

Operator AFFE_MATERIAU

1 Goal

To assign materials to geometrical zones of a grid or a model. There can be to 26 materials on the same mesh (what is necessary for the multifibre beams or polycrystalline materials), but one will be able to use one field of the type `cham_mater` in the operators.

To define the variables of order (temperature, space, hydration, drying, corrosion,...) for digital calculations.

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

2 Syntax

```
chm [cham_mater] = AFFE_MATERIAU

( ♦ | GRID = my , / [grid]
  | MODEL = Mo , / [skeleton]
  | MODEL = Mo , [model]

# assignment of the name of material :
♦ AFFE = (_F (
  ♦ / ALL = 'YES' ,
    / GROUP_MA = lgma , [l_gr_maille]
  ♦ MATER = / chechmate , [to
subdue]
    / l_mat, [l_mater]
  ),),

# assignment of the variables of orders:
♦ AFFE_VARC = (_F (
  ◊ / ALL = 'YES' , [DEFECT]
    / GROUP_MA = lgma, [l_gr_maille]
  ♦ NOM_VARC = / 'TEMP',
                / 'GEOM',
                / 'CORR',
                / 'EPSA',
                / 'HYDR',
                / 'IRRA',
                / 'M_ACIER',
                / 'M_ZIRC',
                / 'NEUT1',
                / 'NEUT2',
                / 'NEUT3',
                / 'PTOT',
                / 'DIVU',
                / 'SECH',

  ◊ CHAM_GD = chvarc [field]
  ◊ EVOL = evovarc [evol_sdaster]
    ◊ NOM_CHAM = nosymb, [TXM]
    ◊ FONC_INST = finst, [function]
  ◊ PROL_DROITE =/'EXCLUDED', [DEFECT]
                / 'CONSTANT',
                / 'LINEAR',
  ◊ PROL_GAUCHE =/'EXCLUDED', [DEFECT]
                / 'CONSTANT',
                / 'LINEAR',

# If NOM_VARC = 'TEMP' (or 'SECH'):
  ♦ VALE_REF = vref, [R]

# "hidden" keywords:
VARC_TEMP = _F (...),
VARC_GEOM = _F (...),
VARC_HYDR = _F (...),
VARC_SECH = _F (...),
...

```

```
    ),),  
  
# assignment of the "behavior" of the multifibre beams :  
◇ AFFE_COMPOR = (_F (  
    ◆ / ALL = 'YES' , [DEFECT]  
    / GROUP_MA = lgma , [l_gr_maille]  
  
    ◆ COMPOR = compor , [compor]  
    ),),  
  
◇ INFORMATION = / 1 , [DEFECT]  
/ 2
```

3 General information

This order is used to affect the material characteristics on the finite elements of the model (even if in fact the meshes of the grid are really affected). These material characteristics are defined in the materials which one affects on the meshes (keyword `MATER`). Each material contains a certain number of parameters (Young modulus, density,...). These parameters can be related to certain variables. We will call these variables of the “variables of order”.

Currently, the variables of order used are:

- the temperature,
- space,
- the hydration,
- drying,
- metallurgical phases,
- irradiation,
- corrosion,
- ...

All these variables of order must be affected with the order `AFFE_MATERIAU` (keyword `AFFE_VARC`).

In the case of model for multifibre beams, it is also necessary to affect in this order the “behaviors” defined with the order `DEFI_COMPOR/MULTI_FIBRE` [U4.43.06].

4 Operands

4.1 GRID

`GRID = my,`

Name of the grid (or the skeleton) which one wants to affect by material characteristics.

Note:

- *The operation of assignment is the same one for the meshes of a skeleton as for the meshes of a grid. In the continuation of the document, one will always say grid to simplify.*
- *When one affects materials on the meshes of a skeleton, it is that one wants to calculate constraints (for example) on the meshes of postprocessing (coarser).*

4.2 MODEL

`MODEL = Mo,`

Name of the model.

If this argument is used, it is checked that the meshes affected in the order are part of the model well.

4.3 Keyword factor `AFFE`

The keyword factor `AFFE` allows to affect various materials on “pieces” of the grid.

4.3.1 Notice concerning calculations of breaking process

In general, the material characteristics must be known on the finite elements modelling the “matter”: “voluminal” elements (or of structure). Finite elements of “skin” are there to apply boundary conditions and do not have to know the material properties of the subjacent matter. An exception exists for the calculation of the option `CALC_K_G` operators `CALC_G_XXXX`. For these

calculations, the finite elements modelling the lips of the crack must be affected by same material as the "voluminal" elements subjacent.

4.3.2 Operands TOUT=' OUI ', GROUP_MA

The keyword ALL and GROUP_MA allow to indicate the whole of the meshes which will be affected.

If a mesh appears in several occurrences of the keyword factor AFFE, the rule of overload is observed: it is the last assignment which precedes [U2.01.08].

4.3.3 Operand MATER

◆ MATER = chechmate,

Name of the material which one wants to affect.

In the case general, each mesh is affected only by one material. Sometimes, it is necessary to indicate a material list when the nonlinear mechanical behavior is obtained by the order DEFI_COMPOR [U4.43.06]. In this last case, the number of materials of the list EST limited to 26.

Caution: if one can have several materials by mesh, one will be able to use one field of the type cham_mater in the operators.

4.4 Keyword AFFE_VARC

This keyword factor makes it possible to affect fields of **variables of order** on the meshes of the grid. This keyword can be repeated. It is necessary to use several occurrences of AFFE_VARC to be able to affect several different variables of order. But one can also use several occurrences for only one variable. For example, on a mixed model (3D + beams), one can affect like temperature:

1 evol_ther calculated on the elements 3D

1 field of temperature (constant in time) on the elements of beam.

A variable of order is a scalar (real) which can influence the material behavior via the parameters which are functions (for example a parameter of DEFI_MATERIAU/ELAS_FO). A variable of order is a known field **front** calculation. This field can be variable in time. In rare cases, the variables of orders can be used like variables of a function defining a loading (ex:AFFE_CHAR_THER_F/SOURCE).

The variables of order were introduced above all for mechanical calculations. The variable of the most usual order for mechanical calculations is the temperature.

For thermal calculations, it is possible to use one (or several) variable of order, but that relates to only some very rare parameters:

```
SECH_NAPPE      / FUNCTION
THER_FO         / LAMBDA
THER_FO         / RHO_CP
THER_COQUE_FO  / COND_LMM, COND_TMM,...
```

Always for thermal calculations of the variables of orders can be used among the variables of the functions given to the following order:

```
AFFE_CHAR_THER_F/SOURCE
```

It is not possible to assign a dependence to the variables of order for the finite elements absorbents (D_PLAN_ABSO , 3D_ABSO).

4.4.1 Operand NOM_VARC

◆ NOM_VARC = nomvarc,

Name of the variable of order which one wants to affect (TEMP, GEOM, IRRA, CORR, HYDR, SECH, ...).

Significance and role of the various variables:

TEMP	temperature
GEOM	directions of space
CORR	corrosion of steels
EPSA	unelastic deformation
HYDR	hydration of the concrete
IRRA	Irradiation
M_ACIER	metallurgical phases of steel
M_ZIRC	metallurgical phases of the zircaloy
NEUT1	"neutral" variable 1: the material coefficients of materials according to a parameter "user allows to vary" (see example 3 below)
NEUT2	"neutral" variable 2 (like NEUT1)
NEUT3	"neutral" variable 3 (like NEUT1)
SECH	drying of the concrete
PTOT	Total pressure of fluid in THM (chained resolution)
DIVU	Voluminal deformation in THM (chained resolution)

Certain variables of order are scalars. Others are "vectors" including several scalar components. One gives in the table below the name of the components of the variables of orders

TEMP	TEMP, TEMP_MIL, TEMP_SUP, TEMP_INF
GEOM	X, Y, Z
CORR	CORR
EPSA	EPSAXX, EPSAYY, EPSAZZ, EPSAXY, EPSAXZ, EPSAYZ
HYDR	HYDR
IRRA	IRRA
M_ACIER	FERRITE, PEARLITE, PBAINITE, PMARTENS, PAUSTENI, PCOLDSUM, TAUSTE, TRANSF, TACIER
M_ZIRC	ALPHPUR, ALPHBETA, BETA, TZIRC, TIME
NEUT1	NEUT1
NEUT2	NEUT2
NEUT3	NEUT3
SECH	SECH
PTOT	PTOT
DIVU	DIVU

The variable of order IRRA corresponds to a fluence, i.e. the intégale in the time of a neutron flux. It is used by several laws of behavior, in specific units:

- it must be expressed in DPA (displacement per atom) for the law IRRAD3M (cf [R5.03.23]);
- it must be expressed in $10^{20} n/cm^2$ for the laws VISC_IRRA_LOG, GRAN_IRRA_LOG, LEMAITRE_IRRA (cf [R5.03.09]).

4.4.2 Operands TOUT= ' OUI ', GROUP_MA

These keywords make it possible to indicate the meshes of the zone to be affected.

4.4.3 Operand CHAM_GD

This keyword makes it possible to associate with the variable of order nomvarc the field chvarc. This field is a field of realities (not functions). It is thus independent of time and will be used throughout transitory calculations.

If the values of the variable of order are dependent on time, the keyword should be used `EVOL` (see below). `Cham_elem ELGA` are authorized only if they result from the operator `PROJ_CHAMP/METHOD = 'SOUS_POINT'`, to assign the values to the under-points of Gauss of the elements under-points.

4.4.4 Operands `EVOL`, `NOM_CHAM`, `FONC_INST`, `PROL_DROITE`, `PROL_GAUCHE`

These keywords make it possible to associate with the variable of order `nomvarc` the transient `evovarc`. The keyword `NOM_CHAM` allows to indicate the reference symbol of the fields of `SD_résultat` to use. By default, the code chooses:

<code>NOM_VARC</code>	<code>NOM_CHAM</code>
<code>TEMP</code>	<code>'TEMP'</code>
<code>GEOM</code>	<code>'GEOM'</code>
<code>CORR</code>	<code>'CORR'</code>
<code>EPSA</code>	<code>'EPSA_ELNO'</code>
<code>HYDR</code>	<code>'HYDR_ELNO'</code>
<code>IRRA</code>	<code>'IRRA'</code>
<code>M_ACIER</code>	<code>'META_ELNO'</code>
<code>M_ZIRC</code>	<code>'META_ELNO'</code>
<code>NEUT1</code>	<code>'NEUT'</code>
<code>NEUT2</code>	<code>'NEUT'</code>
<code>NEUT3</code>	<code>'NEUT'</code>
<code>SECH</code>	<code>'TEMP'</code>
<code>PTOT</code>	<code>'DEPL'</code>
<code>DIVU</code>	<code>'EPSI'</code>

The fields are real fields (neither complexes, nor functions).

`Cham_elem ELGA` are authorized only if they result from the operator `PROJ_CHAMP/METHOD = 'SOUS_POINT'`, to assign the values to the under-points of Gauss of the elements under-points.

The keyword `FONC_INST = first` allows to define a function (time) which is used as correspondence between the “time” of the evolution `evovarc` (`t_evo`) and the “time” of later calculation (`t_calc`). The function can be a simple “translation” (to take account owing to the fact that the beginning of the moments of mechanical calculation is different from the moment of the beginning of thermal calculation, but one can make more complicated, for example to impose a mechanical loading (thermal dilation) “cyclic” by calculating one cycle of temperature. One will be able to consult the case test `zzzz223a` to illustrate the use of this keyword.

Caution: The function `first` is that which transforms `t_calc` in `t_evo` : `t_evo = first (t_calc)`

Keywords `PROL_GAUCHE` and `PROL_DROITE` allow to specify if one can use the transient `evovarc` before the moment “min” of the transient (`PROL_GAUCHE`) and/or after the moment “max” of the transient (`PROL_DROITE`).

The value `'EXCLUDED'` an error will cause if one seeks to use the transient apart from his field.

The value `'CONSTANT'` prolong the transient by the values at the moment “min” (or “max”).

The value `'LINEAR'` prolong the transient linearly with to leave the 2 first (or the last) points of the transient.

4.4.5 Operand `VALE_REF`

This keyword makes it possible to define a value of “reference” for the variable of order `nomvarc` when this one needs a value of reference.

Currently, only two variables of order require a value of reference: `'TEMP'` and `'SECH'`. For these two variables, the keyword `VALE_REF` is obligatory. For the other variables, this keyword is prohibited.

For variable of order `'TEMP'` in the case of the hulls, the temperature of reference is supposed to be the same one for the 3 components. This is why it is returned only once.

♦ VALE_REF = Tref (or c0) [R]

4.4.5.1 Temperature of reference (T_{ref}) :

The temperature of reference T_{ref} introduced behind the keyword VALE_REF is the temperature for which there is no thermal deformation (cf [R4.08.01]).

If the thermal dilation coefficient α (whose value is introduced into the order DEFI_MATERIAU [U4.43.01]) does not depend on the temperature: $\varepsilon^{th}(T) = \alpha(T - T_{ref})$.

If the thermal dilation coefficient depends on the temperature the mathematical expression allowing the calculation of the thermal deformation differs according to the specification from the thermal dilation coefficient in the order DEFI_MATERIAU :

- values of the thermal dilation coefficient (introduced into DEFI_MATERIAU) were determined by tests of dilatometry carried out at the temperature T_{ref} . In this case, the keyword TEMP_DEF_ALPHA does not have to be specified in the order DEFI_MATERIAU and the thermal deformation is calculated by the expression:

$$\varepsilon^{th}(T) = \alpha(T)(T - T_{ref}) \text{ and } \varepsilon^{th}(T_{ref}) = 0$$

where $\alpha(T)$ is well informed under the keyword ALPHA (or ALPHA_*) in DEFI_MATERIAU.

- the values of the thermal dilation coefficient are determined by tests of dilatometry which took place at a temperature T_{def} different from the temperature of reference T_{ref} . It is then necessary to carry out a change of reference mark in the calculation of the thermal deformation [R4.08.01].

$$\varepsilon^{th}(T) = \varepsilon_m^{th}(T) - \varepsilon_m^{th}(T_{ref})$$

where ε_m^{th} is the thermal deformation measured (definite compared to the temperature T_{def}),
 ε^{th} is the thermal deformation calculated (definite compared to the temperature T_{ref}).

The temperature T_{def} is well informed under the keyword TEMP_DEF_ALPHA in DEFI_MATERIAU, and values of the dilation coefficient (definite compared to the temperature T_{def}) are well informed under the keyword ALPHA or (ALPHA_*) in DEFI_MATERIAU.

Note:

- In a modeling THM, the temperature is not a variable of order and the dilation coefficient not can be a function of the temperature. One thus should not use the variable of order TEMP in THM. An error message will stop you
- It is not possible to use a formula nor a tablecloth for ALPHA, because them modifications to be taken into account described above. The parameter ALPHA can depend ONLY on the temperature and provided that it is a function. The user, if it wishes to use a formula, owes initially the tabuler using the order CALC_FONC_INTERP.

4.4.5.2 Drying of reference (c_0) :

c_0 represent the water content initial of the concrete. The user must provide this number when he does a mechanical calculation (MECA_STATIQUE or STAT_NON_LINE) with a loading of type drying.

c_0 must be given in the same units as “drying” (AFFE_MATERIAU/AFFE_VARC= _F (NOM_VARC=' SECH'...)) for example in L/m^3). This unit must be coherent with the parameter DEFI_MATERIAU/ELAS_FO/K_DESSIC.

With this water content initial, the withdrawal of desiccation is null since:
 $EPS_{rd} = K_{DESSIC} (C_0 - C)$.

4.4.6 State initial in STAT_NON_LINE

In the order STAT_NON_LINE, the initial state is not calculated. It is provided by the user. By default, the initial state of a calculation is “virgin”: internal displacements, constraints and variables are worthless.

The possible inconsistency of an initial state can disappear as of the first step from calculated time but it can persist. That depends on the way in which the law of behavior is integrated (choice of the programmer of the law).

If the behavior is integrated in an incremental way (old vocabulary: COMP_INCR), the inconsistency of the initial state will persist for every moment of calculation. If not, inconsistency will disappear with the first step of time.

It is thus very important that the initial state of calculation is “right”.

If the variables of order, at the initial moment are such that they cause nonworthless unelastic deformations, this deformation involves an incorrect initial state. Indeed, one must have:

$$\sigma = A(\epsilon - \epsilon_{anel})$$

As displacement is null, deformation ϵ is worthless. The constraint σ cannot be worthless.

In practice, it is necessary thus that at the initial moment:

- the temperature is equal to the temperature of reference
- the hydration is worthless
- ...

4.4.7 Keywords “hidden” for the assignment of the variables of order

We saw how the user can affect fields (isolated or coming from sd_resultat) like variable of order for its later calculations.

But the variables of order are named scalars and the associated fields have also named components. The problem is to associate each variable of order with a component of the field.

There exists as much of keyword factor “hidden” than of variables of order allowing these associations. These keywords “are hidden” because they have values by default (see table Cibelow). One should use them only when one wishes to do something of a little “special”. For example:

- to use the variables NEUT_1 or NEUT_2,
- to make pass a field of temperature for a field of corrosion,
- ...

Let us explain these keywords factors on two examples:

The user carried out a “thermal” calculation whose solution is actually an evolution evo1 whose fields contain drying. In this evol_ther, fields called TEMP carry a component called also TEMP.

The user who wishes to use such fields as variable of order SECH will be able to write:

```
CHMAT=AFFE_MATERIAU (...  
    VAR_SECH=_F (NOM_VARC=' SECH', GRANDEUR=' TEMP_R',  
                CMP_VARC=' SECH', CMP_GD=' TEMP'))
```

What one can translate by: "the fields which I wish to affect like variable of order 'SECH' are size TEMP_R and the component to be used is 'TEMP'.

When a variable of order is a "vector" having several scalar components, for example the variable M_ACIER who has 9 components ('PFERRITE',..., 'TACIER'). The user can write:

```
CHMAT=AFFE_MATERIAU (...
  VAR_M_ACIER=_F (NOM_VARC=' M_ACIER', GRANDEUR=' VARI_R',
                  CMP_VARC= ('PFERRITE', 'PEARLITE', ..., 'TACIER'),
                  CMP_GD= ('V1', 'V2', ..., 'V9')))
```

What wants to say: "the fields which I wish to affect like variable of order 'M_ACIER' are size VARI_R and the correspondence of the components to be used is: ('PFERRITE', 'V1'), ('TACIER', 'V9').

The hidden keywords have like values by default:

NOM_VARC	SIZE	CMP_VARC	CMP_GD
TEMP	TEMP_R	TEMP	TEMP
GEOM	GEOM_R	X	X
		Y	Y
		Z	Z
SECH	TEMP_R	SECH	TEMP
HYDR	HYDR_R	HYDR	HYDR
CORR	CORR_R	CORR	CORR
IRRA	IRRA_R	IRRA	IRRA
NEUT1	NEUT_R	NEUT1	X1
NEUT2	NEUT_R	NEUT2	X1
NEUT3	NEUT_R	NEUT3	X1
EPSA	EPSI_R	EPSAXX	EPXX
		EPSAYY	EPYY
		EPSAZZ	EPZZ
		EPSAXY	EPXY
		EPSAXZ	EPXZ
		EPSAXZ	EPXZ
M_ACIER	VARI_R	PFERRITE	V1
		PEARLITE	V2
		PBAINITE	V3
		PMARTENS	V4
		PAUSTENI	V5
		PCOLDSUM	V6
		TAUSTE	V7
		TRANSF	V8
		TACIER	V9

M_ZIRC	VARI_R	ALPHPUR	V1
		ALPHBETA	V2
		BETA	V3
		TZIRC	V4
		TIME	V5
PTOT	DEPL_R	PTOT	PTOT
DIVU	EPSI_R	DIVU	DIVU

4.5 Keyword factor **AFFE_COMPOR**

This keyword factor makes it possible to affect the “behavior multifibre” of the multifibre elements of beams.

The keyword `ALL` and `GROUP_MA` allow to indicate the whole of the meshes which will be affected.

Behind the keyword `COMPOR`, the user will indicate the name of a concept of the type `compor` coming from the order `DEFI_COMPOR/MULTIFIBRE`.

4.5.1 Operand **MATER**

♦ `MATER = chechmate`,

Name of the material which one wants to affect.

In the case general, each mesh is affected only by one material. Sometimes, it is necessary to indicate a material list when the nonlinear mechanical behavior is obtained by the order `DEFI_COMPOR [U4.43.06]`.

5 Examples

Example 1: Mechanics without thermal dilation

```
chmat = AFFE_MATERIAU ( GRID = my,  
  AFFE = (  
    _F (ALL = 'YES' , MATER = steel),  
    _F (GROUP_MA= ('Gma1', 'Gma2', 'Gma3'), MATER = aluminum,)),  
  )
```

On the whole of the grid (except the meshes contained in the groups : `gma1`, `gma2`, `gma3`) the material of name is affected `steel`.

On the meshes contained in the groups `gma1`, `gma2`, `gma3` the material is affected `aluminum`.

Example 2: Mechanics with thermal dilation

Assignment on all the grid of material `CHECHMATE` whose certain parameters are functions of the temperature. Moreover the thermal dilation coefficient is defined for this material. The temporal evolution of the temperature is given via the structure of data result `EVOTH` (of type `evol_ther`). The temperature of reference (that for which dilation is worthless) is worth 20 degrees.

```
CHMAT = AFFE_MATERIAU (GRID = MY,  
  AFFE = _F (TOUT=' OUI', MATER = CHECHMATE, ),  
  AFFE_VARC=_F (NOM_VARC=' TEMP', EVOL =EVOTH, VALE_REF=20. ),  
  )
```

Example 3: Mechanics with thermal dilation + complex modeling

In the preceding example, the thermal evolution (EVOTH) applies to all the elements of the model. But it can happen that this situation is unrealistic for certain complex modelings. Several times the keyword should then be repeated `AFFE_VARC/NOM_VARC=' TEMP'` to affect different thermal evolutions on various parts of the model.

In the following example, the model is a model 3D in which are plunged steel reinforcements. A thermal calculation was carried out as a preliminary without taking account of the reinforcements. One got a result which one called `EVOTH3D`. The temperature of the nodes of the elements of the reinforcement is then unknown. So in addition, one is able to evaluate the temperature of the reinforcements (measurements,...) and that this temperature is stored in the field (`TEMP_ARM`), one can then do the mechanical calculation of thermal dilation with the material field according to:

```
CHMAT = AFFE_MATERIAU (GRID = MY,  
  AFFE = _F (...),  
  AFFE_VARC= (  
    _F (NOM_VARC=' TEMP', GROUP_MA=' VOLUM', EVOL =EVOTH3D, VALE_REF=20. ),  
    _F (NOM_VARC=' TEMP', GROUP_MA=' ARMA', CHAM_GD =TEMP_ARM,  
VALE_REF=20. ),  
  ))
```

Example 4: Mechanics with influence of the irradiation

Assignment on all the grid of material `CHECHMATE` whose certain parameters are functions of the irradiation. The temporal evolution of the irradiation is given via the SD result `EVOL = FLUENC`.

```
CHMAT = AFFE_MATERIAU (GRID = MY,  
  AFFE = _F (TOUT=' OUI', MATER = CHECHMATE, ),  
  AFFE_VARC=_F (NOM_VARC=' IRRA', EVOL =FLUENC, ),  
  )
```

Example 5: Mechanical calculation with a field of imposed modulus Young

In this example (resulting from the case test `ssnv130c`), one wants to illustrate the possibility of using a field of modulus Young whom one supposes known (`CHYOUNG`). For example, this field is read in a file (`LIRE_CHAMP`) or it is the result of a calculation.

The “trick” consists in defining a material for which the Young modulus (keyword `ELAS/E`) “identity” of the variable is the function `'NEUT1'` and the field is affected `CHYOUNG` like variable of order `'NEUT1'`.

```
CHYOUNG=...  
NU_F=DEFI_CONSTANTE (VALE=0.3)  
E_F = DEFI_FONCTION (NOM_PARA=' NEUT1', VALE= (- 1.E-9, - 1.E-9,  
1.E+9, 1.E+9))  
MA=DEFI_MATERIAU (ELAS_FO=_F (E=E_F, NU=NU_F, ), );  
  
CM=AFFE_MATERIAU (MAILLAGE=M,  
  AFFE=_F (TOUT= 'YES', MATER= MY),  
  AFFE_VARC=_F (NOM_VARC=' NEUT1', CHAM_GD=CHYOUNG),  
  )
```