Table des matières

1 Travaux pratiques : comment écrire une loi de comportement

1.1 un premier exemple très simple : la loi de Norton

1.1.1 expression de la loi de NORTON

En guise de préliminaire, prenons par exemple la loi de NORTON. Elle est définie par :

$$\begin{cases} \underline{\epsilon}^{\text{to}} = \underline{\epsilon}^{\text{el}} + \underline{\epsilon}^{\text{vis}} \\ \underline{\sigma} = \underline{\underline{\mathbf{D}}} : \underline{\epsilon}^{\text{el}} \\ \dot{\underline{\epsilon}}^{\text{vis}} = \dot{p}\,\underline{n} \\ \dot{p} = A\,(\sigma_{\text{eq}})^m \end{cases}$$

où :
- $\underline{\epsilon}^{\text{to}}$, $\underline{\epsilon}^{\text{el}}$, $\underline{\epsilon}^{\text{vis}}$ représentent respectivement les déformations totale, élastique et visqueuse ;
- $\underline{n} = \dfrac{3}{2}\dfrac{\underline{s}}{\sigma_{\text{eq}}}$ est la direction d'écoulement ;
- $\underline{s}$ est le déviateur des contraintes ;
- $\sigma_{\text{eq}} = \sqrt{\dfrac{3}{2}\,\underline{s} : \underline{s}}$ est la norme de VON MISES.

L'opérateur d'élasticité $\underline{\underline{\mathbf{D}}}$ est classiquement calculé à partir du module d'YOUNG E et du coefficient de POISSON ν.

1.1.2 Discrétisation implicite de loi de NORTON

La discrétisation en temps implique de définir une suite d'instants de calcul $\{t_i\}_{1\leq i\leq I}$.

Pour utiliser un algorithme implicite, il suffit d'écrire toutes les quantités à l'instant t_i et de remplacer les dérivées en temps par leurs incréments sur l'intervalle $\Delta t = t_i - t_{i-1}$:

$$\begin{cases} \Delta\,\underline{\epsilon}^{\text{el}} - \Delta\,\underline{\epsilon}^{\text{to}} + \Delta\,p\,\underline{n} = 0 \\ \Delta\,p - \Delta\,t\,A\,\sigma_{\text{eq}}^m = 0 \end{cases}$$

avec :
- $\underline{\sigma} = \underline{\underline{\mathbf{D}}} : \underline{\epsilon}^{\text{el}}$;
- $\underline{n} = \dfrac{3}{2}\dfrac{\underline{s}\,(t_i)}{\sigma_{\text{eq}}\,(t_i)}$.

On obtient ainsi un système de 7 équations (6 équations — en 3D — relatives à la décomposition additive du tenseur des déformations, et une équation relative à l'écoulement visco-plastique). Les 7 inconnues sont les 6 composantes de $\Delta\underline{\epsilon}^{\text{el}}$ et Δp.

1.1.3 Première implantation

on pourra vérifier que l'intégration implicite de ce modèle avec mfront est opérationnelle, avec le fichier suivant :

```
@Parser Implicit;
@Behaviour Norton;
@Algorithm NewtonRaphson_NumericalJacobian;
@RequireStiffnessTensor;
@MaterialProperty real A;
@MaterialProperty real m;
@StateVariable real    p;
@ComputeStress{  sig = D*eel;}
@Integrator{
  real seq = sigmaeq(sig);
  Stensor n = Stensor(0.);
  if(seq > 1.e-15){
    n   = 1.5*deviator(sig)/seq;
  }
  feel = deel + dp*n-deto;
  fp   = dp - dt*A*pow(seq,m);
}
@TangentOperator{
  Stensor4 Je;
  getPartialJacobianInvert(Je);
  Dt = D*Je;
}
```

`mfront` gère la compilation de la loi. Après avoir positionné l'environnement, par exemple sur la machines Calibre7, en tapant dans un terminal :

```
$ . /home/aster/etc/codeaster/profile_mfront.sh
```

il suffit ensuite de lancer la commande :

```
$ mfront --obuild -interface=aster norton.mfront
```

Ceci génère deuxrépertoires : `src` et `include`. Le répertoire src contient en particulier une bibliothèque dynamique :

```
src/libAsterBehaviour.so
```

1.1.4 Premier test

L'outil `mtest` permet d'effectuer des simulations sur point matériel, permettant de calculer la réponse à des sollicitations en contraintes ou en déformations.

Le fichier de données `norton.mtest` se présente de la façon suivante :

```
@Behaviour<aster> './src/libAsterBehaviour.so' 'asternorton' ;
@MaterialProperty<constant> 'YoungModulus' 178600.0E6 ;
@MaterialProperty<constant> 'PoissonRatio' 0.3 ;
@MaterialProperty<constant> 'A' 8.e-67 ;
@MaterialProperty<constant> 'm' 8.2 ;
@ExternalStateVariable 'Temperature' 300. ;
@ImposedStress 'SXX' {0.:0.,30.:40.e6};
@ImposedStress 'SXY' {0.:0.,30.:30.e6};
@Times {0.,30. in 100};
```

— la ligne 1 définit la bibliothèque à utiliser et le nom du comportement ;
— les lignes 2 à 5 définissent les valeurs des propriétés matériau de la loi ;
— la ligne 6 déclare la valeur de la température (inutile dans le cas présent) ;

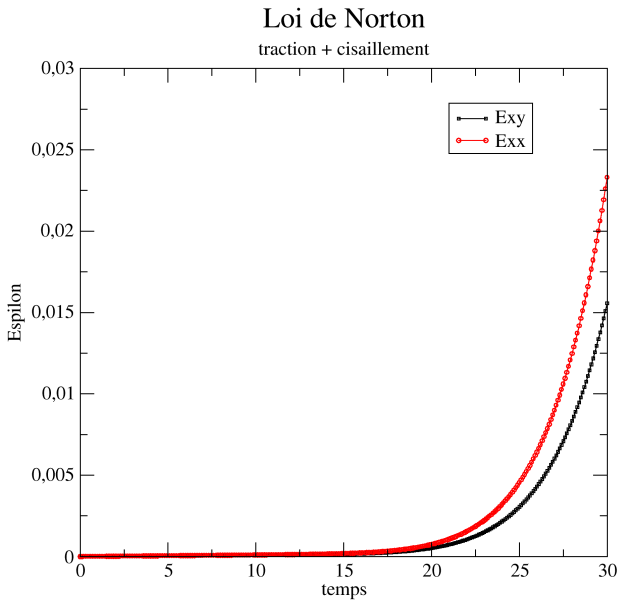

FIGURE 1 – Réponse d'une loi de NORTON à un essai de fluage en traction-cisaillement

— les lignes 7 à 8 spécifient le chargement, sous la forme de composantes de contraintes ou de déformations en fonction du temps ;
— la ligne 9 définit la discrétisation en temps, donc les instants calculés.

Il suffit alors de lancer le calcul par :

```
mtest norton.mtest
```

Ceci produit un fichier résultat `norton.res` contenant les composantes des tenseurs de déformation, de contrainte, et les variables internes en fonction du temps.

La réponse en déformation pour cet essai de traction est représentée en figure 1. Elle peut être tracée par exemple à l'aide de xmgrace.

On pourra également vérifier sur un essai de fluage (modifier le fichier mtest) que celui ci est de nature secondaire.

1.2 la loi visco-plastique de Lemaître

1.2.1 expression de la loi de comportement de LEMAÎTRE

Elle ne diffère de la loi de NORTON que par l'écrouissage. Elle est définie par :

$$\begin{cases} \underline{\epsilon}^{\text{to}} = \underline{\epsilon}^{\text{el}} + \underline{\epsilon}^{\text{vis}} \\ \underline{\sigma} = \underline{\underline{\mathbf{D}}} : \underline{\epsilon}^{\text{el}} \\ \underline{\dot{\epsilon}}^{\text{vis}} = \dot{p}\,\underline{n} \\ \dot{p} = \left(\dfrac{\sigma_{\text{eq}}}{K p^{1/M}}\right)^{N} \end{cases}$$

avec $\underline{n} = \dfrac{3}{2}\dfrac{\underline{s}}{\sigma_{\text{eq}}}$

on propose de modifier le fichier mfront de la loi de Norton pour y intégrer la loi de Lemaître.

Attention à bien écrire les équations sous forme implicite (ou bien avec la theta-méthode, en écrivant les quantités nécessaires à l'intégration sous la forme : a + theta*da) ;

On pourra vérifier sur l'essai de traction défini précédemment la différence par rapport à la loi de Norton (et l'équivalence des résultats dans le cas où UNsurM=0).

@ImposedStress 'SXX' 0. :0.,30. :40.e6 ;
@ImposedStress 'SXY' 0. :0.,30. :40.e6 ;

On pourra ensuite effectuer l'essai de traction précédent et constater la différence par rapport à la loi de Norton. On pourra prendre dans ce cas les paramètres suivants :

@MaterialProperty<constant> 'young' 178600.0E6 ;
@MaterialProperty<constant> 'nu' 0.3 ;
@MaterialProperty<constant> 'E' 8.2 ;
@MaterialProperty<constant> 'UNsurK' 8.6976357924534403e-09 ;
@MaterialProperty<constant> 'UNsurM' 0.2 ;
@ExternalStateVariable 'Temperature' 300.0 ;

1.3 Loi élasto-plastique de CHABOCHE

Nous allons développer pas à pas en `mfront` une loi élasto-plastique, puis visco-plastique (modèle deJ.L. CHABOCHE).

Il s'agit d'une loi de comportement élasto-plastique à écrouissage isotrope et cinématique non linéaires. Les équations du modèle sont résumées brièvement :

— relation contraintes déformations élastiques :

$$\underline{\sigma} = \underline{\underline{\mathbf{D}}} : \left(\underline{\epsilon}^{\mathrm{to}} - \underline{\epsilon}^{\mathrm{p}}\right)$$

— Critère de plasticité :

$$F\left(\underline{\sigma}, \underline{X}\right) = \left(\sigma - \underline{X}\right)_{\mathrm{eq}} - R(p) \leq 0$$

avec, pour tout tenseur A, $A_{\mathrm{eq}} = \sqrt{\frac{3}{2}\tilde{A} : \tilde{A}}$ où $\tilde{A}$ est le déviateur de A

— l'évolution de la déformation plastique est gouvernée par une loi d'écoulement normale au critère de plasticité :

$$\underline{\dot{\epsilon}}^{\mathrm{p}} = \dot{p}\,\underline{n} \quad \text{avec} \quad \underline{n} = \frac{3}{2}\frac{\underline{\tilde{\sigma}} - \underline{X}}{\left(\sigma - \underline{X}\right)_{\mathrm{eq}}}$$

— $\underline{X}$ représente l'écrouissage cinématique non linéaire. Il peut résulter d'une combinaison de plusieurs écrouissages cinématiques $\underline{X} = \underline{X}_1 + \underline{X}_2 + ...$;

— L'évolution de chaque variable d'écrouissage $\underline{X}_i$ est donnée par :

$$\underline{X}_i = \frac{2}{3} C_i \underline{\alpha}_i$$

$$\underline{\dot{\alpha}}_i = \underline{\dot{\epsilon}}^{\mathrm{p}} - \gamma_i\,\underline{\alpha}_i\,\dot{p}$$

— La fonction d'écrouissage isotrope $R\left(p\right)$ est définie par :

$$R\left(p\right) = R^{\infty} + \left(R^{0} - R^{\infty}\right)\exp\left(-b\,p\right)$$

Les propriétés matériau de ce modèle sont donc E, ν, R^0, R^∞, b, C_1, C_2, γ_1, γ_2.

La discrétisation implicite (par une θ-méthode) de ces équations conduit à résoudre un système d'équations où les inconnues sont :

— le tenseur $\Delta\,\underline{\epsilon}^{\mathrm{el}}$;

— le scalaire, $\Delta\,p$;

— les tenseurs $\Delta\,\underline{\alpha}_i$ (pour $i = 1, n$ variables cinématiques) ;

et où chaque équation d'évolution temporelle de la forme $\dot{y} = f\left(y, z, ...\right)$ est remplacée par :

$$\Delta y - \Delta\,t\,f\left(y + \theta\Delta y, z + \theta\Delta z, , ...\right) = 0$$

Dans le cas de la plasticité, on n'écrit pas d'équation d'évolution de la variable p, mais directement le respect du critère de plasticité.

Deux cas se présentent. L'évolution peut être élastique. Il est possible de tester cela en calculant un tenseur de test $\underline{\sigma}^{tr}$ tel que :

$$\underline{\sigma}^{tr} = \underline{D} : \left(\underline{\epsilon}^{\mathrm{el}}\big|_{t} + \Delta\,\underline{\epsilon}^{\mathrm{to}}\right)$$

Si le critère plastique évalué avec cette contrainte de test est négatif, c'est à dire si :

$$F^{el}\left(\underline{\sigma}, \underline{X}\right) = \left(\underline{\sigma}^{tr} - \underline{X}|_{t}\right)_{\mathrm{eq}} - R\left(p|_{t}\right) < 0$$

alors la solution cherchée est triviale et donnée par :

$$\Delta\,\underline{\epsilon}^{\mathrm{p}} = 0 \quad \Delta\,p = 0 \quad \Delta\,\underline{\alpha}_i = \underline{0}$$

Sinon, il faut résoudre le système suivant :

$$F(\sigma, X) = 0 \quad \Leftrightarrow \quad \begin{cases} (\underline{\sigma}|_{t+\Delta t} - \underline{X}|_{t+\Delta t})_{\text{eq}} - R(p(t+\Delta t)) & = 0 \\ \Delta\,\underline{\alpha}_i - \Delta p\ \underline{n}|_{t+\Delta t} + \gamma_i\,\Delta p\,(\underline{\alpha}_i + \theta\Delta\underline{\alpha}_i) & = \underline{0} \\ \Delta\,\underline{\epsilon}^{\text{el}} - \Delta\,\underline{\epsilon}^{\text{to}} + \Delta p\ \underline{n}|_{t+\Delta t} & = \underline{0} \end{cases}$$

où $\Delta\,\underline{\epsilon}^{\text{p}} = \dfrac{3}{2}\Delta p\ \underline{n}|_{t+\Delta t}$.

A faire : On pourra modifier le fichier norton.mfront afin d'intégrer cette loi de comportement. On peut se limiter à une seule variable cinématique dans un premier temps.

Dans le cas d'une seule variable cinématique, le système est de taille $6 + 1 + 1 \times 6 = 13$).

Dans le cas de 2 variables cinématiques, (le système est de taille $6 + 1 + 2 \times 6 = 19$).

Avec cet exemple, on, peut déjà tester le développement sur un point matériel à l'aide de `mtest`.

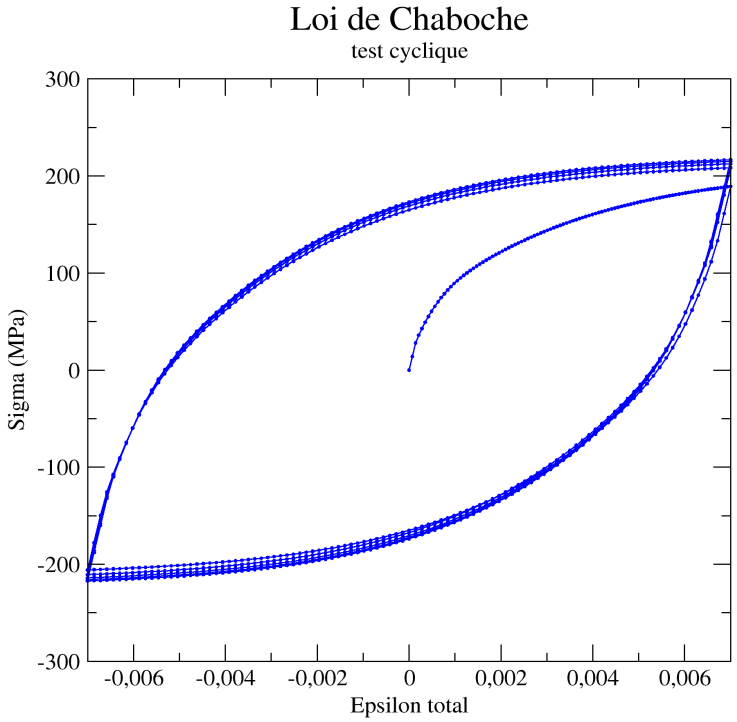

FIGURE 2 – Vérification de l'implantation de la loi élasto-plastique.

Pour une sollicitation uniaxiale de 10 cycles en traction-compression, on obtient le résultat donné en figure 2. Le fichier `mtest` pour cet exemple est le suivant :

```
@Behaviour<aster> 'src/libAsterBehaviour.so' 'asterchaboche';
@MaterialProperty<constant> 'young' 200000. ;
@MaterialProperty<constant> 'nu' 0.33 ;
@MaterialProperty<constant> 'R_inf' 50. ;
@MaterialProperty<constant> 'R_0' 30. ;
@MaterialProperty<constant> 'b' 20. ;
@MaterialProperty<constant> 'C[0]' 187000.;
@MaterialProperty<constant> 'C[1]' 45000.;
@MaterialProperty<constant> 'g[0]' 4460. ;
@MaterialProperty<constant> 'g[1]' 340. ;
```

```
@ExternalStateVariable 'Temperature' 0.;

@ImposedStrain 'EYY' {0.: 0., 1.: 0.007, 2.: -0.007, 3.: 0.007, 4.: -0.007, 5.:
    0.007, 6.: -0.007, 7.: 0.007, 8.: -0.007, 9.: 0.007, 10.: -0.007,11.:
    0.007, 12.: -0.007, 13.: 0.007, 14.: -0.007, 15.: 0.007, 16.: -0.007, 17.:
    0.007, 18.: -0.007, 19.: 0.007, 20.: -0.007, 21.: 0.007, 22.: -0.007};

@Times {0.,10. in 1000};
```

1.4 Utilisation dans Code_Aster

Les premiers tests sur un point matériel à l'aide `mtest`étant effectués, on peut s'intéresser au calcul par éléments finis avec `Code_Aster`. Pour cela, il suffit de fournir comme donnée de l'étude le fichier `libAsterBehabviour.so` (fichier de type "nom" dans `astk`).

Dans le fichier de commandes, il suffit de modifier deux points :

— Dans la commande `DEFI_MATERIAU`, il faut ajouter :
`.......=DEFI_MATERIAU( UMAT=_F( C1 = ... , C2 = ... , C3 = ... , ),`
On fournit les propriétés matériau `Ci` dans l'ordre où elles sont définies dans le fichier `mfront`.
Le mot clé `NB_VALE` permet d'optimiser le temps de lecture des propriétés.

— Le deuxième endroit à modifier est le mot-clé `COMPORTEMENT` dans les commandes globales de calcul (`STAT_NON_LINE`, `DYNA_NON_LINE`, `SIMU_POINT_MAT`, ...),
Le comportement a pour nom « MFRONT ». On spécifie donc dans ces commandes :

1.5 Loi viscoplastique de CHABOCHE

Le modèle diffère peu du précédent, car seul le critère de plasticité : $F(\sigma, X) = (\sigma - X)_{\text{eq}} - R(p) \leq 0$ est remplacé par une loi d'écoulement en puissance :

$$\dot{p} = \langle \frac{F}{K} \rangle^m$$

où $\langle F \rangle$ désigne la partie positive de F définie ci-dessus, soit :

$$\langle F \rangle = \max(0, F)$$

Dans le fichier mfront, seule l'équation donnant l'évolution de Δp change :

```
fp     = (seq_-Rp_)/young;
```

devient :

```
vp  = pow(F*UNsurK,m) ;
fp -= vp*dt;
```

`UNsurK` et `m` ont bien sûr été ajoutés dans la définition des propriétés matériau de la loi :

```
@MaterialProperty real m;
@MaterialProperty real UNsurK;
```

on pourra modifier le fichier `mfront`précédent pour définir ce modèle visco-plastique. On peut ensuite effectuer un test de fluage et des tests de traction à différentes vitesses à l'aide de `mtest` ou de `Code_Aster`.

1.6 Traitement des erreurs, déverminage

Et si, au cours du développement, on rencontre des erreurs, que faire ?

Lors de la compilation avec `mfront`, les erreurs sont la plupart du temps explicites :

— `Viscochab.mfront:94: error: 'F' was not declared in this scope`
— `Viscochab.mfront:74: warning: unused variable 'Rp_'`
— `Viscochab.mfront:91: error: expected ',' or ';' before 'if'` : oubli d'un ' ; ' en fin de ligne
— etc...

Lors de l'exécution, il est possible de localiser les erreurs numériques ou autres via les options de compilation :

— le plus simple est de compiler avec `-debug` (détails sur l'intégration) :
  `mfront --obuild -interface=aster --debug chaboche.mfront`
— un 'classique' du développement, les impressions locales :
  — pour imprimer une variable locale, il suffit d'écrire :
    `cout << " seq calculé " << seq << endl;`
  — pour afficher l'état courant de l'intégration :
    `cout << *this << endl;`
— pour trouver l'endroit où une erreur d'exécution se produit, on peut compiler en mode debug :
  `CXXFLAGS='-g'` `mfront --obuild --interface=aster chaboche.mfront`
— enfin, dans un calcul `Code_Aster`, on peut générer des fichiers mtest en cas d'échec d'intégration :
  `mfront --obuild -interface=aster --@AsterGenerateMTestFileOnFailure=true chaboche.mfront`