Implementation of material constitutive equations in finite element codes

J. Besson
Centre des Matériaux, École des Mines de Paris,
UMR CNRS 7633, BP 87 Evry cedex 91003

Introduction

Why is it important to know how to implement
material constitutive equations in FE codes ?

- Few constitutive equations are avaliable in commercial codes
- Implement new constitutive equations in FE codes (ABAQUS, ANSYS, MARC, . . . , ASTER, CAST3M, . . . , Zébulon, WARP3D)
- Understand convergence problems

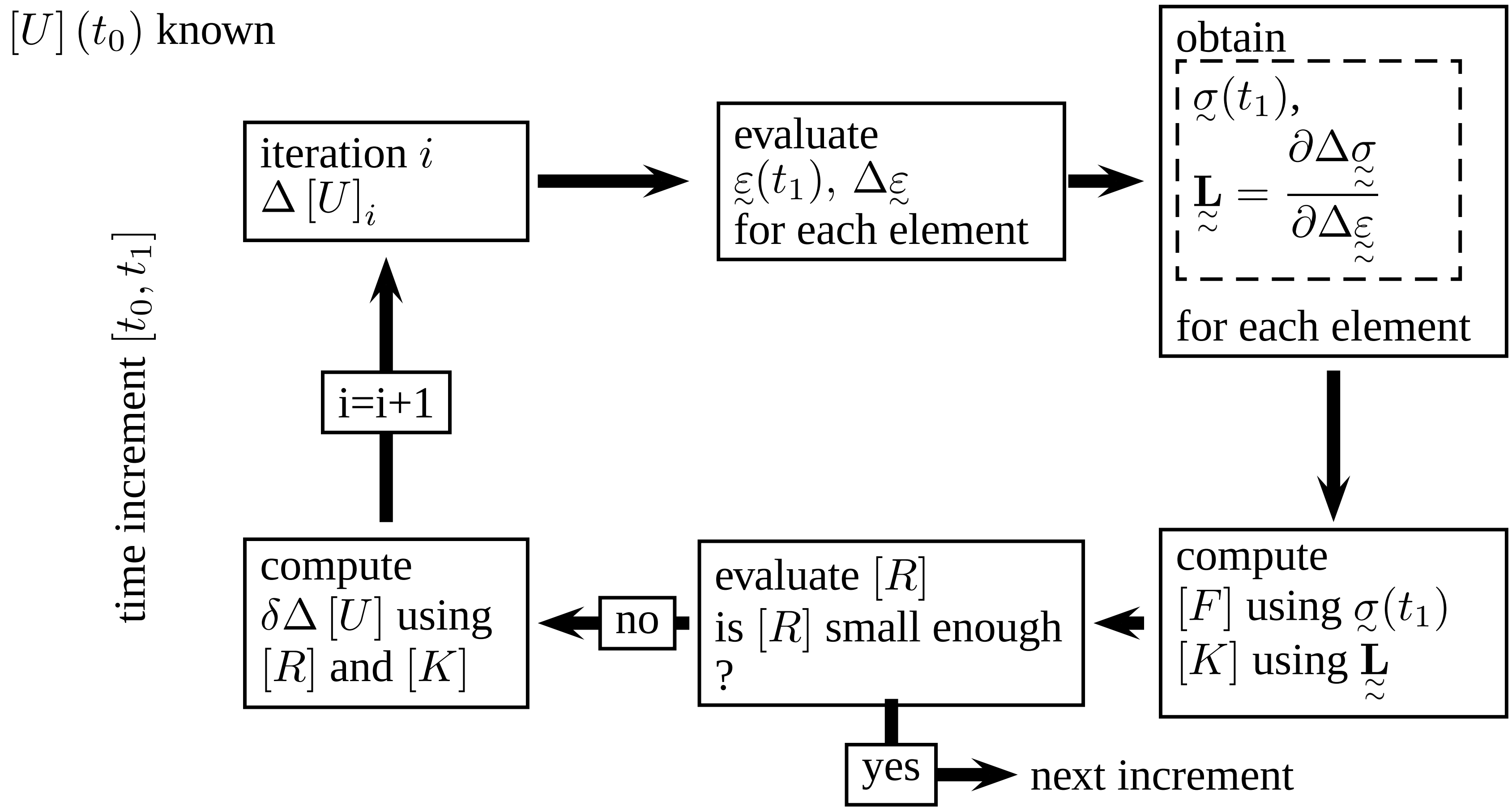
Outline

- Definition of a constitutive equation (FE code point of view)
- Numerical integration methods (explicites/implicites)
- Consistent tangent matrix
- Particular case : von Mises material
- Convergence

Definition of a constitutive equation

- For a displacement based FE formulation, nodal displacements are assumed to be known and therefore the deformations
- The constitutive equation must then supply: (i) stresses $\underset{\sim}{\sigma}$ and (ii) the consistent tangent matrix $\underset{\approx}{\mathbf{L}} = \partial\Delta\underset{\sim}{\sigma}/\partial\Delta\underset{\sim}{\varepsilon}$ for a given strain increment $\Delta\underset{\sim}{\varepsilon}$.
- Complex constitutive equations are characterized by internal state variables $[A]$: the constitutive equation must provide an update of these variables consistent with the strain and time increment.

Role of the constitutive equation in the FEM

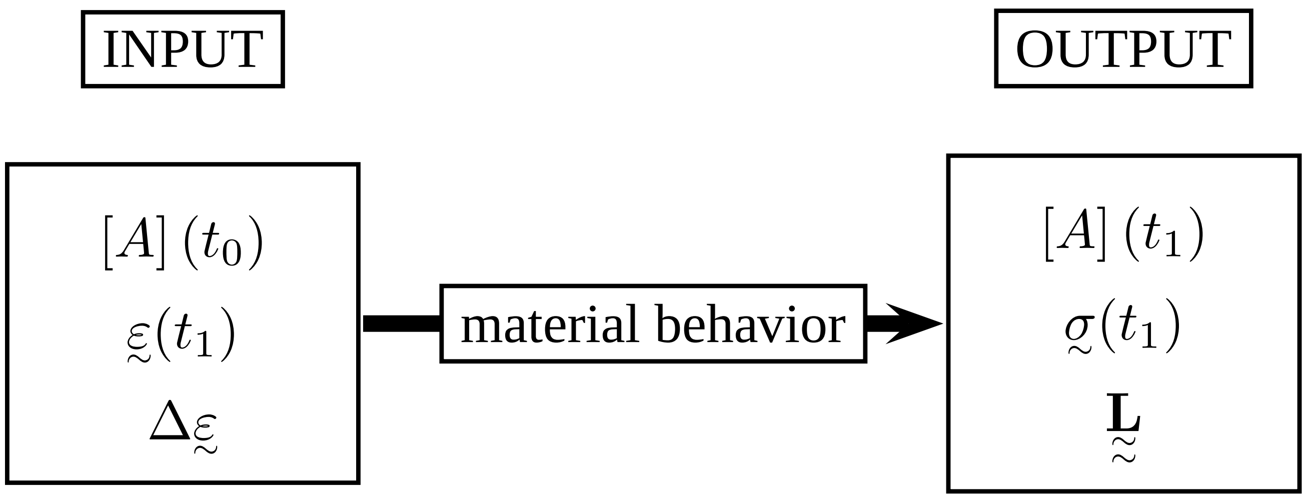


: global computation

: local time integration of the constitutive equations

Generic interface behavior/FEM

- $\Delta t = t_1 - t_0$

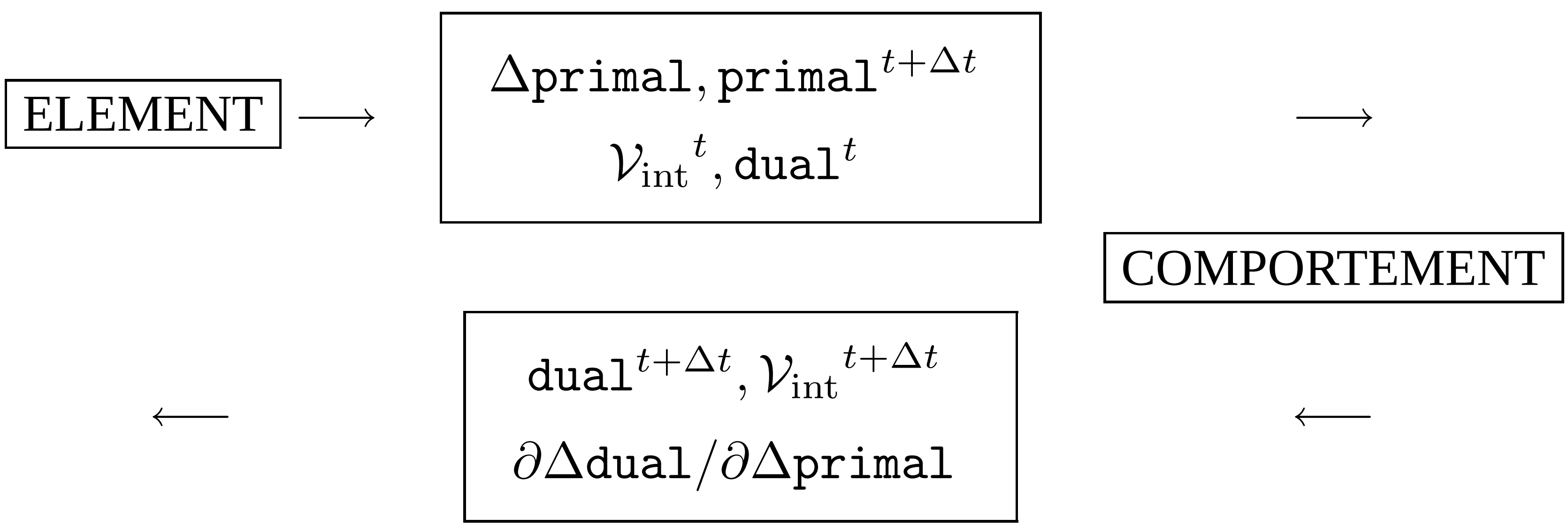


Quantities characterising the material behavior

- Integrated variables/State variables ($\mathcal{V}_{\rm int}$)
- Auxiliary variables ($\mathcal{V}_{\rm aux}$)
- External parameters (EP)
- Coefficients (CO)

$$\texttt{CO} = \texttt{CO}(\texttt{EP}, \mathcal{V}_{\rm int}, \mathcal{V}_{\rm aux})$$

- Interface: input variable (primal), associated dual variable (dual), tangent matrix $\partial\Delta\texttt{dual}/\partial\Delta\texttt{primal}$.



Examples of primal—dual couples

problème	`primal`	`dual`
mechanics (small deformation)	$\underset{\sim}{\varepsilon}$	$\underset{\sim}{\sigma}$
mechanics (finite strain)	$\underset{\sim}{\mathbf{F}}$	$\underset{\sim}{\sigma}$ ou $\underset{\sim}{\mathbf{S}}$
thermal problem	$(T, \underline{\mathrm{grad}} T)$	$(H, \underline{\mathbf{q}})$

Constitutive equations as a differential equation

$$\frac{d\,[A]}{dt} = [\dot{A}] = [G]\left([A]\,,t\right)$$

$$\frac{A_i}{dt} = [\dot{A}]_i = G_i(A_1, \ldots, A_n, t)$$

- Time (t) represents the imposed deformation but also an external parameter such as the temperature ($T(\vec{x}, t)$).
- The FE evaluation of the Constitutive equation for ($\Delta\underset{\sim}{\varepsilon}$, Δ) corresponds to the integration of the previous equation from t_0 to t_1.
- In most cases: $[A] = (\underset{\sim}{\varepsilon}_e, \ldots)$ so that $\underset{\sim}{\sigma}(t_1) = \underset{\approx}{\mathbf{E}}(t_1) : \underset{\sim}{\varepsilon}_e(t_1)$,
- $\underset{\approx}{\mathbf{L}}$ must be computed …

Integration methods of Constitutive equations

Euler explicit method

$$[A]\,(t_1) = [A]\,(t_0) + [\dot{A}]\,([A]\,(t_0), t_0)\,\Delta t = [A]\,(t_0) + [G]\,([A]\,(t_0), t_0)\,\Delta t$$

- The method is not stable and should be avoided
- Explicit : bacause $[\dot{A}]$ is computed at t_0 for the known couple $([A]\,(t_0), t_0)$

Runge–Kutta explicit method

- Numerical estimation of the derivatives of $[\dot{A}]$ (i.e. $d^2\left[F\right]_A/dt^2$, $d^3\left[F\right]_A/dt^3$,$\ldots$)
- Error estimation to control the solution

The Runge–Kutta Integration method is easy to implement because it only uses the differential equation $[\dot{A}] = [G]\left(\left[A\right], t\right)$. It however has some drawbacks:

- Integration may require a large CPU time
- In the case of plastic materials, it is mandatory to compute the plastic multiplier which can be a difficult task (see below) in the case of temperature dependant material coefficients.

Runge–Kutta Method

Using a Taylor expansion, one gets for a time increment $[t, t+\Delta t]$ (which can differ from the FE time step $[t_0, t_1]$:

$$\{v\}(t+\Delta t) = \{v\}(t) + \{\dot{v}\}(t)\Delta t + O(\Delta t^2)$$

The accuracy of the Euler integration is therefore of magnitude $O(\Delta t^2)$. Based on this first estimation of the increment, an other one can be performed using the mid-point (i.e. $t + \Delta t/2$). Let:

$$\{\delta v_1\} = \Delta t \{\dot{v}\}(t)$$

and

$$\begin{aligned} \{\delta v_2\} &= \Delta t \{\dot{v}\}\left(t + \frac{\Delta t}{2}, \{v\}(t) + \frac{1}{2}\{\delta v_1\}\right) \\ &= \Delta t \left(\{\dot{v}\}(t) + \frac{\Delta t}{2}\{\ddot{v}\}(t)\right) \\ &= \{\delta v_1\} + \frac{\Delta t^2}{2}\{\ddot{v}\}(t) \end{aligned}$$

This provides one way to estimate $\{\ddot{v}\}(t)$. The second order Taylor expansion is:

$$\{v\}(t+\Delta t) = \{v\}(t) + \{\dot{v}\}(t)\Delta t + \{\ddot{v}\}(t)\frac{\Delta t^2}{2} + O(\Delta t^3)$$

which can be simplified using the previous estimate of $\{\ddot{v}\}(t)$:

$$\{v\}(t+\Delta t) = \{v\}(t) + \{\delta v_2\} + O(\Delta t^3)$$

The precision has been improved ($O(\Delta t^3)$ instead of $O(\Delta t^2)$). This is a second order method.

The procedure can be generalized. This leads to a 4th order Runge–Kutta method which is written as:

$$
\begin{aligned}
\{\delta v_1\} &= \Delta t\,\{\dot{v}\}\,(t,\{v\}) \\
\{\delta v_2\} &= \Delta t\,\{\dot{v}\}\left(t+\frac{\Delta t}{2},\{v\}+\frac{1}{2}\{\delta v_1\}\right) \\
\{\delta v_3\} &= \Delta t\,\{\dot{v}\}\left(t+\frac{\Delta t}{2},\{v\}+\frac{1}{2}\{\delta v_2\}\right) \\
\{\delta v_4\} &= \Delta t\,\{\dot{v}\}\,(t+\Delta t,\{v\}+\{\delta v_3\}) \\
\{v\}\,(t+\Delta t) &= \{v\}\,(t)+\frac{1}{6}\{\delta v_1\}+\frac{1}{3}\{\delta v_2\}+\frac{1}{3}\{\delta v_3\}+\frac{1}{6}\{\delta v_4\}+O(\Delta t^5)
\end{aligned}
$$

Runge–Kutta Method : error control

Aim: Obtain a given precision while minimize the computational effort

Make large time steps when the $\{\dot{v}\}$ function varies little and smaller time steps if its evolution is rapid.

Let Δt be the time increment over whivh the itegration has to be performed. It can be divided into n sub-steps so that:

$$\Delta t = \sum_k \delta t_k$$

The error is estimated be applying the RK4 method

one time step	δt	$\rightarrow \{v_1\}$
two time steps	$\delta t/2$	$\rightarrow \{v_2\}$

This corresponds to 11 evalutions of $\{\dot{v}\}$.

Let $\{v\}(t+\delta t)$ be the exact solution; one gets

$$\begin{aligned} \{v\}(t+\delta t) &= \{v_1\} + (\delta t)^5 \{\phi\} + O(\delta t^6) \\ \{v\}(t+\delta t) &= \{v_2\} + 2(\delta t/2)^5 \{\phi\} + O(\delta t^6) \end{aligned}$$

$$\{\phi\} \approx \text{constant} \approx \frac{1}{5!}\left\{v^{(5)}\right\}$$

The difference between both estimations is an indicator the error:

$$\{E\} = \{v_2\} - \{v_1\}$$

This difference has to be kept smaller that a prescribed precision by adjusting δt. This equation can be solved neglecting $O(\delta t^6)$ terms:

$$\{v\}(t+\delta t) = \{v_2\} + \frac{1}{15}\{E\} + O(\delta t^6)$$

This is a better estimation (5th order).

$\{E\}$ can then be used to modify the time step. Let $\{E^0\}$ by the requested precision (note that the precision is a vector).

$$\text{if} \;\; E_i < E_i^0, \; \forall i$$

The time step can be increased

$$\text{if} \;\; \exists i, \; E_i > E_i^0$$

The time step must be decreased

The time step is corrected by the following factor:

$$\alpha = \min_i \left| \frac{E_i^0}{E_i} \right|^{0.2}$$

as the error varies as δt^5 for the 4th order Runge–Kutta method.

$\{E^0\}$ must now be chosen.

The required precision must be obtained over the whole time increment Δt and not only on local sub steps δt_k. In that case the error is best defined as:

$$E_i^0 = \epsilon \delta t \left| \frac{dv_i}{dt} \right| = \epsilon \left| \delta v_i \right|$$

Implicite methods (or θ-methods)

- Evaluate $[\dot{A}]$ at t_θ between t_0 et t_1
- $t_\theta = t_0 + \theta\Delta t$ with $0 \leq \theta \leq 1$
- Two solutions :

$$\begin{aligned}
\Delta [A] &= [G] ([A] (t_0) + \theta\Delta [A] \ , \ t_0 + \theta\Delta t)\Delta t \\
&= [G] ([A]_\theta \, , t_\theta)\Delta t \\
\Delta [A] &= [(1-\theta) [G] ([A] (t_0), t_0) + \theta [G] ([A] (t_1), t_1)] \Delta t \\
&= [(1-\theta) [G] ([A]_0 \, , t_0) + \theta [G] ([A]_0 + \Delta [A] \, , t_0 + \Delta t)] \Delta t
\end{aligned}$$

- Implicit : $\Delta [A]$ appears on both left and right handsides of the previous equations
- Integrate the constitutive equation $=$ solve the implicit equations
- $\theta = 0 \rightarrow$ Euler
- ...in the following the first method will only be considered

- Solution obtained by the Newton-Raphson method
- Write a residual vector

$$[R]\left(\Delta[A]\right) = \Delta[A] - [G]\left([A](t_0) + \theta\Delta[A]\ ,\ t_0 + \theta\Delta t\right)\Delta t$$
$$R_i(\Delta A_1, \ldots, \Delta A_n) = \Delta A_i - G_i(A_1(t_0) + \theta\Delta A_1, \ldots, A_n(t_0) + \theta\Delta A_n)\Delta t$$

- 1st order Taylor expansion around an estimation $\Delta[A]_s$:

$$[R] = [R]\left(\Delta[A]_s\right) + \frac{\partial[R]}{\partial\Delta[A]}.\left(\Delta[A] - \Delta[A]_s\right) = [0]$$

- Construction of the next estimation:

$$\Delta[A]_{s+1} = \Delta[A]_s - \left(\frac{\partial[R]}{\partial\Delta[A]}\right)^{-1}_{\Delta[A]=\Delta[A]_s} . [R]\left(\Delta[A]_s\right)$$

- $[J] = \partial[R]/\partial\Delta[A]$ ($J_{ij} = \partial R_j/\partial A_j$): Jacobian matrix , $[J]^\star = [J]^{-1}$

Note

- The internal variable vector $[A]$ often contains 2nd order tensors
- The Voigt notation is used.

$$
\underset{\sim}{\varepsilon} = \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \end{pmatrix}, \quad \underset{\sim}{\sigma} = \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{pmatrix} \qquad \text{—} \qquad \underset{\sim}{\mathbf{x}} = \begin{pmatrix} x_{11} \\ x_{22} \\ x_{33} \\ \sqrt{2}x_{12} \\ \sqrt{2}x_{23} \\ \sqrt{2}x_{31} \end{pmatrix}
$$

Integration — Simple Example : von Mises plasticity

- Additive Decomposition of deformations :

$$\underset{\sim}{\varepsilon} = \underset{\sim}{\varepsilon}_e + \underset{\sim}{\varepsilon}_p$$

- Flow surface

$$\phi = \sigma_{\mathrm{eq}} - R(p)$$

- Normality

$$\dot{\underset{\sim}{\varepsilon}}_p = \dot{p}\frac{\partial \phi}{\partial \underset{\sim}{\sigma}} = \frac{3}{2}\dot{p}\frac{\underset{\sim}{\mathbf{s}}}{\sigma_{\mathrm{eq}}} = \dot{p}\underset{\sim}{\mathbf{n}}$$

- Internal variables : $(\underset{\sim}{\varepsilon}_e, p)$

- $\dot{p}$ is computed using the consitency condition: $\dot{\phi} = 0$

$$\dot{\phi} = \frac{\partial \phi}{\partial \underset{\sim}{\sigma}} : \dot{\underset{\sim}{\sigma}} + \frac{\partial \phi}{\partial p} \dot{p} = \underset{\sim}{\mathbf{n}} : \dot{\underset{\sim}{\sigma}} - H \dot{p}$$

- avec $\underset{\sim}{\sigma} = \underset{\approx}{\mathbf{E}} : \underset{\sim}{\varepsilon}_e = \underset{\approx}{\mathbf{E}} : (\underset{\sim}{\varepsilon} - \underset{\sim}{\varepsilon}_p) \rightarrow \dot{\underset{\sim}{\sigma}} = \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}}_e = \underset{\approx}{\mathbf{E}} : (\dot{\underset{\sim}{\varepsilon}} - \dot{\underset{\sim}{\varepsilon}}_p)$

- so that:

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}}}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

- Differential equations to be integrated:

$$\text{sur } \underset{\sim}{\varepsilon}_e \qquad \dot{\underset{\sim}{\varepsilon}}_e = \dot{\underset{\sim}{\varepsilon}} - \dot{p}\underset{\sim}{\mathbf{n}}$$

$$\text{sur } p \qquad \dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}}}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

- Pay attention to the dependance on external parameters (temperature...)

- Ready for the Runge–Kutta integration

von Mises plasticity : implicit integration

$$\dot{\underset{\sim}{\varepsilon}}_e = \dot{\underset{\sim}{\varepsilon}} - \dot{p}\,\underset{\sim}{\mathbf{n}} \qquad \longrightarrow \qquad \Delta\underset{\sim}{\varepsilon}_e = \Delta\varepsilon - \Delta p\,\underset{\sim}{\mathbf{n}}$$

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}}}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H} \qquad \longrightarrow \qquad \Delta p = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \Delta\varepsilon}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

- Evaluation of $\underset{\sim}{\mathbf{n}}$, $\underset{\approx}{\mathbf{E}}$, H ? …at time $t_\theta = t_0 + \theta\Delta t$.
- Application :

$$\underset{\sim}{\mathbf{n}} = \frac{3}{2}\frac{\underset{\sim}{\mathbf{s}}^\theta}{\sigma^\theta_{\mathrm{eq}}} \quad \text{avec} \quad \underset{\sim}{\sigma}^\theta = \underset{\approx}{\mathbf{E}}^\theta : \underset{\sim}{\varepsilon}^\theta_e \quad \underset{\sim}{\varepsilon}^\theta_e = \underset{\sim}{\varepsilon}^0_e + \theta\Delta\underset{\sim}{\varepsilon}_e$$

$$\underset{\approx}{\mathbf{E}}^\theta = \underset{\approx}{\mathbf{E}}(T^\theta) = \underset{\approx}{\mathbf{E}}(T^0 + \theta\Delta T)$$

$$H^\theta = H(p^\theta) = H(p^0 + \theta\Delta p)$$

- The equation

$$\Delta p = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \Delta\varepsilon}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

is exact but can be replaced by:

$$\phi = \sigma_{\text{eq}} - R(p) = 0$$

It is wrong if R depends on an external parameter (temperature,...) as:

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}} - R_{,T}\dot{T}}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

The correct incremental equation is then:

$$\Delta p = \frac{\underset{\sim}{\mathbf{n}}^{\theta} : \underset{\approx}{\mathbf{E}}^{\theta} : \Delta\underset{\sim}{\varepsilon} - R^{\theta}_{,T}\Delta T}{\underset{\sim}{\mathbf{n}}^{\theta} : \underset{\approx}{\mathbf{E}}^{\theta} : \underset{\sim}{\mathbf{n}}^{\theta} + H^{\theta}}$$

This method should be avoided !

Residual vector

$$
\begin{aligned}
R_e &= \Delta \underset{\sim}{\varepsilon}_e + \Delta p \underset{\sim}{\mathbf{n}}^\theta - \Delta \varepsilon \\
R_p &= \phi = \sigma_{\text{eq}}^\theta - R(p^\theta)
\end{aligned}
$$

$$
R = (R_e, R_p)
$$

Jacobian matrix . . . a though job

- The Jacobian matrix can be written by bloacks:

$$[J] = \begin{pmatrix} \dfrac{\partial R_e}{\partial \Delta \underset{\sim}{\varepsilon}_e} & \dfrac{\partial R_e}{\partial \Delta p} \\ \dfrac{\partial R_p}{\partial \Delta \underset{\sim}{\varepsilon}_e} & \dfrac{\partial R_p}{\partial \Delta p} \end{pmatrix}$$

Computation of the blocks related to $R_e = \Delta \underset{\sim}{\varepsilon}_e + \Delta p \underset{\sim}{\mathbf{n}}^\theta - \Delta \varepsilon$

$$\frac{\partial R_e}{\partial \Delta \underset{\sim}{\varepsilon}_e} = \underset{\approx}{\mathbf{1}} + \Delta p \frac{\partial \underset{\sim}{\mathbf{n}}}{\partial \underset{\sim}{\sigma}} : \frac{\partial \underset{\sim}{\sigma}}{\partial \underset{\sim}{\varepsilon}_e} : \frac{\partial \underset{\sim}{\varepsilon}_e}{\partial \Delta \underset{\sim}{\varepsilon}_e}$$

$$\frac{\partial \underset{\sim}{\mathbf{n}}}{\partial \underset{\sim}{\sigma}} \rightarrow \underset{\approx}{\mathbf{N}} \qquad \frac{\partial \underset{\sim}{\sigma}}{\partial \underset{\sim}{\varepsilon}_e} \rightarrow \underset{\approx}{\mathbf{E}}^\theta \qquad \frac{\partial \underset{\sim}{\varepsilon}_e}{\partial \Delta \underset{\sim}{\varepsilon}_e} \rightarrow \theta \underset{\approx}{\mathbf{1}}$$

$$\underset{\approx}{\mathbf{N}} = \frac{1}{\sigma_{\text{eq}}} \left(\frac{3}{2} \underset{\approx}{\mathbf{J}} - \underset{\sim}{\mathbf{n}} \otimes \underset{\sim}{\mathbf{n}} \right)$$

$$\Rightarrow \quad \frac{\partial R_e}{\partial \Delta \underset{\sim}{\varepsilon}_e} = \underset{\approx}{\mathbf{1}} + \Delta p \underset{\approx}{\mathbf{N}}^\theta : \underset{\approx}{\mathbf{E}}^\theta$$

$$\frac{\partial R_e}{\partial \Delta p} = \underset{\sim}{\mathbf{n}}^\theta$$

Computation of the blocks related to $R_p = \sigma_{\text{eq}}^{\theta} - R(p^{\theta})$

$$\frac{\partial R_p}{\partial \Delta \underset{\sim}{\varepsilon}_e} = \frac{\partial \sigma_{\text{eq}}}{\partial \underset{\sim}{\sigma}} : \frac{\partial \underset{\sim}{\sigma}}{\partial \underset{\sim}{\varepsilon}_e} : \frac{\partial \underset{\sim}{\varepsilon}_e}{\partial \Delta \underset{\sim}{\varepsilon}_e} = \theta \underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}}$$

$$\frac{\partial R_p}{\partial \Delta p} = -\frac{\partial R}{\partial p} \frac{\partial p}{\partial \Delta p} = -\theta H^{\theta}$$

Tangent matrix vs. consistent tangent matrix

- Tangent matrix

$$\dot{\underset{\sim}{\sigma}} = \underset{\approx}{\mathbf{L}}_p : \dot{\underset{\sim}{\varepsilon}}$$

- Calculated as:

$$\dot{\underset{\sim}{\sigma}} = \underset{\approx}{\mathbf{E}} : (\dot{\underset{\sim}{\varepsilon}} - \dot{p}\underset{\sim}{\mathbf{n}})$$

and

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \dot{\underset{\sim}{\varepsilon}}}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

imply

$$\underset{\approx}{\mathbf{L}}_p = \underset{\approx}{\mathbf{E}} - \frac{(\underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}}) \otimes (\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}})}{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{E}} : \underset{\sim}{\mathbf{n}} + H}$$

- Consistent tangent matrix

$$\underset{\approx}{\mathbf{L}} = \frac{\partial \Delta \underset{\sim}{\sigma}}{\partial \Delta \underset{\sim}{\varepsilon}}$$

$$\Delta \underset{\sim}{\sigma} = \underset{\approx}{\mathbf{E}} : (\Delta \underset{\sim}{\varepsilon} - \Delta p \underset{\sim}{\mathbf{n}})$$

$$\begin{aligned} \delta \Delta \underset{\sim}{\sigma} &= \underset{\approx}{\mathbf{E}} : (\delta \Delta \underset{\sim}{\varepsilon} - \delta \Delta p \underset{\sim}{\mathbf{n}} - \Delta p \delta \underset{\sim}{\mathbf{n}}) \\ \ldots & \\ \text{it can be shown that} & \\ \underset{\approx}{\mathbf{L}} \approx \underset{\approx}{\mathbf{L}}_p - \Delta p \underset{\approx}{\mathbf{E}} : \underset{\approx}{\mathbf{N}} : \underset{\approx}{\mathbf{E}} + \underset{\approx}{\mathbf{O}}(\Delta p^2) & \end{aligned}$$

Automatic and generic computation of the consistent tangent matrix

- Internal variables and residuals can be expressed in a very general way as:

$$\begin{aligned} [A] &= (\underset{\sim}{\varepsilon}_e, [a]) \\ [R] &= ([R]_e\,, [R]_a) \\ [R]_e &= \Delta\underset{\sim}{\varepsilon}_e + \Delta\underset{\sim}{\varepsilon}_{\text{irr}} - \Delta\underset{\sim}{\varepsilon} \end{aligned}$$

- Influence of a small varaition of $\Delta\underset{\sim}{\varepsilon}$ on the internal variables $(\underset{\sim}{\varepsilon}_e, [a])$? (around the solution)
- $[R]$ must stay null

$$\delta\,[R] = [0] = \delta \begin{pmatrix} \Delta\underset{\sim}{\varepsilon}_e + \Delta\underset{\sim}{\varepsilon}_{\text{irr}} \\ [R]_a \end{pmatrix} - \delta \begin{pmatrix} \Delta\underset{\sim}{\varepsilon} \\ [0] \end{pmatrix}$$

$$\delta\,[R] = \frac{\partial\,[R]}{\partial\,[A]} - \begin{pmatrix} \delta\Delta\underset{\sim}{\varepsilon} \\ [0] \end{pmatrix} = [J]\,.\delta\Delta A - \begin{pmatrix} \delta\Delta\underset{\sim}{\varepsilon} \\ [0] \end{pmatrix}$$

- Consequently

$$\delta\Delta A = [J]^{-1} \, . \begin{pmatrix} \delta\Delta\underset{\sim}{\varepsilon} \\ [0] \end{pmatrix}$$

- $[J]^{-1}$ can be divided in sub-blocks:

$$[J]^{-1} = [J]^{\star} = \begin{pmatrix} [J]^{\star}_{ee} & [J]^{\star}_{ea} \\ [J]^{\star}_{ae} & [J]^{\star}_{aa} \end{pmatrix} ,$$

- One therefore gets :

$$\delta\Delta\underset{\sim}{\varepsilon}_e = [J]^{\star}_{ee} \, .\delta\Delta\underset{\sim}{\varepsilon}$$

- Using the Hooke law (elasticity):

$$\underset{\sim}{\sigma}(t_1) = \underset{\sim}{\sigma}(t_0) + \Delta\underset{\sim}{\sigma} = \underset{\approx}{\mathbf{E}}(t_1) : \underset{\sim}{\varepsilon}_e(t_1) = \underset{\approx}{\mathbf{E}}(t_1) : (\underset{\sim}{\varepsilon}_e(t_0) + \Delta\underset{\sim}{\varepsilon}_e)$$

so that:

$$\delta\Delta\underset{\sim}{\sigma} = \underset{\approx}{\mathbf{E}}(t_1) : \delta\Delta\underset{\sim}{\varepsilon}_e = \underset{\approx}{\mathbf{E}}(t_1) : \underset{\approx}{\mathbf{J}}^{\star}_{ee} : \delta\Delta\varepsilon$$

- The consistent tangent matrix is therefore given by:

$$\boxed{\underset{\approx}{\mathbf{L}} = \underset{\approx}{\mathbf{E}}(t_1) : \underset{\approx}{\mathbf{J}}^{\star}_{ee}}$$

- In case where $\underset{\approx}{\mathbf{E}}$ depends on an internal variable (e.g. d=damage) then

$$\Delta\underset{\sim}{\sigma} = \frac{\partial\underset{\approx}{\mathbf{E}}}{\partial d}\delta\Delta d : \underset{\sim}{\varepsilon}_e + \underset{\approx}{\mathbf{E}}(t_1) : \Delta\underset{\sim}{\varepsilon}_e$$

$$\boxed{\underset{\approx}{\mathbf{L}} = \frac{\partial\underset{\approx}{\mathbf{E}}}{\partial d} : (\underset{\sim}{\varepsilon}_e \otimes [J]^{\star}_{de}) + \underset{\approx}{\mathbf{E}} : \underset{\approx}{\mathbf{J}}^{\star}_{ee}}$$

Explicit/Implicit

Explicit	Implicit
easy to implement	difficult to implement
slow	fast
$\underset{\approx}{\mathbf{L}}$?	direct computation of $\underset{\approx}{\mathbf{L}}$

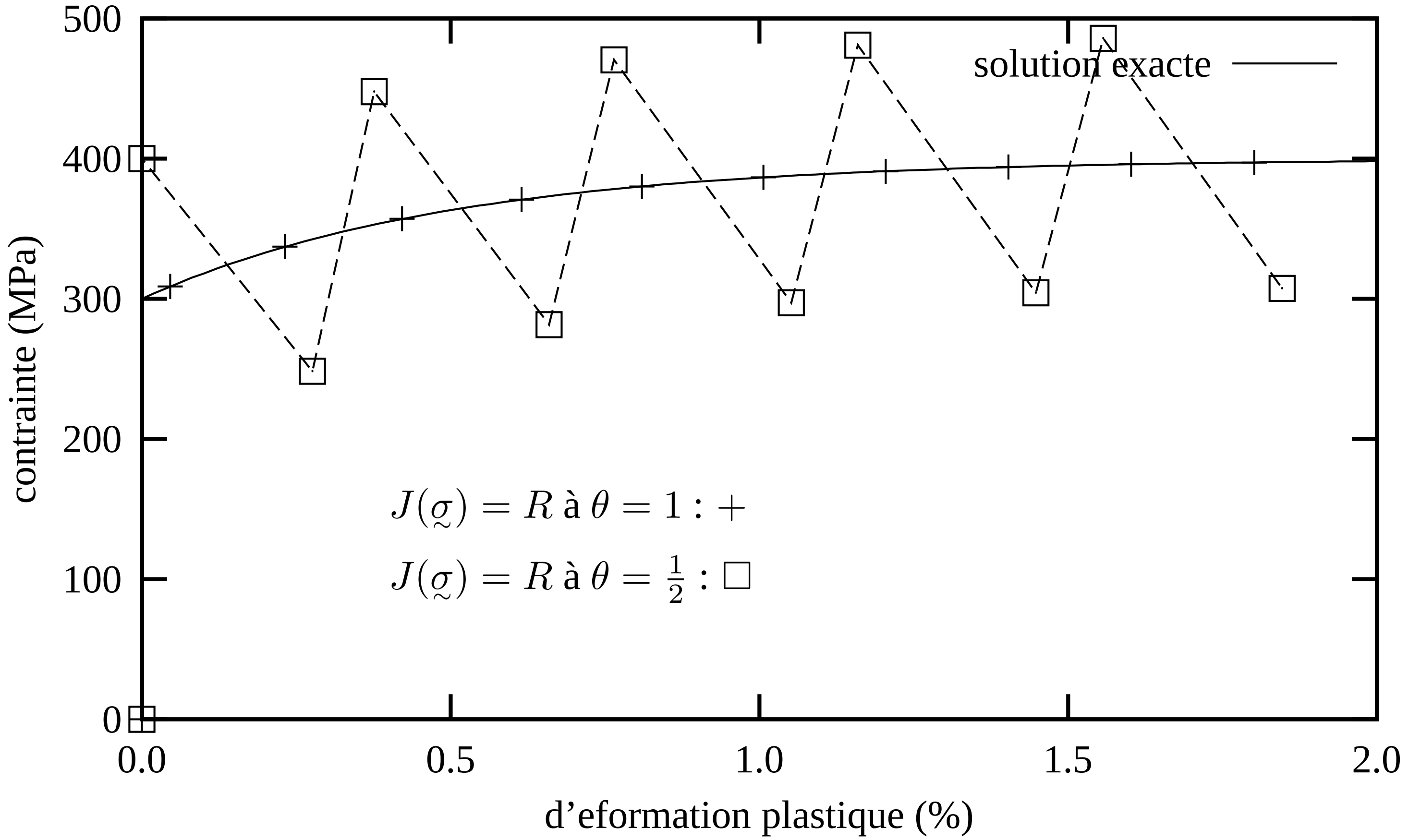
- $\underset{\approx}{\mathbf{L}}$ can be evaluated by perturbation

$$L_{ijkl} = \frac{\sigma_{ij}(\Delta \underset{\sim}{\varepsilon} + \delta\varepsilon \underset{\sim}{\mu}^{kl}) - \sigma_{ij}(\Delta \underset{\sim}{\varepsilon})}{\delta\varepsilon}$$

Choosing de θ in the case of plasticity

The hardening law is

$$R(p) = 300 + 100(1 - \exp(-200p)) \qquad \text{MPa}$$



Solution(s)

- Use $\theta = 1$
- Write the residual R_p for $\theta = 1$

$$
\begin{aligned}
R_e &= \Delta \underset{\sim}{\varepsilon}_e + \Delta p \underset{\sim}{\mathbf{n}}^{\theta} - \Delta \varepsilon \\
R_p &= \sigma_{\text{eq}}^{1} - R(p^{1})
\end{aligned}
$$

- Internal variables must be evaluated at both t_θ (R_e) and $t_1 = t + \Delta t$ (R_p) !

Plasticity : variable temperature

It is assumed that the flow stress R depends on temperature T The consistency condition is expressed as:

$$\dot{f} = \underset{\sim}{\mathbf{n}} : \underset{\sim}{\dot{\sigma}} - R_{,p}\dot{p} - R_{,T}\dot{T} = 0$$

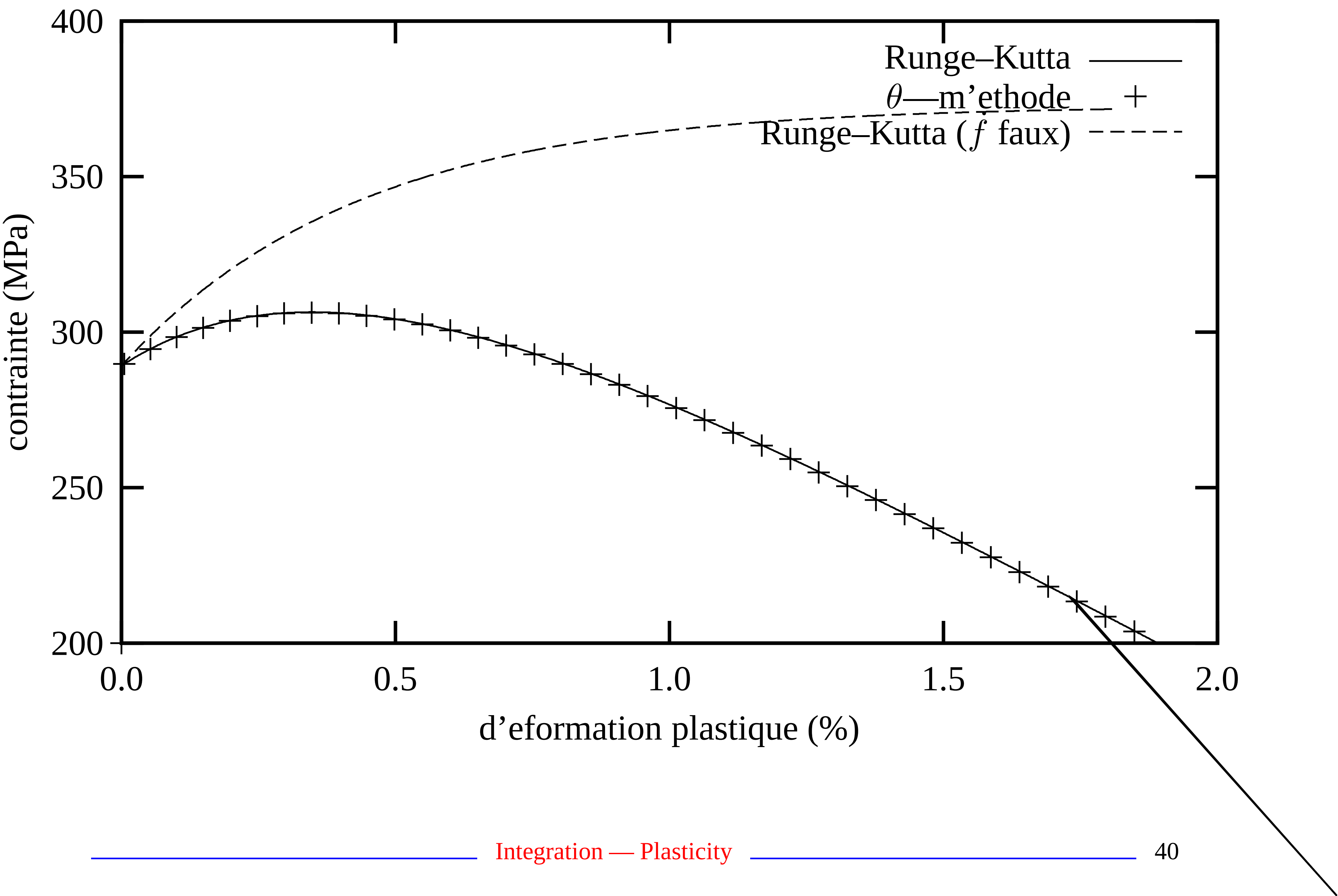
and one gets

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}} : \underset{\sim}{\dot{\epsilon}}_t - R_{,T}\dot{T}}{R_{,p} + \underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}} : \underset{\sim}{\mathbf{n}}}$$

using the previous exemple with:

$$R(p,T) = [300 + 100(1 - \exp(-200p))][1 - T/200]$$

it is shown the omitting the $R_{,T}\dot{T}$ term in the consistency condition leads to wrong results:


400
350
300
250
200
0.0
0.5
1.0
1.5
2.0
contrainte (MPa)
d'eformation plastique (%)
Runge–Kutta
θ—m'ethode
Runge–Kutta (f faux)

In the case where the elasticity coefficients also depend on temperature, this dependance must also be accounted for while writting the consistency condition. The plastic multiplier is then expressed as:

$$\dot{p} = \frac{\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}} : \underset{\sim}{\dot{\epsilon}}_t - \dot{T}\underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}}_{,T} : \epsilon_e - R_{,T}\dot{T}}{R_{,p} + \underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}} : \underset{\sim}{\mathbf{n}}}$$

It may become difficult to take into account the various possible dependancies when several external parameters are prescribed.

This problems are avoided in the case of the θ–method as in all case the yield condition is deirctly used $f_{t+\Delta t} = 0$ and not $\dot{f} = 0$.

Prandtl–Reuss law: creep

In the case of a viscous material, $\dot{p}$ is directly obtained from the creep low:

$$\dot{p} = \phi(\underset{\sim}{\sigma}, A_i)$$

θ–method:

$$r_p = \Delta p - \Delta t\, \phi(J - R, \dots)_\theta = 0$$

The partial derivatives related to the computation of the Jacobian matrix are:

$$\frac{\partial r_p}{\partial \Delta \underset{\sim}{\epsilon}_e} = -\theta \Delta t\, \phi_{,\omega}\, \underset{\approx}{\mathbf{C}} : \underset{\sim}{\mathbf{n}}$$

$$\frac{\partial r_p}{\partial \Delta p} = 1 + \theta\, \Delta t\, R_{,p}\, \phi_{,\omega}$$

There are not longer problems related to the calculation of the consistency condition. A "creep law" can be used to mimic plasticity. For instance using a Norton law

$$\phi(\omega) = \left\langle \frac{\omega}{K} \right\rangle^n$$

if n is high enough, one gets

$$J - R \simeq K$$

Multi–kinematic law : constitutive equations

- The law is expressed using the following state internal variables

$\underset{\sim}{\epsilon}_e$	elastic strain tensor
$\underset{\sim}{\alpha}_i$	kinematic hardening variable tensors
r	isotropic hardening variable

- The following variable is an auxiliary variable:

p	cumulated plastic strain

Forces associated to the state variables are:

$$
\begin{aligned}
\underset{\sim}{\sigma} &= \underset{\approx}{\mathbf{C}} : \underset{\sim}{\epsilon}_e \\
\underset{\sim}{\mathbf{X}}_i &= \underset{\approx}{\mathbf{C}}_i : \underset{\sim}{\alpha}_i \\
R &= cr
\end{aligned}
$$

- The back-stress $\underset{\sim}{\mathbf{X}}$ is given by:

$$\underset{\sim}{\mathbf{X}} = \sum_i \underset{\sim}{\mathbf{X}}_i$$

The evolution laws are given by:

$$\dot{\underset{\sim}{\epsilon}}_e + \dot{\underset{\sim}{\epsilon}}_p = \dot{\underset{\sim}{\epsilon}}_t$$
$$\dot{\underset{\sim}{\alpha}}_i = \dot{\underset{\sim}{\epsilon}}_p - \dot{p}\underset{\approx}{\mathbf{D}}_i : \underset{\sim}{\alpha}_i$$
$$\dot{r} = \dot{p} - \dot{p}br$$

- The plasticity criterion is given by:

$$f = \left|\left|\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}\right|\right|_M - \sigma_y - R \geq 0$$

σ_y is the size of the initial elastic domaine. The norm $||.||_M$ is used to model plastic anisotropy:

$$\left|\left|\underset{\sim}{\mathbf{a}}\right|\right|_M = \left(\underset{\sim}{\mathbf{a}} : \underset{\approx}{\mathbf{M}} : \underset{\sim}{\mathbf{a}}\right)^{\frac{1}{2}}$$

where $\underset{\approx}{\mathbf{M}}$ is a fourth order tensor such that $\underset{\approx}{\mathbf{M}} : \underset{\sim}{\mathbf{1}} = \underset{\sim}{\mathbf{0}}$.

- In the viscous case (studied in the following)

$$\dot{p} = \phi(f, \dots)$$

- The flow direction (normality) is expressed as:

$$\underset{\sim}{\mathbf{n}} = \frac{\partial f}{\partial \underset{\sim}{\sigma}} = \frac{1}{||\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}||_M} \underset{\approx}{\mathbf{M}} : (\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}})$$

- To compute the Jacobian matrix, the following tensor is also needed:

$$\underset{\approx}{\mathbf{N}} = \frac{\partial \underset{\sim}{\mathbf{n}}}{\partial \sigma} = \frac{\partial^2 f}{\partial \sigma^2} = \frac{1}{||\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}||_M} \left(\underset{\approx}{\mathbf{M}} - \frac{1}{||\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}||_M^2} \underset{\approx}{\mathbf{M}} : (\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}) \otimes \underset{\approx}{\mathbf{M}} : (\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}) \right)$$

- $\underset{\approx}{\mathbf{C}}_i$, $\underset{\approx}{\mathbf{D}}_i$ and $\underset{\approx}{\mathbf{M}}$ are used to model anisotropy. The isotropic case corresponds to: $\underset{\approx}{\mathbf{C}}_i = C_i \underset{\approx}{\mathbf{1}}$, $\underset{\approx}{\mathbf{D}}_i = D_i \underset{\approx}{\mathbf{1}}$ and $\underset{\approx}{\mathbf{M}} = \underset{\approx}{\mathbf{J}}$.
- Runge–Kutta is straigthforward !

Multi–kinematic law : θ–method

- The time discretization of the previous equations leads to:

$$\begin{aligned}
\underset{\sim}{\mathbf{r}}_e &= \Delta\underset{\sim}{\epsilon}_e + \Delta p \underset{\sim}{\mathbf{n}} - \Delta\underset{\sim}{\epsilon}_t = \underset{\sim}{\mathbf{0}} \\
\underset{\sim}{\mathbf{r}}_{\alpha_i} &= \Delta\underset{\sim}{\alpha}_i - \Delta p \, \underset{\sim}{\mathbf{n}} + \Delta p \underset{\approx}{\mathbf{D}}_i : \underset{\sim}{\alpha}_i + \theta D_i \Delta\underset{\sim}{\alpha}_i \, \Delta p = \underset{\sim}{\mathbf{0}} \\
r_r &= \Delta r - \Delta p(1 - br) = 0 \\
r_p &= \Delta p - \phi(f, \dots)\Delta t = 0
\end{aligned}$$

- Variables $\underset{\sim}{\varepsilon}_e$, $\underset{\sim}{\alpha}_i$, r are considered e time $t + \theta\Delta t$ and are equal to: $\underset{\sim}{\varepsilon}_e(t) + \theta\Delta\underset{\sim}{\varepsilon}_e$, $\underset{\sim}{\alpha}_i(t) + \theta\Delta\underset{\sim}{\alpha}_i$, $r + \theta\Delta r$.
- Plastcity can be treated writting:

$$r_p = \left|\left|\underset{\sim}{\sigma} - \underset{\sim}{\mathbf{X}}\right|\right|_M - R - \sigma_y = 0\Big|_1$$

- The Jacobian matrix is expressed in the following slides …

$$\boxed{\underset{\sim}{\mathbf{r}}_e = \Delta\underset{\sim}{\epsilon}_e + \Delta p\underset{\sim}{\mathbf{n}} - \Delta\underset{\sim}{\epsilon}_t = \underset{\sim}{\mathbf{0}}}$$

$$\frac{\partial \underset{\sim}{\mathbf{r}}_e}{\partial \Delta \underset{\sim}{\epsilon}_e} = \underset{\approx}{\mathbf{1}} + \theta \Delta p \underset{\approx}{\mathbf{N}}\,\underset{\approx}{\mathbf{C}}$$

$$\frac{\partial \underset{\sim}{\mathbf{r}}_e}{\partial \Delta \underset{\sim}{\alpha}_i} = \Delta p \frac{\partial \underset{\sim}{\mathbf{n}}}{\partial \underset{\sim}{\mathbf{X}}_i} \frac{\partial \underset{\sim}{\mathbf{X}}_i}{\partial \underset{\sim}{\alpha}_i} \frac{\partial \underset{\sim}{\alpha}_i}{\partial \Delta \underset{\sim}{\alpha}_i} = -\theta \Delta p \underset{\approx}{\mathbf{N}}\,\underset{\approx}{\mathbf{C}}_i$$

$$\frac{\partial \underset{\sim}{\mathbf{r}}_e}{\partial \Delta r} = 0$$

$$\frac{\partial \underset{\sim}{\mathbf{r}}_e}{\partial \Delta p} = \underset{\sim}{\mathbf{n}}$$

$$\boxed{\underset{\sim}{\mathbf{r}}_{\alpha_i} = \Delta \underset{\sim}{\alpha}_i - \Delta p\, \underset{\sim}{\mathbf{n}} + \Delta p \underset{\approx}{\mathbf{D}}_i : \underset{\sim}{\alpha}_i + \theta D_i \Delta \underset{\sim}{\alpha}_i\, \Delta p = \underset{\sim}{\mathbf{0}}}$$

$$\begin{aligned}
\frac{\partial \underset{\sim}{\mathbf{r}}_{\alpha_i}}{\partial \Delta \underset{\sim}{\epsilon}_e} &= -\Delta p \frac{\partial \underset{\sim}{\mathbf{n}}}{\partial \underset{\sim}{\sigma}} \frac{\partial \underset{\sim}{\sigma}}{\partial \underset{\sim}{\epsilon}_e} \frac{\partial \underset{\sim}{\epsilon}_e}{\partial \Delta \underset{\sim}{\epsilon}_e} = -\theta \Delta p \underset{\approx}{\mathbf{N}} \underset{\approx}{\mathbf{C}} \\
\frac{\partial \underset{\sim}{\mathbf{r}}_{\alpha_i}}{\partial \Delta \underset{\sim}{\alpha}_i} &= \underset{\approx}{\mathbf{1}} + \theta \Delta p \underset{\approx}{\mathbf{D}}_i \\
\frac{\partial \underset{\sim}{\mathbf{r}}_{\alpha_i}}{\partial \Delta r} &= 0 \\
\frac{\partial \underset{\sim}{\mathbf{r}}_{\alpha_i}}{\partial \Delta p} &= -\underset{\sim}{\mathbf{n}} + \underset{\approx}{\mathbf{D}}_i : \underset{\sim}{\alpha}_i
\end{aligned}$$

$$\boxed{r_r = \Delta r - \Delta p(1 - br) = 0}$$

$$\begin{aligned}
\frac{\partial r_r}{\partial \Delta \underset{\sim}{\epsilon}_e} &= \underset{\sim}{\mathbf{0}} \\
\frac{\partial r_r}{\partial \Delta \underset{\sim}{\alpha}_i} &= \underset{\sim}{\mathbf{0}} \\
\frac{\partial r_r}{\partial \Delta r} &= 1 + \theta \Delta p d \\
\frac{\partial r_r}{\partial \Delta p} &= b\, r
\end{aligned}$$

$$\boxed{r_p = \Delta p - \phi(f, \ldots)\Delta t}$$

$$\begin{aligned}
\frac{\partial r_p}{\partial \Delta \underset{\sim}{\epsilon}_e} &= -\frac{\partial F}{\partial f}\frac{\partial f}{\partial \underset{\sim}{\sigma}}\frac{\partial \underset{\sim}{\sigma}}{\partial \underset{\sim}{\epsilon}_e}\frac{\partial \underset{\sim}{\epsilon}_e}{\partial \Delta \underset{\sim}{\epsilon}_e} = -\theta \Delta t \phi_{,f}\, \underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}} \\
\frac{\partial r_p}{\partial \Delta \underset{\sim}{\alpha}_i} &= \theta \Delta t \phi_{,f}\, \underset{\sim}{\mathbf{n}} : \underset{\approx}{\mathbf{C}}_i \\
\frac{\partial r_p}{\partial \Delta r} &= -\frac{\partial \phi}{\partial R}\frac{\partial R}{\partial r}\frac{\partial r}{\partial \Delta r}\Delta t = \theta \Delta t c \phi_{,f} \\
\frac{\partial r_p}{\partial \Delta p} &= 1
\end{aligned}$$

Multi–kinematic law : static recovery

Static recovery can easily be added by modifying the evolution laws for the hardening (both isotropic and kinematic) variables:

$$\dot{\underset{\sim}{\alpha}}_i = \dot{\underset{\sim}{\epsilon}}_p - \dot{p}\underset{\approx}{\mathbf{D}}_i : \underset{\sim}{\alpha}_i - \underset{\approx}{\mathbf{S}}_i : \underset{\sim}{\alpha}_i$$

$$\dot{r} = \dot{p} - \dot{p}br - sr$$

In the calculation of the Jacobian matrix, the following terms must be added:

$$-\theta\Delta t\underset{\approx}{\mathbf{S}}_i \quad \text{à} \quad \partial\underset{\sim}{\mathbf{r}}_{\alpha_i}/\partial\Delta\underset{\sim}{\alpha}_i$$

$$-\theta\Delta ts \quad \text{à} \quad \partial\underset{\sim}{\mathbf{r}}_{\alpha_i}/\partial\Delta r$$

Multi–kinematic law : variable temperature

- Material coefficients may depend on external parameters and state varaibles
- These coefficients must be evaluated at t, $t + \theta\Delta t$ or $t + \Delta t$.
- In the case of the Runge–Kutta integration, using a viscous creep law allows to bypass the computation of the plastic multiplier using the consistency condition.
- An error is often done…

The relationships

$$\underset{\sim}{\mathbf{X}} = \frac{2}{3} C \underset{\sim}{\alpha} \qquad \dot{\underset{\sim}{\alpha}} = \dot{\underset{\sim}{\epsilon}}_p - \frac{3}{2} \dot{p} \frac{D}{C} \underset{\sim}{\alpha}$$

are replaced by

$$\dot{\underset{\sim}{\mathbf{X}}} = \frac{2}{3} C \dot{\underset{\sim}{\epsilon}}_p - D \dot{p} \underset{\sim}{\mathbf{X}}$$

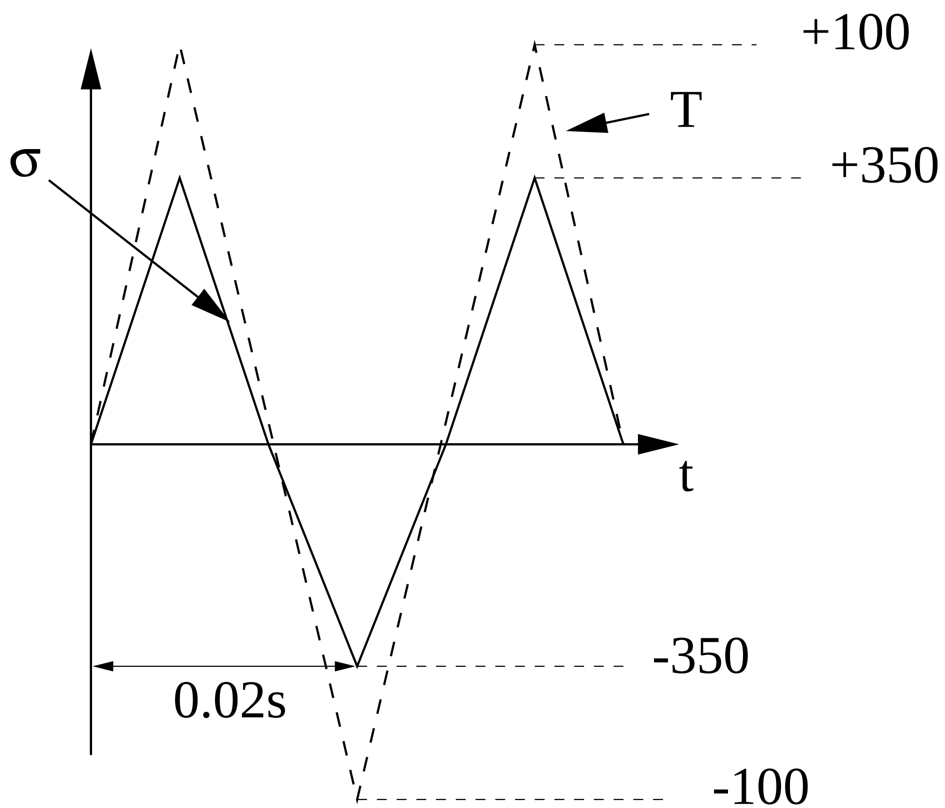
which is only valid if C is a constant. In fact:

$$\dot{\underset{\sim}{\mathbf{X}}} = \frac{2}{3} d(C \underset{\sim}{\alpha})/dt = \frac{2}{3} C_{,T} \dot{T} \underset{\sim}{\alpha} + \frac{2}{3} C \dot{\underset{\sim}{\alpha}}$$

- Comparaison of the results with

$$C = 30000(1 - T/200) \qquad D = 200$$
$$K = 20 \qquad n = 10 \qquad R = 300 \qquad E = 200000$$

and the following load path



In both cases, ratchetting is obtained but the results strongly differ