INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2017; **109**:218–234
Published online 16 June 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.5284

A robust return-map algorithm for general multisurface plasticity

D. P. Adhikary[1], C. T. Jayasundara[1], R. K. Podgorney[2] and A. H. Wilkins[1,*,†]

[1]*Coal Mining Programme, CSIRO Energy Flagship, Queensland Centre for Advanced Technologies, PO Box 883, Kenmore, Q4069, Australia*
[2]*Energy Resource Recovery and Sustainability, Idaho National Laboratory, Idaho Falls, ID, USA*

SUMMARY

Three new contributions to the field of multisurface plasticity are presented for general situations with an arbitrary number of nonlinear yield surfaces with hardening or softening. A method for handling linearly dependent flow directions is described. A residual that can be used in a line search is defined. An algorithm that has been implemented and comprehensively tested is discussed in detail. Examples are presented to illustrate the computational cost of various components of the algorithm. The overall result is that a single Newton-Raphson iteration of the algorithm costs between 1.5 and 2 times that of an elastic calculation. Examples also illustrate the successful convergence of the algorithm in complicated situations. For example, without using the new contributions presented here, the algorithm fails to converge for approximately 50% of the trial stresses for a common geomechanical model of sedementary rocks, while the current algorithm results in complete success. Because it involves no approximations, the algorithm is used to quantify the accuracy of an efficient, pragmatic, but approximate, algorithm used for sedimentary-rock plasticity in a commercial software package. The main weakness of the algorithm is identified as the difficulty of correctly choosing the set of initially active constraints in the general setting. Copyright © 2016 John Wiley & Sons, Ltd.

Received 19 March 2015; Revised 17 March 2016; Accepted 14 April 2016

KEY WORDS: multisurface plasticity; return-map algorithm; Kuhn–Tucker conditions; geomechanics

1. INTRODUCTION

Many mechanical models of real-life materials involve multisurface plasticity. For instance, there are six yield surfaces in Mohr–Coulomb plasticity. In rock mechanics, it is common to use Mohr–Coulomb in conjunction with a tensile (Rankine) cap (an additional 3 yield surfaces), and weak-plane shear and tensile failure (an additional 2 yield surfaces). This rock mechanics example is explored in Section 6. In crystal plasticity, there are typically many different slip planes (e.g. 12 for face-centred-cubic crystals), each contributing a yield surface.

A large computational burden in the numerical simulation of these materials is the return-map algorithm. Given a trial stress that is inadmissible (it lies outside the elastic domain), this algorithm provides a returned stress that lies on the boundary of the admissible domain. Multisurface plasticity considers situations where the admissible region is the intersection of a number of convex domains ('surfaces') in stress space. While each convex domain has a smooth boundary, the intersection does not. This makes the return-map algorithm more complicated than the smooth case, for a returned stress may lie on a 'corner' of the admissible region where the derivative is not defined. Computationally, however, non-smooth domains are not overly difficult: instead, most of the

*Correspondence to: Andy Wilkins, Coal Mining Programme, CSIRO Energy Flagship, Queensland Centre for Advanced Technologies, PO Box 883, Kenmore, Q4069, Australia.
†E-mail: andrew.wilkins@csiro.au

difficulties arise from inequalities in the Kuhn–Tucker constraints, which define physically acceptable solutions. This is described further later.

Because of the computational burden, efficient return-map algorithms have been devised for specific plastic models, or classes of model — see for instance [1–5]. There are a great many of these model-specific algorithms, but this paper considers general situations where no such algorithms exist, such as the aforementioned rock mechanics example with 11 nonlinear yield surfaces, which cannot all be formulated in principal stress space. It is not this paper's purpose to supplant the model-specific algorithms, because in cases for which they have been devised they will be more efficient than the general algorithm presented herein.

An alternative to using multisurface plasticity is to approximate the admissible region with a smooth domain (C1 or C2). This also eliminates problems associated with the Kuhn–Tucker constraints. For instance, the Mohr–Coulomb yield surface may be smoothed with little loss in accuracy [6]. However, smoothing techniques have only been devised for the common plastic models, and for complicated multisurface plasticity models with hardening or softening the smoothing formulation is difficult. When a smoothed formulation is available, it will typically be more efficient than the general algorithm presented herein.

Succinctly, this paper considers the general situation, applicable to cases where model-specific algorithms or smoothing techniques have not yet been devised. This general case was first addressed in detail in the celebrated article of Simo, Kennedy and Govindjee [7]. The book [8] provides a more detailed description and [9] offers other insights, but to our knowledge, no extension of the general formulation has been performed because, perhaps, efforts have been concentrated on specific plasticity models that are frequently used by the solid-mechanics community. The current article describes extensions of [7]. It is an implicit method.

After a successful application of the return-map algorithm, the second ingredient required for efficient numerical performance of computer simulations of plasticity is the consistent tangent operator (eg [10]). This article concentrates upon improving the return-map algorithm and does not derive the consistent tangent operator.

While implementing the ideas of [7] in a computer code, we found some cases where the algorithm failed to converge. In this article, we present our solutions to these following cases:

(1) The situation where the flow directions are linearly dependent was explicitly excluded from consideration in [7]. However, many cases of practical interest contain linearly dependent flow directions. Various solutions to this problem have been presented in the context of crystal plasticity [11–17], and our solution generalises these ideas to the case of nonlinear yield functions.
(2) Although [7] suggest that a line search is necessary to ensure convergence of the Newton-Raphson procedure, they give no details. We present a residual that may be used in a standard cubic line search. The residual is not immediately intuitive in the case of linearly dependent flow directions. In this paper, Newton-Raphson is abbreviated to NR.
(3) The algorithm of [7] involves successively eliminating constraints while converging to a solution. However, there exist examples where constraints must be added, rather than eliminated. This can occur without hardening, as illustrated later, and can also occur because of hardening [18]. We present our generalisation of the algorithm of [7] in detail, including consideration of linearly dependent flow directions.

The algorithm is shown to be efficient and successful in Section 6. We have implemented and thoroughly tested our open-source algorithm [19] in the Multiphysics Object Oriented Simulation Environment (MOOSE) finite-element framework [20].

In cases where there is complicated hardening and softening of the multiple yield surfaces, which results in a non-unique solution to the return-map process, the algorithm contained herein may not converge to the physically correct solution (without subdividing the applied strain increment into an excessive number of substeps). The algorithm may require modification in these cases.

DOI: 10.1002/nme

2. PROBLEM DEFINITION

2.1. Definition

Denote components of the stress tensor by σ_{ij} with $i, j = 1, \ldots, d = 3$. Denote the internal variables by q_a with $a = 1, \ldots, Q$. Denote the yield functions by $f_\alpha = f_\alpha(\sigma, q)$, with $\alpha = 1, \ldots, N$. The admissible region is

$$E_{\text{adm}} = \{(\sigma, q) \mid f_\alpha(\sigma, q) \leqslant 0 \ \forall \alpha\}. \tag{1}$$

It is assumed that each yield function f_α is smooth, and $f_\alpha = 0$ describes a convex domain, but the different f_α may intersect in a non-smooth way. Denote the flow potentials by $r^\alpha_{ij}(\sigma, q)$ and the hardening potentials by $h^\alpha_a(\sigma, q)$.

The return-map algorithm solves the following problem. Given $(\sigma^{\text{trial}}, q^{\text{old}}) \notin E_{\text{adm}}$, find $(\sigma, q, \dot\gamma)$ such that

- Kuhn–Tucker conditions: $\dot\gamma^\alpha \geqslant 0$ and $f_\alpha \leqslant 0$ and $\dot\gamma^\alpha f_\alpha = 0$ (no sum on α).
- Plastic flow fule: $\sigma^{\text{trial}}_{ij} - \sigma_{ij} = E_{ijkl} \sum_{\alpha=1}^{N} \dot\gamma^\alpha r^\alpha_{kl}(\sigma, q)$, where E_{ijkl} are the components of the rank-four elasticity tensor, and the sum on k and l is implied.
- Internal flow rule: $q^{\text{old}}_a - q_a = \sum_{\alpha=1}^{N} \dot\gamma^\alpha h^\alpha_a(\sigma, q)$.

The quantities $\dot\gamma^\alpha$ are the plastic multipliers and are associated with each yield function f_α. The condition $\dot\gamma^\alpha \dot f_\alpha = 0$ ([7] equation 6) is automatically satisfied by the algorithm later. Only linear elasticity is considered in this article.

2.2. A pictorial explanation of the problem

The problem is depicted in Figure 1. This figure shows two yield surfaces, so $\alpha = 1, 2$, $(N = 2)$. The admissible region E_{adm}, where both yield functions are non-positive is heavily shaded. At point

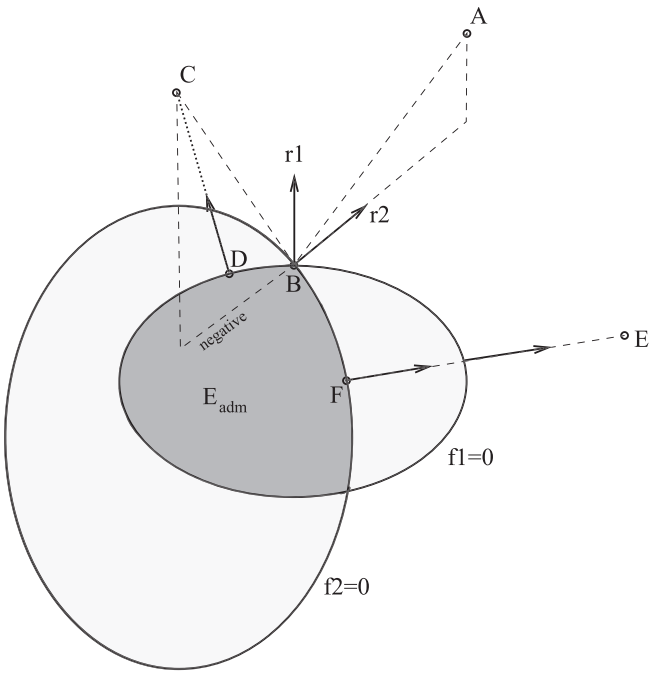

Figure 1. Two yield surfaces, f_1 and f_2 in a two-dimensional stress space. The individual admissible regions where $f < 0$ are lightly shaded, while the dark shading indicates the admissible region E_{adm}. Six stresses are labelled: A, C and E are indamissible, while B, D and F are admissible. Flow potentials r^1 and r^2 are depicted at point B, and flow potentials are also shown at points D and along the vector FE.

DOI: 10.1002/nme

B, the flow-potential directions are shown; in this figure, it is assumed that $r = \partial f/\partial\sigma$ (associative plasticity). Flow-potential directions are also shown at other interesting points. The return-map algorithm solves the following: given an inadmissible stress (A, C or E), find an admissible stress subjected to the three previously mentioned conditions.

Only at point B and the other intersection point are both yield surfaces zero, $f_1 = 0 = f_2$. If the algorithm returns to one of these points, both plastic multipliers must be non-negative, $\dot\gamma^1 \geqslant 0$ and $\dot\gamma^2 \geqslant 0$. Alternatively, if the algorithm returns to the first yield surface $f_1 = 0$ with $f_2 < 0$, then the plastic multipliers must satisfy $\dot\gamma^1 \geqslant 0$ and $\dot\gamma^2 = 0$. The difficulty of achieving this is illustrated by considering each point in careful detail.

- Point A. This returns to point B. Notice that at point B, the yield functions are zero, $f_1 = 0 = f_2$. The plastic flow rule states that $\sigma_{ij}^{\text{trial}} - \sigma_{ij} = E_{ijkl}\sum_{\alpha=1}^{2}\dot\gamma^\alpha r_{kl}^\alpha$, which means geometrically that the vector pointing from B to A is a linear combination of the r^1 and r^2 vectors shown at point B (assume here that E_{ijkl} is isotropic with zero Poisson's ratio). This is illustrated by the dashed triangle lying between B and A. Finally, the Kuhn–Tucker conditions are that $\dot\gamma^1 \geqslant 0$ and $\dot\gamma^2 \geqslant 0$. That is the linear combination must have *non-negative* coefficients, which is clearly true from the dashed triangle.
- Point C. This does not return to point B, and this illustrates the difficulty of satisfying the Kuhn–Tucker conditions. Suppose point C returns to point B. Both the yield functions are zero there, and one can find $\dot\gamma^1$ and $\dot\gamma^2$ that are the coefficients of the linear combination in the plastic flow rule, as illustrated by the triangle lying between B and C. However, $\dot\gamma^2 < 0$: a *negative* amount of vector r^2 is needed to form the triangle! Thus, the Kuhn–Tucker condition is violated. In fact, point C returns to point D. In this case, the second yield function f_2 becomes irrelevant. This is called 'deactivating' a yield function constraint. At point D, $f_1 = 0$, and $f_2 < 0$. The plastic flow rule needs only $\dot\gamma^1 > 0$ and has $\dot\gamma^2 = 0$. Hence all conditions are satisfied. The big question is: given an arbitrary inadmissible point (such as point C), *a priori* which yield functions should be chosen 'active' and which should be "deactivated"?
- Point E. This returns to point F, where $f_1 < 0$ and $f_2 = 0$. To satisfy the Kuhn–Tucker conditions $\dot\gamma^1 = 0$ and $\dot\gamma^2 > 0$. However, the return-map process is iterative, and suppose at the initial iteration, the algorithm activates both yield surfaces. As the algorithm moves iteratively from E to F, the plastic multipliers $\dot\gamma^1$ and $\dot\gamma^2$ will not be uniquely defined, because the flow-potential directions r^1 and r^2 are parallel (in general, the flow potentials need only be linearly dependent). This non-uniqueness causes rather subtle complications in the algorithm as discussed further in Section 3.

2.3. Further discussion

The admissible region, E_{adm}, is an intersection of convex sets, and is assumed to be convex itself [21]. Usually, this is sufficient to ensure uniqueness of the solution; however, there are certain cases with hardening/softening where the solution is not unique. As in [7], this non-uniqueness, and whether the solution obtained is the 'correct' one, is not considered in detail here; our algorithm simply provides a single solution that satisfies the three conditions previously. There are also pathological cases where the flow potentials are linearly dependent in which this uniqueness is spoiled, and this is discussed in further detail later.

The inequalities in the Kuhn–Tucker conditions are what makes the return-map algorithm difficult. The problem is often solved using an NR approach (fully implicit, backward-Euler) with initial values $\sigma = \sigma^{\text{trial}}$, $q = q^{\text{old}}$ and $\dot\gamma = 0$, and this approach is explored here.

In this approach, the main difficulty lies in determining the set of 'active constraints', as explained in [7]. Denote this set by J. Specifically, at the solution of the return-map algorithm

- Iff $\alpha \in J$, then $\dot\gamma^\alpha > 0$ and $f_\alpha = 0$.
- Iff $\alpha \notin J$, then $\dot\gamma^\alpha = 0$ and $f_\alpha \leqslant 0$.

Consider initialising the return-map algorithm at point A of Figure 2. Clearly the solution is point B. The active set consists of just $\alpha = 1$, with $\dot\gamma^1 > 0$, while $\dot\gamma^2 = 0 = \dot\gamma^3$ because those constraints are

DOI: 10.1002/nme

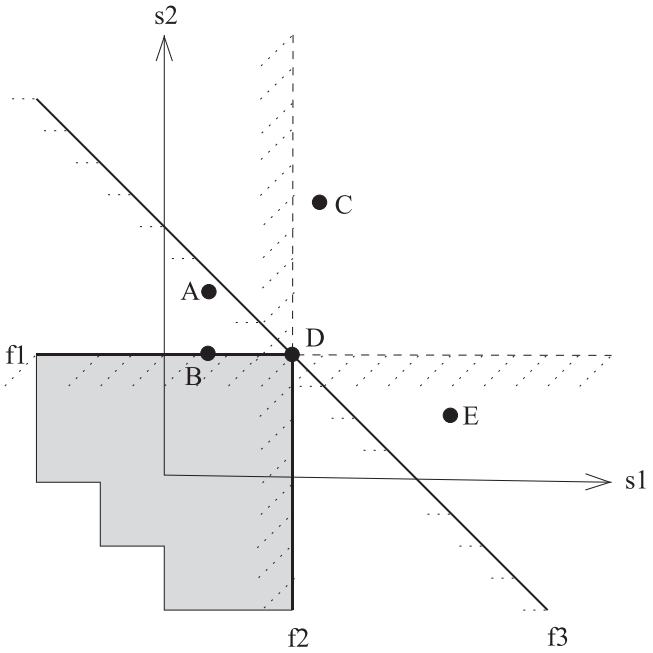

Figure 2. Two-dimensional stress space parameterised by s_1 and s_2, which could be principal stress components, for example. Three yield surfaces are shown: $f_1 = s_2 - 1$, $f_2 = s_1 - 1$ and $f_3 = s_1 + s_2 - 2$. The dashed lines indicate the admissible regions for each yield function, while the shading shows the admissible regions E_{adm}. Points A, B etc. are described in the text. Points A, C and E are inadmissible, while points B and D are admissible.

inactive. Although trivial in this case, finding the set of active constraints is complicated in a fully nonlinear situation.

In all figures we assume the following: (1) the elasticity tensor is isotropic and has zero Poisson's ratio, $E_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$; (2) the flow is associative, meaning the flow potentials are $r^\alpha_{ij} = \partial f_\alpha / \partial \sigma_{ij}$; (3) there is no hardening, $h^\alpha_a = 0$. This is simply for ease of presentation, but the results hold for general elasticity tensors E_{ijkl}, which obey standard conditions [8], general flow potentials r^α and general hardening potentials h^α_a.

3. LINEAR DEPENDENCE

The NR procedure is attempting to solve

$$0 = \begin{pmatrix} f_\alpha \quad \text{with } \alpha \in J \\ \sigma^{\text{trial}}_{ij} - \sigma_{ij} - E_{ijkl} \sum_{\alpha \in J} \dot{\gamma}^\alpha r^\alpha_{kl} \\ q^{\text{old}}_a - q^a - \sum_{\alpha \in J} \dot{\gamma}^\alpha h^\alpha_a \end{pmatrix}, \tag{2}$$

for these degrees of freedom: the plastic multipliers for the active constraints, $\dot{\gamma}^\alpha$ for $\alpha \in J$; the stress σ_{ij}; and all the internal variables q^a, $a = 1, \ldots, Q$. If the flow potentials r^α appearing here are linearly dependent, then the $\dot{\gamma}^\alpha$ cannot be determined uniquely. Blindly including the entire active set (all $\alpha \in J$) results in a singular Jacobian in the NR procedure. This situation was explicitly excluded from consideration in [7].

Consider initialising the return-map algorithm at point C of Figure 2. The solution is point D. However, because of the linear-dependence, the active set could consist of any two, or all three, of the yield surfaces. There is no unique solution to the return-map algorithm in this case.

There are common situations where linear dependence occurs. Because the flow potentials are symmetric $r^\alpha_{ij} = r^\alpha_{ji}$, the flow-direction space is only six dimensional, so with $N > 6$, linear

DOI: 10.1002/nme

dependence will always occur (in the Cosserat case, $N > 9$ is needed). In face-centred-cubic crystal plasticity $N = 12$ and linear-dependence of the flow potentials r^α have been explored in a series of articles [11–17]. In Mohr–Coulomb plasticity, $N = 6$, but because this model is formulated in principal stress space, only three r^α are linearly independent. It is also impossible to avoid linear dependence in complicated geomechanical plasticity models such as those explored in Section 6.

In principal, it is also possible that the h vectors are linearly dependent, but the same techniques may be used.

Our proposal to address linear dependence is to remove from the system of Equation (2), linearly dependent constraints with smallest

$$d_\alpha = \frac{f_\alpha}{\sqrt{\sum_{ij} \frac{\partial f_\alpha}{\partial \sigma_{ij}} \frac{\partial f_\alpha}{\partial \sigma_{ij}}}} \,. \tag{3}$$

The denominator addresses the possibility that yield functions might not be homogeneous of degree one. We retain constraints with the largest d_α because those yield functions are the 'most violated'. In the nonlinear situation, the measure d_α will only be an approximation of the distance to the yield function.

This proposal for eliminating linearly dependent constraints is a generalisation to the nonlinear case of the crystal-plasticity approach described by [22], in which the plastic power is minimised.

Computationally, at every NR iteration, a singular value decomposition of the flow potential r^α ($\alpha \in J$) 'vectors' is performed, and if the smallest singular value is less than ϵ times the largest singular value, then this set of r^α are deemed linearly dependent. The choice $\epsilon = 10^{-4}$ appears to work well in our tests with the implementation in MOOSE [19]. After eliminating constraints using their current values of d_α, a linearly independent set of flow potentials r^α remains. Denote this set of linearly independent and active constraints by $\bar{J}$:

$$\bar{J} = \left\{ \alpha \mid \text{ all } r^\alpha_{ij}(\sigma, q) \text{ are linearly independent} \right\} \,. \tag{4}$$

One NR iteration solving the system

$$0 = \begin{pmatrix} f_\alpha \quad \text{with } \alpha \in \bar{J} \\ \sigma^{\text{trial}}_{ij} - \sigma_{ij} - E_{ijkl} \sum_{\alpha \in J} \dot{\gamma}^\alpha r^\alpha_{kl} \\ q^{\text{old}}_a - q^a - \sum_{\alpha \in J} \dot{\gamma}^\alpha h^\alpha_a \end{pmatrix} , \tag{5}$$

is performed. The summations in Equation (5) correctly include all active constraints ($\alpha \in J$), but the degrees of freedom for this NR iteration are only the plastic multipliers $\dot{\gamma}^\alpha$ for $\alpha \in \bar{J}$, as well as σ and the internal variables q^a. After the solution is found, the eliminated constraints are re-introduced for the next NR iteration.

With this procedure, it is possible to enter an infinite cycle, as explained in the example in the following. To avoid this, if some distances d_α are equal, they are perturbed by a small random amount.

As an example, consider Figure 3, where the algorithm is initialised at point A. There is linear dependence of the flow directions and the distances satisfy $d_2 < d_3 < d_1$, so the second yield function ($\alpha = 2$) is eliminated for the first NR iteration. This iteration's solution is at point B. For the second NR-iteration, there is still linear-dependence and $0 = d_1 = d_3 < d_2$. There are two choices.

(1) $\alpha = 1$ is eliminated. Then the solution is at point D. At this point $0 = d_2 = d_3 < d_1$. Again, there are two choices. If $\alpha = 2$ is eliminated, the solution is once again point B, and an infinite cycle could result. If $\alpha = 3$ is eliminated, the solution is point C.
(2) $\alpha = 3$ is eliminated. Then the solution is point C.

Which of these two choices is made, and whether an infinite cycle occurs, depends in practice on the computer architecture and the code compiler. However, randomly perturbing the d_α prevents

DOI: 10.1002/nme

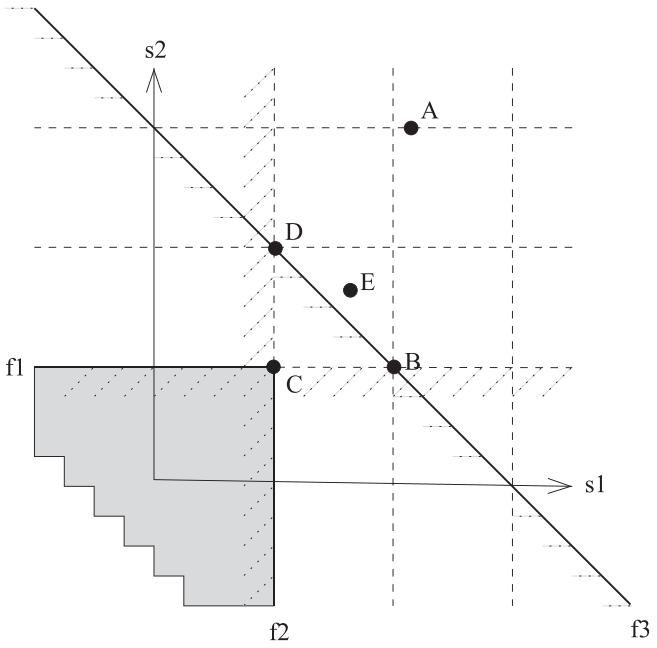

Figure 3. Two-dimensional stress space parameterised by s_1 and s_2, which could be principal stress components, for example. Three yield surfaces are shown: $f_1 = s_2 - 1$, $f_2 = s_1 - 1$ and $f_3 = s_1 + s_2 - 3$. The dashed lines indicate the admissible regions for each yield function, while the shading shows the admissible regions E_{adm}, which is the intersection of the three admissible regions. Points A, B etc. are described in the text. Points A, B, D and E are all inadmissible.

this endless cycle. Note that, while point C is actually the desired solution, this solution path just described actually violates the Kuhn–Tucker conditions because the plastic multiplier $\dot{\gamma}^3 > 0$. This is addressed later.

4. LINE-SEARCH RESIDUAL

In nonlinear situations, it is necessary to use a line search in conjunction with the NR method. While this is mentioned in [7], it is not detailed. We employ a traditional cubic line search [23] in our implementation [19]. The key quantity is the line-search residual.

Define the following tolerances: f_α^{tol} for $\alpha = 1, \ldots, N$, which are tolerances on the yield functions; ϵ^{tol}, which is a tolerance on the plastic strain; q_a^{tol}, which are tolerances on the internal constraints. We propose that the line search should minimise the residual, ϕ, given by

$$\begin{aligned}\phi^2 = &\sum_{\alpha\in\bar{J} \text{ and } \dot{\gamma}^\alpha>0} \left(\frac{f_\alpha}{f_\alpha^{\text{tol}}}\right)^2 + \sum_{\alpha\in\bar{J} \text{ and } \dot{\gamma}^\alpha=0} \left(\frac{\max(f_\alpha,0)}{f_\alpha^{\text{tol}}}\right)^2 \\ &+ \sum_{\alpha\in J-\bar{J}} \left(\frac{\max(f_\alpha,0)}{f_\alpha^{\text{tol}}}\right)^2 \\ &+ \left|\frac{(\sigma^{\text{trial}}-\sigma)E^{-1} - \sum_{\alpha\in J}\dot{\gamma}^\alpha r^\alpha}{\epsilon^{\text{tol}}}\right|^2 \\ &+ \sum_{a=1}^{Q}\left(\frac{q_a^{\text{old}} - q_a - \sum_{\alpha\in J}\dot{\gamma}^\alpha h_a^\alpha}{q_a^{\text{tol}}}\right)^2 .\end{aligned} \tag{6}$$

Here $|x_{ij}|^2 = \sum_{i,j} x_{ij}x_{ij}$.

DOI: 10.1002/nme

This residual needs explanation. The first two terms exactly express the Kuhn–Tucker conditions: if the plastic multiplier is positive ($\dot\gamma^\alpha > 0$), we desire the yield function to be zero ($f_\alpha = 0$); while if $\dot\gamma = 0$, we only need $f_\alpha \leqslant 0$.

The third term involves just the linearly dependent constraints. These still need to satisfy the Kuhn–Tucker conditions. However, if the yield function $f_\alpha \leqslant 0$, then the constraint should not contribute to the residual, because otherwise flow inside yield surfaces would be penalised. For example, consider starting at point E in Figure 3, where the third yield function ($\alpha = 3$) is removed through linear dependence. The solution is point C, where $f_3 < 0$, so it does not contribute to the residual. The alternative residual, containing the terms

$$\sum_{\alpha\in J-\bar J \text{ and } \dot\gamma^\alpha>0} \left(\frac{f_\alpha}{f_\alpha^{\text{tol}}}\right)^2 + \sum_{\alpha\in J-\bar J \text{ and } \dot\gamma^\alpha=0} \left(\frac{\max(f_\alpha,0)}{f_\alpha^{\text{tol}}}\right)^2, \tag{7}$$

is incorrect, even though it expresses the Kuhn–Tucker conditions. For example, starting at point A in Figure 3, and flowing to point B, then eliminating f_3, the algorithm would not arrive at point C because the line search would penalise moving away from $f_3 = 0$, and no solution would be found.

The fourth and fifth terms contain all active α. In particular, the linearly dependent α are included, even though $\Delta\dot\gamma^\alpha = 0$ for this NR iteration. This ensures the correct equations specified in Section 2 are being solved.

Finally, when the NR procedure has converged, if $\dot\gamma^\alpha > 0$ and $f_\alpha < 0$, apparently violating the Kuhn–Tucker conditions, if the flow direction r^α is part of a linearly dependent set, then this plastic multiplier $\dot\gamma^\alpha$ can be set to zero by modifying the other $\dot\gamma$ within this set, and the internal variables q_a where appropriate.

5. RETURN-MAP ALGORITHM

Simo, Kennedy and Govindjee [7] describe algorithms for returning to the yield surface, which primarily involve determining the set of active constraints, and here we describe our extension to these algorithms, which is implemented in [19].

(0) **Preliminary.** It is occasionally efficient to simply apply the strain increment in multiple small steps. If the algorithm fails in some way and returns to this step, then the strain increment is reduced. An input to the algorithm is the minimum fraction of applied strain increment allowed before the algorithm gives up entirely and signals failure to the calling routine.

Another input is the maximum number of NR iterations allowed.

Another input is the 'deactivation scheme'. In [7], two schemes for deactivating constraints are described, which we call:

(a) 'optimised', where constraints are deactivated as soon as their plastic multipliers, $\dot\gamma^\alpha$, become negative.
(b) 'safe', where constraints are deactivated if their $\dot\gamma^\alpha$ is negative when the NR process has converged.

Our implementation also provides another scheme:

(c) 'dumb', where all possible sets of active constraints are tried iteratively. This is a useful in case of repeated failure using the other schemes, but is inefficient.

Our implementation allows one scheme to be tried first, and if that fails, another is tried etc. For instance, first 'optimised' is tried, then 'safe', and finally 'dumb'. Usually just using 'safe' works well.

At this stage, the deactivation scheme is set to that, which the user specified. If the user has allowed it, the scheme may subequently change, as described in the following.

(1) **Initialising the set of active constraints.** If one were to use this algorithm for simple plasticity models, such as Mohr–Coulomb, given $(\sigma^{\text{trial}}, q^{\text{old}})$, the set of constraints leading to a

DOI: 10.1002/nme

solution would be known *a priori* (for example [5]). This is extremely advantageous because initialising the active constraints correctly avoids wasted computational effort. For example, in Figure 2, if $\sigma_1^{\text{trial}} > 1$ and $\sigma_2^{\text{trial}} > 1$ (point C, for example), then both f_1 and f_2 should be made active and f_3 inactive. Similarly, if $\sigma_1^{\text{trial}} > 1$ but $\sigma_2^{\text{trial}} \leqslant 1$ (point E, for example), then only f_2 should be made active. However, for complicated models containing many yield surfaces, this is not possible.

If there is no *a priori* knowledge then

$$J = \left\{\alpha \mid f_\alpha(\sigma^{\text{trial}}, q^{\text{old}}) > f_\alpha^{\text{tol}}\right\} \tag{8}$$

Alternately, if the scheme is 'dumb' then the different possibilities for the active set J are sorted in descending order according to their 'total distance' values $\sum_{\alpha\in J} d_\alpha$, and the first possibility is chosen.

If the active set J is empty, then this is a purely elastic step and no return-mapping is required. Assume this is not the case.

(2) **Initialising NR.** Set $\sigma = \sigma^{\text{trial}}$, $q = q^{\text{old}}$ and $\dot\gamma = 0$. The algorithm may return here after a failure of some sort, so check whether the current set J is empty, and if so, then signal 'failure' and to Step 0. The NR procedure is performed between Steps 3 and 8.

(3) **Single NR step with line search.** This may involve temporarily deactivating linearly dependent constraints, as described previously. If the line search fails to reduce the residual, signal 'failure' and jump to Step 9.

(4) **Optimised?** If the scheme is 'optimised' and if any plastic multipliers $\dot\gamma^\alpha < 0$, remove those from the active set J. In this case, if there are no constraints remaining, signal 'failure' and jump to Step 9; otherwise, redo the NR step (Step 3) with the new active set J.

(5) **Linearly dependent constraints?** If there are linearly dependent constraints then they are re-introduced into the active set J. However, before the reintroduction, if the residual $\phi \leqslant 1$ (indicating convergence), and the yield functions satisfy $f_\alpha \leqslant f_\alpha^{\text{tol}}$ for all α, then the NR process has converged. In this case, signal 'success' and jump to Step 9. This check ensures that the algorithm correctly signals 'success' for the paths B to C in Figure 3, for example. Otherwise, simply recalculate the residual ϕ with the updated J and continue.

(6) **Converged?** If the residual $\phi \leqslant 1$, then NR process has converged, so signal "success" and go to Step 9.

(7) **Check number of NR iterations.** If this exceeds the upper bound, signal 'failure' and go to Step 9.

(8) **End of NR loop.** Go to Step 3 to perform another NR iteration.

(9) **Post NR.** If 'failure' was signalled, then:

   (a) If the scheme can be changed (for instance, 'optimised' to 'safe'), then change the scheme and go to Step 1.
   (b) Else, if the scheme is 'dumb', try a new set of active constraints and go to Step 2. If there are no further untried combinations, signal 'failure' and go to Step 0.
   (c) Otherwise, signal 'failure' and go to Step 0.

The following steps can only be reached if 'success' was signalled by the NR procedure.

(10) **Check Kuhn–Tucker.** Check that: the plastic multipliers $\dot\gamma^\alpha \geqslant 0$ for all α, and $\dot\gamma^\alpha f_\alpha = 0$ for $\alpha \in J$. Note that return to the yield surface $\left(f_\alpha < f_\alpha^{\text{tol}}\right)$ is guaranteed for the active constraints ($\alpha \in J$) because the NR procedure signalled 'success'. If one of these checks fails then:

   (a) If the scheme is not 'dumb', then remove from the active set J all α with $\dot\gamma^\alpha f_\alpha \neq 0$. If this removes no α, then only remove the α with the smallest $\dot\gamma^\alpha$. This is not identical to the 'safe' scheme described in [7], where *all* constraints violating the Kuhn–Tucker conditions are removed. We have found that removing constraints one by one succeeds more frequently. If the new set J has not already been tried, then

DOI: 10.1002/nme

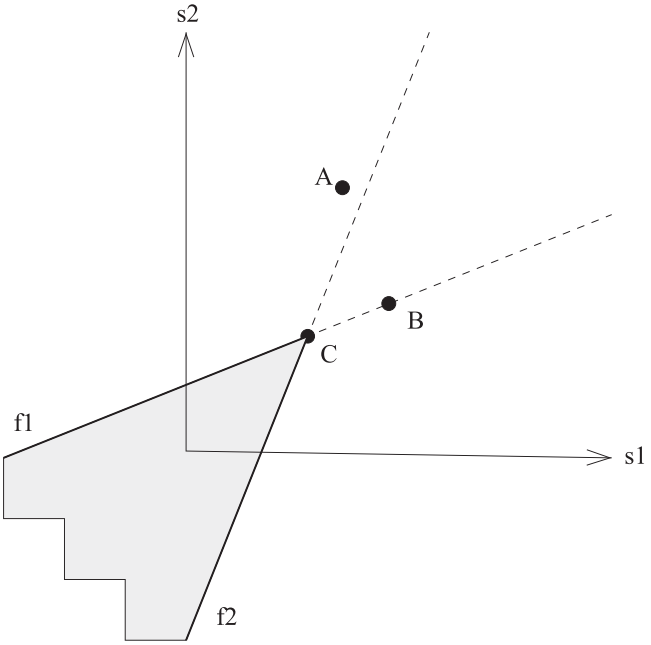

Figure 4. Two-dimensional stress space parameterised by $s1$ and $s2$. Two yield surfaces, $f1$ and $f2$, define a wedge-shaped admissible region. Points A, B and C are described in the text.

go to Step 2. Else, if the scheme can revert to 'dumb', do this and go to Step 2. Else, signal 'failure' and go to Step 0.

(b) If the scheme is 'dumb', try a new set of active constraints and go to Step 2. If there are no further untried combinations, signal 'failure' and go to Step 0.

(11) **Check Admissible.** If $f_\alpha \leqslant f_\alpha^{\text{tol}}$ for all α (this is gauranteed for $\alpha \in J$ by Step 10), then the algorithm has successfully provided a solution, so exit. Otherwise:

(a) Try adding those constraints that are in their inadmissible region $\left(f_\alpha > f_\alpha^{\text{tol}}\right)$ to the set J, and if the new J has not yet been tried, then go to Step 2. This eventuality was not described in [7]. An example is shown in Figure 4. It was also described in the context of hardening in [18]. If the algorithm starts at Point A, then the initial set of constraints is $J = \{1\}$. The algorithm returns to point B, and the Kuhn–Tucker check is successful. However, $f_2 > 0$. Adding 2 to the set J causes the algorithm to converge correctly to point C.

(b) Else, if the scheme can be changed (for instance, 'optimised' to 'safe'), then change the scheme and go to Step 1.

(c) Else, if the scheme is 'dumb', try a new set of active constraints and go to Step 2. If there are no further untried combinations, signal 'failure' and go to Step 0.

(d) Else, signal 'failure' and go to Step 0.

6. QUANTITATIVE EXAMPLES

In this section, examples are presented to illustrate the computational cost of various aspects of the general algorithm. Examples are also presented, that illustrate the necessity of using the three new contributions presented herein (linear-dependence handling, the line search and adding constraints). Finally, because the algorithm produces the exact result (up to specified tolerances), it is used to quantify the accuracy of another common algorithm.

In each example, random displacements are applied to the nodes of a finite-element mesh of 12 5000 hexahedral elements, and the return-map algorithm described previously is used to return

DOI: 10.1002/nme

to the admissible region. The random displacements are chosen so that the components of the strain tensor lie within $\pm 10\%$. Because there are 8 quadrature points in each element, the algorithm is used 10^6 times.

6.1. Computational cost of the basic NR process and linear-dependence checking

The computational costs of the NR process and the linear-dependence algorithm are estimated in this section. Three simulations are performed. The first has a single yield surface, described by $f_1 = \sigma_{zz} - L$, where L is a large number chosen so that the stress never enters the inadmissible region. This simulation acts as a reference to the other two. The second has a single yield surface, $f_2 = \sigma_{zz}$. Approximately, half of the million trial stresses will be inadmissible. The third has two yield surfaces: $f_2 = \sigma_{zz}$ and $f_3 = \sigma_{zz} - \epsilon$, where ϵ is a small value. Again, approximately half of the million trial stresses will be inadmissible, and f_3 will have to be eliminated in almost all of these cases because of linear dependence.

No simulation needs a line search, reactivation of constraints, or any of the other constraint handling procedures mentioned previously. The results are presented in Table I.

The 1.53 in Table I may be decomposed as a 1 (pure elasticity) and a 0.53, which quantifies the cost of performing a single NR iteration for approximately half of the trial stresses. Similarly, $1.72 = 1 + 0.72$, so $0.72/0.53 \approx 1.36$ quantifies the extra cost of checking and enforcing linear dependence.

These simulations all use the full algorithm described previously, for instance, all the components of stress are included in Equation (5), even though this is clearly not unnecessary in this example. Obviously, it would be foolish to use the full algorithm for a 'production' simulation with such simple plasticity models: this example is only meant to quantify computational costs.

6.2. Mohr–Coulomb plasticity

This example is designed to quantify the efficiency of the full constraint-handling and linear-dependence procedures. Mohr–Coulomb plasticity is explored, with Young's modulus 25 GPa, Poisson's ratio 0.2, cohesion 1 MPa, friction angle 30 deg and dilation angle 5 deg. The first model uses six planar surfaces, so that all the multisurface machinery described previously must be used. The 'optimised' deactivation scheme is used. The second model uses a single-surface approximation to the Mohr–Coulomb yield surface, as described in [6], so that none of the multisurface machinery actually need to be used, but the same general algorithm is still used.

Analysis of Table II reveals that the computational cost per NR iteration of the 'six surfaces' simulation (0.87) is over twice that of the 'smoothed' simulation (0.43). This is because all the aspects of the algorithm presented previously are needed (linear-dependence checking, and deactivating and

Table I. Relative compute times for three linear models.

Simulation	Relative compute time
No plasticity	1
Single surface	1.53
Two surfaces	1.72

Table II. Comparison between two Mohr–Coulomb formulations.

Simulation	T	$\overline{\mathrm{NR}}$	$\bar{T}_{\mathrm{NR}}$	LD
Six surfaces	1	1.03	0.97	0.88
Smoothed surface	10.3	23.7	0.43	0

Relative compute time (T), average number of NR iterations ($\overline{\mathrm{NR}}$), average relative time per NR iteration ($\bar{T}_{\mathrm{NR}}$) and fraction of NR iterations that a linear dependence (LD) check was needed.

DOI: 10.1002/nme

reactivating constraints) and that the maximum size of the linear system is 13×13 (six stress components, one internal parameter and six active yield surfaces), rather than 8×8 in the 'smoothed' case. Nevertheless, the linearity of the problem results in a much smaller overall compute time.

For Mohr–Coulomb plasticity, highly efficient algorithms have been devised (for instance [3]), indeed, an NR approach is not even necessary, and our tests suggest that such a specialised algorithm is approximately twice as fast as the six-surface version using the current algorithm. Thus, as stated in the introduction, using the full multi-surface algorithm for Mohr–Coulomb plasticity alone would be foolish for 'production' simulations: this example is only meant to quantify computational costs in a familiar setting.

6.3. Sedimentary geomechanical plasticity

This example demonstrates the necessity of using the new features presented in this paper in a realistic scenario where no specialised algorithms or smoothing techniques are known. The combination of the four plasticity model described later is frequently used when modelling sedimentary rocks. The various deactivation schemes are compared with maximum number of NR iterations set to 30 before the applied strain is halved, and the 'dumb' scheme only being used if the applied strain is reduced to 10% of its original value.

Four plasticity models are used with Young's modulus 3 GPa and Poisson's ratio 0.25. The first is smoothed Mohr–Coulomb [6] with cohesion 40 MPa, friction angle 35 deg and dilation angle 5 deg. The second is tensile (Rankine) failure ($f_2 = \sigma_I - T_r$, where σ_I is the leading principal stress), smoothed [6] with rock tensile strength $T_r = 1$ MPa. The third is ubiquitous-joint tensile failure ($f_3 = \sigma_{zz} - T_j$) with joint tensile strength $T_j = 1$ kPa. The fourth is ubiquitous-joint shear failure $\left(f_4 = \sqrt{\sigma_{zx}^2 + \sigma_{zy}^2} + \sigma_{zz} \tan\phi_j - C_j\right)$ with cohesion $C_j = 0.1$ MPa, friction angle $\phi_j = 25$ deg and dilation angle 5 deg. This is an example where the six-surface Mohr–Coulomb plasticity is inefficient to use because it is highly nonlinear when expressed as a function of the components of the stress tensor. These plasticity models with different physical moduli and a range of applied strains are studied in Section 6.6.

Table III reveals that the Optimised scheme is the most computationally efficient in this case. More importantly, for this scheme, 0.02% of return-maps would fail without removing linearly dependent constraints, 44% would likely fail without using a line search and 33% would fail if constraints were not added at Step 11 of the algorithm. In summary, without using the new features presented in this paper, the return-map algorithm would fail in approximately 50% of cases for this common geomechanics plasticity.

6.4. Capped Mohr–Coulomb plasticity with softening

Unstratified geomaterials are frequently modelled using Mohr–Coulomb plasticity with a tensile (Rankine) cap that models the material's weak tensile strength. In this section, we demonstrate convergence of the algorithm for such a geomaterial with softening. Our algorithm solves simultaneously for the stress, the internal parameters and the plastic strain components (and hence the

Table III. Comparison between deactivation strategies for sedimentary geomechanics plasticity.

Deactivation scheme	T	$\overline{\text{NR}}$	$\text{LD}_{\text{removed}}$	LS	ADD
Optimised	1	42.1	0.0002	0.44	0.33
Safe	1.65	56.0	0.0015	0.60	0.06
Optimised to safe	1.44	54.0	0.0002	0.46	0.33
Optimised to safe to dumb	1.77	71.1	0.0005	0.46	0.33

Relative compute time (T), average number of NR iterations per return-map ($\overline{\text{NR}}$), fraction of return-maps that required removing linearly dependent directions ($\text{LD}_{\text{removed}}$), fraction of return-maps that required a line search (LS) and fraction of return-maps that required constraints to be added in Step 11 of the algorithm (ADD).

DOI: 10.1002/nme

Mohr–Coulomb cohesion, friction angle etc) and does not 'fix' the internal parameters during the return-map process as some other algorithms do.

This example illustrates that knowing the initial set of active constraints is extremely advantageous as mentioned in Step 1 of the algorithm, previously. Without softening, these may be determined (for example [3], and for a commercial implementation see [24, 25]). With softening, it is sometimes necessary to resort to the 'dumb' scheme, which is computationally expensive, as quantified later.

An isotropic material with Young's modulus 3 GPa and Poisson's ratio 0.25 is used. Two plastic formulations are compared, and both simulations use the return-map algorithm detailed previously:

(1) Mohr–Coulomb plasticity is modelled using six yield surfaces, and the tensile cap has three yield surfaces. In terms of the notation previously, this means that $N = 6 + 3 = 9$. The Mohr–Coulomb friction angle softens from 40 to 30 deg, the cohesion from 20 to 10 MPa, the dilation from 10 to 5 deg, and the tension cutoff T_r from 15 MPa to 0. In terms of the notation previously, this means $Q = 2$, and $h_a^1 = -1$ for $a = 1, \ldots, 6$ (the Mohr–Coulomb yield functions), and $h_a^2 = -1$ for $a = 7,\ 8,\ 9$ (the tensile-failure yield functions).
(2) Smoothed Mohr–Coulomb plasticity is used alone, with friction angle, cohesion and dilation softening as previously mentioned. The smoothing is defined in [6], with hyperbolic tip smoothing parameter chosen to be 21.4 MPa. This means that the material can support approximately zero tensile stress, which approximately models the weak tensile strength of the material. This is similar to the plastic models described in [26] where smoothed cap-plasticity models for geomaterials are studied.

The variations in plastic parameters occur between 0% and 100% strain, so that the applied strains of around 10% do not immediately soften the material, which would make this test rather pointless (a range of strains are studied in Section 6.6). For all plastic parameters, a cubic softening relationship is used, for example, when $0 \leqslant q_2 \leqslant 1$ the tensile strength is:

$$T_r(q_2) = 30 \times (q_2 - 0.5)^3 - 22.5(q_2 - 0.5) + 7.5\ , \tag{9}$$

which is measured in MPa, and $T_r(q_2) = 0$ for $q_2 > 1$.

Table IV reveals that the non-smoothed version is over twice as fast as the smoothed version, even though the computational cost per NR iteration is greater, for the reasons explained in Section 6.2. The non-smoothed version has two further advantages: (1) the cohesion and tensile strength are separated, instead of controlling the tensile strength via the hyperbolic smoothing parameter, and this makes softening easier to conceptualise; (2) finite element quadrature points can be categorised as 'failed in shear (Mohr–Coulomb)' or 'failed in tension', or both, instead of simply 'failed', which is advantageous from an engineering perspective.

However, what is most interesting from a computational viewpoint is the fraction of return maps that have to resort to the 'dumb' scheme, and the corresponding large number of NR iterations. This is due to an incorrect choice of initially active constraints and occurs in spite of the judicious use of activating and deactivating constraints during the NR process, as described in Section 5. Although resorting to the 'dumb' scheme occurs only 0.31% of the time, the reader should remember this case has only nine yield surfaces: the chance of having to resort to the 'dumb' scheme surely

Table IV. Comparison between two formulations of unstratified geomechanics plasticity.

Plastic formulation	T	$\overline{\text{NR}}$	LD	LS	ADD	Dumb	$\overline{\text{NR}}_{\text{Dumb}}$
Non-smoothed	1	2.62	0.296	0.0033	0.00519	0.0031	64.8
Smoothed	2.1	9.08	0	0.538	0	0	0

Notation is identical to Table III, along with fraction of return-maps that resorted to the 'dumb' scheme (Dumb) and average number of NR iterations per return-map for those return-maps $\left(\overline{\text{NR}}_{\text{Dumb}}\right)$.

DOI: 10.1002/nme

increases with the number of yield surfaces, and the corresponding number of NR iterations will exponentiate. From a practical point of view, this is the most important weakness in the current algorithm, and for general nonlinear yield functions with hardening/softening, it is not immediately obvious how to formulate a general strategy for always choosing the correct initially active set, and always correctly activating and deactivating constraints during the NR process. Any computational cost associated with the other aspects of the algorithm (formulating in full-dimensional stress space, linear-dependence handling, line searching and constraint handling) are comparitively modest when compared with the risk of having to resort to the 'dumb' scheme.

6.5. Compressive failure with capped J2 plasticity

This section considers J2 plasticity ($f_1 = \sqrt{3 s_{ij} s_{ij}} - S$, where the deviatoric stress is $s_{ij} = \sigma_{ij} - \delta_{ij}/3$) combined with a mean-stress cap ($f_2 = -\sigma_{ii}/3 - C$). In this section, we demonstrate convergence of the algorithm when S and C soften. This combination of plastic models could be taken as a simplified model of a porous geomaterial with low compressive strength: a similar, but smoothed, model is employed to study compaction bands in [27].

The yield strength softens from 20 to 10 MPa, and the compressive strength from 15 to 5 MPa, in the same way Equation (9). An isotropic material with Young's modulus 3 GPa and Poisson's ratio 0.25 is used. In the notation of Section 2, $N = 2$ and $Q = 2$, and the system size of Equation (2) is $2 + 6 + 2 = 10$, assuming both yield functions are active.

Clearly in this case, there is no possibility of linear dependence of the flow directions, because the flow directions r^1 and r^2 are perpendicular. Moreover, because the flow directions are independent of the softening, the initial choice of active constraints specified in Equation (8) is correct.

Table V presents a comparison of this model with the multi-surface formulation of unstratified geomaterials in Section 6.4. The results reveal the costs involved in increasing from a two-yield-surface model to a nine-yield-surface model where the initial set of active constraints is not easily defined: the former is over six times faster, even though the average number of NR iterations is similar.

6.6. Sedimentary geomechanical plasticity — comparison with a pragmatic approach

The combination of four plasticity models described in Section 6.3 is studied further here. These are often used to model sedimentary rock masses, but without the new methods explained in this paper, approximate return-map algorithms have been necessary to date. This section compares the exact approach of this paper, with a pragmatic approach taken in the FLAC commercial software [24, 25].

The parameters used are as follows. Young's modulus 16 GPa; Poisson's ratio 0.25; cohesion softening from 10 to 1 MPa; friction angle softening from 35 to 25 deg (so unconfined compressive strength softens from 38 to 3 MPa); dilation angle softening from 10 to 5 deg; tensile strength softening from 1 to 0; joint tensile strength softening from 1 to 0 kPa; joint cohesion softening from 100 to 10 kPa; joint friction angle softening from 25 to 12 deg; joint dilation angle 5 deg. All the softening parameters obey cubic equations similar to Equation (9), but in contrast, they reach their assymptotic values at 1% strain, which is more realistic for rock plasticity.

The algorithm described in Section 5 solves the return mapping problem of Section 2 exactly, and so it may be used to assess the accuracy of other approaches. For the four plasticity models, a pragmatic approach [24, 25] is to (1) return to the Mohr–Coulomb plus tensile failure yield surface;

Table V. Comparison between the multi-surface formulation unstratified geomechanics plasticity of Section 6.4, and the capped-J2 plasticity of Section 6.5.

Plastic formulation	T	$\overline{\text{NR}}$	LD	LS	ADD	Dumb	$\overline{\text{NR}}_{\text{Dumb}}$
Section 6.4	1	2.62	0.296	0.0033	0.00519	0.0031	64.8
Capped-J2	0.16	2.00	0	0	0	0	0

Notation is identical to Table IV.

DOI: 10.1002/nme

and then, (2) return to the ubiquitous-joint yield surface. This is highly efficient, because both (1) and (2) can use well-known optimisations, but the method introduces some error as quantified in the following.

Two measures of error are used. The first is

$$L_\sigma = \underset{\text{p}}{\text{Mean}} \sqrt{\left(\sigma_{ij}^{\text{ap}} - \sigma_{ij}^{\text{ex}}\right)\left(\sigma_{ij}^{\text{ap}} - \sigma_{ij}^{\text{ex}}\right)}\,, \tag{10}$$

where σ^{ap} is the stress after returning with the pragmatic stepwise method, σ^{ex} is the stress after returning using the exact approach of this paper, and the mean is taken over only quadrature points that have experienced plastic deformation. The second measure is

$$L_\epsilon = \underset{\text{p}}{\text{Mean}} \frac{\sqrt{\left(\sigma_{ij}^{\text{ap}} - \sigma_{ij}^{\text{ex}}\right)\left(\sigma_{ij}^{\text{ap}} - \sigma_{ij}^{\text{ex}}\right)}}{|\sigma^{\text{applied}}|}\,, \tag{11}$$

where $|\sigma^{\text{applied}}|$ is the magnitude of the applied stress increment. In Table VI, these errors appear after only return (1) (Mohr–Coulomb plus tensile) has been employed; after (1) then (2) and so on.

The results of Table VI are for applied strain components within $\pm 0.1\%$. FLAC employs explicit time stepping, so such large applied strain components may be unusual in that setting; however, they are definitely possible for an implicit solver. In either case, the current algorithm allows analysis of the strain dependence of the errors, and the results are shown in Table VII. (For the interested reader: these results are almost independent of softening.) It is interesting that in this case even substepping, which usually increases accuracy (e.g. [28]) would not work well here, because the error accrued is always roughly 0.05 times the applied stress increment.

Table VI. The absolute error L_σ and the fractional error L_ϵ in the stepwise pragmatic approaches for the return-mapping algorithm of sedimentary-rock plasticity.

Method	L_σ (MPa)	L_ϵ
After (1) only	3.0	0.13
After (1) then (2)	1.6	0.07
After (2) only	5.5	0.24
After (2) then (1)	1.4	0.06

Table VII. The absolute error L_σ, and the fractional error L_ϵ, in the '(1) then (2)' stepwise pragmatic approach for the return-mapping algorithm of sedimentary-rock plasticity, as a function of applied strain range.

Range of applied strains	L_σ (MPa)	L_ϵ
$\pm 0.001\%$	0	0
$\pm 0.01\%$	0.09	0.06
$\pm 0.1\%$	1.6	0.07
$\pm 1\%$	4.0	0.04
$\pm 10\%$	61	0.05
$\pm 100\%$	700	0.05

Applied strain components lie within $\pm 0.1\%$.

DOI: 10.1002/nme

7. CONCLUSIONS

The return-map algorithm for general multisurface plasticity presented in [7] failed to converge in a number of test cases. We have presented three new contributions: (1) a method of handling linearly dependent flow directions; (2) a residual for use in a line search; and, (3) an updated algorithm employing (1) and (2), and the possibility of adding constraints to the system. The algorithm has successfully converged for all cases considered.

Examples have quantified the computational cost of various aspects of the algorithm. In these examples, a single NR iteration with line search costs approximately as much as an elastic calculation, while when linear dependence is checked and enforced, this rises to between 1.5 and 2 times (Tables I and II). Studying a common set of plastic models describing sedimentary rock plasticity has shown the necessity of using our three new contributions, because without them, the algorithm would fail to converge in approximately 50% of cases (Table III). Other geomechanically inspired models with softening were also shown to converge. Finally, the algorithm was shown to converge for sedimentary rock plasticity with realistic softening, and it was used to quantify the accuracy of an approximate approach used by the FLAC software [24, 25].

The algorithm allows an arbitrary number of nonlinear yield functions with hardening and softening to be combined, and will provide the returned stress, the internal variables and the plastic strains exactly (to within tolerances, and assuming that the hardening and softening are simple enough that there is a unique solution). Computationally, efficient strategies for specific simple models (such as Mohr–Coulomb) have been devised, but this algorithm treats the return-map problem generally, so it would be inefficient to use for such cases. It is suited to the general setting, where no specific strategies are known, and here the main weakness we have identified is the difficulty of choosing the initial set of active constraints. Despite our sophisticated techniques for deactivating and activating constraints during the return process, there are cases for which the algorithm must resort to the 'dumb' scheme (Table IV), which is computationally very costly.

ACKNOWLEDGEMENTS

The authors thank Ioannis Stefanou (ENPC, France) for confirming that he also used a singular-value decomposition, and that the original algorithm of Simo, Kennedy and Govindjee sometimes failed to converge in his applications. His work was recently published in Godio M, Stefanou, I, Sab K, Sulem J. Multisurface plasticity for Cosserat materials: Plate element implementation and validation. International Journal of Numerical Methods in Engineering 2016; DOI: 10.1002/nme.5219.

REFERENCES

1. Feenstra PH, de Borst R. A composite plasticity model for concrete. *International Journal of Solids and Structures* 1996; **5**:707–730. DOI: 10.1016/0020-7683(95)00060-N.
2. Clausen J, Damkilde L, Andersen L. Efficient return algorithms for associated plasticity with multiple yield planes. *International Journal of Numerical Methods in Engineering* 2006; **66**:1036–1059. DOI: 10.1002/nme.1595.
3. Clausen J, Damkilde L, Andersen L. An efficient return algorithm for non-associated plasticity. *Computers and Structures* 2007; **85**:1795–1807. DOI: 10.1016/j.compstruc.2007.04.002.
4. Karaoulanis FE. Implicit numerical integration of nonsmooth multisurface yield criteria in the principal stress space. *Archives of Computational Methods in Engineering* 2013; **20**:263–308. DOI: 10.1007/s11831-013-9087-3.
5. Pankaj P, Bicanic N. Detection of multiple active yield conditions for Mohr–Coulomb elastoplasticity. *Computers and Structures* 1997; **62**:51–61. DOI: 10.1016/S0045-7949(96)00267-2.
6. Abbo AJ, Lyamin AV, Sloan SW, Hambleton JP. A C2 continuous approximation to the Mohr–Coulomb yield surface. *International Journal of Solids and Structures* 2011; **48**:3001–3010. DOI: 10.1016/j.ijsolstr.2011.06.021.
7. Simo JC, Kennedy JG, Govindjee S. Non-smooth multisurface plasticity and viscoplasticity. Loading/unloading conditions and numerical algorithms. *International Journal for Numerical Methods in Engineering* 1988; **26**:2161–2185. DOI: 10.1002/nme.1620261003.
8. Simo JC, Hughes TJR. *Computational Inelasticity*, Interdisciplinary Applied Mathematics, vol. 7. Springer: New York, 1998. DOI: 10.1007/b98904.
9. Ottosen NS, Ristinmaa M. Corners in plasticity-Koiter's theory revisited. *International Journal of Solids and Structures* 1996; **33**:3697–3721. DOI: 10.1016/0020-7683(95)00207-3.
10. Doghri I. *Mechanics of Deformable Solids: Linear and Nonlinear, Analytical and Computational Asepects*. Springer-Verlag: London, 2000. DOI: 10.1007/978-3-662-04168-0.

11. Borja RI, Wren JR. Discrete micromechanics of elastoplastic crystals. *International Journal for Numerical Methods in Engineering* 1993; **36**:3815–3840. DOI: 10.1002/nme.1620362205.
12. Cuitino AM, Ortiz M. Computational modelling of single crystals. *Modelling and Simulation in Materials Science and Engineering* 1993; **1**:225–263. DOI: 10.1088/0965-0393/1/3/001.
13. Maudlin PJ, Wright SI, Kocks UF, Sahota MS. An application of multisurface plasticity theory: yield surfaces of textured materials. *Acta Materialia* 1996; **44**:4027–4032. DOI: 10.1016/S1359-6454(96)00039-0.
14. Sawischlewski E, Steinmann P, Stein E. Modelling and computation of instability phenomena in multisurface elasto-plasticity. *Computational Mechanics* 1996; **18**:245–258. DOI: 10.1007/BF00364140.
15. Steinmann P, Kuhl E, Stein E. Aspects of non-associated single crystal plasticity: influence of non-Schmid effects and localization analysis. *International Journal of Solids and Structures* 1998; **35**:4437–4456.
16. Knockaert R, Chastel Y, Massoni E. Rate-independent crystalline and polycrystalline plasticity, application to FCC materials. *International Journal of Plasticity* 2000; **16**:179–198. DOI: 10.1016/S0749-6419(99)00071-6.
17. Zuo QH. On the uniqueness of a rate-independent plasticity model for single crystals. *International Journal of Plasticity* 2011; **27**:1145–1164. DOI: 10.1016/j.ijplas.2010.12.002.
18. Pramono E, Willam K. Implicit integration of composite yield surfaces with corners. *Engineering Computations* 1989; **6**:186–197. DOI: 10.1108/eb023774.
19. (Available from: https://github.com/idaholab/moose) [Accessed on 19 March 2015].
20. Gaston D, Newman C, Hansen G, Lebrun-Grandie D. MOOSE: a parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design* 2009; **239**(10):1768–1778. DOI: 10.1016/j.nucengdes.2009.05.021.
21. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press: Cambridge, 2004. ISBN: 9780521833783.
22. Anand L, Kothari M. A computational procedure for rate-independent crystal plasticity. *Journal of the Mechanics and Physics of Solids* 1996; **44**:525–558. DOI: 10.1016/0022-5096(96)00001-4.
23. Press WH, Teukolsky A, Vetterling T, Flannery BP. *Numerical Recipes* (3rd edn), The Art of Scientific Computing. Cambridge University Press: Cambridge, November 2007. ISBN: 9780521880688.
24. Itasca Consulting Group Inc. *FLAC3D – Fast Lagrangian Analysis of Continua in Three-Dimensions, Ver. 5.0*. Itasca: Minneapolis, 2012.
25. Itasca Consulting Group Inc. *FLAC – Fast Lagrangian Analysis of Continua, Ver. 7.0*. Itasca: Minneapolis, 2011.
26. Foster CD, Regueiro RA, Fossum AF, Borja RI. Implicit numerical integration of a three-invariant isotropic/kinematic hardening cap plasticity model for geomaterial. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:5109–5138. DOI: 10.1016/j.cma.2005.01.001.
27. Rudnicki JW. Shear and compaction band formation on an elliptic yield cap. *Journal of Geophysical Research* 2004; **109**:B03402. DOI: 10.1029/2003JB002633.
28. Fellin W, Ostermann A. Consisten tangent operators for constitutive rate equations. *International Journal for Numerical and Analytical Methods in Geomechanics* 2002; **26**:1213–1233. DOI: 10.1002/nag.242.