BACK TO BASICS: AN INTRODUCTION TO MFRONT THROUGH `ISOTROPIC DOMAIN SPECIFIC LANGUAGES'

9th MFront User Day 2023

M. Wangermez, T Helfer

PREAMBLE - GENERAL INFORMATION

MFront is open-source and available here:

https://github.com/thelfer/tfel

The documentation is available here:

https://thelfer.github.io/tfel/web/index.html

This presentation is based on the ongoing MFrontbook

OUTLINE

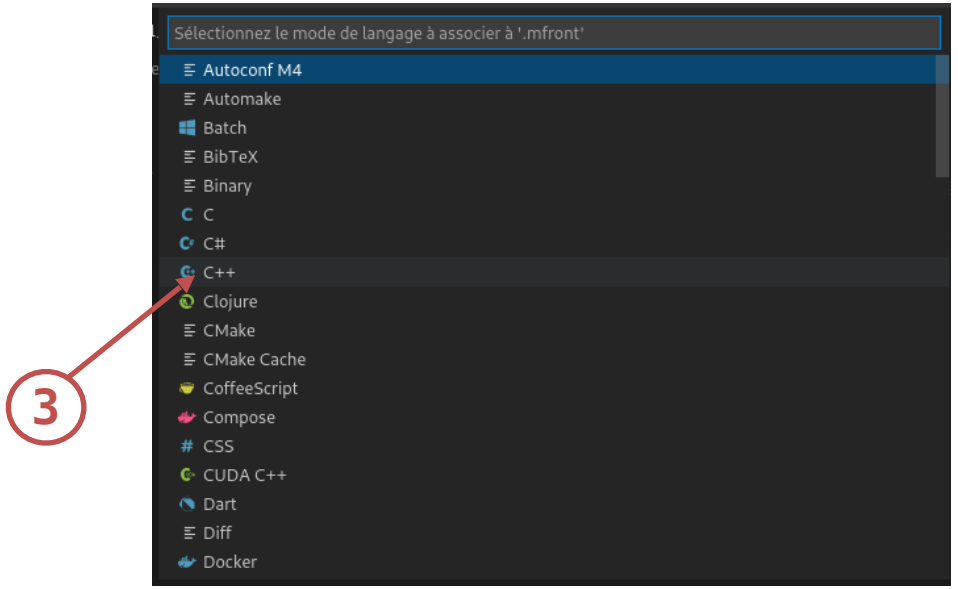
1. First writing and use of a MFront library with MTest
2. Analysis of the MFront file content
3. Isotropic DSLs and @FlowRule directive
4. Improvement and best practices (quality insurance)

FIRST WRITING AND USE OF A MFRONT LIBRARY WITH MTEST

FIRST FILE WRITING - CONFIGURATION

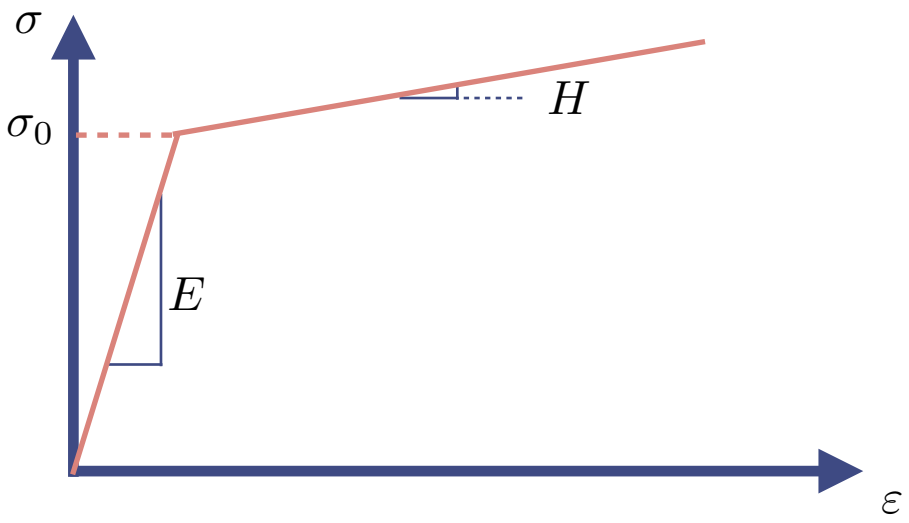
Use C++ syntax highlighting with MFront files

- Use emacs-based tfel-editor (available here: https://github.com/thelfer/tfel-editor.git)
- Associate MFront files with C++ coloring. For example, in Visual Studio Code:



FIRST FILE WRITING - MFRONT FILE

Isotropic linear hardening modeling



The plastic part of the behaviour is described by the following yield surface:

$$f(\sigma_{eq}, p) = \sigma_{eq} - Hp - \sigma_0$$

Minimal MFront file `01-IsotropicPlasticMisesFlow.mfront':

```
@DSL       IsotropicPlasticMisesFlow;
@Behaviour IsotropicLinearHardeningPlasticity;

@FlowRule{
    auto H  = 22.e9;
    auto s0 = 200.e6;
    f       = seq-H*p-s0;
    df_dseq = 1;
    df_dp   = -H;
}
```

FIRST FILE WRITING - COMPILATION

Compiling the Mfront file (PlaneStress not supported yet):

```
$ mfront --obuild --interface=generic 01-IsotropicPlasticMisesFlow.mfront
Treating target : all
The following library has been built :
- libBehaviour.dylib :
IsotropicLinearHardeningPlasticity_AxisymmetricalGeneralisedPlaneStrain
IsotropicLinearHardeningPlasticity_Axisymmetrical
IsotropicLinearHardeningPlasticity_PlaneStrain
IsotropicLinearHardeningPlasticity_GeneralisedPlaneStrain
IsotropicLinearHardeningPlasticity_Tridimensional
```

Name of the library generated by MFront in the src folder

Names of behaviour laws generated (with their modelling assumptions)

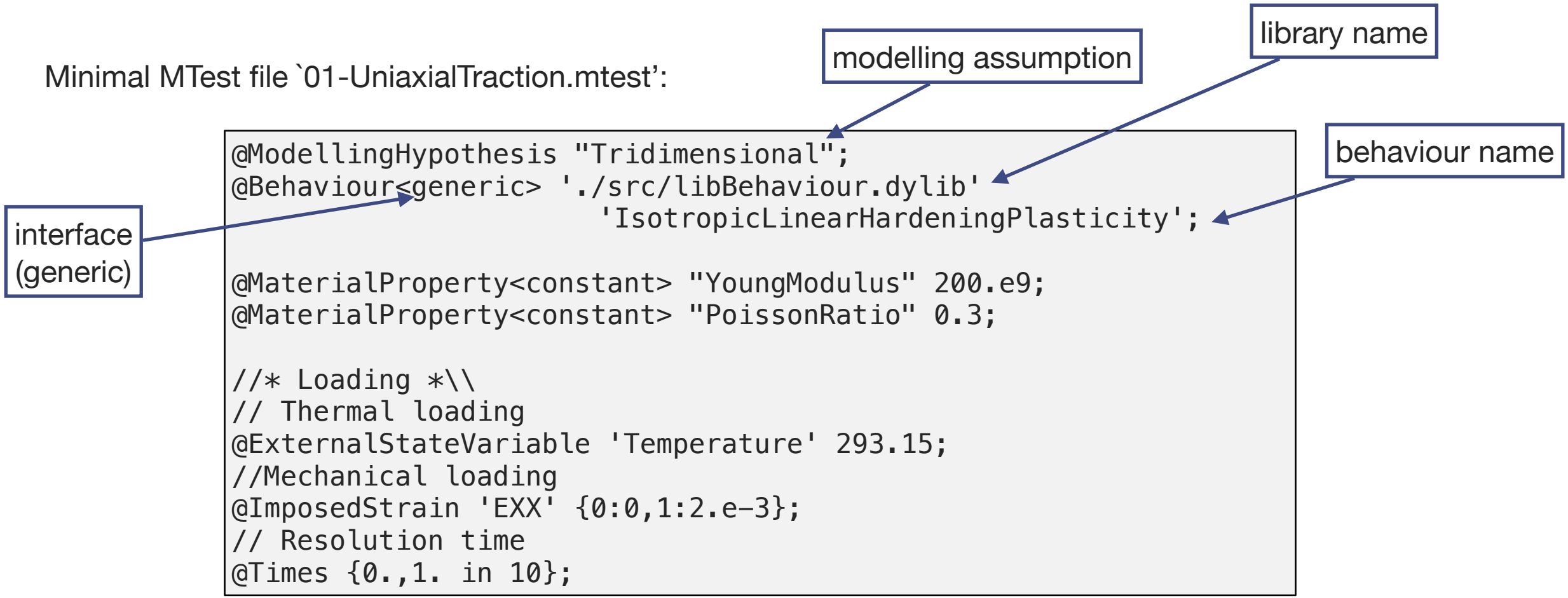
We notice the creation of two folders: src/ and include/

These two directories **are not working directories**, since they are often deleted.

FIRST FILE WRITING - USE WITH MTEST

Use of the library with MTest:

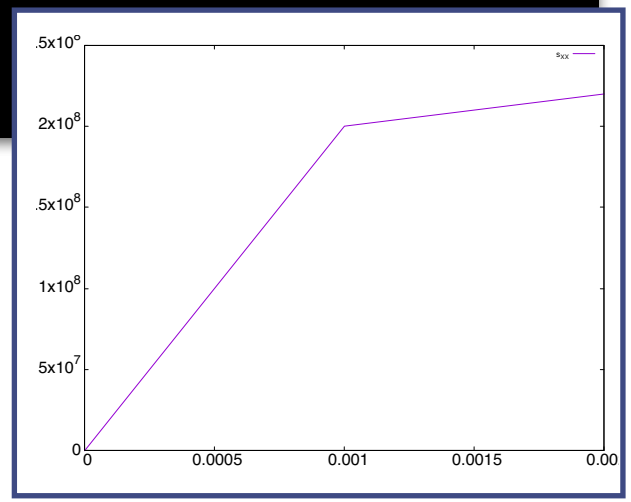
Minimal MTest file `01-UniaxialTraction.mtest':



FIRST FILE WRITING - USE WITH MTEST

Simulation with MTest:

```
$ mtest UniaxialTraction.mtest

. . .
Execution succeeded
-number of period:     10
-number of iterations: 25
-number of sub-steps:  0
Result of test 'unit behaviour test' of group 'MTest'              : SUCCESS
End of Test Suite                                                  : SUCCESS

$ gnuplot
gnuplot> plot "UniaxialTraction.res" u 2:8 w l title « S_{XX}"
```



ANALYSIS OF THE FIRST FILE CONTENT

ANALYSIS OF THE FIRST FILE CONTENT

```
@DSL       IsotropicPlasticMisesFlow;           // domain specific language
@Behaviour IsotropicLinearHardeningPlasticity;

@FlowRule{
    auto H = 22.e9;
    auto s0 = 200.e6;
    f       = seq-H*p-s0;
    df_dseq = 1;
    df_dp   = -H;
}
```

@DSL (Domain Specific Language): Tell MFront how to interpret the file :

- Material Property -> MaterialLaw
- **Material Behaviour (today) -> Implicit, ImplicitParser, RungeKutta, etc.**
- Point-wise model -> Model, DefaultModel, ImplicitModel

To display the list of available DSLs:

```
$ mfront --list-dsl
```

ANALYSIS OF THE FIRST FILE CONTENT - DSL

Each **DSL** has its conventions and **keywords**, fortunately, they are often common to several DSLs.

To display the list of keywords associated with the DSL IsotropicPlasticMisesFlow:

```
$ mfront --help-keywords-list=IsotropicPlasticMisesFlow
```

Each keyword in a DSL is documented! For example:

to display the documentation for the @AuxiliaryStateVariable keyword of the IsotropicPlasticMisesFlow DSL:

```
$ mfront --help-keyword=IsotropicPlasticMisesFlow:@AuxiliaryStateVariable
The `AuxiliaryStateVariable` keyword introduces one or several new
auxiliary state variables. It is followed by a type name and the
name(s) of the variable(s) declared, separated by commas.
```

ANALYSIS OF THE FIRST FILE CONTENT

```
@DSL       IsotropicPlasticMisesFlow;            // domain specific language
@Behaviour sotropicLinearHardeningPlasticity;  // name of the behaviour

@FlowRule{
    auto H = 22.e9;
    auto s0 = 200.e6;
    f       = seq-H*p-s0;
    df_dseq = 1;
    df_dp   = -H;
}
```

@Behaviour: defines the name of the behaviour.

To display the documentation of the @Behaviour keyword related to the IsotropicPlasticMisesFlow DSL:

```
$ mfront --help-keyword=IsotropicPlasticMisesFlow:@Behaviour
```

ANALYSIS OF THE FIRST FILE CONTENT

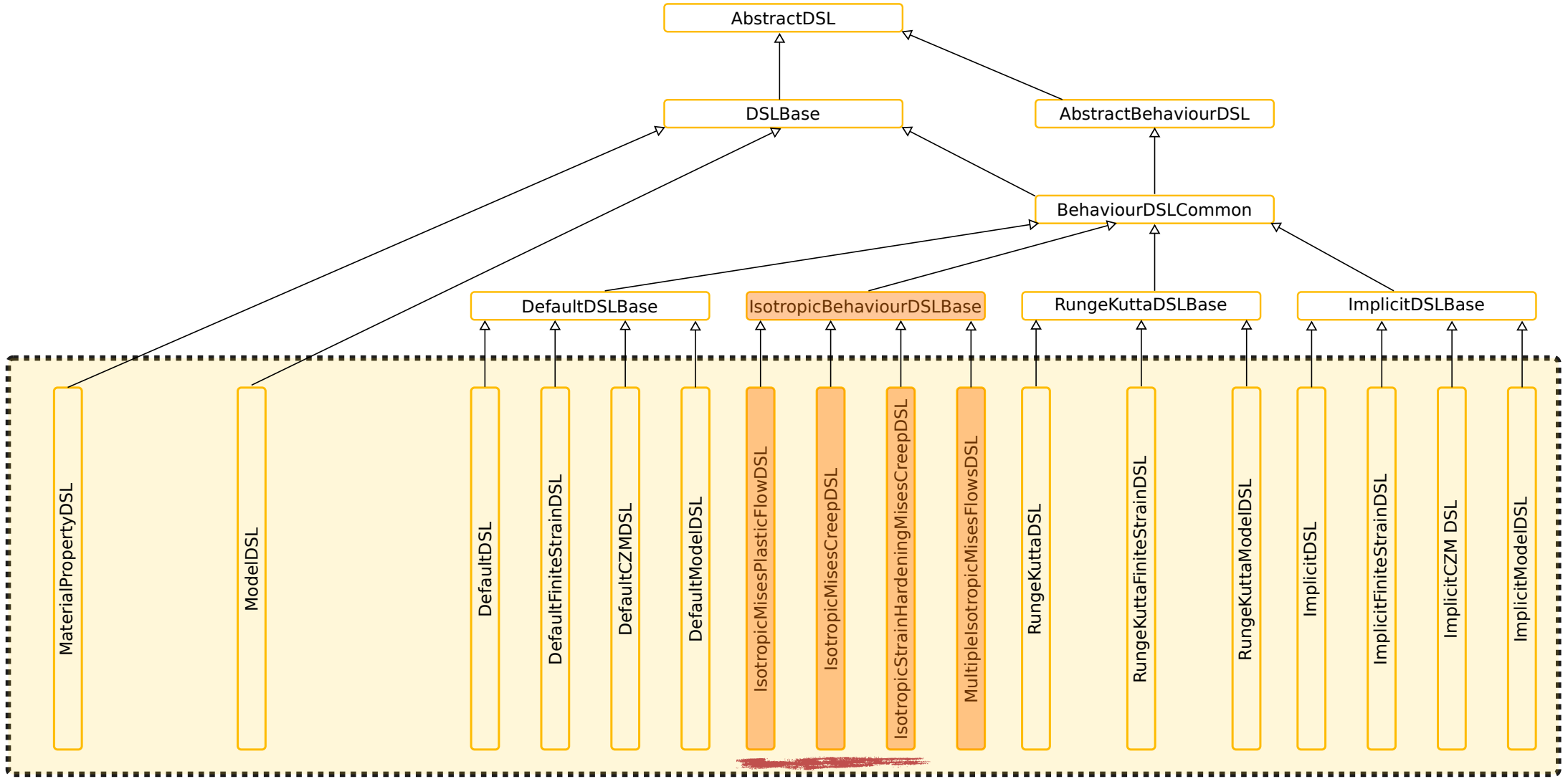
```
@DSL       IsotropicPlasticMisesFlow;           // domain specific language
@Behaviour IsotropicLinearHardeningPlasticity;  // name of the behaviour

@FlowRule{
    auto H = 22.e9;
    auto s0 = 200.e6;
    f       = seq-H*p-s0;
    df_dseq = 1;
    df_dp   = -H;
}
```

The @FlowRule mandatory directive is used to define a plastic flow in conjunction with the DSL IsotropicPlasticMisesFlow.

3 other specific DSLs are available in MFront to describe standard flows

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE



Optimized DSL (scalar equation solving)

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE

Four specialized DSLs:

- **IsotropicPlasticMisesFlow**: standard plastics behaviours characterized by a yield surface of the form:

$$f(\sigma_{eq}, p) = 0$$

- **IsotropicMisesCreep**: standard creep behaviors described by the equation:

$$\dot{p} = f(\sigma_{eq})$$

- **IsotropicStrainHardeningMisesCreep**: standard strain hardening creep behaviours described by the equation:

$$\dot{p} = f(\sigma_{eq}, p)$$

- **MultipleIsotropicMisesFlows**: combining several isotropic flows. Supported flow types include:
  - Plasticity: $f(\sigma_{eq}, p) = 0$
  - Creep: $\dot{p} = f(\sigma_{eq})$
  - StrainHardeningCreep: $\dot{p} = f(\sigma_{eq}, p)$

p : equivalent plastic strain σ_{eq} : equivalent Mises stress

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE

IMPLICIT DECLARATIONS common to the 4 specific DSLs:

- **Material properties:** **young** and **nu**, with references to the glossary entries **YoungModulus** and **PoissonRatio**
- **State variables:** **eel** for elastic strain (**deel** its increment) and **p** for cumulative inelastic strain (**dp** its increment), with glossary references **ElasticStrain** and **EquivalentPlasticStrain**.
- **External state variables:** **T** for the temperature at the start of the time step (**dT** its increment) with glossary reference **Temperature**
- **Thermodynamic forces:** **sig** for the stress tensor
- **Gradients:** **eto** for the total strain (**deto** its increment)
- **Local variables:**
  - **T_**: current temperature defined as: T_ = T + \theta dT
  - **n** : flow direction
  - **seq**: current equivalent stress in time: t + \theta dt
  - **se**: elastic prediction of stress deviator
  - **seq_e**: elastic prediction of equivalent stress

```
$ mfront-query --material-properties Plasticity.mfront
- YoungModulus (young): the Young's modulus of an isotropic material
- PoissonRatio (ν): the Poisson ratio of an isotropic material
```

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE

IMPLICIT DECLARATIONS <u>according to each specific DSL</u>:

- **Local variables definition:**
  - **f**: flow function
  - **df_dseq**: derivative of f with respect to the equivalent stress
  - **df_dp**: derivative of f with respect to the cumulative inelastic strain
  - **p_**: equivalent deformation in t + \theta dt

	IsotropicPlasticMisesFlow	IsotropicMisesCreep	IsotropicStrainHardening MisesCreep	MultipleIsotropicMisesFlows
Local variables	f df_dseq df_dp p_	f df_dseq . .	f df_dseq df_dp p_	. . . p_

Case of **MultipleIsotropicMisesFlows**: several `@FlowRule` blocks can be defined.
A new @FlowRule block declares the associated state variable **pi** and the associated local variables **fi**, **df_dseqi** (eventually **df_dpi**) where **i** is the flow number defined so far.

ISOTROPIC DSLs AND FLOWRULE DIRECTIVE

The @FlowRule block MUST BE FILLED IN with:

- the function **f**
- some of its derivatives (according to **Tab.1**)
  - **df_dseq**: derivative of f with respect to the equivalent stress
  - **df_dp**: derivative of f with respect to the cumulative inelastic strain

	IsotropicPlasticMisesFlow	IsotropicMisesCreep	IsotropicStrainHardeningMisesCreep
Local variables	f df_dseq df_dp	f df_dseq .	f df_dseq df_dp

Tab.1 - mandatory variables according to the flow type

Case of **MultipleIsotropicMisesFlows:** each flow follows the same rules as in **Tab. 1**

IMPROVEMENTS AND BEST PRACTICES

ADD SOME INFORMATIONS

BEST PRACTICE:

- use explicit name of the behaviour.
- change the name of the MFront file to be consistent with the name of the behaviour.
- add **@Author**, **@Date** and **@Description** blocks to fill in informations about your MFront files

```
@DSL       IsotropicPlasticMisesFlow;           // domain specific language
@Behaviour IsotropicLinearHardeningPlasticity;  // name of the behaviour

@Author Maxence Wangermez / Thomas Helfer;
@Date 12/04/2023;

@Description {
  An implementation of a simple
  isotropic plasticity behaviour with
  isotropic linear hardening.
  The yield surface ($f(s, p) = 0$) is defined by:
      f(\sigmaeq,p) = \sigmaeq-s_{0}-H*p
  $p$ represents the equivalent creep strain,
  $s$ represents the equivalent Mises stress.
}

@FlowRule{...}
```

USE OF PARAMETERS

Entering quantities in the form of parameters facilitates:

- sensitivity studies
- taking into account the propagation of uncertainties in the data
- potential re-identifications

```
@Parameter H = 22.e9; // isotropic hardening slope
@Parameter s0 = 200.e6; // initial elasticity limit

@FlowRule{
    auto H = 22.e9;
    auto s0 = 200.e6;
    f       = seq-H*p-s0;
    df_dseq = 1;
    df_dp   = -H;
}
```

MODIFICATION OF PARAMETERS

Modification of the parameters with a .txt file:

- the file contains lines: <parameter name> <new parameter value>
- the expected file name in the current directory is given by command:

```
$ mfront-query --parameters-file 04-IsotropicPlasticMisesFlow_parameters.mfront
IsotropicLinearHardeningPlasticity-parameters.txt
```

IsotropicLinearHardeningPlasticity-parameters.txt:

```
# new material parameters
H       26.e9
s0      180.e6
```

Please note that there is no mention of parameter replacement !
Check that the values are taken into account.

DOCUMENT THE NAMES OF THE PARAMETERS

The names of the variables are not explicit. To improve the clarity and unambiguity of the MFront file, it is possible to use the TFEL Glossary.

```
@Parameter H = 22.e9; // isotropic hardening slope
H.setEntryName("HardeningSlope");
@Parameter s0 = 200.e6; // initial elasticity limit
s0.setGlossaryName("YieldStress");
```

This also helps the interoperability of the library since the TFEL glossary defines a set of uniquely defined names that can be used to qualify a variable.

If the variable name does not exist in the TFEL glossary : use the method setEntryName(str)

Some values are aliases, check with mfront-query or the glossary documentation: https://thelfer.github.io/tfel/web/glossary.html

```
mfront-query --parameters 05-IsotropicPlasticMisesFlow_parameters.mfront
- HardeningSlope (H)
- YieldStrength (s0): the stress corresponding to the yield point at which the
material begins to deform plastically
```

MODIFICATION OF PARAMETERS

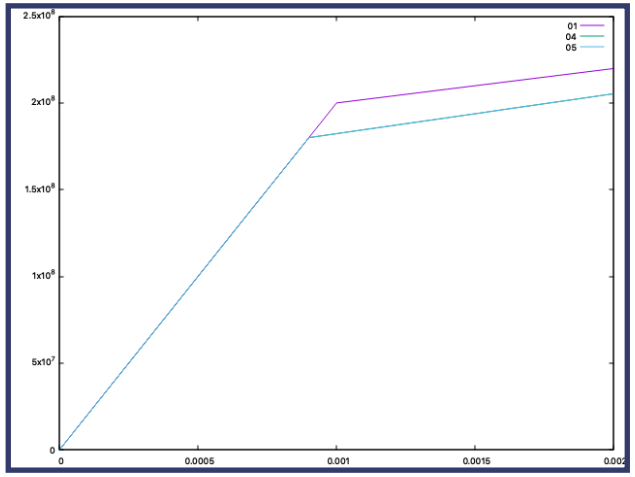
Use their **glossary names** if the parameters are associated with them.

```
@Parameter H = 22.e9; // isotropic hardening slope
H.setEntryName("HardeningSlope");
@Parameter s0 = 200.e6; // initial elasticity limit
s0.setGlossaryName("YieldStress");
```

```
mfront-query --parameters 05-IsotropicPlasticMisesFlow_parameters.mfront
- HardeningSlope (H)
- YieldStrength (s0): the stress corresponding to the yield point at which the
material begins to deform plastically
```

IsotropicLinearHardeningPlasticity-parameters.txt:

```
# new material parameters
HardeningSlope 26.e9
YieldStrength 180.e6
```



DECLARE THE ELASTIC PROPERTIES

Young's modulus and **Poisson's ratio** are automatically declared by the DSL

```
@Parameter stress YoungModulus = 200e9;
```

```
$ mfront --obuild --interface=generic 06-*.mfront
Error while treating file '06-IsotropicPlasticMisesFlow_ElasticProperties.mfront'
BehaviourDSLCommon::analyse: error while treating keyword '@Parameter' at line '27' of file '06-
IsotropicPlasticMisesFlow_ElasticProperties.mfront'.
BehaviourData::registerMemberName: the name 'YoungModulus' is a registred as a glossary name.
```

Two ways to set the Elastic Properties:

1. set from the calling code by adjusting the values linked to the material properties `YoungModulus` and `PoissonRatio` within the law generated by `MFront`.

   For example, as it has been done until now with MTest (slide 8):

```
@MaterialProperty<constant> "YoungModulus" 200.e9;
@MaterialProperty<constant> "PoissonRatio" 0.3;
```

DECLARE THE ELASTIC PROPERTIES

2. set from the MFront file:

The @ElasticMaterialProperties keyword is used to define the material properties for standard mechanical behaviours.
Isotropic case -> two entries in the following order:
- **Young's Modulus**
- **Poisson's ratio**.

can be filled with a **numerical value**, a **formula** or an **external MFront file**

```
// numerical values
@ElasticMaterialProperties {200.e9, 0.3};
```

```
// formula
@Parameter E0 = 2.1421e11, E1 = -3.8654e7, E2 = -3.1636e4;
@ElasticMaterialProperties {"E0+(T-273.15)*(E1+E2*(T-273.15))",0.3};
```

```
// external MFront file
@ElasticMaterialProperties {"YoungModulus_Martin1989.mfront",0.3};
```

USE AUXILIARY STATE VARIABLES

The @AuxiliaryStateVariable directive is used to declare internal variables (scalar or tensor) which are not part of the system to be integrated but which may be useful for post-processing:

```
@AuxiliaryStateVariable StrainStensor evp1;
evp1.setGlossaryName("PlasticStrain");
@AuxiliaryStateVariable StrainStensor evp2;
evp2.setEntryName("PlasticStrain2");
```

They must be updated with the @UpdateAuxiliaryStateVariables:

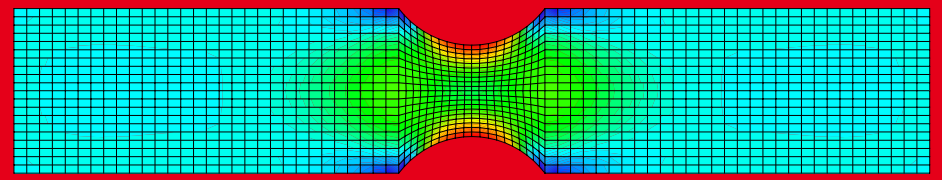
```
@UpdateAuxiliaryStateVariables{
  evp1 += deto-deel;
  evp2 = eto+deto-eel;
}
```

These variables are updated:

- after the integration of internal variables and the updating of stresses,
- before external variables (temperature, total deformation).

Consequence: `eel` has been updated while `eto` has not

Conclusions



CONCLUSIONS

- Specific DSLs:
  - IsotropicPlasticMisesFlow
  - IsotropicMisesCreep
  - IsotropicStrainHardeningMisesCreep
  - MultipleIsotropicMisesFlows

  dedicated to isotropic plastic flow have been investigated
- Easy to use with the @Flowrule block requiring the expression of the function f and its derivatives
- Some examples have been given with the MTest solver through MFront's generic interface
- Could be very efficient for GPU applications (To be validated).
- These models can also be generated by brick StandardElastoViscoPlasticity (less efficient)

```
@DSL       IsotropicPlasticMisesFlow; // domain specific language
@Behaviour IsotropicLinearHardeningPlasticity; // name of the behaviour

@Author Maxence Wangermez;
@Date 12 / 04 / 2023;

@Description {
  An implementation of a simple
  isotropic plasticity behaviour with
  isotropic linear hardening.

  The yield surface ($f(s, p) = 0$) is defined by:
  "\["
  "  f(\sigmaeq,p) = \sigmaeq-s_{0}-H\,p"
  "\]"
  $p$ represents the equivalent creep strain,
  $s$ represents the equivalent Mises stress.
}

@ElasticMaterialProperties {200.e9, 0.3};
@Parameter stress H = 22.e9; // isotropic hardening slope
H.setEntryName("HardeningSlope");
@Parameter stress s0 = 200.e6; // initial elasticity limit
s0.setGlossaryName("YieldStress");

@AuxiliaryStateVariable StrainStensor evp1;
evp1.setGlossaryName("PlasticStrain");

// Mandatory block
@FlowRule{
  f       = seq-H*p-s0;
  df_dseq = 1;
  df_dp   = -H;
}

@UpdateAuxiliaryStateVariables{
  evp1 += deto-deel;
}
```

Questions ?