DOI: 10.5923/j.ajcam.20120201.08

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

Timo Saksala

Department of Mechanics and Design, Tampere University of Technology, Tampere, P.O. Box 589, FI-33101, Finland

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:

$$
\begin{aligned}
&f_{\mathrm{vp},i} = f_i(\boldsymbol{\sigma}, \kappa_i, \dot{\kappa}_i) \\
&\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = \dot{\lambda}_1 \frac{\partial g_{\mathrm{vp1}}}{\partial \boldsymbol{\sigma}} + \dot{\lambda}_2 \frac{\partial g_{\mathrm{vp2}}}{\partial \boldsymbol{\sigma}} \\
&\dot{\kappa}_i = \dot{\lambda}_i k_i(\boldsymbol{\sigma}, \kappa_i), \\
&f_{\mathrm{vp},i} \leq 0, \quad \dot{\lambda}_i \geq 0, \quad \dot{\lambda}_i f_{\mathrm{vp},i} = 0, \quad i = 1,2
\end{aligned}
\tag{1}
$$

where $\kappa_i, \dot{\kappa}_i$ denote the internal variable and its rate, respectively. Moreover, $g_{\mathrm{vp},i}$ is the viscoplastic potential, $\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}$ is the viscoplastic strain tensor (vector), and $\dot{\lambda}_i$ is the viscoplastic increment, i.e. the rate of viscoplastic multiplier. The yield function $f_{\mathrm{vp},i}$ 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:

$$
\begin{aligned}
f_{\mathrm{MC}}(\boldsymbol{\sigma}, \kappa_{\mathrm{MC}}, \dot{\kappa}_{\mathrm{MC}}) &= \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - \\
&\qquad 2c(\kappa_{\mathrm{MC}}, \dot{\kappa}_{\mathrm{MC}})\cos\phi \\
f_{\mathrm{R}}(\boldsymbol{\sigma}) &= \sigma_1 - f_{\mathrm{t}}(\kappa_{\mathrm{R}}, \dot{\kappa}_{\mathrm{R}})
\end{aligned}
\tag{2}
$$

* Corresponding author:
timo.saksala@tut.fi (Timo Saksala)

Published online at http://journal.sapub.org/ajcam

where σ_1, σ_3 are the major and minor principal stresses, φ is the internal friction, c and f_t are the cohesion and tensile strength of the material. The rates of the internal variables are related to viscoplastic increments as $\dot{\kappa}_{\mathrm{R}} = \dot{\lambda}_{\mathrm{R}}$, $\dot{\kappa}_{\mathrm{MC}} = k_{\mathrm{MC}}\dot{\lambda}_{\mathrm{MC}}$ with $k_{\mathrm{MC}} = \sqrt{\tfrac{2}{3}\partial_{\boldsymbol{\sigma}} g_{\mathrm{MC}}{}^{\mathrm{T}} \partial_{\boldsymbol{\sigma}} g_{\mathrm{MC}}}$ where symbol ∂_x denotes the derivative with respect to a variable (scalar or vector) x. The combined yield surface is illustrated in Figure 1.

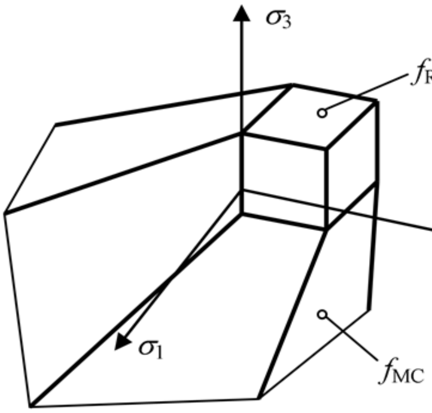


Figure 1. Modified Mohr-Coulomb criterion in principal stress space

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

$$\begin{aligned} c &= c_0 + h_{\mathrm{MC}}\kappa_{\mathrm{MC}} + s_{\mathrm{MC}}\dot{\kappa}_{\mathrm{MC}} \\ f_{\mathrm{t}} &= f_{\mathrm{t}0} + h_{\mathrm{R}}\kappa_{\mathrm{R}} + s_{\mathrm{R}}\dot{\kappa}_{\mathrm{R}} \end{aligned} \tag{3}$$

where c_0 is the initial (intact) value of cohesion while h_{MC}, h_{R} and s_{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

$$g_{\mathrm{MC}}(\boldsymbol{\sigma}) = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\psi \tag{4}$$

where ψ is the dilatation angle. Due to its linearity, the gradients of the MC yield surface and plastic potential are constant vectors (having the angles φ and ψ as parameters). With this choice for the plastic potential, the expression for k_{MC} simplifies as $k_{\mathrm{MC}} = \sqrt{\tfrac{2}{3}(4 - 2\cos\psi)}$. 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.

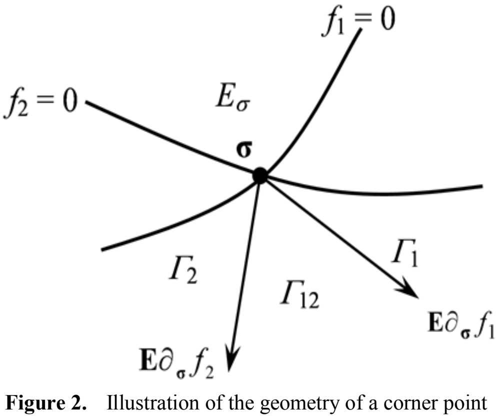

$f_1 = 0$
$f_2 = 0$
E_σ
$\boldsymbol{\sigma}$
Γ_1
Γ_2
Γ_{12}
$\mathbf{E}\partial_{\boldsymbol{\sigma}} f_1$
$\mathbf{E}\partial_{\boldsymbol{\sigma}} f_2$

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_σ in Figure 1. Γ_1 is the region where $\dot{\lambda}_1 > 0, \dot{\lambda}_2 < 0$, in Γ_2, $\dot{\lambda}_1 < 0, \dot{\lambda}_2 > 0$, and in Γ_{12}, $\dot{\lambda}_1 > 0, \dot{\lambda}_2 > 0$. 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 Γ_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 $f_i^{\mathrm{trial}} > 0$ 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:

```
Compute trial state: $\boldsymbol{\sigma}_{\mathrm{trial}} = \mathbf{E}(\boldsymbol{\varepsilon}_{t+\Delta t} - \boldsymbol{\varepsilon}_t^{\mathrm{vp}}), f_{\mathrm{vp1}}^{\mathrm{trial}} = f_1(\boldsymbol{\sigma}_{\mathrm{trial}}, \lambda_{t1}, \dot{\lambda}_{t1}),$
$f_{\mathrm{vp2}}^{\mathrm{trial}} = f_2(\boldsymbol{\sigma}_{\mathrm{trial}}, \lambda_{t2}, \dot{\lambda}_{t2})$
Set: $\Delta\lambda_{1,t}^{(0)} = 0, \Delta\lambda_{2,t}^{(0)} = 0, \boldsymbol{\sigma}_{\mathrm{cor}}^{(0)} = \boldsymbol{\sigma}_{\mathrm{trial}}, \dot{\lambda}_{1,t}^{(0)} = 0, \dot{\lambda}_{2,t}^{(0)} = 0,$
$\Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(0)} = \mathbf{0}$ and go to 1.
Local iteration:
  1. Compute G and F using (5) and solve for $\delta\lambda_1, \delta\lambda_2$
     by $\delta\boldsymbol{\lambda} = \mathbf{G}^{-1}\mathbf{F}$
  2. $\Delta\lambda_{1,t}^{(i+1)} = \Delta\lambda_{1,t}^{(i)} + \delta\lambda_1, \Delta\lambda_{2,t}^{(i+1)} = \Delta\lambda_{2,t}^{(i)} + \delta\lambda_2$
  3. If $\Delta\lambda_{1,t}^{(i+1)} > 0 \,\&\, \Delta\lambda_{2,t}^{(i+1)} > 0$ go to 4. Else restart
     iteration with the criterion for which $\Delta\lambda_{i,t}^{(i+1)} > 0$.
  4. $\dot{\lambda}_{1,t}^{(i+1)} = \Delta\lambda_{1,t}^{(i+1)} / \Delta t, \dot{\lambda}_{2,t}^{(i+1)} = \Delta\lambda_{2,t}^{(i+1)} / \Delta t$
  5. $\delta\boldsymbol{\varepsilon}^{\mathrm{vp}} = \delta\lambda_1 \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp1}} + \delta\lambda_2 \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp2}}$
  6. $\Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(i+1)} = \Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(i)} + \delta\boldsymbol{\varepsilon}^{\mathrm{vp}}$
  7. $\boldsymbol{\sigma}_{\mathrm{cor}}^{(i+1)} = \boldsymbol{\sigma}_{\mathrm{trial}} - \mathbf{E}\Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(i+1)}$
  8. $f_{\mathrm{vp1}}^{(i+1)} = f_1(\boldsymbol{\sigma}_{\mathrm{cor}}^{(i+1)}, \lambda_{1,t} + \Delta\lambda_{1,t}^{(i+1)}, \dot{\lambda}_{1,t}^{(i+1)})$
     $f_{\mathrm{vp2}}^{(i+1)} = f_2(\boldsymbol{\sigma}_{\mathrm{cor}}^{(i+1)}, \lambda_{2,t} + \Delta\lambda_{2,t}^{(i+1)}, \dot{\lambda}_{2,t}^{(i+1)})$
  If $f_{\mathrm{vp1}}^{(i+1)} < TOL \& f_{\mathrm{vp2}}^{(i+1)} < TOL$ then go to Update.
  Else set $i \leftarrow i+1$ and go to 1.
Update: $\boldsymbol{\sigma}_{t+\Delta t} = \boldsymbol{\sigma}_{\mathrm{cor}}^{(i+1)}, \boldsymbol{\varepsilon}_{t+\Delta t}^{\mathrm{vp}} = \boldsymbol{\varepsilon}_t^{\mathrm{vp}} + \Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(i+1)}, \lambda_{1,t+\Delta t} = \lambda_{1,t} + \Delta\lambda_{1,t}^{(i+1)},$
```

$\lambda_{2,t+\Delta t} = \lambda_{2,t} + \Delta\lambda_{2,t}^{(i+1)}$ and exit.

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

$$\mathbf{F} = \begin{bmatrix} f_{\mathrm{vp}1}^{(i)} & f_{\mathrm{vp}2}^{(i)} \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

$$\begin{aligned} G_{11} &= \left(\partial_{\boldsymbol{\sigma}} f_{\mathrm{vp}1}\right)^{\mathrm{T}} \mathbf{E} \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp}1} + (h_1 + s_1 / \Delta t) k_1 \\ G_{12} &= \left(\partial_{\boldsymbol{\sigma}} f_{\mathrm{vp}1}\right)^{\mathrm{T}} \mathbf{E} \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp}2} \\ G_{21} &= \left(\partial_{\boldsymbol{\sigma}} f_{\mathrm{vp}2}\right)^{\mathrm{T}} \mathbf{E} \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp}1} \\ G_{22} &= \left(\partial_{\boldsymbol{\sigma}} f_{\mathrm{vp}2}\right)^{\mathrm{T}} \mathbf{E} \partial_{\boldsymbol{\sigma}} g_{\mathrm{vp}2} + (h_2 + s_2 / \Delta t) k_2 \end{aligned} \tag{5}$$

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 Γ_1 and Γ_2 in Figure 2 and, thus, no genuine corner plasticity has occurred (i.e. the trial stress was not located in region Γ_{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

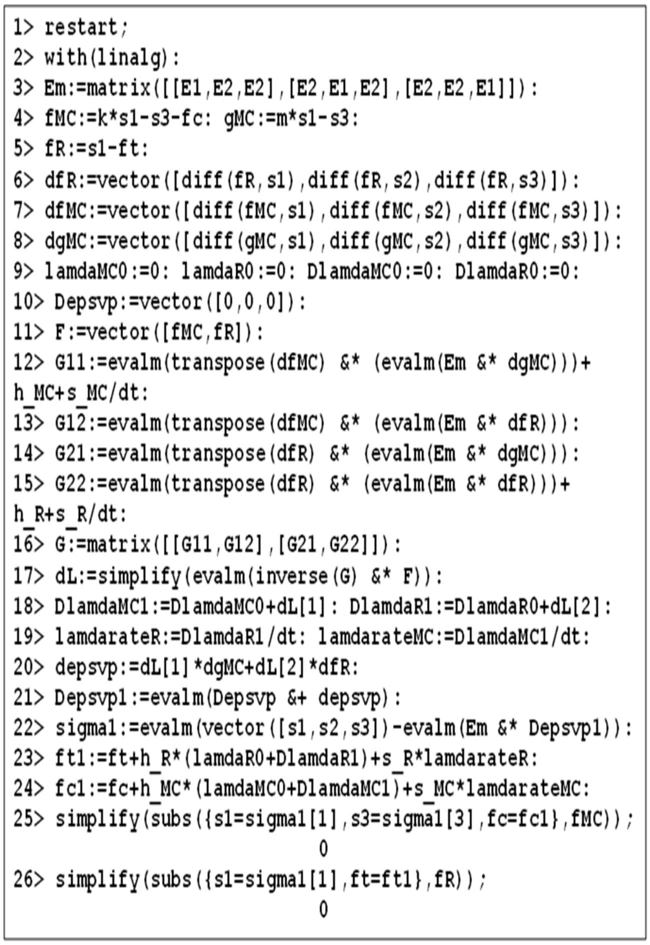
It is convenient for present purposes to write the MC criterion with its viscoplastic potential in the following simpler form:

$$\begin{aligned} f_{\mathrm{MC}}(\boldsymbol{\sigma}, \lambda_{\mathrm{MC}}, \dot{\lambda}_{\mathrm{MC}}) &= k\sigma_1 - \sigma_3 - f_{\mathrm{c}}(\lambda_{\mathrm{MC}}, \dot{\lambda}_{\mathrm{MC}}) \\ g_{\mathrm{MC}}(\boldsymbol{\sigma}) &= m\sigma_1 - \sigma_3 \\ k = \frac{1+\sin\phi}{1-\sin\phi}&, \quad m = \frac{1+\sin\psi}{1-\sin\psi} \end{aligned} \tag{6}$$

where f_{c} is the uniaxial compressive strength which is substituted for c in Equation (3). In addition, the constant k_{MC} becomes now $k_{\mathrm{MC}} = \sqrt{\frac{2}{3}(1+m)}$.

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. $\boldsymbol{\varepsilon}_t^{\mathrm{vp}} = \mathbf{0}, \lambda_{\mathrm{MC},t} = 0, \lambda_{\mathrm{R},t} = 0$, is treated as shown in Figure 3.

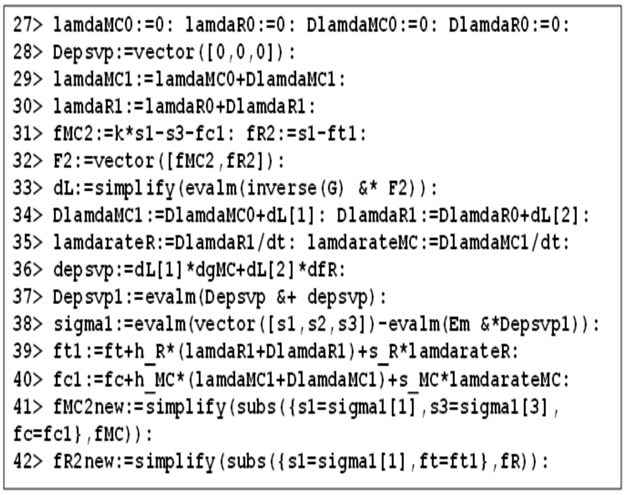


```
1> restart;
2> with(linalg):
3> Em:=matrix([[E1,E2,E2],[E2,E1,E2],[E2,E2,E1]]):
4> fMC:=k*s1-s3-fc: gMC:=m*s1-s3:
5> fR:=s1-ft:
6> dfR:=vector([diff(fR,s1),diff(fR,s2),diff(fR,s3)]):
7> dfMC:=vector([diff(fMC,s1),diff(fMC,s2),diff(fMC,s3)]):
8> dgMC:=vector([diff(gMC,s1),diff(gMC,s2),diff(gMC,s3)]):
9> lamdaMC0:=0: lamdaR0:=0: DlamdaMC0:=0: DlamdaR0:=0:
10> Depsvp:=vector([0,0,0]):
11> F:=vector([fMC,fR]):
12> G11:=evalm(transpose(dfMC) &* (evalm(Em &* dgMC)))+
h_MC+s_MC/dt:
13> G12:=evalm(transpose(dfMC) &* (evalm(Em &* dfR))):
14> G21:=evalm(transpose(dfR) &* (evalm(Em &* dgMC))):
15> G22:=evalm(transpose(dfR) &* (evalm(Em &* dfR)))+
h_R+s_R/dt:
16> G:=matrix([[G11,G12],[G21,G22]]):
17> dL:=simplify(evalm(inverse(G) &* F)):
18> DlamdaMC1:=DlamdaMC0+dL[1]: DlamdaR1:=DlamdaR0+dL[2]:
19> lamdarateR:=DlamdaR1/dt: lamdarateMC:=DlamdaMC1/dt:
20> depsvp:=dL[1]*dgMC+dL[2]*dfR:
21> Depsvp1:=evalm(Depsvp &+ depsvp):
22> sigma1:=evalm(vector([s1,s2,s3])-evalm(Em &* Depsvp1)):
23> ft1:=ft+h_R*(lamdaR0+DlamdaR1)+s_R*lamdarateR:
24> fc1:=fc+h_MC*(lamdaMC0+DlamdaMC1)+s_MC*lamdarateMC:
25> simplify(subs({s1=sigma1[1],s3=sigma1[3],fc=fc1},fMC));
                              0
26> simplify(subs({s1=sigma1[1],ft=ft1},fR));
                              0
```

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

According to lines 25 and 26 of the Maple code in Figure 3, $f_{\mathrm{MC}}(\boldsymbol{\sigma}_{\mathrm{cor}}^{(1)}, \Delta\lambda_{\mathrm{MC}}^{(1)}, \dot{\lambda}_{\mathrm{MC},t}^{(1)}) = 0$ and $f_{\mathrm{R}}(\boldsymbol{\sigma}_{\mathrm{cor}}^{(1)}, \Delta\lambda_{\mathrm{R}}^{(1)}, \dot{\lambda}_{\mathrm{R},t}^{(1)}) = 0$. 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 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: $\boldsymbol{\varepsilon}_{t+\Delta t}^{\mathrm{vp}} = \Delta\boldsymbol{\varepsilon}^{\mathrm{vp},(1)}, \lambda_{\mathrm{MC},t} = \Delta\lambda_{\mathrm{MC}}^{(1)}$ and $\lambda_{\mathrm{R},t} = \Delta\lambda_{\mathrm{R}}^{(1)}$. This is a general time step involving softening/hardening and the strain rate effects as well. The Maple code is shown in Figure 4.



```
27> lamdaMC0:=0: lamdaR0:=0: DlamdaMC0:=0: DlamdaR0:=0:
28> Depsvp:=vector([0,0,0]):
29> lamdaMC1:=lamdaMC0+DlamdaMC1:
30> lamdaR1:=lamdaR0+DlamdaR1:
31> fMC2:=k*s1-s3-fc1: fR2:=s1-ft1:
32> F2:=vector([fMC2,fR2]):
33> dL:=simplify(evalm(inverse(G) &* F2)):
34> DlamdaMC1:=DlamdaMC0+dL[1]: DlamdaR1:=DlamdaR0+dL[2]:
35> lamdarateR:=DlamdaR1/dt: lamdarateMC:=DlamdaMC1/dt:
36> depsvp:=dL[1]*dgMC+dL[2]*dfR:
37> Depsvp1:=evalm(Depsvp &+ depsvp):
38> sigma1:=evalm(vector([s1,s2,s3])-evalm(Em &*Depsvp1)):
39> ft1:=ft+h_R*(lamdaR1+DlamdaR1)+s_R*lamdarateR:
40> fc1:=fc+h_MC*(lamdaMC1+DlamdaMC1)+s_MC*lamdarateMC:
41> fMC2new:=simplify(subs({s1=sigma1[1],s3=sigma1[3],
fc=fc1},fMC)):
42> fR2new:=simplify(subs({s1=sigma1[1],ft=ft1},fR)):
```

Figure 4. Maple code for the first iteration involving viscoplasticity

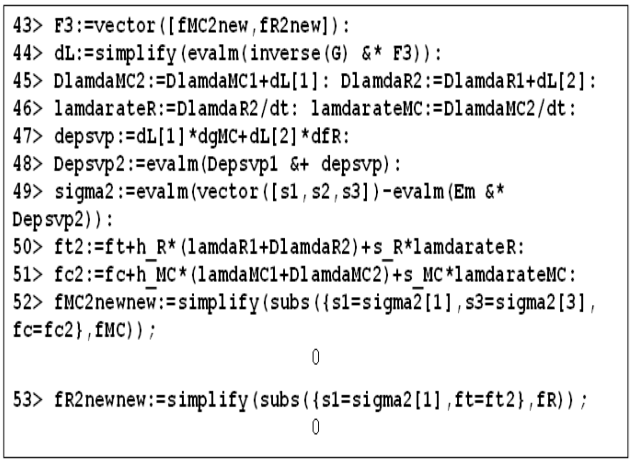
In line 31 above s1 and s2 denote the trial stress state. Moreover, the Maple commands in lines 41 and 42 yield results that can be written in form:

$$
\begin{aligned}
f_{\mathrm{MC2new}} &= s_{\mathrm{MC}}\big((kf_{\mathrm{t}}-\sigma_3-f_{\mathrm{c}})\Delta tE_1+(\sigma_1-f_{\mathrm{t}})\Delta tE_2+\\
&\quad (k\sigma_1-\sigma_3+f_{\mathrm{c}})(h_{\mathrm{R}}\Delta t+s_{\mathrm{R}})\big)/\,DeN\\
f_{\mathrm{R2new}} &= s_{\mathrm{R}}\big((f_{\mathrm{t}}+(kf_{\mathrm{t}}-f_{\mathrm{c}}-\sigma_3)m-\sigma_1)\Delta tE_1+\\
&\quad (f_{\mathrm{c}}+\sigma_3+m\sigma_1-mf_{\mathrm{t}}-kf_{\mathrm{t}})\Delta tE_2+\\
&\quad (s_{\mathrm{MC}}-h_{\mathrm{MC}}\Delta t)f_{\mathrm{t}}-(h_{\mathrm{MC}}\Delta t+s_{\mathrm{MC}})\sigma_1\big)/\,DeN
\end{aligned}
\tag{7}
$$

with

$$
\begin{aligned}
DeN = \big(&E_1\Delta t+(h_{\mathrm{MC}}+kmh_{\mathrm{R}}+h_{\mathrm{R}})\Delta t+kms_{\mathrm{R}}+s_{\mathrm{R}}+\\
&s_{\mathrm{MC}}\big)E_1\Delta t-\big(E_2\Delta t+(mh_{\mathrm{R}}+kh_{\mathrm{R}})\Delta t+ms_{\mathrm{R}}+\\
&ks_{\mathrm{R}}\big)E_2\Delta t+(h_{\mathrm{R}}h_{\mathrm{MC}}\Delta t+s_{\mathrm{MC}}h_{\mathrm{R}}+h_{\mathrm{MC}}s_{\mathrm{R}})\Delta t+\\
&s_{\mathrm{MC}}s_{\mathrm{R}}
\end{aligned}
\tag{8}
$$

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:



```
43> F3:=vector([fMC2new,fR2new]):
44> dL:=simplify(evalm(inverse(G) &* F3)):
45> DlamdaMC2:=DlamdaMC1+dL[1]: DlamdaR2:=DlamdaR1+dL[2]:
46> lamdarateR:=DlamdaR2/dt: lamdarateMC:=DlamdaMC2/dt:
47> depsvp:=dL[1]*dgMC+dL[2]*dfR:
48> Depsvp2:=evalm(Depsvp1 &+ depsvp):
49> sigma2:=evalm(vector([s1,s2,s3])-evalm(Em &*
Depsvp2)):
50> ft2:=ft+h_R*(lamdaR1+DlamdaR2)+s_R*lamdarateR:
51> fc2:=fc+h_MC*(lamdaMC1+DlamdaMC2)+s_MC*lamdarateMC:
52> fMC2newnew:=simplify(subs({s1=sigma2[1],s3=sigma2[3],
fc=fc2},fMC));
                                  0

53> fR2newnew:=simplify(subs({s1=sigma2[1],ft=ft2},fR));
                                  0
```

Figure 5. Maple code for the second iteration involving viscoplasticity

According to lines 52 and 53 of the Maple code:

$$
\begin{aligned}
f_{\mathrm{MC}}(\boldsymbol{\sigma}_{\mathrm{cor}}^{(2)},\lambda_{\mathrm{MC},t}+\Delta\lambda_{\mathrm{MC}}^{(2)},\dot{\lambda}_{\mathrm{MC},t}^{(2)})=0\\
f_{\mathrm{R}}(\boldsymbol{\sigma}_{\mathrm{cor}}^{(2)},\lambda_{\mathrm{R},t}+\Delta\lambda_{\mathrm{R}}^{(2)},\dot{\lambda}_{\mathrm{R},t}^{(2)})=0
\end{aligned}
\tag{9}
$$

Therefore, only two steps are required for the exact solution which was to be proven.

4. Conclusions

The purpose of this paper was to prove that a two-step exact solution exist for the rate-dependent softening continua modelled with the Modified Mohr-Coulomb viscoplastic consistency model with linear hardening/softening rule. According to the proof with the Maple software, the two-step nature of the consistent viscoplasticity model is located in the invention of the viscosity parameter to the hardening/softening rules. On setting the viscosity modulus to zero, the classical rate-independent MMC plasticity with a single-step exact solution is recovered.

Even though the proof was carried out with the MMC model, it can be performed for all other models with linear yield surfaces, such as the Tresca surface, and linear hardening/softening rules.

Finally, it has been observed in numerical simulations that also the Drucker-Prager (DP) viscoplastic consistency model with linear hardening/softening rule has the same two-step nature with respect to convergence. This clearly stems from the fact that a closed form solution exists for the rate-independent DP plasticity problem. However, the proof is somewhat more complicated due to the nonlinearity of the DP criterion.

ACKNOWLEDGEMENTS

This research has been funded by Academy of Finland (Grant no. 251626).

Appendix A

The relation for solving the increment of viscoplastic multiplier vector in bi-surface consistent viscoplasticity with linear isotropic hardening/softening laws is derived here. Following notations are used.

$$
\begin{aligned}
&\mathbf{F}=\begin{pmatrix} f_{\mathrm{vp1}}(\boldsymbol{\sigma},\kappa_1,\dot{\kappa}_1)\\ f_{\mathrm{vp2}}(\boldsymbol{\sigma},\kappa_2,\dot{\kappa}_2)\end{pmatrix},\quad \dot{\boldsymbol{\sigma}}=\mathbf{E}(\dot{\boldsymbol{\varepsilon}}-\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}),\\
&\dot{\boldsymbol{\kappa}}=\begin{pmatrix}\dot{\lambda}_1k_1(\boldsymbol{\sigma})\\ \dot{\lambda}_2k_2(\boldsymbol{\sigma})\end{pmatrix},\quad \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}=\dot{\lambda}_1\partial_{\boldsymbol{\sigma}}g_{\mathrm{vp1}}+\dot{\lambda}_2\partial_{\boldsymbol{\sigma}}g_{\mathrm{vp2}}
\end{aligned}
\tag{A1}
$$

The yield functions are assumed to be of form

$$
f_{\mathrm{vp},i}(\boldsymbol{\sigma},\kappa_i,\dot{\kappa}_i)=\tilde{\sigma}_i-\sigma_i^Y(\kappa_i,\dot{\kappa}_i) \tag{A2}
$$

where $\tilde{\sigma}_i=\tilde{\sigma}_i(\boldsymbol{\sigma})$ is some expression of stress. Yield stresses are assumed to depend linearly on the hardening/softening variables and their rates, i.e.

$$
\sigma_i^Y=\sigma_{0i}^Y+h_i\kappa_i+s_i\dot{\kappa}_i \tag{A3}
$$

where h_i and s_i are the constant hardening/softening and viscosity moduli, respectively. At the end of the time step, condition $f_{\mathrm{vp},i}(\boldsymbol{\sigma}^{t+\Delta t},\kappa_i^{t+\Delta t},\dot{\kappa}_i^{t+\Delta t})=0$ must be satisfied for *i*= 1,2. Following Winnicki et al.[5], it assumed that $\dot{\lambda}_i^{t+\Delta t} = \Delta\lambda_i^{t+\Delta t} / \Delta t$ by which

$$\begin{aligned} \kappa_i^{t+\Delta t} &= \kappa_i^t + \Delta\lambda_i^{t+\Delta t} k_i \\ \dot{\kappa}_i^{t+\Delta t} &= \tfrac{\Delta\lambda_i^{t+\Delta t}}{\Delta t} k_i \end{aligned} \tag{A4}$$

By (A4) the yield functions $f_{\mathrm{vp}i}$ are formulated to depend on $\Delta\lambda_i$: $f_{\mathrm{vp},i} = f_{\mathrm{vp},i}(\boldsymbol{\sigma}^{t+\Delta t}, \Delta\lambda^{t+\Delta t})$. Hence, F can be expanded with the first term of the vector valued Taylor series:

$$\mathbf{F}(\Delta\boldsymbol{\lambda} + \delta\Delta\boldsymbol{\lambda}) = \mathbf{F}(\Delta\boldsymbol{\lambda}) + \nabla_{\Delta\boldsymbol{\lambda}}\mathbf{F}(\Delta\boldsymbol{\lambda})\delta\Delta\boldsymbol{\lambda} \tag{A5}$$

Computing the gradient of f_{vp1} with respect to $\Delta\boldsymbol{\lambda}$ yields (the derivation for f_{vp2} is identical)

$$\begin{aligned} &\nabla_{\Delta\boldsymbol{\lambda}} f_{\mathrm{vp1}} = \\ &\left[\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \frac{\partial\boldsymbol{\sigma}}{\partial\Delta\lambda_1} + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_1}, \quad \left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \frac{\partial\boldsymbol{\sigma}}{\partial\Delta\lambda_2} + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_2}\right] \\ &= \left[-\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \mathbf{E}\frac{\partial\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}}{\partial\Delta\lambda_1} + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_1}, \quad -\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \mathbf{E}\frac{\partial\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}}{\partial\Delta\lambda_2} + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_2}\right] \\ &= \left[-\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \left(\mathbf{E}\frac{\partial}{\partial\Delta\lambda_1}\left(\Delta\lambda_1\frac{\partial g_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}} + \Delta\lambda_2\frac{\partial g_{\mathrm{vp2}}}{\partial\boldsymbol{\sigma}}\right)\right) + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_1},\right. \\ &\quad \left. -\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \left(\mathbf{E}\frac{\partial}{\partial\Delta\lambda_2}\left(\Delta\lambda_1\frac{\partial g_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}} + \Delta\lambda_2\frac{\partial g_{\mathrm{vp2}}}{\partial\boldsymbol{\sigma}}\right)\right) + \frac{\partial f_{\mathrm{vp1}}}{\partial\Delta\lambda_2}\right] \end{aligned} \tag{A6}$$

It is assumed that $g_{\mathrm{vp}i}$ is independent of $\Delta\lambda_i$ and f_1 is independent of $\Delta\lambda_2$. By this, and making use of (A2)-(A4), the first row the gradient (A6) can be written as

$$\nabla_{\Delta\boldsymbol{\lambda}} f_{\mathrm{vp1}} = \left[-\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \mathbf{E}\frac{\partial g_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}} - \left(h_1 + \frac{s_1}{\Delta t}\right)k_1, \quad -\left(\frac{\partial f_{\mathrm{vp1}}}{\partial\boldsymbol{\sigma}}\right)^{\mathrm{T}} \mathbf{E}\frac{\partial g_{\mathrm{vp2}}}{\partial\boldsymbol{\sigma}}\right] \tag{A7}$$

Now, on solving for $\delta\Delta\boldsymbol{\lambda}$ and denoting the gradient with G the desired relation is obtained as

$$\begin{aligned} &\mathbf{F}(\Delta\boldsymbol{\lambda}) + \nabla_{\Delta\boldsymbol{\lambda}}\mathbf{F}(\Delta\boldsymbol{\lambda})\delta\Delta\boldsymbol{\lambda} = \mathbf{0} \\ &\Leftrightarrow \delta\Delta\boldsymbol{\lambda} = -\nabla_{\delta\boldsymbol{\lambda}}\mathbf{F}(\Delta\boldsymbol{\lambda})^{-1}\mathbf{F}(\Delta\boldsymbol{\lambda}) = \mathbf{G}^{-1}\mathbf{F}(\Delta\boldsymbol{\lambda}) \end{aligned} \tag{A8}$$

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., Viscoplasticity 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., Bićanić N., Viscoplastic Hoffman consistency model for concrete. Computers & Structures 2001; 79: 7-19.