Introduire un nouveau type de maille ou un nouvel élément de référence

Résumé :

Ce document décrit ce qu'il faut faire pour introduire un nouveau type de maille dans *Code_Aster* ou un nouvel élément de référence.

En quelques mots, pour ajouter un nouveau type de maille et/ou un nouvel élément de référence, il faut :

- définir et introduire les caractéristiques de la maille et/ou de l'élément de référence dans le source de Code_Aster : position des nœuds, connectivité, fonctions de forme, position des points de Gauss , ...
- penser à l'impact au niveau des fonctionnalités existantes dans Code_Aster ,
- enrichir les documentations «utilisateur» (Doc U) et les documentations «référence» (Doc R) relatives au fichier de maillage, aux fonctions de forme, ...
- enrichir des cas-tests existants afin de tester votre nouvel élément fini.

Un exemple complet sera décrit dans ce document.

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

Table des Matières

1 Introduction....3
2 Introduction d'un nouveau type de maille ou un nouvel élément....3
2.1 Présentation de la maille et de l'élément de référence....4
2.2 Introduction de la maille et de l'élément de référence dans le source de Code_Aster....5
2.2.1 Au niveau des catalogues d'éléments....5
2.2.1.1 Catalogue Commons/mesh_types.py....5
2.2.1.2 Catalogue Commons/phenomenons_modelisations.py....6
2.2.1.3 Catalogue Elements/meca_hexa20.py....6
2.2.2 Au niveau du source fortran....7
2.3 Mise à jour des commandes de pré et post traitement dans Code_Aster....8
2.3.1 Opérateur CREA_MAILLAGE....8
2.3.1.1 Option QUAD_LINE....8
2.3.1.2 Opérateur IMPR_RESU....8
2.3.2 Opérateurs LIRE_RESU, LIRE_CHAMP....9
2.4 Mise à jour des autres opérateurs de Code_Aster....9
2.5 Validation....11
2.6 Documentations....11

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

1 Introduction

Avant de penser à introduire un nouveau type de maille dans Code_Aster, il faut avoir :

- consulté la documentation U3.01.00 («description du fichier de maillage de Code_Aster ») afin de connaître les conventions d'écriture (ordre des nœuds, …),
- consulté la documentation R3.01.01 («fonctions de forme des éléments isoparamétriques») afin de connaître les représentations des éléments de références, leurs fonctions de formes, les familles de points de Gauss existantes, …

Après avoir :

- dimensionné votre élément,
- placé les nœuds,
- déterminé les fonctions de formes et leurs dérivées,
- attribué une ou des familles de points de Gauss,

Vous pouvez passer à l'étape suivante, qui consiste à introduire votre maille ou élément de référence dans Code_Aster.

Cette étape passe par la modification du source de Code_Aster, c'est à dire la modification de :

- catalogues d'éléments,
- fichiers fortran.

Nous décrirons explicitement les interventions dans chacun des fichiers concernés.

Après cette étape, vous pourrez réaliser des calculs sur des modèles acceptant votre nouveau type d'élément. Il est intéressant de pouvoir imprimer, visualiser et/ou relire vos résultats. Il n'est donc pas envisageable de restituer votre développement sans mettre à jour les commandes de pré-traitement et post-traitement. Nous listerons les routines concernées.

A ce stade, il reste à vérifier que l'ensemble des fonctionnalités de Code_Aster soit toujours opérationnel. Nous verrons que le source relatif à d'autres opérateurs aura besoin d'être actualisé. Nous listerons les opérateurs et les routines concernés.

La phase de développement étant finalisée, il reste la phase de validation. Les différents tests à effectuer seront présentés dans ce document.

Enfin, il restera à actualiser les documentations. Nous énumérerons les différentes documentations à enrichir.

Afin de passer en revue ces différentes étapes, nous allons présenter un exemple illustrant l'ajout d'une nouvelle maille et d'un élément de référence.

2 Introduction d'un nouveau type de maille ou un nouvel élément

On propose dans cet exemple le développement d'un élément fini basé sur une nouvelle maille, un pentaèdre à 18 nœuds, dans le domaine de la mécanique non-linéaire.

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

2.1 Présentation de la maille et de l'élément de référence

Un pentaèdre à 18 nœuds est une maille dont les nœuds sont localisés :

- aux sommets (6 nœuds),
- au milieu de chaque arête (9 nœuds),
- au milieu des faces quadrangulaires (3 nœuds).

Une représentation de élément de référence associé à cette maille est la suivante:

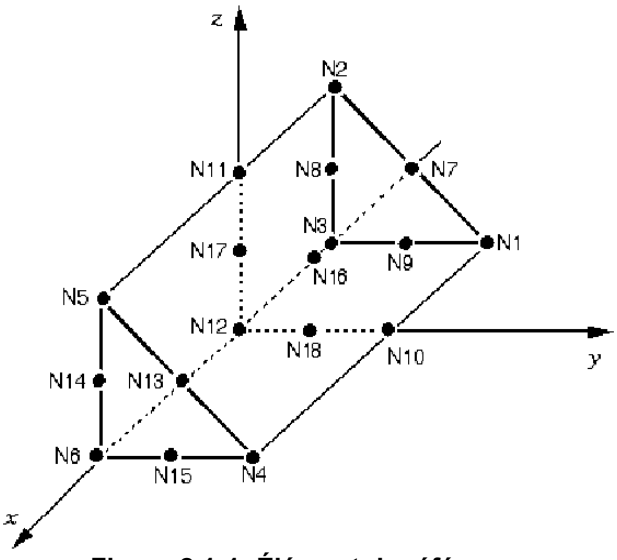


Figure 2.1-1: Élément de référence

Pour réaliser une représentation de cet élément (dimension, position et ordre des nœuds, ...), il est conseillé de consulter la documentation U3.01.00 («description du fichier de maillage de Code_Aster»). Par exemple, une convention sur l'ordre des nœuds y est présentée : on commence par ordonner les nœuds sommets, puis les nœuds milieux (sur les arêtes), et enfin les nœuds centraux (sur les faces).
Les coordonnées des nœuds sont :

	x	y	z
N1	-1.	1.	0.
N2	-1.	0.	1.
N3	-1.	0.	0.
N4	1.	1.	0.
N5	1.	0.	1.
N6	1.	0.	0.
N7	-1.	0.5	0.5.
N8	-1.	0.	0.5.
N9	-1.	0.5	0.
N10	0.	1.	0.
N11	0.	0.	1.
N12	0.	0.	0.
N13	1.	0.5	0.5
N14	1.	0.	0.5
N15	1.	0.5	0.
N16	0.	0.5	0.5
N17	0.	0.	0.5
N18	0.	0.5	0.

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

Les fonctions de forme de cet élément sont les suivantes :

$$w_1=x\,y(x-1)(2\,y-1)/2$$
$$w_2=x\,z(x-1)(2\,z-1)/2$$
$$w_3=x(x-1)(z+y-1)(2\,z+2\,y-1)/2$$
$$w_4=x\,y(x+1)(2\,y-1)/2$$
$$w_5=x\,z(x+1)(2\,z-1)/2$$
$$w_6=x(x+1)(z+y-1)(2\,z+2\,y-1)/2$$
$$w_7=2\,x\,y\,z(x-1)$$
$$w_8=-2\,x\,z(x-1)(z+y-1)$$
$$w_9=-2\,x\,y(x-1)(z+y-1)$$
$$w_{10}=y\left(1-x^2\right)(2\,y-1)$$
$$w_{11}=z\left(1-x^2\right)(2\,z-1)$$
$$w_{12}=\left(1-x^2\right)(z+y-1)(2\,z+2\,y-1)$$
$$w_{13}=2\,x\,y\,z(x+1)$$
$$w_{14}=-2\,x\,z(x+1)(z+y-1)$$
$$w_{15}=-2\,x\,y(x+1)(z+y-1)$$
$$w_{16}=4\,y\,z\left(1-x^2\right)$$
$$w_{17}=4\,z\left(x^2-1\right)(z+y-1)$$
$$w_{18}=4\,y\left(x^2-1\right)(z+y-1)$$

Figure 2.1-2: fonctions de formes

Concernant les familles de points de Gauss à associer à cet élément, il est opportun de consulter la documentation R3.01.01 («fonctions de forme des éléments isoparamétriques»). Pour cet élément, il a été convenu de reprendre celles du pentaèdre à 15 nœuds.

Dans le paragraphe suivant, nous répondrons à la question suivante :

Comment introduire le nouveau type de maille ou élément de référence dans le Source de Code_Aster ?

2.2 Introduction de la maille et de l'élément de référence dans le source de Code_Aster

Avant de se pencher sur les fichiers fortran, on va enrichir les catalogues en question.

2.2.1 Au niveau des catalogues d'éléments

2.2.1.1 Catalogue `Commons/mesh_types.py`

Il faut définir dans ce catalogue, un nouveau type de maille, dont le nom doit être suffisamment explicite. On choisit le nom `PENTA18` pour représenter une maille pendaèdrique à 18 nœuds.

Dans ce catalogue, on va définir ce nouveau `type_maille` : `PENTA18` ainsi qu'un ou plusieurs élément de référence (`ELREFE`) pouvant s'appuyer sur ce type_maille. En général, on ne définit qu'un seul `ELREFE` pour un type_maille . Ici il s'agit de l' `ELREFE` `'P18'` .

Le `type_maille` est purement géométrique (et topologique). Sa définition peut se résumer au schéma le représentant avec la numérotation locale de ses nœuds.

L' `ELREFE` concerne les éléments finis qui s'appuient sur ce `type_maille` : choix des fonctions de forme, choix des familles de points de Gauss.

On enrichit ce catalogue par le bloc suivant :

```
PENTA18 = MeshType(nbno=18, dim=3, code='P18')

P18 = Elrefe()
P18.addLocation('NOEU', 18)
P18.addLocation('NOEU_S', 6)
```


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

```
    ...
    P18.addLocation('FPG21', 21)
    PENTA18.addElrefe(P18)
```

On constate à la lecture de ce bloc que cette maille de type PENTA18 est une maille 3D à 18 nœuds. Elle est associée à l'élément de référence P18. Sur cet elrefe sont définies plusieurs familles de points de Gauss : 'NOEU', 'NOEU_S' 'FPG21', ...

L'étape suivante est d'utiliser cette maille dans une ou plusieurs modélisations de *Code_Aster* . On va enrichir le catalogue des phénomènes et modélisations.

2.2.1.2 Catalogue `Commons/phenomenons_modelisations.py`

Il a été convenu que ce type_maille soit utilisé uniquement avec la modélisation 3D du phénomène MECANIQUE. Ainsi, nous avons ajouté un nouvel élément à cette modélisation, l'élément MECA_PENTA18. Le contenu de ce catalogue relatif à la modélisation 3D est le suivant :

```
phen.add('3D', Modelisation(dim=(3,3), code='3D_',
    attrs=(
        (AT.NBSIGM,'6'),
        (AT.TYPMOD,'COMP3D'),
    ),
    elements=(
        (MT.HEXA8     , EL.MECA_HEXA8),
        (MT.PENTA6    , EL.MECA_PENTA6),
        ...
        (MT.PENTA15   , EL.MECA_PENTA15),
        (MT.PENTA18   , EL.MECA_PENTA18),
```

Pour que cet élément puisse être pris en compte dans les calculs élémentaires, il faut l'ajouter aux catalogues d'éléments.

2.2.1.3 Catalogue `Elements/meca_hexa20.py`

L'enrichissement de la modélisation 3D avec un nouvel élément (MECA_PENTA18) engendre un impact dans la catalogue d'élément meca_hexa20.py .

On doit y ajouter :

```
class MECA_PENTA18(MECA_HEXA20):
    meshType = MT.PENTA18
    nodes = ( SetOfNodes('EN1', (1,2,3,4,5,...,17,18,)), )
    elrefe =(
            ElrefeLoc(MT.P18, gauss = ('RIGI=FPG21','MASS=FPG21', ...),
                      mater=('RIGI','MASS','NOEU','FPG1',),),
            ElrefeLoc(MT.QU9, gauss = ('RIGI=FPG9',...),),
            ElrefeLoc(MT.TR6, gauss = ('RIGI=FPG6', ...),),
            )
```

La déclaration ci-dessus indique que le nouvel élément MECA_PENTA18 hérite de tous les calculs élémentaires définis pour l'élément MECA_HEXA20.
En revanche, on « surcharge » dans l'élément MECA_HEXA20 : son type de maille (meshType), ses « ensembles de noeuds » (nodes) et sa liste d' « elrefe » (elrefe).

L'étape suivante est la saisie des caractéristiques de cet élément de référence. Elle n'a pas lieu dans les catalogues, mais dans des sources fortran.

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

2.2.2 Au niveau du source fortran

On y définit les coordonnées des nœuds, les fonctions de forme et leurs dérivées, c'est à dire toutes les caractéristiques de ce nouvel élément.

Le tableau ci-dessous présente la liste des routines fortran à modifier pour prendre en compte le nouvel ELREFE P18 :

Routines	Fonctionnalités
elraca	Caractéristiques générales de l'élément de référence : nombre et noms des familles de points de Gauss, coordonnées des nœuds, ...
elraga	Définition des poids et coordonnées des points de Gauss
elrfvf	Définition des fonctions de forme
elrfdf	Définition des dérivées des fonctions de forme
elrfd2	Définition des dérivées secondes des fonctions de forme
inmat5	Calcule la matrice de passage Gauss => nœuds à partir de la matrice Gauss => nœuds_sommets
inmat6	Calcule la matrice de passage Gauss => nœuds_sommets
nuelrf	Donne le numéro de la routine jni00i associé à un élément de référence.

Tableau 2.2.2-1: Routines propres au développement d'un nouvel ELREFE

Remarques :

Lors de l'introduction de l'ELREFE P18 , il a été également nécessaire de modifier (mais pour des raisons moins « essentielles » : ECLA_PG , PROJ_CHAMP , ...) les routines suivantes : ecla3d, forme0, forme1,pj3dtr, pjxxut, elref7, ismali

Il faut être très vigilant quand on ajoute un nouveau type de maille car la liste des type_maille (ou du moins leur nombre) est connue en « dur » dans de nombreux sources. La liste est la suivante : iradhs.F90, ircam1.F90, ircame.F90, ircmpe.F90, ircmva.F90, irmhdf.F90, irmmf2.F90, irmmma.F90, irmpga.F90, lrcame.F90, lrfmed.F90, lrmhdf.F90, lrmmdi.F90, lrmmf1.F90, lrmmf3.F90, lrmmfa.F90, lrmmma.F90, lrmtyp.F90, lrvemo.F90.

A titre d'exemple, voici la liste des routines modifiées pour prendre en compte le nouveau type_maille PENTA18 : momaba.F90, forme0.F90, forme1.F90, pjtyma.F90, pjxxut.F90, ermes3.F90, gmgnre.F90, lrmtyp.F90, cm1518.F90, cm2027.F90, cmlqlq.F90, cmqlql.F90, inigms.F90, iradhs.F90, ircam1.F90, irceca.F90, ircecs.F90, ircers.F90, ircmpr.F90, irgmtb.F90, irmac2.F90, irmaca.F90, irmasu.F90, irmmma.F90, lrvemo.F90, creaco.F90, utidea.F90.

A ce stade, on peut construire un exécutable qui peut correspondre à notre besoin. Mais deux questions restent en suspens :

- quelles sont les fonctionnalités de pré et post-traitement à faire évoluer pour pouvoir générer un maillage composé de PENTA18 et pour pouvoir imprimer et visualiser les résultats issus du calcul élémentaire ?
- quelles sont les autres fonctionnalités qui risqueraient d'être impactées (voire cassées) par cette évolution ?

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

2.3 Mise à jour des commandes de pré et post traitement dans *Code_Aster*

2.3.1 Opérateur CREA_MAILLAGE

2.3.1.1 Option QUAD_LINE

Cette option permet de transformer un maillage quadratique en maillage linéaire. Dans notre cas, la maille PENTA18 est quadratique, elle est donc concernée par cette option. Sans intervention dans le source fortran, l'utilisation de cette option en présence de ce type de maille devrait conduire à un arrêt brutal de Code_Aster dans cet opérateur.

Routines	fonctionnalités
cmqlql	Transformation des mailles quadratiques en linéaire
cmqlma	Création des objets TYPMAIL et CONNEX
cmqlnm	Récupération des nœuds milieux
cm2027 cm27ma cm27na cm27nd	Mise à jour de l'option HEXA20_27

Tableau 2.3.1.1-1: Routines de la commande CREA_MAILLAGE/QUAD_LINE impactées

2.3.1.2 Opérateur IMPR_RESU

Pour pouvoir imprimer les résultats calculés à partir d'un modèle composé entre autres d'éléments MECA_PENTA18, il est indispensable de mettre à jour le source fortran de cet opérateur.

Le tableau suivant présente les routines concernées :

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

Format	Routines	Commentaires
GMSH	irgmsh	Écriture d'un champ
	irgmor	Retourne l'ordre d'impression des éléments.
	irgmma, irgmm3	Transforme les mailles en mailles linéaires
	irgmcg, irgmce	Impression d'un cham_elem
	irgmtb	Retourne le tableau de découpage des éléments
	irmgms	Écriture du maillage au format GMSH
IDEAS	ircers, ircecs	Écriture d'un cham_elem
	irmasu	Écriture du maillage
MED	irmmma	Écriture des mailles
	ircam1, ircame	Écriture d'un champ
	ircmpe, ircmpn, ircmpr	Écriture des profils des champs
	ircmpva	Écriture des valeurs
	irmhdf	Écriture du maillage
	irmmf2	Écriture des familles
	irmpga	Écriture des points de Gauss

Tableau 2.3.1.2-1: Routines de la commande IMPR_RESU impactées

2.3.2 Opérateurs LIRE_RESU, LIRE_CHAMP

La modification de l'opérateur IMPR_RESU va de paire avec celle de LIRE_RESU (ou LIRE_CHAMP).

En effet, si l'on imprime un champ on doit être capable de le relire. C'est pourquoi, une intervention dans ces opérateurs est à prévoir.

Routines	Commentaires
lrmtyp	Types de mailles pour med
lrmpga	Localisation des points de Gauss
lrvemo	Vérifie la cohérence entre le modèle fourni et les données
lrcame	Lecture d'un champ
lrmhdf	Lecture du maillage
lrmmdi	Lecture des dimensions du maillage
lrmmf1, lrmmf3, lrmmfa	Lecture des familles
lrmmma	Lecture des mailles
op0150	Lecture d'un résultat

Tableau 2.3.2-1: Routines des opérateurs LIRE_RESU et LIRE_CHAMP impactées

Dans le paragraphe suivant, nous listons les opérateurs indirectement impactés par l'ajout d'une nouvelle maille ou d'un nouvel élément de référence.

2.4 Mise à jour des autres opérateurs de *Code_Aster*

Il n'est pas facile d'établir une liste exhaustive.

Nous avons procédé par :

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

- une recherche des chaines de caractères 'PENTA', 'P15' dans le fortran via des 'grep' pour sortir la liste des fichiers fortran liés aux pentaèdres.
- une analyse minutieuse de chacun d'eux afin de déterminer ceux qui doivent être ajustés.

La liste est la suivante :

Opérateurs	Routines	Fonctionnalités
AFFE_CARA_ELEM: option : LIAISON_GROUP	panbno	Calcule le nombre de nœuds sommets, arêtes d'une maille
CREA_MAILLAGE: ECLA_PG	ecla3d	Décomposer les types des éléments 3D en sous éléments
DEFI_GROUP: CREA_GROUP_NO	gmgnre	Remplir la liste des nœuds sous sous-jacente à la liste de maille
PROJ_CHAMP	pjxxut	Prépare la liste des mailles et les listes de nœuds utiles à la projection
	pjtyma	Retourne le type d'élément à partir du maillage
	pj3dco	Créé la correspondance entre les nœuds et les mailles
	pj3dtr	Transforme le corresp_2_maillage en corresp_2_maillage_final
MODI_MAILLAGE: MODI_MAILLE(option 'NOEU_QUART')	momaba	Modification des mailles via Barsum
	barpen	Traitement des mailles (Barsum)
AFFE_CARA_ELEM	op0019	Ajout de l'élément MECA_PENTA18
CREA_MAILLAGE: LINE_QUAD	cmlqlq	
PRE_GMSH	inigms	Initialisation des types de mailles pour le passage GMSH vers Aster
CALC_ERREUR: OPTION='QIRE_ELEM'	te0368, ermes3	Calcule de l'estimateur d'erreur en mécanique 3D
CALC_ERREUR: OPTION='ERME_ELNO'	te375	Calcule de l'estimateur d'erreur en mécanique 3D

Tableau 2.4-1: Liste des autres opérateurs et routines impactés

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

2.5 Validation

La validation a consisté à :

- tester le développement sur un cas-test interne composé d'un calcul de mécanique linéaire sur un élément dont le type de la maille est un PENTA18 . On a pu comparer les résultats avec ceux de l'étude similaire basée sur des PENTA15 ,
- tester le développement sur un cas-test de référence en mécanique linéaire (ZZZZ238B),
- tester le développement sur un cas-test de référence en mécanique non-linéaire (SSNP121Q),
- tester le mot-clé QUAD_LINE de l'opérateur CREA_MAILLAGE (cas-test ZZZZ206A),
- tester le mot-clé PENTA15_18 de l'opérateur CREA_MAILLAGE ,
- tester les IMPR_RESU aux différents formats, ainsi que LIRE_RESU et LIRE_CHAMP .

Le passage de la liste restreinte a été effectué pour vérifier l'ensemble des fonctionnalités de *Code_Aster* .

2.6 Documentations

- La représentation de votre nouvelle maille doit apparaître dans la documentation U3.01.00 (« description du fichier de maillage de Code_Aster »),
- Les fonctions de forme associées au nouvel élément de référence doivent être présentées dans le document R3.01.01 (« fonctions de forme des éléments isoparamétriques »),
- L'ajout de l'élément MECA_PENTA18 dans la modélisation 3D engendre un travail documentaire dans le document U3.14.01 (Modélisations 3D et 3D_SI mécaniques)
- Le développement de l'option PENTA15_18 de CREA_MAILLAGE s'accompagne par l'enrichissement du document U4.23.02 (opérateur CREA_MAILLAGE)
- Les documentations des cas-tests de validation doivent être mis à jour. Ici, il s'agit des documents V6.03.121 (SSNP121) et V1.01.238 (ZZZZ238).

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