Operator AFFE_MODELE

Usage

Define the physical phenomenon modeled (mechanical, thermal or acoustic) and the type of finite elements.

This operator allows assigning a model on entire mesh or a part of it. It defines

- the degree of freedom on the nodes (and the equation or the associated conservation equation)
- the types of finite elements on the meshes.

The potential of the finite elements that can be selected are described in the booklets [U3].

The types of meshes are described in the document "Description of the mesh file of Code_Aster" [U3.01.00]

This operator can also define a distribution of finite elements to parallelize basic calculations and assemblies.

Produces a data structure of the type `model`.

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

Table of Contents

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

Syntax

```
mo [modele] = AFFE_MODELE (
     ♦ | MAILLAGE = ma,                                 / [maillage]
                                                        / [squelette]
       | GRILLE = grile,                                  [grille]
     ♦ | AFFE = _F (
                    ♦ / TOUT = 'OUI',
                      / MAILLE = mail,                    [l_maille]
                      / NOEUD = noeu,                     [l_noeud ]
                      / GROUP_MA = g_mail,                [l_gr_maille]
                      / GROUP_NO = g_noeu,                [l_gr_noeud ]
                    ♦ / ♦ PHENOMENE = 'MECANIQUE',
                        ♦ MODELISATION = ...              (see [§3.2.1])
                      / ♦ PHENOMENE = 'THERMIQUE'
                        ♦ MODELISATION = ...              (see [§3.2.1])
                      / ♦ PHENOMENE : 'ACOUSTIQUE',
                        ♦ MODELISATION = ...              (see [§3.2.1])
                    ),
       | AFFE_SOUS_STRUC = _F(
                         ♦ / TOUT = 'OUI',
                           / SUPER_MAILLE = l_mail,       [l_maille]
                            ),
      ◊ VERIF = | 'MAILLE',
                | 'NOEUD',
      ◊ VERI_JACOBIEN= / 'OUI',                           [DEFAUT]
                       / 'NON',
      ◊ GRANDEUR_CARA= _F (
                         ◊ LONGUEUR = lcara,              [R]
                         ◊ PRESSION = pcara,              [R]
                         ◊ TEMPERATURE = tcara,           [R]
                         ),
      ◊ PARTITION = _F (
                ◊ PARALLELISME =
                    /'GROUP_ELEM',                        [DEFAUT]
                    /'MAIL_CONTIGU',
                         ◊ CHARGE_PROC0_MA = / 100,       [DEFAUT]
                                             / pct,
                    /'MAIL_DISPERSE',
                         ◊ CHARGE_PROC0_MA = / 100,       [DEFAUT]
                                             / pct,
                    /'SOUS_DOMAINE',
                         ♦ PARTITION = part,              [sd_partit]
```

```
                              ◊ CHARGE_PROC0_SD = / 0,       [DEFAUT]
                                                  / nbsd,
                          /'CENTRALISE',

                              ),
      ◊ INFO = / 1,                                          [DEFAUT]
               / 2,
);
```

Operands

Operand MAILLAGE

```
♦ | MAILLAGE = ma,
```

Name of the associated mesh on which the elements are assigned.

Note:

For the axisymmetric modelisation, the axis of revolution should be Y axis. The entire structure must be meshed in $X \geq 0$.

Operand GRILLE

```
GRILLE = grile,
```

Name of the associated grid on which the elements are assigned. The grid must be defined by `DEFI_GRILLE` (see [U4.24.02]).

Keyword AFFE

```
♦ | AFFE =
```

Defines entities of the mesh and types of elements that will be assigned. For each occurrence, the user can introduce a list of modelisations. The rule of overload applies between the various modelisations from left to right.

For example:

```
AFFE=_F (TOUT='OUI', PHENOMENE='MECANIQUE',
                     MODELISATION= ("AXIS", "AXIS_SI"),)
```

Various modelisation overloads the one on the left. `AXIS_SI` overloads `AXIS` on the meshes where `AXIS_SI` exists (mesh `QUAD4` and `QUAD8`)

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

Note:

The code stops with an <F> Error if the dimension of the modelisation is not the same (for example MODELISATION= ("3D", "D_PLAN")). Moreover the specified mesh must have a dimension equal to the dimension of MODELISATION assigned. If not, the code emits an <A>larm. This alarm protects the user who uses a model that is incomplete (for example, if the user only uses modelisation AXIS_SI on a mesh containing only TRIA6).

The entities of the mesh are specified by the operands:

Operand	Meaning
TOUT	Assigned to the whole mesh (but not the nodes)
GROUP_MA	Assigned to a list of mesh groups
GROUP_NO	Assigned to a list of node groups (see remark)
MAILLE	Assigned to a list of meshes
NOEUD	Assigned to a list of nodes (see remark)

Remark:
The use of elements being based only on nodes does not make it possible to assign materials via AFFE_MATERIAU. So these elements cannot be used in STAT_NON_LINE [U4.51.03] or in DYNA_NON_LINE [U4.53.01]. In this case meshes should be created as a preliminary mesh POI1 using CREA_POI1 keyword in CREA_MAILLAGE [U4.23.02]

The use of such elements is reserved for linear computation on discrete elements, of which all the characteristics are assigned by AFFE_CARA_ELEM.

The type of element is specified by the operands:

Operands	Meaning
PHENOMENE	Physical phenomenon modeled (associated conservation equation)
MODELISATION	Type of interpolation or discretization

Operands PHENOMENE and MODELISATION

♦ PHENOMENE
♦ MODELISATION

These are required for each instance of the keyword factor AFFE. This pair of keyword defines how bijective element type is allocated to a mesh. The possible models are listed below by listing them in groups.

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

ACOUSTIQUE

ACOUSTIQUE 2D continuum

PLAN	U3.33.01

ACOUSTIQUE 3D continuum

3D	U3.33.01

THERMIQUE

THERMIQUE 2D shell

COQUE_AXIS	U3.22.01
COQUE_PLAN	U3.22.01

THERMIQUE 2D continuum

AXIS_DIAG	U3.23.01
AXIS_FOURIER	U3.23.02
AXIS	U3.23.01
PLAN_DIAG	U3.23.01
PLAN	U3.23.01

THERMIQUE 3D shell

COQUE	U3.22.01

THERMIQUE 3D continuum

3D_DIAG	U3.24.01
3D	U3.24.01

MECANIQUE 2D

MECANIQUE 2D discrete elements

2D_DIS_TR
2D_DIS_T

MECANIQUE 2D fluid-structure

2D_FLUIDE	U3.13.03
2D_FLUI_ABSO	U3.13.13
2D_FLUI_PESA	U3.14.02
2D_FLUI_STRU	U3.13.03
AXIS_FLUIDE	U3.13.03
AXIS_FLUI_STRU	U3.13.03
D_PLAN_ABSO	U3.13.12

MECANIQUE 2D continuum

AXIS	U3.13.01
AXIS_FOURIER	U3.13.02

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

```
 AXIS_SI                                    U3.13.05
 C_PLAN_SI                                  U3.13.05
 C_PLAN                                     U3.13.01
 D_PLAN_SI                                  U3.13.05
 D_PLAN                                     U3.13.01

MECANIQUE 2D quasi incompressible
 AXIS_INCO                                  U3.13.07
 D_PLAN_INCO                                U3.13.07
 AXIS_INCO_UP                               R3.06.08
 D_PLAN_INCO_UP                             R3.06.08
 AXIS_INCO_OSGS                             R3.06.08
 D_PLAN_INCO_OSGS                           R3.06.08
 AXIS_INCO_GD                               R3.06.08
 D_PLAN_INCO_GD                             R3.06.08
 AXIS_INCO_LOG                              R3.06.08
 D_PLAN_INCO_LOG                            R3.06.08
 AXIS_INCO_LUP                              R3.06.08
 D_PLAN_INCO_LUP                            R3.06.08

MECANIQUE 2D non local
 C_PLAN_GRAD_EPSI                           U3.13.06
 D_PLAN_GRAD_EPSI                           U3.13.06
 D_PLAN_GRAD_VARI
 D_PLAN_GVNO                                R5.04.04
 AXIS_GVNO                                  R5.04.04
 D_PLAN_GRAD_SIGM                           R5.03.24

MECANIQUE 2D plates and shells
 COQUE_AXIS                                 U3.12.02
 COQUE_C_PLAN                               U3.12.02
 COQUE_D_PLAN                               U3.12.02

MECANIQUE 2D element joined for crack propagation
 PLAN_JOINT                                 U3.13.14
 AXIS_JOINT                                 U3.13.14
 PLAN_JOINT_HYME                            R3.06.09
 PLAN_INTERFACE                             R3.06.13
 PLAN_INTERFACE_S                           R3.06.13
 AXIS_INTERFACE                             R3.06.13
 AXIS_INTERFACE_S                           R3.06.13

MECANIQUE 2D internal discontinuities elements for initiation and
```

```
crack propagation
 PLAN_ELDI                             U3.13.14
 AXIS_ELDI                             U3.13.14

MECANIQUE 2D thermo-hydro-mechanique
 AXIS_HH2MD
 AXIS_HH2MS
 AXIS_HHMD
 AXIS_HHMS
 AXIS_HHM                              U3.13.08
 AXIS_HMD                              U3.13.08
 AXIS_HMS

AXIS_HM
 AXIS_THH2D
 AXIS_THH2S
 AXIS_THH2MD
 AXIS_THH2MS
 AXIS_THHD
 AXIS_THHS
 AXIS_THHMD
 AXIS_THHMS
 AXIS_THMD
 AXIS_THMS
 AXIS_THM                              U3.13.08
 AXIS_HHD                              R5.04.03
 AXIS_HHS                              R5.04.03
 AXIS_HH2D                             R5.04.03
 AXIS_HH2S                             R5.04.03

 D_PLAN_HH2MD
 D_PLAN_HH2MS
 D_PLAN_HHMD
 D_PLAN_HHMS
 D_PLAN_HHM                            U3.13.08
 D_PLAN_HMD
 D_PLAN_HMS
 D_PLAN_HM                             U3.13.08
 D_PLAN_HM_P                           U3.13.08
 D_PLAN_THH2D
 D_PLAN_THH2S
 D_PLAN_THH2MD
 D_PLAN_THH2MS
```

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

D_PLAN_THHD	
D_PLAN_THHS	
D_PLAN_THHMD	
D_PLAN_THHMS	
D_PLAN_THMD	
D_PLAN_THMS	
D_PLAN_THM	U3.13.08
D_PLAN_HHD	R5.04.03
D_ PLAN_HHS	R5.04.03
D_PLAN_HS	R5.04.03
D_PLAN_HH2D	R5.04.03
D_PLAN_HH2S	R5.04.03
D_PLAN_2DG	R5.04.03
D_PLAN_DIL	R5.04.03

MECANIQUE 2D unsaturated hydraulic finite volume
D_PLAN _HH2SUC
D_PLAN_HH2SUDA
D_PLAN_HH2SUDM

MECANIQUE 2D elements joined with hydraulic coupling
AXIS_JHMS
PLAN_JHMS

For 2D meshes, when the contact is defined on the lip of crack, this allows to inform the mesh group or meshes likely to be crossed by crack. Following types of meshes are allowed: QUAD8 and TRIA6 for mesh and SEG3 for the edge. If the meshes are linear, they should be transformed into quadratic mesh as a minimum (using LINE_QUAD of operator CREA_MAILLAGE).

MECANIQUE 3D

MECANIQUE 3D bars and cables

2D_BARRE	
BARRE	U3.11.01
CABLE_POULIE	U3.11.03
CABLE	U3.11.03
CABLE_GAINE	R3.08.10

MECANIQUE 3D discrete elements

DIS_TR	U3.11.02
DIS_T	U3.11.02

MECANIQUE 3D fluid-structure

```
 3D_FAISCEAU
 3D_FLUIDE                            U3.14.02

MECANIQUE 3D absorbine border
 3D_ABSO                              U3.14.09
 3D_FLUI_ABSO                         U3.14.10

MECANIQUE 3D grids of concrete reinforcement
 GRILLE_MEMBRANE
 GRILLE_EXCENTRE                      U3.12.04

MECANIQUE 3D continuum
 3D_SI                                U3.14.01
 3D                                   U3.14.01

MECANIQUE 3D non local
 3D_GRAD_EPSI                         U3.14.11
 3D_GRAD_VARI
 3D_GVNO                              R5.04.04

MECANIQUE 3D plates, shells and membranes
 COQUE_3D                             U3.12.03
 DKT                                  U3.12.01
 DST                                  U3.12.01
 Q4G                                  U3.12.01
 DKTG                                 U3.12.01
 Q4GG                                 U3.12.01
 MEMBRANE                             U3.12.04

MECANIQUE 3D beams
 FLUI_STRU                            U3.14.02
 POU_C_T                              U3.11.01
 POU_D_EM                             U3.11.07
 POU_D_E                              U3.11.01
 POU_D_TGM                            U3.11.04
 POU_D_TG                             U3.11.04
 POU_D_T_GD                           U3.11.05
 POU_D_T                              U3.11.01

MECANIQUE 3D quasi incompressible
 3D_INCO                              U3.14.06
 3D_INCO_UP                           R3.06.08
 3D_INCO_OSGS                         R3.06.08
```

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

```
 3D_INCO_GD                                 R3.06.08
 3D_INCO_LOG                                R3.06.08
 3D_INCO_LUP                                R3.06.08

MECANIQUE 3D thermo-hydro-mechanique
 3D_HHMD
 3D_HHM                                     U3.14.07
 3D_HMD
 3D_HM                                      U3.14.07
 3D_THHD
 3D_THHMD
 3D_THHM                                    U3.14.07
 3D_THMD
 3D_THM                                     U3.14.07
 3D_THVD
 3D_THH2MD
 3D_THH2M
 3D_HH2MD
 3D_HH2MS
 3D_THH2S
 3D_THH2D
 3D_HHD                                     R5.04.03
 3D_HHS                                     R5.04.03
 3D_HS                                      R5.04.03
 3D_HH2D                                    R5.04.03
 3D_HH2S                                    R5.04.03

MECANIQUE 3D  unsaturated hydraulic finite volume
 3D_HH2SUC
 3D_HH2SUDA
 3D_HH2SUDM

MECANIQUE 3D pipes
 TUYAU_3M                                   U3.11.06
 TUYAU_6M                                   U3.11.06

MECANIQUE 3D solid shell elements
 SHB                                        U3.12.05
```

For 3D meshes, when the contact is defined on the lip of crack, this allows to inform the mesh group or meshes likely to be crossed by crack. Following types of meshes are allowed: HEXA20, PENTA15 and TETRA10 for mesh and QUAD8 and TRIA6 for the elements. If the meshes are linear, they should be transformed into quadratic mesh as a minimum (using

LINE_QUAD of operator CREA_MAILLAGE).

```
MECHANICAL 3D element joints for crack propogation
 3D_JOINT                              U3.13.14
 3D_JOINT_HYME                         R3.06.09
 3D_INTERFACE                          R3.06.13
 3D_INTERFACE_S                        R3.06.13
```

Keyword **AFFE_SOUS_STRUC**

| AFFE_SOUS_STRUC

Is used as a model using static substructures [U1.01.04].

♦ / TOUT = 'OUI',

All (super) meshes are affected.

/ SUPER_MAILLE = l_mail, [l_maille]

l_mail is the list of super-mesh you want to assign in the model. As for the finite elements, it is not required to allocate all the meshes of the mesh. It is AFFE_MODELE that confirms what substructures will be used in the model. The difference with the conventional finite element is that on the super-meshes, the user neither chooses the MODELISATION nor the PHENOMENE because the macro-element (built by operator MACR_ELEM_STAT [U4.62.01]) that will be affected on the super-mesh has its own modelisation and its own phenomenon (the ones which were used to calculate it).

Operand **VERIF**

◊ VERIF

Value	**Meaning**
'MAILLE'	verifies the assignment with all the required meshes if no error
'NOEUD'	verifies the assignment with all the required nodes if no error

By Default: no checking is carried out.

Operand **VERI_JACOBIEN**

```
◊ VERI_JACOBIEN= / 'OUI',                   [DEFAUT]
                 / 'NON',
```

This keyword is used to check that the mesh of the model are not too distorted. The Jacobian of the geometric transformation which transforms the reference element in each real model mesh is calculated. If on the different integration points of a mesh, the Jacobian changes sign it means that this mesh is "bad." An alarm (`CALCULEL_7`) is then issued.

Operand GRANDEUR_CARA

◊ GRANDEUR_CARA= _F (◊ LONGUEUR = lcara,....)

This keyword is used to define some physical characteristic variables of the problem addressed. These quantities are being used for "sizing" certain terms estimators error by "HM". See [R4.10.05].

Keyword PARTITION

◊ PARTITION

This keyword is used to distribute the finite element model for parallel computation elements, assemblies and some linear solvers. See [U2.08.06] "User Manual of parallelism."

It defines how (or not) the meshes will be distributed / elements for the parallel phases of Code_Aster. The user thus has the possibility of controlling the distribution between the processors.

Parallelism operates:

- on basic calculations and assemblies matrices and vectors (this is what the keyword factor `PARTITION` can control)
- the resolution of the linear system solver is parallelized if (see [U4.50.01]).

Note:

It is possible to change the mode of distribution during the study. Just use the command `MODI_MODELE [U4.41.02]`.

Operand PARALLELISME

PARALLELISME = / 'CENTRALISE'

The parallelism begins at the level of the linear solver. Each processor builds and supplies to the entire system solver to solve. Elementary calculations are not parallelized.

PARALLELISME = / 'GROUP_ELEM' [DEFAULT]

This is the distribution method chosen by default. It enables a perfect load balancing a priori that is to say that each processor will, for a given element type, the same number of elementary calculations (to almost). Obviously this does not prejudge the final load balancing especially in non-linear where the cost of an elementary calculation depends on parameters other than the element type calculations. In this mode, the model elements are grouped by "group" in order to share some calculations thereby increase efficiency. The number of elements per group may be selected from the DEBUT command [U4.11.01]. Moreover, it is the only way be able to shift basic calculations induced late elements, ie by loads such as boundary conditions dualized or continuous contact.

PARALLELISME = / 'MAIL_DISPERSE'

The distribution takes place on the mesh. They are evenly distributed on the different processors available. The mesh is distributed among different processors as is done when the cards are distributed to multiple players. Also referred to as distribution "cyclical".

For example, with a pattern comprising 8 stitches made on four processors, following distribution is obtained:

Mode of distribution	Mesh1	Mesh2	Mesh3	Mesh4	Mesh5	Mesh6	Mesh7	Mesh8
MAIL_DISPERSE	Proc 0	Proc 1	Proc 2	Proc 3	Proc 0	Proc 1	Proc 2	Proc 3

We see that with this distribution, a processor address mesh evenly spaced in the order of grid cells. The advantage of this distribution is that "Statistically", each processor will process all hexahedral of wedges, ..., and triangles.

Workload for elementary calculations will generally be well distributed. In contrast, the matrix assembly on a processor will be very "dispersed", in contrast to what happens for the user MAIL_CONTIGU.

PARALLELISME = /'MAIL_CONTIGU'

The distribution takes place on the mesh. They are divided into contiguous packets meshes on the different available processors.

For example, with a mesh pattern comprising 8, a 4-processor machine available, the following distribution was obtained:

Mode of distribution	Mesh1	Mesh2	Mesh3	Mesh4	Mesh5	Mesh6	Mesh7	Mesh8
MAIL_DISPERSE	Proc 0	Proc 0	Proc 1	Proc 1	Proc 2	Proc 2	Proc 3	Proc 3

For this distribution, the workload for elementary calculations may be less balanced. For example, a processor may have to be treated as "easy" to edge meshes. However, the matrix assembly on a processor is generally more compact.

Keyword CHARGE_PROC0_MA

```
◊ CHARGE_PROC0_MA = / 100,                              [DEFAULT]
                    / pct,
```

This key word is accessible only for the modes from parallelism “MAIL_DISPERSE” and “MAIL_CONTIGU”. Indeed these modes of distribution do not distribute in general equitably the load of computations because of dualized boundary conditions whose elementary computations are treated by processor 0.

If the user wishes to relieve processor 0 (or overload it), one can use key word CHARGE_PROC0_MA. This key word makes it possible for the user to choose the percentage of load which they wish to assign to processor 0.

For example, if the user chooses CHARGE_PROC0_MA = 80, processor 0 will treat 20% of elements of less than the other processors, that is 80% of the load which processor 0 should support if the sharing were equal between the processors.

PARALLELISME = / 'SOUS_DOMAINE'

The mesh distribution is based on a decomposition into sub-areas via the built in upstream DEFI_PARTITION operator.

```
♦ PARTITION = part,                                [sd_partit]
◊ CHARGE_PROC0_SD = / 0,                           [DEFAUT]
                    / nbsd,
```

The keyword PARTITION receives the product concept by DEFI_PARTITION that describes the partitioning into subdomains.

Keyword CHARGE_PROC0_SD is used to assign the number of sub-domains for processor 0 (master processor). If CHARGE_PROC0_SD = 1, then processor 0 will assume one subdomain.

For example, a data structure sd_partit with 5 subdomains and a machine with two processors, and CHARGE_PROC0_SD = 2, we obtain the following distribution:

Mode of distribution	Sub Dom 1	Sub Dom 2	Sub Dom 3	Sub Dom 4	Sub Dom 5
SOUS_DOMAINE	Proc 0	Proc 0	Proc 1	Proc 1	Proc 1

Warning: This document is translated from the Machine Translated page found on Code_Aster Documentation page. It may be imprecise and inaccurate in w hole or in part and is provided as a convenience.

Production Run

From keywords PHENOMENE and MODELISATION, we create a data structure specifying the type of element attached to each cell. There may be additional creations mesh type POI1 when assignments are made on nodes or groups of nodes. These meshes are not accessible to the user. This is why it is highly advisable to use CREA_MAILLAGE [U4.23.02] to create POI1 mesh used in the command file (STAT_NON_LINE for example).

A brief reminder of the assignments is printed automatically (INFO = 1) in the message file.

```
SUR LES 612 MAILLES DU MAILLAGE MA
 ON A DEMANDE L'AFFECTATION DE  612
 ON A PU EN AFFECTER            612

 MODELISATION   ELEMENT FINI    TYPE MAILLE     NOMBRE
 3D             MECA_TETRA4     TETRA4          52
 3D             MECA_PENTA6     PENTA6          16
…
 3D             MECA_FACE3      TRIA3           60
```

Example

```
mo = AFFE_MODELE ( MAILLAGE = ma,
                     VERIF = ( 'MAILLE', 'NOEUD',),
                     AFFE = ( _F ( GROUP_MA = gma,
                                   PHENOMENE = 'MECANIQUE',
                                   MODELISATION = '3D', ),
                              _F ( GROUP_NO = gno,
                                   PHENOMENE = 'MECANIQUE',
                                   MODELISATION = 'DIS_T', ),
                             ),
                 );
```

For phenomenon MECHANIQUE modelisation, the user assigns
Isoparametric 3D elements on mesh group 'gma'
Discrete elements having 3 degrees of freedom of translation on node group 'gno'