

---

## Notice d'utilisation de MFront avec Code\_Aster

---

### Résumé :

On décrit ici l'utilisation du couplage entre *Code\_Aster* et le module d'intégration de lois de comportements MFront, qui est diffusé en Open Source sous licence GPL et CECILL-A (<http://tfel.sourceforge.net>).

MFront permet de définir une loi de comportement de façon simple, proche des équations physiques, sans avoir à se soucier des méthodes de résolution. Il offre des classes de manipulation de données (tenseurs, par exemple) de façon conviviale et met à disposition des algorithmes de résolution des équations non linéaires associées efficaces en termes de performances. MFront est adapté aussi bien aux schémas temporels implicites qu'explicites.

Ces comportements sont utilisables dans *Code\_Aster* pour les modélisations 3D et 2D, coques, THM et pour les éléments de joints. Les grandes déformations peuvent être utilisées avec `GDEF_LOG`. De plus, MFront produit un code efficace, car les opérations tensorielles sont optimisées comme le montrent les benchmarks réalisés avec des comportements équivalents de *Code\_Aster*.

Il faut bien noter que l'utilisation de ces lois de comportement « à façon » (mot-clé `RELATION='MFRONT'`) implique une validation spécifique pour l'étude envisagée, car on se place hors du domaine qualifié de *Code\_Aster*.

Ce document décrit la façon d'utiliser un comportement défini par l'utilisateur à l'aide de MFront dans un fichier de commandes de *Code\_Aster*.

## 1 Définition d'une loi de comportement avec MFront

### 1.1 Écriture d'un comportement MFront

On écrit la loi de comportement dans le fichier MFront, suivant les conseils fournis dans le tutoriel5, ou en s'inspirant d'un comportement déjà disponible dans les tests de *Code\_Aster*.

Nous prendrons ici pour exemple le comportement visco-plastique de Lemaître écrit sous forme implicite. Le fichier « *Lemaitre.mfront* » s'écrit :

```
@Parser Implicit;
@Behaviour Lemaitre;
@Algorithm NewtonRaphson_NumericalJacobian;
@Theta 0.5 ;

@RequireStiffnessTensor;
@MaterialProperty real m;
@MaterialProperty real UNsurK;
@MaterialProperty real UNsurM;

@StateVariable real    p;

@ComputeStress{  sig = D*eel;}

@Integrator{
  const real seq = sigmaeq(sig);
  const real p_  = max((p+theta*dt),1.e-8) ;
  if(seq > 0.){
    real slen = seq*UNsurK/pow(p_,UNsurM);
    real vp   = pow(slen,m) ;
    Stensor n = 1.5*deviator(sig)/seq;
    feel += vp*dt*n;
    fp   -= vp*dt;
  }
  feel -= deto;
}

@TangentOperator{
  Stensor4 Je;
  getPartialJacobianInvert(Je);
  Dt = D*Je;
}
```

Cette loi est équivalente à la loi LEMAITRE, disponible dans *Code\_Aster* [R5.03.08].

### 1.2 Création de la bibliothèque dynamique à partir d'un comportement MFront

La bibliothèque dynamique contenant le comportement MFront doit être préparée avant l'exécution du calcul. Pour cela, il faut « compiler » ce comportement. Cela peut être fait de deux façons :

1) dans un terminal :

- « sourcer » l'environnement, cf. <http://tfel.sourceforge.net/documentations.html> :  
→ Sur Calibre 7

```
« source
 /mnt/.tgvd/projets/2/simumeca.033/public/V7_5_1_201502/salome_p
 rerequisites.sh »
```

→ Ou encore (pour d'autres installations) :

```
→ export TFELHOME="${PREREQUISITES_ROOT_DIR}/Mfront_TFEL201"
→ export PATH="${PREREQUISITES_ROOT_DIR}/Mfront_TFEL201/bin":"$PATH"
```

• puis « compiler » :

```
→ mfront --obuild -interface=aster Lemaitre.mfront
```

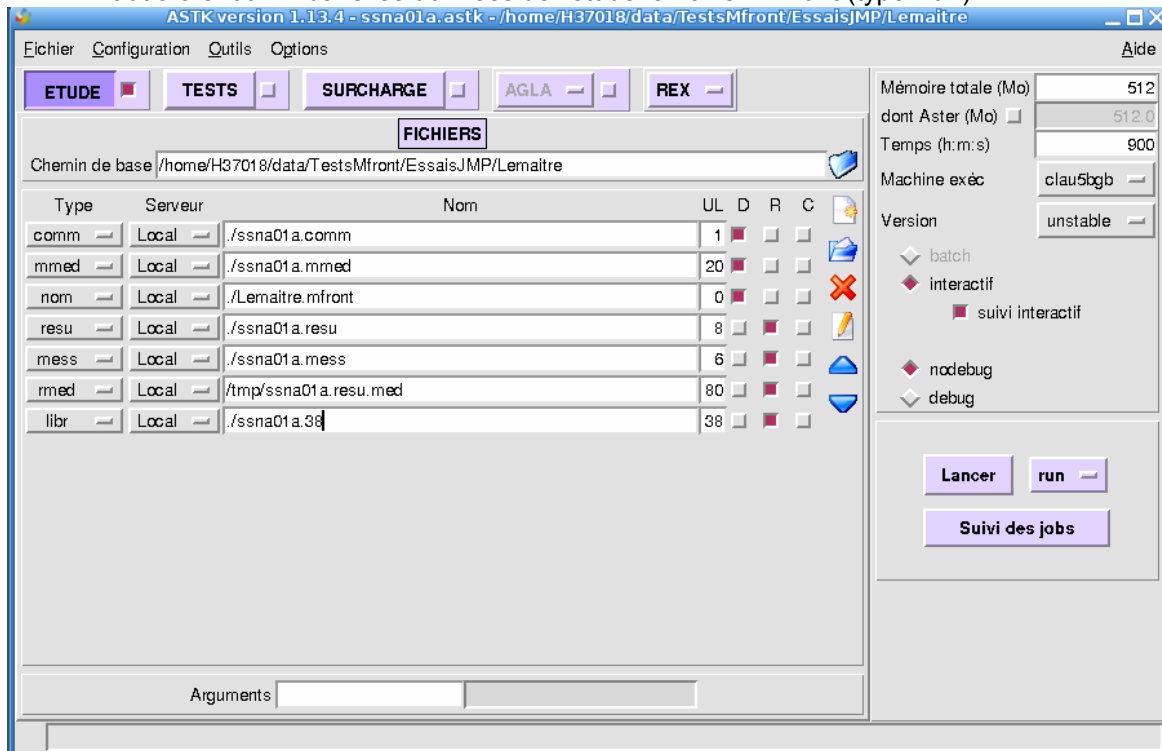
Cela crée deux répertoires, include et src. Ce dernier contient en particulier la bibliothèque dynamique libAsterBehaviour.so qui sera fournie dans les données de l'étude.

• Cette compilation peut être effectuée directement dans le fichier de commandes :

• Insérer pour cela l'instruction suivante en tête de fichier :

```
os.system("mfront --obuild Lemaitre.mfront -interface=aster")
```

Il faut alors fournir dans les données de l'étude le fichier MFront (type nom)



## 1.3 Utilisation du comportement dans le fichier de commandes

L'utilisation du comportement MFront dans le fichier de commandes s'effectue de la façon suivante :

1. Les données matériaux sont fournies dans DEF1\_MATERIAU [U4.43.01], sous UMAT / UMAT\_FO.

Par exemple pour la loi de Lemaitre :

```
young=210000.
poisson=0.3
m=11.
un_sur_k = 0.00032840723
un_sur_m = 0.178571429
```

```
MATFRONT=DEFI_MATERIAU(  
    ELAS=_F(E=young, NU=poisson),  
    UMAT=_F( LISTE_COEF = (young , poisson ,  
                        m, un_sur_k, un_sur_m)),  
    ),  
    )
```

Les valeurs de LISTE\_COEF (young, poisson, ...) sont à donner dans l'ordre des @MaterialProperties du fichier MFront.

2. Sous le mot-clé **COMPOTEMENT** des commandes de calcul non-linéaire **STAT\_NON\_LINE**, **DYNA\_NON\_LINE**, ou **SIMU\_POINT\_MAT** :

```
MFi=STAT_NON_LINE(MODELE=MO,  
    CHAM_MATER=CMF,  
    INCREMENT=_F(LIST_INST = L_INST, NUME_INST_FIN = 5),  
    NEWTON=_F(REAC_ITER=1),  
    EXCIT=( _F(CHARGE = CH1),  
           _F(CHARGE = CH2)),  
    COMPOTEMENT=_F(RELATION='MFRONT',  
                   LIBRAIRIE='./src/libAsterBehaviour.so',  
                   NOM_ROUTINE='asterlemaitre',),  
    )
```

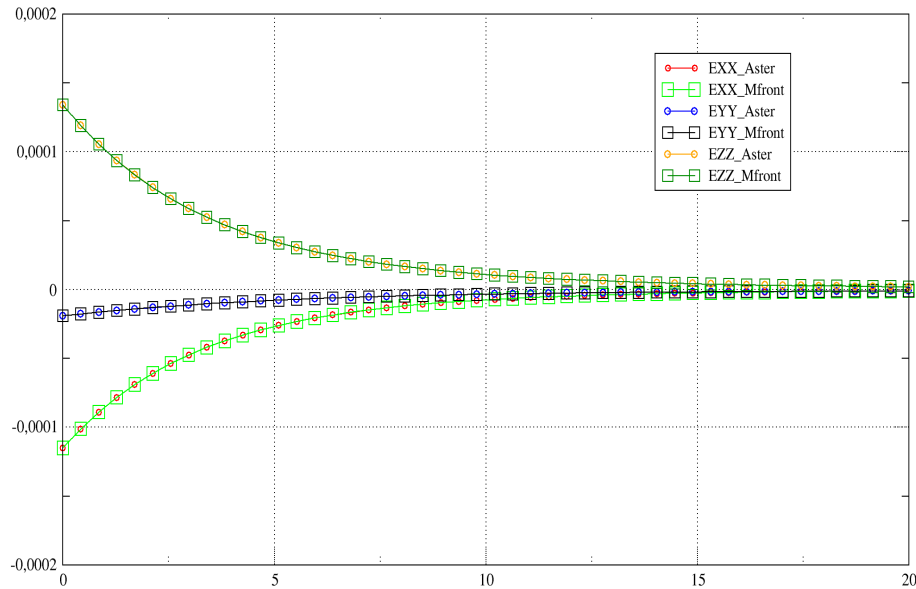
**NOM\_ROUTINE='asterlemaitre'** permet de spécifier la loi de comportement choisie, ce qui est utile si plusieurs lois ont été compilées ensemble. Ce nom est construit par concaténation de « aster » et du nom défini par @Behaviour dans le fichier MFront.

Il est possible d'ajouter sous **COMPOTEMENT**, dans le cas où **RELATION='MFRONT'** les mots-clés suivants (cf. [U4.51.11]) :

- **ITER\_INTE\_MAXI** (100 par défaut)
- **RESI\_INTE\_MAXI** (1.E-8 par défaut)
- **ITER\_INTE\_PAS** (0 par défaut)
- **DEFORMATION** (**PETIT par défaut**) on peut utiliser **GDEF\_LOG** pour tous les comportements écrits en petites déformations et **SIMO\_MIEHE** pour certains comportements écrits spécifiquement avec cette formulation.

Le reste du fichier de commandes est inchangé. Par exemple, si on utilise la loi de Lemaitre définie dans Lemaitre.mfront dans le test ssna01a, on peut comparer les résultats avec ceux obtenus par la loi de **LEMAITRE** de *Code\_Aster*.

test SSNA01A, loi de Lemaitre.  $\text{epsp}=f(x)$ ,  $t=60$ .



## 2 Exemples d'utilisation

D'autres exemples sont fournis dans le tutoriel disponible sur le site internet de MFront : <http://tfel.sourceforge.net/documentations.html>.

Le test `mfron02c` permet de vérifier en gestion de configuration le mot clé `RELATION='MFRONT'`, avec un fichier MFront fourni en donnée du test.

## 3 Références

- [1] T. Helfer, J.-M. Proix. *Écriture de lois de comportement avec MFront : tutoriel*. Décembre 2014. <http://tfel.sourceforge.net/documents/tutoriel/tutoriel.pdf>