans le premier nodedata du fichier plutôt qu'en le repérant dans le nom du fichier .pos. Le nom des fichiers .pos contient malheureusement différents suffixes qui étaient difficiles à tous lister (_pti_Gmsh.pos, _ddl_noe_Gmsh.pos, _evolue_noe_Gmsh.pos, etc...), sans compter qu'il était impossible de gérer l'apparition d'un nouveau type de suffixe en cas d'évolution des sorties Gmsh Herezh++.

  **A noter que suite à cette modif, la subroutine chaine_commune() ne sert plus. On la garde en stock au cas où.
- **version 1.03 (2023/04/05)** : modification mineure (correction d'un test if erroné conduisant à une erreur possible en cas de temps initial nul dans un fichier .pos)
- **version 1.04 (2025/10/31)** : remplacement du module `List::MoreUtils` par le module `List::Util` pour accéder à la fonction `uniq`. Le module `List::MoreUtils` n'est pas natif. A partir de Perl 5.26, on peut accéder à la fonction `uniq` avec le module natif `List::Util`