Notice d'utilisation du modèle THM

Résumé :

Les modélisations THM traitent exclusivement de l'évolution des comportements Thermo-Hydro-Mécaniques des milieux poreux saturés, ou non, par un ou deux fluides. La description numérique et physique des ces modélisations dans Code_Aster est détaillée dans [R7.01.10] et [R7.01.11].

On détaille dans cette documentation la procédure à suivre pour réaliser des simulations dans le contexte des modélisations THM. On décrit dans la première partie les différentes étapes de calculs dans le cadre général des milieux continus dits « généralisés » (cf [R7.01.10]). Dans cette partie, on ne détaillera pas ce qui concerne les modèles mécaniques décrits ailleurs. On restreint dans la seconde partie l'application de ces modélisations pour traiter les milieux poreux qui subissent une dégradation de leur propriétés mécaniques par endommagement du sol ou des roches par exemple. Pour cela, on étend les modélisations THM aux milieux à microstructure en prenant en compte les effets second gradient (cf [R5.04.03]). L'objectif est de corriger la dépendance à la discrétisation spatiale des solutions lorsque la loi de comportement mécanique considérée est de type adoucissante – ce qui est le cas pour tout matériau fragile, et donc les sols en particulier.

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

Table des Matières

1 Les grandes lignes....4
1.1 Contexte des études THM....4
1.2 Généralités....4
1.3 Étapes de calculs....5
2 Les différentes étapes d'un calcul THM....5
2.1 Choix du modèle....5
2.2 Définition du matériau....9
2.2.1 Mot clé simple COMP_THM....10
2.2.2 Mot clé facteur THM_INIT....13
2.2.3 Mot clé facteur THM_LIQU....15
2.2.4 Mot clé facteur THM_GAZ....16
2.2.5 Mot clé facteur THM_VAPE_GAZ....16
2.2.6 Mot clé facteur THM_AIR_DISS....17
2.2.7 Mot clé facteur THM_DIFFU....17
2.2.8 Récapitulatif des fonctions de couplages et leur dépendance....24
Mot clé facteur THM_DIFFU....25
2.3 Initialisation du calcul....28
2.4 Chargements et conditions aux limites....30
2.5 Le calcul non linéaire....33
2.5.1 Résolution par couplage....33
2.5.2 Résolution par chaînage....34
2.5.3 Conseils généraux d'utilisation....36
2.6 Le post-traitement....41
2.6.1 Généralités....41
2.6.2 Variables internes....42
2.6.3 Isovaleurs....44
2.7 Quelques cas tests....45
3 Les modélisations THM par une approche second gradient....46
3.1 Définition du patch....46
3.1.1 Étape 1. Définition du maillage de la structure....47
3.1.2 Étape 2. Duplication du maillage pour définir le patch....47
3.1.3 Étape 3. Modification (éventuelle) du maillage du patch....47
3.2 Choix du modèle....47
3.3 Définition du matériau....48
3.4 Impact sur les conditions aux limites....49
3.5 Résolution du problème....49
Annexe 1 Contraintes généralisées et variables internes....50
Annexe 2 Éléments supplémentaires sur les conditions aux limites en THM....52

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

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

1 Les grandes lignes

1.1 Contexte des études THM

Avant toute chose, il convient de définir le cadre bien précis des calculs Thermo-Hydro-Mécaniques. Ceux-ci ont pour application exclusive l'étude des milieux poreux. Sachant cela, la modélisation THM couvre l'évolution mécanique de ces milieux et les écoulements en leur sein. Ces derniers concernent un ou deux fluides et sont régis par les lois de Darcy (fluides darcéens). Le problème de THM complet traite donc de l'écoulement des ou du fluide(s), de la mécanique du squelette, ainsi que de la thermique: la résolution est très souvent entièrement couplée. Elle peut également être chaînée dans les cas où les phénomènes hydraulique et mécanique sont faiblement couplés.

Remarque :

Dans l'expression des lois de Darcy qui est ici retenue, on néglige l'accélération différentielle de l'eau. Dans le cas de de milieux très perméables et très poreux soumis à un chargement sismique, cela peut constituer une limite.

1.2 Généralités

Les calculs s'appuient sur des familles de lois de comportement THM pour les milieux poreux saturés et non saturés. La mécanique des milieux poreux rassemble une collection très exhaustive de phénomènes physiques touchant aux solides et aux fluides. Elle fait l'hypothèse d'un couplage entre les évolutions mécaniques des solides et des fluides, vus comme des milieux continus, avec les évolutions hydrauliques, qui règlent les problèmes de diffusion de fluides au sein de parois ou de volumes, et les évolutions thermiques. La formulation de la modélisation Thermo-hydro-mécanique (THM) en milieu poreux telle qu'elle est faite dans *Code_Aster* est détaillée dans [R7.01.11] et [R7.01.10]. Toutes les notations employées ici s'y réfèrent donc. On rappelle cependant quelques notations indispensables par la suite :

Concernant les fluides, on considère (cas le plus complet) deux phases (liquide et gaz) et deux constituants appelés par commodité eau et air. On utilise alors les indices suivants :

w pour l'eau liquide
ad pour l'air dissous
as pour l'air sec
vp pour la vapeur d'eau

Les variables thermodynamiques sont :

- les pressions des constituants : $p_w(x,t)$, $p_{ad}(x,t)$, $p_{vp}(x,t)$, $p_{as}(x,t)$,
- la température du milieu $T(x,t)$.

Ces différentes variables ne sont pas totalement indépendantes. En effet, si l'on considère un seul constituant, l'équilibre thermodynamique entre ses phases impose une relation entre la pression de la vapeur et la pression du liquide de ce constituant. Finalement, il n'y a qu'une seule pression indépendante par constituant, de même qu'il n'y a qu'une seule équation de conservation de la masse. Le nombre de pressions indépendantes est donc égal au nombre de constituants indépendants. Le choix de ces pressions varie selon les lois de comportements.

Pour le cas dit saturé (un seul constituant air ou eau), nous avons choisi la pression de cet unique constituant.
Pour le cas dit non saturé (présence d'air et d'eau), nous avons choisi comme variables indépendantes :

- la pression totale du gaz $p_{gz}(x,t) = p_{vp} + p_{as}$,
- la pression capillaire $p_c(x,t) = p_{gz} - p_{lq} = p_{gz} - p_w - p_{ad}$.

Nous verrons par la suite la terminologie Aster pour ces variables.

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

1.3 Étapes de calculs

Pour les étapes nécessaires à la mise en œuvre d'un calcul Aster, indépendamment des aspects purement THM, on se référera à la documentation de chaque commande utilisée.

Dans tout calcul Aster, plusieurs étapes clés doivent être effectuées :

- Choix de la modélisation
- Données matériaux
- Initialisation
- Calcul
- Post-traitement

Ces points sont détaillés dans le chapitre suivant.

2 Les différentes étapes d'un calcul THM

2.1 Choix du modèle

Le traitement numérique en THM nécessite un maillage quadratique puisque les éléments sont de type $P2$ en déplacement et $P1$ en pression et température afin d'éviter des problèmes d'oscillations.

Le choix se fait par l'utilisation de la commande AFFE_MODELE comme dans l'exemple ci-dessous :

```
MODELE=AFFE_MODELE(MAILLAGE=MAIL,
                   AFFE=_F(TOUT='OUI',
                           PHENOMENE='MECANIQUE',
                           MODELISATION='AXIS_THH2MD',),)
```

Dans tous les cas, le phénomène est 'MECANIQUE' (même si la modélisation ne contient pas la mécanique).
L'utilisateur doit renseigner ensuite de manière obligatoire le mot clé MODELISATION. Ce mot clé permet de définir le type d'élément affecté à un type de maille. Les modélisations disponibles en THM sont indiqués dans le tableau 2.1-1.

Remarque concernant le traitement numérique (mot clé se terminant par D ou S) :

Les modélisations se terminant par la lettre D indiquent que l'on fait un traitement permettant de diagonaliser (« lumper ») la matrice afin d'éviter les oscillations pour les problèmes hydrauliques. Pour cela les points d'intégration sont pris aux sommets des éléments. Ce traitement étant peu adapté à la mécanique, on dispose également d'une modélisation dite « sélective ». Dans ce cas, les termes capacitifs sont intégrés aux sommets alors que les termes diffusifs sont intégrés aux points de Gauss. Ces modélisations se terminent par un S. Les autres modélisations intègrent tout aux point de Gauss.

On conseille vivement à l'utilisateur d'utiliser les modélisations D ou S dans les cas sans mécanique et d'utiliser la modélisation S et pour les modélisations avec mécanique.

Les modélisations « classiques » (sans D ni S) sont amenés à être résorbées et sont déconseillées.

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

MODELISATION	Modélisation géométrique	Phénomènes pris en compte
D_PLAN_HM	plane	Mécanique, hydraulique avec une pression inconnue
D_PLAN_HMD	plane	Mécanique, hydraulique avec une pression inconnue (lumpé)
D_PLAN_HMS	plane	Mécanique, hydraulique avec une pression inconnue (sélective)
D_PLAN_HM_SI	plane	Mécanique, hydraulique avec une pression inconnue (sous-intégrée)
D_PLAN_HHM	plane	Mécanique, hydraulique avec deux pressions inconnues
D_PLAN_HHMD	plane	Mécanique, hydraulique avec deux pressions inconnues (lumpé)
D_PLAN_HHMS	plane	Mécanique, hydraulique avec deux pressions inconnues (sélective)
D_PLAN_HH2MD	plane	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
D_PLAN_HH2MS	plane	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
D_PLAN_HH2M_SI	plane	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sous-intégrée). Seule modélisation valide en dynamique.
D_PLAN_THHD	plane	Thermique, hydraulique avec deux pressions inconnues (lumpé)
D_PLAN_THH2D	plane	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
D_PLAN_THH2S	plane	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
D_PLAN_THM	plane	Thermique, mécanique, hydraulique avec une pression inconnue
D_PLAN_THVD	plane	Thermique, mécanique, hydraulique avec deux pressions inconnues (2 phases : eau liquide et vapeur) (lumpé)
D_PLAN_THMD	plane	Thermique, mécanique, hydraulique avec une pression inconnue (lumpé)
D_PLAN_THMS	plane	Thermique, mécanique, hydraulique avec une pression inconnue (sélective)
D_PLAN_THHMD	plane	Thermique, mécanique, hydraulique avec deux pressions inconnues (lumpé)
D_PLAN_THHMS	plane	Thermique, mécanique, hydraulique avec deux pressions inconnues (sélective)
D_PLAN_HHD	plane	hydraulique avec deux pressions inconnues (lumpé)
D_PLAN_HHS	plane	hydraulique avec deux pressions inconnues (sélective)
D_PLAN_HH2D	plane	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
D_PLAN_HH2S	plane	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
D_PLAN_THH2MD	plan	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
D_PLAN_THH2MS	plane	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
AXIS_HM	axisymétrique	Mécanique, hydraulique avec une pression inconnue
AXIS_HMD	axisymétrique	Mécanique, hydraulique avec une pression inconnue (lumpé)
AXIS_HMS	axisymétrique	Mécanique, hydraulique avec une pression inconnue (sélective)
AXIS_HHM	axisymétrique	Mécanique, hydraulique avec deux pressions inconnues
AXIS_HHMD	axisymétrique	Mécanique, hydraulique avec deux pressions inconnues (lumpé)
AXIS_HHMS	axisymétrique	Mécanique, hydraulique avec deux pressions inconnues

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

MODELISATION	Modélisation géométrique	Phénomènes pris en compte
		(sélective)
AXIS_HH2MD	axisymétrique	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
AXIS_HH2MS	axisymétrique	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
AXIS_THHD	axisymétrique	Thermique, hydraulique avec deux pressions inconnues (lumpé)
AXIS_THHS	axisymétrique	Thermique, hydraulique avec deux pressions inconnues (sélective)
AXIS_THH2D	axisymétrique	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
AXIS_THH2S	axisymétrique	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
AXIS_THM	axisymétrique	Thermique, mécanique, hydraulique avec une pression inconnue
AXIS_THMD	axisymétrique	Thermique, mécanique, hydraulique avec une pression inconnue (lumpé)
AXIS_THMS	axisymétrique	Thermique, mécanique, hydraulique avec une pression inconnue (sélective)
AXIS_THVD	axisymétrique	Thermique, mécanique, hydraulique avec deux pressions inconnues (2 phases : eau liquide et vapeur) (lumpé)
AXIS_THHMD	axisymétrique	Thermique, mécanique, hydraulique avec deux pressions inconnues (lumpé)
AXIS_THHMS	axisymétrique	Thermique, mécanique, hydraulique avec deux pressions inconnues (sélective)
AXIS_THH2MD	axisymétrique	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
AXIS_THH2MS	axisymétrique	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
AXIS_HHD	axisymétrique	hydraulique avec deux pressions inconnues (lumpé)
AXIS_HHS	axisymétrique	hydraulique avec deux pressions inconnues (sélective)
AXIS_HH2D	axisymétrique	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
AXIS_HH2S	axisymétrique	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
3D_HM	3D	Mécanique, hydraulique avec une pression inconnue
3D_HMD	3D	Mécanique, hydraulique avec une pression inconnue (lumpé)
3D_HMS	3D	Mécanique, hydraulique avec une pression inconnue (sélective)
3D_HM_SI	3D	Mécanique, hydraulique avec une pression inconnue (sous-intégrée)
3D_HHM	3D	Mécanique, hydraulique avec deux pressions inconnues
3D_HHMD	3D	Mécanique, hydraulique avec deux pressions inconnues (lumpé)
3D_HHMS	3D	Mécanique, hydraulique avec deux pressions inconnues (sélective)
3D_HH2MD	3D	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
3D_HH2MS	3D	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
3D_HH2M_SI	3D	Mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sous-intégrée). Seule modélisation valide en dynamique.
3D_THHD	3D	Thermique, hydraulique avec deux pressions inconnues (lumpé)

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

MODELISATION	Modélisation géométrique	Phénomènes pris en compte
3D_THH2D	3D	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
3D_THH2S	3D	Thermique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
3D_THM	3D	Thermique, mécanique, hydraulique avec une pression inconnue
3D_THMD	3D	Thermique, mécanique, hydraulique avec une pression inconnue (lumpé)
3D_THMS	3D	Thermique, mécanique, hydraulique avec une pression inconnue (sélective)
3D_THVD	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues (2 phases : eau liquide et vapeur) (lumpé)
3D_THHM	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues
3D_THHMD	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues (lumpé)
3D_THHMS	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues (sélective)
3D_THH2MD	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
3D_THH2MS	3D	Thermique, mécanique, hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)
3D_HHD	3D	hydraulique avec deux pressions inconnues (lumpé)
3D_HHS	3D	hydraulique avec deux pressions inconnues (sélective)
3D_HH2D	3D	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (lumpé)
3D_HH2S	3D	hydraulique avec deux pressions inconnues et deux constituants dans la phase gazeuse (sélective)

Tableau 2.1-1: Modélisations THM

Les inconnues principales qui sont aussi les valeurs des degrés de liberté, sont notées dans le cas de la modélisation la plus complète (3D thermique, mécanique, hydraulique avec deux pressions inconnues).

$$\{u\}^{ddl}=\begin{Bmatrix} u_x \\ u_y \\ u_z \\ \mathrm{PRE1}^{ddl} \\ \mathrm{PRE2}^{ddl} \\ T^{ddl} \end{Bmatrix}$$

Le contenu de $PRE1$ et $PRE2$ dépend du couplage choisi et sera explicité dans la section 2.2.2. Suivant la modélisation choisie, seuls certains de ces degrés de liberté existent. Le tableau ci-dessus résume les degrés de liberté utilisés pour chaque modélisation.

MODELISATION	u_x	u_y	u_z	$PRE1^{ddl}$	$PRE2^{ddl}$	T^{ddl}
D_PLAN_HM(_,S,D,SI)	X	X		X		
D_PLAN_HHM(_,S,D)	X	X		X	X	
D_PLAN_HH2M(S,D,_SI)	X	X		X	X	

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

MODELISATION	u_x	u_y	u_z	$PRE1^{ddl}$	$PRE2^{ddl}$	T^{ddl}
D_PLAN_THH(S,D)				X	X	X
D_PLAN_THH2(S,D)				X	X	X
D_PLAN_THM(_,S,D)	X	X		X		X
D_PLAN_THV(_,S,D)				X	X	X
D_PLAN_THHM(S,D)	X	X		X	X	X
D_PLAN_THH2M(S,D)	X	X		X	X	X
D_PLAN_HH(S,D)				X	X	
D_PLAN_HH2(S,D)				X	X	
AXIS_HM(_,S,D)	X	X		X		
AXIS_HHM(_,S,D)	X	X		X	X	
AXIS_HH2M(S,D)	X	X		X	X	
AXIS_THH(S,D)				X	X	X
AXIS_THH2(S,D)				X	X	X
AXIS_THM(_,S,D)	X	X		X		X
AXIS_THV(_,S,D)				X	X	X
AXIS_THHM(S,D)	X	X		X	X	X
AXIS_THH2M(S,D)	X	X		X	X	X
AXIS_HH (S,D)				X	X	
AXIS_HH2(S,D)				X	X	
3D_HM(_,S,D,SI)	X	X	X	X		
3D_HHM(_,S,D)	X	X	X	X	X	
3D_HH2M(S,D,_SI)	X	X	X	X	X	
3D_THH(S,D)				X	X	X
3D_THH2D				X	X	X
3D_THM(_,S,D)	X	X	X	X		X
3D_THVD(_,S,D)				X	X	X
3D_THHM(_,S,D)	X	X	X	X	X	X
3D_THH2M(S,D)	X	X	X	X	X	X
3D_HH(S,D)				X	X	
3D_HH2(S,D)				X	X	

Les contraintes généralisées et les variables internes sont toutes indiquées en [§Annexe 1]. Les notations utilisées sont celles définies dans [R7.01.11].

2.2 Définition du matériau

Le matériau est défini par la commande DEFI_MATERIAU comme dans l'exemple ci-dessous :

```
MATERBO=DEFI_MATERIAU(ELAS=_F(E=5.15000000E8,
                              NU=0.20,
                              ALPHA=0.,),
                      COMP_THM = 'LIQU_AD_GAZ_VAPE',
                      THM_LIQU=_F(RHO=1000.0,
                                  UN_SUR_K=0.,
                                  ALPHA=0.,
                                  CP=0.0,
                                  VISC=VISCOLIQ,
                                  D_VISC_TEMP=DVISCOL,),
                      THM_GAZ=_F(MASS_MOL=0.01,
                                 CP=0.0,
                                 VISC=VISCOGAZ,
                                 D_VISC_TEMP=ZERO,),
                      THM_VAPE_GAZ=_F(MASS_MOL=0.01,
                                      CP=0.0,
                                      VISC=VISCOGAZ,
                                      D_VISC_TEMP=ZERO,),
                      THM_AIR_DISS=_F(
```


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

```
                                CP=0.0,
                                COEF_HENRY=HENRY
                                ),
                    THM_INIT=_F(TEMP=300.0,
                                PRE1=0.0,
                                PRE2=1.E5,
                                PORO=1.,
                                PRES_VAPE=1000.0,
                                ),
                    THM_DIFFU=_F(R_GAZ=8.32,
                                 RHO=2200.0,
                                 CP=1000.0,
                                 BIOT_COEF=1.0,
                                 SATU_PRES=SATUBO,
                                 D_SATU_PRES=DSATBO,
                                 PESA_X=0.0,
                                 PESA_Y=0.0,
                                 PESA_Z=0.0,
                                 PERM_IN=KINTBO,
                                 PERM_LIQU=UNDEMI,
                                 D_PERM_LIQU_SATU=ZERO,
                                 PERM_GAZ=UNDEMI,
                                 D_PERM_SATU_GAZ=ZERO,
                                 D_PERM_PRES_GAZ=ZERO,
                                 FICKV_T=ZERO,
                                 FICKA_T=FICK,
                                 LAMB_T=ZERO,
                                 ),);
```

Nous allons maintenant détailler chacun des mots clés. Nous ne nous attacherons pas ici à la partie mécanique – si mécanique il y a - qui dépend de la loi de comportement choisie. On se reportera pour cela à la documentation de DEFI_MATERIAU (U4.43.01).

2.2.1 Mot clé simple COMP_THM

Permet de sélectionner dès la définition du matériau la loi de couplage THM. Les lois possibles sont

♦ COMP_THM = / 'LIQU_SATU' ,
/ 'LIQU_GAZ' ,
/ 'GAZ' ,
/ 'LIQU_GAZ_ATM' ,
/ 'LIQU_AD_GAZ' ,
/ 'LIQU_VAPE_GAZ' ,
/ 'LIQU_AD_GAZ_VAPE' ,
/ 'LIQU_VAPE' ,
/ 'GAZ'

Loi de comportement d'un gaz parfait c'est-à-dire vérifiant la relation $P/\rho = RT/Mv$ où P est la pression, ρ la masse volumique, Mv la masse molaire, R la constante des gaz parfaits et T la température (confer [R7.01.11] pour plus de détails). Pour un milieu saturé uniquement. Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous le mot clé THM_GAZ.

/ 'LIQU_SATU'

Loi de comportement pour un milieux poreux saturé par un seul liquide (Cf. [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous le mot clé THM_LIQU.

/ 'LIQU_GAZ_ATM'

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

Loi de comportement pour un milieu poreux non saturé avec un liquide et du gaz à pression atmosphérique (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU et THM_GAZ.

/ 'LIQU_VAPE_GAZ'

Loi de comportement pour un milieu poreux non saturé eau/vapeur/air sec avec changement de phase (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU, THM_VAPE et THM_GAZ.

/ 'LIQU_AD_GAZ_VAPE'

Loi de comportement pour un milieu poreux non saturé eau/vapeur/air sec/air dissous avec changement de phase (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU, THM_VAPE, THM_GAZ et THM_AIR_DISS.

/ 'LIQU_AD_GAZ'

Loi de comportement pour un milieu poreux non saturé eau/air sec/air dissous avec changement de phase (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU, THM_GAZ et THM_AIR_DISS.

/ 'LIQU_VAPE'

Loi de comportement pour un milieux poreux saturé par un composant présent sous forme liquide ou vapeur avec changement de phase (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU et THM_VAPE. **Cette loi n'est valable que pour les modélisations de type THVD**.

/ 'LIQU_GAZ'

Loi de comportement pour un milieu poreux non saturé liquide/gaz sans changement de phase (confer [R7.01.11] pour plus de détails). Les données nécessaires du champ matériau sont fournies dans l'opérateur DEFI_MATERIAU, sous les mots clés THM_LIQU et THM_GAZ.

Le tableau ci-dessous résume les mots clés obligatoires pour les sous commandes explicitées dans les sections suivantes en fonction de la loi de couplage choisie.

Légendes : O Mot clé Obligatoire
T Mot clé obligatoire en Thermique
Mot clé inutile pour ce type de loi de couplage
F Utile pour la loi mais facultatif (défini par défaut)

	LIQU_SATU	LIQU_GAZ	GAZ	LIQU_GAZ_ATM	LIQU_VAPE_GAZ	LIQU_AD_GAZ_VAPE	LIQU_AD_GAZ	LIQU_VAPE
THM_INIT	***O***	***O***	***O***	***O***	***O***	***O***	***O***	***O***
PRE1	F	F	F	F	F	F	F	F
PRE2		F			F	F	F	
PORO	O	O	O	O	O	O	O	O
TEMP	F	F	F	F	F	F	F	F
PRES_VAPE					O	O	O	O
THM_DIFFU	***O***	***O***	***O***	***O***	***O***	***O***	***O***	***O***
R_GAZ		O	O		O	O	O	O
RHO	O	O	O	O	O	O	O	O
BIOT_COEF	O	O	O	O	O	O	O	O
PESA_X	O	O	O	O	O	O	O	O
PESA_Y	O	O	O	O	O	O	O	O
PESA_Z	O	O	O	O	O	O	O	O
Si loi* HYDR_UTIL *(cf. chapitre 2.5) :								
SATU_PRES				O	O	O	O	O
D_SATU_PRES				O	O	O	O	O

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

PERM_LIQU				O	O	O	O	O
D_PERM_LIQU_SATU				O	O	O	O	O
PERM_GAZ					O	O	O	O
D_PERM_SATU_GAZ					O	O	O	O
D_PERM_PRES_GAZ					O	O	O	O
Si loi* HYDR_VGM *ou HYDR_VGC (cf. chapitre 2.5) :								
VG_N					O	O	O	
VG_PR					O	O	O	
VG_SR					O	O	O	
VG_SMAX					O	O	O	
VG_SATUR					O	O	O	
FICKV_T					O	O		
FICKV_PV					F	F		
FICKV_PG					F	F		
FICKV_S					F	F		
D_FV_T					F	F		
D_FV_PG					F	F		
FICKA_T						O	O	
FICKA_PA						F	F	
FICKA_PL						F	F	
FICKA_S						F	F	
D_FA_T						F	F	
CP	T	T	T	T	T	T	T	T
PERM_IN/PERM_END/PERM_L,PERM_N, PERM_T	O	O	O	O	O	O	O	O
LAMB_T/LAMB_TL, LAMB_TN, LAMB_TT	T	T	T	T	T	T	T	T
LAMB_S								
LAMB_PHI								
LAMB_CT/LAMB_C_L,LAMB_C_N,LAMB_C_T								
D_LB_T/D_LB_TT, D_LB_TL,D_LB_T,								
D_LB_S								
D_LB_PHI								
THM_LIQU	***O***	***O***		***O***	***O***	***O***	***O***	***O***
RHO	O	O		O	O	O	O	O
UN_SUR_K	O	O		O	O	O	O	O
VISC	O	O		O	O	O	O	O
D_VISC_TEMP	O	O		O	O	O	O	O
ALPHA	T	T		T	T	T	T	T
CP	T	T		T	T	T	T	T
THM_GAZ		***O***	***O***	***O***	***O***	***O***	***O***	
MASS_MOL		O	O	O	O	O	O	
VISC		O	O	O	O	O	O	
D_VISC_TEMP		O	O	O	O	O	O	
CP		T	T	T	T	T	T	
THM_VAPE_GAZ					***O***	***O***		***O***
MASS_MOL					O	O		O
CP					O	O		O
VISC					O	O		O
D_VISC_TEMP					O	O		O
THM_AIR_DISS						***O***	***O***	
CP						O	O	
COEF_HENRY						O	O	

Le tableau ci-dessous résume la compatibilité entre les lois de comportement et les modélisations chosies :

MODELISATION	**Loi de comportement hydraulique autorisée**
*_HM(_,S,D,SI)	'LIQU_SATU', 'GAZ', 'LIQU_GAZ_ATM'
*_HHM(_,S,D)	'LIQU_GAZ', 'LIQU_VAPE_GAZ'
*_HH2M(S,D,_SI)	'LIQU_AD_GAZ', 'LIQU_AD_GAZ_VAPE'
*_THH(S,D)	'LIQU_GAZ', 'LIQU_VAPE_GAZ'

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

MODELISATION	Loi de comportement hydraulique autorisée
*_THH2(S,D)	'LIQU_AD_GAZ', 'LIQU_AD_GAZ_VAPE'
*_THM(_,S,D)	'LIQU_SATU', 'GAZ', 'LIQU_GAZ_ATM'
*_THV(_,S,D)	'LIQU_VAPE'
*_THHM(S,D)	'LIQU_GAZ', 'LIQU_VAPE_GAZ'
*_THH2M(S,D)	'LIQU_AD_GAZ', 'LIQU_AD_GAZ_VAPE'
*_HH(S,D)	'LIQU_GAZ', 'LIQU_VAPE_GAZ'
*_HH2(S,D)	'LIQU_AD_GAZ', 'LIQU_AD_GAZ_VAPE'

2.2.2 Mot clé facteur THM_INIT

Pour tous les comportements Thermo-Hydro-Mécaniques, il permet de décrire un état de référence de la structure (cf. [R7.01.10] et [R7.01.11]). Sa syntaxe est la suivante :

```
THM_INIT = _F (
       ◊   TEMP      = temp ,      [R]
       ◊   PRE1      = pre1 ,      [R]
       ◊   PRE2      = pre2 ,      [R]
       ♦   PORO      = poro ,      [R]
       ◊   PRES_VAPE = pvap ,      [R]
           )
```

Sous ce mot clé, il y a trois grandeurs correspondant à des degrés de liberté. Pour bien comprendre ces données, il faut alors distinguer les inconnues aux nœuds, que nous appelons $\{u\}^{ddl}$ et les valeurs définies sous le mot clé THM_INIT que nous appelons p^{ref} et T^{ref}

$$\{u\}^{ddl}=\begin{Bmatrix} u_x \\ u_y \\ u_z \\ \text{PRE1}^{ddl} \\ \text{PRE2}^{ddl} \\ T^{ddl} \end{Bmatrix}$$

La signification des inconnues PRE1 et PRE2 varie suivant les modèles. En notant p_w la pression d'eau, p_{ad} la pression d'air dissous, p_{lq} la pression de liquide $p_{lq}=p_w+p_{ad}$, p_{as}, p_{vp} la pression de vapeur, p_{as} la pression d'air sec et $p_g=p_{as}+p_{vp}$ la pression totale de gaz et $p_c=p_g-p_{lq}$ la pression capillaire (aussi appelée succion), on a les significations suivantes des inconnues PRE1 et PRE2

Comportement KIT	LIQU_SATU	LIQU_GAZ_ATM	GAZ	LIQU_VAPE_GAZ
PRE1	p_{lq}	$-p_{lq}$	p_g	$p_c=p_g-p_{lq}$
PRE2				p_g

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

Comportement KIT	LIQU_GAZ	LIQU_VAPE	LIQU_AD_GAZ_VAPE ou LIQU_AD_GAZ
PRE1	$p_c = p_g - p_{lq}$	p_{lq}	$p_c = p_g - p_{lq}$
PRE2	p_g		p_g

Tableau 2.2.3-1 : contenu de PRE1 et PRE2

On définit alors les pressions et la température « totales » par :

$$p = p^{ddl} + p^{ref} \; ; T = T^{ddl} + T^{ref}$$

Toutes les valeurs en entrée ou sortie (conditions aux limites ou résultat de IMPR_RESU) sont les inconnues nodales p^{ddl} et T^{ddl} .

Par contre ce sont les pressions et la température totale qui sont utilisées dans les lois de comportement $\frac{p}{\rho} = \frac{R}{M} T$ pour les gaz parfaits, $\frac{d\rho_l}{\rho_l} = \frac{dp_l}{K_l} - 3\alpha_l dT$ pour le liquide et dans la relation saturation/pression capillaire.
On peut donc choisir d'initialiser les les valeurs nodales dans THM_INIT (par défaut ces valeurs sont nulles) ou par le mot clé ETAT_INIT de la commande STAT_NON_LINE (cf. 2.3) ou par une combinaison des deux (cela peut permettre une facile conversion de Kelvin en degré par exemple).

◊ TEMP

Température de référence T_{ref}. Par défaut elle est prise égale à zéro. Attention la valeur de la température **initiale** $T = T^{ddl} + T^{ref}$ doit être strictement supérieure à zéro.

◊ PRE1

Par défaut elle est prise égale à zéro. Comme vu dans le tableau 1 :
Pour les comportements LIQU_SATU , et LIQU_VAPE pression de liquide de référence.
Pour le comportement GAZ pression de gaz de référence. Dans ce cas la pression **initiale** de gaz $p = p^{ddl} + p^{ref}$ doit être non nulle.
Pour le comportement LIQU_GAZ_ATM pression de liquide de référence changée de signe.
Pour les comportements LIQU_VAPE_GAZ , LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE et LIQU_GAZ pression capillaire de référence.

◊ PRE2

Par défaut elle est prise égale à zéro. Pour les comportements LIQU_VAPE_GAZ, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE et LIQU_GAZ pression de gaz de référence. La pression **initiale** de gaz $p = p^{ddl} + p^{ref}$ doit être non nulle.

♦ PORO

Porosité initiale.

♦ PRES_VAPE

Pression de vapeur de référence pour les comportements : LIQU_VAPE_GAZ, LIQU_AD_GAZ_VAPE et LIQU_VAPE.

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

Remarques :

La pression initiale de vapeur doit être prise en cohérence avec les autres données. Bien souvent, on part de la connaissance d'un état initial d'hygrométrie. Le degré hygrométrique est le rapport entre la pression de vapeur et la pression de vapeur saturante à la température considérée. On utilise alors la loi de Kelvin qui donne la pression du liquide en fonction de la pression de vapeur, de la température et de la pression de vapeur saturante : $\frac{p_w - p_w^{ref}}{\rho_w} = \frac{R}{M_{vp}^{ol}} T \ln\left(\frac{p_{vp}}{p_{vp}^{sat}(T)}\right)$. *Cette relation n'est valable que pour des évolutions isothermes. On souligne que* p_w^{ref} *correspond à un état d"équilibre auquel correspond* p_{pv}^{sat}, *cet état d'équilibre correspond en fait à* $p_w^0 = p_{gz}^0 = 1\,atm$. *Pour des évolutions avec variation de température, connaissant une loi donnant la pression de vapeur saturante à la température* T_0, *par exemple :* $p_{pv}^{sat}(T_0) = 10^{\left(2.7858 + \frac{T_0 - 273.5}{31.559 + 0.1354(T_0 - 273.5)}\right)}$, *et un degré d'hygrométrie* HR, *on en déduit la pression de vapeur grâce à* $p_{pv}(T_0) = HR\, p_{pv}^{sat}(T_0)$.
En outre, il ne faut jamais prendre une valeur de `PRES_VAPE` *égale à zéro.*

2.2.3 Mot clé facteur THM_LIQU

Ce mot clé concerne tous les comportements THM faisant intervenir un liquide (confer [R7.01.11]). Sa syntaxe est la suivante :

```
THM_LIQU = _F (
             ♦  RHO          = rho  ,      [R]
             ♦  UN_SUR_K     = usk  ,      [R]
             ♦  VISC         = vi   ,      [fonction **]
             ♦  D_VISC_TEMP  = dvi  ,      [fonction **]
             ◊  ALPHA        = alp  ,      [R]
             ◊  CP           = cp   ,      [R]
             )
```

♦ RHO

Masse volumique initiale du liquide.

♦ UN_SUR_K

Inverse de la compressibilité du liquide : K_l .

♦ VISC [fonction**]

Viscosité du liquide. Fonction de la température.

♦ D_VISC_TEMP [fonction**]

Dérivée de la viscosité du liquide par rapport à la température. Fonction de la température. L'utilisateur doit assurer la cohérence avec la fonction associée à VISC.

◊ ALPHA

Coefficient de dilatation (linéique) du liquide α_l

Si p_l désigne la pression du liquide, ρ_l sa masse volumique et T la température, le comportement du liquide est : $\frac{d\rho_l}{\rho_l} = \frac{dp_l}{K_l} - 3\,\alpha_l dT$

◊ CP

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

Chaleur massique à pression constante du liquide.

2.2.4 Mot clé facteur THM_GAZ

Ce mot clé facteur concerne tous les comportements THM faisant intervenir un gaz (cf. [R7.01.11]). Pour les comportements faisant intervenir à la fois un liquide et un gaz, et quand on prend en compte l'évaporation du liquide, les coefficients renseignés ici concernent le gaz sec. Les propriétés de la vapeur seront renseignées sous le mot clé THM_VAPE_GAZ. Sa syntaxe est la suivante :

```
THM_GAZ = _F (

      ♦ MASS_MOL         = Mgs  ,      [R]
      ♦ CP               = cp   ,      [R]
      ♦ VISC             = vi   ,      [fonction **]
      ♦ D_VISC_TEMP      = dvi  ,      [fonction **]
          )
```

♦ MASS_MOL

Masse molaire du gaz sec. M_{gs}

Si p_{gs} désigne la pression du gaz sec, ρ_{gs} sa masse volumique, R la constante des gaz parfaits et T la température, le comportement du gaz sec est : $\frac{p_{gs}}{\rho_{gs}}=\frac{RT}{M_{gs}}$.

♦ CP

Chaleur massique à pression constante du gaz sec.

♦ VISC [fonction**]

Viscosité du gaz sec. Fonction de la température.

♦ D_VISC_TEMP [fonction**]

Dérivée par rapport à la température de la viscosité du gaz sec. Fonction de la température. L'utilisateur doit assurer la cohérence avec la fonction associée à VISC.

2.2.5 Mot clé facteur THM_VAPE_GAZ

Ce mot clé facteur concerne tous les comportements THM faisant intervenir à la fois un liquide et un gaz, et prenant en compte l'évaporation du liquide (cf. [R7.01.11]). Les coefficients renseignés ici concernent la vapeur. La syntaxe est la suivante :

```
THM_VAPE_GAZ = _F   (
      ♦ MASS_MOL     = m    ,      [R]
      ♦ CP           = cp   ,      [R]
      ♦ VISC         = vi   ,      [fonction **]
      ♦ D_VISC_TEMP  = dvi  ,      [fonction **]
                )
```

♦ MASS_MOL

Masse molaire de la vapeur. M_{vp}

♦ CP

Chaleur massique à pression constante de la vapeur.

♦ VISC [fonction**]

Viscosité de la vapeur. Fonction de la température.

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

♦ D_VISC_TEMP [fonction**]

Dérivée par rapport à la température de la viscosité de la vapeur. Fonction de la température. L'utilisateur doit assurer la cohérence avec la fonction associée à VISC.

2.2.6 Mot clé facteur THM_AIR_DISS

Ce mot clé facteur concerne le comportement THM THM_AIR_DISS prenant en compte la dissolution de l'air dans le liquide (cf. [R7.01.11]). Les coefficients renseignés ici concernent l'air dissous. La syntaxe est la suivante :

```
THM_AIR_DISS = _F   (
        ♦ CP            = cp   ,   [R]
        ♦ COEF_HENRY    = kh   ,       [fonction **]
                    )
```

♦ CP

Chaleur massique à pression constante de l'air dissous.

♦ COEF_HENRY

Constante de Henry K_H, permettant de relier la concentration molaire d'air dissous C_{ad}^{ol} ($moles/m^3$) à la pression d'air sec :

$$C_{ad}^{ol}=\frac{p_{as}}{K_H}$$

Remarque :

La constante de Henry que nous utilisons ici s'exprime en $Pa.m^3.mol^{-1}$. Dans la littérature il existe différentes manières d'écrire la loi de Henry. Par exemple dans les Benchmarks de l'Andra, la loi de Henry est donnée par $\omega_l^a=\frac{P_{as}}{H}\frac{M_{as}^{ol}}{M_w}\rho_w$ avec la concentration d'air dans l'eau que l'on peut ramener à une masse volumique telle que $\omega_l^a=\rho_{ad}$. H est un coefficient qui s'exprime en Pa . Il faudra dans ces cas écrire l'équivalence $K_H=H\frac{M_w}{\rho_w}$

2.2.7 Mot clé facteur THM_DIFFU

Obligatoire pour tous les comportements THM (cf. [R7.01.11]). L'utilisateur doit s'assurer de la cohérence des fonctions et de leur dérivée. La syntaxe est la suivante :

Obligatoire pour tous les comportements THM (cf. [R7.01.11]). L'utilisateur doit s'assurer de la cohérence des fonctions et de leur dérivée.

Remarque importante concernant l'orthotropie :

L'utilisateur a plusieurs possibilités : se placer dans un cadre isotrope classique ou se placer dans un cas d'isotropie transverse en 3D. Dans ce cas il devra renseigner un certain nombre de données matériaux (cf. ci-dessous) dans les directions L et N du repère d'orthotropie, le repère L,T étant le repère d'isotropie (cf. R4.01.02). Dans le cas 2D, la notion d'isotropie transverse perd son sens et dans ce cas seulement, on se place dans un cadre orthotrope. Pour la partie mécanique élastique on se référera à R4.01.02 et les 3 directions seront à prendre en compte. Pour les termes conductifs (hydrauliques et thermiques), seules les directions L et T du plan seront cette fois à renseigner (correspondant au repère X,Y si le repère global coïncide avec le repère local).

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

Attention l'anisotropie en mécanique n'est valable que pour un comportement mécanique élastique (non programmé pour les autres lois).

-

La syntaxe générale est la suivante :

```
THM_DIFFU = _F   (
       ◊  R_GAZ            = rgaz ,           [R]
       ♦  RHO              = rho  ,           [R]
       ◊  CP               = cp   ,           [R]
       ♦  /BIOT_COEF       = bio  ,           [R]
          /|BIOT_L         = biol,            [R]
           |BIOT_N         = bion,            [R]
          /|BIOT_T         = biol,            [R]
           |BIOT_L         = bion,            [R]
           |BIOT_N         = bion,            [R]
       ♦  PESA_X           = px   ,           [R]
       ♦  PESA_Y           = py   ,           [R]
       ♦  PESA_Z           = pz   ,           [R]
       ♦  /PERM_IN         = perm ,           [fonction]
          /|PERMIN_N          = perml,        [fonction]
           |PERMIN_L          = permn,        [fonction]
          /|PERMIN_T          = perml,        [fonction]
           |PERMIN_L          = permn,        [fonction]
       ◊  /   ◊  SATU_PRES        = sp   ,    [fonction]
              ◊  D_SATU_PRES      = dsp  ,       [fonction]
              ◊  PERM_LIQU        = perml,       [fonction]
              ◊  D_PERM_LIQU_SATU = dperm,       [fonction]
              ◊  PERM_GAZ         = permg,       [fonction]
              ◊  D_PERM_SATU_GAZ  = dpsg ,       [fonction]
              ◊  D_PERM_PRES_GAZ  = dppg ,       [fonction]
           /  ◊  VG_N             = vgn  ,       [R]
              ◊  VG_PR            = pr   ,       [R]
              ◊  VG_SR            = sr   ,       [R]
              ◊  VG_SMAX          = smax ,       [R]
              ◊  VG_SATUR         = stur ,       [R]

       ◊  FICKV_T          =  fvt ,           [fonction]
       ◊  FICKV_PV         =  /   fvpv  ,     [fonction]
                              /  1      ,     [DEFAUT]
       ◊  FICKV_PG         =  /  fvpg   ,     [fonction]
                              /  1      ,     [DEFAUT]
       ◊  FICKV_S          =  /  fvs    ,     [fonction]
                              /  1      ,     [DEFAUT]
       ◊  D_FV_T           =  /  dfvt   ,     [fonction]
                              /  0      ,     [DEFAUT]
       ◊  D_FV_PG          =  /  dfvgp  ,     [fonction]
                              /  0      ,     [DEFAUT]
       ◊  FICKA_T          = fat         ,    [fonction]
       ◊  FICKA_PA         =  /  fapv   ,     [fonction]
                              /  1      ,     [DEFAUT]
       ◊  FICKA_PL         =  /  fapg   ,     [fonction]
                              /  1      ,     [DEFAUT]
       ◊  FICKA_S          =  /  fas ,        [fonction]
                              /  1      ,     [DEFAUT]
       ◊  D_FA_T           =  /  dfat   ,     [fonction]
                              /  0      ,     [DEFAUT]
```


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

```
◊  /LAMB_T          =  /  lambt  ,      [fonction]
                       /  0            [DEFAUT]
   /|LAMB_TN     =  /  lambtl ,      [fonction]
                       /  0            [DEFAUT]
    |LAMB_TL     =  /  lambtn ,      [fonction]
                       /  0            [DEFAUT]
   /|LAMB_TT     =  /  lambtl ,      [fonction]
                       /  0            [DEFAUT]
    |LAMB_TL     =  /  lambtn ,      [fonction]
                       /  0            [DEFAUT]
◊  LAMB_S           =  /  lambs  ,      [fonction]
                       /  1      ,      [DEFAUT]
◊  LAMB_PHI         =  /  lambp  ,      [fonction]
                       /  1      ,      [DEFAUT]
◊  /LAMB_CT         =  /  lambct ,      [fonction]
                       /  0      ,      [DEFAUT]
   /|LAMB_C_N          =  /  lambctl,      [fonction]
                       /  0            [DEFAUT]
    |LAMB_C_L      =  /  lambctn,      [fonction]
                       /  0            [DEFAUT]
   /|LAMB_C_T          =  /  lambctl,      [fonction]
                       /  0            [DEFAUT]
    |LAMB_C_L      =  /  lambctn,      [fonction]
                       /  0            [DEFAUT]
◊  D_LB_S           =  /  dlambs ,      [fonction]
                       /  0      ,      [DEFAUT]
◊  /D_LB_T          =  /  dlambt ,      [fonction]
                       /  0      ,      [DEFAUT]
   /|D_LB_TN       =  /  dlambtl,      [fonction]
                       /  0            [DEFAUT]
    |D_LB_TL       =  /  dlambtn,      [fonction]
                       /  0            [DEFAUT]
   /|D_LB_TT       =  /  dlambtl,      [fonction]
                       /  0            [DEFAUT]
    |D_LB_TL       =  /  dlambtn,      [fonction]
                       /  0            [DEFAUT]
◊  D_LB_PHI         =  /  dlambp ,      [fonction]
                       /  0      ,      [DEFAUT]
◊  SIGMA_T          = st   ,            [fonction]
◊  D_SIGMA_T        = dst  ,            [fonction]
◊  PERM_G_INTR      = pgi  ,            [fonction]
◊  CHAL_VAPO        = cv   ,            [fonction **]
◊  EMMAG            = em   ,            [R]
   )
```

♦ R_GAZ

Constante des gaz parfaits.

♦ RHO

Pour les comportements hydrauliques masse volumique homogénéisée initiale [R7.01.11]. C'est celle qui interviendra dans l'équation d'équilibre de quantité de mouvement et dans la partie hydraulique. Attention dans les cas avec dynamique et uniquement dans ces cas, il faudra également renseigner la masse volumique dans ELAS (l'utilisateur devra donc prendre garde à être cohérent entre ces deux valeurs).

◊ CP

Pour les comportements thermiques, chaleur massique à contrainte constante du solide seul (des grains).

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

Remarque :

Attention il s'agit ici de la chaleur massique uniquement et non pas de « ρC_p », comme c'est fait pour d'autres commandes thermiques. La masse volumique des grains est calculée dans le code à partir de la masse volumique homogénéisée [R7.01.11].

♦ BIOT_COEF

Coefficient de Biot utilisé dans le cas isotrope.

♦ BIOT_L,BIOT_N

Dans le cas isotrope transverse (3D seulement), les coefficients de Biot pour les directions L et N du repère local d'orthotropie. Dans ce cas les deux sont obligatoires.

♦ BIOT_L,BIOT_N,BIOT_T

Dans le cas orthotrope (en 2D seulement) , les coefficients de Biot pour les directions L et N T. Dans ce cas les trois sont obligatoires.

♦ PESA_X, PESA_Y, PESA_Z

Pesanteur selon x, y et z

Remarque :

La pesanteur définie ici est celle utilisée dans l'équation de Darcy uniquement. Quand il y a des calculs mécaniques, la pesanteur est également définie dans AFFE_CHAR_MECA .Cette remarque s'applique bien sur pour les trois composantes de la pesanteur.

◊ PERM_IN [fonction**]

Perméabilité intrinsèque fonction de la porosité dans le cas isotrope.
La perméabilité au sens classique K, dont la dimension est celle d'une vitesse se calcule de la façon suivante :
$K=\frac{K_{\text{int}}K_{rel}}{m}\rho_l g$ où K_{int} est la perméabilité intrinsèque, K_{rel} la perméabilité relative, m la viscosité, ρ_l la masse volumique du liquide et g l'accélération de la pesanteur. Obligatoire dans le cas isotrope.

◊ PERMIN_L,PERMIN_N [fonction**]

Dans le cas isotrope transverse 3D, la perméabilité intrinsèque pour les directions L et N du repère local d'orthotropie. Dans ce cas les deux sont obligatoires.

◊ PERMIN_L,PERMIN_T [fonction**]

Dans le cas orthotrope en 2D , la perméabilité intrinsèque pour les directions L et T du repère local (correspondant à X,Y sans rotation d'angle). Dans ce cas les deux sont obligatoires.

◊ SATU_PRES [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_GAZ_ATM, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), isotherme de saturation fonction de la pression capillaire. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

Remarque :

Pour des raisons numériques, il faut éviter que la saturation atteigne la valeur 1. Aussi il est très fortement recommandé de multiplier la fonction capillaire (comprise généralement entre 0 et 1) par 0,999.

◊ D_SATU_PRES [fonction**]

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

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_GAZ_ATM, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), dérivée de la saturation par rapport à la pression. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ PERM_LIQ [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_GAZ_ATM, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), perméabilité relative au liquide : fonction de la saturation. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ D_PERM_LIQ_SATU [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_GAZ_ATM, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), dérivée de la Perméabilité relative au liquide par rapport à la saturation : fonction de la saturation. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ PERM_GAZ [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), perméabilité relative au gaz : fonction de la saturation et de la pression de gaz. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ D_PERM_SATU_GAZ [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), dérivée de la perméabilité au gaz par rapport a la saturation : fonction de la saturation et de la pression de gaz. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ D_PERM_PRES_GAZ [fonction**]

Pour les comportements de matériaux non saturés (LIQU_VAPE_GAZ, LIQU_VAPE, LIQU_GAZ, LIQU_AD_GAZ, LIQU_AD_GAZ_VAPE), dérivée de la perméabilité au gaz par rapport a la pression de gaz : fonction de la saturation et de la pression de gaz. Uniquement pour les lois de couplage HYDR_UTIL ou HYDR_ENDO (voir section 2.5).

◊ VG_N [I]

Pour les comportements de matériaux non saturés à deux inconnues(LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5), désigne le paramètre N de la loi de Mualem Van-Genuchten servant à définir la pression capillaire et les perméabilités relatives à l'eau et au gaz.

◊ VG_PR [R]

Pour les comportements de matériaux non saturés à deux inconnues (LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5), désigne le paramètre Pr de la loi de Mualem Van-Genuchten servant à définir la pression capillaire et les perméabilités relatives à l'eau et au gaz.

◊ VG_SR [R]

Pour les comportements de matériaux non saturés à deux inconnues (LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5), désigne le paramètre Sr de saturation résiduelle de la loi de Mualem Van-Genuchten servant à définir la pression capillaire et les perméabilités relatives à l'eau et au gaz.

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

◊ VG_PENTR [R]

Pour les comportements de matériaux non saturés à deux inconnues (LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5), désigne le paramètre P_e de pression d'entrée de la loi de Mualem Van-Genuchten. Le milieu ne se désature que lorsque la pression capillaire est supérieure à cette valeur (0 par défaut).

◊ VG_SMAX [R]

Pour les comportements de matériaux non saturés à deux inconnues (LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5), désigne la saturation maximum pour laquelle on applique la loi de Mualem Van-Genuchten. Au delà de cette saturation les courbes de Mualem-Van Genuchten sont interpolées. Cette valeur doit être très proche de 1.

◊ VG_SATUR [R]

Pour les comportements de matériaux non saturés à deux inconnues (LIQU_VAPE_GAZ , LIQU_AD_GAZ , LIQU_AD_GAZ_VAPE , LIQU_GAZ) et dans le cas où la loi hydraulique est HYDR_VGM ou HYDR_VGC (voir section 2.5). Au delà de la saturation définie par VG_SMAX, la saturation est multipliée par ce facteur correctif. Cette valeur doit être très proche de 1 (voir doc. R7.01.11).

◊ FICKV_T [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, partie multiplicative du coefficient de Fick fonction de la température pour la diffusion de la vapeur dans le mélange gazeux. Le coefficient de Fick pouvant être fonction de la saturation , la température, la pression de gaz et la pression de vapeur, on le définit comme un produit de 4 fonctions : FICKV_T, FICKV_S, FICKV_PG, FICKV_VP. Seul FICKV_T est obligatoire pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE. Voir remarque en section 2.2.8.

◊ FICKV_S [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, partie multiplicative du coefficient de Fick fonction de la saturation pour la diffusion de la vapeur dans le mélange gazeux. Dans le cas où on utilise cette fonction, on recommande de prendre FICKV_S(1) = 0. Voir remarque en section 2.2.8.

◊ FICKV_PG [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, partie multiplicative du coefficient de Fick fonction de la pression de gaz pour la diffusion de la vapeur dans le mélange gazeux. Voir remarque en section 2.2.8.

◊ FICKV_PV [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, partie multiplicative du coefficient de Fick fonction de la pression de vapeur pour la diffusion de la vapeur dans le mélange gazeux. Voir remarque en section 2.2.8.

◊ D_FV_T [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, dérivée du coefficient FICKV_T par rapport à la température. Voir remarque en section 2.2.8.

◊ D_FV_PG [fonction**]

Pour les comportements LIQU_VAPE_GAZ et LIQU_AD_GAZ_VAPE, dérivée du coefficient FICKV_PG par rapport à la pression de gaz. Voir remarque en section 2.2.8.

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

◊ FICKA_T [fonction**]

Pour les comportement LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ , partie multiplicative du coefficient de Fick fonction de la température pour la diffusion de l'air dissous dans le mélange liquide. Le coefficient de Fick pouvant être fonction de la saturation , la température, la pression d'air dissous et la pression de liquide, on le définit comme un produit de 4 fonctions : FICKA_T, FICKA_S, FICKV_PA, FICKV_PL. Dans le cas de LIQU_AD_GAZ_VAPE, seul FICKA_T est obligatoire. Voir remarque en section 2.2.8.

◊ FICKA_S [fonction**]

Pour les comportement LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ, partie multiplicative du coefficient de Fick fonction de la saturation pour la diffusion de l'air dissous dans le mélange liquide. Voir remarque en section 2.2.8.

◊ FICKA_PA [fonction**]

Pour les comportement LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ, partie multiplicative du coefficient de Fick fonction de la pression d'air dissous pour la diffusion de l'air dissous dans le mélange liquide. Voir remarque en section 2.2.8.

◊ FICKA_PL [fonction**]

Pour les comportement LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ, partie multiplicative du coefficient de Fick fonction de la pression de liquide pour la diffusion de l'air dissous dans le mélange liquide. Voir remarque en section 2.2.8.

◊ D_FA_T [fonction**]

Pour les comportement LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ, dérivée du coefficient FICKA_T par rapport à la température. Voir remarque en section 2.2.8.

◊ LAMB_T [fonction**]

Partie multiplicative de la conductivité thermique du mélange dépendant de la température (Voir remarque en section 2.2.8.). Cette opérande est obligatoire dans le cas thermique et isotrope.

◊ LAMB_TL,LAMB_TN [fonction**]

Dans le cas isotrope transverse, parties multiplicatives de la conductivité thermique du mélange dépendant de la température pour les directions L et N du repère local d'orthotropie. Ces opérandes sont obligatoires dans le cas où on a de la thermique en isotropie transverse.

◊ LAMB_TL,LAMB_TT [fonction**]

Dans le cas orthotrope 2D, parties multiplicatives de la conductivité thermique du mélange dépendant de la température pour les directions L et T. Ces opérandes sont obligatoires dans le cas où on a de la thermique en orthotropie .

◊ LAMB_S [fonction**]

Partie multiplicative (égale à 1 par défaut) de la conductivité thermique du mélange dépendant de la saturation (Voir remarque en section 2.2.8).

◊ LAMB_PHI [fonction**]

Partie multiplicative (égale à 1 par défaut) de la conductivité thermique du mélange dépendant de la porosité (cf. [§2.2.9]).

◊ LAMB_CT [fonction**]

Partie de la conductivité thermique du mélange constante et additive dans le cas isotrope (cf. [§2.2.9]). Cette constante est égale à zéro par défaut.

◊ LAMB_C_L,LAMB_C_N [fonction**]

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

Dans le cas isotrope transverse, parties de la conductivité thermique du mélange constantes et additives (cf. [§2.2.9]) pour les directions L et N du repère local d'orthotropie. Ces constantes sont égales à zéro par défaut.

◊ LAMB_C_L,LAMB_C_T [fonction**]

Dans le cas orthotrope 2D, parties de la conductivité thermique du mélange constantes et additives (cf. [§2.2.9]) pour les directions L et T du repère local. Ces constantes sont égales à zéro par défaut.

◊ D_LB_T [fonction**]

Dérivée de la partie de la conductivité thermique du mélange dépendant de la température par rapport à la température dans le cas isotrope.

◊ D_LB_TL,D_LB_TN [fonction**]

Dans le cas isotrope transverse, dérivées des parties de la conductivité thermique du mélange dépendant de la température par rapport à la température, pour les directions L et N du repère local d'orthotropie.

◊ D_LB_TL,D_LB_TT [fonction**]

Dans le cas orthotrope 2D, dérivées des parties de la conductivité thermique du mélange dépendant de la température par rapport à la température, pour les directions L et TN du repère local d'orthotropie.

◊ D_LB_S [fonction**]

Dérivée de la partie de la conductivité thermique du mélange dépendant de la saturation.

◊ D_LB_PHI [fonction**]

Dérivée de la partie de la conductivité thermique du mélange dépendant de la porosité.

◊ EMMAG [fonction**]

Coefficient d'emmagasinement. Ce coefficient n'est pris en compte que dans les cas des modélisations sans mécanique.

Remarque :

Attention il est important de rappeler à l'utilisateur que les paramètres BIOT_COEF et BIOT_L, BIOT_N sont incompatibles pour une même modélisation. L'utilisateur doit renseigner le paramètre BIOT_COEF si il choisit de mener une étude dans le cas isotrope, les paramètres BIOT_L, BIOT_N si il choisit de mener son étude en isotrope transverse, ou BIOT_L, BIOT_N et BIOT_T dans le cas orthotrope 2D. La même règle est applicable pour les paramètres PERM_IN, LAMB_T, D_LB_T et LAMB_CT. Pour ces termes conductifs, dans le cas 2D orthotrope, seuls les composants L et T sont requis.

2.2.8 Récapitulatif des fonctions de couplages et leur dépendance

Les tableaux ci-dessous rappellent les différentes fonctions et leurs éventuelles dépendances et obligations.

Mot clé facteur THM_LIQU

♦	RHO	ρ_{lq}^{0}
♦	UN_SUR_K	$\frac{1}{K_{lq}}$
◊	ALPHA	α_{lq}
◊	CP	C_{lq}^{p}

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

♦	VISC	$\mu_{lq}(T)$
♦	D_VISC_TEMP	$\frac{\partial \mu_{lq}(T)}{\partial T}$

Mot clé facteur THM_GAZ

♦	MASS_MOL	M_{as}^{ol}
◊	CP	C_{as}^{p}
♦	VISC	$\mu_{as}(T)$
♦	D_VISC_TEMP	$\frac{\partial \mu_{as}(T)}{\partial T}$

Mot clé facteur THM_VAPE_GAZ

♦	MASS_MOL	M_{VP}^{ol}
♦	CP	C_{vp}^{p}
♦	VISC	$\mu_{vp}(T)$
♦	D_VISC_TEMP	$\frac{\partial \mu_{vp}(T)}{\partial T}$

Mot clé facteur THM_AIR_DISS

♦	CP	C_{ad}^{p}
♦	COEF_HENRY	K_H

Mot clé facteur THM_INIT

◊	TEMP	^{init}T
◊	PRE1	$^{init}P_1$
◊	PRE2	$^{init}P_2$
♦	PORO	ϕ^0
◊	PRES_VAPE	p_{vp}^0

•Mot clé facteur THM_DIFFU

◊	R_GAZ	R
♦	RHO	r_0
◊	CP	C_{σ}^{s}
◊	BIOT_COEF	b
◊	BIOT_L	b_L
◊	BIOT_N	b_N
◊	BIOT_T	b_T
◊	SATU_PRES	$S_{lq}(p_c)$

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

◊	D_SATU_PRES	$\frac{\partial S_{lq}(p_c)}{\partial p_c}$
♦	PESA_X	F_x^m
♦	PESA_Y	F_y^m
♦	PESA_Z	F_z^m
◊	PERM_IN	$K^{\text{int}}(\phi)$
◊	PERMIN_L	$K^{\text{int}}(\phi)$ selon L
◊	PERMIN_N	$K^{\text{int}}(\phi)$ selon N
◊	PERMIN_T	$K^{\text{int}}(\phi)$ selon T (2D)
◊	PERM_LIQU	$k_{lq}^{rel}(S_{lq})$
◊	D_PERM_LIQU_SATU	$\frac{\partial k_{lq}^{rel}(S_{lq})}{\partial S_{lq}}$
◊	PERM_GAZ	$k_{gz}^{rel}(S_{lq}, p_{gz})$
◊	D_PERM_SATU_GAZ	$\frac{\partial k_{gz}^{rel}(S_{lq}, p_{gz})}{\partial S_{lq}}$
◊	D_PERM_PRES_GAZ	$\frac{\partial k_{gz}^{rel}(S_{lq}, p_{gz})}{\partial p_{gz}}$
◊	FICKV_T	$f_{vp}^T(T$
◊	FICKV_S	$f_{vp}^S(S$
◊	FICKV_PG	$f_{vp}^{gz}(P_g)$
◊	FICKV_PV	$f_{vp}^{vp}(P_{vp})$
◊	D_FV_T	$\frac{\partial f_{vp}^T(T}{\partial T}$
◊	D_FV_PG	$\frac{\partial f_{vp}^{gz}(P_{gz}}{\partial P_{gz}}$
◊	FICKA_T	$f_{ad}^T(T$
◊	FICKA_S	$f_{ad}^S(S$
◊	FICKA_PA	$f_{ad}^{ad}(P_{ad})$
◊	FICKA_PL	$f_{ad}^{lq}(P_{lq}$
◊	D_FA_T	$\frac{\partial f_{ad}^T(T}{\partial T}$
◊	LAMB_T	$\lambda_T^T(T)$
◊	LAMB_TL	$\lambda_T^T(T)$ selon L
◊	LAMB_TN	$\lambda_T^T(T)$ selon N
◊	LAMB_TT	$\lambda_T^T(T)$ selon T (2D)

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

◊	D_LB_T	$\frac{\partial \lambda_T^T(T)}{\partial T}$
◊	D_LB_TL	$\frac{\partial \lambda_T^T(T)}{\partial T}$ selon L
◊	D_LB_TN	$\frac{\partial \lambda_T^T(T)}{\partial T}$ selon N
◊	D_LB_TT	$\frac{\partial \lambda_T^T(T)}{\partial T}$ selon T (2D)
◊	LAMB_PHI	$\lambda_\varphi^T(\varphi)$
◊	D_LB_PHI	$\frac{\partial \lambda_\varphi^T(\phi)}{\partial \phi}$
◊	LAMB_S	$\lambda_S^T(S)$
◊	D_LB _S	$\frac{\partial \lambda_S^T(S)}{\partial S}$
◊	LAMB_CT	λ_{CT}^T
◊	LAMB_C_L	λ_{CT}^T selon L
◊	LAMB_C_N	λ_{CT}^T selon N
◊	LAMB_C_T	λ_{CT}^T selon T (2D)

Remarque :

Dans le cas où il y a de la thermique :
$\boldsymbol{\lambda}^T$ *est fonction de la porosité, de la saturation et de la température et est donnée sous la forme du produit de trois fonctions :*

$\boldsymbol{\lambda}^T = \lambda_\varphi^T(\phi).\lambda_S^T(S_{lq}).\boldsymbol{\lambda}_T^T(T) + \boldsymbol{\lambda}_{cte}^T$ *avec le tenseur* $\boldsymbol{\lambda}_T^T(T)$ *obligatoire et les autres fonctions par défaut prises égales à un, sauf le tenseur* $\boldsymbol{\lambda}_{cte}^T = \mathbf{0}$.

Pour le coefficient de Fick du mélange gazeux, dans le cas LIQU_VAPE_GAZ *et* LIQU_AD_GAZ_VAPE $F_{vp}(P_{vp}, P_{gz}, T, S) = f_{vp}^{vp}(P_{vp}).f_{vp}^{gz}(P_{gz}).f_{vp}^T(T).f_{vp}^S(S)$ *avec* $f_{vp}^T(T)$ *obligatoire, les autres fonctions étant prises par défaut égales à un, et les dérivées égales à zéro. On négligera les dérivées par rapport à la pression de vapeur et à la saturation.*

Dans les cas LIQU_AD_GAZ_VAPE et LIQU_AD_GAZ, *le coefficient de Fick du mélange liquide sera sous la forme :* $F_{ad}(P_{ad}, P_{lq}, T, S) = f_{ad}^{ad}(P_{ad}).f_{ad}^{lq}(P_{lq}).f_{ad}^T(T).f_{ad}^S(S)$ *, avec* $f_{ad}^T(T)$ *obligatoire, les autres fonctions étant prises par défaut égales à un, et la dérivée égale à zéro. On ne considère que la dérivée par rapport à la température (les autres sont de toutes façons prises égales à zéro).*

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

2.3 Initialisation du calcul

Pour définir un état initial, il faut définir un état de contraintes généralisées (aux éléments), des inconnues nodales et des variables internes.

- **Dans le mot clé THM_INIT de DEFI_MATERIAU , on définit des valeurs initiales pour la porosité et la pression de vapeur (lorsque le modèle tient compte de la vapeur) mais aussi si on le souhaite de référence pour les inconnues nodales.**
- **Par le mot clé DEPL du mot clé facteur ETAT_INIT de la commande STAT_NON_LINE, on affecte le champ d'initialisation des inconnues nodales. (les valeurs physiques d'initialisations seront la somme des valeurs définies sous THM_INIT et sous ETAT_INIT).**
- **Par le mot clé SIGM du mot clé facteur ETAT_INIT de la commande STAT_NON_LINE , on affecte le champs d'initialisation des contraintes.**
- **Par le mot clé VARI du mot clé facteur ETAT_INIT on affecte (éventuellement) le champs d'initialisation des variables internes.**

Afin de préciser les choses, on rappelle à quelle catégorie de variables appartient chaque grandeur physique (ces grandeurs physiques existant ou non suivant la modélisation choisie) :

Inconnues nodales	$p_c, p_g, p_{lq}, T, u_x, u_y, u_z$
Contraintes aux points de Gauss	$\sigma'_{xx}, \sigma'_{yy}, \sigma'_{zz}, \sigma'_{xy}, \sigma'_{xz}, \sigma'_{yz}, \sigma_{p_{xx}}, \sigma_{p_{yy}}, \sigma_{p_{zz}}, \sigma_{p_{xy}}, \sigma_{p_{xz}}, \sigma_{p_{yz}},$ $m_w, \mathrm{M}_{w_x}, \mathrm{M}_{w_y} M_{w_z}, m_{vp}, \mathrm{M}_{vp_x}, \mathrm{M}_{vp_y} M_{vp_z}, m_{as}, \mathrm{M}_{as_x}, \mathrm{M}_{as_y} M_{as_z},$ $m_{ad}, \mathrm{M}_{ad_x}, \mathrm{M}_{ad_y} M_{ad_z}, h^m_w, h^m_{vp}, h^m_{as}, h^m_{ad}, Q', q_x, q_y, q_z$
Variables internes	$\phi, \rho_{lq}, p_{vp}, S_{lq}$

La correspondance entre nom de composante Aster et grandeur physique est explicitée en [§Annexe 1].
L'initialisation des inconnues nodales ainsi que la différence entre état initial et état de référence ont été décrites et détaillées dans la section 2.2.2. On rappelle néanmoins que $p = p^{ddl} + p^{ref}$ pour les pressions PRE1 et PRE2 et $T = T^{ddl} + T^{ref}$ pour les températures, où p^{ref} et T^{ref} sont définis sous le mot clé THM_INIT de la commande DEFI_MATERIAU.

Le mot clé DEPL du mot clé facteur ETAT_INIT de la commande STAT_NON_LINE définit les valeurs initiales de $\{u\}^{ddl}$. Les valeurs initiales des masses volumiques de la vapeur et de l'air sec sont définies à partir des valeurs initiales des pressions de gaz et de vapeur. On remarque que, pour les déplacements, la décomposition $u = u^{ddl} + u^{ref}$ n'est pas faite : le mot clé THM_INIT de la commande DEFI_MATERIAU ne permet donc pas de définir des déplacements initiaux. La seule façon d'initialiser les déplacements est donc de leur donner une valeur initiale par le mot clé facteur ETAT_INIT de la commande STAT_NON_LINE.

Concernant les contraintes, les champs à renseigner sont les contraintes indiquées en annexe I suivant la modélisation choisie.
Les valeurs initiales des enthalpies, qui appartiennent aux contraintes généralisées sont définies à partir du mot clé SIGM du mot clé facteur ETAT_INIT de la commande STAT_NON_LINE.
L'introduction des conditions initiales est très importante, pour les enthalpies. En pratique, on peut raisonner en considérant que l'on a trois états pour les fluides :

- l'état courant,
- l'état de référence : c'est celui des fluides à l'état libre. Dans cet état de référence, on peut considérer que les enthalpies sont nulles,

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

- l'état initial : il doit être en équilibre thermodynamique. Pour les enthalpies de l'eau et de la vapeur, on devra prendre :

$$
\begin{aligned}
&{}^{init}h_w^m = \frac{p_w^{init} - p_l^{ref}}{\rho_w} = \frac{p_w^{init} - p_{atm}}{\rho_w} \\
&{}^{init}h_{vp}^m = L(T^{init}) = \text{chaleur latente de vaporisation} \\
&{}^{init}h_{as}^m = 0 \\
&{}^{init}h_{ad}^m = 0
\end{aligned}
$$

et avec $L(T) = 2500800 - 2443(T - 273.15)\ \mathrm{J/kg}$

Remarque :

La pression initiale de vapeur devra être prise en cohérence avec ces choix (cf 2.2.3).

Concernant les contraintes mécaniques, la partition des contraintes en contraintes totale et effective s'écrit :

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' + \boldsymbol{\sigma}_p$$

où le tenseur $\boldsymbol{\sigma}$ est la contrainte totale, c.a.d celle qui vérifie : $\mathbf{Div}(\boldsymbol{\sigma}) + r\boldsymbol{F}^m = 0$

$\boldsymbol{\sigma}'$ est la contrainte effective. Pour les lois de contraintes effectives : $d\boldsymbol{\sigma}' = f(\boldsymbol{d\varepsilon} - \boldsymbol{\alpha}_0 dT, \alpha)$, où $\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla u} + {}^T\boldsymbol{\nabla u})$ et α représente les variables internes.

Les composantes du tenseur $\boldsymbol{\sigma}_p$ se calculent en fonction des pressions hydrauliques. L'écriture adoptée est incrémentale et, si l'on veut que les valeurs des composantes de $\boldsymbol{\sigma}_p$ soient cohérentes avec les valeurs initiales de pressions définies sous les mots-clés `THM_INIT` et `ETAT_INIT`, il faut initialiser les composantes de $\boldsymbol{\sigma}_p$ par le mot clé `SIGM` du mot clé facteur `ETAT_INIT` de la commande `STAT_NON_LINE`.

Exemple :

Les champs de déplacements initialisés dans `ETAT_INIT` peuvent être définis de la manière suivante :

```
CHAMNO=CREA_CHAMP(MAILLAGE=MAIL,
                  OPERATION='AFFE',
                  TYPE_CHAM='NOEU_DEPL_R',
                  AFFE=(_F(TOUT='OUI',
                           NOM_CMP='TEMP',
                           VALE=0.0,),
                        _F(GROUP_NO='SURFBO',
                           NOM_CMP='PRE1',
                           VALE=7.E7,),
                        _F(GROUP_NO='SURFBG',
                           NOM_CMP='PRE1',
                           VALE=3.E7,),
                        _F(GROUP_NO='SURFBO',
                           NOM_CMP='PRE2',
                           VALE=0.0,),
                        _F(GROUP_NO='SURFBG',
                           NOM_CMP='PRE2',
                           VALE=0.0,),),);
```


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

Et les champs de contrainte de la manière suivante:

```
SIGINIT=CREA_CHAMP(MAILLAGE=MAIL,
                   OPERATION='AFFE',
                  TYPE_CHAM='CART_SIEF_R',
                  AFFE=(_F(GROUP_MA='BO',
                           NOM_CMP=
 ('SIXX','SIYY','SIZZ','SIXY','SIXZ',
  'SIYZ','SIPXX','SIPYY','SIPZZ','SIPXY',
  'SIPXZ','SIPYZ',
  'M11','FH11X','FH11Y','ENT11',
  'M12','FH12X','FH12Y','ENT12',
  'QPRIM','FHTX','FHTY','M21',
  'FH21X','FH21Y','ENT21',
  'M22','FH22X', 'FH22Y','ENT22',),
  VALE=
  (0.0,0.0,0.0,0.0,0.0,
   0.0,0.0,0.0,0.0,0.0,
   0.0,0.0,
   0.0,0.0,0.0,0.0,
   0.0,0.0,0.0, 2500000.0,
   0.0,0.0,0.0,0.0,0.0,0.0,0.0,
   0.,0.,0.,0.),),),);
```

2.4 Chargements et conditions aux limites

Toutes les conditions aux limites ou chargement sont affectés via la commande AFFE_CHAR_MECA [U4.44.01]. Les chargements sont ensuite activés par le mot clé facteur EXCIT de la commande STAT_NON_LINE.

De manière classique, trois types de conditions aux limites sont possibles :

- Des conditions de type Dirichlet qui consistent à imposer sur une partie de frontière des valeurs fixées pour des inconnues principales appartenant à $\{u\}^{ddl}$ (et non $u = u^{ddl} + u^{init}$) pour cela on utilise le mot clé facteur DDL_IMPO ou FACE_IMPO de AFFE_CHAR_MECA.
- Des conditions de type Neumann qui consistent à imposer des valeurs aux « quantités duales », soit en ne disant rien (flux nuls en hydraulique et thermique), soit en leur donnant une valeur via les mots clés FLUN, FLUN_HYDR1 et FLUN_HYDR2 du mot clé facteur FLUX_THM_REP de la commande AFFE_CHAR_MECA. Ce flux est ensuite multiplié par une fonction du temps (par défaut égale à 1) appelée par FONC_MULT dans le sous mot clé EXCIT de la commande STAT_NON_LINE. FLUN, FLUN_HYDR1 et FLUN_HYDR2 représentent respectivement les flux thermiques, les flux d'eau et les flux de composant gazeux (cf, fin du paragraphe).
- Des conditions de type mixtes (ou « d'échange ») qui consistent – en non saturé - à imposer une condition « extérieure » pour chaque inconnue principale d'hydraulique (p_c, p_g) ainsi que les coefficients d'échange correspondant. In fine cela consiste à imposer une valeur aux « quantités duales » fonction des pressions extérieures et des coefficients d'échange (de manière linéaire). Les pressions extérieures et les coefficients d'échanges sont renseignés respectivement par les mots clés PRE1_EXT et PRE2_EXT et COEF_11, COEF_12, COEF_21, COEF_22, en utilisant le mot clé facteur ECHANGE_THM de la commande AFFE_CHAR_MECA.

- Les conditions mécaniques en contraintes totales $\sigma . n$ sont elles données via PRES_REP de la commande AFFE_CHAR_MECA. On se référera à la documentation de cette commande pour en connaître les possibilités.

D'un point de vue syntaxique les conditions de Dirichlet s'appliquent donc comme sur l'exemple suivant

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

```
DIRI=AFFE_CHAR_MECA(MODELE=MODELE,
                        DDL_IMPO=(_F(GROUP_NO=’GAUCHE’,
                                     TEMP=0.0,),
                                  _F(TOUT='OUI',
                                     PRE2=0.0,),
                                  _F(GROUP_NO=’GAUCHE’,
                                     PRE1=0.0,),
                                  _F(TOUT='OUI',
                                     DX=0.0,),
                                  _F(TOUT='OUI',
                                     DY=0.0,),
                                  _F(TOUT='OUI',
                                     DZ=0.0,),
),)
```

Pour les conditions de Neuman, la syntaxe sera alors comme sur l’exemple suivant :

```
NEU1=AFFE_CHAR_MECA(MODELE=MODELE,
                      FLUX_THM_REP=_F(GROUP_MA='DROIT',
                               FLUN=200.,
                               FLUN_HYDR1=0.0,
                               FLUN_HYDR2=0.0),);
NEU2=AFFE_CHAR_MECA(MODELE=MODELE,
                      PRES_REP=_F(GROUP_MA='DROIT',
                               PRES=2.,),);
```

On définit ensuite la fonction multiplicative que l’on veut appliquer, par exemple à $NEU1$:

```
FLUX=DEFI_FONCTION(NOM_PARA='INST',
                   VALE=
                   (0.0, 386.0,
                   315360000.0, 312.0,
                   9460800000.0,12.6),);
```

Les chargements sont ensuite activés dans STAT_NON_LINE via le mot clé EXCIT de la manière suivante :

```
                EXCIT=(
                        _F(CHARGE=DIRI,),
                        _F(CHARGE=NEU2,),
                        _F(CHARGE=NEU1,
                          FONC_MULT=FLUX,),
                        ),
```

FLUN correspond à la valeur du flux de chaleur ; FLUN_HYDR1 et FLUN_HYDR2 correspondent aux valeurs des flux hydrauliques associés aux pressions $PRE1$ et $PRE2$. S’il n’y a pas d’ambiguïté pour la thermique ou la mécanique, en revanche les inconnues principales hydrauliques PRE1 et PRE2 changent suivant le couplage choisi. Comme on le rappelle ci-dessous

Comportement	LIQU_SATU	LIQU_VAPE	LIQU_GAZ_ATM	GAZ	LIQU_VAPE_GAZ LIQU_GAZ LIQU_AD_GAZ_VAPE LIQU_AD_GAZ
$PRE1$	p_{lq}	p_{lq}	$-p_{lq}$	p_g	$p_c = p_g - p_{lq}$
$PRE2$					p_g

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

Les flux associés sont :

Pour PRE1, FLUN_HYDR1 : $\left(M_w+M_{vp}\right).n=M_w^{ext}+M_{vp}^{ext}$
Pour PRE2, FLUN_HYDR2 : $\left(M_{ad}+M_{as}\right).n=M_{ad}^{ext}+M_{as}^{ext}$

Pour les conditions d'échange, qui n'existe que pour les modélisations de type non saturé *HH2*, la syntaxe sera alors comme sur l'exemple suivant :

```
ECH=AFFE_CHAR_MECA(MODELE=MODELE,
                       ECHANGE_THM=_F(GROUP_MA='DROIT',
                                 COEF_11=C11, COEF_12=C12,PRE1_EXT=1.E6,
                                 COEF_21= C21 , COEF_22= C22 ,PRE2_EXT=1.E5,
                                ),);
```

Les chargements sont ensuite activés dans STAT_NON_LINE via le mot clé EXCIT de la manière suivante :

```
                  EXCIT=(
                          _F(CHARGE= ECH, TYPE_CHARGE='SUIV' ),
                         ),
```

Les flux associés sont alors:

Pour PRE1, $\left(M_w+M_{vp}\right).n=C11\left(p_1-p_1^{ext}\right)-C12\left(p_2-p_2^{ext}\right)$
Pour PRE2 : $\left(M_{ad}+M_{as}\right).n=C21\left(p_1-p_1^{ext}\right)-C22\left(p_2-p_2^{ext}\right)$

Nous allons résumer les diverses possibilités en distinguant le cas où on impose des valeurs à PRE1 et/ou PRE2 et celui où on travaille sur des combinaisons des 2. On signale qu'on peut bien sûr avoir différents types de conditions aux limites suivant les morceaux de frontière (groupes de nœuds ou de mailles) que l'on traite. Pour un aperçu plus complet et plus détaillé de la manière dont sont traitées les conditions aux limites dans le cas non saturé, on se reportera à la note reproduite dans l'annexe 2.

- **Cas des conditions aux limites faisant intervenir les inconnues principales PRE1 et PRE2**

  On résume ici le cas habituel où on impose des valeurs à PRE1 et/ou PRE2.

  - Dirichlet sur PRE1 et Dirichlet sur $PRE2$
    L'utilisateur impose une valeur à $PRE1$ et $PRE2$; les flux sont des résultats de calcul.
  - Dirichlet sur PRE1 et Neuman sur $PRE2$
    L'utilisateur impose une valeur à PRE1 et une valeur au flux associé à PRE2 en ne disant rien sur PRE2 ou en donnant une valeur à FLUN_HYDR2.
  - Dirichlet sur PRE2 et Neuman sur PRE1
    L'utilisateur impose une valeur à PRE2 et une valeur au flux associé à PRE1 en ne disant rien sur PRE1 ou en donnant une valeur à FLUN_HYDR1.
  - Neuman sur PRE2 et Neuman sur PRE1
    Les deux flux sont imposés soit en ne disant rien sur PRE1 et/ou PRE2 (flux nuls) soit en donnant une valeur à FLUN_HYDR1 et/ou FLUN_HYDR2

- **Cas des conditions aux limites faisant intervenir une relation linéaire entre les inconnues principales PRE1 et PRE2**

  Il est également possible de manier des combinaisons linéaires de PRE1 et PRE2. Il faut toutefois manier cela avec précaution de manière à partir d'un problème correctement posé. La syntaxe de cet opérateur est détaillée dans la documentation de AFFE_CHAR_MECA, l'exemple ci-dessous illustre ce type de condition:

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

```
P_DDL=AFFE_CHAR_MECA(MODELE=MODELE,
                     LIAISON_GROUP=(_F(
                                  GROUP_NO_1= 'BORDS',
                                  GROUP_NO_2= 'BORDS',
                                  DDL_1='PRE1',
                                  DDL_2='PRE2',
                                  COEF_MULT_1 = x,
                                  COEF_MULT_2 = y.,
                                  COEF_IMPO =z,),),
                                );
```

Cette commande signifie que sur la frontière définie par le groupe de nœuds $BORDS$, les pressions PRE1 et PRE2 sont reliées par la relation linéaire

$$x\,PRE1 + y\,PRE2 = z$$

Remarque :

Les flux imposés sont des quantités scalaires qui peuvent s'appliquer sur une ligne ou une surface interne au solide modélisé. Dans ce cas, ces conditions aux limites correspondent à une source.

2.5 Le calcul non linéaire

La résolution peut être effectuée de 2 manières :

1. la méthode du couplage : c'est la plus fiable, la plus robuste et la plus ancienne
2. la méthode du chaînage : c'est une nouvelle stratégie, adaptée pour les problèmes faiblement couplés et pour les problèmes ne pouvant pas être résolus actuellement par la méthode du couplage.
   Ex :
   - calcul avec la loi de comportement mécanique 'ENDO_HETEROGENE', qui n'est pas disponible dans le kit THM
   - calcul avec les modélisations hydrauliques diphasiques en volumes finis, qui ne sont pas disponibles dans le kit THM couplées avec la mécanique

Le cœur de la résolution est la commande 'STAT_NON_LINE'. A cette commande on affecte le modèle (mot clé MODELE), les matériaux (mot clé CHAM_MATER), le/les chargements (mot clé EXCIT) et l'état initial (mot clé ETAT_INIT) que l'on a définis par toutes les commandes décrites précédemment.

Pour les informations générales concernant cette commande et sa syntaxe, on se reportera à sa documentation. On précise juste que la méthode de calcul est une méthode de Newton. On ne parle donc ci-dessous que de ce qui est spécifique aux calculs THM à savoir les mots clés facteurs RELATION et RELATION_KIT du mot clé COMPORTEMENT qui sont étroitement liés.

Attention :

Sous le mot clé facteur NEWTON, on doit mettre une matrice de type 'TANGENTE' et non 'ELASTIQUE' .

On présente d'abord des exemples d'utilisation pour les deux stratégies de résolution.

2.5.1 Résolution par couplage

Dans ce cas, la résolution est effectuée par une seule commande STAT_NON_LINE :

```
U0=STAT_NON_LINE(MODELE=MODELE,
                 CHAM_MATER=CHMAT0,
                 EXCIT=(
                        _F(CHARGE=T_IMP,),
```


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

```
        _F(CHARGE=CALINT,
           FONC_MULT=FLUX,),),
COMPORTEMENT=_F(RELATION='KIT_THHM',
                RELATION_KIT=
             ('ELAS','LIQU_GAZ' ,'HYDR_UTIL'),),
ETAT_INIT=_F(DEPL=CHAMNO,
             SIGM=SIGINIT),
INCREMENT=_F(LIST_INST=INST1,),
NEWTON=_F(MATRICE='TANGENTE',REAC_ITER=1,),
CONVERGENCE=_F(RESI_GLOB_RELA=1.E-06,
                ITER_GLOB_MAXI=15,
                ITER_INTE_MAXI=5,),
ARCHIVAGE=_F(PAS_ARCH=1,),);
```

2.5.2 Résolution par chaînage

La résolution par chaînage s'applique actuellement aux problèmes Hydro-Mécaniques saturés uniquement. On rappelle d'abord les équations de ce problème :

$$-\nabla\cdot\sigma'(u)+b\nabla p=0$$
$$\partial_t(\rho\phi)+\nabla\cdot M_{lq}=0$$

où la porosité ϕ suit la loi de comportement $d\phi=bd(\nabla\cdot u)+\frac{b-\phi}{K_s}dp$.

Dans le cadre du chaînage, on résout séparément les problèmes mécanique et hydraulique. La communication entre les deux calculs s'effectue via des variables de commande (`'PTOT'` et `'DIVU'`). Pour le problème mécanique, la variable de commande `'PTOT'` joue le rôle de la donnée de pression . Pour le problème hydraulique, la variable de commande `'DIVU'` joue le rôle de la donnée de déformation volumique.
Ces variables de commande sont calculées par la commande `CHAINAGE_THM`.

La méthode illustrée ci-dessous est un algorithme basique de chaînage pur de calculs (résolution hydraulique à l'instant 1, résolution mécanique à l'instant 1, résolution hydraulique à l'instant 2, résolution mécanique à l'instant 2, etc …). On pourrait envisager des algorithmes plus compliqués (de type point fixe ou gradient conjugué non linéaire) mais cela complexifie d'autant la mise en données dans le fichier de commandes. La question de la convergence de tels algorithmes n'est pas non plus du tout acquise !
La solution mécanique finale est `DEPLA` et la solution hydraulique est `PRELIQ`.

```
# boucle sur les pas de temps
for k in range(1,lenlinst) :

   UNPAS=DEFI_LIST_REEL(DEBUT=listinst[k-1],
                        INTERVALLE=_F(JUSQU_A=listinst[k],
                        NOMBRE=1,),);

   UNPASSUB =DEFI_LIST_INST(DEFI_LIST=_F(LIST_INST =UNPAS ),
                                    ECHEC=_F(
                                            SUBD_METHODE='MANUEL',
                                            SUBD_PAS  = 4,
                                            SUBD_NIVEAU=5,),);

   if (k>1) :
```


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

```
            DEFVHY=CHAINAGE_THM(RESULTAT=DEPLA,INFO=1,
                                MODELE=MODHYD,
                                MATR_PROJECTION=MATPROJ,
                                TYPE_CHAINAGE='MECA_HYDR',
                                INST=listinst[k],);

         MATEHY=AFFE_MATERIAU(MAILLAGE=MAIL,
                                AFFE=_F(TOUT='OUI',
                                        MATER=MATERIAU,),
                                AFFE_VARC=_F(NOM_VARC='DIVU',
                                             EVOL=DEFVHY,),);

      else :

         MATEHY=AFFE_MATERIAU(MAILLAGE=MAIL,
                              AFFE=_F(TOUT='OUI',
                                      MATER=MATERIAU,),);

      if (k==1) :

         PRELIQ=STAT_NON_LINE(MODELE=MODHYD,
                              CHAM_MATER=MATEHY,
                              EXCIT=_F(CHARGE=CHARHYD,),
                              SOLVEUR=_F(METHODE='MUMPS',PCENT_PIVOT=100,),
                              CONVERGENCE=_F(ITER_GLOB_MAXI=50,),
                              ARCHIVAGE=_F(LIST_INST=UNPAS,),
                              COMPORTEMENT=_F(RELATION='KIT_H',
                                        RELATION_KIT=('LIQU_SATU','HYDR_UTIL',),),),
                              INCREMENT=_F(LIST_INST=UNPASSUB,),);

      else :

          PRELIQ=STAT_NON_LINE(reuse=PRELIQ,
                              MODELE=MODHYD,
                              CHAM_MATER=MATEHY,
                              EXCIT=_F(CHARGE=CHARHYD,),
                              SOLVEUR=_F(METHODE='MUMPS',PCENT_PIVOT=100,),
                              ARCHIVAGE=_F(LIST_INST=UNPAS,),
                              ETAT_INIT=_F(EVOL_NOLI=PRELIQ,
                                           INST=listinst[k-1]),
                              CONVERGENCE=_F(ITER_GLOB_MAXI=50,),
                              COMPORTEMENT=_F(RELATION='KIT_H',
                                        RELATION_KIT=('LIQU_SATU','HYDR_UTIL',),),),
                              INCREMENT=_F(LIST_INST=UNPASSUB,),);

      if (k>1) :
         DETRUIRE(INFO=1,CONCEPT=_F(NOM=(REPTOT,MATEME,),),);

      REPTOT=CHAINAGE_THM(RESULTAT=PRELIQ,
                          MODELE=MODMEC,
                          TYPE_CHAINAGE='HYDR_MECA',
                          TYPE_RESU='EVOL_VARC',
                          INST=listinst[k],);

      MATEME=AFFE_MATERIAU(MAILLAGE=MAIL,
                           AFFE=_F(TOUT='OUI',
                                   MATER=MATERIAU,),
                           AFFE_VARC=_F(TOUT='OUI',
                                        EVOL=REPTOT,
```


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

```
                                        NOM_VARC='PTOT',),);

      if (k==1) :

           DEPLA=STAT_NON_LINE(MODELE=MODMEC,
                               CHAM_MATER=MATEME,
                               EXCIT=_F(CHARGE=CHARMEC,),
                               COMPORTEMENT=_F(RELATION='ELAS',),
                               INCREMENT=_F(LIST_INST=UNPASSUB,),
                               ARCHIVAGE=_F(LIST_INST=UNPAS,),
                               NEWTON=_F(MATRICE='TANGENTE',
                                         REAC_ITER=10,),
                               CONVERGENCE=_F(ITER_GLOB_MAXI=10,),
                               SOLVEUR=_F(METHODE='MUMPS',PCENT_PIVOT=100,),);

         else :

           DEPLA=STAT_NON_LINE(reuse=DEPLA,
                               MODELE=MODMEC,
                               CHAM_MATER=MATEME,
                               EXCIT=_F(CHARGE=CHARMEC,),
                               ARCHIVAGE=_F(LIST_INST=UNPAS,),
                               COMPORTEMENT=_F(RELATION='ELAS',),
                               ETAT_INIT=_F(EVOL_NOLI=DEPLA,
                                            INST=listinst[k-1]),
                               INCREMENT=_F(LIST_INST=UNPASSUB,),
                               NEWTON=_F(MATRICE='TANGENTE',
                                         REAC_ITER=10,),
                               CONVERGENCE=_F(ITER_GLOB_MAXI=10,),
                               SOLVEUR=_F(METHODE='MUMPS',PCENT_PIVOT=100,),);

         if (k>1) :
            DETRUIRE(INFO=1,CONCEPT=_F(NOM=(DEFVHY,),),);
            DETRUIRE(INFO=1,CONCEPT=_F(NOM=(UNPAS,UNPASSUB,MATEHY,),),);
```

2.5.3 Conseils généraux d'utilisation

RELATION est renseigné par des relations de types KIT_XXXX qui permettent de résoudre simultanément de deux à quatre équations d'équilibre. Les équations considérées dépendent du suffixe XXXX avec la règle suivante :

- M désigne l'équation d'équilibre mécanique,
- T désigne l'équation d'équilibre thermique,
- H désigne une équation d'équilibre hydraulique,
- V désigne la présence d'une phase sous forme vapeur (en plus du liquide).

Une seule lettre H signifie que le milieu poreux est saturé (une seule variable de pression p), par exemple soit de gaz, soit de liquide, soit d'un mélange liquide/gaz (dont la pression du gaz est constante).
Deux lettres H signifient que le milieu poreux est non saturé (deux variables de pression p), par exemple un mélange liquide/vapeur/gaz.
La présence des deux lettres HV signifie que le milieu poreux est saturé par un composant (en pratique de l'eau), mais que ce composant peut être sous forme liquide ou vapeur. Il n'y a alors qu'une équation de conservation de ce composant, donc un seul degré de liberté pression, mais il y a un flux liquide et un flux vapeur. Les relations possibles sont alors les suivantes :

```
/   'KIT_HM'
```


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

```
/   'KIT_THM'
/   'KIT_HHM'
/   'KIT_THH'
/   'KIT_THV'
/   'KIT_THHM'
/   'KIT_HH'
```

Le tableau ci-dessous résume à quel kit correspond chaque modélisation :

KIT_HM	D_PLAN_HM, D_PLAN_HMD, D_PLAN_HMS, D_PLAN_HM_SI, AXIS_HM, AXIS_HMD, AXIS_HMS, 3D_HM, 3D_HMD , 3D_HMS, 3D_HM_SI
KIT_THM	D_PLAN_THM, D_PLAN_THMD, D_PLAN_THMS, AXIS_THM, AXIS_THMD, AXIS_THMS, 3D_THM, 3D_THMD, 3D_THMS
KIT_HHM	D_PLAN_HHM, D_PLAN_HHMD, D_PLAN_HHMS, AXIS_HHM, AXIS_HHMD, AXIS_HHMS, 3D_HHM, 3D_HHMD, 3D_HHMS, D_PLAN_HH2MD, AXIS_HH2MD, 3D_HH2MD, D_PLAN_HH2MS, AXIS_HH2MS, 3D_HH2MS, D_PLAN_HH2M_SI, 3D_HH2M_SI
KIT_THH	D_PLAN_THHD, D_PLAN_THHS, AXIS_THHD, AXIS_THHS, 3D_THHD, 3D_THHS, D_PLAN_THH2D, AXIS_THH2D, 3D_THH2D, D_PLAN_THH2S, AXIS_THH2S, 3D_THH2S
KIT_THV	D_PLAN_THVD , AXIS_THVD, 3D_THVD
KIT_THHM	D_PLAN_THHMD, D_PLAN_THHMS, AXIS_THHMD, AXIS_THHMS, 3D_THHM, 3D_THHMD, 3D_THHMS, D_PLAN_THH2MD, AXIS_THH2MD, 3D_THH2MD, D_PLAN_THH2MS, AXIS_THH2MS, 3D_THH2MS
KIT_HH	D_PLAN_HHD, AXIS_HHD, 3D_HHD, D_PLAN_HH2D, AXIS_HH2D, 3D_HH2D, D_PLAN_HHS, AXIS_HHS, 3D_HHS, D_PLAN_HH2S, AXIS_HH2S, 3D_HH2S

Pour chaque phénomène modélisé (hydraulique et/ou mécanique), on doit préciser dans RELATION_KIT :

- Le modèle de comportement mécanique du squelette s'il y a modélisation mécanique (M),

```
/   'ELAS'
/   'MOHR_COULOMB'
/   'CJS'
/   'HUJEUX'
/   'CAM_CLAY'
/   'BARCELONE'
/   'LAIGLE'
/   'LETK'
/   'DRUCK_PRAGER'
/   'DRUCK_PRAG_N_A'
/   'VISC_DRUCK_PRAG'
/   'GONF_ELAS'
/   'HOEK_BROWN_TOT'
/   'HOEK_BROWN_EFF'
/   'MAZARS'
/   'ENDO_ISOT_BETON'
```

- Le comportement des liquides/gaz ,(le même que celui indiqué dans COMP_THM sous DEFI_MATERIAU, cf. [§2.2.2])

```
/   'LIQU_SATU'
/   'LIQU_GAZ'
/   'GAZ'
```


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

```
/   'LIQU_GAZ_ATM'
/   'LIQU_VAPE_GAZ'
/   'LIQU_AD_GAZ_VAPE'
/   'LIQU_AD_GAZ'
/   'LIQU_VAPE'
```

- Concernant le comportement hydraulique, l'utilisateur dispose actuellement de 4 choix :

```
/   'HYDR_UTIL'
/   'HYDR_ENDO'
/   'HYDR_VGM'
/   'HYDR_VGC'
```

  - Soit il choisit HYDR_UTIL : ce mot clé permet alors de renseigner la courbe de saturation et sa dérivée en fonction de la pression capillaire ainsi que la perméabilité relative et sa dérivée en fonction de la saturation. L'utilisateur rentre ces courbes via une série de formules ensuite appelée par DEFI_MATERIAU (cf 2.2.7).
  - Soit il choisit HYDR_ENDO : c'est la même chose mais permet l'utilisation des lois avec endommagement 'MAZARS' et 'ENDO_ISOT_BETON' (la perméabilité est alors couplée à l'endommagement).
  - Soit il choisit HYDR_VGM ou HYDR_VGC ce qui signifie que les lois de capillarité et perméabilités obéiront au modèle de Mualem-Van-Genuchten prédéfini dans le code, autrement dit, la capillarité est donnée par une loi de Van Genuchten :

$$S_{we}=\frac{S-S_{wr}}{1-S_{wr}} \text{ et } m=1-\frac{1}{n}$$

$$S_{we}=\frac{1}{\left[1+\left(\frac{P_c-P_e}{P_r}\right)^n\right]^m}$$

La perméabilité relative à l'eau s'exprime alors en intégrant le modèle de prédiction proposé par Mualem (1976) dans le modèle de capillarité de Van-Genuchten :

$$k_r^w=\sqrt{S_{we}}\left(1-\left(1-S_{we}^{1/m}\right)^m\right)^2$$

La perméabilité au gaz est formulée de façon similaire par une loi de parker pour HYDR_VGM :

$$k_r^{gz}=\sqrt{(1-S_{we})}\left(1-S_{we}^{1/m}\right)^{2m}$$

ou par une loi cubique pour HYDR_VGC :

$$k_r^{gz}=(1-S)^3$$

avec dans tous les cas les paramètres n, Pr, Pe, Swr, $Smax$ définis dans DEFI_MATERIAU :

- n correspond au paramètre VG_N
- Pr correspond au paramètre VG_PR
- Pe correspond au paramètre VG_P ENTR (0 par défaut)
- Swr correspond au paramètre VG_SR

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

A cela doivent être rajoutés deux paramètres supplémentaires correspondant à un traitement que l'on effectue sur ces courbes :

$Smax$ qui correspond au paramètre VG_SMAX
$CSAT$ qui correspond au paramètre VG_SATUR

Nous allons maintenant expliciter ce qu'est ce traitement et ce que sont $CSAT$ et $Smax$. Pour $S>Smax$, ces courbes sont interpolées par un polynôme de degré 2, $C1$ en $Smax$, de manière à éviter d'avoir à traiter des dérivées de valeurs infinies. En effet, pour $S=1$:

$$\frac{\partial k_r^w(S)}{\partial S}=\infty$$

et dans le cas de HYDR_VGM :

$$\frac{\partial k_r^{gz}(S)}{\partial S}=\infty$$

Pour éviter d'avoir à traiter ce problème (qui n'a a priori pas de signification physique) on remplace ces fonctions à partir d'une saturation $Smax$ par un polynôme du second ordre $C1$ en ce point.

Ce qui donne pour la fonction $k_r^w(S)$:
Pour $S=Smax$, on détermine le polynôme $PL(S)$ tel que

$$\begin{cases} PL(S_{max})=k_r^w(S_{max}) \\ PL'(S_{max})=\dfrac{\partial k_r^w}{\partial S}(S_{max}) \end{cases} \text{ et } PL(1)=0$$

Pour $S>Smax$, $k_r^w(S)$ est remplacé par $PL(S)$.
Et, dans le cas de HYDR_VGM, pour $k_r^{gz}(S)$:
Pour $S=Smax$, on détermine le polynôme $PG(S)$ tel que

$$\begin{cases} PG(S_{max})=k_r^{gz}(S_{max}) \\ PG'(S_{max})=\dfrac{\partial k_r^{gz}}{\partial S}(S_{max}) \end{cases} \text{ et } PG(0)=1$$

Pour $S>Smax$, $k_r^{gz}(S)$ est remplacé par $PG(S)$ comme l'exemple Figure 2.5.3-a. Dans le cas d'une loi cubique, aucun traitement spécial n'est nécessaire.

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

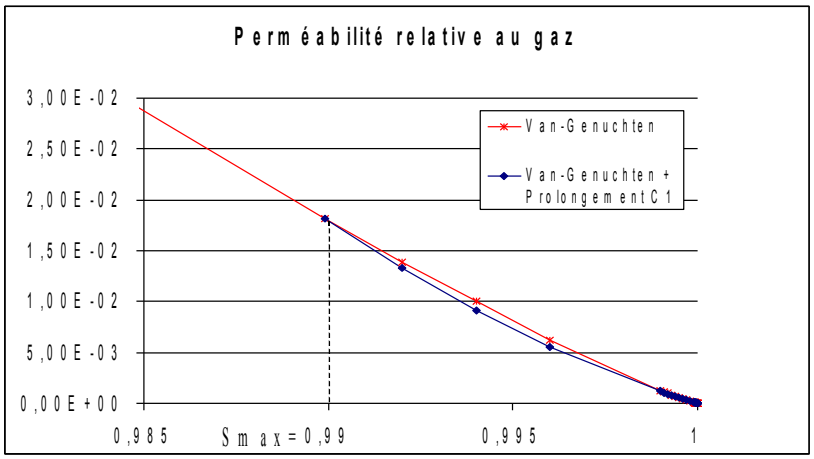


Figure 2.5.3-a: Exemple de prolongement d'une perméabilité relative

Pour la succion $S(Pc)$ et pour $Pc<Pcmin$ (avec $S(Pcmin)=Smax$) on prolonge la courbe $S(Pc)$ par une hyperbole telle que la courbe soit $C1$ en ce point :
Pour $S>Smax$:

$$S(Pc)=1-\frac{A}{B-P_c}$$

Avec A et B tels que la courbe soit $C1$ en $Smax$.
On a donc bien une courbe décroissante qui tend vers 1 quand Pc tend vers $-\infty$. Ce traitement nous permet de gérer des pressions capillaires négatives dans Code_Aster (voir l'exemple Figure 2.5.3-b).

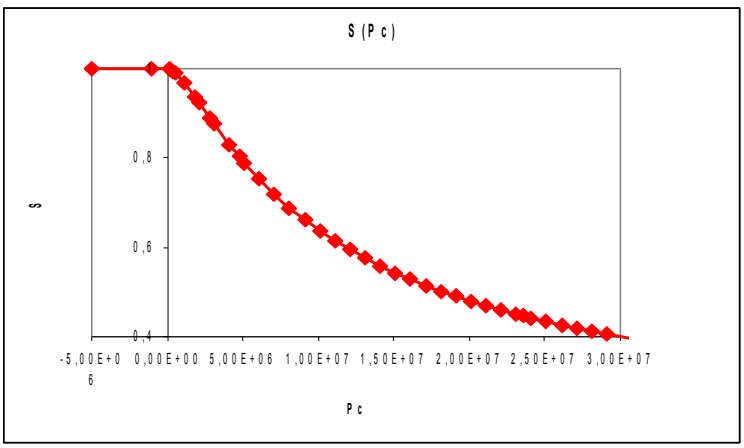


Figure 2.5.3-b: Exemple de prolongement pour PC<0

$S(Pc)$ est ensuite multipliée par un coefficient « de sécurité » $CSAT$ de manière à ce que la saturation n'atteigne jamais 1 (problème que l'on ne sait pas traiter).
Conseil : on conseille une valeur de $CSAT$ le plus proche possible de 1 ($0,99999$ par exemple).

Attention :

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

Selon le KIT_XXXX choisi, tous les comportements ne sont pas licites (par exemple si on choisit un milieux poreux non saturé, on ne peut pas affecter un comportement de type gaz parfait). toutes les combinaisons possibles sont résumées ci-dessous

	Mécanique	Couplage hydraulique	Loi hydraulique
KIT_HM KIT_THM	'ELAS','MOHR_COULOMB','CJS', 'HUJEUX','LAIGLE','CAM_CLAY', 'DRUCK_PRAGER','LEK', 'DRUCK_PRAG_N_A', 'VISC_DRUCK_PRAG','GONF_ELAS', 'HOEK_BROWN_TOT', 'HOEK_BROWN_EFF',	'GAZ', 'LIQU_SATU', 'LIQU_GAZ_ATM'	'HYDR_UTIL'
KIT_HM KIT_THM	'MAZARS', 'ENDO_ISOT_BETON',	'GAZ', 'LIQU_SATU', 'LIQU_GAZ_ATM'	'HYDR_ENDO'
KIT_HHM KIT_THHM	'ELAS','MOHR_COULOMB','CJS', 'HUJEUX''LAIGLE', 'CAM_CLAY','DRUCK_PRAGER', 'DRUCK_PRAG_N_A', 'VISC_DRUCK_PRAG', 'GONF_ELAS','LEK', 'HOEK_BROWN_TOT', 'HOEK_BROWN_EFF','BARCELONE'	'LIQU_GAZ','LIQU_VAPE_GAZ', 'LIQU_AD_VAPE_GAZ' 'LIQU_AD_GAZ'	'HYDR_UTIL' 'HYDR_VGM' 'HYDR_VGC'
KIT_HHM KIT_THHM	'MAZARS', 'ENDO_ISOT_BETON',	'LIQU_GAZ','LIQU_VAPE_GAZ', 'LIQU_AD_VAPE_GAZ' 'LIQU_AD_GAZ'	'HYDR_ENDO'
KIT_THH KIT_HH		'LIQU_GAZ','LIQU_VAPE_GAZ', 'LIQU_AD_VAPE_GAZ' 'LIQU_AD_GAZ'	'HYDR_UTIL' 'HYDR_VGM' 'HYDR_VGC'
KIT_THV		'LIQU_VAPE'	'HYDR_UTIL'

Remarque :

En cas de problème de convergence, il peut être très utile d'activer la recherche linéaire. La recherche linéaire n'améliore cependant pas systématiquement la convergence, elle est donc à manier avec précaution.

2.6 Le post-traitement

2.6.1 Généralités

Le post-traitement des données en THM ne varie pas du post-traitement Aster habituel. On rappelle juste que pour toute impression des valeurs aux nœuds qui ne sont pas les inconnues nodales, il est nécessaire de calculer ces valeurs par la commande CALC_CHAMP dont on donne un exemple ci-après.

Pour les contraintes :

```
U0=CALC_CHAMP(reuse =U0,
              MODELE=MODELE,
              CHAM_MATER=CHMAT0,
              TOUT_ORDRE='OUI',
              CONTRAINTE=('SIGM_ELNO'),
              RESULTAT=U0,);
```


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

Pour les variables internes :

```
U0=CALC_CHAMP(reuse =U0,
              MODELE=MODELE,
              CHAM_MATER=CHMAT0,
              TOUT_ORDRE='OUI',
              VARI_INTERNE=('VARI_ELNO'),
              RESULTAT=U0,);
```

Il faut toutefois rappeler que toutes les valeurs de déplacements en sorties correspondent à u^{ddl} et non $u=u^{ddl}+u^{ref}$.

Il est également important de connaître le nom des contraintes et les numéros des variables internes. Tout cela est consigné dans l'annexe I.
Ainsi l'exemple suivant permet d'imprimer la masse d'eau liquide sur le groupe de nœuds $HAUT$ à tous les instants.

```
TAB1=POST_RELEVE_T(ACTION=_F(INTITULE='CONT',
                             GROUP_NO=('HAUT'),
                             RESULTAT=U0,
                             NOM_CHAM='SIEF_ELNO',
                             TOUT_ORDRE='OUI',
                             NOM_CMP=('M11'),
                             OPERATION='EXTRACTION',),);

IMPR_TABLE(TABLE=TAB1,
           FICHIER='RESULTAT',
           FORMAT='AGRAF',
           PAGINATION='INST',
           NOM_PARA=('INST','COOR_X','M11'),);
```

L'exemple suivant permet d'imprimer les valeurs de porosité au nœud 1 et au premier instant.

```
TAB2=POST_RELEVE_T(ACTION=_F(INTITULE='DEPL',
                             NOEUD='NO1',
                              RESULTAT=U0,
                              NOM_CHAM='VARI_ELNO',
                              NUME_ORDRE=1,
                              NOM_CMP=('V2'),
                              OPERATION='EXTRACTION',),);
IMPR_TABLE(TABLE=TAB2,
           FICHIER='RESULTAT',
           FORMAT='AGRAF',
           PAGINATION='INST',
           NOM_PARA=('INST','COOR_X','V2'),);
```

2.6.2 Variables internes

Outre les généralités proposées ci-dessus, il existe désormais un moyen plus convivial d''extraire les variables internes (cf doc U4.81.01). Pour cela des champs ont été créés dont le principe est d'extraire du champ VARI_ELGA (ou VARI_ELNO pour les champs calculés aux nœuds) la variable interne qui nous intéresse via un mot clé plus parlant que $V1$, $V2$, ...

En tant que post-traitement ces champs sont calculés par CALC_CHAMP. La syntaxe à utiliser est la suivante :

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

- pour un champ aux éléments cham_elem

```
GAMP=CALC_CHAMP(RESULTAT=U0,
      OPTION='VAEX_ELNO',
       NOM_VARI='GAMP');
```

- pour un champ aux nœuds cham_no :

```
GAMP=CALC_CHAMP(reuse=GAMP,
  RESULTAT=GAMP,
  VARI_INTERNE='VAEX_NOEU');
```

Puisqu'il s'agit juste d'extraire une (et une seule!!) variable interne, les cham_elem correspondants doivent avoir été calculés au préalable :

```
U0=CALC_CHAMP(reuse =U0,
              MODELE=MODELE,
              CHAM_MATER=CHMAT0,
              TOUT_ORDRE='OUI',
              VARI_INTERNE=('VARI_ELNO'),
              RESULTAT=U0,);
```

La liste des différents noms symboliques des variables internes est :

"DPORO"	variation de la porosité du matériau
"DRHOLQ"	variation de la masse volumique du matériau
"DPVP"	variation de la pression de vapeur
"SATLIQ"	saturation du liquide
"EVP"	déformation plastique volumique cumulée
"IND_ETA"	Indicateur d'état mécanique
"D"	Valeur de l'endommagement
"IND_END"	Indicateur d'endommagement
"TEMP_MAX"	Température maximale
"GAMP"	Déformation déviatoire plastique cumulée
"PCR"	Pression critique
"SEUIL_HYD"	Seuil hydrique
"IND_HYD"	Indicateur d'irréversibilité hydrique
"PCOHE"	Pression de cohésion
"COMP_ROC"	Comportement de la roche
"SEUIL_ISO"	Seuil isotrope
"ANG_DEV"	Angle du seuil déviatoire
"X11"	Composantes du tenseur d'écrouissage cinématique
"X22"	Composantes du tenseur d'écrouissage cinématique
"X33"	Composantes du tenseur d'écrouissage cinématique
"X12"	Composantes du tenseur d'écrouissage cinématique
"X13"	Composantes du tenseur d'écrouissage cinématique
"X23"	Composantes du tenseur d'écrouissage cinématique
"DIST_DEV"	Distance normalisée au seuil déviatoire
"DEV_SUR_CRIT"	Rapport entre le seuil déviatoire et le seuil déviatorique critique
"DIST_ISO"	Distance normalisée au seuil isotrope
"NB_ITER"	Nombre d'itérations internes
"ARRET"	Valeur du test local d'arrêt du processus itératif
"NB_REDE"	Nombre de redécoupage local du pas de temps
"SIGNE"	Signe du produit contracté de la contrainte déviatorique par la déformation plastique déviatorique

Remarque :

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

Lorsque la variable à extraire ne fait pas partie des variables internes des lois concernées, une alarme est émise mais le champ est tout de même affecté à `R8VIDE()`.

2.6.3 Isovaleurs

Attention : distinction résolution linéaire et maillage quadratique

On rappelle que pour tout problème THM et afin d'éviter toute ambiguïté pour l'utilisateur, le maillage utilisé est toujours quadratique. En revanche la résolution du problème hydraulique et thermique est linéaire : les inconnues *PRE1* , *PRE2* et *TEMP* n'existent donc que sur les nœuds sommets du maillage.

A l'issue du calcul, on obtient un résultat sous cette forme:
- Les pressions *PRE1* , *PRE2* et les températures *TEMP* sont sur les nœuds sommets des éléments.
- Les variables internes, les contraintes sont sur les points de gauss et/ou les nœuds du maillage quadratique (suivant que l'on soit en sélectif ou en lumpé) et les déplacements mécaniques sont sur les sommets et les nœuds milieux.

Il y a donc un conflit entre les maillages quadratiques et les calculs linéaires ce qui va poser un problème par exemple pour tracer des isovaleurs.

Il existe alors une manière simple de s'affranchir de ce problème en projetant l'ensemble des résultats sur le maillage linéaire issu du maillage quadratique. Ainsi tous les résultats seront définis au même endroit (sur les sommets).

Pour *TEMP* , *PRE1* et *PRE2* , il suffit de projeter comme sur l'exemple suivant :

projection du maillage quadratique sur le maillage linéaire :

```
MAILQ4 = CREA_MAILLAGE(
                          MAILLAGE  = MAIL,
                          QUAD_LINE = _F(TOUT = 'OUI'),
                        )
```

affectation du modèle correspondant :

```
MODELQ4=AFFE_MODELE(MAILLAGE=MAILQ4,
                    AFFE=_F(TOUT='OUI',
                            PHENOMENE='MECANIQUE',
                            MODELISATION='D_PLAN',),);
```

projection du résultat sur ce nouveau modèle :

```
RESULIN = PROJ_CHAMP(
              METHODE='ELEM',
              RESULTAT=resu,
              MODELE_1=MODELQ8,
              MODELE_2=MODELQ4,
              NOM_CHAM ='DEPL',
              TOUT_ORDRE='OUI')
```

post-traitement habituel mais sur le maillage linéaire :

```
IMPR_RESU(FORMAT='MED',
          RESU=_F(
                  RESULTAT=RESULIN,
```


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

```
                                NUME_ORDRE=(1,10,20,),
                                NOM_CMP    = ('PRE1','PRE2',),
                                MAILLAGE=MAILQ4,),);
```

Il existe une méthode alternative, qui peut s'avérer plus rapide pour les grosses études industrielles. Cette méthode utilise MACR_ADAP_MAIL. On suppose que resu est composé de nbinst instants stockés dans la liste linst. Le post-traitement précédent peut également se faire de la manière suivante :

```
PRE  = [None]*nbinst;
PREB = [None]*nbinst;

for k in range(nbinst) :

  PRE[k]=CREA_CHAMP(TYPE_CHAM='NOEU_DEPL_R',
                    OPERATION='EXTR',
                    RESULTAT=resu,
                    NOM_CHAM='DEPL',
                    INST=linst[k],);

motcle1={}
motcle2={}

motcle1['MAJ_CHAM']=[]
motcle2['AFFE']=[]

for k in range(nbinst) :

  PREB[k] =CO('PREB_%d' % (k))

  motcle1['MAJ_CHAM'].append(
  _F(TYPE_CHAM='NOEU_DEPL_R',CHAM_GD  = PRE[k],
                                  CHAM_MAJ = PREB[k],),);

  motcle2['AFFE'].append(
  _F(CHAM_GD=PREB[k],INST=linst[k],),);

MACR_ADAP_MAIL( ADAPTATION='MODIFICATION',DEGRE='OUI',
                MAILLAGE_N   = MAILQ8,
                MAILLAGE_NP1 = CO('MAILQ4'),
                **motcle1);

RESULINB=CREA_RESU(OPERATION='AFFE',TYPE_RESU='EVOL_NOLI',NOM_CHAM='DEPL',**
motcle2);
```

2.7 Quelques cas tests

A titre informatif, on récapitule ici quelques cas tests disponibles dans la base de Code_Aster. Cette liste n'est absolument pas exhaustive mais permet d'accéder directement à un certain nombre d'exemples représentatifs.

Nom de cas-test	Type de modélisation THM	Phénomène représenté	Modélisation géométrique	Loi de couplage
Wtna109a	AXIS_THVD	Désaturation par chauffage d'un milieu sans air	axisymétrique	LIQU_VAPE
Wtna110a	AXIS_HH2MS	Modélisation d'un gonflement d'une argile	axisymétrique	LIQU_AD_GAZ_VAPE
Wtna107	AXIS_THMS	Chauffage d'un milieu	axisymétrique	LIQU_SATU

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

		saturé (THM)		
wtna112	AXIS_THMS	Pressurisation thermique d'une éprouvette drainée	axisymétrique	LIQU_SATU
Wtnp117a	D_PLAN_HHS	Rééquilibrage capillaire de 2 matériaux (BO/BG)	plane	LIQU_GAZ
Wtnp118a	D_PLAN_HH2D	Rééquilibrage gravitaire d'une colonne	plane	LIQU_AD_GAZ_VAPE
Wtnp112d	D_PLAN_HMD	Resaturation d'une colonne (hypothèse de Richards)	plane	LIQU_GAZ_ATM
Wtnp101a	D_PLAN_HM	Modélisation d'un milieu saturé en gaz	plane	GAZ
Wtnl100a	D_PLAN_HMD D_PLAN_HM_SI	Problème de consolidation (Therzagi)	plane	LIQU_SATU
Wtnv123a	3D_HHM	Essai hydrostatique puis triaxial à succion fixe (modèle de Barcelone)	3D	LIQU_GAZ
Wtnp113a	D_PLAN_THHD	Resaturation d'une alvéole par le massif	plane	LIQU_GAZ
Wtnv121a	3D_HHMD	Mouillage du béton avec endommagement	3D	LIQU_GAZ
Wtnv130a	3D_THH2S	Chauffage d'un milieu poreux avec air dissous	3D	LIQU_AD_GAZ_VAPE
wtnv140	3D_HMS	Essai triaxial drainé anisotrope	3D	LIQU_SATU

3 Les modélisations THM par une approche second gradient

Les modélisations de type second gradient et second gradient de dilatation sont décrites dans la documentation [R5.04.03] dans le cadre des milieux poreux. Leur utilisation est indispensable lorsque les lois de comportements mécaniques modélisent une dégradation du matériau – on parle de lois adoucissantes. En effet, dans ce cas, les solutions dépendent du maillage y compris lorsque le milieu est couplé avec les équations de l'hydraulique.

L'intégration numérique de ces modèles a suivi un protocole atypique lors de sa mise en œuvre pour faciliter son utilisation dans Code_Aster. Ainsi, la partie régularisante (second gradient) est définie comme un patch qu'il suffit de « coller » sur la structure à modéliser. On décrit ci-dessous les étapes à suivre pour produire une telle simulation.

1) Définition du patch
2) Choix du modèle
3) Définition du matériau
4) Conditions aux limites
5) Calcul

3.1 Définition du patch

Pour la simulation d'une étude THM par une approche de type second gradient, il faut définir un maillage quadratique pour la discrétisation de la structure, puis un second maillage pour le patch régularisant. La particularité de ce second maillage est qu'il doit nécessairement être topologiquement identique au premier maillage (mêmes nœuds et mêmes éléments géométriques. Pour des raisons de simplification numérique et pour des raisons de cohérence de modèles, on conseille d'utiliser la procédure suivante :

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

1) Définition d'une discrétisation spatiale pour la structure et la modélisation THM ;
2) Duplication du maillage précédent sur l'ensemble de la structure par la commande CREA_MAILLAGE pour définir le patch
3) Modification éventuelle du type de mailles pour une étude avec multiplicateurs de Lagrange aux centres des éléments.

3.1.1 Étape 1. Définition du maillage de la structure

Cette première étape est à produire par un mailleur pour discrétiser la structure. On récupère ensuite le maillage par la commande *Code_Aster* :

lecture du maillage (quadratique)

```
MA = LIRE_MAILLAGE()
```

3.1.2 Étape 2. Duplication du maillage pour définir le patch

Il existe dans *Code_Aster* une procédure qui permet de dupliquer un maillage en conservant les mêmes nœuds que le maillage original – il est indispensable que les nœuds du premier maillage soient inclus dans le patch (cf [R5.04.03]). Il est nécessaire de définir ce second maillage pour permettre de définir la modélisation second gradient au patch régularisant.

Duplication du maillage (quadratique). Seules les mailles sont dupliquées, les nœuds restent communs. Le but est d'associer à chacun de ces maillages une modélisation différente.

```
MAIL=CREA_MAILLAGE(MAILLAGE      = MA,
                   CREA_GROUP_MA = ( _F( NOM      = 'ROCHE_REG',
                                         GROUP_MA = 'ROCHE')))
```

3.1.3 Étape 3. Modification (éventuelle) du maillage du patch

Il existe différentes discrétisations éléments finis pour interpoler les modélisations second gradient (avec ou sans multiplicateurs de Lagrange, multiplicateurs de Lagrange au centre des éléments ou aux nœuds sommets). Par conséquent il faut adapter le maillage du patch à l'espace de discrétisation de la modélisation second gradient.

Par exemple pour prendre en compte une interpolation avec des multiplicateurs de Lagrange aux centres des éléments, il faut définir les nœuds aux centres des éléments car ceux-ci ne sont pas prévus, en général, dans les discrétisations spatiales proposées par la plupart des mailleurs. Ainsi dans ce cas, on peut utiliser la commande *Code_Aster* suivante :

Introduction de nœuds centraux aux éléments finis du nouveau maillage pour une interpolation $P2-P1-P0$. Procédure nécessaire pour prendre en compte les interpolations des multiplicateurs de Lagrange.

```
MAILLAGE=CREA_MAILLAGE(MAILLAGE    = MAIL,
                       MODI_MAILLE =( _F( GROUP_MA = 'ROCHE_REG',
                                         OPTION   = 'QUAD8_9'),
                                      _F( GROUP_MA = 'ROCHE_REG',
                                         OPTION   = 'TRIA6_7')))
```

Cette étape n'est possible qu'en 2D lorsque l'interpolation élément fini considère des multiplicateurs de Lagrange constant par élément.

3.2 Choix du modèle

Les « patchs régularisants » second gradient ou second gradient de dilatation sont à combiner avec n'importe quel type de modélisation THM. Ils n'ont d'intérêt, bien entendu, qu'en présence d'une composante mécanique (adoucissante). Les modélisations disponibles sont :

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

Modélisation	Modélisation géométrique	Phénomène pris en compte
D_PLAN_2DG	Plane	Second gradient
D_PLAN_DIL	Plane	Second gradient de dilatation
3D_DIL	3D	Second gradient de dilatation

Remarque :

A noter qu'il est fortement conseillé d'utiliser les modélisations second gradient de dilatation pour traiter les comportements des sols et des roches ou plus particulièrement tout type de matériau présentant une variation volumique lors de sa dégradation .

Les inconnues nodales des modélisations THM sont alors enrichies après ajout du « patch » second gradient. Le nombre d'inconnues varie selon le patch considéré et le type de mailles caractérisant la discrétisation spatiale du patch. Pour la modélisation second gradient de dilatation c'est donc par le choix de la topologie de la maille que l'interpolation des multiplicateurs de Lagrange sera déterminée (voir le tableau ci-dessous). On résume dans le tableau ci-dessous les différentes possibilités où l'on note en accord avec [R5.04.03], f_{ij} les composantes du tenseur des déformations microscopiques, χ la déformation volumique microscopique et λ le multiplicateur de Lagrange:

Modélisation	Type de mailles	Degrés de liberté	Position du degré de liberté
D_PLAN_2DG	TRIA7, QUAD9	f_{ij}	Aux sommets de chaque élément
		λ_{ij}	Au centre de chaque élément
D_PLAN_DIL	TRIA7, QUAD9	χ	Aux sommets de chaque élément
		λ	Au centre de chaque élément
	TRIA6, QUAD8	χ	Aux sommets de chaque élément
3D_DIL	TETRA10, HEXA20, PENTA15	χ , λ	Aux sommets de chaque élément

On procède alors de la façon suivante :

```
MODELE = AFFE_MODELE( MAILLAGE = MAILLAGE,
                      AFFE     = ( _F( GROUP_MA     = 'ROCHE',
                                       PHENOMENE    = 'MECANIQUE',
                                       MODELISATION = 'D_PLAN_HMS'),
                                   _F( GROUP_MA     = 'ROCHE_REG',
                                       PHENOMENE    = 'MECANIQUE',
                                       MODELISATION  = 'D_PLAN_DIL')))
```

3.3 Définition du matériau

Il faut, en plus de la définition des paramètres matériaux nécessaires pour le calcul THM décrit au chapitre précédent, ajouter un paramètre qui caractérise la longueur caractéristique de la modélisation second gradient (cf [R5.04.03]). Cette donnée caractérise en fait la dimension du voisinage qui contribue à la description d'un point matériel.

Cette longueur est définie à partir des composantes matériau $A1$ et $A3$ de la loi de comportement d'élasticité linéaire second gradient inspirée des travaux de Mindlin dans le cas de la modélisation second gradient de dilatation. C'est dans la détermination de ces paramètres que réside la plus grande difficulté. Il n'existe pas aujourd'hui de méthode analytique pour les identifier. La pratique est de faire quelques essais préalable. Dans le cas des modélisations second gradient cette longueur est fonction des 5 composantes $A1$, $A2$, $A3$, $A4$ et $A5$, à renseigner sous le mot-clé ELAS_2NDG de DEFI_MATERIAU (cf [R5.04.03]).

On peut, de plus, ajouter un paramètre numérique de pénalisation. Il a été montré que le terme de pénalisation des modélisations second gradient de dilatation améliore la convergence numérique sans perturber la qualité des résultats.

Voici un exemple de définition des paramètres matériau pour la partie second gradient.

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

```
SOL2 = DEFI_MATERIAU( ELAS_2NDG  = _F( A1 = 4.0E4,
                                       A2 = 0.0,
                                       A3 = 0.0,
                                       A4 = 0.0,
                                       A5 = 0.0,),
                      NON_LOCAL  = _F(PENA_LAGR=1.0E12,),);
```

Puis affectation des paramètres matériaux suivant la même procédure que pour la définition des modélisations.

```
MATE = AFFE_MATERIAU( MAILLAGE = MAILLAGE,
                      AFFE     = (_F( TOUT     = 'ROCHE',
                                      MATER    =  SOL_HM,),
                                  _F( GROUP_MA = 'ROCHE_REG',
                                      MATER    =  SOL2)))
```

3.4 Impact sur les conditions aux limites

Il n'y a aucun impact à apporter dans le fichier de commande Code_Aster pour prendre en compte les conditions aux limites avec les modélisations second gradient par rapport aux modélisations classiques THM. En revanche, il est essentiel de noter que la signification physique des conditions aux limites de pression de contraintes normales est modifiée avec les modélisations second gradient (cf [R5.04.03]).

3.5 Résolution du problème

Le calcul est à effectuer par la commande STAT_NON_LINE. Il suffit d'ajouter au calcul classique THM la relation de comportement d'élasticité second gradient sur le maillage correspond au patch comme sur l'exemple ci-dessous (mot-clé ELAS). Il n'y a aucune incompatibilité avec les lois de comportement. A noter, tout de même, qu'une simulation THM par une approche second gradient n'a d'intérêt qu'en présence d'une composante mécanique dans le calcul couplé.

Il est important de noter que lorsque la modélisation prend en compte une interpolation des multiplicateurs de Lagrange constants par éléments il est nécessaire d'utiliser le solveur Mumps pour la résolution.

Définition du calcul statique non linéaire avec une loi de comportement associé à chacune des modélisation : comportement de type Drucker-Prager pour le premier gradient, et élasticité linéaire pour le second gradient

```
U1 = STAT_NON_LINE( MODELE     = MODELE,
                    CHAM_MATER = MATE,
                    EXCIT      = ( _F( CHARGE = CHCI ),…),
                    SOLVEUR    = ( _F( METHODE='MUMPS',)),
                    COMPORTEMENT  = ( _F(GROUP_MA='ROCHE',
                                         RELATION = 'DRUCK_PRAGER',),
                                      _F(GROUP_MA = 'ROCHE_REG',
                                         RELATION = 'ELAS',),),
                  NEWTON    = _F( MATRICE  = 'TANGENTE',REAC_ITER = 1 ),
                  INCREMENT = _F( LIST_INST = TEMPS ))
```

On trouvera un exemple de modélisation par second gradient, par exemple dans le cas test ssll117.

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

Annexe 1Contraintes généralisées et variables internes

Les contraintes :

Numéro	Nom de composante Aster	Contenu	Modélisations
1	SIXX	σ'_{xx}	Si mécanique (.. M …)
2	SIYY	σ'_{yy}	Si mécanique (.. M …)
3	SIZZ	σ'_{zz}	Si mécanique (.. M …)
4	SIXY	σ'_{xy}	Si mécanique (.. M …)
5	SIXZ	σ'_{xz}	Si mécanique (.. M …)
6	SIYZ	σ'_{yz}	Si mécanique (.. M …)
7	SIPXX	$\sigma_{p_{xx}}$	Si mécanique (.. M …)
8	SIPYY	$\sigma_{p_{yy}}$	Si mécanique (.. M …)
9	SIPZZ	$\sigma_{p_{zz}}$	Si mécanique (.. M …)
10	SIPXY	$\sigma_{p_{xy}}$	Si mécanique (.. M …)
11	SIPXZ	$\sigma_{p_{xz}}$	Si mécanique (.. M …)
12	SIPYZ	$\sigma_{p_{yz}}$	Si mécanique (.. M …)
13	M11	m_w	Dans tous les cas
14	FH11X	M_{w_x}	Dans tous les cas
15	FH11Y	M_{w_y}	Dans tous les cas
16	FH11Z	M_{w_z}	Dans tous les cas
17	ENT11	h_w^m	Dans tous les cas avec thermique
18	M12	m_{vp}	Si 2 pressions inconnues (.. HH …)
19	FH12X	M_{vp_x}	Si 2 pressions inconnues (.. HH …)
20	FH12Y	M_{vp_y}	Si 2 pressions inconnues (.. HH …)
21	FH12Z	M_{vp_z}	Si 2 pressions inconnues (.. HH …)
22	ENT12	h_{vp}^m	Si 2 pressions inconnues et thermique (.. THH …)
23	M21	m_{as}	Si 2 pressions inconnues (.. HH …)
24	FH21X	M_{as_x}	Si 2 pressions inconnues (.. HH …)
25	FH21Y	M_{as_y}	Si 2 pressions inconnues (.. HH …)
26	FH21Z	M_{as_z}	Si 2 pressions inconnues (.. HH …)
27	ENT21	h_{as}^m	Si 2 pressions inconnues et thermique (.. THH …)
28	M22	m_{ad}	Si modélisation de l'air dissous (…HH2…)
29	FH22X	M_{ad_x}	Si modélisation de l'air dissous (…HH2…)
30	FH22Y	M_{ad_y}	Si modélisation de l'air dissous (…HH2…)

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

31	FH22Z	M_{ad_z}	Si modélisation de l'air dissous (…HH2…)
32	ENT22	h_{ad}^{m}	Si modélisation de l'air dissous et thermique (…THH2…)
33	QPRIM	Q'	Si thermique
34	FHTX	q_x	Si thermique
35	FHTY	q_y	Si thermique
36	FHTZ	q_z	Si thermique

Dans le cas sans mécanique, et pour les lois de comportements (LIQU_VAPE_GAZ,LIQU_VAPE, LIQU_AD_GAZ_VAPE , 'LIQU_AD_GAZ') les variables internes sont :

Numéro	Nom de composante Aster	Contenu
1	$V1$	$\rho_{lq}-\rho_{lq}^{0}$
2	$V2$	$\varphi-\varphi^{0}$
3	$V3$	$p_{vp}-p_{vp}^{0}$
4	$V4$	S_{lq}

Dans le cas sans mécanique, et pour les lois de comportements (LIQU_GAZ,LIQU_GAZ_ATM,) les variables internes sont :

Numéro	Nom de composante Aster	Contenu
1	$V1$	$\rho_{lq}-\rho_{lq}^{0}$
2	$V2$	$\varphi-\varphi^{0}$
3	$V3$	S_{lq}

Dans le cas sans mécanique, et pour les lois de comportements (LIQU_SATU,) les variables internes sont :

Numéro	Nom de composante Aster	Contenu
1	$V1$	$\rho_{lq}-\rho_{lq}^{0}$
2	$V2$	$\varphi-\varphi^{0}$

Dans le cas avec mécanique les premiers numéros seront ceux correspondant à la mécanique ($V1$ dans le cas élastique, $V1$ et suivantes pour des modèles plastiques). Le numéro des variables internes ci-dessus devra alors être incrémenté d'autant.

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

Annexe 2 Éléments supplémentaires sur les conditions aux limites en THM

Dans ce qui suit on ne prend pas en compte l'air dissous (l'indice lq correspond alors à celui de l'eau w) et on s'attache au cas non saturé.

Nous rappelons ici le choix des inconnues de pression.

Comportement	**LIQU_GAZ et LIQU_VAPE_GAZ**
PRE1	Pression capillaire : $p_c = p_{gz} - p_{lq}$
PRE2	Pression de gaz $p_{gz} = p_{vp} + p_{as}$

A2.1 Formulation variationnelle des équations de conservation

On se réfère ici à [R7.01.11]. Ces équations sont

$$\dot{m}_{lq} + \dot{m}_{vp} + Div\left(M_{lq} + M_{vp}\right) = 0 \qquad \text{éq A5.1-1}$$

$$\dot{m}_{as} + Div\left(M_{as}\right) = 0 \qquad \text{éq A5.1-2}$$

La formulation variationnelle déduite est donnée par

$$-\int_\Omega \left(\dot{m}_{lq} + \dot{m}_{vp}\right)\pi_1 d\Omega + \int_\Omega \left(M_{lq} + M_{vp}\right).\nabla \pi_1 d\Omega = \int_{\partial\Omega}\left(M_{lq_{ext}} + M_{vp_{ext}}\right).\pi_1 d\Gamma \;\forall \pi_1 \in P_{1_{ad}} \qquad \text{éq A5.1-3}$$

$$-\int_\Omega \dot{m}_{as}\pi_2 d\Omega + \int_\Omega M_{as}.\nabla \pi_2 d\Omega = \int_{\partial\Omega} M_{as_{ext}}.\pi_2 d\Gamma \;\forall \pi_2 \in P_{2_{ad}} \qquad \text{éq A5.1-4}$$

Les pressions capillaires et de gaz sont liées aux pression d'eau, de vapeur et d'air sec par les relations :

$$p_c = p_{gz} - p_{lq} \qquad \text{éq A5.1-5}$$

$$p_{gz} = p_{vp} + p_{as} \qquad \text{éq A5.1-6}$$

La pression de vapeur n'est pas une variable indépendante. Elle est reliée à la pression de liquide p_{lq} par les relations

$$\frac{dp_{vp}}{\rho_{vp}} = \frac{dp_{lq}}{\rho_{lq}} + \left(h^m_{vp} - h^m_{lq}\right)\frac{dT}{T} \qquad \text{éq A5.1-7}$$

$$dh^m_{lq} = C^p_{lq}dT + \left(1 - 3\alpha_{lq}T\right)\frac{dp_{lq}}{\rho_{lq}} \qquad \text{éq A5.1-8}$$

$$dh^m_{vp} = C^p_{vp}dT \qquad \text{éq A5.1-9}$$

Ces relations montrent que la pression de vapeur est déterminée complètement pas la connaissance de p_{lq} (et de son évolution). Souvent, ces relations servent à établir la loi de Kelvin, $\frac{p_{lq} - p^0_{lq}}{\rho_{lq}} = \frac{R}{M^{ol}_{vp}} T \ln\left(\frac{p_{vp}}{p^{sat}_{vp}(T)}\right)$, mais cette loi n'est pas utilisée directement dans Aster.

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

Les documents de référence Aster ne disent rien sur ce que sont les variables π_1 et π_2. Mais deux éléments peuvent nous mettre sur la piste :

- D'une part, $\pi_1 \in P_{1_{ad}}$ et $\pi_2 \in P_{2_{ad}}$ alors que $P_{1_{ad}}$ et $P_{2_{ad}}$ sont les espaces d'appartenance de $PRE1$ et $PRE2$ (incluant donc leurs conditions aux limites).
- D'autre part, au chapitre 7. de [R7.01.10], on voit que la déformation virtuelle $E_g^{*el} = (v, \varepsilon(v), \pi_1, \nabla \pi_1, \pi_2, \nabla \pi_2, \tau, \nabla \tau)$ est liée au vecteur de déplacement nodaux virtuel $U^{*el} = (v, \pi_1, \pi_2, \tau)$ par le même opérateur Q_g^{el} que celui qui relie entre eux la déformation $E_g^{el} = (u, \varepsilon(u), p_1, \nabla p_1, p_2, \nabla p_2, T, \nabla T)$ et le déplacement nodal $U^{el} = (u, p_1, p_2, T)$:
  - $E_g^{*el} = Q_g^{el} U^{*el}$
  - $E_g^{el} = Q_g^{el} U^{el}$

Il est alors clair que π_1 et π_2 sont des variations virtuelles de p_1 et p_2
D'où le tableau :

$$p_1 = p_c \Rightarrow \pi_1 = \delta p_c = p_c^*$$
$$p_1 = p_{lq} \Rightarrow \pi_1 = \delta p_{lq} = p_{lq}^*$$
$$p_2 = p_{gz} \Rightarrow \pi_2 = \delta p_{gz} = p_{gz}^*$$

A2.2 Cas de conditions aux limites faisant intervenir les inconnues principales

Ce que nous disons dans ce paragraphe et les suivants est relatif à une partie de la frontière $\partial\Omega_d$ sur laquelle des conditions sont prescrites : rien n'empêche bien sûr que ces conditions ne soient pas les mêmes sur des parties de frontières différentes. Nous traitons dans ce chapitre le cas habituel où on impose des conditions sur $PRE1$ et/ou $PRE2$, par opposition au chapitre suivant où nous parlerons de relations linéaires entre inconnues.

$$p_c = p_{gz} - p_{lq} = p_c^{imp}$$
$$p_{gz} = p_{as} + p_{vp} = p_{gz}^{imp}$$

Les flux sont alors des résultats de calculs par [éq A5.1-3] et [éq A5.1-4]

- **Dirichlet** $PRE1$ **, Neuman** $PRE2$

C'est le cas où on impose une valeur à $PRE1$ et une valeur au flux associé à $PRE2$, en ne disant rien sur $PRE2$ ou en donnant une valeur à FLUN_HYDR2 de FLUX_THM_REP dans AFFE_CHAR_MECA. Appelons $M_{2_{ext}}$ cette quantité imposée, qui vaudra 0 si rien n'est dit relativement à $PRE2$. Nous noterons $p_1 = p_1^{imp}$ la condition imposée sur $PRE1$
Ceci correspond à :

$$p_c = p_{gz} - p_{lq} = p_c^{imp}$$
$$p_1^{imp} = p_c^{imp}$$

Pour faire la démonstration dans le cadre non homogène, il faudrait introduire un relèvement de la condition $p_1 = p_1^{imp}$ (c'est à dire un champ particulier vérifiant cette condition). Cela alourdit les écritures et n'apporte rien, on se place donc dans le cadre homogène $p_1^{imp} = 0$

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

Dans [éq A5.1-3] et [éq A5.1-4], on peut donc prendre et π_2 quelconque et π_1 vérifiant $\pi_1=0$ sur $\partial\Omega_d$ On commence alors à prendre $\pi_1=0$ et $\pi_2=0$ sur tout le bord $\partial\Omega$ et on obtient [éq A5.1-1] et [éq A5.1-2] au sens des distributions. On multiplie alors [éq A5.1-1] par π_1 tel que $\pi_1=0$ sur $\partial\Omega_d$ on multiplie [éq A5.1-2] par π_2 quelconque, on intègre par partie, on tient compte de [éq A5.1-3] et [éq A5.1-4] et on obtient, en désignant par n la normale au bord :

$$\int_{\partial\Omega_d} M_{as}.n.\pi_2 d\Gamma=\int_{\partial\Omega_d} M_{2ext}\pi_2 d\Gamma,\quad \forall\pi_2$$

On en déduit

$$M_{as}.n=M_{2ext}\,\text{sur}\,\partial\Omega_d$$

- **Dirichlet** $PRE2$ **, Neuman** $PRE1$

C'est le cas où on impose une valeur à $PRE2$ et une valeur au flux associé à $PRE1$, en ne disant rien sur PRE1 ou en donnant une valeur à FLUN_HYDR1 de FLUX_THM_REP dans AFFE_CHAR_MECA. Appelons $M_{1_{ext}}$ cette quantité imposée, qui vaudra 0 si rien n'est dit relativement à $PRE2$. Nous noterons $p_2=p_2^{imp}$ la condition imposée sur $PRE2$.
Ceci correspond à :

$$p_{gz}=p_{as}+p_{vp}=p_{gz}^{imp}$$
$$p_2^{imp}=p_{gz}^{imp}$$

La démonstration est la même qu'au paragraphe précédent et aboutit à :

$$\left(M_{lq}+M_{vp}\right).n=M_{1ext}\,\text{sur}\,\partial\Omega_d$$

A2.3 Cas de conditions aux limites faisant intervenir des relations linéaires entre inconnues principales

Code_Aster permet d'introduire comme conditions aux limites des relations entre degrés de liberté, portés par le même nœud ou par des nœuds différents. Cette possibilité est atteinte via le mot clé LIAISON_DDL de la commande AFFE_CHAR_MECA.

Soit p_{lq}^{imp} la valeur que l'on veut imposer à la pression de liquide sur $\partial\Omega_d$. Compte tenu de [éq A5.1-5], et du choix des inconnues principales pour ce comportement, on écrit :

$$p_{gz}-p_c=p_2-p_1=p_{lq}^{imp} \qquad \text{éq A5.3-1}$$

Les relations linéaires sont traitées dans Aster par introduction de multiplicateurs de Lagrange. Ceci correspond en l'espèce à la formulation suivante :

Trouver p_1, p_2, μ tels que :

$$-\int_\Omega\left(\dot{m}_{lq}+\dot{m}_{vp}\right)\pi_1 d\Omega+\int_\Omega\left(M_{lq}+M_{vp}\right).\nabla\pi_1 d\Omega-\int_\Omega \dot{m}_{as}\pi_2 d\Omega+\int_\Omega M_{as}.\nabla\pi_2 d\Omega+$$
$$+\int_{\partial\Omega_d}\mu^*\left(p_2-p_1-p_{lq}^{imp}\right)d\Gamma+\int_{\partial\Omega_d}\mu\left(\pi_2-\pi_1-p_{lq}^{imp}\right)d\Gamma\quad\forall\pi_1,\pi_2,\mu^* \qquad \text{éq A5.3-2}$$

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

Pour faire la démonstration dans le cadre non homogène, il faudrait introduire un relèvement de la condition $p_2 - p_1 - p_{lq}^{imp} = 0$ (C'est à dire des champs particuliers vérifiant cette condition). Cela alourdit les écritures et n'apporte rien, on se place donc dans le cadre homogène $p_{lq}^{imp} = 0$.

On commence alors à prendre $\pi_1 = 0$ et $\pi_2 = 0$ sur tout le bord $\partial\Omega$ et on obtient [éq A5.1-1] et [éq A5.1-2] au sens des distributions. On multiplie alors [éq A5.1-1] par π_1 quelconque on multiplie [éq A5.1-2] par π_2 quelconque, on intègre par partie, on porte les résultats trouvés dans [éq A5.3-2] et on obtient :

$$\int_{\partial\Omega_d} (M_{lq} + M_{vp}).n.\pi_1 d\Gamma + \int_{\partial\Omega_d} M_{as}.n.\pi_2 d\Gamma + \int_{\partial\Omega_d} \mu^* (p_2 - p_1) d\Gamma + \int_{\partial\Omega_d} \mu (\pi_2 - \pi_1) d\Gamma = 0 \quad \forall \pi_1, \pi_2, \mu^* \qquad \text{éq A5.3-3}$$

Il est clair que [éq A5.3-3] redonne bien $p_2 - p_1 = p_{lq}^{imp} = 0$

En prenant de plus $\pi_2 - \pi_1 = 0$, on trouve :

$$\int_{\partial\Omega_d} (M_{lq} + M_{vp} + M_{as}).n.\pi_1 d\Gamma = 0 \quad \forall \Gamma_1$$

D'où l'on déduit :

$$(M_{lq} + M_{vp} + M_{as}).n = 0 \text{ sur } \partial\Omega_d \qquad \text{éq A5.3-4}$$

A2.4 Cas de conditions aux limites mixtes

Nous abordons dans cette section le cas des conditions aux limites mixtes (ou conditions d'échange) appliquées sous forme d'une condition de Neumann. Ces conditions peuvent porter sur l'une ou l'autre des équations de conservation ou bien sur les deux.

L'écriture des flux se fait alors sous la forme :
$(M_{lq} + M_{vp}).n = M_{1ext}$ sur $\partial\Omega_d$ avec $M_{1ext} = \lambda_{11}(p_1 - p_1^{ext}) - \lambda_{12}(p_2 - p_2^{ext})$

et/ou

$M_{as}.n = M_{2ext}$ sur $\partial\Omega_d$ avec $M_{2ext} = \lambda_{21}(p_1 - p_1^{ext}) - \lambda_{22}(p_2 - p_2^{ext})$

Cette condition est implémentée de manière explicite (la valeur de $PRE1$ et/ou $PRE2$ étant prise au temps précédent). Ces conditions mixtes sont donc appliquées comme des conditions de Neumann (cf. précédemment).

A2.5 Les cas non linéaires

Nous ne faisons ici qu'aborder des questions plus difficiles consistant à imposer soit la pression de vapeur soit la pression d'air sec. Compte tenu des relations [éq A5.1-7], [éq A5.1-8] et [éq A5.1-9] imposer une valeur sur la pression de vapeur revient à imposer une relation non linéaire sur la pression de liquide; de même qu'imposer une pression d'air sec sur la pression de gaz.

A titre d'exemple, nous abordons le cas d'un pression d'air sec imposée pour un comportement LIQU_VAPE_GAZ, et nous supposons que nous savons écrire la relation non linéaire reliant la pression de vapeur et la pression de liquide.

La relation à imposer est donc :

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

$$p_{as} = p_{gz} - p_{vp} = p_2 - p_{vp} = p_{as}^{imp}$$ éq A5.4-1

En différenciant cette relation, on trouvera une condition sur les variations virtuelles de pressions :

$$dp_{as} = dp_{gz} - \frac{\partial p_{vp}}{\partial p_{lq}} dp_{lq} = dp_{gz} - \frac{\partial p_{vp}}{\partial p_{lq}} \left(dp_{gz} - dp_c \right)$$

Soit encore

$$dp_{as} = dp_2 - \frac{\partial p_{vp}}{\partial p_{lq}} \left(dp_2 - dp_1 \right) = \frac{\partial p_{vp}}{\partial p_{lq}} dp_1 + \left(1 - \frac{\partial p_{vp}}{\partial p_{lq}} \right) dp_2$$

La formulation variationnelle serait alors :

$$-\int_\Omega \left(\dot{m}_{lq} + \dot{m}_{vp} \right) \pi_1 \, d\Omega + \int_\Omega \left(M_{lq} + M_{vp} \right) . \nabla \pi_1 \, d\Omega - \int_\Omega \dot{m}_{as} \, \pi_2 \, d\Omega + \int_\Omega M_{as} . \nabla \pi_2 \, d\Omega +$$
$$+ \int_{\partial \Omega_d} \mu^* \left(p_2 - p_{vp} - p_{as}^{imp} \right) d\Gamma + \int_{\partial \Omega_d} \mu \left(\frac{\partial p_{vp}}{\partial p_{lq}} \pi_1 + \left(1 - \frac{\partial p_{vp}}{\partial p_{lq}} \right) \pi_2 \right) d\Gamma \quad \forall \pi_1, \pi_2, \mu^*$$

Et on trouverait :

$$\int_{\partial \Omega_d} \left(M_{lq} + M_{vp} \right) . n . \pi_1 \, d\Gamma + \int_{\partial \Omega_d} M_{as} . n . \pi_2 d\Gamma + \int_{\partial \Omega_d} \mu \left(\frac{\partial p_{vp}}{\partial p_{lq}} \pi_1 + \left(1 - \frac{\partial p_{vp}}{\partial p_{lq}} \right) \pi_2 \right) d\Gamma = 0 \quad \forall \pi_1, \pi_2$$

En prenant $\frac{\partial p_{vp}}{\partial p_{lq}} \pi_1 + \left(1 - \frac{\partial p_{vp}}{\partial p_{lq}} \right) \pi_2 = 0$ on trouverait :

$$\left(1 - \frac{\partial p_{vp}}{\partial p_{lq}} \right) \left(M_{lq} + M_{vp} \right) . n - \frac{\partial p_{vp}}{\partial p_{lq}} M_{as} . n = 0$$ éq A5.4-2

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