Elasto-viscoplastic behaviour law for irradiated low-allow steels

G. Monnet (EDF R&D, MMC), J. Dalphin (EDF R&D, ERMES)

11th MFront User Meeting - 20/11/2025

Outline

1. Motivation and strategy

2. Constitutive equations and modeling

3. MFront implementation

1.1 Motivation

Prediction of mechanical properties of low-alloy steels accounting for

- Temperature
- Strain rate
- Irradiation damage (size and density of solute clusters)

1.1 Motivation

Prediction of mechanical properties of low-alloy steels accounting for

- Temperature
- Strain rate
- Irradiation damage (size and density of solute clusters)

Requirements

- Isotropic flow behaviour
- Input parameters ≡ microstructure characteristics
- ONE adjusted parameter (unirradiated) and TWO for irradiated steels

1.2 Strategy: multiscale modeling

Extensive modeling effort on mechanisms of plastic deformation

- Lattice friction (temperature and strain rate effects)
- Solid solution hardening
- Forest hardening
- Slip system activity
- Precipitation hardening (addressing radiation hardening)

1.2 Strategy: multiscale modeling

Extensive modeling effort on mechanisms of plastic deformation

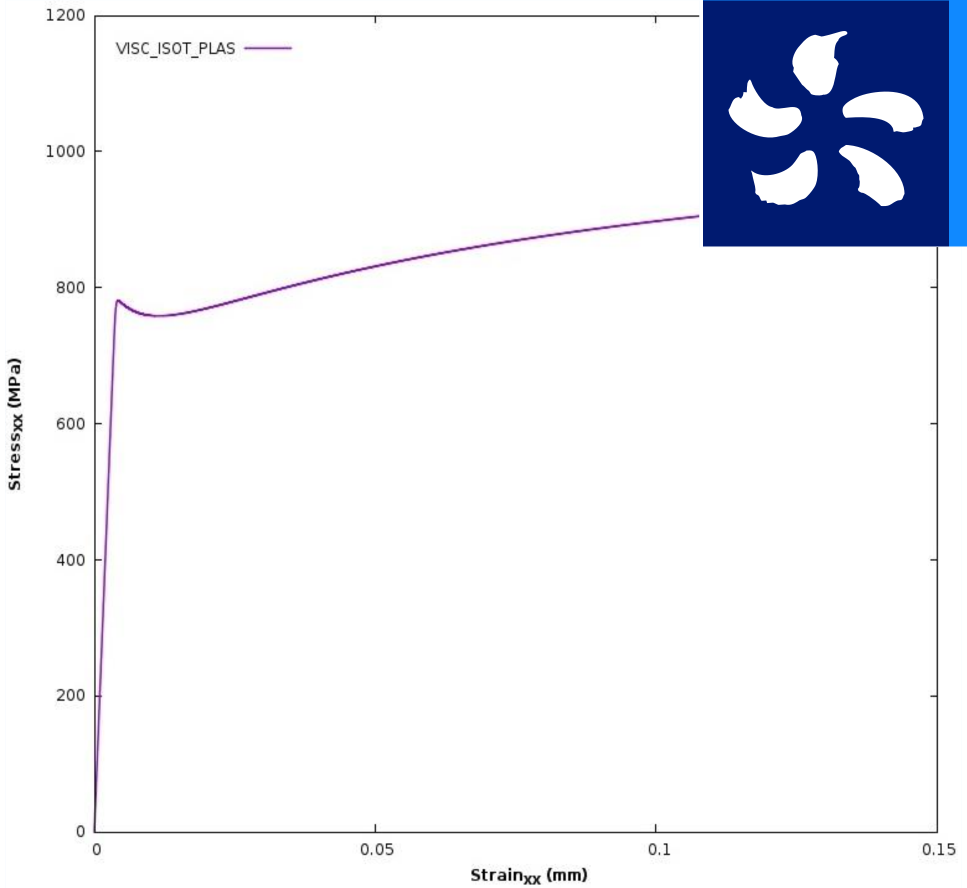
- Lattice friction (temperature and strain rate effects)
- Solid solution hardening
- Forest hardening
- Slip system activity
- Precipitation hardening (addressing radiation hardening)

Empirical mechanisms

- Hall-Petch or grain-size effect
- Static and dynamic ageing
- Jog drag rate-controlled flow

2.1 Constitutive equations

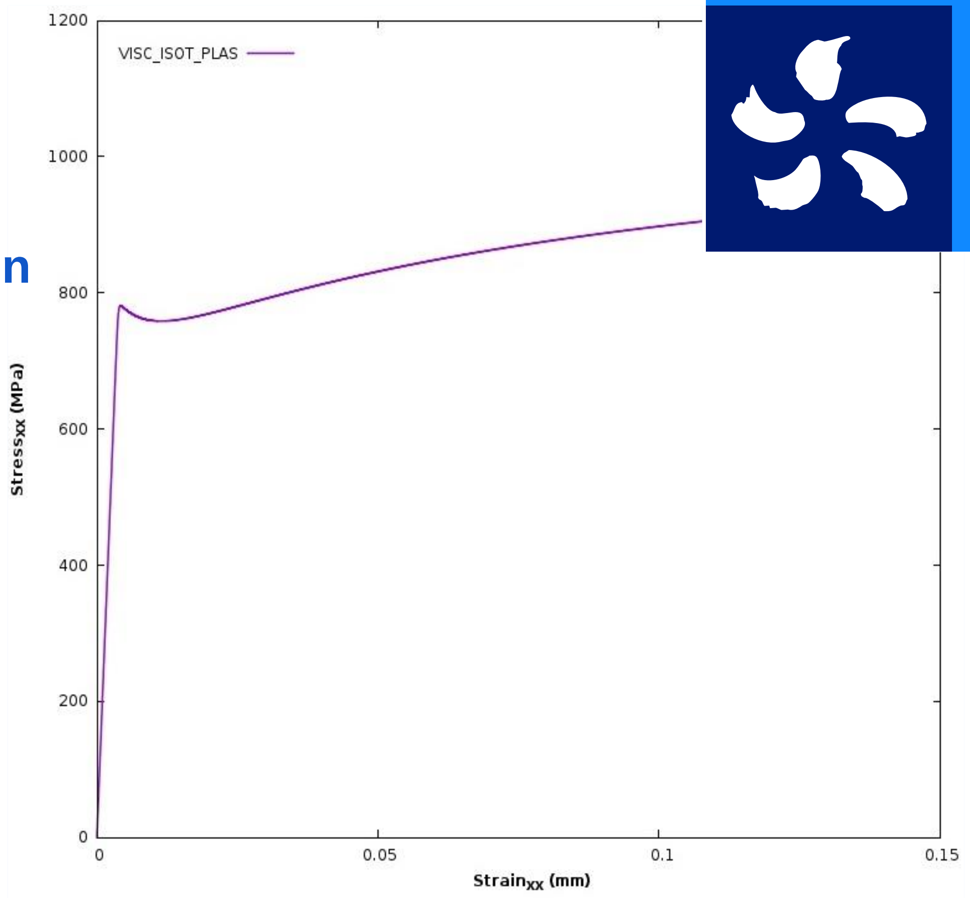
- Temperature $T(°C)$ is used and assumed constant



2.1 Constitutive equations

- Temperature $T(°C)$ is used and assumed constant
- Infinitesimal deformations and strain decomposition

$$\underline{\varepsilon}^{to} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{vp}$$



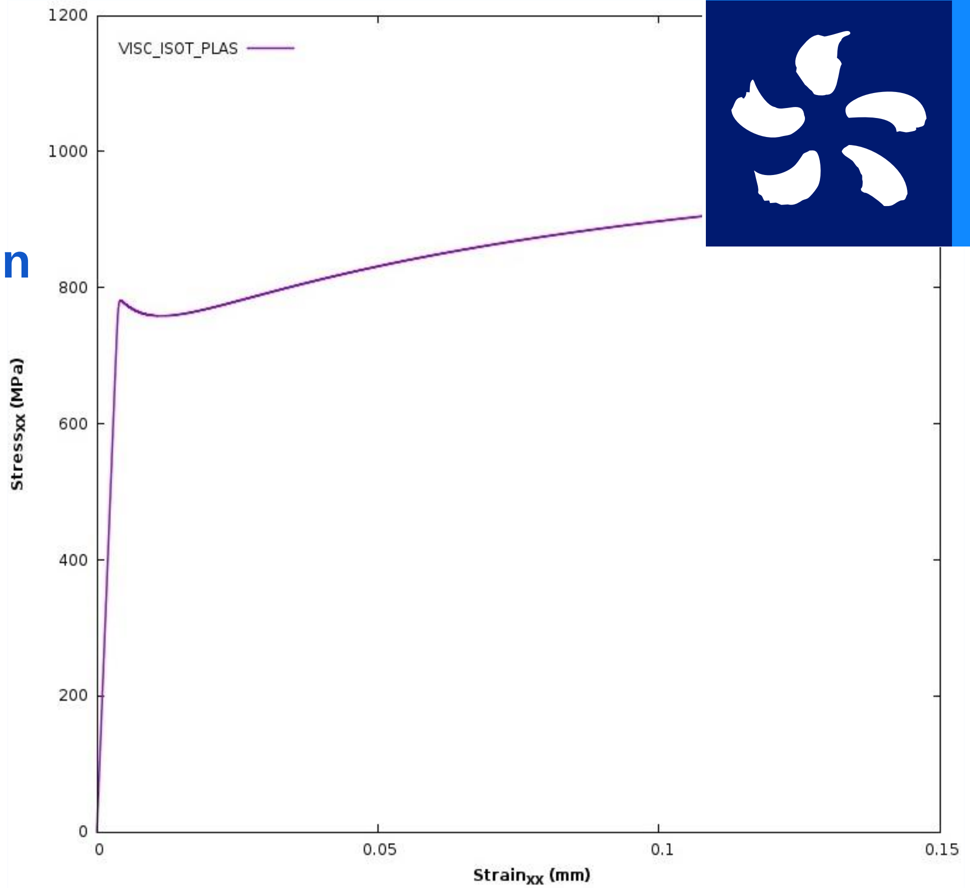
2.1 Constitutive equations

- Temperature $T(°C)$ is used and assumed constant
- Infinitesimal deformations and strain decomposition

$$\underline{\varepsilon}^{to} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{vp}$$

- Hooke's law (E,ν) with shear modulus $\mu := \frac{E(T(°C))}{2(1+\nu)}$

$$\underline{\sigma} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \underline{\varepsilon}^{el}$$



2.1 Constitutive equations

- **Temperature $T(°C)$ is used and assumed constant**
- **Infinitesimal deformations and strain decomposition**

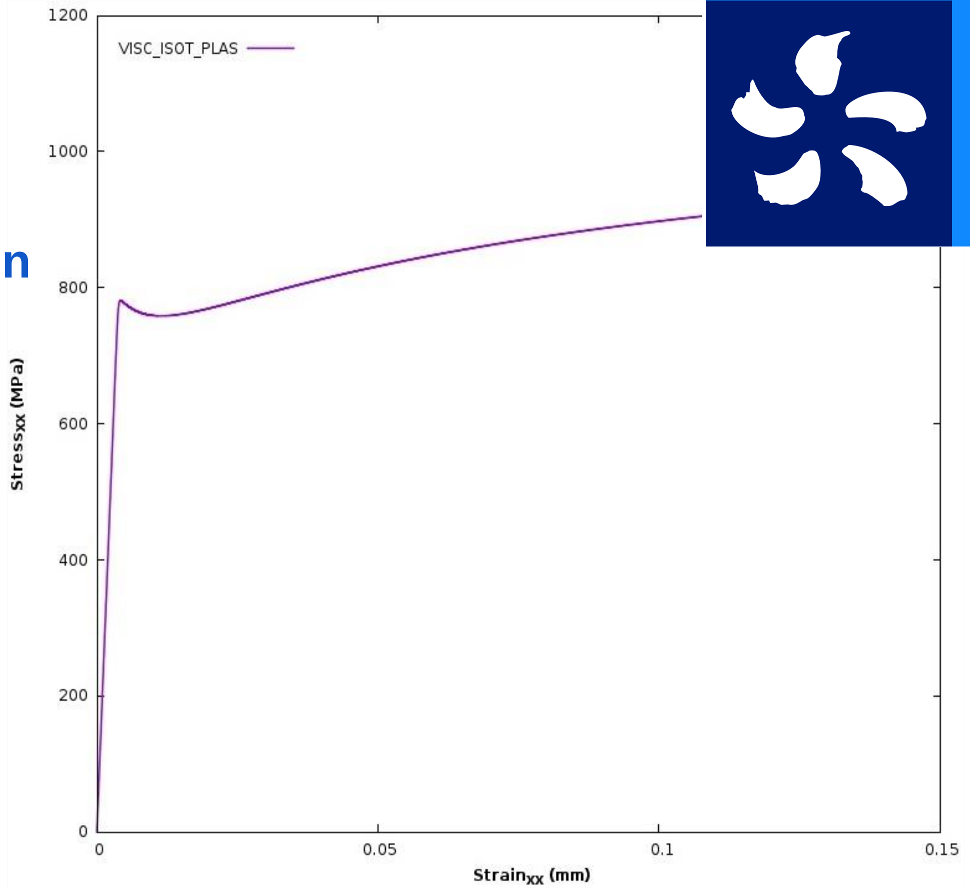
$$\underline{\varepsilon}^{to} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{vp}$$

- **Hooke's law (E,ν) with shear modulus $\mu := \frac{E(T(°C))}{2(1+\nu)}$**

$$\underline{\sigma} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \underline{\varepsilon}^{el}$$

- **Von Mises stress criterion with $\underline{\tilde{\sigma}} := \left(\underline{\underline{I}} - \frac{1}{3}\underline{I}\otimes\underline{I}\right) : \underline{\sigma}$**

$$\sigma_{eq} = \sqrt{\frac{3}{2}\underline{\tilde{\sigma}} : \underline{\tilde{\sigma}}}$$



2.1 Constitutive equations

- **Temperature $T(°C)$ is used and assumed constant**
- **Infinitesimal deformations and strain decomposition**

$$\underline{\varepsilon}^{to} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{vp}$$

- **Hooke's law (E,ν) with shear modulus $\mu := \frac{E(T(°C))}{2(1+\nu)}$**

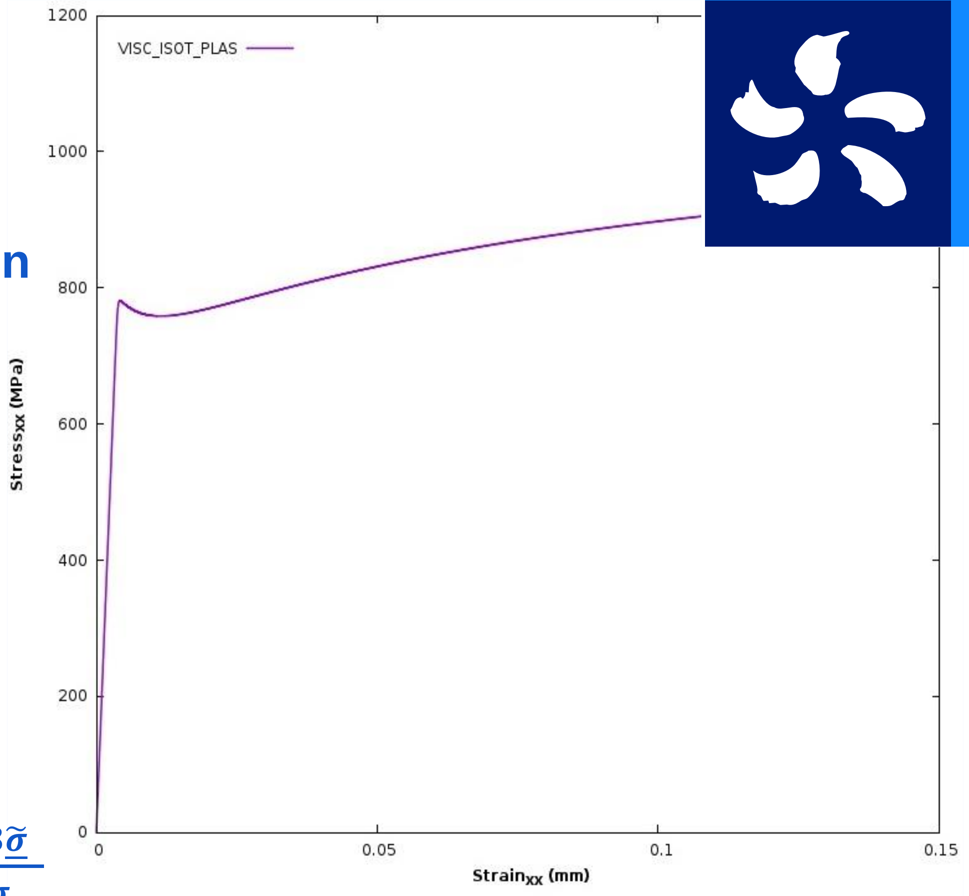
$$\underline{\sigma} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \underline{\varepsilon}^{el}$$

- **Von Mises stress criterion with $\underline{\tilde{\sigma}} := \left(\underline{\underline{I}} - \frac{1}{3}\underline{I}\otimes\underline{I}\right) : \underline{\sigma}$**

$$\sigma_{eq} = \sqrt{\frac{3}{2}\,\underline{\tilde{\sigma}} : \underline{\tilde{\sigma}}}$$

- **Associative viscoplastic flow rule with $\underline{n} := \frac{\partial \sigma_{eq}}{\partial \underline{\sigma}} = \frac{3\underline{\tilde{\sigma}}}{2\sigma_{eq}}$**

$$\underline{\dot{\varepsilon}}^{vp} = \dot{p}\,\underline{n}$$



Remark: in the uniaxial case, $\sigma_{eq} = \sigma = E\varepsilon^{el}$ and $\varepsilon^{vp} = p$.

2.1 Constitutive equations

- **Temperature $T(°C)$ is used and assumed constant**
- **Infinitesimal deformations and strain decomposition**

$$\underline{\varepsilon}^{to} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{vp}$$

- **Hooke's law (E,ν) with shear modulus $\mu := \frac{E(T(°C))}{2(1+\nu)}$**

$$\underline{\sigma} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \underline{\varepsilon}^{el}$$

- **Von Mises stress criterion with $\underline{\tilde{\sigma}} := \left(\underline{\underline{I}} - \frac{1}{3}\underline{I}\otimes\underline{I}\right) : \underline{\sigma}$**

$$\sigma_{eq} = \sqrt{\frac{3}{2}\underline{\tilde{\sigma}}:\underline{\tilde{\sigma}}}$$

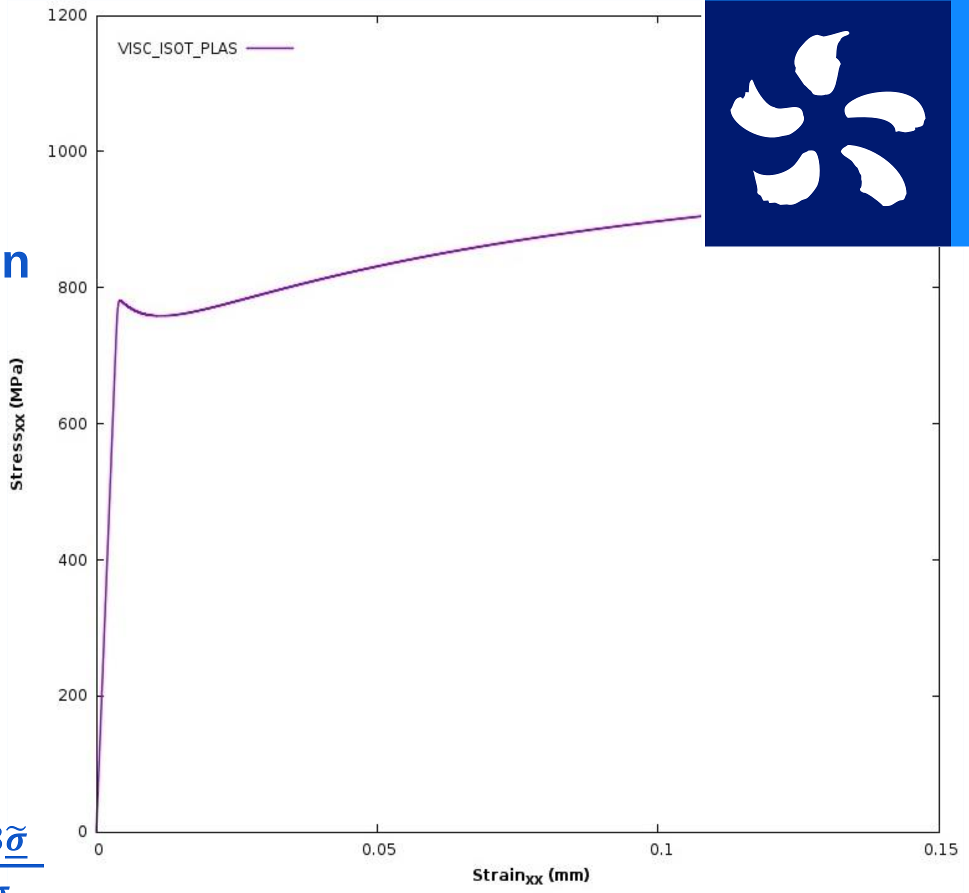
- **Associative viscoplastic flow rule with $\underline{n} := \frac{\partial\sigma_{eq}}{\partial\underline{\sigma}} = \frac{3\underline{\tilde{\sigma}}}{2\sigma_{eq}}$**

$$\underline{\dot{\varepsilon}}^{vp} = \dot{p}\,\underline{n}$$

- **Rate-dependent isotropic hardening rule**

$$\sigma_{eq} < R(p,\dot{p}) \implies \dot{p} = 0$$

$$\dot{p} > 0 \implies \sigma_{eq} = R(p,\dot{p})$$



<u>Remark:</u> in the uniaxial case, $\sigma_{eq} = \sigma = E\varepsilon^{el}$ and $\varepsilon^{vp} = p$.

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_{D}(\dot{p}) + \sigma_{E}(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \boldsymbol{\sigma_{HP}} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution

$$\boldsymbol{\sigma_{HP}} \coloneqq \mu \sqrt{\frac{3.3090\,10^{-11}\,LengthUnit}{TAILLE\_GRAIN}}$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \boldsymbol{\sigma_{VD}} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing

$$\boldsymbol{\sigma_{VD}} \coloneqq (2.5\ 10^7\ StressUnit)\min\left(\left\langle \frac{T(°C)}{100\ °C} - 2\right\rangle_+, 1\right)$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \boldsymbol{\sigma_{VS}(p)} + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing

$$\boldsymbol{\sigma_{VS}(p)} \coloneqq (5.0\ 10^7\ StressUnit) \left\langle 1 - \frac{T(°C)}{250\ °C} \right\rangle_+ e^{-200\ \boldsymbol{p}}$$

<u>Remark:</u> this term is responsible for softening at the elasto-viscoplastic transition.

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \boldsymbol{\sigma_D}(\dot{\boldsymbol{p}}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing
- Jog drag resistance

$$\boldsymbol{\sigma_D}(\dot{\boldsymbol{p}}) \coloneqq (5.0\ 10^6\ StressUnit)\left(\frac{\dot{\boldsymbol{p}}\left(\boldsymbol{s^{-1}}\right)}{10^{-6}\ s^{-1}}\right)^{1/10}$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \boldsymbol{\sigma_E}(\dot{\boldsymbol{p}}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing
- Jog drag resistance
- Lattice friction

$$\boldsymbol{\sigma_E}(\dot{\boldsymbol{p}}) \coloneqq (9.0\ 10^8\ StressUnit)\left(\left\langle 1 - \frac{T(°C) + 273.15\ °C}{9744.78\ °C}\left\langle 19.3289 - \ln\left(\frac{\dot{\boldsymbol{p}}\left(\boldsymbol{s^{-1}}\right)}{1\ s^{-1}}\right)\right\rangle_+\right\rangle_+\right)^2$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_{D}(\dot{p}) + \sigma_{E}(\dot{p}) + \sqrt{(\boldsymbol{\sigma_I})^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing
- Jog drag resistance
- Lattice friction
- Irradiation hardening

$$\boldsymbol{\sigma_I} \coloneqq \mu \sqrt{3.35\ 10^{-8} \left(\frac{10^9\ TAILLE\_AMAS}{LengthUnit}\right)^{2.3} \left(10^{-22}\ (LenghtUnit)^3\ C\_AMAS\right)^{\frac{8}{7}}}$$

provided that $(LenghtUnit)^3\ C\_AMAS > 10^{20}$ and $\boldsymbol{\sigma_I} \coloneqq 0$ otherwise.

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\boldsymbol{\sigma_{auto}}(\boldsymbol{p})\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing
- Jog drag resistance
- Lattice friction
- Irradiation hardening
- Self-interaction

$$\boldsymbol{\sigma_{auto}}(\boldsymbol{p}) \coloneqq \mu \sqrt{1.5376\ 10^{-20}\ (LengthUnit)^2\ D\_DISLOC\ e^{-3f\boldsymbol{p}} + \frac{1}{4Lf}(4 - e^{-3f\boldsymbol{p}} - 3e^{-\boldsymbol{p}})}$$

with annihilation distance $f \coloneqq \min\left(1.178 + 1.9\frac{T(°C)+273.15\ °C}{100\ °C}, 6.8\right)$ normalized by the Burgers vector $b \coloneqq 2.48\ 10^{-10}\ LengthUnit$

and normalized free mean distance defined by $L \coloneqq \left(\frac{1}{4500} + 6.7925\ 10^{-6}\sqrt{10^{-22}\ (LenghtUnit)^3\ C\_AMAS}\right)^{-1}$.

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\boldsymbol{\sigma_{LT}}(\boldsymbol{p},\dot{\boldsymbol{p}})\right)^2}$$

- Hall-Petch contribution
- Dynamic ageing
- Static ageing
- Jog drag resistance
- Lattice friction
- Irradiation hardening
- Self-interaction
- Forest hardening

$$\boldsymbol{\sigma_{LT}}(\boldsymbol{p},\dot{\boldsymbol{p}}) \coloneqq \left\langle \sqrt{3}\sigma_{auto}(\boldsymbol{p}) - \sigma_E(\dot{\boldsymbol{p}}) \right\rangle_+$$

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\sigma_{auto}(p)\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Monnet Ghiath, Analytical flow equation for irradiated low-alloy steels established by multiscale modeling. *Journal of Nuclear Materials*, 586(2023).

2.2 Modeling: description of the flow stress components

Decomposition of the hardening rule into several flow stress components

$$R(p,\dot{p}) = \sigma_{HP} + \sigma_{VD} + \sigma_{VS}(p) + \sigma_D(\dot{p}) + \sigma_E(\dot{p}) + \sqrt{(\sigma_I)^2 + \left(\boldsymbol{\sigma_{auto}}(\boldsymbol{p})\right)^2 + \left(\sigma_{LT}(p,\dot{p})\right)^2}$$

- Monnet, G., Analytical flow equation for irradiated low-alloy steels established by multiscale modeling. *Journal of Nuclear Materials*, 586(2023).

Coming back to the evolution of self-interaction $\boldsymbol{\sigma_{auto}}(\boldsymbol{p}) \coloneqq \mu\sqrt{\frac{b^2\boldsymbol{\rho}(\boldsymbol{p})}{4}}$

Apart from static ageing, only dislocation density $\boldsymbol{\rho}$ evolves with the equivalent plastic strain $\boldsymbol{p}(t) = \int_0^t \sqrt{\frac{2}{3}\underline{\dot{\varepsilon}}^{vp}(\tau):\underline{\dot{\varepsilon}}^{vp}(\tau)}\,\mathrm{d}\tau$

Two independant populations evolves: jog density ρ_j and stored dislocation density ρ_s

$$\begin{cases} \dfrac{\mathrm{d}\rho}{\mathrm{d}p} = \dfrac{\mathrm{d}\rho_j}{\mathrm{d}p} + \dfrac{\mathrm{d}\rho_s}{\mathrm{d}p}\,, & \rho(0) = \rho_0 \\ \dfrac{\mathrm{d}\rho_j}{\mathrm{d}p} = \dfrac{3}{b^2L} - \rho_j\,, & \rho_j(0) = 0 \\ \dfrac{\mathrm{d}\rho_s}{\mathrm{d}p} = \dfrac{3}{b^2L} - 3f\rho_s\,, & \rho_s(0) = \rho_0 \end{cases}$$

which gives $\boldsymbol{\rho}(\boldsymbol{p}) = \rho_0 + \frac{1}{b^2Lf}(4 - e^{-3\boldsymbol{p}} - 3e^{-\boldsymbol{p}})$.

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{\coloneqq \sigma_C(p,\dot{p})}$$

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{\coloneqq \sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1})\left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p})\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10}$$

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{:=\,\sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1})\left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p})\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. @DSL IsotropicMisesCreep; / which treats the case* $\dot{p} = f\left(\sigma_{eq}\right)$ **/*

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{:=\,\sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1}) \left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p})\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. *@DSL IsotropicMisesCreep; /* which treats the case* $\dot{p} = f\left(\sigma_{eq}\right)$ **/*

2. *@DSL IsotropicStrainHardeningMisesCreep; /* which treats the case* $\dot{p} = f\left(\sigma_{eq}, p\right)$ **/*

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{\coloneqq \sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1}) \left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p})\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. *@DSL IsotropicMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq})$ **/*
2. *@DSL IsotropicStrainHardeningMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq}, p)$ **/*
3. *@DSL IsotropicPlasticMisesFlow; /* which treats the case* $f(\sigma_{eq}, p) \leq 0$ *with* $\dot{p} \geq 0$ *and* $\dot{p}\ f(\sigma_{eq}, p) = 0$ **/*

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{:= \sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1}) \left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p}) \rangle_+}{5.0\ 10^6\ StressUnit} \right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. *@DSL IsotropicMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq})$ */
2. *@DSL IsotropicStrainHardeningMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq}, p)$ */
3. *@DSL IsotropicPlasticMisesFlow; /* which treats the case* $f(\sigma_{eq}, p) \leq 0$ *with* $\dot{p} \geq 0$ *and* $\dot{p}\ f(\sigma_{eq}, p) = 0$ */
4. *@DSL ImplicitII; /* No need to predefine* $\underline{\varepsilon^{el}}$ as the first internal state variable since p is the only unkown here */

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{:=\sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\ s^{-1})\left(\frac{\langle\sigma_{eq} - \sigma_C(p,\dot{p})\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. *@DSL IsotropicMisesCreep; /* which treats the case* $\dot{p} = f\left(\sigma_{eq}\right)$ **/*

2. *@DSL IsotropicStrainHardeningMisesCreep; /* which treats the case* $\dot{p} = f\left(\sigma_{eq}, p\right)$ **/*

3. *@DSL IsotropicPlasticMisesFlow; /* which treats the case* $f\left(\sigma_{eq}, p\right) \leq 0$ *with* $\dot{p} \geq 0$ *and* $\dot{p}\ f\left(\sigma_{eq}, p\right) = 0$ **/*

4. ***@DSL ImplicitII;** /* No need to predefine* $\underline{\varepsilon^{el}}$ as the first internal state variable since p is the only unkown here */

3.1 MFront implementation

Isolate $\boldsymbol{\sigma_D(p,\dot{p})}$ from the hardening rule to define a *pseudo*-yield surface $\boldsymbol{\sigma_C(p,\dot{p})}$

$$\sigma_D(\dot{p}) = \sigma_{eq} - \underbrace{\left(R(p,\dot{p}) - \sigma_D(\dot{p})\right)}_{\coloneqq \sigma_C(p,\dot{p})}$$

Invert the relation to get back to a Perzyna-like formulation

$$\dot{p}(s^{-1}) = (10^{-6}\, s^{-1}) \left(\frac{\langle \sigma_{eq} - \sigma_C(p,\dot{p}) \rangle_+}{5.0\ 10^6\ StressUnit} \right)^{10}$$

In addition to the *StandardElastoViscoPlasticity brick, we can choose among Domain Specific Languages (DSL):*

1. *@DSL IsotropicMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq})$ */
2. *@DSL IsotropicStrainHardeningMisesCreep; /* which treats the case* $\dot{p} = f(\sigma_{eq}, p)$ */
3. *@DSL IsotropicPlasticMisesFlow; /* which treats the case* $f(\sigma_{eq}, p) \leq 0$ *with* $\dot{p} \geq 0$ *and* $\dot{p}\ f(\sigma_{eq}, p) = 0$ */
4. ***@DSL ImplicitII;*** */* No need to predefine* $\underline{\varepsilon}^{el}$ as the first internal state variable since p is the only unkown here */

What I should have done: write $\boldsymbol{\dot{p} \approx \frac{p - p_{old}}{\Delta t(s^{-1})}}$ thanks to a local variable $\boldsymbol{p_{old}}$ and use case 2.

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;
```

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                  where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                  in Implicit DSL */
```

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                 where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                 in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                 where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                 in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                 where it is | Δp^{iter+1} – Δp^{iter} | < ϵ
                 in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments
2. Make an elastic prediction σ_{eq}^{pred} and $\underline{n}^{pred}$ by assuming
$$\Delta\underline{\varepsilon}^{el} = \Delta\underline{\varepsilon}^{to}$$

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                  where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                  in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments
2. Make an elastic prediction σ_{eq}^{pred} and $\underline{n}^{pred}$ by assuming
$$\Delta\underline{\varepsilon}^{el} = \Delta\underline{\varepsilon}^{to}$$
3. Correct it with the plastic strain increment if needed
$$\Delta\underline{\varepsilon}^{vp} = \int_t^{t+\Delta t} \dot{p}(\tau)\,\underline{n}(\tau)\,\mathrm{d}\tau \approx \Delta p\,\underline{n}(t+\theta\Delta t)$$

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                 where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                 in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments
2. Make an elastic prediction σ_{eq}^{pred} and $\underline{n}^{pred}$ by assuming
$$\Delta\underline{\varepsilon}^{el} = \Delta\underline{\varepsilon}^{to}$$
3. Correct it with the plastic strain increment if needed
$$\Delta\underline{\varepsilon}^{vp} = \int_{t}^{t+\Delta t} \dot{p}(\tau)\,\underline{n}(\tau)\,\mathrm{d}\tau \approx \Delta p\,\underline{n}(t+\theta\Delta t)$$
4. Observe $\sigma_{eq}(t+\theta\Delta t) = \sigma_{eq}^{pred} - 3\mu\theta\Delta p$ and $\underline{n}(t+\theta\Delta t) = \underline{n}^{pred}$ in order to solve
$$\boldsymbol{\Delta p} - \Delta t(s^{-1})(10^{-6}\,s^{-1})\left(\frac{\left\langle \sigma_{eq}^{pred} - 3\mu\theta\boldsymbol{\Delta p} - \sigma_C\left(p+\theta\boldsymbol{\Delta p}, \frac{\boldsymbol{\Delta p}}{\Delta t(s^{-1})}\right)\right\rangle_+}{5.0\ 10^6\ StressUnit}\right)^{10} = 0$$

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                  where it is | Δp^{iter+1} − Δp^{iter} | < ϵ
                  in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments
2. Make an elastic prediction σ_{eq}^{pred} and $\underline{n}^{pred}$ by assuming
$$\Delta\underline{\varepsilon}^{el} = \Delta\underline{\varepsilon}^{to}$$
3. Correct it with the plastic strain increment if needed
$$\Delta\underline{\varepsilon}^{vp} = \int_t^{t+\Delta t} \dot{p}(\tau)\,\underline{n}\,(\tau)\,\mathrm{d}\tau \approx \Delta p\,\underline{n}(t+\theta\Delta t)$$
4. Observe $\sigma_{eq}(t+\theta\Delta t) = \sigma_{eq}^{pred} - 3\mu\theta\Delta p$ and $\underline{n}(t+\theta\Delta t) = \underline{n}^{pred}$ in order to solve
$$\boldsymbol{\Delta p} - \Delta t(s^{-1})(10^{-6}\,s^{-1})\left(\frac{\left\langle \sigma_{eq}^{pred} - 3\mu\theta\boldsymbol{\Delta p} - \sigma_C\left(p+\theta\boldsymbol{\Delta p}, \frac{\boldsymbol{\Delta p}}{\Delta t(s^{-1})}\right)\right\rangle_+}{5.0\;10^6\;StressUnit}\right)^{10} = 0$$
5. Update the stress increment from Hooke's law
$$\boldsymbol{\Delta\underline{\sigma}} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \left(\Delta\underline{\varepsilon}^{to} - \boldsymbol{\Delta p}\,\underline{n}^{pred}\right)$$

3.1 MFront implementation

VISC_ISOT_PLAS.mfront

```
@DSL ImplicitII;
@Behaviour VISC_ISOT_PLAS;
@Algorithm NewtonRaphson; /* Analytical Jacobian */
@UseQt true;

@Theta 1.0; /* Full implicit scheme */
@Epsilon 5.0e-14;
@IterMax 100; /* Warning: stop criteria is | f(Δp) | < ϵ
                 where it is | Δp^{iter+1} – Δp^{iter} | < ϵ
                 in Implicit DSL */

@StateVariable strain p;
p.setEntryName("EPSPEQ");
@AuxiliaryStateVariable real INDIPLAS;
@AuxiliaryStateVariable StrainStensor eel_old;
eel_old.setGlossaryName("ElasticStrain");
```

MFront strategy of resolution

1. Initialize local variables, affine approximation of increments
2. Make an elastic prediction σ_{eq}^{pred} and $\underline{n}^{pred}$ by assuming
$$\Delta\underline{\varepsilon}^{el} = \Delta\underline{\varepsilon}^{to}$$
3. Correct it with the plastic strain increment if needed
$$\Delta\underline{\varepsilon}^{vp} = \int_t^{t+\Delta t} \dot{p}(\tau)\,\underline{n}(\tau)\,\mathrm{d}\tau \approx \Delta p\,\underline{n}(t+\theta\Delta t)$$
4. Observe $\sigma_{eq}(t+\theta\Delta t) = \sigma_{eq}^{pred} - 3\mu\theta\Delta p$ and $\underline{n}(t+\theta\Delta t) = \underline{n}^{pred}$ in order to solve
$$\boldsymbol{\Delta p} - \Delta t(s^{-1})(10^{-6}\,s^{-1})\left(\frac{\left\langle \sigma_{eq}^{pred} - 3\mu\theta\boldsymbol{\Delta p} - \sigma_C\left(p+\theta\boldsymbol{\Delta p}, \frac{\boldsymbol{\Delta p}}{\Delta t(s^{-1})}\right)\right\rangle_+}{5.0\;10^6\;StressUnit}\right)^{10} = 0$$
5. Update the stress increment from Hooke's law
$$\boldsymbol{\Delta\underline{\sigma}} = 2\mu\left(\underline{\underline{I}} + \frac{\nu}{1-2\nu}\underline{I}\otimes\underline{I}\right) : \left(\Delta\underline{\varepsilon}^{to} - \boldsymbol{\Delta p}\,\underline{n}^{pred}\right)$$
6. Compute analytically the tangent operator $\mathrm{D_t} \coloneqq \frac{\partial(\Delta\underline{\sigma})}{\partial(\Delta\underline{\varepsilon}^{to})}$ from the derivation of the above two relations

3.1 MFront implementation

Difficulties encountered solving $F(\Delta p) = 0$

- Protect algebraic quantities (division by zero, undefined function value, infinite slope, ...) to avoid SIGFPE signal
- Multi-scale modeling: $F(\Delta p, \ln(\Delta p), e^{\Delta p})$ very sensitive to Δp

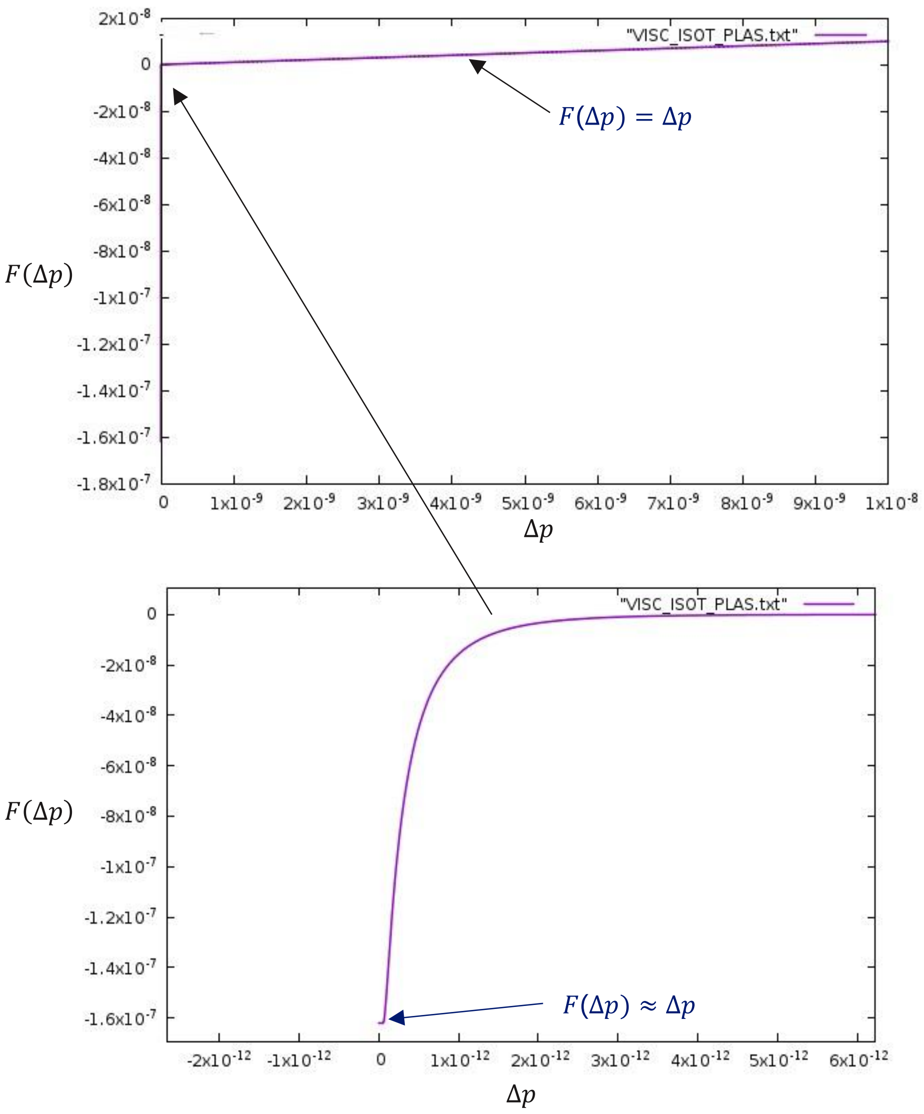
<u>Debug MFront tips</u>

```
@Includes {
    #include <cfenv>
    #include "TFEL/Math/General/floating_point_exceptions.hxx"
}

@InitLocalVariables {
    /* To let Code_Aster handles signals instead of MFront via MGIS */
    fedisableexcept(FE_DIVBYZERO);
    fedisableexcept(FE_UNDERFLOW);
    fedisableexcept(FE_OVERFLOW);
    fedisableexcept(FE_INVALID);
    fedisableexcept(FE_INEXACT);
}

@Predictor{
    If (bloading){
        std::cout<<"starting with dp="<<dp<<" and sigC="<<sigC;
        std::cout<<" and p="<<p<<" and seq_elas="<<seq_elas<<std::endl;
    }
}

@Integrator{
    if (iter>iterMax-5) {
        std::cout<<dp=<<dp<<...<<std::endl;
    }
}
```

3.1 MFront implementation

Difficulties encountered solving $F(\Delta p) = 0$

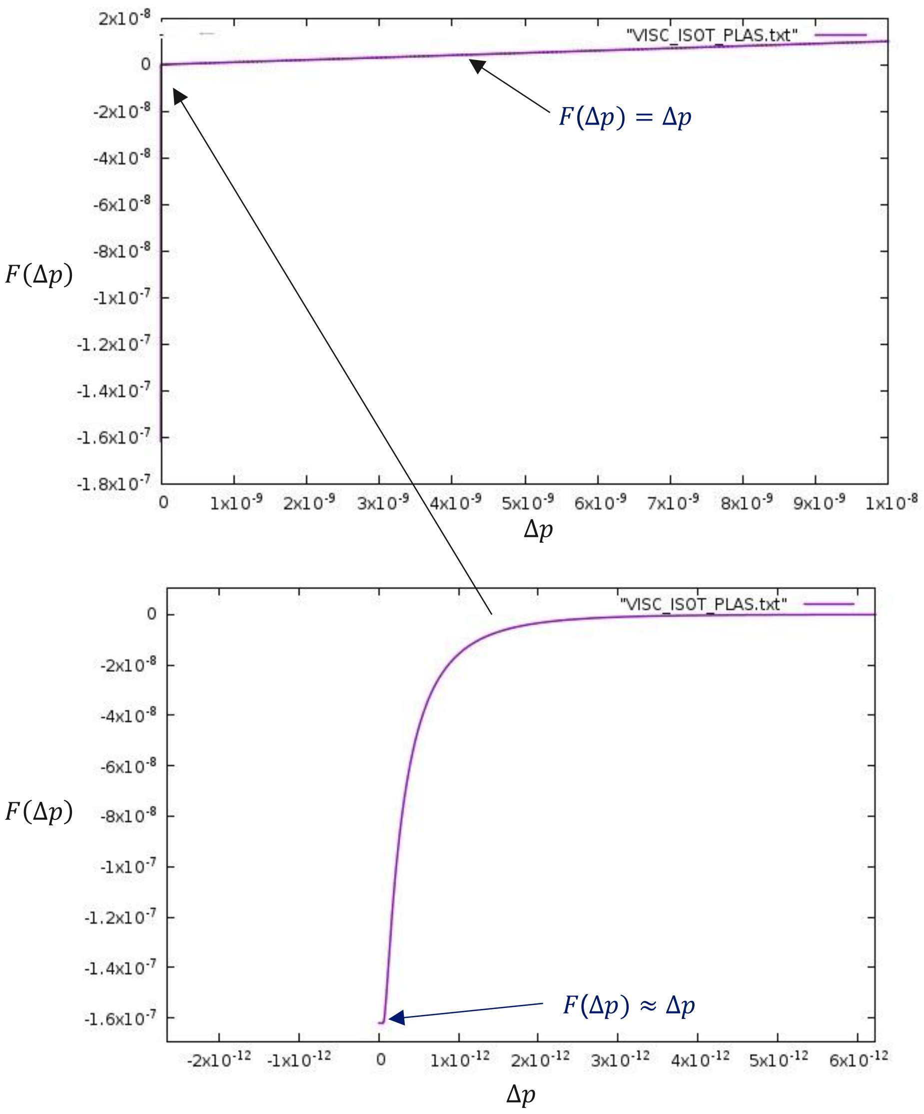
- Protect algebraic quantities (division by zero, undefined function value, infinite slope, ...) to avoid SIGFPE signal
- Multi-scale modeling: $F(\Delta p, \ln(\Delta p), e^{\Delta p})$ very sensitive to Δp
- Starting point $\Delta p = 0$ yields no convergence and hard to setup when viscoplasticity starts
- Stable interval needed for Newton method (dichotomy)



3.1 MFront implementation

Difficulties encountered solving $F(\Delta p) = 0$

- Protect algebraic quantities (division by zero, undefined function value, infinite slope, ...) to avoid SIGFPE signal
- Multi-scale modeling: $F(\Delta p, \ln(\Delta p), e^{\Delta p})$ very sensitive to Δp
- Starting point $\Delta p = 0$ yields no convergence and hard to setup when viscoplasticity starts
- Stable interval needed for Newton method (dichotomy)
- Stop criteria $|F(\Delta p)| < \epsilon$ reaches precision machine and oscillations found obstructing convergence.
- Control and reduce the interval during Newton's method



Conclusion

Closed form equation for flow stress of irradiated low-alloy steels

Expression involving microstructure features

Temperature and strain rate effects *naturally* accounted for

NEW (2025): MFront law VISC_ISOT_PLAS incorporated in Code_Aster via MGIS.

Thank you

Questions ?