
To introduce a new type of mesh or a new element of reference

Summary:

This document describes what it is necessary to do to introduce a new type of mesh into *CoDe_Aster* or a new element of reference.

In a few words, to add a new type of mesh and/or a new element of reference, it is necessary:

- to define and introduce the characteristics of the mesh and/or the element of reference into the source of Code_Aster: position of the nodes, connectivity, functions of form, position of the points of Gauss,...
- to think of the impact on the level of the existing features in Code_Aster,
- to enrich documentations "user" (Doc. U) and documentations "reference" (Doc. R) relating to the file by grid, the functions of form,...
- to enrich by the existing CAS-tests in order to test your new finite element.

A complete example will be described in this document.

Contents

1 Introduction.....	3
2 Introduction of a new type of mesh or a new element.....	3
2.1 Presentation of the mesh and the element of reference.....	3
2.2 Introduction of the mesh and the element of reference into the source of Code_Aster.....	5
2.2.1 On the level of the catalogues of elements.....	5
2.2.1.1 Catalogue Commons/mesh_types.pthere.....	5
2.2.1.2 Catalogue Commons/phenomenons_modelisations.py.....	6
2.2.1.3 Catalogue Elements/meca_hexa20.py.....	6
2.2.2 On the level of source FORTRAN.....	6
2.3 Update of the orders the pre one and post treatment in Code_Aster.....	8
2.3.1 Operator CREA_MALLAGE.....	8
2.3.1.1 Option QUAD_LINE.....	8
2.3.1.2 Operator IMPR_RESU.....	8
2.3.2 Operators LIRE_RESU, LIRE_CHAMP.....	9
2.4 Update of the other operators of Code_Aster.....	9
2.5 Validation.....	11
2.6 Documentations.....	11

1 Introduction

For thinking of introducing a new type of mesh into Code_Aster, it is necessary to have:

- consulted U3.01.00 documentation ("description of the file of grid of Code_Aster") in order to know conventions of writing (order of the nodes,...),
- consulted R3.01.01 documentation ("functions of the shape of the isoparametric elements") in order to know the representations of the elements of references, their functions of forms, the existing families of points of Gauss,...

After having:

- dimensioned your element,
- placed the nodes,
- determined the functions of forms and their derivative,
- allotted one or of the families of points of Gauss,

You can pass at the following stage, which consists in introducing your mesh or element of reference into Code_Aster.

This stage passes by the modification of the source of Code_Aster, i.e. the modification of:

- catalogues of elements,
- files FORTRAN.

We will explicitly describe the interventions in each file concerned.

After this stage, you will be able to carry out calculations on models accepting your new type of element. It is interesting to be able to print, display and/or read again your results. It is not thus possible to restore your development without updating the orders of preprocessing and postprocessing. We will list the routines concerned.

This stage, it remains to check that the whole of the features of Code_Aster is always operational. We will see that the source relative to other operators will need to be updated. We will list the operators and the routines concerned.

The phase of development being finalized, it remains the phase of validation. The various tests to be carried out will be presented in this document.

Lastly, it will remain to bring up to date documentations. We will enumerate various documentations to be enriched.

In order to review these various stages, we will present an example illustrating the addition of a new mesh and an element of reference.

2 Introduction of a new type of mesh or a new element

One proposes in this example the development of a finite element based on a new mesh, a pentahedron with 18 nodes, in the field of non-linear mechanics.

2.1 Presentation of the mesh and the element of reference

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

A pentahedron with 18 nodes is a mesh whose nodes are localised:

- at the tops (6 nodes),
- in the middle of each edge (9 nodes),
- in the middle of the quadrangular faces (3 nodes).

A representation of element of reference associated with this mesh is the following one:

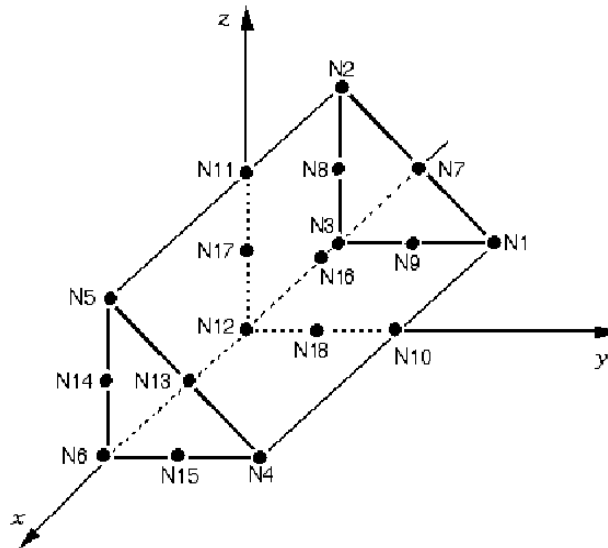


Figure 2.1-1: Element of reference

To carry out a representation of this element (dimension, position and order of the nodes,...), it is advised to consult U3.01.00 documentation ("description of the file of grid of Code_Aster"). For example, a convention on the order of the nodes is presented there: one starts by ordering the nodes tops, then the nodes mediums (on the edges), and finally the nodes central (on the faces).

The coordinates of the nodes are:

	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.

The functions of form of this element are the following ones:

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

Figure 2.1-2: functions of forms

Concerning the families of points of Gauss to be associated with this element, it is convenient to consult R3.01.01 documentation ("functions of form of the isoparametric elements"). For this element, it was agreed to take again those of the pentahedron with 15 nodes.

In the following paragraph, we will answer the following question:

How to introduce the new type of mesh or element of reference into the Source of Code_Aster?

2.2 Introduction of the mesh and the element of reference into the source of Code_Aster

Before leaning on files FORTRAN, one will enrich the catalogues in question.

2.2.1 On the level of the catalogues of elements

2.2.1.1 Catalogue Commons/mesh_types.pthere

It is necessary to define in this catalogue, a new type of mesh, whose name must be sufficiently explicit. The name is chosen PENTA18 to represent a pendaedric mesh with 18 nodes.

In this catalogue, one will define this new `type_maille : PENTA18` like one or more element of reference (`ELREFE`) being able to rest on this `type_maille`. In general, one is defined `ELREFE` for a `type_maille`. Here it is about `ELREFE 'P18'` .

`type_maille` is purely geometrical (and topological). Its definition can summarize with the diagram the representative with the local classification of its nodes.

`ELREFE` relate to the finite elements which rest on it `type_maille` : choice of the functions of form, choice of the families of points of Gauss.

One enriches this catalogue by the following block:

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

P18 = Elrefe ()
P18.addLocation ('NOEU', 18)
P18.addLocation ('NOEU_S', 6)
...
P18.addLocation ('FPG21', 21)
PENTA18.addElrefe (P18)
```

One notes with the reading of this block that this mesh of the type `PENTA18` is a mesh 3D with 18 nodes. It is associated with the element of reference `P18`. On this elrefe several families of points of Gauss are defined: `'NOEU'`, `'NOEU_S'` `'FPG21'`, ...

The following stage is to use this mesh in one or more modelings of `Code_Aster`. One will enrich the catalogue by the phenomena and modelings.

2.2.1.2 Catalogue `Commons/phenomenons_modelisations.py`

It was agreed that this `type_maille` is used only with modeling 3D of the phenomenon `MECHANICS`. Thus, we added a new element to this modeling, the element `MECA_PENTA18`. The contents of this catalogue relating to modeling 3D are the following:

```
phen.add ('3D', Modeling (dim= (3.3), code=' 3D_ ',
    attrs= (
        (AT.NBSIGM, '6'),
        (AT.TYPMOD, 'COMP3D'),
    ),
    elements= (
        (MT.HEXA8      , EL.MECA_HEX8),
        (MT.PENTA6     , EL.MECA_PENTA6),
        ...
        (MT.PENTA15    , EL.MECA_PENTA15),
        ( MT.PENTA18  , EL.MECA_PENTA18 ),
    )
```

So that this element can be taken into account in elementary calculations, it should be added to the catalogues of elements.

2.2.1.3 Catalogue `Elements/meca_hexa20.py`

The enrichment of modeling 3D with one new element (`MECA_PENTA18`) generate an impact in the catalogue of element `meca_hexa20.py`.

One must add to it:

```
class MECA_PENTA18 (MECA_HEX20):
    meshType = MT.PENTA18
    nodes = (SetOfNodes ('EN1', (1,2,3,4,5,..., 17.18)),)
    elrefe = (
        ElrefeLoc (MT.P18, gauss = ('RIGI=FPG21', 'MASS=FPG21',...),
                    mater= ('RIGI', 'FARMHOUSE', 'NOEU', 'FPG1',)),
        ElrefeLoc (MT.QU9, gauss = ('RIGI=FPG9',...),),
        ElrefeLoc (MT.TR6, gauss = ('RIGI=FPG6',...),),
    )
```

The declaration above indicates that the new element `MECA_PENTA18` inherit all definite elementary calculations for the element `MECA_HEX20`.

On the other hand, one "overloads" in the element `MECA_HEX20`: its type of mesh (`meshType`), its "whole of nodes" (`nodes`) and its list of "elrefe" (`elrefe`).

The following stage is the seizure of the characteristics of this element of reference. It does not take place in the catalogues, but in sources FORTRAN.

2.2.2 On the level of source FORTRAN

One defines in it the coordinates of the nodes, the functions of form and their derivative, i.e. all the characteristics of this new element.

The table below presents the list of routines FORTRAN to be modified to take into account the new one ELREFE P18 :

Routines	Features
elraca	General characteristics of the element of reference: number and names of the families of points of Gauss, coordinates of the nodes,...
elraga	Definition of the weights and coordinates of the points of Gauss
elrfvf	Definition of the functions of form
elrfdf	Definition of the derivative of the functions of form
elrfd2	Definition of the derivative second of the functions of form
inmat5	Calculate the matrix of passage Gauss => nodes starting from the matrix Gauss => nœuds_sommets
inmat6	Calculate the matrix of passage Gauss => nœuds_sommets
nuelrf	Give the number of the routine jni00i associated with an element of reference.

Table 2.2.2-1: Routines suitable for the development again ELREFE

Note:

During the introduction of the ELREFE P18, it was also necessary to modify (but for "essential" reasons less: ECLA_PG, PROJ_CHAMP, ...) following routines: ecla3d, forme0, forme1, pj3dtr, pjxxut, elref7, ismali

It is necessary to be very vigilant when one adds a new type of mesh because the list of type_maille (or at least their number) is known in "hard" in many sources. The list is the following one: iradhs. F90, ircaml. 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, lrmffa. F90, lrmmma. F90, lrmtyp. F90, lrvemo. F90.

By way of an example, here the list of the routines modified to take into account new the 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, ircaml. 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.

This stage, one can build achievable who can correspond to our need. But two questions remain outstanding:

- which are the features the pre one and postprocessing to make evolve to be able to generate a grid made up of PENTA18 and to be able to print and display the results resulting from elementary calculation?
- which are the other features which would be likely to be impacted (even broken) by this evolution?

2.3 Update of the orders the pre one and post treatment in Code_Aster

2.3.1 Operator CREA_MAILLAGE

2.3.1.1 Option QUAD_LINE

This option makes it possible to transform a quadratic grid into linear grid. In our case, mesh PENTA18 is quadratic, it is thus concerned with this option. Without intervention in the source FORTRAN, the use of this option in the presence of this kind of mesh should lead to a brutal stop of Code_Aster in this operator.

Routines	features
cmqlql	Transformation of the quadratic meshes into linear
cmqlma	Creation of the objects TYPMAIL and CONNEX
cmqlnm	Recovery of the nodes mediums
cm2027 cm27ma cm27na cm27nd	Update of the option HEXA20_27

Table 2.3.1.1-1: Routines of the order CREA_MAILLAGE/QUAD_LINE impacted

2.3.1.2 Operator IMPR_RESU

To be able to print the results calculated starting from a made up model inter alia elements MECA_PENTA18, it is essential to update source FORTRAN of this operator.

The following table presents the routines concerned:

Format	Routines	Comments
GMSH	irgmsh	Writing of a field
	irgmor	Turn over the order of impression of the elements.
	irgmma, irgmm3	Transform the meshes into linear meshes
	irgmcg, irgmce	Impression of one cham_elem
	irgmtb	Turn over the table of cutting of the elements
	irmgms	Writing of the grid to the format GMSH
IDEAS	ircers, ircecs	Writing of one cham_elem
	irmasu	Writing of the grid
MED	irmma	Writing of the meshes
	ircaml, ircame	Writing of a field
	ircmpe, ircmpn, ircmpr	Writing of the profiles of the fields
	ircmpva	Writing of the values
	irmhdf	Writing of the grid
	irmmf2	Writing of the families
	irmpga	Writing of the points of Gauss

Table 2.3.1.2-1: Routines of the order IMPR_RESU impacted

2.3.2 Operators LIRE_RESU, LIRE_CHAMP

Modification of the operator IMPR_RESU goes from pair with that of LIRE_RESU (or LIRE_CHAMP).

Indeed, if one prints a field one must be able to read again it. This is why, an intervention in these operators is to be envisaged.

Routines	Comments
lrmtyp	Types of meshes for med
lrmpga	Localization of the points of Gauss
lrvemo	Check coherence between the provided model and the data
lrcame	Reading of a field
lrmhdf	Reading of the grid
lrmmdi	Reading of dimensions of the grid
lrmmf1, lrmmf3, lrmffa	Reading of the families
lrmma	Reading of the meshes
op0150	Reading of a result

Table 2.3.2-1: Routines of the operators LIRE_RESU and LIRE_CHAMP impacted

In the following paragraph, we list operators indirectly impacted by the addition of a new mesh or a new element of reference.

2.4 Update of the other operators of Code_Aster

It is not easy to draw up an exhaustive list.

We proceeded by:

- a research of 'the P15', character strings 'PENTA' in FORTRAN via 'grep' to leave the list files FORTRAN related to the pentahedrons.
- a meticulous analysis of each one of them in order to determine those which must be adjusted.

The list is the following one:

Operators	Routines	Features
AFFE_CARA_ELEM: option: LIAISON_GROUP	panbno	Calculate the number of nodes tops, edges of a mesh
CREA_MALLAGE: ECLA_PG	ecla3d	To break up the types of the elements 3D into under elements
DEFI_GROUP: CREA_GROUP_NO	gmgnre	To fill out the list of the nodes under subjacent with the list of mesh
PROJ_CHAMP	pjxxut	Prepare the list of the meshes and the lists of nodes useful for projection
	pjtyma	Turn over the type of element starting from the grid
	pj3dco	Created the correspondence enters the nodes and the meshes
	pj3dtr	Transform it <code>corresp_2_maillage</code> in <code>corresp_2_maillage_final</code>
MODI_MALLAGE: MODI_MAILLE (option 'NOEU_QUART')	momaba	Modification of the meshes via Barsum
	barpen	Treatment of the meshes (Barsum)
AFFE_CARA_ELEM	op0019	Addition of the element MECA_PENTA18
CREA_MALLAGE: LINE_QUAD	cmlqlq	
PRE_GMSH	inigms	Initialization of the types of meshes for the passage GMSH towards Aster
CALC_ERREUR: OPTION=' QIRE_ELEM'	te0368, ermes3	Calculate of the estimator of error in mechanics 3D
CALC_ERREUR: OPTION=' ERME_ELNO'	te375	Calculate of the estimator of error in mechanics 3D

Table 2.4-1: List of the other operators and routines impacted

2.5 Validation

The validation consisted with:

- to test the development on a CAS-test interns made up of a linear calculation of mechanics on an element whose type of the mesh is a PENTA18. One could compare the results with those of the similar study based on PENTA15,
- to test the development on a CAS-test of reference in linear mechanics (ZZZZ238B),
- to test the development on a CAS-test of reference in non-linear mechanics (SSNP121Q),
- to test keyword QUAD_LINE of operator CREA_MALLAGE (CAS-test ZZZZ206A),
- to test keyword PENTA15_18 of operator CREA_MALLAGE,
- to test the IMPR_RESU with the various formats, like LIRE_RESU and LIRE_CHAMP.

The passage of the short list was carried out to check the whole of the features of *Code_Aster* .

2.6 Documentations

- The representation of your new mesh must appear in U3.01.00 documentation ("description of the file of grid of Code_Aster"),
- The functions of form associated with the new element with reference must be presented in the R3.01.01 document ("functions of the shape of the isoparametric elements"),
- The addition of the element MECA_PENTA18 in modeling 3D generate a documentary work in the document U3.14.01 (Modelings 3D and 3D_SI mechanics)
- The development of the option PENTA15_18 of CREA_MALLAGE be accompanied by enrichment by the document U4.23.02 (operator CREA_MALLAGE)
- Documentations of the CAS-tests of validation must be updated. Here, it is the documents V6.03.121 (SSNP121) and V1.01.238 (ZZZZ238).