Two-Step Exact Solution for Modified Mohr-Coulomb Viscoplastic Consistency Model with Linear Hardening/Softening: A Proof with Maple Software

Timo Saksala

Abstract It has been observed numerically that the viscoplastic consistency model by Wang (1997) with a linear yield surface and a linear hardening/softening rule converges, using the standard stress return mapping, with two steps. In this paper this numerical observation is proved analytically using the Maple software. The proof is carried out for the Modified Mohr-Coulomb viscoplastic consistency model in the corner plasticity situation, i.e. when both the Rankine and Mohr-Coulomb criteria are violated.

Keywords Viscoplastic consistency model, Modified Mohr-Coulomb criterion, Stress return mapping, Exact solution

1. Introduction

The classical viscoplasticity formulations, namely the Perzyna and Duvaut-Lions models, do not utilize the consistency condition[1]. Consequently, trial stress states violating the yield criterion are not returned to the yield surface defined by the yield criterion. In those models, the viscoplastic strain rate depends on the amount of violation[1]. In contrast, the consistency condition is imposed in the more recent viscoplastic consistency model by Wang[2]. Thereby, the trial stresses violating the yield function are returned to the yield surface as in the classical rate-independent plasticity. The major difference to the rate-independent plasticity is the dependence of the yield function on the rate of the internal variable(s). The basic components of the bi-surface viscoplastic consistency model are:

\[ f_{vp,i} = f_i^{\prime}(\sigma, \kappa_i, \dot{k}_i) \]

\[ \dot{\varepsilon}^{vp} = \dot{\lambda}_1 \frac{\partial g_{vpi}}{\partial \sigma} + \dot{\lambda}_2 \frac{\partial g_{vpi}}{\partial \sigma} \]

\[ \dot{k}_i = \dot{\lambda}_i k_i (\sigma, \kappa_i) \]

\[ f_{vp,i} \leq 0, \quad \dot{\lambda}_i \geq 0, \quad \dot{\lambda}_i f_{vp,i} = 0, \quad i = 1, 2 \]

where \( \kappa, \kappa_i \) denote the internal variable and its rate, respectively. Moreover, \( g_{vp,i} \) is the viscoplastic potential, \( \dot{\varepsilon}^{vp} \) is the viscoplastic strain tensor (vector), and \( \dot{\lambda}_i \) is the viscoplastic increment, i.e. the rate of viscoplastic multiplier. The yield function \( f_{vp,i} \) can be called a dynamic yield function since it can expand and shrink in relation to its static position depending on the loading rate. Moreover, the loading-unloading conditions of Kuhn-Tucker form must hold.

As a consequence, the robust stress return algorithms of rate-independent computational plasticity can be employed with minor modifications concerning the presence of the rate of the internal variable(s). It is shown in this paper that a two-step exact solution exists for a viscoplasticity problem with a linear bi-surface yield function and linear hardening/softening laws. This fact has, of course, been observed in numerical simulations but a formal proof is given here. The proof is performed for the Modified Mohr-Coulomb (MMC) viscoplastic consistency model in the corner plasticity situation, i.e. when both the Rankine and Mohr-Coulomb criteria are violated. Moreover, the proof is carried out with the Maple technical computing software due to the complexity of the equations involved. The MMC model is chosen due its wide usage in geotechnical engineering analyses. The MC criterion is usually modified with the Rankine, or the principal stress, criterion to predict the correct tensile strength of geomaterials such as rock and concrete.

2. Modified Mohr-Coulomb Viscoplastic Consistency Model

The MMC yield criterion in the principal stress space, assuming the ordering \( \sigma_1 \geq \sigma_2 \geq \sigma_3 \), consists of the MC and Rankine criteria:

\[ f_{mc}(\sigma, \kappa_{mc}, \dot{k}_{mc}) = \sigma_i - \sigma_i + (\sigma_i + \sigma_i) \sin \phi - 2c(\kappa_{mc}, \dot{k}_{mc}) \cos \phi \]

\[ f_{R}(\sigma) = \sigma_1 - f_i(\kappa_i, \dot{k}_i) \]

* Corresponding author: timo.saksala@tut.fi (Timo Saksala)
Published online at http://journal.sapub.org/ajcam
Copyright © 2012 Scientific & Academic Publishing. All Rights Reserved
where $\sigma_i$, $\sigma_j$ are the major and minor principal stresses, $\varphi$ is the internal friction, $c$ and $f_i$ are the cohesion and tensile strength of the material. The rates of the internal variables are related to viscoplastic increments as $\dot{\kappa}_{\text{MC}} = k_{\text{MC}}\dot{\lambda}_{\text{MC}}$ with $k_{\text{MC}} = \sqrt{2}\frac{\partial}{\partial\sigma}g_{\text{MC}}\partial\sigma g_{\text{MC}}$ where symbol $\partial_x$ denotes the derivative with respect to a variable (scalar or vector) $x$. The combined yield surface is illustrated in Figure 1.

$\varphi$ are related to viscoplastic increments as $\dot{\sigma}_i = f_i + h_i\kappa_i + s_i \kappa_i$ where $c_0$ is the initial (intact) value of cohesion while $h_{\text{MC}}$, $h_R$ and $s_{\text{MC}}$, $s_R$ are the softening/hardening and viscosity moduli in compression and tension, respectively.

In order to account for correct dilatation behaviour of the material in compression, the plastic potential is chosen as $g_{\text{MC}}(\sigma) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\psi$. Therefore, since the relation between the internal variables and plastic increments is a constant relation, it is more convenient to use the plastic multipliers and increments in the following derivations.

**Figure 1.** Modified Mohr-Coulomb criterion in principal stress space

The rate-dependent softening/hardening rules for cohesion and tensile strength are assumed to be linear by

$$c = c_0 + h_{\text{MC}}\kappa_{\text{MC}} + s_{\text{MC}}\kappa_{\text{MC}}$$

$$f_i = f_i + h_i\kappa_i + s_i \kappa_i$$

where $c_0$ is the initial value of cohesion while $h_{\text{MC}}$, $h_R$ and $s_{\text{MC}}$, $s_R$ are the softening/hardening and viscosity moduli in compression and tension, respectively.

In order to account for correct dilatation behaviour of the material in compression, the plastic potential is chosen as $g_{\text{MC}}(\sigma) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\psi$. Therefore, since the relation between the internal variables and plastic increments is a constant relation, it is more convenient to use the plastic multipliers and increments in the following derivations.

**Figure 2.** Illustration of the geometry of a corner point

### 3. Return Mapping Scheme for Bisurface Viscoplastic Consistency Model

The intricacy involved in the multisurface plasticity is the stress return to an intersection of yield criteria. At such a point the gradient of the combined yield surface is not unique as illustrated in Figure 2 depicting a corner point in a bi-criteria case.

The set of admissible stresses is denoted by $E_{\sigma}$ in Figure 1. $I_1$ is the region where $\dot{\lambda}_i > 0$, $\dot{\lambda}_j < 0$, in $I_2$, $\dot{\lambda}_i < 0$, $\dot{\lambda}_j > 0$, and in $I_{12}$, $\dot{\lambda}_j > 0$, $\dot{\lambda}_j > 0$. The stress return algorithm should be able to distinguish between these regions and restore the consistency accordingly. This means, e.g., that if the trial stress lies in $I_1$ the single surface integration should be performed in order to return the trial stress to the surface $f_1 = 0$. The determination of the right return can be performed by trial and error only because the condition $f_1^{\text{trial}} > 0$ does not guarantee that eventually $\dot{\lambda}_i > 0$. Here the procedure presented in [3] is employed for determining the active set of yield surfaces.

Stress integration, i.e. return mapping, algorithms for classical single and multi-surface plasticity and viscoplasticity are considered, e.g. in [1,3], while algorithms for consistent viscoplasticity are developed in [2,4,5]. For present purposes, the stress integration algorithm by Wang et al.[2] is utilized as modified for the linear bi-surface case. It is assumed that both criteria are violated.

**Return mapping algorithm for bi-surface viscoplastic consistency model:**

1. **Compute G and F using (5) and solve for $\delta\lambda_1, \delta\lambda_2$** by $\delta\lambda = G^{-1}F$

2. $\Delta\lambda_j^{(i+1)} = \Delta\lambda_j^{(i)} + \delta\lambda_j, \Delta\lambda_{2j}^{(i+1)} = \Delta\lambda_j^{(i)} + \delta\lambda_{2j}$

3. If $\Delta\lambda_j^{(i+1)} > 0$ & $\Delta\lambda_{2j}^{(i+1)} > 0$ go to 4. Else restart iteration with the criterion for which $\Delta\lambda_j^{(i+1)} > 0$.

4. $\dot{\lambda}_j^{(i+1)} = \Delta\lambda_j^{(i+1)}/\Delta t, \dot{\lambda}_{2j}^{(i+1)} = \Delta\lambda_j^{(i+1)}/\Delta t$

5. $\delta\epsilon^{\text{vp}} = \delta\lambda_j\partial_{\epsilon^{\text{vp}}}g_{\text{vp1}} + \delta\lambda_{2j}\partial_{\epsilon^{\text{vp}}}g_{\text{vp2}}$

6. $\Delta\epsilon^{\text{vp}}^{(i+1)} = \Delta\epsilon^{\text{vp}}^{(i)} + \delta\epsilon^{\text{vp}}$

7. $\sigma^{(i+1)} = \sigma^{(i)} - E\Delta\epsilon^{\text{vp}}^{(i+1)}$

8. $f_1^{(i+1)} = f_1(\sigma^{(i+1)}, \dot{\lambda}_j + \Delta\lambda_j^{(i+1)}, \dot{\lambda}_{2j}^{(i+1)})$

9. $f_2^{(i+1)} = f_2(\sigma^{(i+1)}, \dot{\lambda}_j + \Delta\lambda_j^{(i+1)}, \dot{\lambda}_{2j}^{(i+1)})$

If $f_1^{(i+1)} < \text{TOL}$ & $f_2^{(i+1)} < \text{TOL}$ then go to Update. Else set $i ← i + 1$ and go to 1.

**Update:** $\sigma^{(i+1)} = \sigma^{(i)} + \epsilon^{\text{vp}} = \epsilon^{\text{vp}} + \Delta\epsilon^{\text{vp}}^{(i+1)}, \dot{\lambda}_j + \Delta\lambda_j^{(i+1)}, \dot{\lambda}_{2j}^{(i+1)}$. 

**Figure 2. Illustration of the geometry of a corner point**
\[ \lambda_{2, t + \Delta t} = \lambda_{2, t} + \Delta \lambda_{2, t}^{(1)} \] and exit.

The expressions for \( G \) and \( F \) needed in step 1 of the algorithm, derived in Appendix A, are:

\[ G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \]
\[ G_{11} = \left( \hat{\varepsilon}_a \alpha_{s, \text{sp1}} \right)^	op \mathbf{E} \hat{\varepsilon}_a \alpha_{s, 1} + (h_1 + s_1 / \Delta t) k_1 \\
G_{12} = \left( \hat{\varepsilon}_a \alpha_{s, \text{sp2}} \right)^	op \mathbf{E} \hat{\varepsilon}_a \alpha_{s, 2} \\
G_{21} = \left( \hat{\varepsilon}_a \alpha_{s, \text{sp1}} \right)^	op \mathbf{E} \hat{\varepsilon}_a \alpha_{s, 2} \\
G_{22} = \left( \hat{\varepsilon}_a \alpha_{s, \text{sp2}} \right)^	op \mathbf{E} \hat{\varepsilon}_a \alpha_{s, 2} + (h_2 + s_2 / \Delta t) k_2 \] (5)

The purpose of step 3 in the algorithm is to ensure that a genuine corner plasticity situation is realized. This can be performed on trial and error only, as noted above. If one of the cumulative plastic increments is found non-positive, the trial stress was located in one of the regions denoted by \( \Gamma_1 \) and \( \Gamma_2 \) in Figure 2 and, thus, no genuine corner plasticity has occurred (i.e. the trial stress was not located in region \( \Gamma_{12} \)). This means that the bi-surface iteration must be terminated and a single surface integration should be performed to return the trial stress onto the correct yield surface as explained above.

In fact, in this form the algorithm above is similar to the generalized cutting plane algorithm \[3\]. Therefore, it is not limited to linear yield surfaces but can be used for stress integration of nonlinear models as well.

4. A Two-Step Solution for the MMC Viscoplasticity Problem

It is convenient for present purposes to write the MC criterion with its viscoplastic potential in the following simpler form:

\[ f_{MC}(\sigma, \lambda_{MC}, \dot{\lambda}_{MC}) = k \sigma_1 - \sigma_3 - f_4(\lambda_{MC}, \dot{\lambda}_{MC}) \]
\[ g_{MC}(\sigma) = m \sigma_1 - \sigma_3 \]
\[ k = \frac{1 + \sin \phi}{1 - \sin \phi}, \quad m = \frac{1 + \sin \psi}{1 - \sin \psi} \] (6)

where \( f_4 \) is the uniaxial compressive strength which is substituted for \( c \) in Equation (3). In addition, the constant \( k_{MC} \) becomes now \( k^\text{MC} = \sqrt{\frac{1}{1 + m}} \).

Next, it is shown by Maple software that a two-step exact solution, using the algorithm above, exists for the MMC viscoplasticity problem in the corner plasticity situation with the linear hardening/softening rules (3). The proof begins at the onset of viscoplasticity. Vector notation is used and only the relevant intermediate results of the Maple command executions are shown.

First time step: transition from elasticity to viscoplasticity, i.e. \( \varepsilon^\text{p} = 0 \), \( \lambda_{MC,j} = 0 \), \( \lambda_{R,j} = 0 \). is treated as shown in Figure 3.

1> restart:
2> with(linalg):
3> Em:=matrix([[E1,E2,E2], [E2,E1,E2], [E2,E2,E1]]);
4> FMC:=k*s1-s3: gMC:=m*s1-s3:
5> Fr:=s1-ft:
6> dfr:=vector([diff(Em,s1),diff(Em,s2),diff(Em,s3)]):
7> dFMC:=vector([diff(FMC,s1),diff(FMC,s2),diff(FMC,s3)]):
8> ddFMC:=vector([diff(gMC,s1),diff(gMC,s2),diff(gMC,s3)]):
9> lmandMC[0]:0: lmandR[0]:0: DlmandMC[0]:0: DlmandR[0]:0:
10> Denspv:=vector([0,0,0,0]):
11> F:=vector([FMC,F]):
12> G11:=evalv(transpose(dFMC) &* (evalv(Em &* ddFMC))+ h MC*s MC/dt):
13> G12:=evalv(transpose(dFMC) &* (evalv(Em &* dFMC)))+ h MC*s MC/dt:
14> G21:=evalv(transpose(dFMC) &* (evalv(Em &* ddFMC)))+ h MC*s MC/dt:
15> G22:=evalv(transpose(dFMC) &* (evalv(Em &* dFMC)))+ h MC*s MC/dt:
16> G:=matrix([[G11,G12],[G12,G22]]):
17> dl:=simplify(evalv(transpose(G) &* F)):
18> DlmandMC:=DlmandMC[0]+Dl[M1]: DlmandR:=DlmandR[0]+Dl[M2]:
19> lmandare:=lmandare[0]: Dlmandare:=Dlmandare[0]: Dlmandare[0]:
20> depspv:=evalv(transpose(Dl[M1])&*Dl[M2]):
21> Depspv:=evalv(Depspv &* Depspv):
22> signal:=evalv(0.0):
23> ftc:=ft+h MC&*lmandare[0]+s R&*lmandare:
24> fcl:=fc+h MC&*lmandare[0]+s MC&*lmandare:
25> simplify(subs(s,[s1,s2,s3],[s1,s2,s3])).
26> simplify(subs(s=[s1,s2,s3],fcl,FC)):

Figure 3. Maple code for the first iteration involving a transition from elasticity to viscoplasticy

According to lines 25 and 26 of the Maple code in Figure 3, \( f_{MC}(\varepsilon^\text{p}, \Delta \lambda_{MC}, \Delta \dot{\lambda}_{MC}) = 0 \) and \( f_4(\sigma^\text{p}, \Delta \lambda_{MC}, \Delta \dot{\lambda}_{MC}) = 0 \). Therefore, at the onset of viscoplasticity, a single step solution exists. Before continuing, the above Maple script is elaborated in some detail. After loading the linear algebra package in line 2, the linear elasticity matrix is defined in line 3 with \( E_1 = E(1 - \nu)/(1 + \nu)(1 - 2\nu) \) and \( E_2 = E\nu/(1 + \nu)(1 - 2\nu) \). Lines 4-8 define the trial values of the MC and Rankine criteria, the plastic potential (for MC criterion), and their gradients determined with the \text{diff} command. In lines 9 and 10 the initial values are set for the viscoplastic multipliers and strain. Vector \( F \) and matrix \( G \) according to Equation (5) are defined in lines 11-16. Then, the viscoplastic increments are solved in line 17 corresponding to step 1 in the algorithm. Lines 18 and 19 corresponds to steps 2 and 4 in the algorithm. Lines 20 and 21 do the same operations as steps 5 and 6 in the algorithm. Line 22 corrects the stress according to step 7. Lines 23, 24 update the values of uniaxial tensile and compressive strengths and, finally, lines 25 and 26 compute the new values of the yield criteria (step 8 in the algorithm).

In the next time step: \( \varepsilon^\text{p}_t = \Delta \varepsilon^\text{p}_t, \lambda_{MC,t} = \Delta \lambda_{MC,t} \) and \( \lambda_{R,t} = \Delta \lambda_{R,t} \). This is a general time step involving softening/hardening and the strain rate effects as well. The Maple code is shown in Figure 4.
In line 31 above \( s_1 \) and \( s_2 \) denote the trial stress state. Moreover, the Maple commands in lines 41 and 42 yield results that can be written in form:

\[
\begin{align*}
&\frac{f_{MC2new}}{f_{MC1new}} = MC_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&\frac{f_{R2new}}{f_{R1new}} = R_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
\end{align*}
\]

with

\[
\begin{align*}
&f_{MC1new} = s_1 (f_1 + \frac{1}{3} c_1 s_1) + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R1new} = s_1 (f_1 + \frac{1}{3} c_1 s_1) + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
\end{align*}
\]

Thus, in the end of the first iteration the values of the yield functions are given by (7). Accordingly, the exact solution is obtained with the first step if the viscosity moduli \( s_{MC} \) and \( s_{R} \) are set to zero which is the rate-independent case. In the rate-dependent (viscous) case another step is needed:

\[
\begin{align*}
&f_{MC2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3.
\end{align*}
\]

Thus, in the end of the first iteration the values of the yield functions are given by (7). Accordingly, the exact solution is obtained with the first step if the viscosity moduli \( s_{MC} \) and \( s_{R} \) are set to zero which is the rate-independent case. In the rate-dependent (viscous) case another step is needed:

\[
\begin{align*}
&f_{MC2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3.
\end{align*}
\]

Thus, in the end of the first iteration the values of the yield functions are given by (7). Accordingly, the exact solution is obtained with the first step if the viscosity moduli \( s_{MC} \) and \( s_{R} \) are set to zero which is the rate-independent case. In the rate-dependent (viscous) case another step is needed:

\[
\begin{align*}
&f_{MC2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3.
\end{align*}
\]

Thus, in the end of the first iteration the values of the yield functions are given by (7). Accordingly, the exact solution is obtained with the first step if the viscosity moduli \( s_{MC} \) and \( s_{R} \) are set to zero which is the rate-independent case. In the rate-dependent (viscous) case another step is needed:

\[
\begin{align*}
&f_{MC2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3.
\end{align*}
\]

Thus, in the end of the first iteration the values of the yield functions are given by (7). Accordingly, the exact solution is obtained with the first step if the viscosity moduli \( s_{MC} \) and \( s_{R} \) are set to zero which is the rate-independent case. In the rate-dependent (viscous) case another step is needed:

\[
\begin{align*}
&f_{MC2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3,
&f_{R2new} = s_1 (f_1 + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3) + \frac{1}{3} c_1 s_1 + \frac{1}{3} c_3 s_3 + \frac{1}{3} c_1 s_2 + \frac{1}{3} c_3 s_3.
\end{align*}
\]
Following Winnicki et al. [5], it is assumed that
\[
\dot{\lambda}_i^{i+M} = \Delta \dot{\lambda}_i^{i+M} / \Delta t
\]
by which
\[
k_i^{i+M} = k_i + \Delta \dot{\lambda}_i^{i+M} \dot{k}_i
\]
(A4)

By (A4) the yield functions \( f_{vi} \) are formulated to depend on \( \lambda + \Delta \lambda \):
\[
f_{vi} = g_{vi}(\sigma, \lambda + \Delta \lambda).
\]
Hence, \( F \) can be expanded with the first term of the vector valued Taylor series:
\[
F(\Delta \lambda + \delta \Delta \lambda) = F(\Delta \lambda) + \nabla_{\Delta \lambda} F(\Delta \lambda) \delta \Delta \lambda.
\]
(A5)

Computing the gradient of \( f_{vi} \) with respect to \( \Delta \lambda \) yields (the derivation for \( f_{v2} \) is identical)
\[
\nabla_{\Delta \lambda} f_{vi} = \begin{bmatrix}
\left( \frac{\partial f_{vi}}{\partial \sigma} \right)^T \frac{\partial \sigma}{\partial \Delta \lambda} + \left( \frac{\partial f_{vi}}{\partial \sigma} \right)^T \frac{\partial \sigma}{\partial \Delta \lambda}
\left( \frac{\partial f_{vi}}{\partial \sigma} \right)^T \frac{\partial \sigma}{\partial \Delta \lambda} + \left( \frac{\partial f_{vi}}{\partial \sigma} \right)^T \frac{\partial \sigma}{\partial \Delta \lambda}
\end{bmatrix}
\]
(A6)

It is assumed that \( g_{vi} \) is independent of \( \Delta \lambda \) and \( f_1 \) is independent of \( \Delta \lambda_2 \). By this, and making use of (A2)-(A4), the first row of the gradient (A6) can be written as
\[
\nabla_{\Delta \lambda} f_{vi} = \begin{bmatrix}
\left( \frac{\partial f_{vi}}{\partial \sigma} \right)^T \frac{\partial \sigma}{\partial \Delta \lambda} & \frac{\partial f_{vi}}{\partial \Delta \lambda} & \frac{\partial f_{vi}}{\partial \Delta \lambda}
\end{bmatrix}
\]
(A7)

Now, on solving for \( \delta \Delta \lambda \) and denoting the gradient with \( G \) the desired relation is obtained as
\[
F(\Delta \lambda) + \nabla_{\Delta \lambda} F(\Delta \lambda) \delta \Delta \lambda = 0
\]
\[
\iff \delta \Delta \lambda = -\nabla_{\Delta \lambda} F(\Delta \lambda)^{-1} F(\Delta \lambda) = G^{-1} F(\Delta \lambda)
\]
(A8)
where \( G \) is a 2×2 matrix with the entries in (5).

REFERENCES
[1] Sluys D.R.J., Wave propagation, localisation and dispersion is softening solids, PhD Thesis, Delft University of Technology, 1992.
[2] Wang W.M., Sluys L.J., De Borst R., Viscoelasticity for instabilities due to strain softening and strain-rate softening, International Journal for Numerical Methods in Engineering 1997; 40: 3839-3864.
[3] Simo J.C., Hughes T.J.R., Computational inelasticity, Springer, 1998.
[4] Barpi F., Impact behaviour of concrete: a computational approach. Engineering Fracture Mechanics 2004; 71: 2197-2213.
[5] Winnicki A., Pearce C.J., Bičanić N., Viscoelastic Hoffman consistency model for concrete. Computers & Structures 2001; 79: 7-19.