INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
*Int. J. Numer. Meth. Engng* 2007; **69**:592–626
Published online 5 June 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1771

# An interior-point algorithm for elastoplasticity

K. Krabbenhoft[1,*,†], A. V. Lyamin[1], S. W. Sloan[1] and P. Wriggers[2]

[1]*Geotechnical Research Group, University of Newcastle, NSW 2308, Australia*
[2]*Institut fr Baumechanik und Numerische Mechanik, Universität Hannover, D-30167 Hannover, Germany*

SUMMARY

The problem of small-deformation, rate-independent elastoplasticity is treated using convex programming theory and algorithms. A finite-step variational formulation is first derived after which the relevant potential is discretized in space and subsequently viewed as the Lagrangian associated with a convex mathematical program. Next, an algorithm, based on the classical primal–dual interior point method, is developed. Several key modifications to the conventional implementation of this algorithm are made to fully exploit the nature of the common elastoplastic boundary value problem. The resulting method is compared to state-of-the-art elastoplastic procedures for which both similarities and differences are found. Finally, a number of examples are solved, demonstrating the capabilities of the algorithm when applied to standard perfect plasticity, hardening multisurface plasticity, and problems involving softening. Copyright © 2006 John Wiley & Sons, Ltd.

Received 19 October 2005; Revised 24 March 2006; Accepted 30 March 2006

KEY WORDS: plasticity; optimization; interior-point; finite elements

## 1. INTRODUCTION

The solution of elastoplastic boundary value problems typically proceeds by computing a displacement increment on the basis of the current state of the system. At a selected number of points, usually the Gauss points, the stress increments are then found by solving the local constitutive relations and an improved estimate of the elastoplastic stiffness modulus is computed. The possible deviation between the initially assumed stiffness and the actual one may yield a residual in the form of a global out-of-balance force which is subsequently sought minimized by a Newton type procedure that alternates between global equilibrium iterations and local constitutive updates. The main difference between the various methods lies in how the local stress update is performed and how the resulting stiffness is computed. Since the solution of the local constitutive relations is

*Correspondence to: K. Krabbenhoft, Geotechnical Research Group, University of Newcastle, NSW 2308, Australia.
†E-mail: kristian.krabbenhoft@newcastle.edu.au

equivalent to solving a set of non-linear ordinary differential equations, the common methods in use can be classified as being either explicit or implicit.

The implicit methods, which have received by far the most attention, usually employ the backward-Euler scheme, e.g. Reference [1], for the local integration whereas Newton's method is used for the global equilibrium iterations. As was first pointed out by Simo and Taylor [2], the use of the continuum elastoplastic modulus is not consistent with Newton's method where an exact linearization is implied. Rather, a slightly different tangent modulus which depends on the method of local integration, should be used. The use of this modulus leads to a quadratic rate of convergence as the solution is approached. That is, when all points in yielding have been identified, the resulting equations are solved using an exact linearization with all the ramifications that this has. Although the implicit methods were originally developed for metal plasticity, where they perform very well, procedures have also been derived for relatively complex soil, rock, concrete, and masonry models [3–12]. The robustness of these procedures is, however, much less documented than in the case of metal plasticity. An exception is the work of by Perez-Foguet *et al*. [13, 14] where it is shown that local substepping is crucial when integrating complex constitutive relations such as the ones mentioned above.

Alternatively, the local constitutive relations can be integrated explicitly. Here the substepping schemes developed by Sloan and his co-workers [15–20] have proved particularly efficient, not least in forming a reliable basis for formulating a general purpose solver that can be used for a wide variety of models. On the material level the elastoplastic constitutive relations are integrated explicitly to within a specified tolerance and on the structural level an inexact Newton method, possibly enhanced by an error-based selection of the load step [16, 20–22], is used.

A fundamentally different approach, which is the one explored in this paper, consists of casting the problem as a non-linear mathematical program which is subsequently solved using general optimization methods. This approach was first suggested by Maier and his co-workers [23–25] almost 40 years ago and, although often perceived as somewhat outdated, it is still occasionally used, see e.g. References [26–32]. So far, however, the main emphasis has been on formulating various standard and non-standard problems as mathematical programs rather than on devising solution procedures capable of solving large scale problems of practical interest. Thus, although the approach is elegant and has some obvious advantages over the more *ad hoc* methods, it lacks seriously behind in the crucial aspect of practical solution algorithms.

The main virtue of the mathematical programming approach is that it offers a coherent variational framework for treating a broad class of problems, one of these being classical rate-independent elastoplasticity. Thus, once a relevant variational principle has been established, it can be discretized by any method of choice and the resulting discrete potential then viewed as the Lagrangian associated with a mathematical program which is easily identified and, ideally, also easily solved. In this way, all possible displacement, equilibrium, mixed and hybrid finite element formulations can be treated in a unified manner, as can multi-surface plasticity, contact conditions and other external or internal inequality constraints. Moreover, variational formulations constitute a natural basis for error estimation. In perfect plasticity a particularly powerful feature is the possibility of computing rigorous upper and lower bounds on the exact collapse load [33–40].

Until about 15 years ago, the solution of general mathematical programs posed a significant challenge, with the simplex method and its derivatives being the only reliable algorithms available. Although relatively robust these algorithms suffer from a significant increase in computational effort as the problem size increases. Indeed, the number of iterations required is usually proportional



DOI: 10.1002/nme

to the number of constraints, i.e. the problem size, and the theoretical worst-case performance is even more discouraging.

In recent years, however, a number of new and very powerful methods of optimization have been developed and have now largely replaced earlier methods in theoretical and applied optimization. Thus, the paper of Karmarkar [41] in 1984 sparked a revolution which led to the development of the algorithms now known as interior-point methods [42–45]. The basic idea behind these methods is to solve the optimality conditions associated with a suitably penalized mathematical program, an approach which is radically different from the combinatorical approach of the simplex method. Furthermore, as is often the case with Newton based procedures, the number of iterations required to converge to the solution is largely independent of the problem size. Thus, the methods have been applied in many of the fields that traditionally make heavy use of optimization, i.e. economics, finance, planning and scheduling, control theory, etc. In computational mechanics, however, the application has been very limited and seems largely to have been confined to the problem of plastic limit analysis, e.g. References [46–50].

In this paper we explore the application of these new methods to boundary value problems arising in classical small-deformation rate-independent elastoplasticity. The resulting algorithm shares many traits with the method of Simo and Taylor [2]. For example, a quantity which is similar to the consistent tangent modulus derived in Reference [2] plays an important role in the algorithm. There are also some links to the fully implicit return mapping algorithm although a major difference between this and previous methods is that no explicit reference to a local integration phase is made. Regarding efficiency, the proposed method is also competitive with the Simo–Taylor method. The cost of each iteration is roughly the same and the iteration counts appear to be similar to, or perhaps slightly better than, those reported for the Simo–Taylor method. Furthermore, a quadratic rate of convergence is attained as the solution is approached. The robustness of the algorithm also appears to be excellent and can be further enhanced by a simple line search procedure. Finally, the algorithm extends naturally to multi-surface plasticity without any complications or additional computational effort.

The paper is organized as follows. In Section 2 a variational formulation of classical small-deformation rate-independent elastoplasticity is derived. We here make use of the complementary energy formulation of Simo *et al*. [51]. In Section 3 the corresponding finite-step mathematical programming formulation is derived and various finite element discretization issues discussed. A solution algorithm is then presented in Section 4 and a number of test examples are solved in Section 5. Finally, conclusions are drawn in Section 6.

## 2. VARIATIONAL FORMULATION OF FINITE-STEP ELASTOPLASTICITY

The current derivation of the finite-step complementary formulation of elastoplasticity follows that of Simo *et al*. [51]. A Helmholtz free energy function is first assumed and by a Legendre transformation the dual complementary energy function is constructed. Secondly, the plastic dissipation is stated and the principle of maximum dissipation postulated. As is well-known this principle implies associated flow and also produces the loading/unloading conditions.

### *2.1. Free energy*

We assume the existence of a Helmholtz free energy function of the type

$$\mathscr{A}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^{\mathrm{p}}, \boldsymbol{\alpha}) = \phi^{\mathrm{e}}(\boldsymbol{\varepsilon}^{\mathrm{e}}) + \phi^{\mathrm{p}}(\boldsymbol{\alpha}), \quad \boldsymbol{\varepsilon}^{\mathrm{e}} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{\mathrm{p}} \tag{1}$$



DOI: 10.1002/nme

where $\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^{\mathrm{e}}$, and $\boldsymbol{\varepsilon}^{\mathrm{p}}$ are the total, elastic, and plastic strains, respectively, and $\boldsymbol{\alpha}$ is a set of internal plastic variables. Next, consider the Legendre transformation

$$\phi^{\mathrm{p}}(\boldsymbol{\alpha}) = -\psi^{\mathrm{p}}(\boldsymbol{\kappa}) - \boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{\alpha} \tag{2}$$

where $\boldsymbol{\kappa}$ are the internal variables conjugate to $\boldsymbol{\alpha}$. We then have the relations

$$\boldsymbol{\kappa} = -\nabla\phi^{\mathrm{p}}(\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = -\nabla\psi^{\mathrm{p}}(\boldsymbol{\kappa}) \tag{3}$$

### 2.2. Plastic dissipation

The local plastic dissipation is given by [51]

$$\mathring{\mathscr{D}}(\boldsymbol{\sigma}, \boldsymbol{\kappa}; \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}, \dot{\boldsymbol{\alpha}}) = \boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} + \boldsymbol{\kappa}^{\mathsf{T}}\dot{\boldsymbol{\alpha}} \tag{4}$$

where $\boldsymbol{\sigma}$ are the stresses and a superposed circle (rather than a dot) is used to emphasize path-dependence. The principle of maximum plastic dissipation states that, for given $\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}$ and $\dot{\boldsymbol{\alpha}}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\kappa}$ will be such that the dissipation is at a maximum. This is subject to the constraint that $(\boldsymbol{\sigma}, \boldsymbol{\kappa})$ lie within some admissible domain given by the yield condition. Thus, the principle of maximum plastic dissipation can be stated as

$$\begin{aligned} &\underset{\boldsymbol{\sigma},\boldsymbol{\kappa}}{\text{maximize}} \quad \boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} + \boldsymbol{\kappa}^{\mathsf{T}}\dot{\boldsymbol{\alpha}} \\ &\text{subject to} \quad f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) \leqslant 0 \end{aligned} \tag{5}$$

If the yield function is convex, this is a standard convex program.

Following the interior-point methodology [42–44], problem (5) is solved by first adding a positively restricted variable, a so-called slack variable, to the inequality constraint, thus converting it to an equality. Secondly, a penalty term is added to the objective function which effectively avoids making explicit reference to the fact that the slack variable is positively restricted. The modified problem then reads

$$\begin{aligned} &\underset{\boldsymbol{\sigma},\boldsymbol{\kappa}}{\text{maximize}} \quad \boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} + \boldsymbol{\kappa}^{\mathsf{T}}\dot{\boldsymbol{\alpha}} + \mu \log s \\ &\text{subject to} \quad f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) + s = 0 \end{aligned} \tag{6}$$

where $s$ is a slack variable and $\mu \log s$, with $\mu$ being an arbitrarily small positive constant, is the so-called logarithmic barrier function. It should be noted here that numerous other barrier and penalty functions have been proposed (the standard reference being Reference [52]). In elasto-plasticity the viscoplastic regularization of Perzyna [53, 54] has been widely applied, see, e.g. Reference [55] and references therein. However, the logarithmic barrier function is the most satisfying from both theoretical and algorithmic points of view, see, e.g. Reference [56], and is a key ingredient in most modern large-scale optimization algorithms.

The Lagrangian associated with (6) is given by

$$\mathscr{L}_{\mathscr{D}} = \mathring{\mathscr{D}}(\boldsymbol{\sigma}, \boldsymbol{\kappa}; \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}, \dot{\boldsymbol{\alpha}}) + \mathring{\mathscr{R}}(\boldsymbol{\sigma}, \boldsymbol{\kappa}; \mathring{\gamma}, s) \tag{7}$$

where

$$\mathring{\mathscr{R}} = \mu \log s - \mathring{\gamma}[f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) + s] \tag{8}$$



DOI: 10.1002/nme

and $\mathring{\gamma}$ is a Lagrange multiplier. The first-order necessary and sufficient Karush–Kuhn–Tucker (KKT) optimality conditions associated with (6) follow from the Lagrangian (7) as

$$\begin{aligned}
\nabla_\sigma \mathscr{L}_\mathscr{D} &= \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} - \mathring{\gamma}\nabla_\sigma f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = \mathbf{0} \\
\nabla_\kappa \mathscr{L}_\mathscr{D} &= \dot{\boldsymbol{\alpha}} - \mathring{\gamma}\nabla_\kappa f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = \mathbf{0} \\
\nabla_{\mathring{\gamma}} \mathscr{L}_\mathscr{D} &= -f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) - s = 0 \\
\nabla_s \mathscr{L}_\mathscr{D} &= \mu s^{-1} - \mathring{\gamma} = 0 \;\Rightarrow\; s\mathring{\gamma} = \mu
\end{aligned} \tag{9}$$

The first optimality condition is here the associated flow rule, where it is clear that the Lagrange multiplier $\mathring{\gamma}$ is the plastic multiplier. The second condition establishes the evolution of the internal variables. By the Legendre transformation (2)–(3) we have

$$\boldsymbol{\alpha} = -\nabla\psi^{\mathrm{p}}(\boldsymbol{\kappa}) \;\Rightarrow\; \dot{\boldsymbol{\alpha}} = -\nabla^2\psi^{\mathrm{p}}(\boldsymbol{\kappa})\dot{\boldsymbol{\kappa}} \tag{10}$$

so that the evolution of $\boldsymbol{\kappa}$ is given by

$$\dot{\boldsymbol{\kappa}} = -\mathring{\gamma}[\nabla^2\psi^{\mathrm{p}}(\boldsymbol{\kappa})]^{-1}\nabla_\kappa f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) \tag{11}$$

Since $\psi^{\mathrm{p}}$ is a function of $\boldsymbol{\kappa}$ only, there is a natural limitation to the type of hardening laws that can be considered within this framework.

The third optimality condition restates the yield condition and finally, the last condition is the plastic consistency condition. It should be noted here that since $s>0$ by virtue of the logarithmic barrier function, we also have $\mathring{\gamma}>0$ since $\mu>0$. These conditions, together with the two last optimality conditions, define the loading/unloading conditions, which are usually written as

$$\mathring{\gamma} \geqslant 0, \quad f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) \leqslant 0, \quad \mathring{\gamma} f(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = 0 \tag{12}$$

Note, however, that since $\mu>0$ we can only satisfy strict inequality for first two equations of (12) whereas the last condition is fulfilled to within some arbitrarily small positive constant proportional to $\mu$.

Finally, it should be mentioned that the extension to multi-surface plasticity is straightforward. In this case the penalized form of the principle of maximum dissipation is

$$\begin{aligned}
&\underset{\boldsymbol{\sigma},\boldsymbol{\kappa}}{\text{maximize}} \quad \boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} + \boldsymbol{\kappa}^{\mathsf{T}}\dot{\boldsymbol{\alpha}} + \mu\sum_{j=1}^{\mathrm{NS}} \log s_j \\
&\text{subject to} \quad f_j(\boldsymbol{\sigma}, \boldsymbol{\kappa}) + s_j = 0, \quad j = 1, \ldots, \mathrm{NS}
\end{aligned} \tag{13}$$

where NS is the number of yield surfaces. The associated Lagrangian is

$$\mathscr{L}_\mathscr{D} = \boldsymbol{\sigma}^{\mathsf{T}}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} + \boldsymbol{\kappa}^{\mathsf{T}}\dot{\boldsymbol{\alpha}} - \sum_{j=1}^{\mathrm{NS}} [\mathring{\gamma}_j(f_j(\boldsymbol{\sigma}, \boldsymbol{\kappa}) + s_j) + \mu \log s_j] \tag{14}$$



DOI: 10.1002/nme

and the optimality conditions follow as

$$\begin{aligned}
\nabla_\sigma \mathscr{L}_{\mathscr{D}} &= \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} - \sum_{j=1}^{\mathrm{NS}} \mathring{\gamma}_j \nabla_\sigma f_j(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = \mathbf{0} \\
\nabla_\kappa \mathscr{L}_{\mathscr{D}} &= \dot{\boldsymbol{\alpha}} - \sum_{j=1}^{\mathrm{NS}} \mathring{\gamma}_j \nabla_\kappa f_j(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = \mathbf{0} \\
\nabla_{\mathring{\gamma}_j} \mathscr{L}_{\mathscr{D}} &= -f_j(\boldsymbol{\sigma}, \boldsymbol{\kappa}) - s_j = 0, \quad j = 1, \ldots, \mathrm{NS} \\
\nabla_{s_j} \mathscr{L}_{\mathscr{D}} &= \mu s_j^{-1} - \mathring{\gamma}_j = 0 \;\Rightarrow\; s_j \gamma_j = \mu, \quad j = 1, \ldots, \mathrm{NS}
\end{aligned} \tag{15}$$

The first of these is Koiter's rule which, as seen, is a direct consequence of the principle of maximum plastic dissipation (due to von Mises). This and the rest of the conditions are analogous to those in the single-surface case.

### 2.3. Complementary energy

A complementary stored energy function, where $\boldsymbol{\sigma}$ and $\boldsymbol{\kappa}$ replace $\boldsymbol{\varepsilon}$ and $\boldsymbol{\alpha}$ as the independent variables, is achieved through the following Legendre transformation:

$$\phi^{\mathrm{e}}(\boldsymbol{\varepsilon}^{\mathrm{e}}) = -\psi^{\mathrm{e}}(\boldsymbol{\sigma}) + \boldsymbol{\sigma}^{\mathsf{T}} \boldsymbol{\varepsilon}^{\mathrm{e}} \tag{16}$$

where $\psi^{\mathrm{e}}(\boldsymbol{\sigma})$ is the complementary stored energy function. We then have the relations

$$\boldsymbol{\sigma} = \nabla \phi^{\mathrm{e}}(\boldsymbol{\varepsilon}^{\mathrm{e}}), \quad \boldsymbol{\varepsilon}^{\mathrm{e}} = \nabla \psi^{\mathrm{e}}(\boldsymbol{\sigma}) \tag{17}$$

and the total free energy is given by

$$\mathscr{A}(\boldsymbol{\sigma}, \boldsymbol{\kappa}) = -\psi^{\mathrm{e}}(\boldsymbol{\sigma}) - \psi^{\mathrm{p}}(\boldsymbol{\kappa}) + \boldsymbol{\sigma}^{\mathsf{T}} \boldsymbol{\varepsilon}^{\mathrm{e}} - \boldsymbol{\kappa}^{\mathsf{T}} \boldsymbol{\alpha} \tag{18}$$

where $\boldsymbol{\varepsilon}^{\mathrm{e}}$ and $\boldsymbol{\alpha}$ are given in terms of $\boldsymbol{\sigma}$ and $\boldsymbol{\kappa}$ by (17) and (3), respectively.

### 2.4. Power of deformation

The power of deformation over an infinitesimal increment is given by [57]

$$\mathring{\mathscr{P}} = \boldsymbol{\sigma}^{\mathsf{T}} \dot{\boldsymbol{\varepsilon}} = \dot{\mathscr{A}} + \mathring{\mathscr{D}} \tag{19}$$

That is, the power of deformation is equal to the sum of the rate of change in free energy $\mathscr{A}$ and the dissipation $\mathring{\mathscr{D}}$ over an infinitesimal increment.

### 2.5. Finite-step formulation

The power of deformation (19) is extended to steps of a finite magnitude as

$$\mathring{\mathscr{P}}^{n}_{n+1} \simeq \mathscr{A}_{n+1} - \mathscr{A}_n + \mathring{\mathscr{D}}^{n}_{n+1} \tag{20}$$

where the notation $\mathring{X}^{n}_{n+1}$ is to be understood as the change in a path dependent quantity $X$ over the interval from $n$ to $n+1$.



DOI: 10.1002/nme

The free energies at two points $n$ and $n+1$ are given by

$$\mathscr{A}_n = -\psi^{\mathrm{e}}(\boldsymbol{\sigma}_n) - \psi^{\mathrm{p}}(\boldsymbol{\kappa}_n) + \boldsymbol{\sigma}_n^{\mathsf{T}}(\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_n^{\mathrm{p}}) - \boldsymbol{\kappa}_n^{\mathsf{T}}\boldsymbol{\alpha}_n \tag{21}$$

$$\mathscr{A}_{n+1} = -\psi^{\mathrm{e}}(\boldsymbol{\sigma}_{n+1}) - \psi^{\mathrm{p}}(\boldsymbol{\kappa}_{n+1}) + \boldsymbol{\sigma}_{n+1}^{\mathsf{T}}(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{\mathrm{p}}) - \boldsymbol{\kappa}_{n+1}^{\mathsf{T}}\boldsymbol{\alpha}_{n+1} \tag{22}$$

The plastic dissipation over a finite increment from $n$ to $n+1$ is evaluated in a backward Euler manner to give

$$\mathring{\mathscr{D}}_{n+1}^{n} \simeq \boldsymbol{\sigma}_{n+1}^{\mathsf{T}}(\boldsymbol{\varepsilon}_{n+1}^{\mathrm{p}} - \boldsymbol{\varepsilon}_n^{\mathrm{p}}) + \boldsymbol{\kappa}_{n+1}^{\mathsf{T}}(\boldsymbol{\alpha}_{n+1} - \boldsymbol{\alpha}_n) \tag{23}$$

Finally, the penalty term (8) is evaluated as

$$\mathring{\mathscr{R}}_{n+1}^{n} \simeq \mu \log s_{n+1} - \mathring{\gamma}_{n+1}^{n}[f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) + s_{n+1}] \tag{24}$$

Using this discretization we define a penalized power of deformation function as

$$\widehat{\mathring{\mathscr{P}}}_{n+1}^{n} = \mathscr{A}_{n+1} - \mathscr{A}_n + \mathring{\mathscr{D}}_{n+1}^{n} + \mathring{\mathscr{R}}_{n+1}^{n} \tag{25}$$

After collecting terms in the above expressions the total penalized power of deformation over a finite step can be written as

$$\begin{aligned}\widehat{\mathring{\mathscr{P}}}_{n+1}^{n} = &-(\psi_{n+1}^{\mathrm{e}} - \psi_n^{\mathrm{e}}) - (\psi_{n+1}^{\mathrm{p}} - \psi_n^{\mathrm{p}}) + (\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n)^{\mathsf{T}}\nabla\psi_n^{\mathrm{e}} + (\boldsymbol{\kappa}_{n+1} - \boldsymbol{\kappa}_n)^{\mathsf{T}}\nabla\psi_n^{\mathrm{p}} \\ &+ \boldsymbol{\sigma}_{n+1}^{\mathsf{T}}\boldsymbol{\nabla}(\mathbf{u}_{n+1} - \mathbf{u}_n) + \mu \log s_{n+1} - \mathring{\gamma}_{n+1}^{n}(f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) + s_{n+1})\end{aligned} \tag{26}$$

where the small-displacement assumption

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}\mathbf{u} \tag{27}$$

has been introduced.

At this point it is convenient to further introduce the following quantities:

$$\hat{\boldsymbol{\sigma}} = [\boldsymbol{\sigma}, \boldsymbol{\kappa}]^{\mathsf{T}}, \quad \hat{\boldsymbol{\nabla}}^{\mathsf{T}} = [\boldsymbol{\nabla}^{\mathsf{T}}, \mathbf{0}], \quad \psi^{\mathrm{ep}}(\hat{\boldsymbol{\sigma}}) = \psi^{\mathrm{e}}(\boldsymbol{\sigma}) + \psi^{\mathrm{p}}(\boldsymbol{\kappa}) \tag{28}$$

so that (26) can be written as

$$\begin{aligned}\widehat{\mathring{\mathscr{P}}}_{n+1}^{n} = &-(\psi_{n+1}^{\mathrm{ep}} - \psi_n^{\mathrm{ep}}) + (\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\nabla\psi_n^{\mathrm{ep}} + \hat{\boldsymbol{\sigma}}_{n+1}^{\mathsf{T}}\hat{\boldsymbol{\nabla}}(\mathbf{u}_{n+1} - \mathbf{u}_n) \\ &+ \mu \log s_{n+1} - \lambda_{n+1}[f(\hat{\boldsymbol{\sigma}}_{n+1}) + s_{n+1}]\end{aligned} \tag{29}$$

where $\lambda_{n+1} = \mathring{\gamma}_{n+1}^{n}$.

### *2.6. Variational principle*

A Hellinger–Reissner type functional is now defined as

$$\mathring{\Pi}_{n+1}^{n} = \int_{\Omega} \widehat{\mathring{\mathscr{P}}}_{n+1}^{n}\, \mathrm{d}\Omega - \int_{\Omega} \mathbf{b}^{\mathsf{T}}(\mathbf{u}_{n+1} - \mathbf{u}_n)\, \mathrm{d}\Omega - \int_{\Gamma_\sigma} \mathbf{t}^{\mathsf{T}}(\mathbf{u}_{n+1} - \mathbf{u}_n)\, \mathrm{d}\Gamma \tag{30}$$

where $\mathbf{b}$ are the body forces acting in the domain $\Omega$ and $\mathbf{t}$ are the tractions acting on the boundary $\Gamma_\sigma$.



DOI: 10.1002/nme

*2.6.1. Euler–Lagrange equations.* The functional (30) verifies both the weak and strong forms of the governing equations. In the following we summarize only the strong form. Taking functional derivatives, see, e.g. Reference [55], the following Euler–Lagrange equations appear

$$\begin{aligned}
&\frac{\delta \mathring{\Pi}^n_{n+1}}{\delta \mathbf{u}_{n+1}} = \begin{cases} \boldsymbol{\nabla}^{\mathsf{T}}\boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} & \text{in } \Omega \\ \mathbf{P}^{\mathsf{T}}\boldsymbol{\sigma} - \mathbf{t} = \mathbf{0} & \text{on } \Gamma_\sigma \end{cases} \\
&\frac{\delta \mathring{\Pi}^n_{n+1}}{\delta \boldsymbol{\sigma}_{n+1}} = -\nabla\psi^{\mathrm{e}}(\boldsymbol{\sigma}_{n+1}) + \nabla\psi^{\mathrm{e}}(\boldsymbol{\sigma}_n) + \boldsymbol{\nabla}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \lambda_{n+1}\nabla_\sigma f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) = \mathbf{0} \\
&\frac{\delta \mathring{\Pi}^n_{n+1}}{\delta \boldsymbol{\kappa}_{n+1}} = -\nabla\psi^{\mathrm{p}}(\boldsymbol{\kappa}_{n+1}) + \nabla\psi^{\mathrm{p}}(\boldsymbol{\kappa}_n) - \lambda_{n+1}\nabla_\kappa f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) = \mathbf{0} \\
&\frac{\delta \mathring{\Pi}^n_{n+1}}{\delta \lambda_{n+1}} = -f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) - s_{n+1} = 0 \\
&\frac{\delta \mathring{\Pi}^n_{n+1}}{\delta s_{n+1}} = \mu s^{-1}_{n+1} - \lambda_{n+1} = 0 \;\Rightarrow\; s_{n+1}\lambda_{n+1} = \mu
\end{aligned} \tag{31}$$

where the stress boundary conditions are retrieved by integration by parts and the displacement boundary conditions are assumed to hold *a priori*.

## 3. MATHEMATICAL PROGRAMMING FORMULATION

In the following the mathematical programming formulation of the continuous variational problem is derived. For convenience this will be limited to the case where $\psi^{\mathrm{ep}}(\hat{\boldsymbol{\sigma}})$ is a quadratic function given by

$$\psi^{\mathrm{ep}}(\hat{\boldsymbol{\sigma}}) = \tfrac{1}{2}\hat{\boldsymbol{\sigma}}^{\mathsf{T}}\mathbb{M}\hat{\boldsymbol{\sigma}}, \quad \mathbb{M} = \begin{bmatrix} \mathbb{E}^{-1} & \\ & \mathbb{H}^{-1} \end{bmatrix} \tag{32}$$

where $\mathbb{E}$ is the elastic stiffness modulus and $\mathbb{H}$ is a matrix of hardening moduli. The corresponding finite-step power of deformation is given by

$$\hat{\mathring{\mathscr{P}}}^n_{n+1} = -\tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbb{M}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n) + \hat{\boldsymbol{\sigma}}^{\mathsf{T}}_{n+1}\hat{\boldsymbol{\nabla}}(\mathbf{u}_{n+1} - \mathbf{u}_n) + \mathring{\mathscr{R}}^n_{n+1} \tag{33}$$

where the penalty term $\mathring{\mathscr{R}}^n_{n+1}$ is

$$\mathring{\mathscr{R}}^n_{n+1} = \mu \log s_{n+1} - \lambda_{n+1}[f(\hat{\boldsymbol{\sigma}}_{n+1}) + s_{n+1}] \tag{34}$$

A mixture of isotropic and kinematic hardening can be modelled by introducing

$$\boldsymbol{\kappa} = [\boldsymbol{\kappa}_i, \boldsymbol{\kappa}_{\mathrm{k}}]^{\mathsf{T}} \tag{35}$$



DOI: 10.1002/nme

in conjunction with a yield function given by

$$F(\hat{\boldsymbol{\sigma}}) = f(\boldsymbol{\sigma} - \boldsymbol{\kappa}_k) - h(\boldsymbol{\kappa}_i) \tag{36}$$

which should be convex with respect to all variables, i.e. $\boldsymbol{\sigma}$ as well as $\boldsymbol{\kappa}_k$ and $\boldsymbol{\kappa}_i$.

### *3.1. Spatial discretization*

We introduce the following finite element approximations for the generalized stresses and displacements

$$\hat{\boldsymbol{\sigma}}(\mathbf{x}) \approx \mathbf{N}_{\hat{\sigma}}(\mathbf{x})\hat{\boldsymbol{\sigma}}^h, \quad \mathbf{u}(\mathbf{x}) \approx \mathbf{N}_u(\mathbf{x})\mathbf{u}^h \tag{37}$$

where $\hat{\boldsymbol{\sigma}}^h$, and $\mathbf{u}^h$ are the nodal generalized stresses and displacements, respectively. In order to avoid an excessive number of sub- and superscripts we make the following change of notation:

$$\hat{\boldsymbol{\sigma}} := \hat{\boldsymbol{\sigma}}^h, \quad \mathbf{u} := \mathbf{u}^h \tag{38}$$

so that $\hat{\boldsymbol{\sigma}}$ and $\mathbf{u}$ now refer to nodal, rather than continuous, quantities. Inserting the above approximations into (30) gives the following fully discrete functional:

$$\begin{aligned} \mathring{\Pi}^n_{n+1} = &-\frac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n) + \hat{\boldsymbol{\sigma}}^{\mathsf{T}}_{n+1}\hat{\mathbf{B}}(\mathbf{u}_{n+1} - \mathbf{u}_n) \\ &- \mathbf{p}^{\mathsf{T}}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \int_\Omega \lambda_{n+1}[f(\hat{\boldsymbol{\sigma}}_{n+1}) + s_{n+1}] - \mu \log s_{n+1}\, \mathrm{d}\Omega \end{aligned} \tag{39}$$

where

$$\mathbf{M} = \int_\Omega \mathbf{N}^{\mathsf{T}}_{\hat{\sigma}}\mathbb{M}\mathbf{N}_{\hat{\sigma}}\, \mathrm{d}\Omega, \quad \hat{\mathbf{B}} = \int_\Omega \mathbf{N}^{\mathsf{T}}_{\hat{\sigma}}\hat{\nabla}\mathbf{N}_u\, \mathrm{d}\Omega, \quad \mathbf{p} = \int_\Omega \mathbf{N}^{\mathsf{T}}_u\mathbf{b}\, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \mathbf{N}^{\mathsf{T}}_u\mathbf{t}\, \mathrm{d}\Gamma \tag{40}$$

In the following we will only consider discretizations where (i) the displacements are continuous and differentiable inside the elements and continuous between elements and (ii) the generalized stresses are continuous and differentiable inside the elements and discontinuous between elements. Obviously, standard displacement finite elements fall within this scope.

### *3.2. Numerical integration*

Following standard practice the integrals in (40) are evaluated using a quadrature rule. Extending this procedure to the last integral in (39) implies that the yield condition is enforced at a finite number of points, i.e.

$$\begin{aligned} \mathring{\mathscr{R}}^n_{n+1} &= \int_\Omega \lambda_{n+1}[f(\hat{\boldsymbol{\sigma}}_{n+1}) + s_{n+1}] + \mu \log s_{n+1}\, \mathrm{d}\Omega \\ &\approx \sum_{I=1}^{N_q} W^I[\lambda^I_{n+1}[f(\mathbf{N}_{\hat{\sigma}}(\mathbf{x}^I)\hat{\boldsymbol{\sigma}}_{n+1}) + s^I_{n+1}] + \mu \log s^I_{n+1}] \end{aligned} \tag{41}$$

where superscript $I$ refers to the $N_q$ quadrature points and $W^I$ are the usual weighting factors. The location of the points at which the yield conditions are enforced can be chosen



DOI: 10.1002/nme

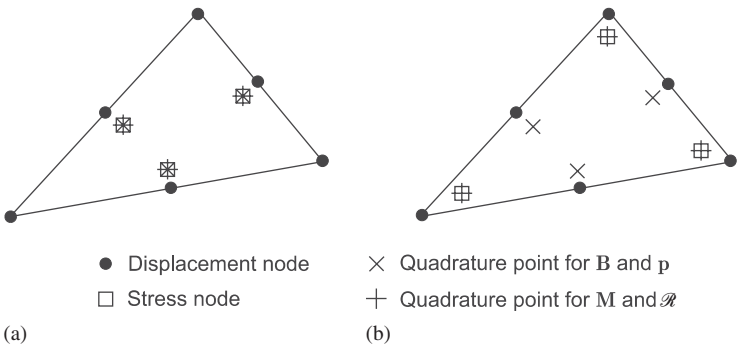


Figure 1. Standard displacement and mixed stress–displacement elements: (a) displacement element; and (b) mixed stress-displacement element.

from a number of obvious candidates which, on the other hand, are linked to the numerical integration scheme used for computing the system matrices (40). As an example, consider a linear stress/quadratic displacement triangle. Suppose that the stresses are interpolated from the midside nodes and the displacements from the corner and midside nodes, see Figure 1(a). A choice of points at which the yield conditions should be enforced must then be made. If all vectors and matrices are integrated using standard three-point quadrature, where the integration points are located at the midside nodes, and the yield conditions are enforced at these nodes, the standard displacement formulation appears.

There are, however, a number of obvious alternatives to this choice of interpolation, integration and yield enforcement points. Firstly, as will become clear later on, it is desirable that $\mathbf{M}$ be a block-diagonal matrix. This implies that $\mathbf{M}$ should always be evaluated by a scheme where the quadrature points coincide with the stress interpolation points whereas $\hat{\mathbf{B}}$ can be evaluated at any point. In this way it is possible to derive an element where the stresses, instead of being interpolated from the midside nodes, are interpolated from the corner nodes and $\mathbf{M}$ evaluated at these points with equal weights, Figure 1(b). For such an element it is natural to impose the yield condition at corner nodes since, for a convex yield function, this implies satisfaction of $f(\hat{\boldsymbol{\sigma}}) \leqslant 0$ throughout the element. The discrete strain–displacement operator $\hat{\mathbf{B}}$, on the other hand, is evaluated using the standard three-point scheme with the integration points located at the midside nodes. For the case of perfect plasticity, the incremental stress–strain relations resulting from these choices are given by

$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{\sigma}_{n+1}} = \mathbf{0} \;\Rightarrow\; \sum_{m=1}^{3} W^m \underbrace{[\mathbf{N}_\sigma^{\mathrm{T}}(\mathbf{x}^m)\nabla \mathbf{N}_u(\mathbf{x}^m)](\mathbf{u}_{n+1} - \mathbf{u}_n)}_{(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n)^m}$$

$$= \sum_{c=1}^{3} W^c \underbrace{[\mathbf{N}_\sigma^{\mathrm{T}}(\mathbf{x}^c)\mathbb{E}^{-1}\mathbf{N}_\sigma(\mathbf{x}^c)](\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n)}_{(\boldsymbol{\varepsilon}^{\mathrm{e}}_{n+1} - \boldsymbol{\varepsilon}^{\mathrm{e}}_n)^c} + \sum_{c=1}^{3} W^c \underbrace{\lambda^c_{n+1} \nabla_\sigma f(\boldsymbol{\sigma}^c_{n+1})}_{(\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} - \boldsymbol{\varepsilon}^{\mathrm{p}}_n)^c} \tag{42}$$

where superscript $m$ refers to the midside nodes and $c$ to the corner nodes. Thus, the stress–strain relations are enforced in an approximate or 'mixed' way. The total strain increment refers



DOI: 10.1002/nme

to the midside nodes whereas the elastic and plastic strain increments refer to the corner nodes. A rigid-plastic version of this element has been used extensively by Borges, Zouain *et al.* for plastic limit and shakedown analysis [58–62] where it appears to furnish excellent results. Zienkiewicz and Taylor [63] further note its satisfactory performance for linear elasticity and thus, it can be expected to perform well also for elastoplasticity. Concerning the ultimate limit state we note that since it is more restrictive to enforce the yield conditions at the corner nodes than at the midside nodes, it is immediately clear that the element will always produce collapse loads that are less than or equal to those obtained by the standard displacement formulation.

### *3.3. Mathematical program*

With the above approximation of the penalty term, the discrete functional (39) can be seen as the Lagrangian associated with the following mathematical program:

$$\begin{aligned} \min_{\mathbf{u}_{n+1}} \max_{\hat{\boldsymbol{\sigma}}_{n+1}} : \quad & -\tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)+\hat{\boldsymbol{\sigma}}_{n+1}\hat{\mathbf{B}}(\mathbf{u}_{n+1}-\mathbf{u}_n) \\ & -\mathbf{p}^{\mathsf{T}}(\mathbf{u}_{n+1}-\mathbf{u}_n)+\mu\sum_{J=1}^{N}\log s^J \\ \text{subject to:} \quad & f_J(\hat{\boldsymbol{\sigma}}^J_{n+1})+s^J_{n+1}=0, \quad J=1,\ldots,N_\sigma \end{aligned} \tag{43}$$

To reduce this problem the generalized stresses are first considered known. This gives the following optimality condition for the minimization part of the problem

$$\hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}^*_{n+1}-\mathbf{p}=\mathbf{0} \tag{44}$$

which are simply the discrete equations of equilibrium. Next, the displacements are considered known. In doing so we observe that the above condition gives

$$(\mathbf{u}_{n+1}-\mathbf{u}_n)^{\mathsf{T}}(\hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}^*_{n+1}-\mathbf{p})=\mathbf{0} \tag{45}$$

for arbitrary $(\mathbf{u}_{n+1}-\mathbf{u}_n)$. Thus, the min–max problem (43) reduces to

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)+\sum_{J=1}^{N}\mu\log s^J \\ \text{subject to} \quad & \hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_{n+1}=\mathbf{p} \\ & f_J(\hat{\boldsymbol{\sigma}}^J_{n+1})+s^J_{n+1}=0, \quad J=1,\ldots,N_\sigma \end{aligned} \tag{46}$$

If $\mathbf{M}$ is positive semidefinite and $f_J(\hat{\boldsymbol{\sigma}}^J_{n+1})$ are convex, this is a standard convex optimization problem whose solution is discussed in detail in Section 5.

The geometric interpretation of (46) is very clear: from a known point lying in the plane given by $h_n : \hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_n-\mathbf{p}_n=0$, the new point is to be found by minimizing the distance (in the metric $\mathbf{M}$) to a parallel plane given by $h_{n+1} : \hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_{n+1}-\mathbf{p}_{n+1}=\mathbf{0}$ so that the new point satisfies all $f_J(\hat{\boldsymbol{\sigma}}_{n+1})\leqslant 0$. This geometric interpretation is illustrated in Figure 2 for the case of perfect plasticity.



DOI: 10.1002/nme

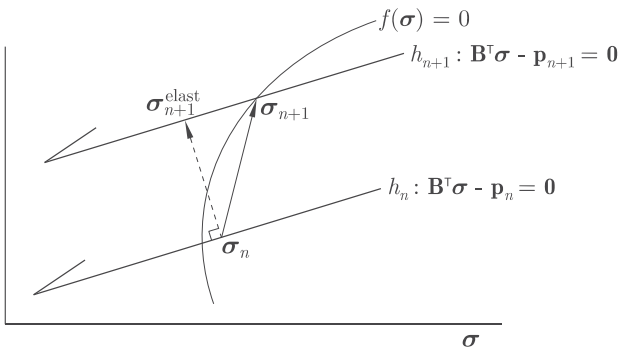


Figure 2. Geometric interpretation linear elasticity/perfect plasticity in stress space.

### *3.4. Optimality conditions*

The Lagrangian associated with (46) is given by

$$\begin{aligned}\mathscr{L} = &-\frac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n) + \mathbf{v}_{n+1}(\hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_{n+1} - \mathbf{p}) \\ &- \sum_{J=1}^{N_\sigma}[\lambda_{n+1}^J[f_J(\hat{\boldsymbol{\sigma}}_{n+1}^J, \boldsymbol{\kappa}_{n+1}^J) + s_{n+1}^J] - \mu \log s^J]\end{aligned} \tag{47}$$

where $\mathbf{v}_{n+1}$ and $\lambda_{n+1}^J$ are Lagrange multipliers. The KKT conditions follow as

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \hat{\boldsymbol{\sigma}}_{n+1}} &= \mathbf{r}_{\text{ss}} = -\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n) - \nabla_{\hat{\sigma}}\mathbf{f}_{n+1}\boldsymbol{\lambda}_{n+1} + \hat{\mathbf{B}}(\mathbf{u}_{n+1} - \mathbf{u}_n) = \mathbf{0} \\ \frac{\partial \mathscr{L}}{\partial \hat{\mathbf{u}}_{n+1}} &= \mathbf{r}_{\text{eq}} = \hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_{n+1} - \mathbf{p} = \mathbf{0} \\ \frac{\partial \mathscr{L}}{\partial \boldsymbol{\lambda}_{n+1}} &= \mathbf{r}_{\text{yc}} = -\mathbf{f}_{n+1} - \mathbf{s}_{n+1} = \mathbf{0} \\ \frac{\partial \mathscr{L}}{\partial \mathbf{s}_{n+1}} &= \mathbf{r}_{\text{cs}} = \mu\mathbf{S}_{n+1}^{-1}\mathbf{e} - \boldsymbol{\lambda}_{n+1} = \mathbf{0} \;\Rightarrow\; \mathbf{S}_{n+1}\boldsymbol{\lambda}_{n+1} = \mu\mathbf{e}\end{aligned} \tag{48}$$

where the following matrix and vectors have been introduced:

$$\boldsymbol{\lambda}_{n+1} = [\lambda_{n+1}^1, \ldots, \lambda_{n+1}^{N_\sigma}]^{\mathsf{T}}, \quad \mathbf{f}_{n+1} = [f_1(\hat{\boldsymbol{\sigma}}_{n+1}^1), \ldots, f_{N_\sigma}(\hat{\boldsymbol{\sigma}}_{n+1}^{N_\sigma})]^{\mathsf{T}} \tag{49}$$

$$\mathbf{s}_{n+1} = [s_{n+1}^1, \ldots, s_{n+1}^{N_\sigma}]^{\mathsf{T}}, \quad \mathbf{S}_{n+1} = \text{diag}(\mathbf{s}_{n+1}) \tag{50}$$

and $\mathbf{e}$ is a vector of ones. Finally, the Lagrange multipliers $\mathbf{v}_{n+1}$ have been identified as the displacement increment $\mathbf{u}_{n+1} - \mathbf{u}_n$. The task ahead is to solve these optimality conditions.



DOI: 10.1002/nme

### *3.5. Duality*

As an alternative to solving the minimization part of the min–max problem (46) first, we can begin by solving the maximization problem. The following minimization problem then appears:

$$\begin{aligned}
\underset{(\mathbf{u},\boldsymbol{\lambda})_{n+1}}{\text{minimize}} \quad & \underbrace{(\hat{\boldsymbol{\sigma}}_{n+1})^{\mathsf{T}}\nabla_{\hat{\boldsymbol{\sigma}}}\mathbf{f}_{n+1}\boldsymbol{\lambda}_{n+1}}_{\int_\Omega \mathring{\mathscr{D}}^{n}_{n+1}\,\mathrm{d}\Omega} + \underbrace{\tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}+\hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)}_{\int_\Omega \mathscr{A}_{n+1}-\mathscr{A}_n\,\mathrm{d}\Omega} - \underbrace{\mathbf{p}^{\mathsf{T}}(\mathbf{u}_{n+1}-\mathbf{u}_n)}_{(\mathring{\mathscr{W}}^{\mathrm{ext}})^{n}_{n+1}} \\
\text{subject to} \quad & \hat{\mathbf{B}}(\mathbf{u}_{n+1}-\mathbf{u}_n) = \mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n) + \nabla_{\hat{\boldsymbol{\sigma}}}\mathbf{f}_{n+1}\boldsymbol{\lambda}_{n+1} \\
& \mathbf{f}(\hat{\boldsymbol{\sigma}}_{n+1}) \leqslant \mathbf{0}, \quad \boldsymbol{\lambda}_{n+1} \geqslant \mathbf{0}
\end{aligned} \tag{51}$$

It is here clear that the objective function is the difference between the total power of deformation and the rate of external work. Furthermore, it is easy to verify that the optimality conditions (48) are also the KKT conditions for the above problem. Using these optimality conditions it can be shown that the objective function in (46) is equal to the one in (51), i.e. when all optimality conditions are fulfilled we have

$$\begin{aligned}
\mathring{\Pi}^{n}_{n+1} &= \hat{\boldsymbol{\sigma}}^{\mathsf{T}}_{n+1}\nabla_{\hat{\boldsymbol{\sigma}}}\mathbf{f}_{n+1}\boldsymbol{\lambda}_{n+1} + \tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}+\hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n) - \mathbf{p}^{\mathsf{T}}(\mathbf{u}_{n+1}-\mathbf{u}_n) \\
&= -\tfrac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}-\hat{\boldsymbol{\sigma}}_n)
\end{aligned} \tag{52}$$

The programs (46) and (51) can therefore be seen as being dual to each other. In practice this feature can be exploited so that, in addition to the residual of the optimality conditions, the gap between the primal and dual objective functions of (46) and (51) can also be used as a measure of the progress of the solution algorithm.

## 4. SOLUTION ALGORITHM

Although the idea of applying standard mathematical programming algorithms to the solution of incremental elastoplastic problems is not new, the approach has never been computationally competitive with the more traditional methods described in Section 1. However, as also mentioned in Section 1, in recent years the field of applied optimization has undergone a profound change with the so-called interior point method replacing earlier algorithms in many, if not most, practical applications. This development was initiated in 1984 by the paper of Karmarkar [41] on a linear programming algorithm with a proven polynomial complexity (a significant result since the complexity of the simplex algorithm, the most widely used algorithm until then, is exponential). It was soon realized [64], however, that the method of Karmarkar could be derived from the barrier methods developed by Fiacco and McCormick [52] in the 1960s. Another important milestone is the so-called primal–dual interior-point method of Lustig [65] which has become the prototype of many modern convex programming algorithms. The primal–dual interior-point method, which will form the basis of the algorithm developed in the following, aims to solve the optimality conditions resulting from a problem where the objective function has been augmented by a logarithmic barrier term as in (48). This is done using Newton's method with a reduced step length in order to ensure that all positively restricted variables remain positive throughout the iterations, i.e. that all points $\mathbf{s}_{n+1}$ and $\boldsymbol{\lambda}_{n+1}$ remain in the *interior* of the feasible solution space. Furthermore, as the solution is approached the barrier parameter is reduced according to a certain rule usually relating the current value of the parameter to the 'duality gap' $\delta_{n+1} = \mathbf{s}^{\mathsf{T}}_{n+1}\boldsymbol{\lambda}_{n+1}$.



DOI: 10.1002/nme

Besides the problems with ensuring that $\mathbf{s}_{n+1}$ and $\boldsymbol{\lambda}_{n+1}$ remain strictly feasible throughout the iterations, the main difficulty involved with applying Newton's method to sets of optimality conditions resulting from general mathematical programs is that usually very little knowledge about the optimal solution is available. Thus, choosing an initial point within the convergence radius of Newton's method is generally impossible. A very important point in the practical implementation of the primal–dual interior-point method is therefore a rule for how the barrier parameter should be decreased in the course of the iterations. That is, if it is reduced too much the iterations will diverge and if it is not reduced enough the iteration count may be unreasonably high. In most cases, with a good barrier parameter reduction rule, the number of iterations required is around 20–50 independent of the size of the problem. In typical elastoplastic computations, however, the situation is quite different. Here, a good estimate of the optimal solution is usually available, namely the last converged solution. Of course, if large changes occur over a given load step this initial point may not be particularly good, but the load step can then be reduced. It seems therefore, that the safeguard provided by the interior-point methods in terms of the logarithmic barrier function is not appropriate for general elastoplastic analysis where, using conventional methods, the number of iterations in each load step rarely exceeds 10. In the following, the standard primal–dual interior-point method is first briefly reviewed and a simple modification then proposed. As will be demonstrated by example, this modification leads to iteration counts similar to, or perhaps slightly better than, those typically encountered using conventional methods.

### *4.1. Standard primal–dual interior-point method*

With the standard primal–dual interior-point method, Newton's method is applied to the optimality conditions (48) so that, in each iteration, the following set of equations is solved:

$$\begin{bmatrix} -(\mathbf{M}+\mathbf{H}) & \hat{\mathbf{B}} & -\nabla\mathbf{f} & \cdot \\ \hat{\mathbf{B}}^{\mathsf{T}} & \cdot & \cdot & \cdot \\ \nabla\mathbf{f}^{\mathsf{T}} & \cdot & \cdot & \mathbf{I} \\ \cdot & \cdot & \mathbf{S} & \boldsymbol{\Lambda} \end{bmatrix}_{n+1}^{j} \begin{bmatrix} \Delta\hat{\boldsymbol{\sigma}} \\ \Delta\mathbf{u} \\ \Delta\boldsymbol{\lambda} \\ \Delta\mathbf{s} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathrm{ss}} \\ \mathbf{r}_{\mathrm{eq}} \\ \mathbf{r}_{\mathrm{yc}} \\ \mathbf{r}_{\mathrm{cs}} \end{bmatrix}_{n+1}^{j} \tag{53}$$

where superscript $j$ refers to the iteration number, $\mathbf{I}$ is a unit matrix, $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$, and

$$\mathbf{H} = \sum_{J=1}^{N_\sigma} \lambda_J \nabla^2_{\hat{\sigma}\hat{\sigma}} \mathbf{f}(\hat{\boldsymbol{\sigma}}) \tag{54}$$

System (53) is not symmetric, but can easily be symmetrized, for example by multiplication of the third row by $-1$ and the last row by $-\mathbf{S}^{-1}$. The computation of the increments of the variables, or in optimization terminology, the search direction, is followed by the computation of the maximum allowable step length subject to the condition that the new points $\boldsymbol{\lambda}_{n+1}^{j+1}$ and $\mathbf{s}_{n+1}^{j+1}$ remain strictly positive. If these variables are considered separately, two different maximum step lengths can be computed as

$$\omega_\lambda = \max_{\Delta\lambda_I<0} ((\lambda_{n+1}^{j})_I/\Delta\lambda_I) \quad \text{and} \quad \omega_s = \max_{\Delta s_I<0} (s_{n+1}^{j})_I/\Delta s_I) \tag{55}$$

Alternatively, a common maximum step length given by

$$\omega = \min(\omega_\lambda, \omega_s) \tag{56}$$



DOI: 10.1002/nme

can be used. The new point is then determined as

$$\mathsf{x}_{n+1}^{j+1} = \mathsf{x}_{n+1}^{j} + \theta\omega\Delta\mathsf{x}, \quad \mathsf{x} = \mathbf{u}, \hat{\boldsymbol{\sigma}}, \boldsymbol{\lambda}, \mathbf{s} \tag{57}$$

where $\theta$ is a scalar which is usually chosen as being close to unity, e.g. $\theta \simeq 0.95$–$0.9995$.

Finally, at the end of each iteration, a new barrier parameter is computed. There are numerous heuristic rules for this, but typically it is determined on the basis of the duality gap $\delta_{n+1}^{j} = (\mathbf{s}_{n+1}^{j})^{\mathsf{T}}\boldsymbol{\lambda}_{n+1}^{j}$. Thus, a very simple, but often effective strategy consists of computing the new barrier parameter as

$$\mu_{n+1}^{j+1} = \eta\delta_{n+1}^{j}/N_{\sigma} \tag{58}$$

where $\eta \simeq 0.2$–$0.7$. More elaborate rules are given in the interior-point literature, see, e.g. References [42–44, 66] and references therein. The algorithm described above has previously been applied to plastic limit analysis and design in References [48, 67].

### *4.2. Modified interior-point method*

The modification of the above algorithm when applied to incremental elastoplasticity concerns only the update rule. We first note that with a heuristic rule for updating $\mu$ and a damping factor less than unity, the quadratic rate of convergence that characterizes Newton's method under ideal circumstances in the limit of a vanishing residual cannot possibly be achieved. In fact, practical experience shows that, even with a good strategy for reducing $\mu$, the convergence rate is linear at best and often deteriorates as the problem size grows. In order to achieve convergence rates similar to those offered by conventional methods we have therefore chosen to make two modifications. First of all, the barrier parameter is chosen as $\mu = 0$ throughout. Secondly, we have found that the best performance is generally achieved if a full Newton step ($\theta = \omega = 1$) is always taken and the positively restricted variables then corrected after each iteration, should they become negative. Thus, after each update of the variables, negative entries in $\boldsymbol{\lambda}_{n+1}^{j+1}$ and $\mathbf{s}_{n+1}^{j+1}$ are set equal to a small positive number. Also, in order to ensure a numerically stable equation system, it is beneficial to limit the minimum values of $\boldsymbol{\lambda}_{n+1}^{j+1}$ and $\mathbf{s}_{n+1}^{j+1}$. Thus, at the end of each iteration, $\boldsymbol{\lambda}_{n+1}^{j+1}$ and $\mathbf{s}_{n+1}^{j+1}$ are corrected as

$$(\lambda_{n+1}^{j+1})_I := \max((\lambda_{n+1}^{j+1})_I, \varepsilon_\lambda), \quad (s_{n+1}^{j+1})_I := \max((s_{n+1}^{j+1})_I, \varepsilon_s) \tag{59}$$

where $\varepsilon_\lambda \simeq 10^{-9}$–$10^{-12}$ and $\varepsilon_s \simeq 10^{3}\varepsilon_\lambda$ are appropriate. Although very simple, this strategy is surprisingly efficient.

### *4.3. Implementation issues*

It is possible to reduce system (48) significantly prior to numerical solution. The two last sets of equations give

$$\Delta\boldsymbol{\lambda} = [-\mathbf{S}^{-1}(\boldsymbol{\Lambda}\Delta\mathbf{s} + \mathbf{r}_{\mathrm{cs}})]_{n+1}^{j}, \quad \Delta\mathbf{s} = [-\nabla\mathbf{f} - \mathbf{r}_{\mathrm{yc}}]_{n+1}^{j} \tag{60}$$

These can be used to eliminate $\Delta\boldsymbol{\lambda}$ from the first set of equations to give

$$\begin{bmatrix} -\mathbf{M}^{\mathrm{ep}} & \hat{\mathbf{B}} \\ \hat{\mathbf{B}}^{\mathsf{T}} & \mathbf{0} \end{bmatrix}_{n+1}^{j} \begin{bmatrix} \Delta\hat{\boldsymbol{\sigma}} \\ \Delta\mathbf{u} \end{bmatrix} = -\begin{bmatrix} \mathbf{g} \\ \mathbf{r}_{\mathrm{eq}} \end{bmatrix}_{n+1}^{j} \tag{61}$$



DOI: 10.1002/nme

where

$$\mathbf{M}^{\mathrm{ep}} = \mathbf{M} + \mathbf{H} + \nabla\mathbf{f}\mathbf{S}^{-1}\boldsymbol{\Lambda}\nabla\mathbf{f}^{\mathsf{T}}$$
$$\mathbf{g} = \mathbf{r}_{\mathrm{ss}} + \nabla\mathbf{f}\mathbf{S}^{-1}(\mathbf{r}_{\mathrm{cs}} - \boldsymbol{\Lambda}\mathbf{r}_{\mathrm{yc}}) \tag{62}$$

and finally the displacement and stress iterates can be determined separately as

$$\Delta\mathbf{u} = [\hat{\mathbf{B}}^{\mathsf{T}}\mathbf{C}^{\mathrm{ep}}\mathbf{B}]^{-1}\mathbf{r} \tag{63}$$

$$\Delta\hat{\boldsymbol{\sigma}} = \mathbf{C}^{\mathrm{ep}}(\hat{\mathbf{B}}\Delta\mathbf{u} + \mathbf{g}) \tag{64}$$

where

$$\mathbf{r} = -\mathbf{r}_{\mathrm{eq}} - \hat{\mathbf{B}}^{\mathsf{T}}\mathbf{C}^{\mathrm{ep}}\mathbf{g}, \quad \mathbf{C}^{\mathrm{ep}} = (\mathbf{M}^{\mathrm{ep}})^{-1} \tag{65}$$

Suppose now that the yield conditions are applied at the same integration points as those used for the integration of $\mathbf{M}$ and that only one yield condition is applied at each stress node. Suppose further that the stresses are interpolated on the basis of values at these points. The matrix $\mathbf{M}^{\mathrm{ep}}$ is then a block-diagonal matrix with each block representing a stress point. Thus, at each stress point $I$ we have

$$\mathbf{M}_I^{\mathrm{ep}} = \mathbf{M}_I + \mathbf{H}_I + \mathbf{a}_I s_I^{-1}\lambda_I\mathbf{a}_I^{\mathsf{T}} \tag{66}$$

where

$$\mathbf{a}_I = \nabla f_I(\hat{\boldsymbol{\sigma}}_I), \quad \mathbf{H}_I = \lambda_I\nabla^2 f_I(\hat{\boldsymbol{\sigma}}) \tag{67}$$

Using the Sherman–Morrison formula the inverse of (66) is given by

$$(\mathbf{M}_I^{\mathrm{ep}})^{-1} = \mathbf{C}_I^{\mathrm{ep}} = \mathbf{C}_I - \mathbf{C}_I\mathbf{a}_I(\lambda_I^{-1}s_I + \mathbf{a}_I^{\mathsf{T}}\mathbf{C}_I\mathbf{a}_I)^{-1}\mathbf{a}_I^{\mathsf{T}}\mathbf{C}_I \tag{68}$$

or

$$\mathbf{C}_I^{\mathrm{ep}} = \mathbf{C}_I - \mathbf{C}_I\mathbf{a}_I D_I\mathbf{a}_I^{\mathsf{T}}\mathbf{C}_I \tag{69}$$

where

$$\mathbf{C}_I = [\mathbf{M}_I + \mathbf{H}_I]^{-1}, \quad D_I = \lambda_I(s_I + \mathbf{a}_I^{\mathsf{T}}\mathbf{C}_I\mathbf{a}_I\lambda_I)^{-1} \tag{70}$$

Similarly, for each stress point a residual vector can be computed as

$$\mathbf{r}_I = -\mathbf{r}_{\mathrm{eq},I} - \hat{\mathbf{B}}_I^{\mathsf{T}}\mathbf{C}_I^{\mathrm{ep}}\mathbf{g}_I \tag{71}$$

By some rather lengthy but otherwise straightforward manipulations this reduces to

$$\mathbf{r}_I = \hat{\mathbf{B}}_I^{\mathsf{T}}(\mathbf{q}_I - \hat{\boldsymbol{\sigma}}_I) + \mathbf{p}_I \tag{72}$$

where

$$\mathbf{q}_I = -\mathbf{C}_I^{\mathrm{ep}}\mathbf{r}_{\mathrm{ss},I} + \mathbf{C}_I\mathbf{a}_I D_I f_I(\hat{\boldsymbol{\sigma}}_I) \tag{73}$$

The important point is here that, in contrast to (62), neither the elastoplastic modulus (69) nor the residual vector (73) involve the term $s_I^{-1}$. Thus, it is in fact possible to use a value of $s_I$ identically equal to zero and the same of course goes for $\lambda_I$ in which case the linear elastic stiffness modulus appears. This is utilized in the following section. Before that, however, the calculation of $\Delta\hat{\boldsymbol{\sigma}}$, $\Delta\boldsymbol{\lambda}$, and $\Delta\mathbf{s}$ is considered. Recall that the sequence in which the iterates are computed is

$$\underset{(63)}{\Delta\mathbf{u}} \rightarrow \underset{(64)}{\Delta\hat{\boldsymbol{\sigma}}} \rightarrow \underset{(60)_2}{\Delta\mathbf{s}} \rightarrow \underset{(60)_1}{\Delta\boldsymbol{\lambda}} \tag{74}$$



DOI: 10.1002/nme

Obviously, any inaccuracy contained in $\Delta\mathbf{u}$ can potentially be amplified in each of the following steps. The situation is further worsened by the fact that the final step, namely the computation of the plastic multiplier increments, is extremely sensitive. Thus, in the case of yielding $s_I^{-1}$ tends to infinity whereas $\Delta\lambda_I = -s_I^{-1}(\lambda_I \Delta s_I + r_{\mathrm{cs},I})$ should ultimately approach zero. Such computations are bound to suffer from inaccuracies which ultimately destroy the overall convergence and in many cases even lead to divergence. Alternatively, with $\Delta\mathbf{u}$ determined, $\Delta\hat{\boldsymbol{\sigma}}$, $\Delta\boldsymbol{\lambda}$, and $\Delta\mathbf{s}$ can be computed by solving a reduced system of equations. Again, this can be done separately for each stress point. From (53) it is immediately seen that these equations can be written as

$$\begin{bmatrix} \mathbf{M}_I + \mathbf{H}_I & \mathbf{a}_I & \cdot \\ \mathbf{a}_I^{\mathsf{T}} & \cdot & 1 \\ \cdot & s_I & \lambda_I \end{bmatrix}_{n+1}^{j} \begin{bmatrix} \Delta\hat{\boldsymbol{\sigma}}_I \\ \Delta\lambda_I \\ \Delta s_I \end{bmatrix} = \begin{bmatrix} \Delta\boldsymbol{\varepsilon}_I + \mathbf{r}_{\mathrm{ss},I} \\ -f(\hat{\boldsymbol{\sigma}}_I) - s_I \\ -s_I\lambda_I \end{bmatrix}_{n+1}^{j} \tag{75}$$

where $\Delta\boldsymbol{\varepsilon}_I = (\hat{\mathbf{B}}\Delta\mathbf{u})_I$ is the iterative strain increment at the $I$th stress node. Finally, all variables are updated as

$$\mathsf{x}_{n+1}^{j+1} = \mathsf{x}_{n+1}^{j} + \Delta\mathsf{x}, \quad \mathsf{x} = \mathbf{u}, \hat{\boldsymbol{\sigma}}, \boldsymbol{\lambda}, \mathbf{s} \tag{76}$$

and the iterative procedure, starting with the solution of (63) and then skipping to (75), is repeated until convergence.

### *4.4. Active set strategy*

The procedure described above is generally quite robust and a point in very close proximity to the optimal solution is determined in typically 3–10 iterations, depending on the extent of plasticity and the change in structural behaviour from one increment to another. However, setting the slack variables and plastic multipliers equal to some small number, when the final solution dictates that they should be equal to zero, leads to a less than optimal rate of convergence, although in practice it is in fact often almost quadratic. Nevertheless, it is desirable to improve the convergence characteristics if possible. As with all inequality constrained problems the primary difficulty lies in identifying the *active set* during the iterations. The knowledge of which points are yielding of course does not guarantee that Newton's method converges when applied to the resulting equations. On the other hand, if convergence does occur the rate of this will be quadratic as the solution is approached. In order to achieve this in the present context, the values of $\varepsilon_s$ and $\varepsilon_\lambda$ should be adjusted automatically so that they attain values of zero once we are confident that the active set has been identified. Thus, once a slack variable $s_I$ has been identified as being equal to zero, solution of system (75) implies that $\Delta s_I$ will be equal to zero in all of the subsequent iterations so that no further correction of this variable is needed. Conversely, if $\lambda_I = 0$ has been identified, $\lambda_I$ remains equal to zero in all subsequent iterations. With these considerations in mind the identification of the active set proceeds as follows. First of all, unless the residual is below some tolerance $ATOL$, nothing is done. This provides a safeguard for making a wrong identification of the active set. If the criterion is met, the active (and hence the inactive) set is identified after each update and



DOI: 10.1002/nme

correction of the variables as

$$\begin{aligned} &\textit{For all yield constraints}, \quad k = 1, \ldots : \\ &\quad \text{if } \lambda_k = \varepsilon_\lambda \quad \text{and} \quad s_k > \varepsilon_s \rightarrow \lambda_k := 0 \ (\text{elastic}) \\ &\quad \text{if } \lambda_k > \varepsilon_\lambda \quad \text{and} \quad s_k = \varepsilon_s \rightarrow s_k := 0 \ (\text{yielding}) \end{aligned} \tag{77}$$

### 4.5. Multisurface generalization

The generalization of the algorithm to multisurface plasticity is very straightforward and essentially involves substituting scalar and vector quantities for vector and matrix quantities, respectively. Consider the situation where $\mathrm{NS}_I$ yield conditions are applied at each stress node $I$. We introduce the following quantities:

$$\begin{aligned} \mathbf{f}_I &= [f_I^1(\hat{\boldsymbol{\sigma}}_I), \ldots, f_I^{\mathrm{NS}_I}(\hat{\boldsymbol{\sigma}}_I)]^{\mathsf{T}} \\ \mathbf{A}_I &= [\mathbf{a}_I^1, \ldots, \mathbf{a}_I^{\mathrm{NS}_I}] \\ \mathbf{s}_I &= [s_I^1, \ldots, s_I^{\mathrm{NS}_I}], \quad \mathbf{S}_I = \mathrm{diag}(\mathbf{s}_I) \\ \boldsymbol{\lambda}_I &= [\lambda_I^1, \ldots, \lambda_I^{\mathrm{NS}_I}], \quad \boldsymbol{\Lambda}_I = \mathrm{diag}\,(\boldsymbol{\lambda}_I) \end{aligned} \tag{78}$$

and the multisurface equivalents to (69) and (73) are then

$$\mathbf{C}_I^{\mathrm{ep}} = \mathbf{C}_I - \mathbf{C}\mathbf{A}_I\mathbf{D}_I\mathbf{A}_I^{\mathsf{T}}\mathbf{C}_I \tag{79}$$

$$\mathbf{q}_I = -\mathbf{C}_I^{\mathrm{ep}}\mathbf{r}_{\mathrm{ss},I} + \mathbf{C}\mathbf{A}_I\mathbf{D}_I\mathbf{f}_I \tag{80}$$

where

$$\mathbf{C}_I = \left[\mathbf{M}_I + \sum_{k=1}^{\mathrm{NS}_I} \lambda_I^k \nabla^2 f_I^k(\hat{\boldsymbol{\sigma}}_I)\right]^{-1}, \quad \mathbf{D}_I = \boldsymbol{\Lambda}_I(\mathbf{S}_I + \mathbf{A}_I^{\mathsf{T}}\mathbf{C}_I\mathbf{A}_I\boldsymbol{\Lambda}_I)^{-1} \tag{81}$$

Likewise, system (75) generalizes to

$$\begin{bmatrix} \Delta\hat{\boldsymbol{\sigma}}_I \\ \Delta\boldsymbol{\lambda}_I \\ \Delta\mathbf{s}_I \end{bmatrix} = \begin{bmatrix} \mathbf{M}+\mathbf{H} & \mathbf{A} & \cdot \\ \mathbf{A}^{\mathsf{T}} & \cdot & \mathbf{I} \\ \cdot & \mathbf{S} & \boldsymbol{\Lambda} \end{bmatrix}_I^{-1} \begin{bmatrix} \mathbf{B}\Delta\mathbf{u} + \mathbf{r}_{\mathrm{ss}} \\ -\mathbf{f} - \mathbf{s} \\ -\mathbf{S}\boldsymbol{\lambda} \end{bmatrix}_I \tag{82}$$

and the variables are updated as in (76). Obviously, the above relations contain the single surface ones as the special case where $\mathrm{NS}_I = 1$ for all $I$.

The final multisurface algorithm is summarized in Box 1. As indicated, the iterations are broken off when the norm of the total residual becomes less than some tolerance ITOL, typically, $\mathrm{ITOL} = 10^{-9}$.

It is noted that constraints which have initially been wrongly identified as active may at a later stage be excluded from the active set. For example if a given constraint $k$ has been identified as active, we set $s_k = 0$. In the next iteration the tangent modulus and residual vector will reflect this, but it will not make much difference as compared to the case where $s_k = \varepsilon_s$. In the local solution step, on the other hand, it does make a difference as we will have $\Delta s_k = 0$. However, at some stage $\lambda_k$ may well become negative and after correction we would then have $\lambda_k = \varepsilon_\lambda$ and $s_k = \varepsilon_s$ and the constraint would no longer be classified as active. However, in order to avoid too many of these



DOI: 10.1002/nme

classification changes it is prudent to not begin identifying the active set before the solution is well on its way to converging. In Box 1 the measure for this is given by the parameter ATOL. Also, at the beginning of each new load step all information about the active set in the previous step is discarded.

Box 1. Solution algorithm.

*Initialize:* $\hat{\boldsymbol{\sigma}}_0,\ \mathbf{u}_0,\ \boldsymbol{\lambda}_0,\ \mathbf{s}_0$

*Load steps,* $n = 0, \ldots, n_{\max}$:

$\mathbf{u}^0_{n+1} = \mathbf{u}_n,\ \hat{\boldsymbol{\sigma}}^0_{n+1} = \hat{\boldsymbol{\sigma}}_n,$

$\mathbf{s}^0_{n+1} := \mathbf{max}(\mathbf{s}_n, \varepsilon_s \mathbf{e}),\ \boldsymbol{\lambda}^0_{n+1} := \mathbf{max}(\boldsymbol{\lambda}_n, \varepsilon_\lambda \mathbf{e}),$

*Iterations,* $j = 0, \ldots, j_{\max}$:

$$\mathbf{r}_{\mathrm{ss}} = -\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1} - \hat{\boldsymbol{\sigma}}_n) - \mathbf{A}_{n+1}\boldsymbol{\lambda}_{n+1} + \hat{\mathbf{B}}(\mathbf{u}_{n+1} - \mathbf{u}_n)$$

$$\mathbf{r}_{\mathrm{eq}} = \hat{\mathbf{B}}^{\mathsf{T}}\hat{\boldsymbol{\sigma}}_{n+1} - \mathbf{p}$$

$$\mathbf{r}_{\mathrm{yc}} = -\mathbf{f}_{n+1} - \mathbf{s}_{n+1}$$

$$\mathbf{r}_{\mathrm{rc}} = \mathbf{S}_{n+1}\boldsymbol{\lambda}_{n+1}$$

*Tangent matrix* $I = 1, \ldots, N_\sigma$:

$$\mathbf{C}_I = \left[\mathbf{M}_I + \sum_{k=1}^{\mathrm{NS}_I} \lambda_I^k \nabla^2 f_I^k(\hat{\boldsymbol{\sigma}}_I)\right]^{-1}$$

$$\mathbf{D}_I = \boldsymbol{\Lambda}_I(\mathbf{S}_I + \mathbf{A}_I^{\mathsf{T}}\mathbf{C}_I\mathbf{A}_I\boldsymbol{\Lambda}_I)^{-1}$$

**** $$\mathbf{C}_I^{\mathrm{ep}} = \mathbf{C}_I - \mathbf{C}_I\mathbf{A}_I\mathbf{D}_I\mathbf{A}_I^{\mathsf{T}}\mathbf{C}_I$$

*Global solve*:

$$\mathbf{q} = -\mathbf{C}^{\mathrm{ep}}\mathbf{r}_{\mathrm{ss}} + \mathbf{CADf},\quad \mathbf{r} = \hat{\mathbf{B}}^{\mathsf{T}}(\mathbf{q} - \hat{\boldsymbol{\sigma}}) + \mathbf{p}$$

$$\mathbf{K} = \hat{\mathbf{B}}^{\mathsf{T}}\mathbf{C}^{\mathrm{ep}}\hat{\mathbf{B}}$$

$$\Delta\mathbf{u} = \mathbf{K}^{-1}\mathbf{r}$$

*Local solve,* $I = 1, \ldots, N_\sigma$:

$$\begin{bmatrix} \Delta\hat{\boldsymbol{\sigma}}_I \\ \Delta\boldsymbol{\lambda}_I \\ \Delta\mathbf{s}_I \end{bmatrix} = \begin{bmatrix} \mathbf{M}+\mathbf{H} & \mathbf{A} & \cdot \\ \mathbf{A}^{\mathsf{T}} & \cdot & \mathbf{I} \\ \cdot & \mathbf{S} & \boldsymbol{\Lambda} \end{bmatrix}_I^{-1} \begin{bmatrix} \mathbf{B}\Delta\mathbf{u} + \mathbf{r}_{ss} \\ -\mathbf{f} - \mathbf{s} \\ -\mathbf{S}\boldsymbol{\lambda} \end{bmatrix}_I$$

*Update and correct*:

$$\mathsf{x}^{j+1}_{n+1} = \mathsf{x}^j_{n+1} + \Delta\mathsf{x},\qquad \mathsf{x} = \mathbf{u},\ \hat{\boldsymbol{\sigma}},\ \boldsymbol{\lambda},\ \mathbf{s}$$

$$\mathbf{s}^{j+1}_{n+1} := \mathbf{max}(\mathbf{s}^{j+1}_{n+1}, \varepsilon_s\mathbf{e}),\quad \boldsymbol{\lambda}^{j+1}_{n+1} := \mathbf{max}(\boldsymbol{\lambda}^{j+1}_{n+1}, \varepsilon_\lambda\mathbf{e}),$$

*Identify active set (optional)*:

*If* $|||[\mathbf{r}_{\mathrm{ss}}, \mathbf{r}_{\mathrm{eq}}, \mathbf{r}_{\mathrm{yc}}, \mathbf{r}_{\mathrm{cs}}]|| < ATOL$

if $\lambda_k = \varepsilon_\lambda$ and $s_k > \varepsilon_s \ \rightarrow\ \lambda_k := 0$ (elastic)

if $\lambda_k > \varepsilon_\lambda$ and $s_k = \varepsilon_s \ \rightarrow\ s_k := 0$ (yielding)

*End iterations if* $|||[\mathbf{r}_{\mathrm{ss}}, \mathbf{r}_{\mathrm{eq}}, \mathbf{r}_{\mathrm{yc}}, \mathbf{r}_{\mathrm{cs}}]|| < ITOL$

*End load step*



DOI: 10.1002/nme

### 4.6. Line search procedure

Occasionally, when the load step is above some critical magnitude or when the curvature of the yield function is very high (or a combination of the two), the algorithm fails to converge. Besides reducing the step size such difficulties can in many cases be overcome by a line search so that a damped, rather than a full, Newton step is taken. In the current implementation, we compare the current residual to the residual resulting from a full Newton step before the variables are updated, and if the norm of the latter is greater than the norm of the former a line search procedure is evoked. This line search can be performed in a number of ways and a large body of literature exists on the subject, see, e.g. Reference [68] and references therein. We have implemented a simple bisection procedure where, in order to locate the minimum of the residual norm as function of the step size with sufficient accuracy, some 5–10 residual evaluations are typically required.

### 4.7. Relation to Simo–Taylor method

It is apparent that the quantities defining the solution algorithm have much in common with those of the celebrated method of Simo and Taylor [2]. Thus, the elastoplastic tangent modulus (68)–(69) is identical to that derived by Simo and Taylor if either $s_I = 0$ or $\lambda_I = 0$. In view of the fact that the same physical problem is solved using the same method, namely Newton's method, this is of course not too surprising. Furthermore, the algorithm also contains an analogy to the return-mapping iterations employed in the Simo–Taylor method. As is well-known, the return-mapping procedure is equivalent to solving the following closest point projection (i.e. distance minimization) problem at each Gauss point

$$\begin{array}{ll} \underset{\hat{\boldsymbol{\sigma}}_{n+1}^{j+1}}{\text{minimize}} & \frac{1}{2}(\hat{\boldsymbol{\sigma}}_{n+1}^{j+1} - \hat{\boldsymbol{\sigma}}_{\text{trial}})^{\mathsf{T}}\mathbf{M}(\hat{\boldsymbol{\sigma}}_{n+1}^{j+1} - \hat{\boldsymbol{\sigma}}_{\text{trial}}) \\ \text{subject to} & f(\hat{\boldsymbol{\sigma}}_{n+1}^{j+1}) \leqslant 0 \end{array} \tag{83}$$

where the trial stress state $\hat{\boldsymbol{\sigma}}_{\text{trial}}$ follows from an elastic prediction given by

$$\hat{\boldsymbol{\sigma}}_{\text{trial}} = \hat{\boldsymbol{\sigma}}_n + \mathbf{M}^{-1}\Delta\mathbf{e} \tag{84}$$

where $\Delta\mathbf{e}$ is the *total* strain increment

$$\Delta\mathbf{e} = \hat{\mathbf{B}}(\mathbf{u}_{n+1}^{j+1} - \mathbf{u}_n) \tag{85}$$

The principle behind the method is now to solve the closest point projection problem (83) in each iteration and thus obtain the new iterative stress state $\hat{\boldsymbol{\sigma}}_{n+1}^{j+1}$ and plastic multiplier $\lambda_{n+1}^{j+1}$, i.e. the Lagrange multiplier associated with the constraint in (83).

If Newton's method is applied to the KKT conditions associated with (83) and the iterations are initiated from the point $(\cdot)_{n+1}^{j}$ it is easily verified that the first iteration corresponds to the solution of (75). In contrast to the Simo–Taylor method, however, the present algorithm only takes this one iteration at the local level after which a global iteration is performed and so on until convergence. Since the local constitutive conditions are not fulfilled at intermediate stages of the iterative procedure the resulting residuals appear in the global residual, namely via $\mathbf{q}_I$ in (72)–(73).

Thus, whereas there are some similarities between the two methods, there are also fundamental differences. The question of course then arises as to which method is 'best'. Since the cost of solving the global system equations is usually much higher than the cost of performing the local



DOI: 10.1002/nme

return mapping iterations, at least for large-scale problems, the computational effort required in each iteration appears to be quite similar for the two methods. Furthermore, on the basis of extensive numerical experimentation we can conclude that the number of iterations used in each load step for typical problems are quite similar. Regarding the issue of robustness, experience also shows that the current method performs very well and special procedures such as line searches are rarely needed, at least for moderately curved yield conditions such as that of von Mises. In contrast, such procedures are often reported to be necessary for the Simo–Taylor method, even with a moderate number of elements and for simple constitutive relations [55, 69].

Which method is more 'natural' in the sense of tackling the problem at hand in the most direct way is another question which of course is highly debatable. In any case, the method presented here breaks with the tradition of satisfying the local constitutive equations at the end of each iteration. From a mechanical point of view this could be seen as being somewhat dubious whereas from an optimization point of view it is completely in accordance with the well established rule-of-thumb that it is desirable to have all residuals converge at more or less the same rate.

Also the basic idea of viewing the problem as one containing two distinct levels, the global (structural) and local (material), is initially abandoned although, if a specific spatial discretization implies a decoupling of the stress points, two such levels can indeed be identified and exploited to formulate an efficient algorithm.

## 5. EXAMPLES

In the following a number of examples problems are solved to demonstrate the performance of the proposed algorithm. In all examples the algorithm parameters were chosen as: $\varepsilon_\lambda = 10^{-12}$, $\varepsilon_s = 10^{-9}$, ITOL $= 10^{-9}$, ATOL $= 10^{-4}$.

### *5.1. Example 1: von Mises plasticity*

The first example concerns the perforated plate shown in Figure 3. A total of ten displacement increments of magnitude $\Delta u = 2 \times 10^{-3}$ are applied along the top boundary, giving rise to a load–displacement curve as shown in Figure 4. In addition to applying the proposed algorithm, the problem is also solved using the second-order cone optimizer MOSEK (www.mosek.com) which has recently been applied to large-scale limit and shakedown analysis with some success [70–72]. A comparison of the two methods in terms of the number of iterations in each load step and the total CPU time is given in Table I. As seen the number of iterations needed by MOSEK is fairly constant and appears to be independent of the load level. In contrast, the proposed algorithm requires slightly more iterations for load steps that bring about significant changes in the spread of plasticity. Overall the present algorithm requires roughly one-third of the iterations used by MOSEK. A smaller total CPU time is therefore to be expected. However, in comparing the CPU time per iteration, the present method outperforms MOSEK by an additional factor of three. It should be mentioned here that since the present method was implemented in MATLAB, only the time spent solving linear equations was recorded. However, for problems such as the one considered in this example the bulk of the total CPU time will be used on this task and the comparison is therefore meaningful. Since it is not to be expected that the solver used by MOSEK is less efficient than the UMFPACK solver used in MATLAB the discrepancy between the solution times must be due to the structure of the equations actually solved. Indeed, if the original KKT system (53) is solved without



DOI: 10.1002/nme

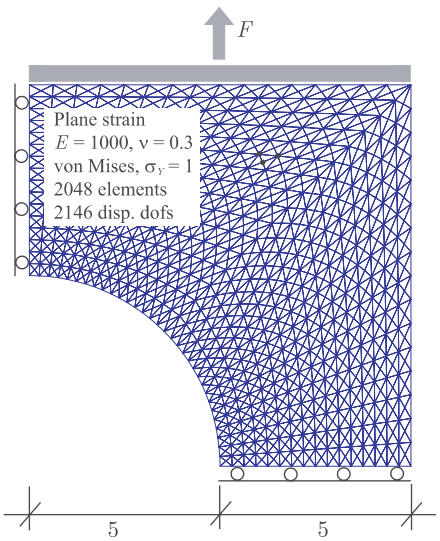


Figure 3. Perforated plate.

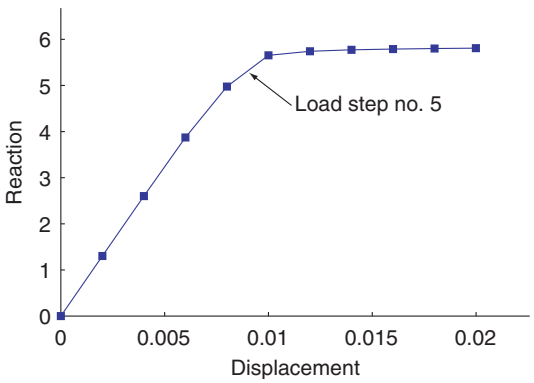


Figure 4. Load–displacement curves for perforated plate.

Table I. Iteration counts and CPU times for perforated plate example.

| Step no. | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Tot. iter. | CPU (s) | CPU/Iter. (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MOSEK | 18 | 18 | 18 | 17 | 17 | 17 | 18 | 18 | 141 | 66 | 0.47 |
| Present | 4 | 5 | 7 | 6 | 6 | 5 | 4 | 4 | 41 | 6* | 0.15* |

*Solution of linear equations only.

reduction we achieve similar CPU times as those offered by MOSEK. These results illustrate the fact that although it may be convenient to apply general purpose black box optimizers to this and similar problems, the potential savings that can be made by paying due attention to the structure



DOI: 10.1002/nme

Table II. Residual norm and primal and dual objectives (52) for load step 5.

| | MOSEK | | | Present | | |
|---|---|---|---|---|---|---|
| Iter. no. | Res. norm | Primal | Dual | Res. norm | Primal | Dual |
| 1 | 7.64e−03 | 0.000038 | 66593.75 | 1.71e+02 | 1.094380 | 1.095379 |
| 2 | 2.08e−03 | 0.007244 | 345.1532 | 1.41e+01 | 1.093736 | 1.079687 |
| 3 | 7.00e−02 | 0.086998 | 13.50159 | 3.66e−01 | 1.078826 | 1.077050 |
| 4 | 7.30e−02 | 0.125820 | 10.37086 | 7.17e−02 | 1.077909 | 1.077034 |
| 5 | 7.40e−02 | 0.144572 | 8.448395 | 2.36e−03 | 1.077041 | 1.077035 |
| 6 | 6.30e−02 | 0.157813 | 8.038101 | 1.21e−05 | 1.077035 | 1.077035 |
| 7 | 8.10e−01 | 0.401708 | 6.662264 | 6.31e−11 | 1.077035 | 1.077035 |
| ⋮ | ⋮ | ⋮ | ⋮ | — | — | — |
| 13 | 1.20e−02 | 1.076804 | 1.077013 | | | |
| 14 | 2.90e−03 | 1.076987 | 1.077019 | | | |
| 15 | 3.20e−04 | 1.077030 | 1.077031 | | | |
| 16 | 4.20e−05 | 1.077034 | 1.077035 | | | |
| 17 | 2.90e−06 | 1.077035 | 1.077035 | | | |
| 18 | 2.40e−07 | 1.077035 | 1.077035 | | | |

of the resulting KKT system is significant. Regarding the convergence of the residuals within a given load step, the two methods are compared for the critical load step no. 5 in Table II. It is evident here that MOSEK and the proposed algorithm are characterized by linear and quadratic convergence rates, respectively.

### *5.2. Example 2: Drucker–Prager plasticity*

The next example concerns the problem of a centrally loaded strip footing on a weightless, cohesive-frictional soil. The material is linear elastic/perfectly plastic with yielding governed by a smoothed Drucker–Prager criterion:

$$f(\boldsymbol{\sigma}) = \alpha I_1 + \sqrt{J_2 + \delta^2} - k \tag{86}$$

where the parameters $\alpha$ and $k$ are chosen as

$$\alpha = \frac{\tan\phi}{3\sqrt{1+\frac{4}{3}\tan^2\phi}}, \quad k = \frac{c}{\sqrt{1+\frac{4}{3}\tan^2\phi}}, \quad \delta = \frac{0.05c}{\sqrt{1+\frac{4}{3}\tan^2\phi}} \tag{87}$$

so that (86) matches the rounded Mohr–Coulomb criterion of Abbo and Sloan [73], with cohesion $c$ and friction angle $\phi$, in plane strain.

In the following analyses we use a Young's modulus of $E = 3000$, a Poisson's ratio of $\nu = 0.3$, and a cohesion of $c = 1$. Two meshes as shown in Figure 5 are used. For $B/L = 0$ and $\delta = 0$ the ultimate load is given by the well-known Prandtl solution

$$V_u/B = c\left(\tan^2\left(\tfrac{\pi}{4}+\tfrac{\phi}{2}\right)e^{\pi\tan\phi} - 1\right)\cot\phi \tag{88}$$

First the problem is solved with the coarse mesh using friction angles of $\phi = 20°$, $30°$, and $\phi = 40°$. For these analyses the two quadratic displacement/linear stress triangles discussed in Section 3.2 are used and 25 uniform displacement increments of $\Delta u = 0.02$ are applied underneath the footing.



DOI: 10.1002/nme

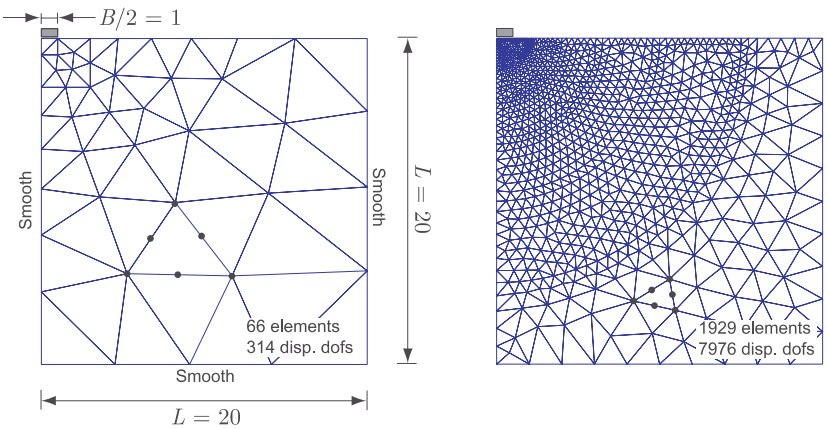


Figure 5. Strip footing problem: coarse and fine meshes.

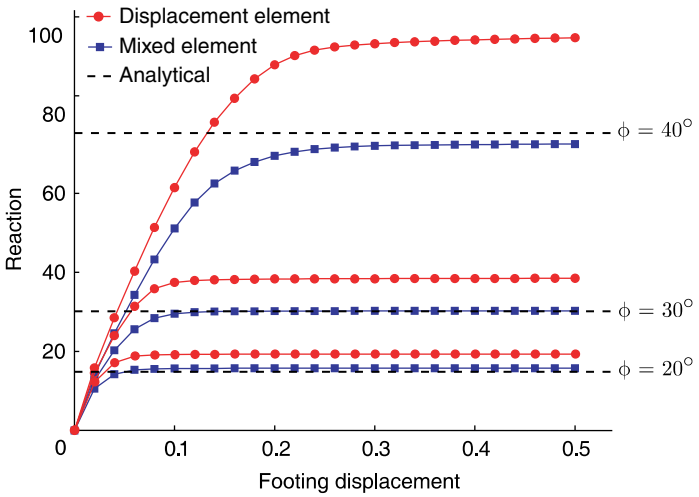


Figure 6. Strip footing problem: results for coarse mesh.

The results of the analyses are shown in Figure 6. As seen, there is a marked difference between the results produced by the two elements with the mixed element being far superior to the standard displacement element. Since the difference between the two elements is essentially quite small these results are perhaps somewhat surprising. Nevertheless, the same trend has been observed for a large number of other problems and is consistent with the results obtained by Pontes *et al.* [60] who, for a similar problem, used relatively coarse meshes to compute collapse loads to within a few percent of the exact ones.

Next, the problem where $\phi = 30^\circ$ is considered in some detail for the case of the fine mesh. The load–displacement curve for this case, using the mixed element and 100 displacement increments of $\Delta u = 0.0015$, is shown in Figure 7. In Table III some solution statistics are given.



DOI: 10.1002/nme

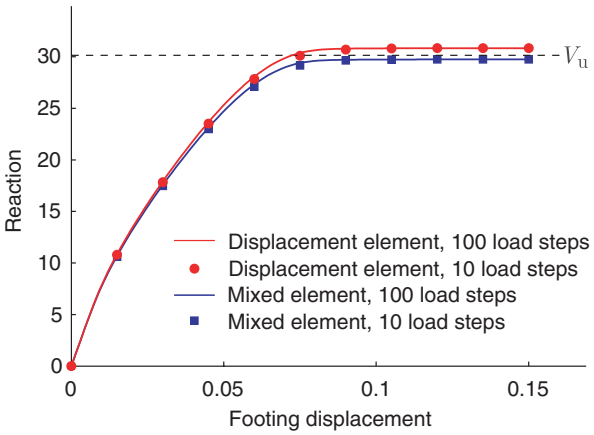


Figure 7. Strip footing problem: load–displacement curves for fine mesh.

Table III. Solution statistics for strip footing problem using fine mesh and mixed quadratic displacement/linear stress triangle.

| No. steps | Iter, no l.s. | Iter (l.s.) | Iter/step | $V(u=0.15)$ |
|---|---|---|---|---|
| 10 | * | 114 (17) | 11.4 | 29.7418 |
| 20 | * | 167 (12) | 8.4 | 29.7426 |
| 40 | 273 | 275 (5) | 6.9 | 29.7430 |
| 60 | 377 | 378 (5) | 6.3 | 29.7432 |
| 80 | 474 | 473 (3) | 5.9 | 29.7433 |
| 100 | 546 | 545 (4) | 5.5 | 29.7433 |
| 200 | 952 | 951 (2) | 4.8 | 29.7434 |
| 400 | 1656 | 1656 (0) | 4.1 | 29.7434 |

*Convergence not achieved after 200 iterations in the same load step.

As expected the number of iterations per load step decreases as the number of steps increases. For 10 and 20 load steps it is further necessary, occasionally, to employ a line search procedure in order to achieve convergence. On the other hand, when the number of steps increases above 40 the basic Newton method suffices and the effect of line search becomes marginal.

Regarding the accuracy of the analysis we note that the effect of the number of steps only has a small influence on the final footing force at a displacement of $u=0.15$. Thus, regardless of the number of steps, the final footing force is only slightly more than 1% *below* the analytical solution of $V_u=30.14$.

Next, we examine the effects of relaxing the convergence tolerance ITOL. The results, in terms of the total number of iterations required for different tolerances and number of steps, is shown in Table IV. Whereas it is not too surprising that relaxing the convergence tolerance leads to a smaller number of total iterations, it is somewhat surprising that the footing load–displacement curve is relatively unaffected, even for very large tolerances and a small number of steps. The results for the extreme case of 10 steps and a tolerance of ITOL = 1.0 are shown in Figure 8 and,



DOI: 10.1002/nme

Table IV. Effect of convergence criterion for strip footing problem using fine mesh and mixed quadratic displacement/linear stress element.

| Steps | ITOL = 1.0 Iter (l.s.) | ITOL = 0.1 Iter (l.s.) | ITOL = $10^{-3}$ Iter (l.s.) | ITOL = $10^{-9}$ Iter (l.s.) |
|---|---|---|---|---|
| 10 | 51 (10) | 65 (11) | 89 (18) | 114 (17) |
| 25 | 59 (2) | 85 (5) | 139 (11) | 195 (10) |
| 50 | 79 (0) | 135 (0) | 211 (8) | 331 (5) |
| 100 | 114 (0) | 196 (2) | 336 (4) | 545 (4) |

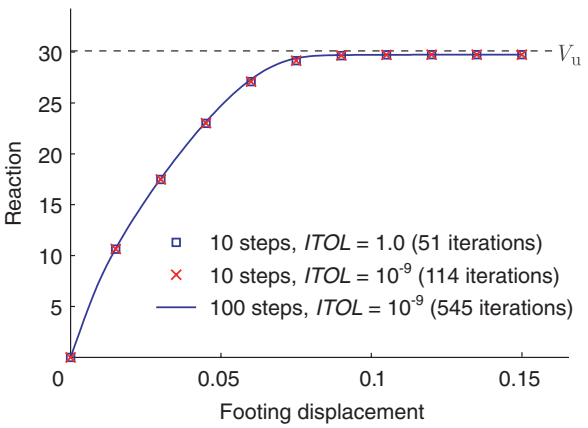


Figure 8. Strip footing problem: load–displacement curves for fine mesh.

as can be seen, the solution is virtually indistinguishable from the solution produced using 100 steps and ITOL = $10^{-9}$. In terms of the ultimate load, for example, the difference is less than one hundredth of a percent. A partial explanation for this is that even if the tolerance is very high, there is a tendency for the optimality conditions to be violated only at a very limited number of points. These violations concern primarily the yield criterion, with the other residuals usually being much smaller. Since the total complementary energy (which is the quantity optimized) is comprised of contributions from all stress points in the domain, a significant violation of the yield criterion at a few nodes does not necessarily induce major global errors.

Finally, the progress of the algorithm for a selected number of load steps in the case where 10 steps are used and ITOL = $10^{-9}$ is shown in Figure 9. Clearly, the steps in which the largest changes occur require the most iterations whereas the iteration counts decrease as the ultimate limit state is approached. Also, a quadratic rate of convergence is achieved soon after the active set has been identified.

### *5.3. Example 3: Multi-surface plasticity*

The next example concerns a model where the overall yield surface is defined by three surfaces—a cone, a hardening cap, and tension cut-off as shown in Figure 10. This type of model has been widely applied to geomaterials and a large number of models, containing a smaller or a larger



DOI: 10.1002/nme

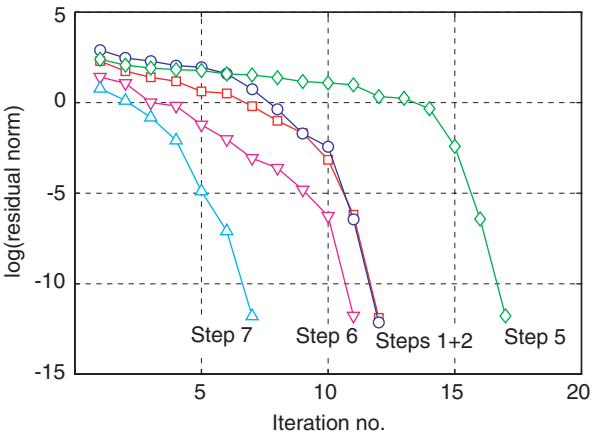


Figure 9. Strip footing problem: convergence results for fine mesh, 10 steps, and ITOL $= 10^{-9}$.

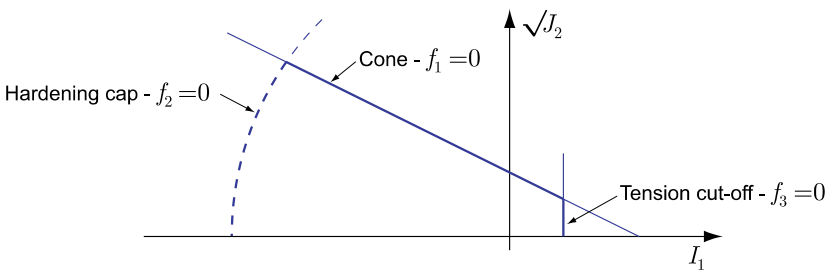


Figure 10. Three-surface yield criterion.

number of material parameters, can be found in the literature, e.g. References [74–78]. In the following a somewhat simplified model is used, with the three yield surfaces given by:

$$\begin{aligned} &\text{Cone:} && f_1(\boldsymbol{\sigma}) = \alpha I_1 + \sqrt{J_2} - k \\ &\text{Hardening cap:} && f_2(\boldsymbol{\sigma}, \kappa) = \sqrt{I_1^2 + J_2} - (c_0 + h\kappa) \\ &\text{Tension cut-off:} && f_3(\boldsymbol{\sigma}) = I_1 - c_t \end{aligned} \tag{89}$$

where $\alpha$, $k$, $c_0$, $h$ and $c_t$ are material parameters. The hardening potential is taken to be

$$\psi^p(\kappa) = \tfrac{1}{2} H^{-1} \kappa^2 \tag{90}$$

so that the evolution of $\kappa$ is given by

$$\dot{\kappa} = \mathring{\gamma} h H \tag{91}$$

The model is applied to the problem of a rigid footing on the top of a slope as shown in Figure 11 where a state of plane deformation is assumed. The model parameters used are shown in Table V.



DOI: 10.1002/nme

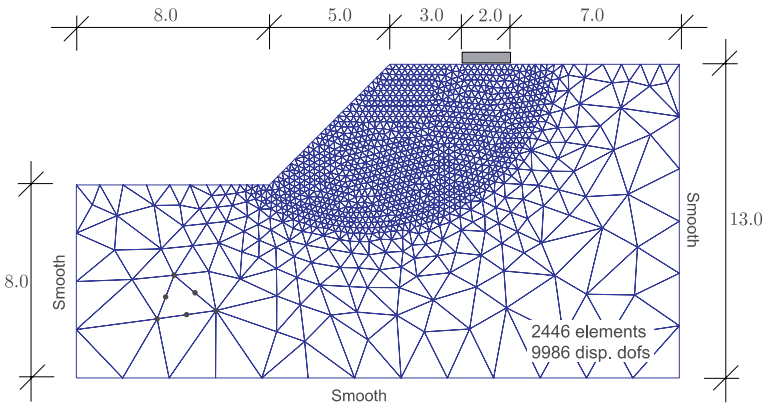


Figure 11. Strip footing on top of a slope (all measurements in m).

Table V. Model parameters for slope example. In all cases the soil unit weight is $\lambda = 18\,\mathrm{kN/m^3}$, Young's modulus is $E = 100\,000\,\mathrm{kPa}$, and Poisson's ratio is $\nu = 0.3$.

| | |
|---|---|
| $k/c$ (kPa/kPa) | 73.8/80.0 |
| $\alpha/\phi$ (dimensionless/°) | 0.112/20.0 |
| $c_0$ (kPa) | 450.0 |
| $c_t$ (kPa) | 5.0 |
| $H$ (kPa) | 100.0 |
| $h$ (dimensionless) | 2000; 3000;... 8000 |

The parameters $\alpha$ and $k$ are related to the plane strain Mohr–Coulomb $c$ and $\phi$ by (87). A total of 50 equally sized displacement increments were used.

The load–displacement curves for different values of $h$ are shown in Figure 12. Clearly, the effective stiffness of the footing after the initial stages, where the cap does not limit the possible states of stress, decreases as $h$ decreases.

In the course of the load stepping there are numerous stress points where two surfaces are active, in particular the cone and the cap. The evolution of the number of active yield constraints is shown in Figure 13where it is seen that activity of the cap for most part is accompanied by activity of the cone. Eventually, since the cap is capable of unlimited hardening, the ultimate failure is by way of activity of the cone and, to a much lesser extent, the tension cut-off. The distribution of the plastic multipliers associated with the cone at a point close to collapse is shown in Figure 14.

Regarding the performance of the algorithm there did not seem to be any noteworthy difference between this and the previous examples, the number of iterations per load step varied between 4 and 8 and a quadratic rate of convergence was observed in all load steps.

### *5.4. Example 4: Softening plasticity*

The last example tests the applicability of the proposed solution algorithm to a non-convex problem. The top boundary of the rectangular sheet shown in Figure 15(a) is subjected to a uniform vertical



DOI: 10.1002/nme

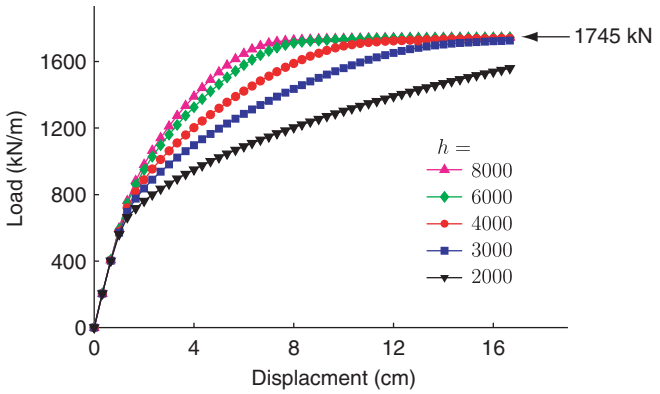


Figure 12. Load–displacement curves for strip footing on top of a slope.

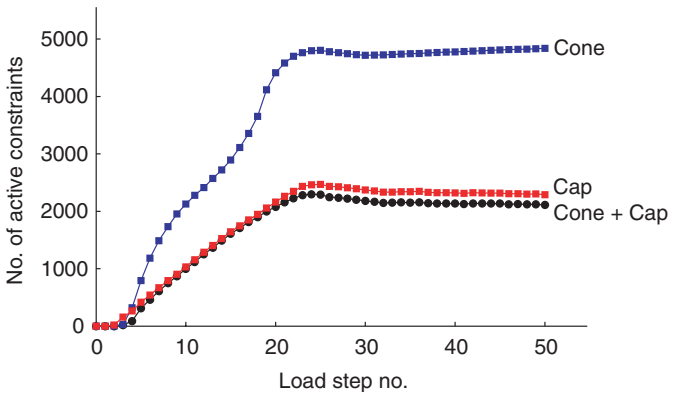


Figure 13. Activity of yield constraints for $h = 6000$.

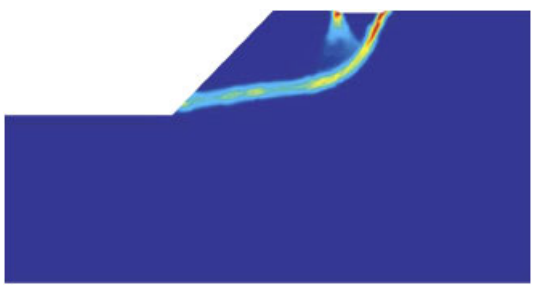

Figure 14. Cone plastic multiplier field close to point of collapse.



DOI: 10.1002/nme

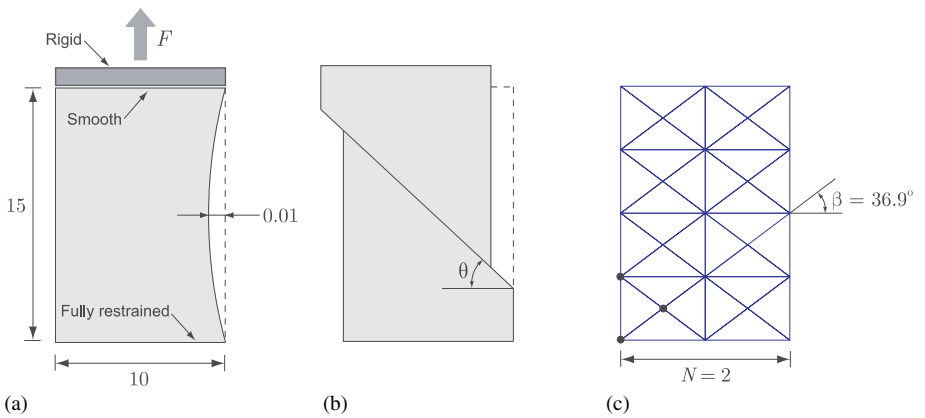


Figure 15. Example 4. Problem setup (a), discontinuous bifurcation (b), and finite element mesh (c).

displacement while the bottom boundary is fully restrained. Yielding is governed by an isotropically softening von Mises criterion given by

$$f(\boldsymbol{\sigma}, \kappa) = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x \sigma_y + 3\tau_{xy}^2} - k\mathrm{e}^{hl_{\mathrm{ch}}\kappa} \tag{92}$$

where $k>0$ and $h>0$ are material parameters and $l_{\mathrm{ch}}$ is a characteristic length. Following standard practice, e.g. Reference [79] and references therein, the characteristic length is related to the volume of the elements. For the constant strain triangles used in this example we set $l_{\mathrm{ch}} = \sqrt{A}$ with $A$ being the element area. The hardening (softening) potential is given by

$$\psi^{\mathrm{p}}(\kappa) = -\tfrac{1}{2} l_{\mathrm{ch}} H^{-1} \kappa^2 \tag{93}$$

where $H>0$. The evolution of $\kappa$ then follows from (11) as

$$\dot{\kappa} = -\mathring{\gamma} H_{\mathrm{eff}} \quad (<0) \tag{94}$$

where an effective modulus $H_{\mathrm{eff}} = Hhk\mathrm{e}^{hl_{\mathrm{ch}}\kappa}$ has been introduced. Since $\partial^2 f/\partial\kappa^2 < 0$ and $\partial^2 \psi^{\mathrm{p}}/\partial\kappa^2 < 0$ the finite-step mathematical program is non-convex in both the objective function and the inequality constraints. Such problems often contain multiple local optima which of course reflects the non-uniqueness of the corresponding continuous problem.

The material parameters are chosen as $k = 1.0\,(\mathrm{N\,m^{-2}})$, $h = 0.2\,(\mathrm{N^{-1}\,m})$, and $H = 1000\,(\mathrm{N\,m^{-1}})$. Young's modulus is $E = 1000\,(\mathrm{N\,m})^{-2}$ and Poisson's ratio is $\nu = 0.3$. This gives an effective softening modulus of $H_{\mathrm{eff}} = 200\mathrm{e}^{hl_{\mathrm{ch}}\kappa}\,(\mathrm{N\,m^{-2}})$. For a problem without imperfections or other features capable of inducing inhomogeneous stress states the acoustic tensor can be used to assess the inclination of localization bands characterizing the discontinuous bifurcations that may occur as a result of softening [80]. Thus, with reference to Figure 15(b), the determinant of the acoustic tensor predicts possible localization band inclinations of

$$\tfrac{1}{3}[1 + 4(H_{\mathrm{eff}}/E)^{1/2}] \leqslant \cos 2\theta \leqslant \tfrac{1}{3}[1 - 4(H_{\mathrm{eff}}/E)^{1/2}] \tag{95}$$

For $\kappa = 0$ this range is $10.8^\circ \leqslant \theta \leqslant 52.6^\circ$ whereas for $\kappa = -\infty$ the only possibility of localization is given by a band inclined at $\theta = 35.3^\circ$. The mesh shown in Figure 15(c) where the diagonals



DOI: 10.1002/nme

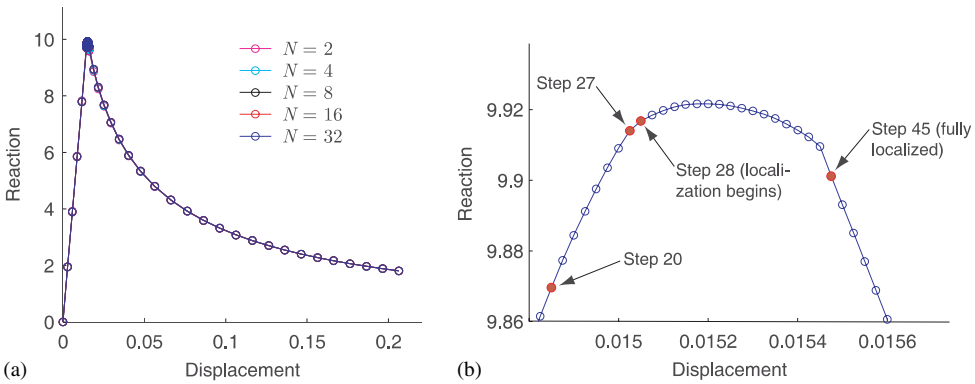


Figure 16. Load–displacement curves: (a) $N = 2, \ldots, 32$; and (b) $N = 32$.

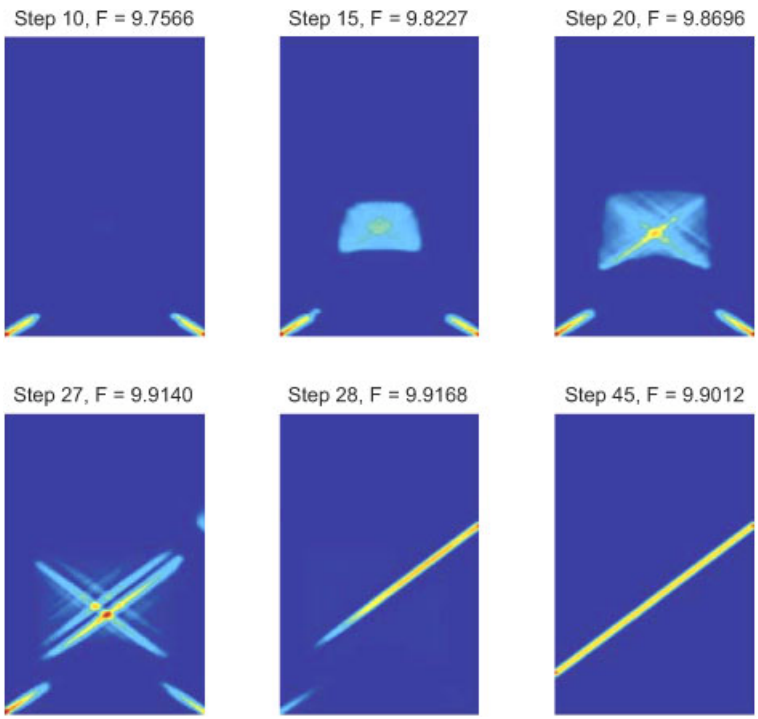


Figure 17. Evolution of equivalent plastic strain rate ($N = 32$).

are inclined at angles of $\beta = 36.9°$ with vertical should thus be capable of capturing the expected localization relatively well.

The problem is analysed for $N = 2$, 4, 8, 16, and 32 (32, 128, 512, 2048, and 8192 elements). The load–displacement curves are shown in Figure 16(a) with a zoom of the conditions around the peak shown in Figure 16(b). The load steps were adjusted in such a way that finer steps



DOI: 10.1002/nme

were used around the peak. The results appear to be quite insensitive to the mesh density which, however, primarily is a result of a favourable mesh arrangement and should not be taken as being representative of arbitrary meshes. From an algorithmic point of view the interesting feature is that there does not seem to be any significant problems with handling the considerable stress and strain redistributions that take place around the peak, cf. Figures 16(b) and 17, although this of course to some degree depends on the magnitude of the load step. Neither does the non-convex character of the problem imply any observable difference in performance as compared to that experienced for convex problems. For all analyses a total of 90 load steps were used with the average number of iterations per load step ranging from 3.2 for $N = 2$ to 4.1 for $N = 32$. In all cases an asymptotically quadratic rate of convergence was observed in all load steps.

## 6. CONCLUSIONS

An interior-point based algorithm for classical small-deformation rate-independent elastoplasticity has been presented. Compared to conventional implementations, several modifications have been made to exploit key features of common elastoplastic boundary value problems. The resulting algorithm is both efficient and robust and leads to a quadratic rate of convergence as the solution is approached. Furthermore, multisurface plasticity is dealt with in a straightforward manner and does not require any additional or specialized procedures. Finally, the mathematical programming framework presented appears to be convenient for implementing other finite element formulations than those based on standard displacement elements. This has been exemplified by the implementation of a mixed stress–displacement element which yields very accurate results, particularly regarding the ultimate load, and which, to our knowledge, has not previously been used for elastoplasticity.

## REFERENCES

1. Ortiz M, Popov EP. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *International Journal for Numerical Methods in Engineering* 1985; **21**:1561–1576.
2. Simo JC, Taylor RL. Consistent tangent operators for rate-independent elastoplasticity. *Computer Methods in Applied Mechanics and Engineering* 1985; **48**:101–118.
3. Borja RI. Cam-clay elastoplasticity, part I: Implicit integration of elasto-plastic relations. *Computer Methods in Applied Mechanics and Engineering* 1990; **78**:49–92.
4. Borja RI. Cam-clay elastoplasticity, part I: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor. *Computer Methods in Applied Mechanics and Engineering* 1991; **88**:225–240.
5. Feenstra PH, de Borst R. A composite plasticity model for concrete. *International Journal of Solids and Structures* 1996; **33**:707–730.
6. Jeremic B, Sture S. Implicit integrations in elastoplastic geotechnics. *Mechanics of Cohesive-Frictional Materials* 1997; **2**:165–183.
7. Lourenco PB, Rots JG. Multisurface interface model for analysis of masonry structures. *Journal of Engineering Mechanics* 1996; **123**:660–668.
8. de Borst R, Groen AE. Computational strategies for standard soil plasticity models. In *Modelling in Geomechanics*, Zaman M, Booker J, Gioda G (eds). Wiley: New York, 2000; 23–50.
9. Manzari MT, Prachathananukit R. On integration of a cyclic soil plasticity model. *International Journal for Numerical and Analytical Methods in Geomechanics* 2001; **25**:525–549.
10. Ahadi A, Krenk S. Implicit integration of plasticity models for granular materials. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:3471–3488.
11. Borja RI, Sama KM, Sanz PF. On the numerical integration of three-invariant elastoplastic constitutive models. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:1227–1258.



DOI: 10.1002/nme

12. Wang X, Wang LB, Xu LM. Formulation of the return mapping algorithm for elastoplastic soil models. *Computers and Geotechnics* 2004; **31**:315–338.
13. Perez-Foguet A, Rodriguez-Ferran A, Huerta A. Key issues in computational geomechanics. *Monograph M-58*. CIMNE, Barcelona, 2000.
14. Perez-Foguet A, Rodriguez-Ferran A, Huerta A. Consistent tangent matrices for substepping schemes. *Computer Methods in Applied Mechanics and Engineering* 2001; **53**:4627–4647.
15. Sloan SW. Substepping schemes for the numerical integration of elastoplastic stress–strain relations. *International Journal for Numerical Methods in Engineering* 1987; **24**:893–911.
16. Abbo AJ, Sloan SW. An automatic load stepping algorithm wiht automatic error control. *International Journal for Numerical Methods in Engineering* 1996; **39**:1737–1759.
17. Abbo AJ. *Finite Element Algorithms for Elastoplasticity and Consolidation*. *Ph.D. Thesis*, University of Newcastle, Newcastle, Australia, 1997.
18. Sloan SW, Booker JR. Integration of Tresca and Mohr–Coulomb constitutive relations in plane strain elastoplasticity. *International Journal for Numerical Methods in Engineering* 1992; **33**:163–196.
19. Sloan SW, Abbo AJ, Sheng D. Refined explicit integration of elastoplastic models with automatic error control. *Engineering Computations* 2001; **18**:121–154.
20. Sheng D, Sloan SW, Yu HS. Aspects of finite element implementation of critical state models. *Computational Mechanics* 2000; **26**:185–196.
21. Luccioni LX, Pestana JM, Taylor RL. Finite element implementation of non-linear elastoplastic constitutive laws using local and global explicit algorithms with automatic error control. *International Journal for Numerical Methods in Engineering* 2001; **50**:1191–1212.
22. Sheng D, Sloan SW, Abbo AJ. An automatic Newton–Raphson scheme. *International Journal of Geomechanics* 2002; **2**:471–502.
23. Maier G. Quadratic programming theory for elastic perfectly plastic structures. *Meccanica* 1968; **3**:31–39.
24. Maier G. A quadratic programming approach for certain classes on nonlinear structural problems. *Meccanica* 1968; **3**:121–130.
25. Capurso M, Maier G. Incremental elastoplastic analysis and quadratic optimization. *Meccanica* 1970; **5**:107–116.
26. Xiaoming G, Roulei Z, Yinghe S. On the mathematical modeling for elastoplastic contact problem and its solution by quadratic programming. *International Journal of Solids and Structures* 2001; **38**:8133–8150.
27. Zhang HW, Xu WL, Di SL, Thomson PF. Quadratic programming method in numerical simulation of metal forming process. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:5555–5578.
28. Liew KM, Wu YC, Zou GP, Ng TY. Elasto-plasticity revisited: numerical analysis via reproducing kernel particle method and parametric quadratic programming. *International Journal for Numerical Methods in Engineering* 2002; **55**:669–683.
29. Su RKL, Zhu J, Leung AYT. Parametric quadratic programming method for elastic contact fracture analysis. *International Journal of Fracture* 2002; **117**:143–157.
30. Hjiaj M, Fortin J, de Saxce G. A complete stress update algorithm for the non-associated Drucker–Prager model including treatment of the apex. *International Journal of Engineering Science* 2003; **41**:1109–1143.
31. Tin-Loi F. An iterative complementarity approach for elastoplastic analysis involving frictional contact. *International Journal of Mechanical Sciences* 2003; **45**:197–216.
32. Contrafatto L, Ventura G. Numerical analysis of augmented Lagrangian algorithms in complementary elastoplasticity. *International Journal for Numerical Methods in Engineering* 2004; **60**:2263–2287.
33. Pastor J. Analyse limite: determination de solutions statiques completes—application au talus vertical. *European Journal of Mechanics – A/Solids* 1978; **2**:176–196.
34. Bottero A, Negre R, Pastor J, Turgeman S. Finite element method and limit analysis theory for soil mechanics problems. *Computer Methods in Applied Mechanics and Engineering* 1980; **22**:131–149.
35. Sloan SW. Lower bound limit analysis using finite elements and linear programming. *International Journal for Numerical and Analytical Methods in Geomechanics* 1988; **12**:61–77.
36. Sloan SW. Upper bound limit analysis using finite elements and linear programming. *International Journal for Numerical and Analytical Methods in Geomechanics* 1989; **13**:263–282.
37. Sloan SW, Kleeman PW. Upper bound limit analysis using discontinuous velocity fields. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:293–314.
38. Lyamin AV, Sloan SW. Lower bound limit analysis using non-linear programming. *International Journal for Numerical Methods in Engineering* 2002; **55**:573–611.



DOI: 10.1002/nme

39. Lyamin AV, Sloan SW. Upper bound limit analysis using linear finite elements and non-linear programming. *International Journal for Numerical and Analytical Methods in Geomechanics* 2002; **26**:181–216.
40. Krabbenhoft K, Damkilde L. Lower bound limit analysis of slabs with nonlinear yield criteria. *Computers and Structures* 2002; **80**:2043–2057.
41. Karmarkar N. A new polynomial-time algorithm for linear programming. *Combinatorica* 1984; **4**:373–395.
42. Nash SG, Sofer A. *Linear and Nonlinear Programming*. McGraw-Hill: New York, 1996.
43. Wright SJ. *Primal–Dual Interior-Point Methods*. SIAM: Philadelphia, 1997.
44. Vanderbei RJ. *Linear Programming*: *Foundations and Extensions*. Springer-Verlag: New York, 2001.
45. Wright MH. The interior-point revolution in optimization: history, recent developments and lasting consequences. *Bulletin of the American Mathematical Society* 2004; **42**:39–56.
46. Andersen KD, Christiansen E. Limit analysis with the dual affine scaling algorithm. *Journal of computational and Applied Mathematics* 1995; **59**:233–243.
47. Christiansen E, Andersen KD. Computation of collapse loads with von Mises type yield condition. *International Journal for Numerical Methods in Engineering* 1999; **45**:1185–1202.
48. Krabbenhoft K, Damkilde L. A general nonlinear optimization algorithm for lower bound limit analysis. *International Journal for Numerical Methods in Engineering* 2003; **56**:165–184.
49. Pastor J, Thai TH, Francescato P. Interior point optimization and limit analysis: an application. *Communications in Numerical Methods in Engineering* 2003; **19**:769–786.
50. Pastor F, Loute E. Solving limit analysis problems: an interior-point method. *Communications in Numerical Methods in Engineering* 2005; **21**:631–642.
51. Simo JC, Kennedy JG, Taylor RL. Complementary mixed finite element formulations for elastoplasticity. *Computer Methods in Applied Mechanics and Engineering* 1989; **74**:177–206.
52. Fiacco AV, McCormick GP. *Nonlinear Programming*: *Sequential Unconstrained Minimization Techniques*. SIAM: Philadelphia, 1968.
53. Perzyna P. Fundamental problems in viscoplasticity. *Advances in Applied Mechanics* 1966; **9**:244–378.
54. Perzyna P. Thermodynamic theory of viscoplasticity. *Advances in Applied Mechanics* 1971; **11**:313–354.
55. Simo JC, Hughes TJR. *Computational Inelasticity*. Springer-Verlag: New York, 1998.
56. Lasserre JB. Why the logarithmic barrier function in convex and linear programming? *Operations Research Letters* 2000; **27**:149–152.
57. Collins IF, Houlsby G. Application of thermomechanical principles to the modelling of geotechnical materials. *Proceedings of the Royal Society of London*, *Series A* 1997; **453**:1975–2000.
58. Zouain N, Herskovits J, Borges LA, Feijóo RA. An iterative algorithm for limit analysis with nonlinear yield functions. *International Journal of Solids and Structures* 1993; **30**(10):1397–1417.
59. Borges LA, Zouain N, Huespe AE. A nonlinear optimization procedure for limit analysis. *European Journal of Mechanics – A/Solids* 1996; **15**(3):487–512.
60. Pontes IDS, Borges LA, Zouain N, Lopes FR. An approach to limit analysis with cone-shaped yield surfaces. *International Journal for Numerical Methods in Engineering* 1997; **40**:4011–4032.
61. Zouain N, Borges L, Silveira JL. An algorithm for shakedown analysis with nonlinear yield functions. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:2463–2481.
62. Borges LA, Zouain N, Costa C, Feijoo R. An adaptive approach to limit analysis. *International Journal of Solids and Structures* 2001; **38**:1707–1720.
63. Zienkiewicz OC, Taylor RL. *The Finite Element Method*. McGraw-Hill: New York, 1989.
64. Gill PE, Murray W, Saunders MA, Wright MH. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Technical Report SOL 85-11*, Stanford University Systems Optimization Laboratory, 1985.
65. Lustig IJ. Feasibility issues in a primal–dual interior-point method for linear programming. *Mathematical Programming* 1990; **49**:145–162.
66. Vanderbei RJ, Shanno DF. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications* 1999; **13**:231–252.
67. Krabbenhoft K, Damkilde L, Krabbenhoft S. Ultimate limit state design of sheet pile walls by finite elements and nonlinear programming. *Computers and Structures* 2005; **83**:383–393.
68. Fletcher R. *Practical Methods of Optimization*. Wiley: New York, 1987.
69. Alfano G, Rosati L, Valoroso N. A displacement-like finite element model for $J_2$ plasticity. *Computer Methods in Applied Mechanics and Engineering* 1998; **74**:177–206.
70. Gilbert M, Tyas A. Layout optimization of large-scale pin-jointed frames. *Engineering Computations* 2003; **8**:1044–1064.




DOI: 10.1002/nme

71. Bisbos CD, Makrodimopoulos A, Pardalos PM. Second-order cone programming approaches to static shakedown analysis in steel plasticity. *Optimization Methods and Software* 2005; **20**:25–52.
72. Makrodimopoulos A, Martin CM. Limit analysis using large-scale SOCP optimization. In *Proceedings of the 13th National Conference of UK Association for Computational Mechanics in Engineering*, Sheffield, 2005; 21–24.
73. Abbo AJ, Sloan SW. A smoothed hyperbolic approximation to the Mohr–Coulomb yield criterion. *Computers and Structures* 1995; **54**:427–441.
74. Sandler IS, DiMaggio FL, Baladi GY. Generalized cap model for geological materials. *Journal of the Geotechnical Engineering Division* 1976; **102**:683–699.
75. Resende L, Martin JB. Formulation of Drucker–Prager cap model. *Journal of Engineering Mechanics* 1985; **111**:855–881.
76. Hofstetter G, Simo JC, Taylor RL. A modified cap model: closest point solution algorithm. *Computers and Structures* 1993; **46**:203–214.
77. Sture S, Runesson K, Macari-Pasqualino EJ. Analysis and calibration of a three-invariant plasticity model for granular materials. *Ingenieur Archiv* 1989; **59**:253–266.
78. Jeremic B, Sture S. Implicit integration rules in plasticity: theory and implementation. *Technical Report*, University of Colorado, Boulder, 1994.
79. Cervera M, Chiumenti M, Agelet de Saracibar C. Softening, localization and stabilization: capture of discontinuous solutions in $J_2$ plasticity. *International Journal for Numerical and Analytical Methods in Geomechanics* 2004; **28**:373–393.
80. Bigoni D. Bifurcation and instability of non-associative elastoplastic solids. In *Material Instabilities in Elastic and Plastic Solids*, Petryk H (ed.). CISM Courses and Lectures No. 414. Springer-Verlag: New York, 2000.



DOI: 10.1002/nme