Computational Inelasticity - Final Project

Philip Moseley

October 25, 2010

1 Introduction

Pressure-dependent materials such as soil or concrete require advanced material models to predict yielding. A commonly used yield criterion is the two-invariant Mohr-Coulomb model, as described in (1.1), where the σ_i's correspond to principal stresses, c is the cohesion, and ϕ is the friction angle. On the π-plane, the Mohr-Coulomb yield surface is similar to a distorted Tresca yield surface with lower tensile strengths and higher compressive strengths. It is well suited to any porous or granular material in compression loading.

$$\begin{aligned} F(\sigma_1,\sigma_2,\sigma_3,\kappa) &= |\sigma_A-\sigma_B| - \kappa \leq 0 \\ \kappa &= 2c\cos\phi - (\sigma_A+\sigma_B)\sin\phi \end{aligned} \tag{1.1}$$

The corners of the Mohr-Coulomb yield surface visible on the π-plane, as well as the pressure vertex in the meridian plane, can cause problems for numerical integration techniques. Several different smoothed approximations have been developed in order to simplify calculations, and it has been shown that these methods may actually be more accurate than the unmodified Mohr-Coulomb model.

This paper will derive the Drucker-Prager smoothed approximation to the Mohr-Coulomb plasticity model, including a detailed account of the integration algorithm and numerical implementation. The Drucker-Prager yield surface and plastic potential are defined by

$$F = q + \alpha p \tag{1.2}$$

$$G = q + \beta p \tag{1.3}$$

with deviatoric stress $q = \sqrt{\frac{3}{2}}\|\boldsymbol{s}\|$ and mean normal stress (pressure) $p = \frac{1}{3}\boldsymbol{\sigma} : \boldsymbol{\delta}$, with $\boldsymbol{s} = \boldsymbol{\sigma} - p\boldsymbol{\delta}$ and $\boldsymbol{\delta}$ as the Kronecher delta. The hardening laws are given by

$$\begin{cases} \hat{\alpha}(\lambda) &= \alpha_0 + \frac{2a\sqrt{k\lambda}}{k+\lambda} \\ \hat{\beta}(\lambda) &= \hat{\alpha}(\lambda) - \beta_0 \end{cases} \tag{1.4}$$

Here λ is the cumulative plastic multiplier and α_0,β_0,a, and k are material parameters.

2 Integration Algorithm

The Drucker-Prager material model is non-linear, so it is integrated with an iterative numerical scheme. In this paper a simple Newton-Raphson scheme is used in the material subroutine. Although the Drucker-Prager method only depends on two invariants, in this paper the more general three-invariant return mapping scheme is used. This is more difficult to implement, but can be applied to a broader range of material models. The method is a spectral method and therefore frame-invariant, and so can be easily applied to a broader range of material loading conditions. It requires the repeated calculation of eigenvalues and eigenvectors which can become computationally expensive in large models, but for the small numerical models presented here this is irrelevant.

The stress and strain tensors can be decomposed into their spectral forms using

$$\boldsymbol{\sigma}^{\mathrm{tr}}_{n+1} = \sum_{A=1}^{3} \sigma^{\mathrm{tr}}_{A} \boldsymbol{m}^{\mathrm{tr}}_{A} = \boldsymbol{\sigma}_n + \mathbb{C}^{e}_{n+1} : \Delta\boldsymbol{\varepsilon} \tag{2.1a}$$

$$\boldsymbol{\varepsilon}^{\mathrm{e\ tr}}_{n+1} = \sum_{A=1}^{3} \varepsilon^{\mathrm{e}}_{A}{}^{\mathrm{tr}} \boldsymbol{m}^{\mathrm{tr}}_{A} = \boldsymbol{\varepsilon}^{\mathrm{e}}_{n} + \Delta\boldsymbol{\varepsilon} \tag{2.1b}$$

The subscripts n and n+1 refer to the timestep, and unless specified are assumed to be n+1. σ^{tr}_{A} and $\varepsilon^{\mathrm{e}}_{A}{}^{\mathrm{tr}}$ are the trial spectral magnitudes (eigenvalues) and $\boldsymbol{m}^{\mathrm{tr}}_{A}$ is the trial spectral direction, defined as

$$\boldsymbol{m}^{\mathrm{tr}}_{A} = \boldsymbol{n}^{\mathrm{tr}}_{A} \otimes \boldsymbol{n}^{\mathrm{tr}}_{A} \tag{2.2}$$

The trial eigenvectors are $\boldsymbol{n}^{\mathrm{tr}}_{A}$. The eigenvectors for $\boldsymbol{\sigma}^{\mathrm{tr}}$, $\boldsymbol{\varepsilon}^{\mathrm{tr}}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are coaxial and therefore equivalent, and hereafter will be written without the trial indicator $^{\mathrm{tr}}$.

Using the principal stress axes, the algorithm calculates an elastic predictor followed by a plastic corrector to evaluate the stresses. This can be written as

$$\sigma_A = \sigma^{\mathrm{tr}}_{A} - \Delta\lambda \sum_{B=1}^{3} a^{e}_{AB} \frac{\partial G}{\partial \sigma_B} \tag{2.3a}$$

$$\varepsilon^{e}_{A} = \varepsilon^{\mathrm{e}}_{A}{}^{\mathrm{tr}} - \Delta\lambda \frac{\partial G}{\partial \sigma_A} \tag{2.3b}$$

$\Delta\lambda$ is the change in the cumulative plastic multiplier, and $\boldsymbol{a}^{e}_{AB}$ is the elasticity matrix in the principal axes,

$$\boldsymbol{a}^{e}_{AB} = \frac{\partial \sigma_A}{\partial \varepsilon^{e}_{B}} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda \\ \lambda & \lambda + 2\mu & \lambda \\ \lambda & \lambda & \lambda + 2\mu \end{bmatrix} \tag{2.4}$$

Here λ refers to the Lamé material constant, not the plastic multiplier.

Additional constraints are imposed by the hardening equations in (1.4). Combining the plastic internal variables α and β into a single matrix simplifies our notation,

$$\boldsymbol{\kappa}(\lambda) = \begin{bmatrix} \alpha(\lambda) \\ \\ \beta(\lambda) \end{bmatrix}, \quad \boldsymbol{h} = \frac{\partial \boldsymbol{\kappa}}{\partial \lambda} = \begin{bmatrix} \dfrac{\partial \alpha}{\partial \lambda} \\ \dfrac{\partial \beta}{\partial \lambda} \end{bmatrix} \tag{2.5}$$

Similarly, the following vectors are created for notational convenience

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix}, \quad \boldsymbol{\varepsilon}^{\text{e tr}} = \begin{bmatrix} \varepsilon_1^{\text{e tr}} \\ \varepsilon_2^{\text{e tr}} \\ \varepsilon_3^{\text{e tr}} \end{bmatrix}, \quad \boldsymbol{g} = \begin{bmatrix} \dfrac{\partial G}{\partial \sigma_1} \\ \dfrac{\partial G}{\partial \sigma_2} \\ \dfrac{\partial G}{\partial \sigma_3} \end{bmatrix} \tag{2.6}$$

The Newton-Raphson iteration scheme solves all these equations simultaneously, along with the consistency condition $F = 0$. We can write this as a residual vector $\boldsymbol{r}$ and a vector of unknowns $\boldsymbol{x}$,

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{a}_{AB}^{\text{e}-1} \cdot \boldsymbol{\sigma} - \boldsymbol{\varepsilon}^{\text{e tr}} + \Delta\lambda \boldsymbol{g} \\ \boldsymbol{\kappa} - \boldsymbol{\kappa}_n - \Delta\lambda \boldsymbol{h} \\ F \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{\sigma} \\ \boldsymbol{\kappa} \\ \Delta\lambda \end{bmatrix} \tag{2.7}$$

where the iterations are defined by

$$\boldsymbol{\Delta x} = -\boldsymbol{J}_k^{-1} \cdot \boldsymbol{r}_k, \quad \boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \Delta \boldsymbol{x}, \quad k \leftarrow k+1 \tag{2.8}$$

and continue to loop until the norm $||\boldsymbol{r}||$ is sufficiently small. $\boldsymbol{J}$ is the Jacobian, defined by

$$\boldsymbol{J} = \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{\varepsilon}^{\text{e tr}}} = \begin{bmatrix} \dfrac{\partial r_1}{\partial \boldsymbol{\sigma}}_{\text{3x3}} & \dfrac{\partial r_1}{\partial \boldsymbol{\kappa}}_{\text{3x}\,\text{n}_\text{piv}} & \dfrac{\partial r_1}{\partial \boldsymbol{\lambda}}_{\text{3x1}} \\ \dfrac{\partial r_2}{\partial \boldsymbol{\sigma}}_{\text{n}_\text{piv}\,\text{x1}} & \dfrac{\partial r_2}{\partial \boldsymbol{\kappa}}_{\text{n}_\text{piv}\,\text{x}\,\text{n}_\text{piv}} & \dfrac{\partial r_2}{\partial \boldsymbol{\lambda}}_{\text{n}_\text{piv}\,\text{x1}} \\ \dfrac{\partial r_3}{\partial \boldsymbol{\sigma}}_{\text{1x3}} & \dfrac{\partial r_3}{\partial \boldsymbol{\kappa}}_{\text{1x}\,\text{n}_\text{piv}} & \dfrac{\partial r_3}{\partial \boldsymbol{\lambda}}_{\text{1x1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}^{\text{e}-1} + \Delta\lambda \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} & \Delta\lambda \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{\kappa}} & \boldsymbol{q} \\ -\Delta\lambda \dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{\sigma}} & \boldsymbol{I} & -\boldsymbol{h} - \Delta\lambda \dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{\lambda}} \\ \dfrac{\partial F}{\partial \boldsymbol{\sigma}} & \dfrac{\partial F}{\partial \boldsymbol{\kappa}} & 0 \end{bmatrix} \tag{2.9}$$

The first matrix in (2.9) is provided as a reference for the sizes of each submatrix, where n_piv is the number of plastic internal variables (n_piv equals two in this model, α and β). The derivatives in the Jacobian are included in Appendix A.

Once the material subroutine integration algorithm described here has converged, the consistent tangent operator $\mathbb{C}$ can be calculated to achieve an optimal convergence rate in the loading iterations. In spectral form,

$$\mathbb{C} = \sum_{A=1}^{3} \sum_{B=1}^{3} a_{AB} \boldsymbol{m}_\text{A} \otimes \boldsymbol{m}_\text{B} + \frac{1}{2} \sum_{A=1}^{3} \sum_{B \neq A} \left(\frac{\sigma_B - \sigma_A}{\varepsilon_B^{\text{e tr}} - \varepsilon_A^{\text{e tr}}} \right) [\boldsymbol{m}_\text{AB} \otimes (\boldsymbol{m}_\text{AB} + \boldsymbol{m}_\text{BA})] \tag{2.10}$$

with $a_{AB} = \partial\sigma_A / \partial\varepsilon_B^{\text{e tr}}$. It can be shown that the coefficients of a_{AB} come from the inverse of the local Jacobian, $a_{AB} = J_{AB}^{-1}$.

The steps of the algorithm are described in detail in Box 1.

1. Compute $\boldsymbol{\sigma}^{\mathrm{tr}}$ and $\boldsymbol{\varepsilon}^{\mathrm{e\ tr}}$.
2. Spectrally decompose $\boldsymbol{\sigma}^{\mathrm{tr}}$ and $\boldsymbol{\varepsilon}^{\mathrm{e\ tr}}$ using (2.1).
3. Check if $F \leq 0$
    - (a) If yes, the material is still in the elastic region. Set the following variables, then exit the loop.
      $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{tr}}$, $\boldsymbol{\kappa}_{n+1} = \boldsymbol{\kappa}_n$, $\lambda_{n+1} = \lambda_n$
    - (b) If no, the material is in yielding. Continue to the next step.
4. Newton-Raphson iterations on (2.8) to converge on a solution of $\boldsymbol{x}$.
5. Set $\lambda_{n+1} = \lambda_n + \Delta\lambda$
6. Calculate the consistent tangent operator (2.10) for the global iteration loop.
7. Compute the final stresses in original coordinate system using (2.1a), where the values σ_A^{tr} are the values of $\boldsymbol{\sigma}$ from $\boldsymbol{x}$, and the directions $\boldsymbol{m}_A^{\mathrm{tr}}$ are the same spectral directions found in Step 2.

Box 1: Return mapping algorithm in principal stress directions.

3 Numerical Experiments

Two numerical experiments are discussed. These are designed to test both the global loading iteration loop and the local material subroutine iteration loop. The common initial stress state is $p_0 = -50$ kPa and $q = 0$ kPa, ie a hydrostatic state. Both experiments use vertical strain control and horizontal stress control. During loading, the vertical strain increment is $\Delta\varepsilon_1 = -0.2\%$ and the horizontal stress is kept constant. Both experiments stop at $\varepsilon_1 = -3\%$.

Material constants are given in Table 1.

Constant	Value
E	25000 kPa
ν	0.3
α_0	0.7
β_0	0.7
a	0.25
k	0.1

Table 1: Material constants for numerical experiments.

3.1 Axisymmetric Loading

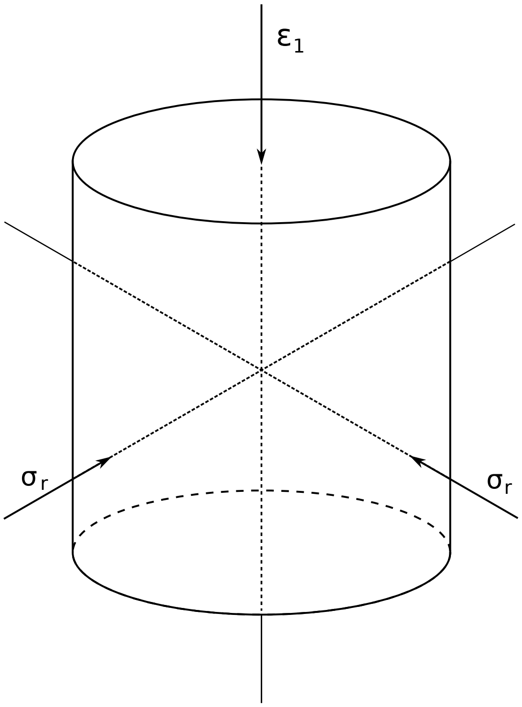


Figure 1: Loading conditions for axisymmetric loading.

For the case of axisymmetric loading, a strain $\Delta\varepsilon$ is applied along the axis while the radial stresses σ_r are kept constant (Figure 1). In order to apply these complicated loading conditions, a global Newton-Raphson loop is implemented.

The Newton-Raphson iteration scheme solves for the appropriate stresses and strains to meet the boundary conditions. We can write this as a residual vector $\boldsymbol{R}$ and a vector of unknowns $\boldsymbol{X}$,

$$\boldsymbol{R} = \begin{bmatrix} \hat{\varepsilon}_1(t) - \varepsilon_1 \\ \hat{\sigma}_2(t) - \sigma_2 \\ \hat{\sigma}_3(t) - \sigma_3 \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \Delta\varepsilon_1 \\ \Delta\varepsilon_2 \\ \Delta\varepsilon_3 \end{bmatrix} \tag{3.1}$$

This problem is difficult to write a residual for, so we rewrite it

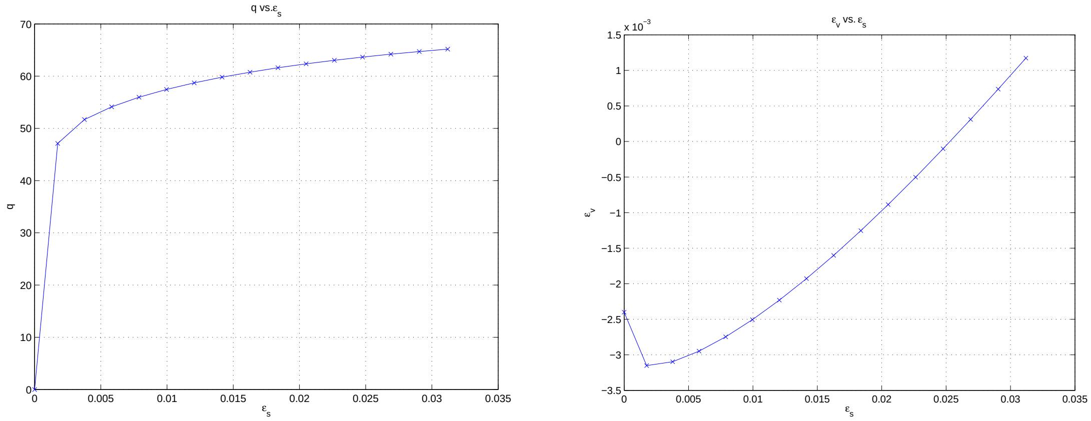
$$\boldsymbol{R} = \begin{bmatrix} -\sigma_1 \\ \hat{\sigma}_2(t) - \sigma_2 \\ \hat{\sigma}_3(t) - \sigma_3 \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \Delta\varepsilon_1 - \hat{\varepsilon}_1(t) \\ \Delta\varepsilon_2 \\ \Delta\varepsilon_3 \end{bmatrix} \tag{3.2}$$

Now the Newton iterations can be defined by

$$\boldsymbol{\Delta X} = \mathbb{C}_m^{-1} \cdot \boldsymbol{R}_m, \quad \boldsymbol{X}_{m+1} \leftarrow \boldsymbol{X}_m + \Delta\boldsymbol{X}, \quad m \leftarrow m + 1 \tag{3.3}$$

where $\mathbb{C}$ is the consistent tangent operator from the material subroutine. Because the problem is strain-controlled, $\Delta\varepsilon_1$ must equal the -0.2% loading. In order to ensure this, before solving an iteration step we back-substitute and solve for a σ_1 that satisfies the $\Delta\varepsilon_1$ boundary condition. Now we can solve the iteration step for $\Delta\varepsilon_2$ and $\Delta\varepsilon_3$.

Under this loading, the Drucker-Prager is elastic for one loading step before going plastic. Once the material response is plastic, the material begins to soften.



(a) q vs. ε_s

(b) ε_v vs. ε_s

Figure 2: Evolution of stresses and strains.

The convergence for this experiment is quadratic, as expected from the use of the consistent tangent operator. In the final step the iterations converge to 1e-16, which is machine precision, not a reflection of the method or the implementation.

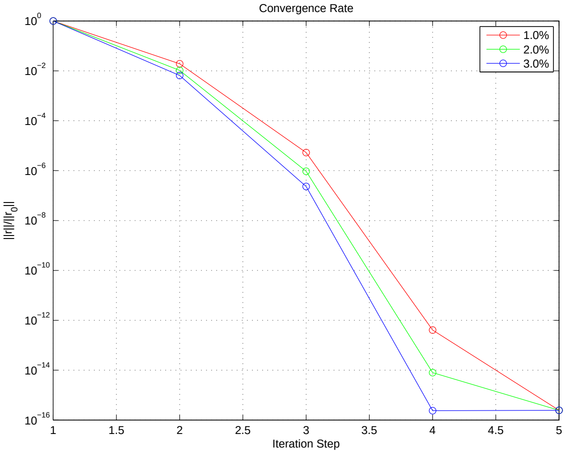


Figure 3: Global convergence rates at three different time steps.

3.2 Plane Strain Loading

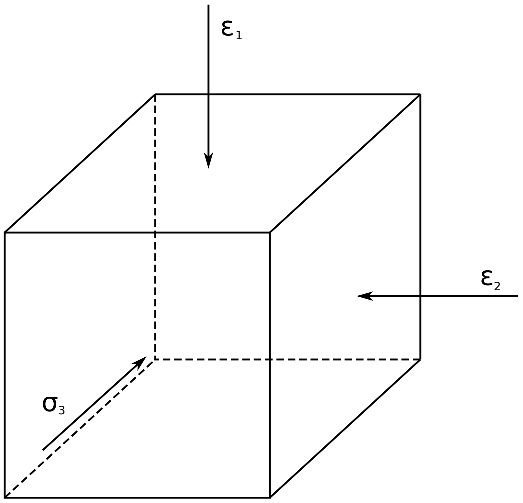
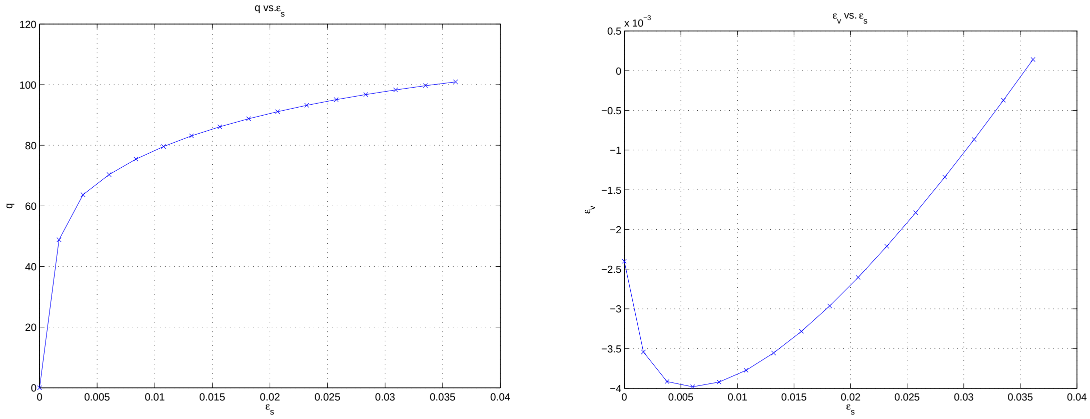


Figure 4: Loading conditions for axisymmetric loading.

For the plane strain loading, a strain $\Delta\varepsilon$ is applied in the vertical (ε_1) direction. The strain ε_2 is maintained at zero, and only the stress is controlled in the third direction. Again, a global Newton-Raphson iteration scheme is used to apply these conditions. The same Newton-Raphson loop as in Section 3.1 is used here, with a minor modification to control two strains instead of just one. Here we solve for the σ_1 and σ_2 values that correspond to the desired strain state before inverting the matrix.

The results are similar to the case of axisymmetric loading in Section 3.1. The additional strain constraint in the horizontal direction from plane strain causes higher stresses under loading.



(a) q vs. ε_s

(b) ε_v vs. ε_s

Figure 5: Evolution of stresses and strains.

Convergence for the plane strain loading is still quadratic, due to the consistent tangent operator.

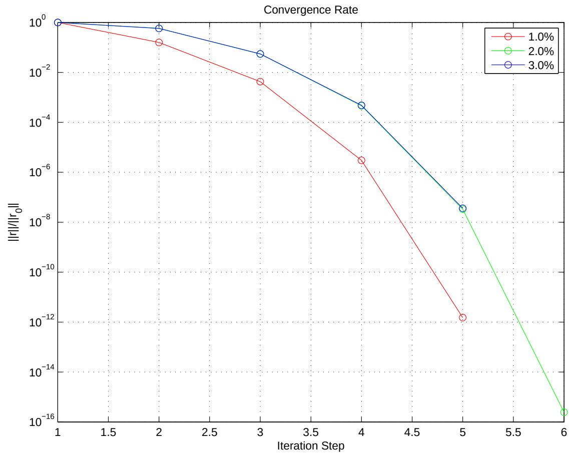


Figure 6: Global convergence rates at three different time steps.

References

[1] R.I. Borja and J.E. Andrade. Critical state plasticity. part vi: Meso-scale finite element simulation of strain localization in discrete granular materials. *Computer Methods in Applied Mechanics and Engineering*, 195(37-40):5115–5140, 2006.

[2] R.I. Borja, K.M. Sama, and P.F. Sanz. On the numerical integration of three-invariant elastoplastic constitutive models. *Computer Methods in Applied Mechanics and Engineering*, 192(910):1227–1258, 2003.

[1, 2]

A Derivatives

All the necessary derivatives for the integration procedure described in Section 2 are included here. Material constants are a, k, α_0, β_0. Additionally, K is the bulk modulus, ν is the Poisson's ratio, E is the Young's modulus, and μ is one of the Lamé constants. Unless specified otherwise, λ is the cumulative plastic multiplier.

$$F = q + \alpha p = \sqrt{\frac{3}{2}} \|\boldsymbol{\sigma} - p\boldsymbol{\delta}\| + \alpha p \tag{A.1a}$$

$$\frac{\partial F}{\partial \boldsymbol{\sigma}} = \frac{\alpha}{3} + \frac{1}{2\sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_2\sigma_3 + \sigma_3^2 - \sigma_1\left(\sigma_2 + \sigma_3\right)}} \begin{bmatrix} 2\sigma_1 - \sigma_2 - \sigma_3 \\ -\sigma_1 + 2\sigma_3 - \sigma_3 \\ -\sigma_1 - \sigma_2 + 2\sigma_3 \end{bmatrix} \tag{A.1b}$$

$$\frac{\partial F}{\partial \boldsymbol{\kappa}} = \begin{bmatrix} \partial F/\partial \alpha \\ \partial F/\partial \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\left(\sigma_1 + \sigma_2 + \sigma_3\right) \\ 0 \end{bmatrix} \tag{A.1c}$$

Box 2: Yield surface derivatives.

$$G = q + \beta p = \sqrt{\frac{3}{2}} \|\boldsymbol{\sigma} - p\boldsymbol{\delta}\| + \beta p \tag{A.2a}$$

$$\boldsymbol{g} = \frac{\partial G}{\partial \boldsymbol{\sigma}} = \frac{\beta}{3} + \frac{1}{2\sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_2\sigma_3 + \sigma_3^2 - \sigma_1\left(\sigma_2 + \sigma_3\right)}} \begin{bmatrix} 2\sigma_1 - \sigma_2 - \sigma_3 \\ -\sigma_1 + 2\sigma_3 - \sigma_3 \\ -\sigma_1 - \sigma_2 + 2\sigma_3 \end{bmatrix} \tag{A.2b}$$

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} = A \begin{bmatrix} (\sigma_2 - \sigma_3)^2 & (\sigma_1 - \sigma_3)(\sigma_3 - \sigma_2) & (\sigma_1 - \sigma_2)(\sigma_2 - \sigma_3) \\ (\sigma_1 - \sigma_3)(\sigma_3 - \sigma_2) & (\sigma_1 - \sigma_3)^2 & (\sigma_1 - \sigma_2)(\sigma_3 - \sigma_1) \\ (\sigma_1 - \sigma_2)(\sigma_2 - \sigma_3) & (\sigma_1 - \sigma_2)(\sigma_3 - \sigma_1) & (\sigma_1 - \sigma_2)^2 \end{bmatrix} \tag{A.2c}$$

$$A = \frac{3}{4\left(\sigma_1 + \sigma_2 - \sigma_2\sigma_3 + \sigma_3^2 - \sigma_1(\sigma_2 + \sigma_3)\right)^{3/2}} \tag{A.2d}$$

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\kappa}} = \frac{1}{3} \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \tag{A.2e}$$

Box 3: Plastic potential derivatives.

$$\boldsymbol{\kappa}(\lambda) = \begin{bmatrix} \alpha(\lambda) \\ \beta(\lambda) \end{bmatrix} = \begin{bmatrix} \alpha_0 + \frac{2a\sqrt{k\lambda}}{k+\lambda} \\ \alpha - \beta_0 \end{bmatrix} \tag{A.3a}$$

$$\boldsymbol{h} = \frac{\partial \boldsymbol{\kappa}}{\partial \lambda} = \frac{ak\left(k-\lambda\right)}{\sqrt{k\lambda}\left(k+\lambda\right)^2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{A.3b}$$

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.3c}$$

$$\frac{\partial \boldsymbol{h}}{\partial \lambda} = -\frac{ak^2\left(k^2 + 6k\lambda - 3\lambda^2\right)}{2\left(k\lambda\right)^{3/2}\left(k+\lambda\right)^3} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{A.3d}$$

Box 4: Plastic internal variable derivatives.

Note that for the plastic internal variables, β is a direct function of α and does not need to be included in $\boldsymbol{\kappa}$. However, doing so keeps the algorithm and code more general, so it was implemented as seen in Box 4.