NEML: Nuclear Engineering Material Library

Mark Messner

June 13, 2017

Chapter 1

Conventions

1.1 Mechanics

We use the Mandel notation to convert symmetric second and fourth order tensors to vectors and matrices. The convention transforms the second order tensor

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \rightarrow \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sqrt{2}\sigma_{23} & \sqrt{2}\sigma_{13} & \sqrt{2}\sigma_{12} \end{bmatrix} \tag{1.1}$$

and, after transformation, a fourth order tensor $\mathbf{C}$ becomes

$$\begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & \sqrt{2}C_{1123} & \sqrt{2}C_{1113} & \sqrt{2}C_{1112} \\ C_{1122} & C_{2222} & C_{2233} & \sqrt{2}C_{2223} & \sqrt{2}C_{2213} & \sqrt{2}C_{2212} \\ C_{1133} & C_{2233} & C_{3333} & \sqrt{2}C_{3323} & \sqrt{2}C_{3313} & \sqrt{2}C_{3312} \\ \sqrt{2}C_{1123} & \sqrt{2}C_{2223} & \sqrt{2}C_{3323} & 2C_{2323} & 2C_{2313} & 2C_{2312} \\ \sqrt{2}C_{1113} & \sqrt{2}C_{2213} & \sqrt{2}C_{3313} & 2C_{2313} & 2C_{1313} & 2C_{1312} \\ \sqrt{2}C_{1112} & \sqrt{2}C_{2212} & \sqrt{2}C_{3312} & 2C_{2312} & 2C_{1312} & 2C_{1212} \end{bmatrix}. \tag{1.2}$$

For symmetric two second order tensors $\mathbf{a}$ and $\mathbf{b}$ and their Mandel vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ the relation

$$\mathbf{a} : \mathbf{b} = \hat{\mathbf{a}} \cdot \hat{\mathbf{b}} \tag{1.3}$$

expresses the utility of this convention. Similarly, given the symmetric fourth order tensor $\mathbf{C}$ and its equivalent Mandel matrix $\hat{\mathbf{C}}$ contraction over two adjacent indices

$$\mathbf{a} = \mathbf{C} : \mathbf{b} \tag{1.4}$$

simply becomes matrix-vector multiplication

$$\hat{\mathbf{a}} = \hat{\mathbf{C}} \cdot \hat{\mathbf{b}}.$$

1.2 Interfaces

NEML supports three interfaces into the material library.

1. update_ldF: Large deformations with kinematics expressed directly through the deformation gradient $\mathbf{F}$.
2. update_ldI: Incremental large deformations with kinematics expressed through the spatial velocity gradient $\mathbf{l}$.
3. update_sd: Incremental small deformations with kinematics expressed through the small strain rate $\dot{\boldsymbol{\varepsilon}}$.

A user defining a new material model selects one of these four interfaces when implementing a new material model. NEML takes the interface the user elected to implement and automatically translates the implemented interfaces to conform to the remaining two. The translation from interfaces 1 or 2 to interface 3 does not fundamentally change the implemented material model – it simply applies the assumptions of small deformation kinematics. However, the conversion from interface 3 to interface 1 or 2 assumes a hypoelastic formulation using the Jaumann objective stress rate. This can fundamentally and detrimentally alter the material constitutive response [].

The library also provides several non-implementable interfaces:

1. update_warp3d: Interface linking the material library to the warp3d finite element code.
2. update_umat_incremental: Interface for linking the material library to ABAQUS Implicit via the UMAT interface and using the incremental strains.

These interfaces will introduce similar errors as the small deformation to large deformation translation step described above.

Chapter 2

Material models

2.1 Yield surfaces

2.1.1 Von Mises with isotropic hardening

Internal variables are $\mathbf{q} = [\sigma_F] \boldsymbol{\alpha} = [\bar{\varepsilon}_p]$

$$f = J_2(\boldsymbol{\sigma}) + \sqrt{\frac{2}{3}} \sigma_F(\bar{\varepsilon}_p, T) = \|\mathbf{s}\| + \sqrt{\frac{2}{3}} \sigma_F$$

with $\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3} \operatorname{tr} \boldsymbol{\sigma} \mathbf{I}$for various yield strength functions.

The derivatives are:

$$\begin{aligned} \frac{\partial f}{\partial \boldsymbol{\sigma}} &= \frac{\mathbf{s}}{\|\mathbf{s}\|} \\ \frac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} &= \frac{1}{\|\mathbf{s}\|} \left(\mathbf{1} - \frac{1}{3} (\mathbf{I} \otimes \mathbf{I}) - \frac{\mathbf{s}}{\|\mathbf{s}\|} \otimes \frac{\mathbf{s}}{\|\mathbf{s}\|} \right) \\ \frac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial \mathbf{q}} &= \mathbf{0} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{q}} &= \sqrt{\frac{2}{3}} \\ \frac{\partial^2 f}{\partial \mathbf{q} \partial \boldsymbol{q}} &= 0 \\ \frac{\partial^2 f}{\partial \mathbf{q} \partial \boldsymbol{\sigma}} &= \mathbf{0} \end{aligned}$$

2.1.2 Von Mises with isotropic and kinematic hardening

Internal variables are $\mathbf{q} = \left[\begin{array}{cc} \sigma_F & \mathbf{X} \end{array} \right] \boldsymbol{\alpha} = \left[\begin{array}{cc} \bar{\varepsilon}_p & \boldsymbol{\varepsilon}_p \end{array} \right]$

$$f = J_2\left(\boldsymbol{\sigma} + \mathbf{X}\right) + \sqrt{\frac{2}{3}}\sigma_F\left(\bar{\varepsilon}_p, T\right) = \left\|\mathbf{s} + \mathbf{X}\right\| + \sqrt{\frac{2}{3}}\sigma_F.$$

The derivatives are:

$$\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{\sigma}} &= \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \\
\frac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} &= \frac{1}{\left\|\mathbf{s}+\mathbf{X}\right\|}\left(\mathbf{1} - \frac{1}{3}\left(\mathbf{I} \otimes \mathbf{I}\right) - \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \otimes \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|}\right) \\
\frac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial \mathbf{q}} &= \left[\frac{1}{\left\|\mathbf{s}+\mathbf{X}\right\|}\left(\mathbf{1} - \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \otimes \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|}\right)\right]
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{q}} &= \begin{bmatrix} \sqrt{\frac{2}{3}} & \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \end{bmatrix} \\
\frac{\partial^2 f}{\partial \mathbf{q} \partial \boldsymbol{q}} &= \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \frac{1}{\left\|\mathbf{s}+\mathbf{X}\right\|}\left(\mathbf{1} - \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \otimes \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|}\right) \end{bmatrix} \\
\frac{\partial^2 f}{\partial \mathbf{q} \partial \boldsymbol{\sigma}} &= \begin{bmatrix} 0 \\ \frac{1}{\left\|\mathbf{s}+\mathbf{X}\right\|}\left(\mathbf{1} - \frac{1}{3}\left(\mathbf{I} \otimes \mathbf{I}\right) - \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|} \otimes \frac{\mathbf{s}+\mathbf{X}}{\left\|\mathbf{s}+\mathbf{X}\right\|}\right) \end{bmatrix}
\end{aligned}$$

2.2 Hardening Rules

2.2.1 Associative

2.2.1.1 Isotropic

Linear

$$\sigma_F\left(\bar{\varepsilon}_p, T\right) = -\left(\sigma_0 + K\bar{\varepsilon}_p\right)$$

$$\frac{\partial \sigma_F}{\partial \bar{\varepsilon}_p} = -K$$

Voce

$$\sigma_F\left(\bar{\varepsilon}_p, T\right) = -\left[\sigma_0 + R\left(1 - \exp\left(-\delta\bar{\varepsilon}_p\right)\right)\right]$$

$$\frac{\partial \sigma_F}{\partial \bar{\varepsilon}_p} = -\delta R \exp\left(-\delta\bar{\varepsilon}_p\right)$$

2.2.1.2 Kinematic

Linear

$$\mathbf{X}\left(\boldsymbol{\varepsilon}_p, T\right) = -H\boldsymbol{\varepsilon}_p$$

$$\frac{d\mathbf{X}}{d\boldsymbol{\varepsilon}_p} = -H\mathbf{I}$$

2.2.2 Nonassociative

2.2.2.1 Frederick-Armstrong & Chaboche

A traditional model of this type uses an associative flow rule for the plastic strain, with a yield surfaces expecting an isotropic hardening parameter and a backstress. The isotropic hardening rule can be selected from Section 2.2.1.1 while the backstress follows the formulation:

$$\begin{aligned} \mathbf{X} &= \sum_{i=1}^{n} \mathbf{X}_i \\ \dot{\mathbf{X}}_i &= -\left(\frac{2}{3}C_i\mathbf{n} + \sqrt{\frac{2}{3}}\gamma_i\left(\bar{\varepsilon}_p\right)\mathbf{X}_i\right)\dot{\gamma} - A_i\sqrt{\frac{3}{2}}\left\Vert\mathbf{X}_i\right\Vert^{a_i-1}\mathbf{X}_i \end{aligned}$$

A Frederick-Armstrong model has $n = 1$.

So far the library has two forms for $\gamma\left(\bar{\varepsilon}_p\right)$: constant and a form proposed by Chaboche in his book:

$$\gamma\left(\bar{\varepsilon}_p\right) = \gamma_s + \left(\gamma_0 - \gamma_s\right)e^{-\beta\bar{\varepsilon}_p}.$$

2.3 Viscoplastic models

2.3.1 Perzyna

This is implemented as associative viscoplasticity. The user provides a yield surface and a hardening rule from the list of associative models above. The formulation then uses the associative flow and hardening rules and the rate function:

$$y\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) = \left\langle \frac{g\left(f\left(\boldsymbol{\sigma}, \mathbf{q}\left(\boldsymbol{\alpha}\right)\right)\right)}{\eta} \right\rangle$$

where g is some monotonic function. The library currently only provides a power law for g:

$$g\left(f\right) = f^n.$$

2.3.2 Chaboche

The Chaboche viscoplastic model, as described in [], uses the flow rule associated to surface $f\left(\boldsymbol{\sigma}, \boldsymbol{q}\right)$, a Chaboche backstress, described above, Voce isotropic hardening, and the rate rule:

$$y\left(\boldsymbol{\sigma}, \mathbf{q}\right) = \sqrt{\frac{3}{2}}\left\langle \frac{f\left(\boldsymbol{\sigma}, \boldsymbol{q}\right)}{\sqrt{2/3}\eta} \right\rangle^n.$$

The NEML implementation allows for an arbitrary backstress and hardening rule.

2.3.3 Yaguchi & Takahashi

This model is described by the equations:

$$\dot{\boldsymbol{\varepsilon}}_p = \mathbf{n}\dot{p}$$

$$\mathbf{n} = \frac{3}{2}\frac{\boldsymbol{\sigma}' - \mathbf{X}'}{J_2\left(\boldsymbol{\sigma}' - \mathbf{X}'\right)}$$

$$J_2\left(\mathbf{Y}'\right) = \sqrt{\frac{3}{2}\mathbf{Y}' : \mathbf{Y}'}$$

$$\dot{p} = \left\langle \frac{J_2\left(\boldsymbol{\sigma}' - \mathbf{X}'\right) - \sigma_a}{D} \right\rangle^n$$

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$$

$$\dot{\mathbf{X}}_1 = C_1\left(\frac{2}{3}\left(a_{10} - Q\right)\mathbf{n} - \mathbf{X}_1\right)\dot{p} - \gamma_1 J_2\left(\mathbf{X}_1\right)^{m-1}\mathbf{X}_1$$

$$\dot{\mathbf{X}}_2 = C_2\left(\frac{2}{3}a_2\mathbf{n} - \mathbf{X}_2\right)\dot{p} - \gamma_2 J_2\left(\mathbf{X}_2\right)^{m-1}\mathbf{X}_2$$

$$\dot{Q} = d\left(q - Q\right)\dot{p}$$

$$\begin{aligned}
\dot{\sigma}_a &= b\left(\sigma_{as} - \sigma_a\right)\dot{p} \\
b &= \begin{cases} b_h & \sigma_{as} - \sigma_a \geq 0 \\ b_r & \sigma_{as} - \sigma_a < 0 \end{cases} \\
\sigma_{as} &= \left\langle A + B\log_{10}\dot{p}\right\rangle
\end{aligned}$$

The authors provided interpolation schemes for the model parameters in the range $473\,\mathrm{K} < T < 873\,\mathrm{K}$. These parameters are hard-coded in the implementation, so this model takes no parameters.

2.4 Creep models

In general these models have the form

$$\dot{\boldsymbol{\varepsilon}}^{cr} = \boldsymbol{f}\left(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}^{cr}, t, T\right)$$

so we need to keep in mind the explicit time dependence. The idea is to add the integrated creep strain to a rate-independent plasticity model for a classical “non-unified” approach. We solve the creep rate equation implicitly as

$$\boldsymbol{\varepsilon}^{cr}_{n+1} = \boldsymbol{\varepsilon}^{cr}_{n} + \boldsymbol{f}_{n+1}\Delta t_{n+1}$$

with the residual

$$\boldsymbol{R} = \boldsymbol{\varepsilon}^{cr}_{n+1} - \boldsymbol{\varepsilon}^{cr}_{n} - \boldsymbol{f}_{n+1}\Delta t_{n+1}$$

and the Jacobian

$$\boldsymbol{J} = I - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\varepsilon}_{n+1}^{cr}} \Delta t_{n+1}$$

which the user must provide the derivative for.

To solve the eventual outer nonlinear equation and to calculate the tangent we need the additional derivative

$$\frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}}.$$

Play our usual game of

$$\boldsymbol{R} = \boldsymbol{0} \implies 0 = d\boldsymbol{R} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\sigma}_{n+1}} : d\boldsymbol{\sigma}_{n+1} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\varepsilon}_{n+1}^{cr}} : d\boldsymbol{\varepsilon}_{n+1}^{cr} + \frac{\partial \boldsymbol{R}}{\partial t_{n+1}} : dt_{n+1}$$

$$\boldsymbol{R} = \boldsymbol{0} \implies 0 = d\boldsymbol{R} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\sigma}_{n+1}} : d\boldsymbol{\sigma}_{n+1} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\varepsilon}_{n+1}^{cr}} : d\boldsymbol{\varepsilon}_{n+1}^{cr} + \frac{\partial \boldsymbol{R}}{\partial t_{n+1}} : dt_{n+1} + \frac{\partial \boldsymbol{R}}{\partial t_{n+1}} : dT_{n+1}$$

$$\boldsymbol{0} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\sigma}_{n+1}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\varepsilon}_{n+1}^{cr}} : \frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}} + \frac{\partial \boldsymbol{R}}{\partial t_{n+1}} : \frac{dt_{n+1}}{d\boldsymbol{\sigma}_{n+1}} + \frac{\partial \boldsymbol{R}}{\partial t_{n+1}} : \frac{dT_{n+1}}{d\boldsymbol{\sigma}_{n+1}}$$

and pretending the time and temperature variation is small. Then

$$\frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}} = -\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\varepsilon}_{n+1}^{cr}}\right)^{-1} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\sigma}_{n+1}}$$

$$\frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}} = \boldsymbol{J}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\sigma}_{n+1}} \Delta t_{n+1}.$$

Therefore the whole model is defined by the function $\boldsymbol{f}$ and derivatives $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\varepsilon}^{cr}}$, $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\sigma}}$, $\frac{\partial \boldsymbol{f}}{\partial t}$, and $\frac{\partial \boldsymbol{f}}{\partial T}$.

2.4.1 J_2 creep

The above gives the specific framework, but we'll go further are assume that

$$\dot{\boldsymbol{\varepsilon}}^{cr} = g\left(\sigma_{eq}, \varepsilon_{eq}, t, T\right) \frac{3}{2} \frac{\boldsymbol{s}}{\sigma_{eq}}$$

with

$$\boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3} \operatorname{tr}\left(\boldsymbol{\sigma}\right)$$

$$\sigma_{eq} = \sqrt{\frac{3}{2} \boldsymbol{s} : \boldsymbol{s}}$$

and

$$\varepsilon_{eq} = \sqrt{\frac{2}{3} \boldsymbol{\varepsilon}^{cr} : \boldsymbol{\varepsilon}^{cr}}.$$

Then we can define the whole thing in terms of the nice scalar function g:

$$\boldsymbol{f} = g\frac{3}{2}\frac{\boldsymbol{s}}{\sigma_{eq}}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\sigma}} = \left[\frac{3}{2}\left(\frac{\partial g}{\partial \sigma_{eq}} - \frac{g}{\sigma_{eq}}\right)\left(\frac{\boldsymbol{s}}{\sigma_{eq}} \otimes \frac{\boldsymbol{s}}{\sigma_{eq}}\right) + \frac{g}{\sigma_{eq}}\boldsymbol{I}\right]\frac{3}{2}\left(\boldsymbol{I} - \frac{1}{3}\mathbf{1} \otimes \mathbf{1}\right)$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\varepsilon}^{cr}} = \left(\frac{\boldsymbol{s}}{\sigma_{eq}}\right) \otimes \left(\frac{\partial g}{\partial \varepsilon_{eq}}\frac{\boldsymbol{\varepsilon}^{cr}}{\varepsilon_{eq}}\right)$$

$$\frac{\partial \boldsymbol{f}}{\partial t} = \frac{\partial g}{\partial t}\frac{3}{2}\frac{\boldsymbol{s}}{\sigma_{eq}}$$

$$\frac{\partial \boldsymbol{f}}{\partial T} = \frac{\partial g}{\partial T}\frac{3}{2}\frac{\boldsymbol{s}}{\sigma_{eq}}$$

2.4.2 Specific creep laws

2.4.2.1 Power law

$$\begin{aligned} g = & \quad A\sigma_{eq}^{n} \\ \frac{\partial g}{\partial \sigma_{eq}} = & \quad An\sigma_{eq}^{n-1} \\ \frac{\partial g}{\partial \varepsilon_{eq}} = & \quad 0 \\ \frac{\partial g}{\partial t} = & \quad 0 \end{aligned}$$

2.4.2.2 Norton-Bailey

$$\begin{aligned} g = & \quad mA^{1/m}\sigma_{eq}^{n/m}\left(\varepsilon_{eq}\right)^{(m-1)/m} \\ \frac{\partial g}{\partial \sigma_{eq}} = & \quad nA^{1/m}\sigma_{eq}^{n/m-1}\left(\varepsilon_{eq}\right)^{(m-1)/m} \\ \frac{\partial g}{\partial \varepsilon_{eq}} = & \quad \left(m-1\right)A^{1/m}\sigma_{eq}^{n/m}\left(\varepsilon_{eq}\right)^{-1/m} \\ \frac{\partial g}{\partial t} = & \quad 0 \end{aligned}$$

Chapter 3

Algorithms

3.1 General Newton algorithm for rate-independent plasticity

For associative flow this algorithm has the interpretation of a closest point projection.

3.1.1 Continuous equations

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}}^p &= \gamma \mathbf{g}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right)\\ \dot{\boldsymbol{\alpha}} &= \gamma \mathbf{h}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right)\end{aligned}$$

and the Kuhn-Tucker and consistency conditions

$$\begin{aligned}\gamma &\geq 0\\ f\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) &\leq 0\\ \gamma f\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) &= 0\end{aligned}$$

3.1.2 Newton integration scheme

- Trial state: $\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n$,$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n$, $\boldsymbol{\sigma}_{n+1} = \mathbf{C}_{n+1} : \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_{n+1}\right)$
- Evaluate $f\left(\boldsymbol{s}_{n+1}, \boldsymbol{\alpha}_{n+1}\right)$. If less than zero return, else do Newton iteration on the scheme described here.

Equations:

$$\begin{aligned}\boldsymbol{\varepsilon}^p_{n+1} &= \boldsymbol{\varepsilon}^p_n + \mathbf{g}_{n+1}\Delta\gamma_{n+1}\\ \boldsymbol{\alpha}_{n+1} &= \boldsymbol{\alpha}_n + \mathbf{h}_{n+1}\Delta\gamma_{n+1}\\ f\left(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\alpha}_{n+1}\right) &= 0\end{aligned}$$

with $\boldsymbol{\sigma}_{n+1} = \mathbf{C}_{n+1} : \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_{n+1}\right)$.

Residual:

$$\mathbf{R} = \begin{bmatrix} -\boldsymbol{\varepsilon}^p_{n+1} + \boldsymbol{\varepsilon}^p_n + \mathbf{g}_{n+1}\Delta\gamma_{n+1} \\ -\boldsymbol{\alpha}_{n+1} + \boldsymbol{\alpha}_n + \mathbf{h}_{n+1}\Delta\gamma_{n+1} \\ f_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ R_3 \end{bmatrix}.$$

Jacobian. Helpful note is that

$$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}_{n+1} - \mathbf{C}^{-1}_{n+1} : \boldsymbol{\sigma}_{n+1}$$

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}^p_{n+1}} = -\mathbf{C}_{n+1}$$

$$\mathbf{J} = \begin{bmatrix} -\mathbf{I} - \frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1}\Delta\gamma_{n+1} & \frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}}\Delta\gamma_{n+1} & \mathbf{g}_{n+1} \\ -\frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1}\Delta\gamma_{n+1} & -\mathbf{I} + \frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}}\Delta\gamma_{n+1} & \mathbf{h}_{n+1} \\ -\frac{\partial f_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1} & \frac{\partial f_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \mathbf{J}_{13} \\ \mathbf{J}_{21} & \mathbf{J}_{22} & \mathbf{J}_{23} \\ \mathbf{J}_{31} & \mathbf{J}_{32} & J_{33} \end{bmatrix}$$

3.1.3 Algorithmic tangent

Consider the implicit function theorem applied to the residual at the final iteration:

$$\mathbf{R}\left(\boldsymbol{\varepsilon}_{n+1}, \boldsymbol{\varepsilon}^p_{n+1}, \boldsymbol{\alpha}_{n+1}, \Delta\gamma_{n+1}\right) = 0$$

Provided several conditions are met (notably a non-singular Jacobian at this point), we can solve for the derivatives $\frac{d\boldsymbol{\varepsilon}_{n+1}}{d\boldsymbol{\varepsilon}^p_{n+1}}$, $\frac{d\boldsymbol{\varepsilon}_{n+1}}{d\boldsymbol{\alpha}_{n+1}}$, and $\frac{d\boldsymbol{\varepsilon}_{n+1}}{d\Delta\gamma_{n+1}}$ in terms of the other derivatives. As it turns out, these other derivatives are parts of the Jacobian we already computed for the Newton scheme.

$$\begin{aligned} d\mathbf{R}_1 &= \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}_{n+1}} d\boldsymbol{\varepsilon}_{n+1} + \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}^p_{n+1}} d\boldsymbol{\varepsilon}^p_{n+1} + \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\alpha}_{n+1}} d\boldsymbol{\alpha}_{n+1} + \frac{\partial \mathbf{R}_1}{\partial \Delta\gamma_{n+1}} d\Delta\gamma_{n+1} = 0 \\ d\mathbf{R}_2 &= \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}_{n+1}} d\boldsymbol{\varepsilon}_{n+1} + \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}^p_{n+1}} d\boldsymbol{\varepsilon}^p_{n+1} + \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\alpha}_{n+1}} d\boldsymbol{\alpha}_{n+1} + \frac{\partial \mathbf{R}_2}{\partial \Delta\gamma_{n+1}} d\Delta\gamma_{n+1} = 0 \\ dR_3 &= \frac{\partial R_3}{\partial \boldsymbol{\varepsilon}_{n+1}} d\boldsymbol{\varepsilon}_{n+1} + \frac{\partial R_3}{\partial \boldsymbol{\varepsilon}^p_{n+1}} d\boldsymbol{\varepsilon}^p_{n+1} + \frac{\partial R_3}{\partial \boldsymbol{\alpha}_{n+1}} d\boldsymbol{\alpha}_{n+1} + \frac{\partial R_3}{\partial \Delta\gamma_{n+1}} d\Delta\gamma_{n+1} = 0 \end{aligned}$$

$$\begin{aligned} \mathbf{0} &= \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}^p_{n+1}} \frac{d\boldsymbol{\varepsilon}^p_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\alpha}_{n+1}} \frac{d\boldsymbol{\alpha}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \Delta\gamma_{n+1}} \frac{d\Delta\gamma_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \\ \mathbf{0} &= \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}^p_{n+1}} \frac{d\boldsymbol{\varepsilon}^p_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\alpha}_{n+1}} \frac{d\boldsymbol{\alpha}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \Delta\gamma_{n+1}} \frac{d\Delta\gamma_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \\ 0 &= \frac{\partial R_3}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\partial R_3}{\partial \boldsymbol{\varepsilon}^p_{n+1}} \frac{d\boldsymbol{\varepsilon}^p_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial R_3}{\partial \boldsymbol{\alpha}_{n+1}} \frac{d\boldsymbol{\alpha}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} + \frac{\partial R_3}{\partial \Delta\gamma_{n+1}} \frac{d\Delta\gamma_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \end{aligned}$$

Make some associations...

$$\mathbf{0} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{KK} & \mathbf{J}_{KE} \\ \mathbf{J}_{EK} & \mathbf{J}_{EE} \end{bmatrix} \begin{bmatrix} \mathbf{K} \\ \mathbf{E} \end{bmatrix}$$

where $\mathbf{K} = \frac{d\boldsymbol{\varepsilon}^{p}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$. Solve for $\mathbf{K}$:

$$\mathbf{K} = \left(\mathbf{J}_{KK} - \mathbf{J}_{KE}\mathbf{J}_{EE}^{-1}\mathbf{J}_{EK}\right)^{-1}\left(\mathbf{J}_{KE}\mathbf{J}_{EE}^{-1}\mathbf{B} - \mathbf{A}\right)$$

Here

$$\mathbf{A} = \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}_{n+1}} = \frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1}\Delta\gamma_{n+1}$$

and

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}_{n+1}} \\ \frac{\partial R_3}{\partial \boldsymbol{\varepsilon}_{n+1}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1}\Delta\gamma_{n+1} \\ \frac{\partial f_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} : \mathbf{C}_{n+1} \end{bmatrix}.$$

With $\frac{d\boldsymbol{\varepsilon}^{p}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$ in hand we have

$$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \mathbf{C} : \left(\mathbf{I} - \frac{d\boldsymbol{\varepsilon}^{p}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}\right)$$

3.2 Combined rate-independent plasticity and creep

Start with two functions:

$$\boldsymbol{\sigma}^{ep}_{n+1}\left(\boldsymbol{\varepsilon}^{ep}_{n+1}, \boldsymbol{\varepsilon}^{ep}_{n}, \boldsymbol{h}_n, \Delta t_{n+1}, T_{n+1}\right)$$

as defined in the previous section and

$$\boldsymbol{\varepsilon}^{cr}_{n+1}\left(\boldsymbol{\varepsilon}^{cr}_{n}, \boldsymbol{\sigma}^{cr}_{n+1}, \Delta t_{n+1}, T_{n+1}\right)$$

defined above. We want to make a combined update so that

$$\begin{aligned} \boldsymbol{\varepsilon}_{n+1} &= \boldsymbol{\varepsilon}^{ep}_{n+1} + \boldsymbol{\varepsilon}^{cr}_{n+1} \\ \boldsymbol{\sigma}^{cr}_{n+1} &= \boldsymbol{\sigma}^{ep}_{n+1}. \end{aligned}$$

Combine these two equations into the residual

$$\boldsymbol{R} = \boldsymbol{\varepsilon}^{ep}_{n+1} + \boldsymbol{\varepsilon}^{cr}_{n+1}\left(\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}^{ep}_{n}, \boldsymbol{\sigma}_{n+1}\left(\boldsymbol{\varepsilon}^{ep}_{n+1}, \boldsymbol{\varepsilon}^{ep}_{n}, \boldsymbol{h}_n, \Delta t_{n+1}, T_{n+1}\right), \Delta t_{n+1}, T_{n+1}\right) - \boldsymbol{\varepsilon}_{n+1}$$

which means we need to supplement the history variables of the elastic-plastic model with $\boldsymbol{\varepsilon}^{ep}$. We want to solve this residual equation for $\boldsymbol{\varepsilon}^{ep}_{n+1}$ using Newton's method. The Jacobian of this residual is

$$\frac{d\boldsymbol{R}}{d\boldsymbol{\varepsilon}^{ep}_{n+1}} = \boldsymbol{I} + \frac{d\boldsymbol{\varepsilon}^{cr}_{n+1}}{d\boldsymbol{\sigma}_{n+1}}\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}^{ep}_{n+1}}$$

where $\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}^{ep}}$ is the algorithmic tangent of the rate independent model and $\frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}}$ is an output of the creep model. Finally, the overall algorithmic tangent is

$$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \left(\frac{d\boldsymbol{\varepsilon}_{n+1}^{cr}}{d\boldsymbol{\sigma}_{n+1}} + \left(\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}^{ep}}\right)^{-1}\right)^{-1}$$

which is annoying but doable.

3.3 General Newton algorithm for visco-plasticity

3.3.1 Continuous equations

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}}^{p} &= \gamma \mathbf{g}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \\ \dot{\boldsymbol{\alpha}} &= \gamma \mathbf{h}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \\ \gamma &= y\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \end{aligned}$$

Note these are nearly the same equations as above. We can condense out the last equation:

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}}^{p} &= y\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \mathbf{g}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \\ \dot{\boldsymbol{\alpha}} &= y\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \mathbf{h}\left(\boldsymbol{\sigma}, \boldsymbol{\alpha}\right) \end{aligned}$$

3.3.2 Newton integration scheme

- Trial state: $\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_{n}$,$\boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n}^{p}$, $\boldsymbol{\sigma}_{n+1} = \mathbf{C}_{n+1} : \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{p}\right)$
- Evaluate $y\left(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\alpha}_{n+1}\right)$. If zero return, else do Newton iteration on the scheme described here.

Equations:

$$\begin{aligned} \boldsymbol{\varepsilon}_{n+1}^{p} &= \boldsymbol{\varepsilon}_{n}^{p} + \mathbf{g}_{n+1} y_{n+1} \Delta t_{n+1} \\ \boldsymbol{\alpha}_{n+1} &= \boldsymbol{\alpha}_{n} + \mathbf{h}_{n+1} y_{n+1} \Delta t_{n+1} \end{aligned}$$

with $\boldsymbol{\sigma}_{n+1} = \mathbf{C}_{n+1} : \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{p}\right)$.

Residual:

$$\mathbf{R} = \left[\begin{array}{c} -\boldsymbol{\varepsilon}_{n+1}^{p} + \boldsymbol{\varepsilon}_{n}^{p} + \mathbf{g}_{n+1} y_{n+1} \Delta t_{n+1} \\ -\boldsymbol{\alpha}_{n+1} + \boldsymbol{\alpha}_{n} + \mathbf{h}_{n+1} y_{n+1} \Delta t_{n+1} \end{array}\right] = \left[\begin{array}{c} \mathbf{R}_1 \\ \mathbf{R}_2 \end{array}\right].$$

Jacobian. Helpful note is that

$$\boldsymbol{\varepsilon}_{n+1}^{p} = \boldsymbol{\varepsilon}_{n+1} - \mathbf{C}_{n+1}^{-1} : \boldsymbol{\sigma}_{n+1}$$

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}^{p}} = -\mathbf{C}_{n+1}$$

$$\mathbf{J} = \left[\begin{array}{cc} -\mathbf{I} - \left(\frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} y_{n+1} + \mathbf{g}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}}\right) : \mathbf{C}_{n+1} \Delta t_{n+1} & \left(\frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}} y_{n+1} + \mathbf{g}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}}\right) \Delta t_{n+1} \\ -\left(\frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} y_{n+1} + \mathbf{h}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}}\right) : \mathbf{C}_{n+1} \Delta t_{n+1} & -\mathbf{I} + \left(\frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}} y_{n+1} + \mathbf{h}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\alpha}_{n+1}}\right) \Delta t_{n+1} \end{array} \right] = \left[\begin{array}{l} \mathbf{J} \\ \mathbf{J} \end{array} \right.$$

3.3.3 Algorithmic tangent

We can use a very similar algorithm to the one above.

Let

$$\mathbf{A} = \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}_{n+1}} = \left(\frac{\partial \mathbf{g}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} y_{n+1} + \mathbf{g}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\sigma}_{+1}}\right) : \mathbf{C}_{n+1} \Delta t_{n+1}$$

$$\mathbf{B} = \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}_{n+1}} = \left(\frac{\partial \mathbf{h}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} y_{n+1} + \mathbf{h}_{n+1} \otimes \frac{\partial y_{n+1}}{\partial \boldsymbol{\sigma}_{+1}}\right) : \mathbf{C}_{n+1} \Delta t_{n+1}$$

$$\mathbf{0} = \left[\begin{array}{c} \mathbf{A} \\ \mathbf{B} \end{array} \right] + \left[\begin{array}{cc} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{array} \right] \left[\begin{array}{c} \mathbf{K} \\ \mathbf{E} \end{array} \right]$$

$$\mathbf{K} = \left(\mathbf{J}_{11} - \mathbf{J}_{12} \mathbf{J}_{22}^{-1} \mathbf{J}_{21}\right)^{-1} \left(\mathbf{J}_{12} \mathbf{J}_{22}^{-1} \mathbf{B} - \mathbf{A}\right)$$

$$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \mathbf{C} : (\mathbf{I} - \mathbf{K}) \, .$$

So it's the same as above but with slightly smaller blocks.

3.4 General integration algorithm for thermo-elasto-viscoplasticity

3.4.1 Continuous equations

The system is posed as:

$$\begin{aligned} \dot{\boldsymbol{\sigma}} = \quad & \dot{\boldsymbol{\sigma}} \left(\boldsymbol{\sigma}, \mathbf{q}, \dot{\varepsilon}, T, \dot{T}, t\right) \\ \dot{\mathbf{q}} = \quad & \dot{\mathbf{q}} \left(\boldsymbol{\sigma}, \mathbf{q}, \dot{\varepsilon}, T, \dot{T}, t\right) \end{aligned}$$

3.4.2 Newton algorithm

$$\begin{aligned} \mathbf{R}_1 = \quad & -\boldsymbol{\sigma}_{n+1} + \boldsymbol{\sigma}_n + \dot{\boldsymbol{\sigma}} \left(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \dot{\boldsymbol{\varepsilon}}_{n+1}, T_{n+1} \dot{T}_{n+1}, , t_{n+1}\right) \Delta t_{n+1} = \quad & \mathbf{0} \\ \mathbf{R}_2 = \quad & -\mathbf{q}_{n+1} + \mathbf{q}_n + \dot{\mathbf{q}} \left(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \dot{\boldsymbol{\varepsilon}}_{n+1}, T_{n+1}, \dot{T}_{n+1}, t_{n+1}\right) \Delta t_{n+1} = \quad & \mathbf{0} \end{aligned}$$

The Jacobian is simple:

$$\mathbf{J} = \left[\begin{array}{cc} \mathbf{J}_{\sigma\sigma} & \mathbf{J}_{\sigma q} \\ \mathbf{J}_{q\sigma} & \mathbf{J}_{qq} \end{array} \right] = \left[\begin{array}{cc} -\mathbf{I} + \frac{\partial \dot{\boldsymbol{\sigma}}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} \Delta t_{n+1} & \frac{\partial \dot{\boldsymbol{\sigma}}_{n+1}}{\partial \boldsymbol{q}_{n+1}} \Delta t_{n+1} \\ \frac{\partial \dot{\mathbf{q}}_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} \Delta t_{n+1} & -\mathbf{I} + \frac{\partial \dot{\mathbf{q}}_{n+1}}{\partial \boldsymbol{q}_{n+1}} \Delta t_{n+1} \end{array} \right] .$$

3.4.3 Algorithmic tangent

The tangent requires several unusual derivatives. It is formed as:

$$\begin{aligned}
d\mathbf{R}_1 = &\quad \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\sigma}_{n+1}} : d\boldsymbol{\sigma}_{n+1} + \frac{\partial \mathbf{R}_1}{\partial \mathbf{q}_{n+1}} : d\mathbf{q}_{n+1} + \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\varepsilon}_{n+1}} : d\dot{\boldsymbol{\varepsilon}}_{n+1} + \frac{\partial \mathbf{R}_1}{\partial T_{n+1}} : dT_{n+1} + \frac{\partial \mathbf{R}_1}{\partial t_{n+1}} : dt_{n+1} = &\quad \mathbf{0} \\
d\mathbf{R}_2 = &\quad \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\sigma}_{n+1}} : d\boldsymbol{\sigma}_{n+1} + \frac{\partial \mathbf{R}_2}{\partial \mathbf{q}_{n+1}} : d\mathbf{q}_{n+1} + \frac{\partial \mathbf{R}_2}{\partial \boldsymbol{\varepsilon}_{n+1}} : d\dot{\boldsymbol{\varepsilon}}_{n+1} + \frac{\partial \mathbf{R}_2}{\partial T_{n+1}} : dT_{n+1} + \frac{\partial \mathbf{R}_2}{\partial t_{n+1}} : dt_{n+1} = &\quad \mathbf{0}
\end{aligned}$$

Divide through and substitute...

$$\begin{aligned}
\mathbf{J}_{\sigma\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{\sigma q} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial T_{n+1}} : \frac{dT_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \dot{T}_{n+1}} : \frac{d\dot{T}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial t_{n+1}} : \frac{dt_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} = &\quad \mathbf{0} \\
\mathbf{J}_{q\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{qq} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial T_{n+1}} : \frac{dT_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \dot{T}_{n+1}} : \frac{d\dot{T}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial t_{n+1}} : \frac{dt_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} = &\quad \mathbf{0}
\end{aligned}$$

We will need to revisit this, but as an approximation assume:

1. There is no explicit time dependence or the time dependence is small compared to the dependence on the physical variables.

2. The temperature change is gradual (need to consider this...).

With these assumptions:

$$\begin{aligned}
\mathbf{J}_{\sigma\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{\sigma q} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_1}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} = &\quad \mathbf{0} \\
\mathbf{J}_{q\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{qq} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \frac{\partial \mathbf{R}_2}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} = &\quad \mathbf{0}
\end{aligned}$$

and this is basically our old algorithm:

$$\begin{aligned}
\mathbf{X} = &\quad \frac{\partial \dot{\boldsymbol{\sigma}}}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} \Delta t_{n+1} \\
\mathbf{Y} = &\quad \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\boldsymbol{\varepsilon}}_{n+1}} \Delta t_{n+1}
\end{aligned}$$

$$\begin{aligned}
\mathbf{J}_{\sigma\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{\sigma q} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \mathbf{X} = &\quad \mathbf{0} \\
\mathbf{J}_{q\sigma} : \mathbf{A}_{n+1} + \mathbf{J}_{qq} : \frac{d\mathbf{q}_{n+1}}{d\dot{\boldsymbol{\varepsilon}}_{n+1}} + \mathbf{Y} = &\quad \mathbf{0}
\end{aligned}$$

$$\mathbf{A}_{n+1} = \left(\mathbf{J}_{\sigma\sigma} - \mathbf{J}_{\sigma q} : \mathbf{J}_{qq}^{-1} : \mathbf{J}_{q\sigma}\right)^{-1} : \left(\mathbf{J}_{\sigma q} : \mathbf{J}_{qq}^{-1} : \mathbf{Y} - \mathbf{X}\right)$$

And the actual tangent is just

$$\mathbf{T}_{n+1} = \mathbf{A}_{n+1} \frac{1}{\Delta t_{n+1}}$$

3.4.4 Application to plasticity

Define the rates as:

$$\begin{aligned}
\dot{\boldsymbol{\sigma}}\left(\boldsymbol{\sigma},\mathbf{q},\boldsymbol{\varepsilon},T,t\right) &= \mathbf{C}:\left(\dot{\boldsymbol{\varepsilon}}-\mathbf{g}_{\gamma}\dot{\gamma}-\mathbf{g}_{T}\dot{T}-\mathbf{g}_{t}\right)\\
\dot{\mathbf{q}}\left(\boldsymbol{\sigma},\mathbf{q},\boldsymbol{\varepsilon},T,t\right) &= \mathbf{h}_{\gamma}\dot{\gamma}+\mathbf{h}_{T}\dot{T}+\mathbf{h}_{t}
\end{aligned}$$

The derivatives are then

$$\begin{aligned}
\frac{\partial\dot{\boldsymbol{\sigma}}}{\partial\boldsymbol{\sigma}} &= \mathbf{C}:\left(-\frac{\partial\mathbf{g}_{\gamma}}{\partial\boldsymbol{\sigma}}\dot{\gamma}-\left(\mathbf{g}_{\gamma}\otimes\frac{\partial\dot{\gamma}}{\partial\boldsymbol{\sigma}}\right)-\frac{\partial\mathbf{g}_{T}}{\partial\boldsymbol{\sigma}}\dot{T}-\frac{\partial\mathbf{g}_{t}}{\partial\boldsymbol{\sigma}}\right)\\
\frac{\partial\dot{\boldsymbol{\sigma}}}{\partial\mathbf{q}} &= \mathbf{C}:\left(-\frac{\partial\mathbf{g}_{\gamma}}{\partial\mathbf{q}}\dot{\gamma}-\left(\mathbf{g}_{\gamma}\otimes\frac{\partial\dot{\gamma}}{\partial\mathbf{q}}\right)-\frac{\partial\mathbf{g}_{T}}{\partial\mathbf{q}}\dot{T}-\frac{\partial\mathbf{g}_{t}}{\partial\mathbf{q}}\right)\\
\frac{\partial\dot{\mathbf{q}}}{\partial\boldsymbol{\sigma}} &= \frac{\partial\mathbf{h}_{\gamma}}{\partial\boldsymbol{\sigma}}\dot{\gamma}+\left(\mathbf{h}_{\gamma}\otimes\frac{\partial\dot{\gamma}}{\partial\boldsymbol{\sigma}}\right)+\frac{\partial\mathbf{h}_{T}}{\partial\boldsymbol{\sigma}}\dot{T}+\frac{\partial\mathbf{h}_{t}}{\partial\boldsymbol{\sigma}}\\
\frac{\partial\dot{\mathbf{q}}}{\partial\mathbf{q}} &= \frac{\partial\mathbf{h}_{\gamma}}{\partial\mathbf{q}}\dot{\gamma}+\left(\mathbf{h}_{\gamma}\otimes\frac{\partial\dot{\gamma}}{\partial\boldsymbol{\sigma}}\right)+\frac{\partial\mathbf{h}_{T}}{\partial\mathbf{q}}\dot{T}+\frac{\partial\mathbf{h}_{t}}{\partial\mathbf{q}}\\
\frac{\partial\dot{\boldsymbol{\sigma}}}{\partial\dot{\boldsymbol{\varepsilon}}_{n+1}} &= \mathbf{C}\\
\frac{\partial\dot{\mathbf{q}}}{\partial\dot{\boldsymbol{\varepsilon}}_{n+1}} &= \mathbf{0}
\end{aligned}$$

Chapter 4

Implemented class hierarchy

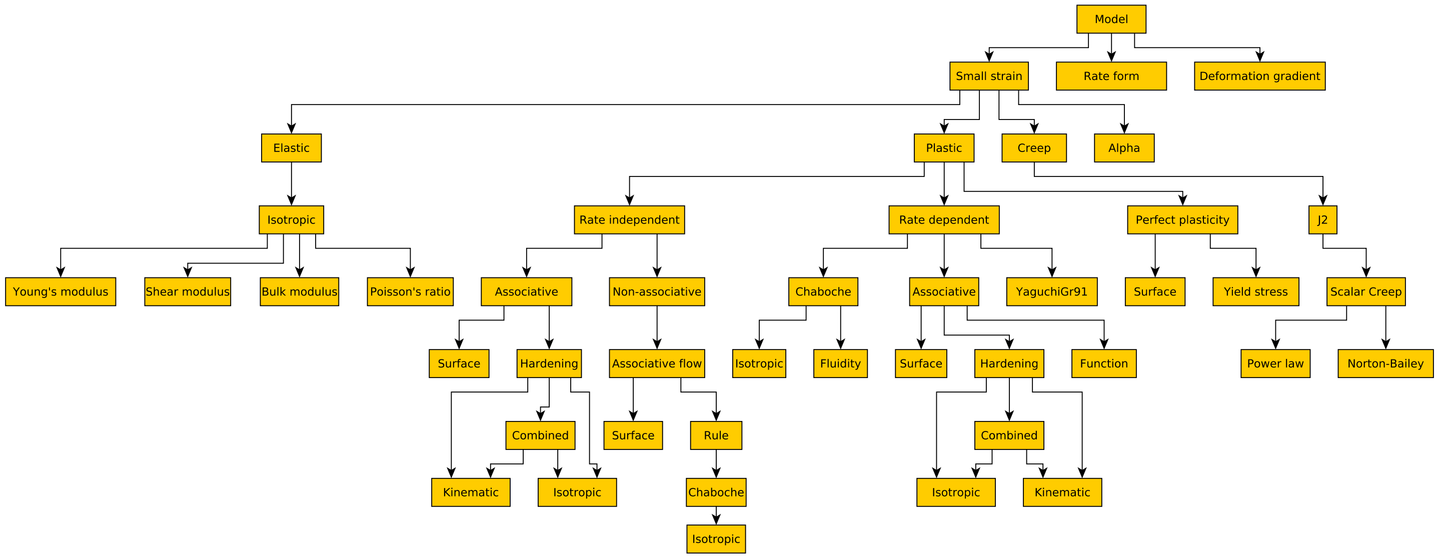

Figure 4.1: Structure of NEML. The diagram shows the available model combinations and guides the XML based input format. Essentially each box in the flow chart corresponds to a node in the XML input file.