Description of the mechanical behaviour interface to the Abaqus solver

T. Helfer, Kulbir Singh

SOMMAIRE

1 USAGE OF LIBRARIES GENERATED USING MFRONT

The mechanisms used by `Abaqus` to incorporate external subroutines is not suitable for mechanical behaviours generated by `MFront`.

We propose a solution which is meant to be flexible and easy to set up for the end-user.

This solution is based on a :

- a generic `umat` subroutine which aims at loading dynamically libraries generated by `MFront`.
- a naming convention of the material behaviour in the `Abaqus` input file which allow the user to specify both the function to called and the library in which this function is implemented.

This approach allows the user to build `MFront` libraries *before the computations*. Those libraries can be shared among various computations by setting the appropriate environment variable[1].

1.1 A GENERIC `umat` SUBROUTINE

A generic `umat` subroutine is delivered with `MFront`. It is implemented in a file called `umat.cpp`.

The aim of this subroutine is to dynamically load libraries generated by `MFront` using the name given to the behaviour in the input file. This subroutine can with little modifications (under comments in the source file) be made compatible with other user subroutines so mixing « standard umat » implementations and `MFront` implementations shall be feasible.

We must insist : *this file is the only source that must be compiled along with `Abaqus` : generation of `MFront` libraries is a different process that is done independently*.

As such, `abaqus` shall be called like this :

```
abaqus user=umat.cpp [options] inputfile.inp
```

1.1.1 Setting the compiler flags

The generic `umat` subroutine is written using the `C++-11` standard. Depending on the compiler and compiler version, appropriate flags shall be added for the compilation. Those flags are defined in the `abaqus_v6.env` file that can be overridden by the user.

```
1  cppCmd  = "g++"      # <-- C++ compiler
2  compile_cpp = [cppCmd,
3                 '-c', '-fPIC', '-w', '-Wno-deprecated', '-DTYPENAME=typename',
4                 '-D_LINUX_SOURCE', '-DABQ_LINUX', '-DABQ_LNX86_64', '-DSMA_GNUC',
5                 '-DFOR_TRAIL', '-DHAS_BOOL', '-DASSERT_ENABLED',
6                 '-D_BSD_TYPES', '-D_BSD_SOURCE', '-D_GNU_SOURCE',
7                 '-D_POSIX_SOURCE', '-D_XOPEN_SOURCE_EXTENDED', '-D_XOPEN_SOURCE',
8                 '-DHAVE_OPENGL', '-DHKS_OPEN_GL', '-DGL_GLEXT_PROTOTYPES',
9                 '-DMULTI_THREADING_ENABLED', '-D_REENTRANT',
10                '-DABQ_MPI_SUPPORT', '-DBIT64', '-D_LARGEFILE64_SOURCE',
11                '-D_FILE_OFFSET_BITS=64', '-std=c++11',
12                mpiCppImpl, '-I\%I']
```

FIGURE 1 : Declaration of the `C++` compiler flags for `gcc` under `LiNuX` in the `abaqus_v6.env` file. The `--std=c++11` flag was added at Line 11.

1. `LD_LIBRARY_PATH` under `Unix` systems, `PATH` under `Windows` systems

For gcc, one have to add the --std=c++11 flag. The modification made to the abaqus_v6.env are reported on Figure 1 in this case.

1.2 NAME OF THE BEHAVIOUR IN THE ABAQUS INPUT FILE

The name of the behaviour shall define the function to be called and the library in which this function is implemented. It is important to notice that the name of the behaviour is automatically converted to upper-case by Abaqus.

The name of the libraries generated by MFront though the Abaqus interface are thus upper-cased. *The user shall thus be aware that he/she must not rename MFront generated libraries using lower-case letters.*

By convention, this name is splitted into two parts, separed by the underscore character ('_'). The first part is the name of library, without prefix (lib) or suffix (.dll or .so). This convention implies that the library name does not contain an underscore character.

For example, on UNIX systems, if one want to call the ELASTICITY behaviour in libABAQUSBEHAVIOUR.so library, the name of the behaviour in the Abaqus input file has to be : ABAQUSBEHAVIOUR_ELASTICITY. This leads to the following declaration of the material :

```
*Material, name=ABAQUSBEHAVIOUR_ELASTICITY
```

1.3 NOTE ON LIBRARIES LOCATIONS

As explained above, MFront libraries will be loaded at the runtime time. This means that the libraries must be found by the dynamic loader of the operating system.

1.3.1 Under Linux

Under Linux, the search path for dynamic libraries are specified using the LD_LIBRARY_PATH variable environment. This variable defines a colon-separated set of directories where libraries should be searched for first, before the standard set of directories.

Depending on the configuration of the system, the current directory can be considered by default.

1.3.2 Under Windows

Under Windows, the dynamic libraries are searched :
- in the current directory ;
- in the directories listed in the PATH environment. This variable defines a semicolon-separated set of directories.

2 DESCRIPTION OF THE INTERFACE FUNCTIONALITIES

2.1 SUPPORTED MODELLING HYPOTHESIS

2.2 SETTING THE OUT-OF-BOUNDS POLICY

2.3 SETTING PARAMETERS VALUES

The values of the parameters can be set in an external text file which is automatically read. If the behaviour name is `Norton`, the name of this file must be `Norton-parameters.txt`. This file must be in the current directory.

The parameters file must be formatted like this :

```
ParameterName ParameterValue
OtherParameterName OtherParameterValue
... ...
```

If a parameter is omitted, this parameter will have its default value.

3 TEST CASES

3.1 UNIT TESTS IN SMALL STRAIN

Isotropic elastic behaviour under tensile loading in small strain

Isotropic elastic behaviour under shear loading in small strain

Isotropic viscoplastic behaviour under tensile loading in small strain

Isotropic plastic behaviour with isotropic hardening under shear loading in small strain

3.1.1 Unit tests in finite strain

Isotropic elastic behaviour under shear loading in finite strain

Saint-Venant Kirchhoff hyperelastic behaviour under tensile loading

Saint-Venant Kirchhoff hyperelastic behaviour under shear loading

3.2 ISOTROPIC PLASTIC BEHAVIOUR WITH ISOTROPIC HARDENING ON A NOTCHED BEAM

ANNEXE A DEFINITION OF THE CONSISTENT TANGENT OPERATOR FOR FINITE STRAIN BEHAVIOURS

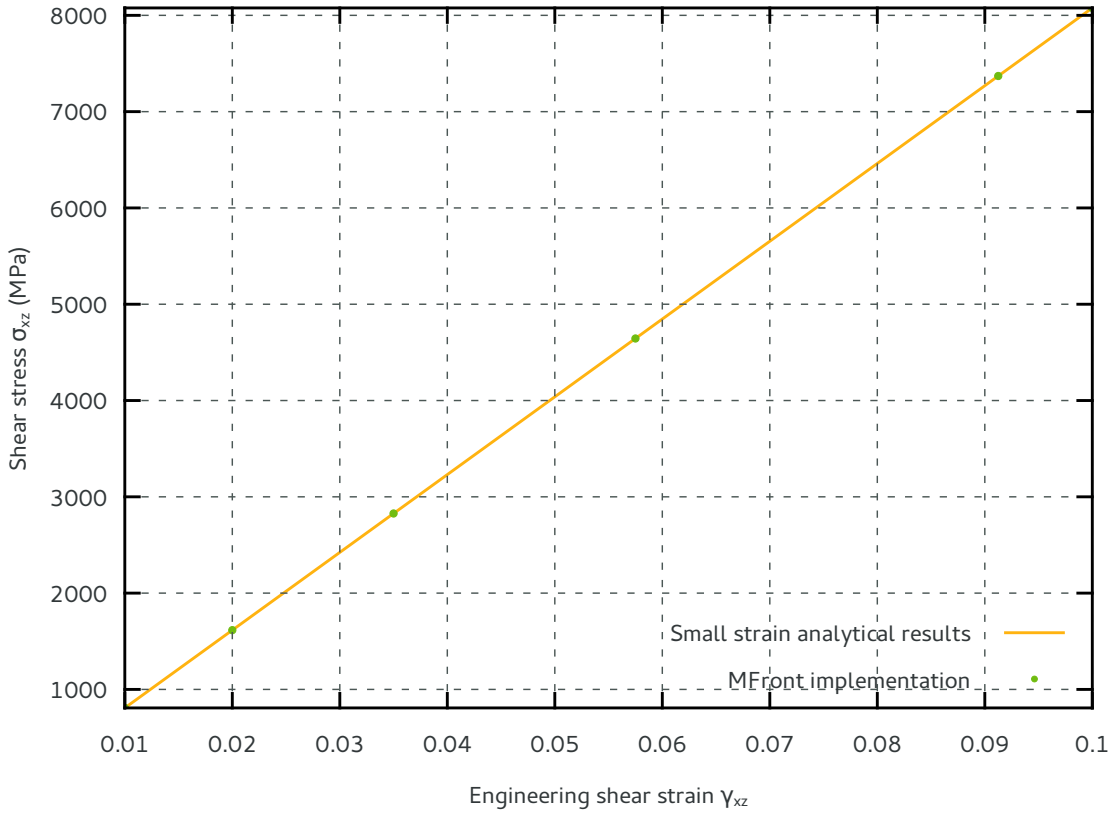

FIGURE 2 : Comparison of the results obtained by the `MFront` implementation of an isotropic elastic behaviour and the results obtained with the `Abaqus` build-in elastic behaviour under shear loading.

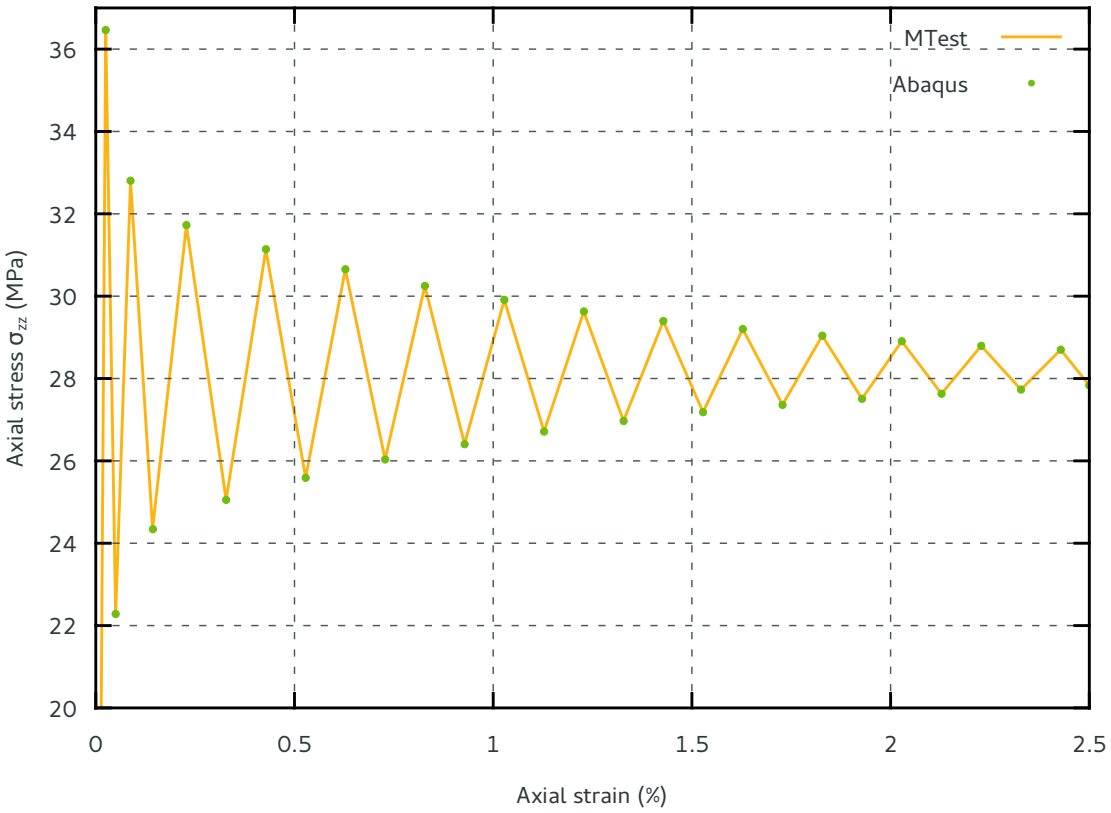

FIGURE 3 : Comparison of the results obtained with `Abaqus` and `MTest` using an `MFront` implementation of an isotropic viscoplastic behaviour under tensile loading.

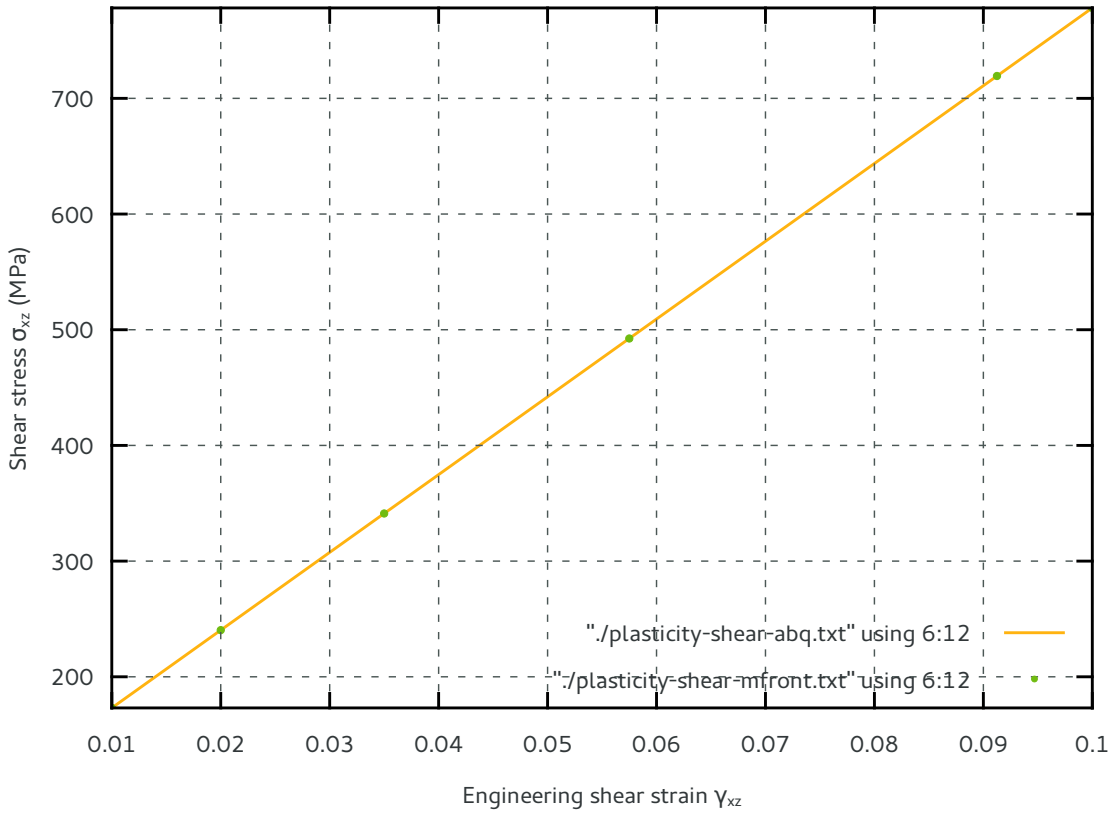

FIGURE 4 : Comparison of the results obtained by the `MFront` implementation of an isotropic plastic behaviour with isotropic hardening and the results obtained with the `Abaqus` build-in behaviour under shear loading.

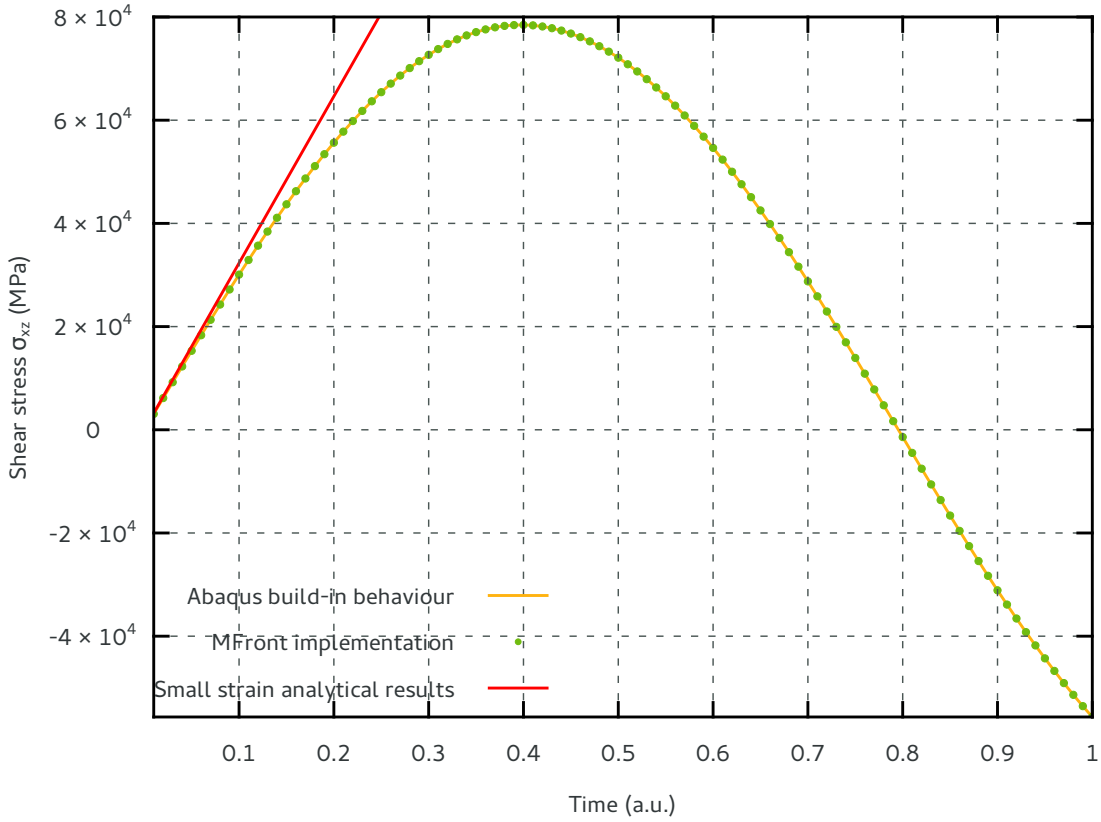

FIGURE 5 : Comparison of the results obtained by the `MFront` implementation of an isotropic elastic behaviour and the results obtained with the `Abaqus` build-in elastic behaviour under shear loading when the Hughes-Winglet hypo-elastic formulation is used.

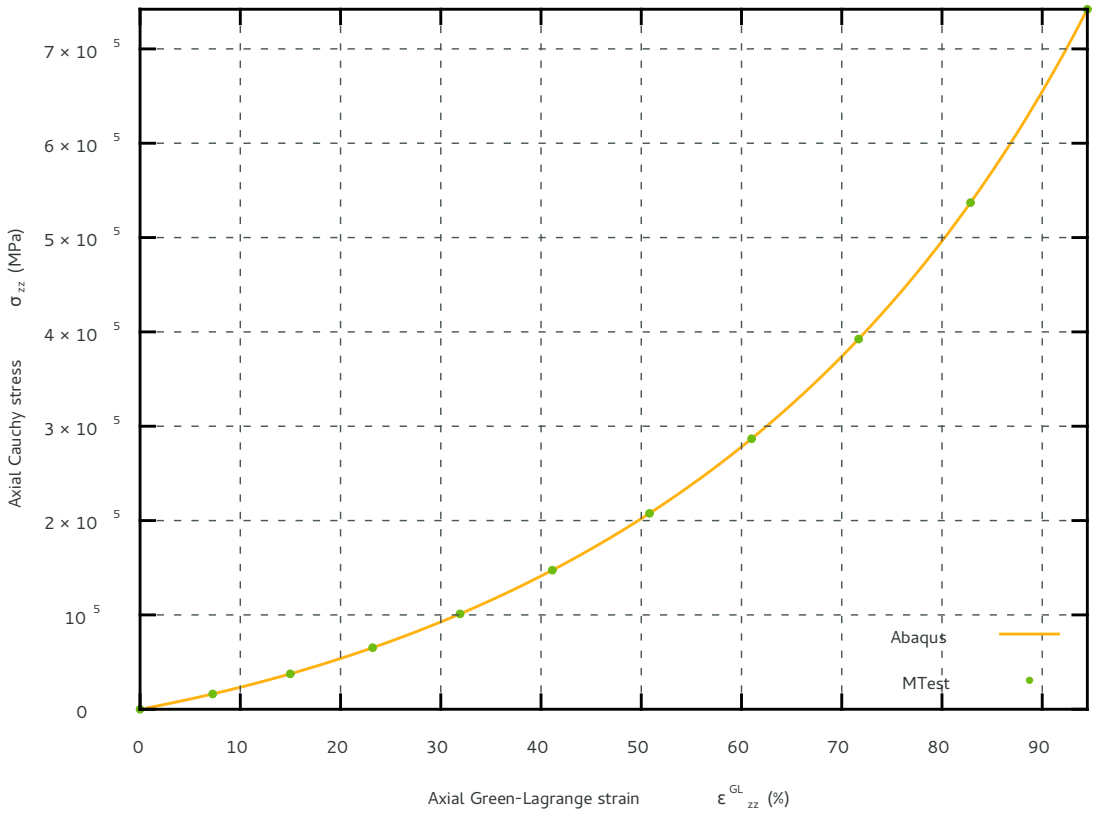

FIGURE 6 : Comparison of the results obtained the `MFront` implementation of the Saint-Venant Kirchhoff hyperelastic behaviour using `Abaqus` and `MTest`.

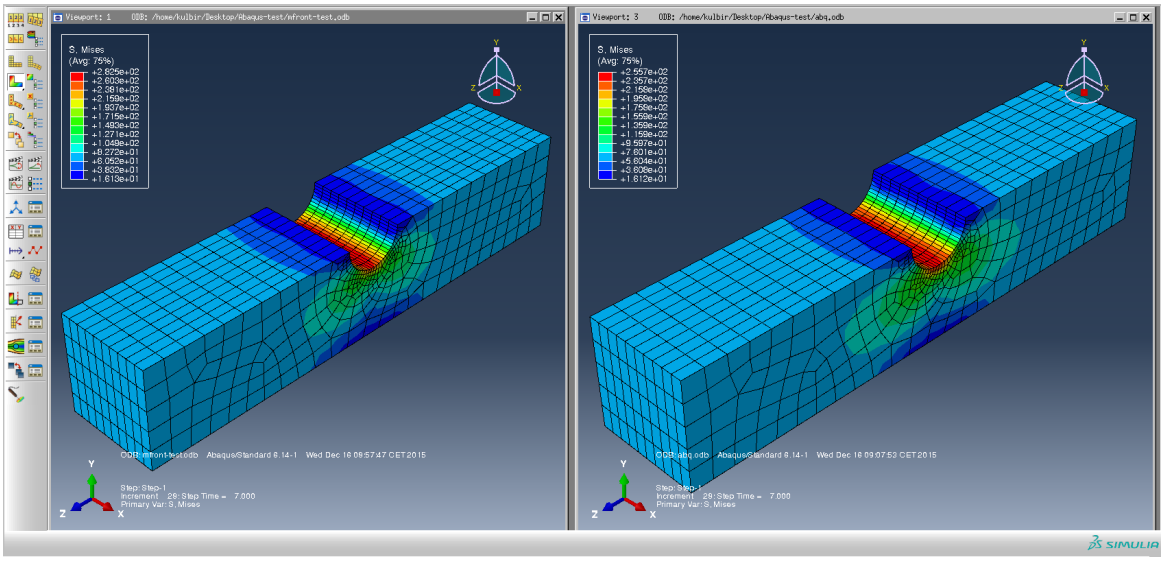

FIGURE 7 : Comparison of the VON MISES stress distribution obtained with `MFront` and the results obtained with the `Abaqus` build-in plasticity behaviour on a notched beam.

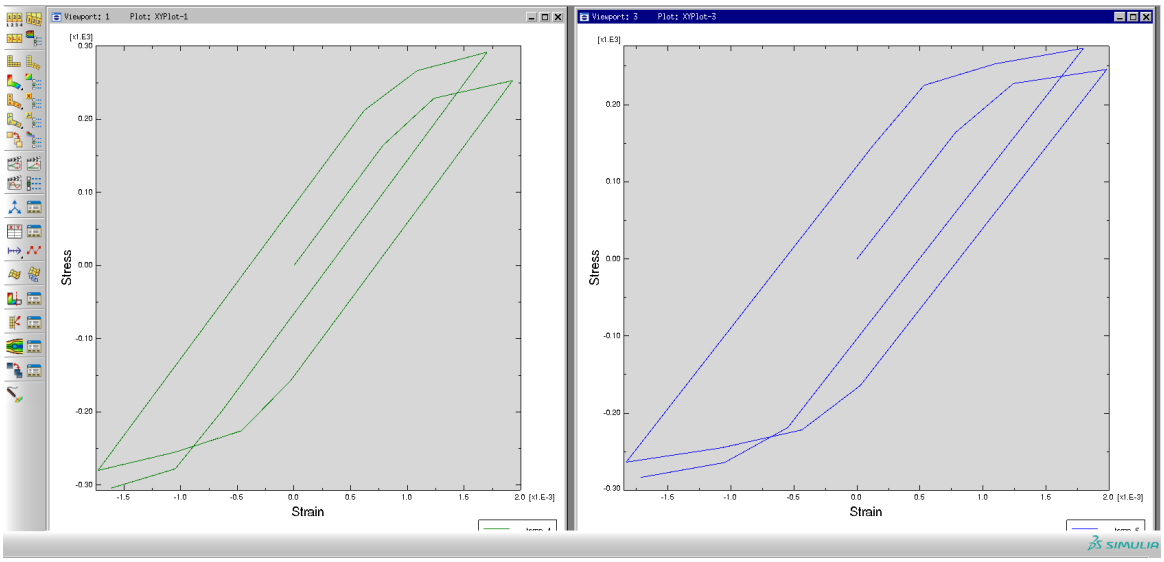

FIGURE 8 : Comparison of the stress vs. strain curves obtained with `MFront` and with the `Abaqus` build-in plasticity behaviour on a notched beam.

`Abaqus` expresses the equilibrium in the rotated frame. To our current knowledge, the derivation of the consistent tangent operator $\underline{\underline{\mathbf{C}}}$ is not described in any reference and the expression defined in the `Abaqus` manuel differs from the one given in the book of BELYTSCHKO et al. [Belytschko 00].

Let $\delta\,\underline{\mathbf{F}}$ a variation of the deformation gradient at the end of the time step. This variation induces a variation of :

— a variation $\delta\,\underline{\mathbf{L}}$ of the gradient defined by :

$$\delta\,\underline{\mathbf{L}} = \delta\,\underline{\mathbf{F}}\,.\,\underline{\mathbf{F}}\Big|_{t+\Delta\,t}^{-1}$$

— a variation $\delta\,\underline{\mathbf{D}}$ of the strain rate defined by the symmetric part of $\delta\,\underline{\mathbf{L}}$.

— a variation $\delta\,\underline{\mathbf{W}}$ of the spin rate defined by the unsymmetric part of $\delta\,\underline{\mathbf{L}}$.

— a variation of the $\Delta\,\underline{\tau}$ of the Kirchhoff stress tensor.

For the sake of conciseness, $\underline{\mathbf{F}}\Big|_{t+\Delta\,t}$ will be noted $\underline{\mathbf{F}}$ in the rest of this section.

The abaqus manual defines the consistent tangent operator $\underline{\underline{\mathbf{C}}}$ as the tangent moduli associated with the JAUMAN rate of the KIRCHHOFF stress $\underline{\underline{\mathbf{C}}}^{J\,\tau}$ divided by J. $\underline{\underline{\mathbf{C}}}$ satisfies :

$$J\,\underline{\underline{\mathbf{C}}} :\ \delta\,\underline{\mathbf{D}} = \underline{\underline{\mathbf{C}}}^{J\,\tau} :\ \delta\,\underline{\mathbf{D}} = \delta\,\underline{\tau} - \delta\,\underline{\mathbf{W}}\,.\,\underline{\tau} + \underline{\tau}\,.\,\delta\,\underline{\mathbf{W}} \tag{1}$$

The mechanical behaviour allows us to compute $\delta\,\underline{\tau}$ as :

$$\delta\,\underline{\tau} = \frac{\partial\,\underline{\tau}}{\partial\,\underline{\mathbf{F}}} :\ \delta\,\underline{\mathbf{F}}$$

The tangent operator $\dfrac{\partial\,\underline{\tau}}{\partial\,\underline{\mathbf{F}}}$ is well defined and given by most mechanical behaviour implementations.

To compute the other terms, we will need $\dfrac{\partial W}{\partial F}$:

$$\frac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}} = \frac{1}{2}\left(\partial^l_\star\left(\underset{\sim}{\mathbf{F}}^{-1}\right) - \partial^r_\star\left(\underset{\sim}{\mathbf{F}}^{-T}\right) \cdot \left(\frac{\partial \underset{\sim}{\mathbf{F}}^T}{\partial \underset{\sim}{\mathbf{F}}}\right)\right)$$

The term $\partial^l_\star\left(\underset{\sim}{\mathbf{F}}^{-1}\right)$ can be computed using the method `tpld` of the `t2tot2` class. The term $\partial^r_\star\left(\underset{\sim}{\mathbf{F}}^{-1}\right)$ can be computed using the method `tprd` of the `t2tot2` class.

This derivative allow us to compute the remaining terms of Equation (1) :

$$\delta \underset{\sim}{\mathbf{W}} \,.\, \underline{\tau} = \partial^l_\star\left(\underline{\tau}\right)\left(\frac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right) \,:\, \delta \underset{\sim}{\mathbf{F}}$$

$$\underline{\tau} \,.\, \delta \underset{\sim}{\mathbf{W}} = \partial^r_\star\left(\underline{\tau}\right)\left(\frac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right) \,:\, \delta \underset{\sim}{\mathbf{F}}$$

The terms $\partial^l_\star\left(\underline{\tau}\right)\left(\dfrac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right)$ and $\partial^r_\star\left(\underline{\tau}\right)\left(\dfrac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right)$ can be computed with the `tpld` and `tprd` of the `t2tot2` class.

Finally, Equation (1) can be recast as :

$$J\,\underline{\underline{\mathbf{C}}} :\, \delta\,\underline{\mathbf{D}} = \underline{\underline{\mathbf{K}}} :\, \delta \underset{\sim}{\mathbf{F}}$$

where $\underline{\underline{\mathbf{K}}}$ is given by :

$$\underline{\underline{\mathbf{K}}} = \frac{\partial \underline{\tau}}{\partial \underset{\sim}{\mathbf{F}}} - \partial^l_\star\left(\underline{\tau}\right)\left(\frac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right) + \partial^r_\star\left(\underline{\tau}\right)\left(\frac{\partial \underset{\sim}{\mathbf{W}}}{\partial \underset{\sim}{\mathbf{F}}}\right)$$

Following HAN [Han 12], the consistent tangent operator $\underline{\underline{\mathbf{C}}}$ can then be deduced from the $\underline{\underline{\mathbf{K}}}$ matrix using the following relationship :

$$J\,\underline{\underline{\mathbf{C}}}_{ijkl} = K_{ijkm}\,F_{lm}$$

RÉFÉRENCES

[Belytschko 00] BELYTSCHKO TED. *Nonlinear Finite Elements for Continua and Structures*. Wiley-Blackwell, Chichester ; New York, 2000.

[Han 12] HAN XU. *Modélisation de la fragilisation due au gonflement dans les aciers inoxydables austénitiques irradiés*. Thèse : Ecole Doctorale Sciences des Métiers de l'Ingénieur, Paris, France, 2012.