Operator AFFE_MODELE

1 Goal

To define the modelled physical phenomenon (mechanical, thermal or acoustic) and the type of finite elements.

This operator allows to affect modelings on whole or part of the grid, which defines:

- degrees of freedom on the nodes (and the equation or the associated conservation equations),
- types of finite elements on the meshs,

The possibilities of the finite elements being able to be selected are described in the booklets [U3].

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

This operator also allows to define a distribution of the finite elements in order to parallel elementary calculations and the assemblies.

Product a structure of data of the type `model`.

2 Syntax

```
Mo [model] = AFFE_MODELE      (

     ♦     GRID   =  my,                          /  [grid]
                                                     /  [skeleton]

     ♦  |  AFFE   =  _F  (
                          ♦  /  ALL     =  'YES',
                             /  GROUP_MA  =  g_mail,          [l_gr_maille]

                          ♦  /  ♦  PHENOMENON =     'MECHANICAL',
                                ♦  MODELING =… (see [§3.2.1])

                             /  ♦  PHENOMENON =     'THERMAL'
                                ♦  MODELING =… (see [§3.2.1])

                             /  ♦  PHENOMENON:      'ACOUSTIC',
                                ♦  MODELING =… (see [§3.2.1])

                     ),

        |  AFFE_SOUS_STRUC = _F (
                       ♦  /  ALL    =  'YES',
                          /  SUPER_MAILLE =  l_mail,          [l_maille]

                                   )
     ◊  VERI_JACOBIEN=  /  'YES'                          [DEFECT]
                        /  'NOT'
     ◊  GRANDEUR_CARA=  _F  (
             ◊  LENGTH     =  will lcara,                   [R]
             ◊  PRESSURE      =  will pcara,                   [R]
             ◊  TEMPERATURE  =  will tcara,                   [R]

     ◊  DISTRIBUTION  =  _F  (
             ◊  METHOD =
                        /'SOUS_DOMAINE'                   [DEFECT]
                            ◊ NB_SOUS_DOMAINE =/ nb_proc  [DEFECT]
                                              / nb_sous_dom
                            ◊ PARTITIONNEUR   =/ 'MONGREL'
[DEFECT]
                                              / 'SCOTCH TAPE'

                        /'MAIL_CONTIGU'
                            ◊ CHARGE_PROC0_MA =/100       [DEFECT]
                                              / pct
                        /'MAIL_DISPERSE'
                            ◊ CHARGE_PROC0_MA =/100       [DEFECT]
                                              / pct
                        /'GROUP_ELEM'
                        /'CENTRALIZES'
                          )
     ◊  INFORMATION  =  /  1
   [DEFECT]
                    /  2,

                          )
```

3 Operands

3.1 Operand GRID

♦ GRID = my

Name of the associated grid on which one affects the elements.

Note:

For axisymmetric modelings, the axis of revolution is the axis Y grid. All the structure must be with a grid in $X \geq 0$.

3.2 Keyword AFFE

♦ | AFFE

Defines the entities of the grid and the types of elements which will be affected for them. For each occurrence, one can introduce a list of modelings. The rule of overload applies between various modelings, from left to right.

For example:

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

Various modelings "overload" the ones the others: AXIS_SI overload AXIS on the meshs where AXIS_SI exist (mesh QUAD4 and QUAD8).

Note:

The code stops in error <F> if modelings of the list are not the very same "dimension" (for example MODELISATION=('3D', 'D_PLAN')). Moreover, for an occurrence of AFFE, the specified meshs whose dimension is that of the dimension of modeling must be all affected. If not the code emits one <A>larme. This alarm protects the user who uses modelings "with holes". If for example, it uses only modeling AXIS_SI on a grid containing only TRIA6.

The entities of the grid are specified by the operands:

Operands	Contents/significance
ALL	Assignment with the totality of the meshs
GROUP_MA	Assignment with a list of groups of meshs

The type of element is specified by the operands:

Operands	Contents/significance
PHENOMENON	Modelled physical phenomenon (associated conservation equation)
MODELING	Type of interpolation or discretization

3.2.1 Operands PHENOMENON and MODELING

♦ PHENOMENON
♦ MODELING

Are obligatory for each occurrence of the keyword factor AFFE. This couple of keywords defines in a bijective way the type of affected element in a kind of mesh. Possible modelings are indicated below by listing them by "packages":

ACOUSTICS

ACOUSTICS 2D continuous mediums

PLAN	U3.33.01

ACOUSTICS 3D continuous mediums

3D	U3.33.01

THERMICS

THERMICS 2D hull

COQUE_AXIS	U3.22.01
COQUE_PLAN	U3.22.01

THERMICS 2D continuous mediums

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

THERMICS 3D hull

HULL	U3.22.01

THERMICS 3D continuous mediums

3D_DIAG	U3.24.01
3D	U3.24.01

MECHANICS 2D

MECHANICS 2D discrete elements

2D_DIS_TR
2D_DIS_T

MECHANICS 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

MECHANICS 2D continuous mediums

AXIS	U3.13.01
AXIS_FOURIER	U3.13.02
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

MECHANICS 2D quasi incompressible

AXIS_INCO_UP	R3.06.08
D_PLAN_INCO_UP	R3.06.08
AXIS_INCO_UPG	U3.13.07 and R3.06.08
D_PLAN_INCO_UPG	U3.13.07 and R3.06.08
AXIS_INCO_UPO	R3.06.08
D_PLAN_INCO_UPO	R3.06.08

MECHANICS 2D not room

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

MECHANICS 2D plates and hulls

COQUE_AXIS	U3.12.02

Mechanics 2D elements joined for the propagation of crack

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

Mechanics 2D elements with internal discontinuities for the starting and the propagation of crack

PLAN_ELDI	U3.13.14
AXIS_ELDI	U3.13.14

MECHANICS 2D thermo-hydro-mechanics

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	
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

MECHANICS 2D hydraulics unsaturated with finished volumes
D_PLAN_HH2SUDA

MECHANICS 2D elements joined with hydraulic coupling
AXIS_JHMS
PLAN_JHMS

For the grids 2D, allows to inform the groups of meshs or the meshs likely to be crossed by the crack when the contact is defined on the lips of the crack. Are allowed the following types of meshs: QUAD8 and TRIA6 and the meshs of edge of these elements, are them SEG3. If the meshs are linear, they should as a preliminary be transformed into quadratic meshs (with LINE_QUAD of the operator CREA_MAILLAGE).

MECHANICS 3D

MECHANICS 3D bars and cables

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

MECHANICS 3D discrete elements

DIS_TR	U3.11.02
DIS_T	U3.11.02

MECHANICS 3D fluid-structure

3D_FAISCEAU	
3D_FLUIDE	U3.14.02

MECHANICS 3D absorbing border

3D_ABSO	U3.14.09
3D_FLUI_ABSO	U3.14.10

MECHANICS 3D grids of concrete reinforcements

GRILLE_MEMBRANE	
GRILLE_EXCENTRE	U3.12.04

MECHANICS 3D continuous mediums

3D_SI	U3.14.01
3D	U3.14.01

MECHANICS 3D not room	
3D_GRAD_EPSI	U3.14.11
3D_GRAD_VARI	
3D_GVNO	R5.04.04
MECHANICS 3D plates, hulls 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
MECHANICS 3D beams	
FLUI_STRU	U3.14.02
LOUSE_FLUI_STRU	U3.14.02
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
MECHANICS 3D quasi incompressible	
3D_INCO _UP	R3.06.08
3D_INCO_UPG	U3.14.06 and R3.06.08
3D_INCO_UPO	R3.06.08
MECHANICS 3D thermo-hydro-mechanics	
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
MECHANICS 3D hydraulics unsaturated with finished volumes	
3D_HH2SUDA	
MECHANICS 3D pipes	
TUYAU_3M	U3.11.06
TUYAU_6M	U3.11.06

MECHANICS 3D massive element of hull

SHB	U3.12.05

For the grids 3D, allows to inform the groups of meshs or the meshs likely to be crossed by the crack when the contact is defined on the lips of the crack. Are allowed the following types of meshs: HEXA20, PENTA15, TETRA10, and the meshs of edges of these elements, are them QUAD8 and TRIA6. If the meshs are linear, they should as a preliminary be transformed into quadratic meshs (with LINE_QUAD of the operator CREA_MAILLAGE).

Mechanics 3D elements joined for the propagation of crack

3D_JOINT	U3.13.14
3D_JOINT_HYME	R3.06.09
3D_INTERFACE	R3.06.13
3D_INTERFACE_S	R3.06.13

3.3 Keyword AFFE_SOUS_STRUC

♦ | AFFE_SOUS_STRUC

Is usable only for one using model of the static substructures [U1.01.04].

♦ / SUPER_MAILLE = l_mail

l_mail is the list of the super-meshs which one wants to affect in the model. As for the finite elements, it is not obligatory to affect all the meshs of the grid. It is AFFE_MODELE who confirms which are the substructures which will be used in the model. The difference with the classical finite elements is that on the super-meshs, one does not choose nor MODELING nor it PHENOMENON because the macronutrient (built by the operator MACR_ELEM_STAT [U4.62.01]) who will be affected on the super-mesh has his own modeling and his own phenomenon (those which were used to calculate it).

Caution! Your model must contain at least a finite element (keyword AFFE with the §3.2) when you use definite static substructures starting from a physical grid (read by LIRE_MAILLAGE) because it is not possible to have only macronutrients in this case.

/ ALL = 'YES'

All them (super) meshs are affected.

3.4 Operand VERI_JACOBIEN

◊ VERI_JACOBIEN = 'YES'/'NOT'

This keyword is used to check that the meshs of the model are not distorted too much. One calculates the jacobien of the geometrical transformation which transforms the element of reference into each real mesh of the model. So on the various points of integration of a mesh, the jacobien changes sign, it is that this mesh is very "badly rotten".
An alarm (CALCULEL_7) is then emitted.

3.5 Operand GRANDEUR_CARA

◊ GRANDEUR_CARA = _F (LENGTH = will lcara,…)

This keyword is used to define some physical sizes characteristic of with the dealt problem. These sizes are currently used "have-to dimension" certain terms of the estimators of error in " HM ". See [R4.10.05].

3.6 Keyword DISTRIBUTION

◊ DISTRIBUTION = _F (METHOD = methoof,…)

This keyword makes it possible to distribute the finite elements of the model for the parallelism of elementary calculations, the assemblies and certain linear solveurs. *Cf.* [U2.08.06] "Note of use of parallelism".

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

Parallelism operates:
- on elementary calculations and the assemblies of matrices and vectors (it is what the keyword factor DISTRIBUTION allows to control),
- with the resolution of the linear system if the solvor is paralleled (*cf.* [U4.50.01]).

Note:

It is possible to modify the mode of distribution during its study. It is enough to use the order MODI_MODELE *[U4.41.02].*

Note:

It can be practical to continue a parallel calculation with a number of processors different from that used for initial calculation. In particular, one can want to carry out certain postprocessings into sequential. It is recommended to use the order MODI_MODELE *to define the distribution to be used in continuation. More precisely, Lorsque initial calculation used it parallelism "by groups of elements" ('* GROUP_ELEM *' or '* SOUS_DOMAINE *'), the order* MODI_MODELE *is useless. On the other hand, Lorsque initial calculation has used it parallelism "by elements" (* 'MAIL_CONTIGU' *or* 'MAIL_DISPERSE' *), the order* MODI_MODELE *is obligatory. If it is forgotten, one is stopped during calculation by one error message.*

3.6.1 Keyword METHOD

3.6.1.1 METHOD =/'CENTRALIZES'

Parallelism starts only on the level of the linear solvor. Each processor builds and provides to the solvor the entirety of the system to be solved. Elementary calculations are not paralleled.

3.6.1.2 METHOD =/'GROUP_ELEM'

CE mode of distribution allows a perfect balancing of load (in term of numbers of calculations elementary) *a priori*, i.e. each processor will carry out, for a kind of element given, the same number of elementary calculations (with near). Obviously that does not prejudge of anything the final balancing of load in particular in non-linear calculations where the cost of an elementary calculation depends on other parameters but the type of element.
In this mode, the elements of the model are gathered by "group" in order to pool certain calculations what makes it possible to gain in effectiveness. The number of elements by group can be selected in the order BEGINNING [U4.11.01].
In addition, it is a question of the only mode able of distributing elementary calculations induced by the late elements, i.e. by the loadings such as the boundary conditions dualized or the continuous contact.

3.6.1.3 METHOD =/'MAIL_DISPERSE'

The distribution takes place on the meshs. They are distributed equitably on the various processors available. The meshs are distributed on the various processors as it is made it when one distributes cards to several players. One also speaks about "cyclic" distribution.

For example, with a model comprising 8 meshs, carried out on 4 processors, one obtains the following distribution:

Mode of distribution	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6	Mesh 7	Mesh 8
MAIL_DISPERSE	Proc. 0	Proc. 1	Proc. 2	Proc. 3	Proc. 0	Proc. 1	Proc. 2	Proc. 3

It is seen that with this mode of distribution, a processor will treat meshs regularly spaced in the order of the meshs of the grid. The advantage of this distribution is that "statistically", each processor will treat as many hexahedrons, of pentahedrons,…, and of triangles.
The workload for elementary calculations in general will be well distributed. On the other hand, the matrix assembled on a processor "will be very dispersed", contrary to what occurs for the mode 'MAIL_CONTIGU'.

3.6.1.4 METHOD =/'MAIL_CONTIGU'

The distribution takes place on the meshs. They are divided into packages of contiguous meshs on various processors available.

For example, with a model comprising 8 meshs, a machine of 4 processors available, the following distribution is obtained:

Mode of distribution	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6	Mesh 7	Mesh 8
MAIL_CONTIGU	Proc. 0	Proc. 0	Proc. 1	Proc. 1	Proc. 2	Proc. 2	Proc. 3	Proc. 3

For this mode of distribution, the workload for elementary calculations can be less balanced. For example, a processor can have to treat only “easy” meshs of edge. On the other hand, the matrix assembled on a processor is in general more compact.

3.6.1.5 Keyword CHARGE_PROC0_MA

```
◊ CHARGE_PROC0_MA =/100 [DEFECT]
                   / pct
```

This keyword is accessible only for the modes from distribution 'MAIL_DISPERSE'and'MAIL_CONTIGU'. Indeed these modes of distribution do not distribute in general equitably the load of calculations because of boundary conditions dualized whose elementary calculations are treated by processor 0.

If one wishes to relieve processor 0 (or on the contrary to overload it), one can use the keyword CHARGE_PROC0_MA. This keyword makes it possible to the user to choose the percentage of load which one wishes 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, is 80% of the load which it should support if the division were equitable between the processors.

3.6.1.6 METHOD =/ 'SOUS_DOMAINE' [DEFECT]

This distribution of the meshs is based on a decomposition grid under-fields, built by a tool external of partitioning defined by the keyword PARTITIONNEUR

```
◊ PARTITION NEUR     =  / 'MONGREL'     [ DEFECT ]
                        / 'SCOTCH TAPE'
```

The number of under-fields can be given by the user, via the keyword NB_SOUS_DOMAINE

```
◊ NB_SOUS_DOMAINE    =     / nbproc     [ DEFECT ]
                           / nb_sous_dom
```

By default, the number of under-fields is taken equal to the number of processors implied in calculation (nbproc).

The elements of the model finite elements carried by the meshs of each under-field are then distribute by groups of similar elements (as in the distribution corresponding to the method GROUP_ELEM), in order to balance elementary calculations as well as possible.
The preliminary partitioning of the grid under-fields makes it possible to ensure that all the elements of a group of finite elements belong to only one under-field.

4 Production run

Starting from the keywords `PHENOMENON` and `MODELING`, one creates a structure of data specifying the type of element attached to each mesh.

A brief recall of the assignments is systematically printed (`INFO=1`) in the file `message`.

For example:

```
ON         612 MESHS OF THE GRID MY
   ONE WITH REQUEST THE ASSIGNMENT OF         612
   ONE WITH PU TO AFFECT SOME                   612

   MODELING      FINITE ELEMENT      TYPE NETS         NUMBER
   3D                MECA_TETRA4       TETRA4                  52
   3D                MECA_PENTA6       PENTA6                  16
…
   3D                MECA_FACE3        TRIA3                   60
```

5 Example

```
Mo = AFFE_MODELE    (  GRID = my,
                       AFFE  = ( _F (  GROUP_MA     =  gma,
                                       PHENOMENON      =  'MECHANICAL',
                                       MODELING  =   '3D'   ),
                    ) )
```

For a modeling of the phenomenon `'MECHANICAL'`, one affects on the group of meshs `gma` elements `3D` isoparametric.