

**TFEL/MFront 2.0.4 : note de version**  
**T. Helfer**

**RÉSUMÉ**

## S O M M A I R E

<b>1 DESCRIPTION DES ÉVOLUTIONS .....</b>	<b>3</b>
1.1 CORRECTIONS D'ANOMALIES .....	3
1.1.1 <i>Opérateur tangent cohérent mal calculé par MTest en grandes transformations pour les lois compilées avec l'interface aster (Ticket #12)</i> .....	3
1.1.2 <i>Convention de rangement des tenseurs non symétriques (Ticket #13)</i> .....	3
1.1.3 <i>Définition des macros F77_FUNC et F77_FUNC_ avec cmake (Ticket #15)</i> .....	3
1.1.4 <i>Correction du calcul de la matrice d'élasticité « altérée »pour un matériau orthotrope en contraintes planes (Ticket #18)</i> .....	4
1.2 NOUVELLES FONCTIONNALITÉS .....	4
1.2.1 <i>Définition du repère d'orthotropie à l'aide des angles d'Euler ou des indices de Miller dans MTest</i> .....	4
1.2.2 <i>Ticket # 30</i> .....	4
1.2.3 <i>Ticket # 31</i> .....	4
1.2.4 <i>Ticket # 32</i> .....	5
<b>2 CONCLUSIONS .....</b>	<b>5</b>
2.1 REMERCIEMENTS .....	5
<b>RÉFÉRENCES .....</b>	<b>5</b>

# 1 DESCRIPTION DES ÉVOLUTIONS

## 1.1 CORRECTIONS D'ANOMALIES

Les numéros de ticket correspondent au forum public de `tfel` consultable à l'adresse suivante :

<http://sourceforge.net/p/tfel/tickets>

### 1.1.1 Opérateur tangent cohérent mal calculé par MTest en grandes transformations pour les lois compilées avec l'interface aster (Ticket #12)

L'opérateur tangent cohérent était mal calculé par `MTest` en grandes transformations pour les lois compilées avec l'interface `aster`.

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/12/>

### 1.1.2 Convention de rangement des tenseurs non symétriques (Ticket #13)

La convention de rangement des tenseurs non symétriques était détaillée dans une annexe de la documentation `MFront` sur les lois en grandes transformations et était donc peu accessible. Ce point a été corrigé et cette convention est maintenant également décrite dans la documentation du mot clé `@ImposedDeformation-Gradient` de `MTest`.

### 1.1.3 Définition des macros F77\_FUNC et F77\_FUNC\_ avec cmake (Ticket #15)

La version 2.0.3 ne compilait plus avec les versions récentes de `CMake`. Le problème est lié à la définition de macros `F77_FUNC` et `F77_FUNC_` qui permettent de trouver la correspondance entre le nom d'une fonction en `C` et son nom en `fortran`.

Plus précisément, le problème est apparu suite à une correction de bug dans les versions de `CMake` supérieures à 3.1 qui conduit à traiter correctement le caractère dièse ('#'). Hors, pour contourner ce bug, nous utilisions un contournement : nous préfixions le caractère dièse par le caractère '\'. C'est ce contournement qui conduit à des erreurs dans les versions récentes de `CMake`.

Afin de supporter toutes les versions de `CMake`, le problème a été résolu ainsi :

```
if (CMAKE_VERSION)
    if (CMAKE_VERSION VERSION_GREATER 3.1)
        # Nouvelle definition de la macro F77_FUNC
    else (CMAKE_VERSION VERSION_GREATER 3.1)
        # Ancienne definition de la macro F77_FUNC
    endif (CMAKE_VERSION VERSION_GREATER 3.1)
else (CMAKE_VERSION)
    # Ancienne definition de la macro F77_FUNC
endif (CMAKE_VERSION)
```

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/15/>

#### **1.1.4 Correction du calcul de la matrice d'élasticité « altérée »pour un matériau orthotrope en contraintes planes (Ticket #18)**

Le terme de cisaillement  $D_{xyxy}$  dans le calcul de la matrice d'élasticité « altérée »pour un matériau orthotrope en contraintes planes était mal calculé.

Pour cette hypothèse de modélisation, `MFront` peut fournir deux types de matrices d'élasticité à la loi de comportement : la matrice standard et la matrice altérée qui prend en compte l'hypothèse de contraintes planes. Seule la première est utilisée en pratique, ce qui limite la portée de cette anomalie dont la correction ne devrait pas avoir d'impact sur les utilisateurs.

Deux cas test supplémentaires, pour les codes `Cast3M` et `Code_Aster`, ont été rajoutées, pour tester spécifiquement cette fonctionnalité. Le code `Zebulon` n'est pas affecté.

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/18/>

## **1.2 NOUVELLES FONCTIONNALITÉS**

### **1.2.1 Définition du repère d'orthotropie à l'aide des angles d'Euler ou des indices de Miller dans MTest**

Il est maintenant possible de définir le repère d'orthotropie à l'aide des angles d'Euler ou des indices de Miller dans `MTest`. Pour cela, le mot clé `RotationMatrix` peut prendre une option (précisée entre les caractères < et >) dont la valeur peut être :

- `Standard` (la valeur par défaut). Dans ce cas, l'utilisateur doit préciser la matrice de rotation.
- `Euler`. Dans ce cas, l'utilisateur doit préciser les 3 angles d'`EULER`.
- `Miller`. Dans ce cas, l'utilisateur doit préciser les 3 indices de `MILLER`. Ce dernier cas est intéressant pour le test de lois mono-cristallines.

### **1.2.2 Ticket # 30**

La classe `CxxTokenizer` contenait une erreur dans la méthode `stripComments` si le premier élément du fichier était un commentaire.

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/30/>

### **1.2.3 Ticket # 31**

Par cohérence avec le jeu de données `MTest`, les méthodes suivantes ont été ajoutées à la classe `python` éponyme :

- `setStrainEpsilon`
- `setDeformationGradientEpsilon`
- `setOpeningDisplacementEpsilon`
- `setStressEpsilon`
- `setCohesiveForceEpsilon`

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/31/>

#### **1.2.4 Ticket # 32**

L'interface `python de MTest` ne permettait pas de fixer les valeurs initiales des inconnues (déformation, gradient de la déformation ou saut de déplacement suivant les cas) et des forces (contrainte de Cauchy, force cohésive). Les méthodes sont `setDrivingVariablesInitialValues` et `setThermodynamicForcesInitialValues` ont été ajoutées pour corriger ce point.

Par cohérence avec le jeu de données `MTest`, les méthodes suivantes ont également été ajoutées à la classe `python éponyme` :

- `setStrain`
- `setDeformationGradient`
- `setOpeningDisplacement`
- `setStress`
- `setCohesiveForce`

Plus de détails peuvent être trouvés à la page :

<https://sourceforge.net/p/tfel/tickets/32/>

`@RotationMatrix<Miller> {1,5,9};`

## **2 CONCLUSIONS**

### **2.1 REMERCIEMENTS**

La page suivante donne les noms des principaux contributeurs à TFEL :

<http://tfel.sourceforge.net/about.html>

## **RÉFÉRENCES**