CAST3M - Implantation d'une nouvelle loi d'évolution / loi de comportement mécanique

SEMT/LM2S
J. Kichenin et T. Charras

22 septembre 2003

Table des matières

Chapitre 1

Préambule

Le logiciel CAST3M est écrit en langage Fortran77 auquel ont été ajoutées quelques instructions. Ce nouveau langage est baptisé ESOPE et sa structure est identique à celle du Fortran. Il existe donc des sous-programmes qui s'appellent les uns les autres. En pratique la connaissance de ESOPE n'est pas demandée pour implanter une nouvelle loi simple, mais peut s'avérer utile pour des lois très compliquées. De plus CAST3M propose à l'utilisateur un langage dénommé GIBIANE qui permet de créer et manipuler des OBJETS, avec lequel on peut créer des procédures encapsulant des séquences de calcul. L'utilisation de ces procédures conserve aux jeux données proprement dits, exprimés en GIBIANE, un caractère paramétrique et synthétique.
A l'installation du logiciel, un répertoire est créé qui contient l'ensemble des fichiers source. Ainsi, le sous-programme fortran(esope), de nom ABCDEF, se trouve dans le fichier **abcdef.eso**. C'est pourquoi nous parlerons par la suite de sous-programme **abcdef.eso**. Le suffixe **.eso** est donné pour préciser que le langage est ESOPE et qu'il faudra lui appliquer la moulinette de traduction en pur Fortran77.
A priori pour compiler un sous-programmes il faut passer la commande :

```
compil abcdef
```

Ceci compile le contenu du fichier **abcdef.eso** après l'avoir traduit en pur Fortran77. Pour faire un load module exécutable de castem il suffit ensuite de passer la commande :

```
essai
```

On est conduit à travailler dans un répertoire contenant des sous-programmes modifiés et/ou créés pour le développement. L'exécution de CAST3M par la commande :

```
castem aa.dgibi
```

dans le répertoire contenant ce nouvel exécutable entrainera l'utilisation du load module. L'opérateur `UTIL` permet d'utiliser des procédures crées ou modifiées rassemblées dans un fichier **mesprocedures**, au cours de la prochaine exécution de CAST3M :

```
util proc 'mesprocedures' ;
fin ;
```

Chapitre 2

Introduction

Pour des raisons de régularité et de généralité, un nouvel opérateur `COMP` — comportement — remplace l'opérateur `ECOU` — écoulement. Ce document annule et remplace le rapport [5].
L'implantation d'une nouvelle loi d'évolution et en particulier d'une loi de comportement mécanique nécessite la création d'un nouvel exécutable `cast` obtenu à partir de la modification et/ou l'ajout d'un certain nombre de sous-programmes, mais également la mise à jour de procédures de calculs telles `PASAPAS`, et bien entendu la définition de jeux de données `moncas.dgibi` appropriés. Un tel travail s'articule autour des points suivants, repris dans la construction de ce document :

- $1^{eme}phase$: définition d'un nouveau modèle de matériau — type de formulation ;
- $2^{eme}phase$: description des champs physiques participants à la loi — en mécanique, déformations, contraintes, caractéristiques du matériau, variables internes, ... ;
- $3^{eme}phase$: calcul incrémental.

Les deux premières phases servent à initialiser le logiciel pour qu'il soit capable automatiquement de récupérer les données fournis par l'utilisateur : le développement porte sur les opérateurs `MODE, MATE`, ...La troisième phase correspond à la programmation de la loi proprement dite dans l'opérateur `COMP` — qui remplace l'opérateur `ECOU` —, et à la mise au point des procédures qui facilitent l'utilisation de la loi — par exemple `PASAPAS`.

Trois possibilités sont proposées pour l'implantation :

1. description des champs dans des sous-programmes — listes des noms des composantes — : pour les phases 1 et 2 rien n'est changé vis-à-vis de l'opérateur `ECOU` [5]. Puis modification des sous-programmes de `COMP` avec appel éventuel à de nouveaux sous-programmes dans **coml7.eso** ou **coml8.eso**. il peut être nécessaire de modifier `PASAPAS` pour adresser les données convenables à `COMP`.
2. modèles VISCO_EXTERNE : en mécanique, un intégrateur dédié au fluage polynomial permet de traiter des expressions où interviennent des champs paramètres

et/ou des variables internes non-standards [6-7]. L'intégrateur est contenu dans le sous-programme **ccreep.eso**. Les listes des noms de composantes de ces paramètres "externes" et de ces variables internes sont associées à l'objet `MMODEL` dans le jeu de données (voir exemple `creep04_cisXY.dgibi`). Les champs de paramètres externes servent à composer le chargement figurant dans la table argument de la procédure PASAPAS ; celle-ci sait gérer le calcul a priori sans modification.

3. modèle NON_LINEAIRE UTILISATEUR : en mécanique, CAST3M offre la possiblilité d'exploiter des sous-programmes fortran77 au standard UMAT d'ABAQUS, permettant de traiter des comportements très variés [6-7]. La description des composantes matériaux, des noms de variables internes et des noms de champs de paramétrages est possible dans le fichier utilisateur (voir exemple `umat05.dgibi`). Il faut modifier le contenu du sous-programme **umat.eso** qui existe dans CAST3M. La procédure `PASAPAS` sait gérer le calcul a priori sans modification .

Les variantes 2 et 3 — réservées à la mécanique — dispensent de réaliser les phases 1 et 2 du développement et conduisent à intervenir en phase 3 sur des sous-programmes spécifiques : les lois de comportement mécanique complexes seront difficiles à réaliser dans ce cadre. La variante 1 offre beaucoup plus de possibilités.
Ce document essaye de donner un aperçu sur l'organisation du logiciel, il donne donc parfois des informations qui ne sont pas strictement nécessaires à l'implantation d'une nouvelle loi de comportement.

Chapitre 3

Phase 1 - Définition d'un nouveau modèle de matériau

Il est nécessaire de mettre à jour l'opérateur `MODE`, qui crée un objet `MMODEL`, association d'une géométrie (objet MAILLAGE), d'une formulation physique (ex : MECANIQUE), d'un comportement linéaire (ex : ELASTIQUE ANISOTROPE), d'un comportement non-linéaire (ex : VISCOPLASTIQUE CHABOCHE) et d'une formulation élément fini (ex : CUB8,DKT). C'est le sous-programme `modeli.eso` qui effectue le travail en appelant d'autres sous-programmes. Aucun calcul n'a lieu pendant cette étape, seule la description d'un nouveau type de données est à faire. Dans de nombreux opérateurs où un objet `MMODEL` figure en argument, la première action consiste à analyser les informations de celui-ci, ce qui oriente la suite des calculs. En mécanique, l'utilitaire contenu dans le sous-programme **nomate.eso**, qui attribue un numéro à la loi de comportement, est systématiquement utilisé ; il convient donc de le mettre à jour, si besoin est, en même temps que `MODE`.

3.1 Variantes 2 et 3

Il n'est pas nécessaire d'intervenir dans l'opérateur `MODE` ni l'utilitaire **nomate.eso** — l'utilisateur attribue un numéro.

– exemple syntaxe modèle VISCO_EXTERNE dans le fichier de données :

```
LCPAR24 = MOTS 'TFIS' 'TUO2' 'FACF' 'DSIU' 'DGRA' ;
LCVAR24 = MOTS 'PSUP' 'QSUP' ;

mo_uti2 = MODE mail1 'MECANIQUE' 'ELASTIQUE' 'ISOTROPE'
                'VISCO_EXTERNE' 'GENERAL' 'NUME_LOI' 24
                'PARA_LOI' LCPAR24 'C_VARINTER' LCVAR24 ;
```

– exemple syntaxe modèle NON_LINEAIRE UTILISATEUR dans le fichier de données :

```
LCM05_0 = MOTS 'YOUN' 'NU  ' ;

LCM05_1 = MOTS 'E001' 'S001' 'E002' 'S002' 'E003' 'S003'
               'E004' 'S004' 'E005' 'S005' 'E006' 'S006'
               'E007' 'S007' 'E008' 'S008' 'E009' 'S009' ;

LCMAT05 = LCM05_0 et LCM05_1 ;
LCVAR05 = MOTS 'EPSE' 'ENXX' 'ENYY' 'ENZZ' 'GNXY' 'GNXZ' 'GNYZ' ;

mo_util = MODE vol1 'MECANIQUE' 'ELASTIQUE' 'ISOTROPE'
                    'NON_LINEAIRE' 'UTILISATEUR' 'NUME_LOI' 5
                    'C_MATERIAU' LCMAT05 'C_VARINTER' LCVAR05 ;
```

3.2 Variante 1

Liste des formulations disponibles (définies dans **modeli.eso** dans le tableau `MOFORM`). Au besoin il faudra en ajouter une nouvelle.

```
'THERMIQUE'  'MECANIQUE'  'LIQUIDE'       'CONVECTION'
'POREUX'     'DARCY'      'FROTTEMENT'    'RAYONNEMENT'
'MAGNETODYNAMIQUE'        'NAVIER_STOKES' 'MELANGE'
```

modeli.eso appele un sous-programme **modelj.eso** (j=1,n) suivant la formulation lue (ex : "model2.eso" est appele dans le cas de la formulation MECANIQUE). Ces sous-programmes servent à dresser la liste de tous les mots acceptés et ceci en fonction de la formulation.

Dans le cas MECANIQUE, (géré par **model2.eso**) le seul comportement linéaire est : ELASTIQUE. La liste des options disponibles aujourd'hui (définies dans **modela.eso**) est :

```
'ISOTROPE'     'ORTHOTROPE'         'ANISOTROPE'    'PORPOR'
'HOMOGENEISE'  'UNIDIRECTIONNEL'    'SECTION'       'ARMATURE'
```

Les comportements non-linéaires disponibles aujourd'hui sont définis dans **modnli.eso**, la liste est :

```
'PLASTIQUE'        'FLUAGE'            'VISCOPLASTIQUE'    'VISCO_EXTERNE'
'ENDOMMAGEMENT'    'PLASTIQUE_ENDOM'    'NON_LINEAIRE'
```

Remarque : La classification en catégorie PLASTIQUE FLUAGE VISCO... est très formelle. Elle permet de donner un semblant de structuration au logiciel car elle oriente certaines actions de type déclaratives (et non calculatoires). Par exemple les noms des composantes d'un modèle de comportement PLASTIQUE seront définies dans **idplas.eso** *alors que celles pour un modèle de type fluage seront définies dans* **idflua.eso**. *Cependant, vis-à-vis du traitement dans* `COMP` *ou* `PASAPAS` *ces catégories sont indifférentes.*

Suivant le comportement non-linéaire **model2.eso** se branche sur un des sous-programmes :

```
PLASTIQUE         ----->  modpla.eso
FLUAGE            ----->  modflu.eso
VISCOPLASTIQUE    ----->  modvis.eso
ENDOMMAGEMENT     ----->  modend.eso
PLASTIQUE_ENDOM   ----->  modple.eso
NON_LINEAIRE      ----->  modenl.eso
VISCO_EXTERNE     ----->  modvix.eso
```

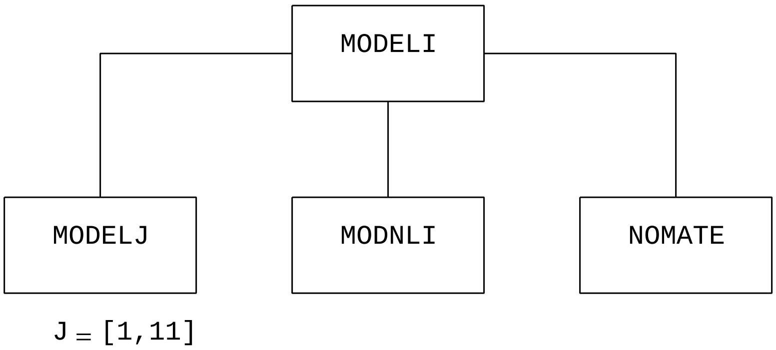
Dans chacun de ces sous-programmes on initialise la liste des modèles disponibles. Par exemple dans **modflu.eso** on trouve :

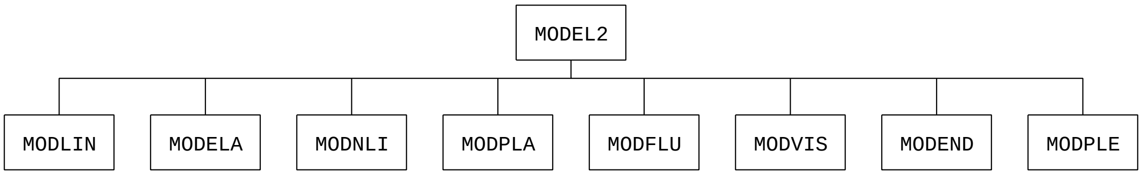
```
'NORTON'     'BLACKBURN'    'RCCMR_316'     'RCCMR_304'
'LEMAITRE'   'POLYNOMIAL'   'BLACKBURN_2'   'CERAMIQUE'
'MAXWELL'    'COMETE'       'CCPL'          'X11'
'BPEL_RELAX''BETON_URGC'    'SODERBERG'     'MAXOTT'
```

Enfin **nomate.eso** affecte un numero au modèle de matériau et ensuite quelques vérifications sont faites dans **modeli.eso**.

Organigramme des procédures et sous programmes utilisés pour la description d'un modèle.

Exemple : J=2 $\Rightarrow$ MODEL2 $\Leftrightarrow$ "Formulation mécanique"





3.3 Sous-programmes à modifier

En résumé :

MODELI.ESO Création de l'objet MMODEL. En général rien à faire ici sauf dans le cas d'une nouvelle formulation simple ou couplée.

1. En cas d'une nouvelle formulation, différente des formulations simples ou couplées déjà existantes, aiguiller le programme sur un nouveau sous programme MODELJ (J>13) pour définir les noms des modèles de matériau et les noms de type d'éléments finis utilisables par une telle formulation. Exemple en mécanique des fluides :

```
ELSE IF(LESFOR(1).EQ.'NAVIER_STOKES') THEN
   CALL MODE11(MOPROP,NPROP,MOTEF,NBTEF,N1MAX,N2MAX)
```

2. En cas d'une nouvelle formulation mécanique, différente de celles déjà existantes dans MODNLI.ESO, on doit charger la valeur par défaut dans `LESPRO(NMAT)`. Dans ce cas les lignes à ajouter sont de la forme :

```
(    ....       ELSE IF (IPLAC.EQ.7) THEN
                 --
                 --                  )
            ELSE IF (IPLAC.EQ.8) THEN
*
*      Par defaut : NOUVFORMULATION ---->  NOUVCOMPORTEMENT
*
                NMAT=NMAT+1
                LESPRO(NMAT)='NOUVCOMPORTEMENT'
```

MODELJ.ESO Création d'un objet `MMODEL` spécifique à une formulation nouvelle.

- Dans le cas d'une formulation déjà existante, i.e. thermique, mécanique, liquide, convection, poreux etc.., modifier un des sous programmes appelés par MODELJ($1 \leq J \leq 11$), pour la définition des noms des modèles de matériaux et des noms des types d'éléments finis utilisables par une telle formulation.

Remarque : Pour introduire un nouveau modele de comportement mécanique il faudrait modifier MODEL2 mais la programmation de MODEL2 entraine que les modifications sont à faire dans les sous-programmes appelés (modnli, modpla, mod...)

MODNLI.ESO Définition des noms de comportement non-linéaires dans le cas d'une formulation mécanique. Ce sous-programme n'est à modifier que pour ajouter un nouveau type de comportement non-linéaire (plastique, fluage, viscoplastique,..). **En général rien à faire ici non plus ! Par contre il faut modifier un des sous-programmes modpla, modflu, modvis, modend, modple ou modenl.**

MODPLA.ESO Définition des comportements de type PLASTIQUE, ajouter le nouveau. Par exemple, pour introduire un modèle de comportement plastique s'appelant "TRESMOU" on aura besoin d'insérer les lignes :

```
NMOD=62
  .
  .
MOMODL(62)='TRESMOU'
```

MODFLU.ESO Définition des comportements de type FLUAGE, ajouter le nouveau.

MODVIS.ESO Définition des comportements de type VISCOPLASTIQUE, ajouter le nouveau.

MODEND.ESO Définition des comportements de type ENDOMMAGEMENT, ajouter le nouveau.

MODPLE.ESO Définition des comportements de type ENDOMMAGEMENT, ajouter le nouveau.

MODENL.ESO Définition des comportements de type NON_LINEAIRE, ajouter le nouveau.

MODVIX.ESO Définition des comportements de type VISCO_EXTERNE, ajouter le nouveau.

NOMATE.ESO Attribue un numéro à un modèle de matériau, ce numéro sera celui qui gouvernera l'exécution de l'écoulement non-linéaire faite dans le sous-programme approprié de **coml7eso**. Pour cela il faut initialiser la variable `INATU` (qui s'appelera `INPLAS` dans **coml6.eso**) par une séquence d'instructions comme suit :

```
                          ELSE IF (IMOD.EQ.??) THEN
*      NOUVCOMPORTEMENT
                             INATU=120
```

Cette séquence d'instructions est à placer en fin de la séquence de `IF ... ELSEIF...` qui est elle-même située après l'appel au sous-programme MODXXX (XXX = ELA,PLA,FLU,VIS,END,PLE) concerné. La valeur qu'il faut confronter à `IMOD` est à trouver dans le sous-programme MODXXX concerné en face de NOUVCOMPORTEMENT. Par exemple dans le cas déjà cité de l'incorporation d'un modèle plastique "TRESMOU", il faudra insérer derrière :

```
                          ELSE IF(IMOD.EQ.61) THEN
*     JOINT_COAT
                             INATU=119
```

les lignes suivantes : (la valeur 120 représente le fait que c'est le 120ème modèle de comportement toutes catégories confondues, information tirée des commentaires du sous-programme **nomate.eso**)

```
                          ELSE IF (IMOD.EQ.62) THEN
*      TRES_MOU
                             INATU=120
```

Chapitre 4

Phase 2 - Champs relatifs — caractéristiques du matériau

La loi physique associée à l'objet `MMODEL` est une relation entre un ensemble de champs physiques décrits généralement dans `CAST3M` à l'aide de champs par éléments `MCHAML`. On notera que certains champs sont étroitement liés au `MMODEL` — les variables internes pour un modèle de mécanique non-linéaire —, alors que d'autres ont une portée plus générale — température, temps, ...L'utilisateur et le logiciel identifient les variables à l'aide des noms de composantes de ces `MCHAML`. Jusqu'à présent les noms de composantes devaient être écrits dans les sous-programmes. Les modèles VISCO_EXTERNE et UTILISATEUR permettent désormais de renseigner certains noms dans le jeu de données. Nous citons dans ce chapitre des utilitaires qui servent à décrire le noms des composantes des champs par élément associés principalement à la formulation mécanique. Ces utilitaires sont appelés par des opérateurs tels `MODE, MATE, EXTR, EXIS, COMP, BSIGM, RIGI,` ...

4.1 Opérateur `VARI`

Cet opérateur ne crée pas de nouveaux champs physiques, mais calcule la valeur effective des champs selon celle des paramètres, en utilisant des objets `MMODEL et MCHAML` en argument. Une application fréquente est la variation de caractéristiques matériaux en fonction de la température : l'utilisateur fait figurer la courbe des caractéristiques selon la température dans un `MCHAML` à partir duquel `VARI` établit le `MCHAML` des valeurs pour une température donnée. Il faut envisager de modifier `VARI` si l'un des champs de paramètres est définie par une fonction qui ne peut se traduire ni par un objet `EVOLUTION`, ni par un objet `NUAGE` à deux colonnes : plus clairement, `VARI` sait réaliser une interpolation linéaire dans une courbe, ou entre une famille de courbes.
`CAST3M` dispose désormais de l'utilitaire **comput.eso**, appelé par `VARI`, qui facilite les calculs d'instanciation plus complexes. L'utilisateur substitue un développement approprié au contenu du sous-programme standard et crée un nouvel exécutable. Dans le jeu de données, des instructions telles que

```
mo_uti2 = MODE mail1 'MECANIQUE' 'ELASTIQUE' 'ISOTROPE' ;
lpyoun = MOTS 'T   ' 'PORO' 'YOGC' ;
lpnu   = MOTS 'T   ' 'PORO' 'YOGC' ;
ma_uti2 = MATE mo_uti2 'YOUN' lpyoun 'NU  ' lpnu   'RHO ' xrho ;
```

permettent de construire le `MCHAML` de caractéristiques donné en argument de `VARI`, qui fait alors appel à `comput.eso` pour instancier `YOUN et NU`. Si la température `T` apparaît, il est nécessaire qu'elle soit placée en première position dans les objets `LISTMOTS` tels `lpyoun`.

4.2 Sous-programmes utilitaires

Ceux-ci doivent notamment être mis à jour si l'on formule un nouvel élément fini :

IDCONT.ESO Ce sous-programme renvoie le nombre et le nom des composantes du tableau des contraintes.

IDDEFO.ESO Ce sous-programme renvoie le nombre et le nom des composantes du tableau des déformations.

IDDEIN.ESO Ce sous-programme renvoie le nombre et le nom des composantes du tableau des déformations inélastiques.

IDGRAD.ESO Ce sous-programme renvoie le nombre et le nom des composantes du tableau des gradients de déplacements.

4.3 Variables internes

Le sous-programme **idvari.eso** renvoie le nombre et le nom des composantes des variables internes. Cependant la variante VISCO_EXTERNE dispose de 4 variables internes prédéfinies `ECO, ESWO, P et QTLD`, auxquelles l'utilisateur peut adjoindre d'autres composantes dans le jeu de données `moncas.dgibi`. Dans cette variante, l'utilisateur ne dispose que des composantes matériau élastique `YOUN, NU, ALPH et RHO`. Les autres seront calculées dans **creep.eso**.
La variante NON_LINEAIRE UTILISATEUR conduit à spécifier les composantes des champs de variables internes et de caractéristiques dans le jeu de données. Aucune composante par défaut.

– syntaxe pour les caractéristiques du modèle UTILISATEUR, suite du Chapitre 3 :

```
ma_u0 = MATE mo_util 'YOUN' youn0 'NU  ' 0.33 ;

epsp2 = (prog 0. 1. 2. 3. 4. 5.)*1.e-2 ;
```

```
Y1 = (prog 0. 200. 250. 280. 300. 310.)*1.e6 ;
ma_u11 = MATE mo_util 'E001' (EXTR epsp2  1) 'S001' (EXTR Y1  1)
                      'E002' (EXTR epsp2  2) 'S002' (EXTR Y1  2)
                      'E003' (EXTR epsp2  3) 'S003' (EXTR Y1  3)
                      'E004' (EXTR epsp2  4) 'S004' (EXTR Y1  4)
                      'E005' (EXTR epsp2  5) 'S005' (EXTR Y1  5) ;

ma_util = ma_u0  et ma_u11
```

Pour ce modèle, seul le type `REAL*8` est admis pour le champ de caractéristiques utilisé par `COMP` ; ainsi une courve est traduite par une double liste étiquetée et non par un objet `EVOLUTION`.

Dans le cas le plus général, on est conduit à modifier **idvari.eso** quand on veut introduire un nouveau modèle de comportement.

IDVARI.ESO Déclarations des noms de composantes des variables internes :

- pour un matériau linéaire : le paramètre `MATEPL` est mis à 0
- pour un matériau non linéaire :

1. Matériau plastique : le paramètre `IPLAC` correspond au type de plasticité défini dans **modpla.eso**. Le paramètre `MATEPL` correspond au numéro du modèle de matériau défini dans `nomate.eso`. En cas de rajout d'un nouveau type de plasticité, charger les noms des variables internes dans le vecteur `LESOBL` et les variables facultatives dans `LESFAC`. Charger la variable `NBROBL` au nombre total de variables internes obligatoires et `NBRFAC` le nombre de variables internes facultatives. Les variables internes sont chargées dans l'un des sous-programmes **idvar6.eso et idvar7.eso**.
2. Matériau fluage : `IPLAC et MATEPL` sont définis respectivement dans **modflu.eso et nomate.eso**. En cas de rajout d'un nouveau type de fluage, charger les variables internes dans l'un des sous-programmes **idvar1.eso, idvar2.eso ou idvar3.eso**.
3. Matériau viscoplastique : `IPLAC et MATEPL` sont définis respectivement dans **modvis.eso et nomate.eso**. En cas de rajout d'un nouveau type de viscoplasticité, charger les variables internes dans le sous programme **idvar4.eso**.
4. Matériau endommageable : `IPLAC et MATEPL` sont définis respectivement dans **modend.eso et nomate**. En cas de rajout d'un nouveau type d'endommagement, charger les variables internes dans **idvar5.eso**.
5. Matériau plastique endommageable : `IPLAC et MATEPL` sont définis respectivement dans **modple.eso et idplen.eso**. En cas de rajout d'un nouveau type d'endommagement, charger les variables internes dans **idvar6.eso**.
6. Matériaux VISCO_EXTERNE et NON_LINEAIRE UTILISATEUR : non concernés

En résumé : Il faut modifier idvari.eso pour définir `MATEPL` et pour diriger l'appel vers l'un des IDVARX, — par défaut c'est **idvar6.eso** qui sera appelé ;

modifier bien entendu **idvarx.eso**. (Exemple en Annexe).

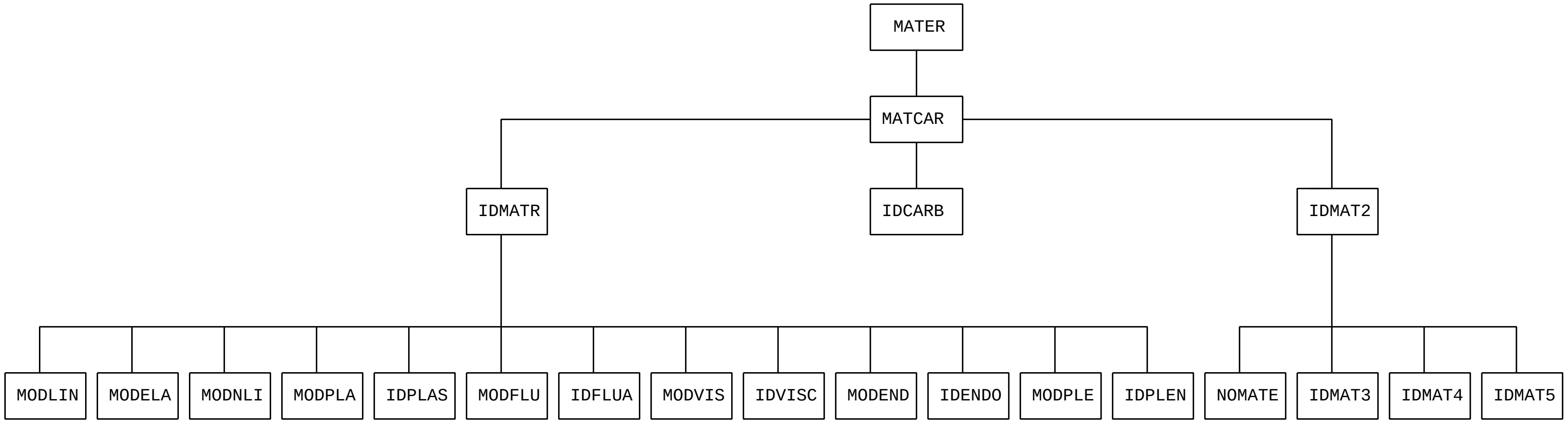
Remarque : La variable interne représentant la déformation plastique équivalente doit de préférence s'appeler `EPSE`, car c'est ce nom qui est utilisé, par défaut, par la procédure `PASAPAS` .

4.4 Caractéristiques

La lecture des caractéristiques des matériaux se fait par l'opérateur `MATE` qui appele le sous-programme fortran **matcar.eso** (via **mater.eso**). Celui-ci commence par appeler le sous-programme **idmatr.eso** qui fabrique une liste de données obligatoires et une liste de données facultatives qui sont fonction de la formulation, du comportement linéaire et du comportement non-linéaire.

Aprés avoir donné l'organigramme des séquences d'appels des différents sous-programmes (peu sont à modifier), nous dresserons la liste des actions à mener.

Organigramme des procédures et sous-programmes utilisés pour la gestion des caractéristiques du matériau.


MATER
MATCAR
IDMATR
IDCARB
IDMAT2
MODLIN
MODELA
MODNLI
MODPLA
IDPLAS
MODFLU
IDFLUA
MODVIS
IDVISC
MODEND
IDENDO
MODPLE
IDPLEN
NOMATE
IDMAT3
IDMAT4
IDMAT5

MATER.ESO et MATCAR.ESO Création d'un champ de caractéristiques matérielles et/ou géométriques. Rien à faire ici.

IDMATR.ESO Chargement des noms de composantes de matériaux selon le type de formulation. Cette étape est aussi purement descriptive, il s'agit de définir la liste des composantes obligatoires et facultatives à lire puis à la rigueur de faire quelques vérifications ou mises en forme. Certaines données peuvent etre fournies par l'utilisateur sous une forme facile pour lui puis transcrite sous une autre forme pour le logiciel (par exemple la lecture des directions d'orthotropie dans un repère cylindrique permet de fabriquer pour chaque élément les composantes de la loi de Hooke dans un repere cartesien), dans ce cas il faudra faire la mise en forme dans un nouveau sous-programme qui sera appelé par IDMAT2.
Ainsi :

1. Si l'on a ajouté un nouveau comportement linéaire dans **modela.eso**, alors il faut déclarer le nom des composantes des caractéristiques du matériau dans **idmatr.eso** (exemple en Annexe). *Cette situation est très rare quand on introduit un modèle de comportement sans introduire simultanéement de nouveaux éléments.*
2. Pour la partie non-linéaire du comportement, la déclaration des noms de caractéristiques du comportement se fait dans le sous-programme appelé par **idmatr.eso** qui suivant la formulation est (voir annexe) :

```
PLASTIQUE         ----->  idplas.eso
FLUAGE            ----->  idflua.eso
VISCOPLASTIQUE    ----->  idvisc.eso
ENDOMMAGEMENT     ----->  idendo.eso
PLASTIQUE_ENDOM   ----->  idplen.eso
NON_LINEAIRE      ----->  idelnl.eso
```

IDPLAS.ESO Chargement des noms de composantes de matériaux plastiques pour chaque type de plasticité définie dans MODPLA. Ajouter les noms de composantes pour le NOUVCOMPORTEMENT plastique.

IDFLUA.ESO,IDVISC.ESO,IDENDO.ESO,IDPLEN.ESO,IDELNL.ESO
Similaires à **idplas.eso** en cas de FLUAGE,VISCOPLASTICITE, ENDOMMAGEMENT et comportement PLASTIQUE ENDOMMAGEABLE, ou NON_LINEAIRE — autre que UTILISATEUR.

IDMAT2.ESO Création du `MCHAML` correspondant à un matériau orthotrope.

Remarque : s'il est inutile de prétraiter l'information donnée par l'utilisateur ne rien faire ici ni dans les sous-programmes suivants

IDMAT3.ESO Définition des directions d'orthotropie d'une coque en 3D, et d'un élément joint 3D.

IDMAT4.ESO et IDMAT5.ESO Calcul des cosinus directeurs des axes d'orthotropie par rapport au repère local de l'élément.

Chapitre 5

Phase 3 - Calcul incrémental, opérateur COMP

5.1 Présentation

Le recours à des procédures encapsulant l'appel à l'opérateur `COMP`, telles que `PASAPAS`, est recommandé dès que l'on étudie des phénomènes évolutifs. Il peut être nécessaire de modifier ces procédures pour utiliser de nouvelles lois en ajoutant les instructions traitant les nouveaux champs nécessaires. Le détail de l'algorithme utilisé dans **pasapas.procedur et increme.procedur** est décrit dans [2].
Nous décrivons dans ce chapitre la programmation de l'opérateur `COMP` dont la notice est :

```
Opérateur COMP                           Voir aussi : MODE MATE
--------------                                        CARA

CHE1 = COMP  MOD1 CHE2 CHE3 ;

Objet :
-------

L'opérateur COMP établit l'évolution des champs relatifs à un
modèle physique, lois de comportement, d'état ou bien cinétique,
entre un instant initiale et un instant final.
Il est nécessaire de préciser l'objet MMODEL qui induit les lois,
l'état initial des champs nécessaires à la formulation contenue
dans le modèle et l'état final des variables de controle.

Applications possibles : lois de comportement en mécanique,
transitions de phase en métallurgie ...

Commentaire :
```

```
-------------

MOD1  : type MMODEL

CHE2  : type MCHAML, ensemble des grandeurs décrivant l'état
        initial pour chaque modèle élémentaire, les champs étant
        identifiés par des noms de composantes (en 4 lettres)
        et de constituants. Sont inclus notamment la date 'TEMP',
        la température 'T   '.

CHE3  : type MCHAML, ensemble des grandeurs décrivant l'état
        final pour chaque modèle élémentaire, les champs étant
        identifiés de la meme maniere que ci-dessus.
        Sont inclus notamment la date 'TEMP', la température 'T   '.

CHE1  : type MCHAML, ensemble des grandeurs décrivant l'état final.


Remarques :
-------------

Mecanique : Les contraintes, variables internes, déformations
            inélastiques, déformations totales, ainsi éventuellement
            que les caractéristiques matériau et géométrique, les
            parametres externes du modele (s'il en existe) et autres
            grandeurs relatives à l'état initial sont rangées dans CHE2.
            La déformation totale, les caractéristiques de matériau et
            géométriques, les parametres externes du modele (s'il en
            existe) relatifs à l'état final sont rangés dans CHE3.
            CHE1 contient alors entre autre les nouvelles contraintes,
            variables internes et déformations inélastiques.

Métallurgie : La température, le temps  et les caractéristiques
            matériau initiales, telles les proportions de phases ou
            les tailles de grain, sont rangées dans CHE2. La
            température finale est rangée dans CHE3. CHE1 contient
            les caractéristiques finales, notamment les proportions
            de phases.
```

5.2 Organigramme

Le propos de COMP est de traiter au niveau de chaque volume élémentaire représentatif du milieu des lois et relations liant des grandeurs physiques a priori quelconques.

L'objet MMODEL induit le calcul réalisé. Les objets MCHAML représentent précisément le milieu puisqu'on peut décrire une grandeur en tout point à partir des valeurs aux points de Gauss et des fonctions de forme. On distingue l'état initial — CHE2 — et l'état final — CHE1 reprend les valeurs de CHE3 complétées de celles calculées.

L'opérateur COMP est associé à un certain nombre d'objets temporaires définis dans DECHE.INC. Les variables et tableaux nommés dans ces objets mettent l'accent sur les calculs thermo-poro-mécaniques. **Cependant toute grandeur figurant dans CHE2 ou CHE3 est collectée**. Le segment DECHE permet de recenser toutes les données fournies.

L'opérateur se compose de 4 niveaux de boucles imbriquées dont la fonction est de trier les données, renseigner les objets temporaires wrk52 , wrk522, wrk53 et wrk54, réaliser les calculs avec ceux-ci puis ranger les résultats.

```
                      STRUCTURE INFORMATIQUE DE COMP

- appel à "coml.eso"
|  - lecture des données
|  - décompose les MCHAML de données en objets temporaires DECHE
|  - appel à "coml2.eso"
|     |  - boucle sur les formulations envisageables (voir notice MODE)
|     |  - boucle sur les zones de modèles élémentaires
|     |    *  - reconstruit les champs de caractéristiques pour les états 1
|     |    *    et 2 afin de réaliser si besoin des projections
|     |    *  - trie les DECHE associés à la sous-zone, change au besoin les
|     |    *    points supports et projette éventuellement les grandeurs dont
|     |    *    le support n'est pas approprié
|     |    *  - appel à "coml6.eso"
|     |    *    |   - initialise et renseigne wrk53
|     |    *    |   - appel à "comouw.eso" : initialise wrk52 et wrk522
|     |    *    |   - appel à "comcri.eso" : crée les DECHE associées aux
|     |    *    |     grandeurs à calculer
|     |    *    |   - appel à "comtri.eso" : vérifie le type des variables
|     |    *    |     des MELVAL
|     |    *    |   - appel à "comdef.eso" : renseigne wrk53 et les
|     |    *    |     segments IECOU et NECOU
|     |    *    |   - boucle sur les éléments
|     |    *    |     *   - appel à "comrot.eso" si matériau non isotrope
|     |    *    |     *   - boucle sur les points de calcul
|     |    *    |     *     *  - appel à  "comval.eso" : renseigne wrk52,
|     |    *    |     *     *     wrk53, wrk522 données aux points de GAUSS
```

```
|    |    *    |     *     *  - appel à "comara.eso" : réorganisation
|    |    *    |     *     *  - appel à "coml7.eso"
|    |    *    |     *     *    |   - appel à un sous-programme associé à la
C    C    *    C     *     *    C     loi ou la relation physique induite par
O    O    *    O     *     *    O     le modèle selon la formulation puis la
M    M    *    M     *     *    M      valeur de INPLAS pour la mécanique
L    L    *    L     *     *    L     (dans ce cas, le numéro du comportement
|    2    *    6     *     *    7       est attribué dans "nomate.eso")
|    |    *    |     *     *    |    - appel à "coml8.eso" qui relaie
|    |    *    |     *     *    |      "coml7.eso" pour la mécanique
|    |    *    |     *     *  - fin de "coml7.eso"
|    |    *    |     *     *  - appel à "defer1.eso" puis "defer2.eso" pour
|    |    *    |     *     *    messages d'erreur
|    |    *    |     *     *  - appel à "comsor.eso" qui range les résultats
|    |    *    |     *     *    dans les MELVAL
|    |    *    |     *   - fin boucle sur points de calcul
|    |    *    |     *   - appel si besoin à "compou.eso" (poutres et tuyaux)
|    |    *    |   - fin de boucle sur les éléments
|    |    *    |   - appel à "comfin.eso" : fermeture et destruction des
|    |    *    |     objets temporaires
|    |    *  - fin de "coml6.eso
|    |    *  - met à jour l'inventaire des DECHE
|    |    *  - désactivation et suppression de segments
|    |  - fin de boucle sur les zones de modèles élémentaires
|    |  - fin de boucle sur les formulations
|  - fin de "coml2.eso"
|  - construction du MCHAML des résultats
|  - compression des MELVAL constants
|  - destruction des DECHE
|  - écriture des résultats pour gibiane (ecrobj.eso)
-fin de "coml.eso"
```

Le contenu de DECHE.INC :

```
* segment deroulant le mcheml
       SEGMENT,DECHE
         INTEGER INDEC
         CHARACTER*16 CONDEC
         CHARACTER*8 PHADEC
         INTEGER  IMADEC, IELDEC, IFODEC
         INTEGER INFDEC(N3)
         CHARACTER*8 NOMDEC
         CHARACTER*16 TYPDEC
```

```
       ENDSEGMENT
       POINTEUR DEC1.DECHE,DEC2.DECHE
* INDEC : indice du MCHELM
* CONDEC : constituant
* PHADEC : phase
* IMADEC : pointeur maillage
* IELDEC : pointeur melval
* IFODEC : CORRESPOND A IFOCHE (voir MCHAML)
* INFDEC : CORRESPOND A INFCHE (voir MCHAML)
* NOMDEC : nom de composante
* TYPDEC : type du champ

c segment des noms de composantes relatives a un champ physique
      SEGMENT NOMID
        CHARACTER*8 LESOBL(NBROBL),LESFAC(NBRFAC)
      ENDSEGMENT
* LESOBL : compoantes obligatoires
* LESFAC : composantes facultatives

* segment des types de composantes relatives a un champ physique
      SEGMENT NOTYPE
        CHARACTER*16 TYPE(NBTYPE)
      ENDSEGMENT

* segment des adresses des deche associes a un NOMID
      segment pilnec
       integer pilobl(mobl,mran),pilfac(mfac,mran)
      endsegment

c segment des pointeurs sur nomid et pilnec
      segment domdec
        integer jnomid, jilnec
      endsegment

* pile des domdec (les indices se referent au DATA LISMOT ci-dessous)
      segment piluc(iiluc)
* pile des deche concernes par un modele elementaire
      segment pilcon(iilcon)
* pile de tous les deche correspondant aux melvals vus par l utilisateur
       segment pilmel(iimel)

*
c information sur l element fini
```

```
      SEGMENT INFO
       INTEGER INFELE(JG)
      ENDSEGMENT

C  liste de reference identifiant les champs
      PARAMETER (NMOT=25)
      CHARACTER*8 LISMOT(NMOT),MOT
      DATA LISMOT / 'NOEUD   ', 'GRAVITE ', 'RIGIDITE', 'MASSE   ',
     1              'STRESSES', 'DEPLACEM', 'FORCES  ', 'REACTUAL',
     1              'FVOLUMIQ', 'GRADIENT', 'CONTRAIN', 'DEFORMAT',
     1              'MATERIAU', 'CARACTER', 'TEMPERAT', 'PRINCIPA',
     1              'MAHOOKE ', 'HOTANGEN', 'DILATATI', 'VARINTER',
     1              'GRAFLEXI', 'VONMISES', 'VINMETAL', 'DEFINELA',
     1              'PARAMEXT'/

* segments pour passer les informations au point d integration

* donnees brutes en entree de COMP, completees par les valeurs calculees
      SEGMENT WRK52
* temps initial, final
        REAL*8 temp0, tempf
*
        REAL*8 scal0(nsca), scalf(nsca)
* deplacements
        REAL*8 depl0(ndep), deplf(ndep)
* forces
        REAL*8 forc0(nfor), forcf(nfor)
* gradients
        REAL*8 grad0(ngra), gradf(ngra)
* contraintes
        REAL*8 SIG0(NSTRS), SIGF(NSTRS),DSIGT(NSTRS)
* deformations totales
        REAL*8 epst0(NDEFO),DEPST(NDEFO),epstf(NDEFO)
* caracteristiques materiau
        REAL*8 xmat0(ncara),XMAT(NCARA),xmatf(NCARA),
     &   VALMAT(NCARA),valma0(ncara)
c type et nom de composantes relatifs a VALMAT et valma0
        CHARACTER tyval(ncara)*16, commat(ncara)*8
c pointeur sur les melval de caracteristiques materiau a l etat final
        INTEGER ivalma(ncara)
* caracteristiques geometriques
        REAL*8 xcar0(ncarb),XCARB(NCARB),xcarbf(NCARB),VALCAR(NCARB)
c type et nom de composantes pour les caracteristiques
```

```
        CHARACTER tycar(ncarb)*16,comcar(ncarb)*8
* temperatures
        REAL*8 ture0(ntur), turef(ntur)
* contraintes principales
        REAL*8 prin0(npri), prinf(npri)
* matrices de hook
        REAL*8 maho0(nmah), mahof(nmah)
* matrices hook tangent
        REAL*8 hota0(nhot), hotaf(nhot)
* variables internes
        REAL*8 VAR0(NVARI),VARF(NVARI)
* gradients flexion
        REAL*8 graf0(ngrf), graff(ngrf)
*
        REAL*8 rhas0(nrhi), rhasf(nrhi)
* deformations inelastiques
        REAL*8 DEFP(NDEIN),EPIN0(NDEIN),EPINF(NDEIN)
* parametres externes
        REAL*8 PAREX0(NPAREX),PAREXF(NPAREX)
* fourre tout pour les comp non prevues et/ou munies d un autre constituant
        CHARACTER*16 typexo(nexo)
        CHARACTER*8 nomexo(nexo)
        CHARACTER*16 conexo(nexo)
        REAL*8 exova0(nexo),exova1(nexo)
      ENDSEGMENT
      segment wrk522
* temps initial, final
        integer  mkktp0, mkktpf
*
         integer  mkkal0(nsca), mkklaf(nsca),mkkalz
* deplacements
        integer  mkkpl0(ndep), mkkplf(ndep),mkkplz
* forces
         integer  mkkrc0(nfor), mkkrcf(nfor),mkkrcz
* gradients
         integer  mkkad0(ngra), mkkadf(ngra),mkkadz
* contraintes
          integer  mkkIG0(NSTRS), mkkIGF(NSTRS),mkkIGT(NSTRS),mkkigz
* deformations totales
         integer  mkkst0(NDEFO),mkkPST(NDEFO),mkkstf(NDEFO),mkkstz
* caracteristiques materiau
         integer  mkkat0(ncara),mkkMAT(NCARA),mkkatf(NCARA),
     &   mkkVAT(NCARA),mkkva0(ncara),mkkatz
```

```
c pointeur sur les melval de caracteristiques materiau a l etat final
        INTEGER mkklma(ncara)
* caracteristiques geometriques
        integer  mkkar0(ncarb),mkkARB(NCARB),mkkrbf(NCARB),mkkVCR(NCARB)
        integer mblcar,mkkarz
* temperatures
        integer  mkkre0(ntur), mkkref(ntur),mkkrez
* contraintes principales
        integer mkkin0(npri), mkkinf(npri),mkkinz
* matrices de hook
        integer  mkkho0(nmah), mkkhof(nmah),mkkhoz
* matrices hook tangent
        integer mkkta0(nhot), mkktaf(nhot),mkktaz
* variables internes
        integer  mkkVR0(NVARI),mkkVRF(NVARI),mkkvrz
* gradients flexion
        integer  mkkaf0(ngrf), mkkaff(ngrf),mkkafz
*
        integer  mkkas0(nrhi), mkkasf(nrhi),mkkasz
* deformations inelastiques
        integer mkkEFP(NDEIN),mkkPN0(NDEIN),mkkPNF(NDEIN),mkkpnz
* parametres externes
        integer  mkkEX0(NPAREX),mkkEXF(NPAREX),mkkexz
* fourre tout pour les comp non prevues et/ou munies d un autre constituant
        integer mkkvx0(nexo),mkkvx1(nexo),mkkvxz
      ENDSEGMENT
*
* quelques donnees pratiques
      SEGMENT WRK53
        INTEGER MATE,INPLAS,MELE,IPMAIL,NPINT,NBNN,NBELEM
        INTEGER NFOR,NMAT,NUMAT,NUCAR,NMATR,NCARF
        INTEGER MFR,NBG,NBGS,NSTRS,LRE,LW,LW2,LHOOK,LHOO2,IPORE,NBNO
        INTEGER NVART,NMATT,NCARR,JECHER,ISTEP,NBINT,JNOID
        INTEGER NBPTEL,NEL,N2PTEL,N2EL,KERRE
        INTEGER ITHHER
        CHARACTER*8  CMATE
        CHARACTER*16 CONM
        LOGICAL LOGSUC,LOGVIS,LUNI1,LUNI2,LOGRE,LOGIN
        REAL*8 PRECIS,SECT,COB,XMOB,EPAIST
        REAL*8 BID(6),BID2(6)
        REAL*8 COORGA(3),LCARAC
      ENDSEGMENT
* MATE : numero de materiau
```

```
* INPLAS : numero loi de comportement
* MELE : numero element fini
* IPMAIL : pointeur maillage
* NPINT : nombre points d integration coque
* NBNN : nombre de noeuds
* NBELEM : nombre d elements
* MFR : numero formulation
* NBPTEL : nombre de points d integrations
* NEL : nombre d elements
* KERRE : indicateur d erreur
* CMATE : description comportement
* CONM : constituant du modele
* PRECIS : precision
* SECT : section
* EPAIST : epaisseur
* COORGA(3) : coordonnees cartesiennes du point d'integration courant
* LCARAC : longueur caracteristique de l'element courant

* tableaux intermediaires
       SEGMENT WRK54
        REAL*8 DDHOOK(LHOOK,LHOOK)
        REAL*8 DDAUX(LHOOK,LHOOK)
        REAL*8 TXR(IDIM,IDIM),DDHOMU(LHOOK,LHOOK)
        REAL*8 XLOC(IDIM,IDIM),XGLOB(IDIM,IDIM)
        REAL*8 D1HOOK(LHOOK,LHOOK),ROTHOO(LHOOK,LHOOK)
        CHARACTER*16 TYMAT(NCXMAT)
        REAL*8 XMAT1(NCXMAT),XMAT2(NCXMAT)
      ENDSEGMENT
*
```

5.3 Développement

Il s'agit avec `COMP` d'être à même d'utiliser toutes les grandeurs figurant parmi les données transmises par l'opérateur, et de retrouver celles-ci dans des tableaux aux appellations conventionnelles au niveau des sous-programmes appelés par **coml7.eso et coml8.eso**. A ce titre la liste de mots `LISMOT` contenue dans `DECHE.INC` décrit les champs a priori recherchés et est exploitée systématiquement dans les sous-programmes

- **comou2.eso** , appelé par **comouw.eso**
- **cotype.eso**, appelé par **comcre.eso**, appelé par **comcri.eso** appelé par **comtri.eso**
- **comval.eso**
- **comsor.eso**

En particulier, les noms de champs spécifiés, pour les modèles VISCO_EXTERNE ou NON_LINEAIRE UTILISATEUR, sous l'étiquette `PARA_LOI` par l'utilisateur lors de la déclaration du modèle, correspondent à l'étiquette '`PARAMEXT`' de `LISMOT` et aux tableaux `PAREXO et PAREXF` de `wrk52`.

Les tableaux `typexo, nomexo, conexo, exava0 et exova1` sont renseignés avec les champs qui n'ont pas été identifiés au travers de la liste `LISMOT`.

L'objet temporaire `wrk52` est construit en distinguant état initial — terminaison 0 — et état final — terminaison f. En mécanique, les contraintes initiales sont dans `sig0`, l'incrément de déformation `depst` est la différence des déformations finales et initiales, les caractéristiques `xmat et xmatf` correspondent à l'état final. L'opérateur calcule les contraintes et les variables internes finales dans `sigf et varf`.

Quelques tableaux utilitaires sont inclus — par exemple `dsigt` pour les contraintes. Les tableaux peuvent paraître redondants : `valmat et xmat` pour les caractéristiques matériau. On notera que pour les caractéristiques on dispose des noms des composantes et de leurs types. Par ailleurs, des tableaux permettent d'exploiter les champs non-désignés dans la liste `LISMOT`.

L'objet temporaire `wrk522` est un préconditionnement pour renseigner `wrk52` dans **comval.eso**.

L'objet temporaire `wrk54` est un utilitaire permettant de calculer les opérateurs d'élasticité avec le sous-programme **calsig.eso**.

L'objet temporaire `wrk53` consigne des données communément employées.

Pour prendre en compte une nouvelle loi de comportement en mécanique il suffit généralement de modifier :

1. **coml8.eso** : il faut insérer le sous-programme propre à la loi, par exemple sous la forme

```
          IF (INPLAS.EQ.???)   THEN
            CALL CMALOI(wrk52,wrk53,wrk54,wrk27,IB,IGAU,
     &                     NBPGAU,ecou,necou,iecou)
          ENDIF
```

   `wrk27` représentant éventuellement un segment utilitaire propre à cette loi.

2. **cotype.eso** : quand ce n'est pas une loi VISCO_EXTERNE ou UTILISATEUR et quand la loi utilise des champs qui ne sont pas décrits localement par des réels, il est nécessaire de rajouter autant de séquences que nécessaire, en suivant les adresses induites par la liste `LISMOT`, suivant le modèle

```
IF (INPLAS.EQ.???)   THEN
   NBTYPE=YYYY
   SEGINI NOTYPE
   MOTYPE=NOTYPE
   .
   .
   .
   TYPE(K)=TYPEVAR (K=1,...,YYYY)
   .
   .
   .
 ENDIF
```

Signification des variables

`INPLAS` Le paramètre `INATU` de **NOMATE.ESO** (numéro de nature du matériau). est à confronter au numéro affecté au nouveau modèle dans ce sous-programme.

`YYYY` Nombre de composantes

`TYPEVAR` Chaine de caractères valant, soit `REAL*8` si la K^{eme} composante est un réel, soit par exemple `POINTEUREVOLUTIO` s'il s'agit d'une courbe. On n'écrit la séquence que si l'une des composantes est associée à un `POINTEURZZZZZZZZ`, pour la grandeur physique — caractéristiques, variables internes, ...— envisagée.
Remarque : Dans les instructions `TYPE(K)=TYPEVAR` concernant les caractéristiques on mettra d'abord les caractéristiques plastiques obligatoires, puis les caractéristiques facultatives.

Pour introduire des ensembles de lois faisant intervenir des grandeurs physiques non recensées dans la liste `LISMOT` — champs électromagnétiques, vecteurs flux, ...—, on peut envisager d'introduire les labels adéquats dans la liste `LISMOT`, en allongeant celle-ci ou encore en renommant les étiquettes '`RIGIDITE`', '`MASSE` ' et '`STRESSES`' de la liste qui sont inemployées. Par la suite on sera conduit à ajouter les tableaux nécessaires dans `wrk52` et `wrk522`, puis à modifier principalement

- **comou2.eso**
- **cotype.eso**
- **comcri.eso**
- **comval.eso**
- **comsor.eso**
- **coml7.eso**

On aura soin de recompiler tous les sous-programmes utilisant `DECHE.INC`.

Chapitre 6

Sous-programmes principaux de COML6.ESO

COMOUW.ESO Dimensionne les objets temporaires `wrk52 et wrk522` et détermine l'association entre les segments `DECHE` et les composantes des différents tableaux.

COMCRI.ESO Crée les segments `DECHE` destinés à recueillir les résultats.

COMTRI.ESO Vérifie le type des données.

COMDEF.ESO Renseigne les segments `wrk53, necou, iecou et xecou`.

COMROT.ESO Matériaux inélastiques orthotropes, anisotropes et unidirectionnels :
- en formulation massive : on cherche les coordonnées des noeuds de l'élément courant pour le calcul des axes locaux.
- cas particulier de l'ACIER_UNI

COMVAL.ESO Renseigne l'objet intermédiare `wrk52` à partir des segments `MELVAL`. Ce programme se réfère à la liste `LISMOT`. Si besoin est, on cherche la section de l'élément courant, on prend en compte l'épaisseur et l'excentrement dans le cas des coques minces avec ou sans cisaillement transverse.

COMARA.ESO Quelques calculs intermédiaires et réarrangement éventuel des données des tableaux `xmat0, xmat et tymat`, de sorte que les caractéristiques du matériau sont souvent placées dans l'ordre suivant :

– caractéristiques élastiques obligatoires :

$$E, \nu$$

– caractéristiques "élastiques" facultatives :

$$\alpha, \rho$$

– caractéristiques non linéaires obligatoires (dans l'ordre des déclarations dans les sous-programmes IDPLAS,IDFLUA,IDVISC,IDENDO,IDPLEN)
– caractéristiques non linéaires facultatives

COML7.ESO Gère l'appel des sous-programmes traitant les lois et relations ; est relayé par **coml8.eso** pour la mécanique.

DEFER1.ESO Traite les messages d'erreur en cas de problème de convergence

DEFER2.ESO Traite les autres messages d'erreur en général à partir de la variable `KERRE` de l'objet temporaire `wrk53` (0 si pas d'erreur). rappelons qu'il faut également modifier **gibi.erreur** lorsqu'un nouveau message est ajouté .

COMSOR.ESO On remplit les `MELVAL` concernés à partir de `wrk52`. Ce programme se réfère à la liste `LISMOT`.

COMPOU.ESO Réalise des moyennes pour les cas particuliers des poutres et tuyaux (sauf Timoshenko) et renseigne les `MELVAL`.

COMFIN.ESO Supprime les objets temporaires `wrk52 et wrk522` et quelques autres segments.

Signalons l'existence de quelques utilitaires pour la mécanique :
– **calsig.eso** permet de calculer un incrément de contraintes en supposant un comportement élastique, connaissant un incrément de déformation. Cet utilitaire, appelé directement dans **coml7.eso** ou bien par les sous-programmes propres aux diverses lois, a besoin d'un certain nombre d'informations qui se trouvent en partie dans les variables suivantes :

IMAT	entier	
ICAR	entier	
VALMAT	real*8	
VALCAR	real*8	
LHOOK	entier	
N2EL	entier	
N2PTEL	entier	
NBNO	entier	Ces quantités sont là pour etre passées à ”calsig.eso”
NBPGAU	entier	qui calcule l'incrément de contraintes en comportement
LW	entier	élastique .Pour la dimension des tableaux prendre exemple
NBGMAT	entier	dans le sous-programme ”prpvar.eso”
NELMAT	entier	
TXR	real*8	
XLOC	real*8	
D1HOOK	real*8	
ROTHOO	real*8	
DDHOMU	real*8	
CRIGI	real*8	
NUMAT	entier	
NUCAR	entier	
DSIGT	real*8	sortie de calsig, incrément des contraintes en élastique

- **ccoinc.eso**, appelé dans **coml7.eso** traite l'écoulement plastique par une méthode inspirée du “radial-return”.
- **cconst.eso**, appelé dans **coml7.eso** traite les lois de viscosité avec une méthode de Runge-Kutta d'ordre 4.

Chapitre 7

Conclusion

Nous avons évoqué les 3 phases d'implantation d'un nouveau modèle dans CAST3M : i) déclaration, ii) reconnaissance des champs impliqués, iii) calcul incrémental.

Pour introduire une loi d'évolution physique ou un modèle de comportement mécanique non-linéaire dans Castem2000, la connaissance approfondie d'ESOPE n'est pas utile. Si le développeur doit intervenir dans une petite dizaine de sous-programmes, il peut constater que dans la plupart des cas très peu de lignes fortran sont à ajouter.

Nous espérons que le lecteur aura bien pris conscience que s'il posséde déjà une boite fortran réalisant l'écoulement de son modèle, l'incorporation dans Cast3M est normalement l'affaire de un ou deux jours.

Dans les cas de modèles très compliqués (par exemple le modèle de MAXWELL dont le nombre de variables internes n'est pas connu par avance), une maîtrise plus grande du logiciel est souhaitable (contacter l'équipe de développement).

Nous éspérons que ce document s'enrichira et nous sommes prêts à le rendre plus explicite, en fonction de vos suggestions.

Chapitre 8

Bibliographie

– [1] FELIX S.
Implantation d'un nouvel élément fini dans le code CASTEM 2000.
Rapport DMT/93-654
– [2] DE GAYFFIER A.
Algorithme non linéaire de CASTEM 2000 (NONLIN et INCREME)
Rapport DMT/94-188
– [3] FELIX S.
Implantation d'un nouveau modèle de comportement dans le code de calcul CASTEM 2000
Rapport DMT/94-530
– [4] LE BOULCH D.
Implantation d'un nouveau modèle de comportement dans le code de calcul CASTEM 2000
Rapport DMT/97-215
– [5] CHARRAS T.
CASTEM2000. Implantation d'une nouvelle loi de comportement.
Rapport DMT/98-029
– [6] LAINET M.
PROJET PLEÏADES. Spécifications / Conception d'une évolution du code CAST3M pour la prise en compte d'un modèle de comportement mécanique non-linéaire défini à l'externe par l'utilisateur. Révisions pour compatibilité avec ABAQUS. Guide d'utilisation.
Rapport n CS/311-1/AJ02A009/RAP/02/07/15 Version 2.2
– [7] LAINET M.
PROJET PLEÏADES. Evolution du code CAST3M pour la prise en compte d'un modèle de comportement mécanique non-linéaire défini à l'externe par l'utilisateur.
Rapport de validation.
Rapport n CS/311-1/AJ02A009/RAP/03/01/09 Version 1.0

Ce document résulte de l'amélioration des références 3 à 7.

Chapitre 9

ANNEXE

Nous considérons par exemple l'implantation du modèle viscoplastique de KOCKS.

9.1 Phase 1 : Déclaration d'un modèle mécanique non-linéaire

1. Le premier travail est à faire dans **modvis.eso** où il faut déclarer le modèle en introduisant la ligne :

```
MOMODL(9)='KOCKS'
```

2. Il faut ensuite déclarer un numéro global de comportement non-linéaire dans **nomate.eso**. Cela se traduit par les lignes :

```
  ....
  ....
*   'KOCKS'                                     INATU = 70
  ....
  ....
  ....
                        ELSE IF (IMOD.EQ.9 ) THEN
*   KOCK
                          INATU = 70
```

9.2 Phase 2 : Déclaration des paramètres d'un modèle mécanique non-linéaire

1. Ensuite dans **idvari.eso** il faut remplir la variable `MATEPL` par la séquence suivant :

```
               ELSE IF (IPLAC.EQ.9 ) THEN
*    KOCK
                  MATEPL=70
                  INMAT=INMAT+1
```

   Puis prévoir l'appel à un des **idvarX.eso** (ici **idvar4.eso**) par la séquence suivante ou seule la partie `.OR.MATEPL.EQ.70` (en majuscule) a été ajoutée :

```
*
      else if (matepl.eq.25.or.matepl.eq.63.OR.MATEPL.EQ.70) then
         call idvar4(matepl,mfr,ifour,npint,ipcomp,nbrobl,nbrfac)
```

2. Il faut enfin déclarer le noms des variables internes dans **idvar4.eso**. Le retour des **idvarX.eso** est le pointeur `IPCOMP` qui pointe sur un segment de type `NOMID`. Dans notre exemple il fallait introduire dans **idvar4.eso** la séquence :

```
*--------------------------------------------------------------------
*      MODELE  ELASTIQUE VISCOPLASTIQUE  DE KOCKS
*
      ELSE IF ( MATEPL .EQ. 70 ) THEN
         IF ( MFR.EQ.1.AND.(IFOUR.NE.-2.AND.(IFOUR.NE.1))) THEN
           NBROBL = 2
           SEGINI NOMID
           LESOBL(1) = 'EPSE'
           LESOBL(2) = 'S   '
           IPCOMP=NOMID
           SEGDES NOMID
           RETURN
         ENDIF
*   OPTION INDISPONIBLE
      CALL ERREUR(19)
      RETURN
```

   Remarque : `IFOUR` *traduit le MODE de l'opérateur OPTIO :*

- *-3 : DEFORMATION PLANE GENERALISEE*
- *-2 : CONTRAINTES PLANES*
- *-1 : DEFORMATION PLANE*
- *0 : AXISYMETRIQUE*
- *1 : SERIE DE FOURIER*
- *2 : TRIDIMENSIONNEL*

et `MFR` *est le numéro de formulation élément fini voir* **nummfr.eso**. `MFR` *est égal à 1 pour les éléments massifs.*

3. Puis on doit déclarer dans **idvisc.eso** le nom des paramètres matériau de la loi de comportement de KOCKS. Ce qui est réalisé en introduisant la séquence suivante dans le sous-programme :

```
      ELSE IF (IPLAC.EQ.9) THEN

*     VISCOPLASTIQUE MODELE DE KOCK

         MLMOTS=MOOBL
         JGA=MOTS(/2)
         JGM=JGA+10
         SEGADJ MLMOTS
         MOOBL=MLMOTS
         MOTS(JGA+1)='A    '
         MOTS(JGA+2)='B    '
         MOTS(JGA+3)='M    '
         MOTS(JGA+4)='Q    '
         MOTS(JGA+5)='R    '
         MOTS(JGA+6)='H0   '
         MOTS(JGA+7)='AP   '
         MOTS(JGA+8)='SB   '
         MOTS(JGA+9)='N    '
         MOTS(JGA+10)='S0   '
         GOTO 9999
```

Dans cet exemple, il n'y a pas de paramètre facultatif, sinon on aurait procédé de la même façon à partir de `MLMOTS = MOFAC`.

L'instruction `MOTS(/2)` *demande la 2ème dimension du tableau* `MOTS` *(étant un tableau de chaînes de caractères, on considère le nombre de caractères par chaîne comme la première dimension et le nombre de chaînes comme la deuxième dimension.* `SEGADJ MLMOTS` *demande de redimensionner le segment avec les nouvelles dimensions courantes (* `JGN et JGM`*).*

9.3 Phase 3 : Préparation de l'appel à l'écoulement non-linéaire

1. Pas d'intervention dans **cotype.eso** : en effet les paramètres matériaux et les variables internes sont de type `REAL*8`, qui est le type par défaut.
2. La variable `LOGVIS` est initialisée à `.true.` dans **comdef.eso**

```
        if (  inplas .eq. 17 .or.
       2   (  inplas .ge. 19 .and. inplas .le. 25) .or.
       4      inplas .eq. 61 .or.
       4      inplas .eq. 63 .or.
       4      inplas .eq. 65 .or.
       1      inplas .eq. 53 .or.
       1      inplas .eq. 29 .or.
       2      inplas .eq. 44 .or.  inplas .eq. 45  .or.
-->    2      INPLAS .EQ. 70 .OR.  inplas .eq. 74  .or.
       9      inplas .eq. 76 .or.
       9      inplas .eq. 77 .or.  inplas .eq.  82     ) then
           logvis=.true.
        endif
```

La variable `IND` prend la valeur 1, le nombre de sous-pas et la précision des itérations interne sont initialisées

```
        IF ( INPLAS .EQ. 17 .OR. INPLAS .EQ. 19 .OR.
       2     INPLAS .EQ. 20 .OR. INPLAS .EQ. 61 .OR.
       3     INPLAS .EQ. 21 .OR. INPLAS .EQ. 65 .OR.
       4     INPLAS .EQ. 22 .OR. INPLAS .EQ. 23 .OR.
       5     INPLAS .EQ. 24 .OR. INPLAS .EQ. 70 .OR.
       6     INPLAS .EQ. 25 .OR. INPLAS .EQ. 29 .OR.
       7     INPLAS .EQ. 43 .OR. INPLAS .EQ. 44 .OR.
       8     INPLAS .EQ. 45 .OR. INPLAS .EQ. 53 .OR.
       9     INPLAS .EQ. 63 .OR. INPLAS .EQ. 66 .OR.
       a     INPLAS .EQ. 74 .OR. INPLAS .EQ. 76 .OR.
       b     INPLAS .EQ. 84 .OR. INPLAS .EQ. 85 .OR.
       c     INPLAS .EQ. 86 .OR.
       d     INPLAS .EQ. 77 .OR. INPLAS .EQ. 82 .OR.
       e     INPLAS .EQ. 106 .OR. INPLAS .EQ. 102 .OR.
       f     INPLAS .EQ. 107 .OR. INPLAS .EQ. 108) THEN
*
```

```
         IND = 1
*
       if (inplas.eq.44.or.inplas.eq.45) then
         MSOUPA = 2000
       else
         MSOUPA = 200
       endif
         PRECIS = 5.d-7
      ENDIF
```

3. Il est nécessaire d'intervenir dans **coml7.eso**. En effet, le modèle de KOCKS est intégré par une méthode classique de Runge-Kutta pour lequel un utilitaire est disponible : le sous-programme **cconst.eso**. Il faut donc appeler ce sous-programme dans la séquence suivante :

```
*----------------------------------------------------------------------
*     modeles de viscoplasticite integres par consti
*----------------------------------------------------------------------
      IF ( INPLAS .EQ. 17 .OR.
     2     (INPLAS .GE. 19 .AND. INPLAS .LE. 25) .OR.
     3      INPLAS .EQ. 61 .OR. INPLAS .EQ. 63  .OR.
     4      INPLAS .EQ. 53 .OR. INPLAS .EQ. 102 .OR.
     5      INPLAS .EQ. 44 .OR. INPLAS .EQ. 76  .OR.
     6      INPLAS .EQ. 45 .OR. INPLAS .EQ. 77  .OR.
     7      INPLAS .EQ. 84 .OR. INPLAS .EQ. 85  .OR.
     8      INPLAS .EQ. 86 .OR. INPLAS .EQ. 70  .OR.
     9      INPLAS .EQ. 107) THEN
*
 ...
*
       CALL CCONST(wrk52,wrk53,wrk54,WRK7,WRK8,WRK9,
     1        NVARI,NSSINC,INV,IFORB,TETA1,TETA2,FI1,FI2,
     4     TLIFE,NCcor,IB,IGAU,NBPGAU,KERREU1,iecou,xecou)

...
```

4. Dans cet exemple, l'écoulement profite de la structure de **cconst.eso** ; cependant pour le modèle de KOCKS il faut appeler dans **cconst.eso** un nouveau sous-programme **incre8.eso** qui réalise le calcul d'un incrément de déformation et de variables internes. le sous-programme **advan1.eso** a lui aussi été légèrement modifié.

9.4 Introduction d'un nouveau comportement élastique

Nous prendrons l'exemple du modèle de la SECTION de la poutre à fibre.

1. On introduit dans **modela.eso** la ligne :

```
        MOMODL(7)='SECTION'
```

2. Puis il est nécessaire de définir dans **idmatr.eso** le nom des caractéristiques obligatoires et facultatives du comportement linéaire. Ceci se fait par :

```
           ELSE IF (IPLAC.EQ.7) THEN
C
C      MODELE ET MATERIAU DE LA SECTION DU MODELE A FIBRE
C
              JGM=2
              SEGINI MLMOTS
              MOOBL=MLMOTS
              MOTS(1)='MODS'
              MOTS(2)='MATS'
              JGM=1
              SEGINI MLMOTS
              MOFAC=MLMOTS
              MOTS(1)='MAHO'
```

`SEGINI MLMOTS` *est un ordre ESOPE demandant de créer un segment de type* `MLMOTS`*, qui d'aprés sa définition dans le* `-INC SMLMOTS` *que voici :*

```
*=================================================================
*
*     OBJET DE TYPE 'LISTMOTS'. DEFINIT UNE LISTE DE MOTS.
*
      SEGMENT MLMOTS
       CHARACTER*(JGN) MOTS(JGM)
      ENDSEGMENT
      POINTEUR MLMOT1.MLMOTS,MLMOT2.MLMOTS,MLMOT3.MLMOTS
      POINTEUR MLMOT4.MLMOTS,MLMOT5.MLMOTS,MLMOT6.MLMOTS
*
*======================================================
```

contient un tableau de `JGM` *chaînes de* `JGN` *caractères chacune. Les valeurs des pointeurs sont sauvées dans* `MOOBL` *pour le pointeur des noms de caractéristiques obligatoires et dans* `MOFAC` *pour celui des noms de caractéristiques facultatives. Ce sont ces tableaux qui permettront de remplir le segment* `NOMID` *qui contient :*

```
SEGMENT NOMID
  CHARACTER*8 LESOBL(NBROBL),LESFAC(NBRFAC)
ENDSEGMENT
```