Descriptif informatique de CALC_MISS

Résumé :

Ce document décrit l'architecture informatique de la macro-commande CALC_MISS.
La commande se décompose en deux étapes principales que l'on retrouve dans son architecture : une première où on fait appel à MISS3D et une deuxième dans laquelle on effectue un post-traitement des résultats MISS3D avec des commandes de *Code_Aster*.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Modules de CALC_MISS

Le point d'entrée de CALC_MISS est calc_miss_ops (dans calc_miss_ops.py)
Cette méthode est très simple.
Tout d'abord, on stocke les paramètres fournis par l'utilisateur. Il s'agit simplement de stocker les valeurs des mots-clés, de les compléter avec d'autres pour simplifier les traitements futurs et d'y ajouter des vérifications de cohérence. C'est le travail de l'instance MISS_PARAMETER (définie dans miss_utils.py).
Ensuite, en fonction des paramètres, on crée un objet dérivé de CALCUL_MISS (de miss_calcul.py) qui décrit les tâches réalisées par CALC_MISS (*cf.* §2).

Remarque

Le mot-clé « post-traitement » peut prêter à confusion : il s'agit ici d'un « post-traitement » de la résolution faite par MISS3D, c'est donc la finalisation du calcul d'ISS dans Code_Aster*.*

2 Objet de type CALCUL_MISS

C'est la fonction CalculMissFactory qui crée, en fonction de TYPE_RESU, l'objet qui organise le traitement effectué par CALC_MISS. Actuellement, il y a deux types possibles :

- CALCUL_MISS_IMPE : traite l'appel à MISS3D pour le calcul des impédances et enchaîne éventuellement un post-traitement,
- CALCUL_MISS_POST : traite une exécution sans appel à MISS3D, seul le post-traitement est lancé.

Systématiquement, on enchaîne ces 3 tâches :

- prepare_donnees()
- execute()
- post_traitement()

Dans le cas d'un appel à MISS3D, la méthode prepare_donnees a pour rôle de produire les fichiers de données qui seront lus par MISS3D. Ces fichiers sont créés à partir des résultats imprimés par IMPR_MACR_ELEM, qui sont retraités en utilisant les fonctions dédiées des modules miss_fichier_sol (écriture des caractéristiques du sol), miss_fichier_interf (écriture du maillage, des modes statiques et dynamiques, du fichier de commandes MISS), miss_fichier_option (écriture des options du calcul MISS).

Remarque

Aujourd'hui on relit dans CALC_MISS le fichier produit par IMPR_MACR_ELEM. On pourrait optimiser cette étape en produisant, directement dans IMPR_MACR_ELEM, les deux fichiers contenant le maillage de l'interface d'une part et les modes statiques et dynamiques d'autre part.

La méthode execute, comme son nom l'indique, exécute MISS3D en utilisant les fichiers préparés.

La méthode post_traitement récupère les fichiers produits par MISS3D afin de les déposer dans le répertoire d'exécution de *Code_Aster* sous les numéros d'unité logique fournis par l'utilisateur, puis procède au « post-traitement » (*cf.* §3).

Dans le cas du post-traitement seul (CALCUL_MISS_POST), les méthodes prepare_donnees et execute ne font rien, et les fichiers ont été obtenus auparavant et sont fournis par l'utilisateur.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Objet de type POST_MISS

Dans la méthode post_traitement de l'objet CALCUL_MISS, on fait appel à la fonction PostMissFactory pour créé l'objet d'un type dérivé de POST_MISS (dans miss_post.py) en fonction des choix de l'utilisateur.
Les types disponibles sont :

- POST_MISS_HARM : calcule la réponse (déplacement, vitesse, accélération) harmonique de la structure,
- POST_MISS_TRAN : calcule la réponse transitoire,
- POST_MISS_TAB : calcule les réponses harmonique et transitoire, et le spectre d'oscillateur en certains lieux,
- POST_MISS_FICHIER : on ne fait rien (les fichiers issus de MISS3D sont récupérés par l'utilisateur).

La documentation d'utilisation de CALC_MISS [U7.03.12] détaille le type de traitement effectué.

Systématiquement, on enchaîne ces 3 tâches :

- argument()
- execute()
- sortie()

La méthode argument vérifie les données d'entrée, procède aux initialisations et prépare les objets nécessaires aux traitements futurs (comme par exemple, l'interpolation des accélérogrammes fournis sur une liste d'instants commune et à pas constant).

La méthode execute effectue les projections des matrices de masse et rigidité, des seconds membres en cas de chargement harmonique, puis enchaîne les commandes DYNA_LINE_HARM pour chaque fréquence de calcul.

La méthode sortie prépare et produit les concepts de sortie.
POST_MISS_HARM est une spécialisation de POST_MISS_TRAN afin d'arrêter le traitement plus tôt. La méthode sortie de POST_MISS_TRAN appelle par exemple REST_SPEC_TEMP pour calculer la réponse transitoire ; celle de POST_MISS_TAB se contentant de produire la table résultat.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)