Note of use of MFront with code_aster

Summary:

One describes the use of the coupling here enters **Code_hasster** and integration of laws of MFront behavior modulates it, which is diffused in Open Source under licence LPG and CECILL-A (http://tfel.sourceforge.net).

MFront makes it possible to define a law of simple behavior of way, near to the physical equations, without having to worry about the methods of resolution. It offer of the classes of handling of data (tensors, for example) in a convivial way and places at the disposal of the algorithms of resolution of the nonlinear equations effective partners in terms of performances. MFront is adapted as well to the implicit temporal diagrams as explicit.

These behaviors are usable in **Code_hasster** for modelings 3D and 2D, hulls, THM and for the elements of joints. The great deformations can be used with `GDEF_LOG`. Moreover, MFront produces an effective code, because the tensorial operations are optimized as show it them *benchmarks* realized with equivalent behaviors of **Code_hasster**.

It should well be noted that the use of these laws of behavior “made-to-order” (keyword `RELATION=' MFRONT'` under `BEHAVIOR`) imply a specific validation for the study considered, because one places oneself out of field described as **Code_hasster**.

This document describes the way of using a behavior defined by the user using MFront in a command file of **Code_hasster**.

1 Definition of a law of behavior with MFront

1.1 Writing of a MFront behavior

One writes the law of behavior in the MFront file, according to the advices provided in the tutorial[1], or while taking as a starting point a behaviour already available in the tests by **Code_hasster**.

We will take here as an example the viscoplastic behavior of Lemaître written in implicit form. The file " `Lemaitre.mfront` " is written:

```
@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*dp), 1.e-8);
  yew (seq > 0.) {
     real slem  = seq*UNsurK/pow (p_, UNsurM);
     real vp  = pow (slem, m);
     Stensor N = 1.5*deviator (sig) /seq;
     feel += vp*dt*n;
     FP   - = vp*dt;
  }
  feel - = deto;
}

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

This law is equivalent to the law `LEMAITRE`, available in **Code_hasster** [R5.03.08].

1.2 Creation of the dynamic library starting from a MFront behavior

To be usable in **Code_hasster**, the MFront behavior thus programmed must be transformed into dynamic library. For that, it is necessary "to compile". That can be made in two ways:

1) **Solution recommended : use of `CREA_LIB_MFRONT`**

Lmacro-order has `CREA_LIB_MFRONT` allows to build the library automatically. It is enough to provide the MFront file as starter on a logical unit.

```
CREA_LIB_MFRONT ( UNITE_MFRONT=10,
                  UNITE_LIBRAIRIE=11, )
```

This operating process is to be privileged, in particular because it facilitates the use in AsterStudy. For more information, to see [U7.03.04].

2) **For the experts: construction of the library to the hand**

In a terminal, "to compile" the MFront file:

```
will mfront --obuild – interface=aster Lemaitre.mfront
```

That creates two repertoires, `include` and `src`. This last contains in particular the dynamic library `libAsterBehaviour.so` who will be provided in the data of the study.

Note: Compilation in mode DEBUG

SI the user wishes to debug its law, it must build it this way:

```
CXXFLAGS=" - G 'tfel-config --oflags --compile-flags '" will mfront --obuild --interface=aster Lemaitre.mfront
```

1.3 Use of the behavior in the command file

The use of the MFront behavior in the command file is carried out in the following way:

1. The data materials are provided in `DEFI_MATERIAU` [U4.43.01], under `MFRONT / MFRONT_FO`.
   For example for the law of Lemaître:

```
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),
                      MFRONT=_F (LISTE_COEF = (Young, fish,
                                    m, un_sur_k, un_sur_m,)),
                       ),
                              )
```

   Values of `LISTE_COEF` (`Young, fish`,...) are to be given in the order of `@MaterialProperties` MFront file.

2. Under the keyword `BEHAVIOR` orders of non-linear calculation `STAT_NON_LINE`, `DYNA_NON_LINE`, or `SIMU_POINT_MAT` :

```
MFi=STAT_NON_LINE (MODELE=MO,
                  CHAM_MATER=CMF,
                  INCREMENT=_F (LIST_INST = L_INST, NUMÉRIQUE_INST_FIN
= 5),
                  NEWTON=_F (REAC_ITER=1),
                  EXCIT= (_F (LOAD = CH1),
                         _F (LOAD = CH2)),
                  COMPORTEMENT=_F (RELATION=' MFRONT',
                        UNITE_LIBRAIRIE=11,
                        NOM_ROUTINE=' asterlemaitre',),
              )
```

   UNITE_LIBRAIRIE=11 must be coherent with the value used in `CREA_LIB_MFRONT` to build the library.

NOM_ROUTINE=' asterlemaitre ' allows to specify the law of behavior chosen, which is useful if several laws were compiled together. This name is built by concatenation of " aster " and of the name defined by @Behaviour in the MFront file.
It is possible to add under BEHAVIOR, if RELATION= 'MFRONT' following keywords (*cf.* [U4.51.11]):

- ITER_INTE_MAXI (100 by defaults)
- RESI_INTE_MAXI (1.E-8 by default)
- ITER_INTE_PAS (0 by default)
- DEFORMATION (SMALL by default) one can use GDEF_LOG for all the behaviors written in small deformations and SIMO_MIEHE for certain behaviors written specifically with this formulation.

The rest of the command file is unchanged. For example, if one uses the law of Lemaître defined in Lemaitre.mfront in the test ssna01a, one can compare the results with those obtained by the law of LEMAITRE of **Code_hasster**.

Note:

If one compiled the MFront library with the hand, it to replace the keyword UNITE_LIBRAIRIE by BOOKSTORE and to provide the name of the file in the repertoire of execution (cf [U4.51.11]).

1.4 Modelings available

Laws of behavior MFront is usable natively in 3D (and its derived like 3D_SI and 3D_INCO_*), in plane deformations (D_PLAN_*) and into axisymmetric (AXIS_*). For the case of CoNTraintes plane (C_PLAN), there are two situations:

- either one uses the mode forced plane native of MFrhave in selecting ALGO_CPLAN=' ANALYTIQUE' in the keyword factor BEHAVIOR. In this case, the file MFront will have to use, following the value of @ModellingHypothesis, the adequate integrator;
- that is to say one utilisE the algorithm of Borst (see [R5.03.03]) with ALGO_CPLAN=' DEBORST' in the keyword factor BEHAVIOR. In this case, the file MFront will have to be able to use a modeling of the type axisymmetric.

For the unidimensional behaviors (TUYAU_*, POU_D_EM and POU_D_TGM), it is necessarily necessary to use the algorithm of Borst because MFront does not support CES modelings.

1.5 Models of great deformations

All the models of great deformations existing in code_aster are usable with MFront behaviors, in particular 'PETIT_REAC', 'GBELCH_GDEP', 'SIMO_MIEHE' and 'GDEF_LOG'.

Models PETIT_REAC and GDEF_LOG of code_aster rest necessarily on a written law in small deformations, there is thus nothing to make of individual in the MFront file.

For SIMO_MIEHE and GROT_GDEP, it will be necessary to choose in the MFront file one of the following instructions, according to the model used in the law :

@AsterFiniteStrainFormulation [aster] SIMO_MIEHE

or

@AsterFiniteStrainFormulation [aster] GROT_GDEP

Notice S :

I L has two operating processes of the model there GROT_GDEP in code_aster.

*When the model of behavior is written in **small deformations** then the structure can undergo **only great rotations and great displacements** (it is the case for the totality of the native laws* *to the format code_aster except `ELAS_HYPER` and MFront laws which do not make use of `@AsterFiniteStrainFormulation`).*

On the other hand, Q uand one uses the law `ELAS_HYPER` or a law MFront which uses `@AsterFiniteStrainFormulation [aster] GROT_GDEP` , then the behavior is solved in ***great deformations*** *.*

2 Examples of use

Other examples are provided in the tutorial available on the website of MFront (http://tfel.sourceforge.net/documentations.html).

S casetestS `mfron*` allowstent to check the keyword `RELATION=' MFRONT'`, with a MFront file provided in data of the test.

3 References

[1] T. Helfer, J. - M. Proix. *Writing of laws of behavior with MFront: tutorial*. December 2014. http://tfel.sourceforge.net/documents/tutoriel/tutoriel.pdf