Christian Carow, Frank Rackwitz

Comparison of implicit and explicit numerical integration schemes for a bounding surface soil model without elastic range

Comparison of implicit and explicit numerical integration schemes for a bounding surface soil model without elastic range

Christian Carow*, Frank Rackwitz

Technische Universität Berlin, Chair of Soil Mechanics and Geotechnical Engineering, Straße des 17. Juni 135, 10623 Berlin, Germany

Abstract

Standard integration schemes for rate constitutive equations were designed within the classical theory of plasticity. Consequently, they rely on the assumption that a yield criterion defines a range of purely elastic material behaviour. Many constitutive models for non-cohesive soils discard that assumption. They account for inelastic deformations without employing a yield criterion. This simplifies the formulation of the models but raises questions concerning their numerical implementation. In particular, it is not fully clear which algorithmic method is most appropriate for the numerical integration of constitutive models without elastic range. To investigate this, two different stress point algorithms for a critical state bounding surface model for sands were developed and implemented. The explicit method employs substepping to automatically control the local error. The implicit method updates stresses and state variables through a local Newton iteration, the Jacobian of which is computed by numerical differentiation. The two algorithms were compared by means of calculations at integration point level as well as with respect to a boundary value problem. The results show that for a given level of accuracy, the explicit update procedure is significantly more efficient than the implicit one. This holds regardless of initial state parameters and input strain increment magnitudes.

Keywords: Constitutive models, Non-Cohesive soils, Bounding Surface Plasticity, Implicit Numerical Integration, Explicit Numerical Integration, Numerical Differentiation

1. Introduction

For constitutive models which employ the classical theory of plasticity, a yield criterion must be defined. It describes a domain in the space of stresses and state variables where the material under consideration behaves purely elastic. This domain is briefly referred to as *elastic range*.

In contrast to that, non-cohesive soils behave inelastically even if strains are very small. Consequently, some models for such soils do not exhibit yield surfaces. Instead, the frameworks of Generalized Plasticity (Pastor et al., 1990) and Bounding Surface Plasticity (Dafalias, 1986) are frequently used (e. g. Li, 2002; Khalili et al., 2005; Andrianopoulos et al., 2010; Sadeghian and Namin, 2013). A special kind of bounding surface theories without elastic range was termed *hypoplastic* by Dafalias (1986). The characteristic feature of these models are stress rate dependent mapping rules, rendering them incrementally non-linear. The concept was adopted in (Wang et al., 1990; Dafalias and Taiebat, 2016), for example. Apart from that, Kolymbas (1991) established the name *Hypoplasticity* for a class of soil models without yield criteria which were mostly developed at the University of Karlsruhe (e. g. von Wolffersdorff, 1996; Niemunis and Herle, 1997). They abandon the concept of elastic-plastic strain decomposition completely and are incrementally non-linear in the strain rate. Constitutive relations of this type are called *Karlsruhe Hypoplasticity* here.

All constitutive theories mentioned above are formulated by means of rate equations. In order to be used for the solution of initial boundary value problems, such differential equations have to be integrated over time increments of finite size. This can be accomplished by either implicit or explicit numerical methods. Explicit integrators are straightforward to construct and sufficiently accurate if time steps are small. Implicit integrators, on the other hand, are supposed to be stable even for large time steps, but require a local iteration if non-linear functions are to be handled. That renders the implementation rather cumbersome and may induce considerable computational overhead (Neto et al., 2008).

Common integration algorithms were developed with respect to the classical theory of plasticity. Thus, the algorithms' key point is to enforce the yield criteria of the underlying models. Such methods have to be adapted if no elastic range is defined. This has been thouroughly investigated with respect to Karlsruhe Hypoplasticity by comparing different types of integration schemes (Tamagnini et al., 2000; Fellin et al., 2009; Ding et al., 2015; Wang et al., 2018).

*Corresponding author

Email address: christian.carow@tu-berlin.de (Christian Carow)

For elasto-plastic models without yield surfaces, implicit algorithms (de Borst and Heeres, 2002; Mira et al., 2009; Petalas and Dafalias, 2019) as well as explicit ones (Andrianopoulos et al., 2010; Shu et al., 2011; Kan and Taiebat, 2014; Shi et al., 2018) were adopted. However, each of these references was confined to either explicit or implicit methods and thus did not compare the performance of different classes of integrators directly.

Therefore, one question has been left unanswered by previous publications: *Which method is most appropriate for the numerical implementation of bounding surface plasticity without elastic range?* This issue is investigated here through numerical experiments with the critical state bounding surface model for sands proposed by Li (2002).

It is important to note beforehand that a radial mapping rule is employed to define the bounding surface image stress, as will be discussed further in Section 2.4. This approach seems to be quite popular, since it has been adopted extensively in soil mechanics (e. g. by Andrianopoulos et al., 2010; Kan and Taiebat, 2014; Wang and Xie, 2014; Shi et al., 2018). Still, there are bounding surface models which use different mapping rules. Major implications for the numerical implementation may result from that, which cannot be covered by the work presented here. This is particularly true for hypoplastic mappings in the sense of Dafalias (1986), because they require knowledge of the stress rate direction, which is not available *a priori* during numerical stress integration. Please refer to Petalas and Dafalias (2019) for further insights into that special subject.

The model by Li (2002) is briefly introduced in the following section. Afterwards, an implicit as well as an explicit integration scheme are presented. Then, the two methods are verified, tested and compared comprehensively by means of example calculations.

2. Theory

2.1. Preliminary remarks

The usual conventions of soil mechanics are adopted. The effective stress $\boldsymbol{\sigma}$ and the infinitesimal strain $\boldsymbol{\varepsilon}$ are assumed to be positive in compression. The coordinate frame is three dimensional and Cartesian. Tensors, vectors and matrices are typeset boldface or are referred to by their components, e. g. tensor $\boldsymbol{\sigma}$ has components σ_{ij}. Quantities referring to a bounding surface are labelled by a superimposed bar, e. g. $\bar{\boldsymbol{r}}$ is the bounding surface image of $\boldsymbol{r}$. Summation over repeated indices in products is implied, e. g. $a_i v_i = a_1 v_1 + a_2 v_2 + a_3 v_3$.

Stresses, strains and state variables are functions of their location x in the ambient space and of the pseudo-time t. All processes considered here are rate-independent, therefore t has no physical meaning, but is used to label loading stages. The dependency on t is indicated by subscripts. The dependency on x is hidden in all equations because stress integration takes place at discrete points with x fixed. That means for example $\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}(t_{n+1}, x)$.

Formally, deformations are assumed to be small, allowing to employ a linearised relationship between displacements and strains. Nevertheless, the theory and the algorithms presented here may be transferred into a finite deformation context without structural modifications. This can be accomplished by using a numerical method for the global problem which preserves incremental objectivity for the integration of stresses and state variables. Most major finite element codes provide that. The theoretical background is comprehensively described in (Simo and Hughes, 1998, Chapters 7, 8) and more concisely in (Aubram, 2017).

The subsequent description of the model by Li (2002) focuses on basic concepts and on the major rate equations, the numerical integration of which is discussed in Section 3. The set of constitutive equations is completed by the contents of Appendix B.

See Appendix A for a list of operators and symbols which are not defined in the text.

2.2. Elastic and plastic mechanisms

The constitutive model for sands by Li (2002) is based on the additive decomposition of the strain rate tensor

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \,. \tag{1}$$

Soil behaviour is assumed to be instantaneously inelastic, i. e. the elastic part $\dot{\boldsymbol{\varepsilon}}^{\mathrm{e}}$ and the plastic part $\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}$ always act simultaneously.

The rates of strains and stresses are linked by the usual elastic law

$$\dot{\boldsymbol{\sigma}} = \mathbf{c}^{\mathrm{e}}(e,p) : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}) \tag{2}$$

The elastic stiffness $\mathbf{c}^{\mathrm{e}}$ is a non-linear function of void ratio e and mean effective stress p through the empirical formula for the shear modulus of Richart et al. (1970), see (B.2).

For the plastic strains $\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}$, the bounding surface framework of Wang et al. (1990) is adopted. Since non-cohesive soils are particularly sensitive to changes in stress ratio $\boldsymbol{r} = \boldsymbol{\sigma}_{\mathrm{dev}}/p$, the stress rate is decomposed by the chain rule of differentiation into

$$\dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{r}} p + \dot{p}\frac{\boldsymbol{\sigma}}{p} \,. \tag{3}$$

The two rate quantities in (3) are being accounted for by separate plastic mechanisms. Each of them computes a plastic strain rate by means of its own bounding surface. These surfaces are described in Section 2.3.

The contributions of the two plastic mechanisms are added up to

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} &= \dot{\boldsymbol{\varepsilon}}^{\mathrm{p1}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{p2}} \\ &= \dot{\lambda}_1 \left(\bar{\boldsymbol{n}} + \sqrt{\frac{2}{27}} D_1 \boldsymbol{I} \right) + \dot{\lambda}_2 \left(\bar{\boldsymbol{m}} + \sqrt{\frac{2}{27}} D_2 \boldsymbol{I} \right) \,. \end{aligned} \tag{4}$$

According to this equation, the deviatoric plastic flow is directed along the unit tensors $\bar{\boldsymbol{n}}$ and $\bar{\boldsymbol{m}}$, while its magnitude is governed by the plastic loading indices $\dot{\lambda}_1$ and $\dot{\lambda}_2$. The volumetric plastic strains are scaled against the loading indices by the dilatancy functions D_1 and D_2.

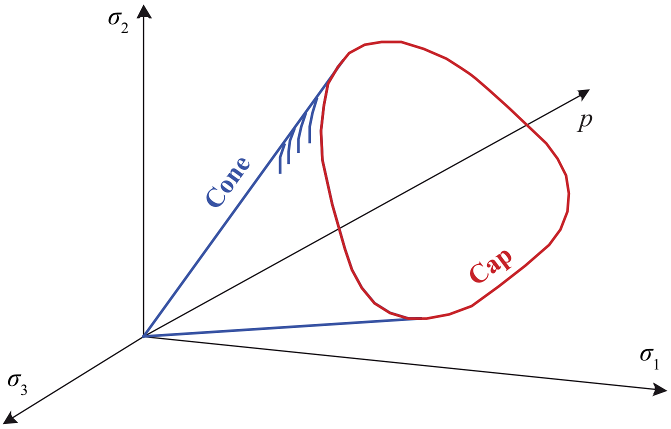


Figure 1: Illustration of bounding surfaces in principal stress space; adapted with permission from (Carow et al., 2017, p. 606); Copyright (2017) Wiley

2.3. Bounding surfaces

Plastic mechanism $\dot{\varepsilon}^{\mathrm{p}1}$ accounts for the stress ratio part of (3) and relies on the bounding surface

$$F_1 = \frac{\bar{R}}{g(\bar{\theta})} - H_1 = 0 \,. \tag{5}$$

The surface is a cone in effective stress space. Its shape, as sketched in Figure 1, is prescribed by the function $g(\bar{\theta})$ and its size is defined by the internal state variable H_1. The quantities $\bar{R}$ and $\bar{\theta}$ in (5) are invariants of the image stress ratio $\bar{\boldsymbol{r}}$, as defined in Appendix A. To find $\bar{\boldsymbol{r}}$, the current stress ratio $\boldsymbol{r}$ is projected onto F_1 using the method described in Section 2.4.

The hardening rule of the cone is

$$\dot{H}_1 = \frac{A}{p} \bar{K}_{\mathrm{p}1} \dot{\lambda}_1 \,, \tag{6}$$

in which $\bar{K}_{\mathrm{p}1}$ is the cone's plastic modulus and A is the magnitude of the outward normal to F_1 at $\bar{\boldsymbol{r}}$, see (B.10).

Plastic mechanism $\dot{\varepsilon}^{\mathrm{p}2}$ accounts for the second term of (3). Its bounding surface

$$F_2 = \bar{p} - H_2 = 0 \tag{7}$$

is a flat cap perpendicular to the hydrostatic axis, as shown in Figure 1. The image mean stress $\bar{p}$ is defined by a one-dimensional mapping rule as described in Section 2.4. The evolution of the hardening parameter H_2 is linked to the cap's loading index and the plastic modulus $\bar{K}_{\mathrm{p}2}$ by

$$\dot{H}_2 = \bar{K}_{\mathrm{p}2} \dot{\lambda}_2 \,. \tag{8}$$

2.4. Mapping rules

In order to calculate the loading indices $\dot{\lambda}_1$ and $\dot{\lambda}_2$, plastic moduli $K_{\mathrm{p}1}$ and $K_{\mathrm{p}2}$ at the current stress point are required. The corresponding formulas are listed here in Appendix B. They differ from their bounding surface counterparts $\bar{K}_{\mathrm{p}1}$ and $\bar{K}_{\mathrm{p}2}$ by scaling factors, which take into account the scalar stress distances

$$\begin{aligned} \bar{\rho}_1 &= \|\bar{\boldsymbol{r}} - \boldsymbol{\alpha}\| \,, \quad \rho_1 = \|\boldsymbol{r} - \boldsymbol{\alpha}\| \,, \\ \bar{\rho}_2 &= |\bar{p} - \beta| \,, \quad \rho_2 = |p - \beta| \,. \end{aligned} \tag{9}$$

The quantities $\boldsymbol{\alpha}$ and β in (9) are projections centres, located at the points of last stress reversal. They are known from the loading history. The stress images $\bar{\boldsymbol{r}}$ and $\bar{p}$ are obtained by means of mapping rules as follows.

To define the image stress ratio $\bar{\boldsymbol{r}}$, Li (2002) adopted a radial mapping rule of the form

$$\bar{\boldsymbol{r}} = \boldsymbol{\alpha} + \frac{\bar{\rho}_1}{\rho_1} (\boldsymbol{r} - \boldsymbol{\alpha}) \,. \tag{10}$$

The distance ρ_1 can directly be computed through (9). That leaves $\bar{\boldsymbol{r}}$ and $\bar{\rho}_1$ as unknowns of (10). Unfortunately, the equation cannot be solved analytically. Instead, a first estimate for the unknowns is computed by assuming the shape of the cone to be circular, implying $g(\bar{\theta}) = 1$. Then, the solution is improved by iteratively enforcing (5) and (10).

The image stress $\bar{p}$ of the cap bounding surface is much easier to find, because the projection is along the hydrostatic axis only. Li (2002) simply defined

$$\bar{p} = \begin{cases} H_2 & \text{if } \dot{p} > 0 \\ 0 & \text{if } \dot{p} < 0 \end{cases} \,. \tag{11}$$

2.5. State dependend plastic parameters

The model features presented so far constitute a comprehensive bounding surface framework without elastic range. The main novelty which Li (2002) added to this platform are functions for the dilatancy D_1 and the plastic moduli $\bar{K}_{\mathrm{p}1}$ and $K_{\mathrm{p}1}$ that depend on the material internal state. The latter is described by the weighted invariants of $\bar{\boldsymbol{r}}$ and $\boldsymbol{r}$

$$\bar{\eta} = \frac{\bar{R}}{g(\bar{\theta})} \,, \qquad \eta = \frac{R}{g(\theta)} \,, \tag{12}$$

and by the state parameter of Been and Jefferies (1985)

$$\psi = e - e_{\mathrm{c}} \,. \tag{13}$$

For the critical void ratio e_{c} in (13), the equation

$$e_{\mathrm{c}} = e_{\Gamma} - \lambda_{\mathrm{c}} \, (p/p_{\mathrm{a}})^{\xi} \tag{14}$$

is adopted from Li and Wang (1998). Therein, p_{a} is the atmospheric pressure and e_{Γ}, λ_{c} as well as ξ are constitutive parameters, the meaning of which is depicted in Figure 2. Earlier approaches assumed a linear relationship between e_{c} and $\ln p$. Compared to that, the power law (14) fits experimental data for non-cohesive soils across a greater range of pressures.

The idea of modelling the behaviour of non-cohesive soils by means of the state parameter ψ was developed and gradually refined by a series of pioneering works. Jefferies (1993) and Muir Wood et al. (1994) were the first to include a dependence on ψ in the hardening rules of plasticity models. Manzari and Dafalias (1997) adopted this approach. They furthermore expressed the phase transformation stress ratio at which the soil response changes from contractive to dilative as a linear function of ψ. Thus, both hardening and dilatancy became directly related to ψ. The concept was taken up by Li and Dafalias (2000). They combined it with the power law (14) and developed a model for triaxial compression in which the hardening modulus and the dilatancy are exponential functions of ψ. These functions were then adapted for general loading conditions by Li (2002).

The actual equations for D_1, $\bar{K}_{\mathrm{p1}}$ and K_{p1} are provided here in Appendix B. For a detailed discussion, please refer to (Li and Dafalias, 2000; Li, 2002). In essence, their dependency on the material internal state enables the model to capture the following crucial aspects of the mechanical behaviour of non-cohesive soils during shear loading:

- The soil response depends on whether the initial state is to be characterized as *dense* or *loose*.
- Dense states and loose states cannot be distinguished by density or void ratio e only. In addition, the mean effective stress p must be taken into account, as illustrated in Figure 2.
- For dense states, the soil contracts initially, then undergoes phase transformation and finally dilates towards e_{c}. Meanwhile, the stresses reach a sharp peak, followed by softening towards the residual strength.
- For loose states, the soil continuously contracts towards e_{c} and the stresses evolve without a distinctive peak.

The above explanations refer to drained conditions. Nevertheless, the model is also able to simulate the excess pore pressure evolution during undrained loading with the same set of equations and constitutive parameters.

2.6. Overall structure

The elastic and plastic constitutive equations are combined into the tensors $\boldsymbol{\Theta}$ and $\boldsymbol{Z}$, which are defined in (B.5) and (B.6). These are used to calculate the plastic loading indices from the strain rate according to

$$\dot{\lambda}_1 = \boldsymbol{\Theta} : \dot{\boldsymbol{\varepsilon}}\,, \qquad \dot{\lambda}_2 = \boldsymbol{Z} : \dot{\boldsymbol{\varepsilon}}\,. \tag{15}$$

Subsequently, a rate evolution equation for the effective stress can be derived, the general shape of which is

$$\dot{\boldsymbol{\sigma}} = \mathbf{c}^{\mathrm{ep}}(\boldsymbol{\sigma}, \boldsymbol{q}) : \dot{\boldsymbol{\varepsilon}}\,. \tag{16}$$

The elastoplastic stiffness $\mathbf{c}^{\mathrm{ep}}$ is given by (B.4). It is a function of stress $\boldsymbol{\sigma}$ and internal state variables $\boldsymbol{q}$. The component matrix of $\boldsymbol{q}$ is

$$\boldsymbol{q} = \begin{bmatrix} e & \lambda_1 & H_1 & H_2 & \boldsymbol{\alpha} & \beta \end{bmatrix}^{\mathrm{T}}\,. \tag{17}$$

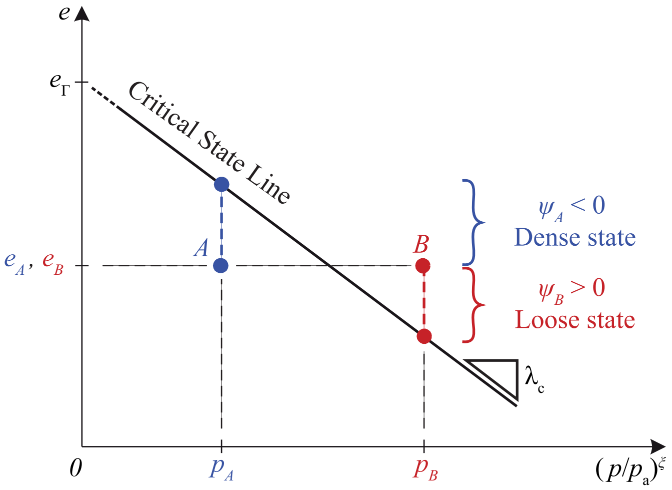


Figure 2: Illustration of Critical State Line and state parameter ψ; adapted with permission from (Carow et al., 2017, p. 607); Copyright (2017) Wiley

2.7. Description of model operation

2.7.1. Monotonic loading

When the model by Li (2002) is used to simulate monotonic loading of a virgin soil sample, the bounding surfaces expand isotropically while the current stress point resides on them. At this stage, the bounding surface projection centres are defined through the initial conditions. As noted before, there is no elastic range, so the simulated soil behaviour is elastic-plastic even for the very first loading increment. Only in case of a neutral loading path along the cone bounding surface the model response would be purely elastic because $\dot{\lambda}_1 = \dot{\lambda}_2 = 0$. However, according to Li (2002) this situation is “highly hypothetical”.

If softening occurs, for example in drained triaxial compression of dense sand, the two bounding surfaces evolve differently, as defined by their plastic moduli $\bar{K}_{\mathrm{p1}}$ and $\bar{K}_{\mathrm{p2}}$ (see Appendix B):

- The cone shrinks because $\bar{K}_{\mathrm{p1}}$ becomes negative. Meanwhile, the stress point remains on the cone.
- The cap stays at the position $H_2 = p_{\max}$ as long as $p < p_{\max}$, in which $p_{\max}$ is the maximum mean effective stress that has been experienced.

2.7.2. Reversal of loading direction

According to Li (2002) the loading direction of the cone is said to reverse if $\dot{\lambda}_1 < 0$ is encountered. In that case, the cone's projection centre $\boldsymbol{\alpha}$ is relocated onto the current stress ratio $\boldsymbol{r}$. Consequently, $\rho_1 = \|\boldsymbol{r} - \boldsymbol{\alpha}\| = 0$ and thus K_{p1} according to (B.16) becomes infinitely large, meaning that no plastic strains are produced by the cone. Due to the mechanics of the bounding surface framework, this situation is only instantaneous, as Li (2002) explains in detail. Immediately afterwards, $\dot{\lambda}_1$ becomes positive again.

The loading direction of the cap can be monitored through the sign of the mean effective stress rate. As long as $\operatorname{sgn}\dot{p} = \text{const}$, plastic strains are continuously produced. When $\operatorname{sgn}\dot{p}$ changes, the projection centre β jumps to the current p, while the cap is momentarily inactive. As with the cone, this situation lasts for an instant only. For any further changes of p in the same direction, the loading index becomes positive again, no matter if $\operatorname{sgn}\dot{p}$ has changed from positive to negative or vice versa.

2.7.3. Cyclic un- and reloading

Consider a situation in which the loading directions of both bounding surfaces have reversed after monotonic loading. Then, continued loading in the new directions induces the following processes.

1) The stress point detaches from the bounding surfaces.

2) The plastic strain magnitude becomes a function of the distances defined in (9).

3) The plastic modulus K_{p1} becomes a function of the accumulated plastic loading index

$$\lambda_1(t) = \lambda_1(t_0) + \int_{t_0}^{t} \dot{\lambda}_1(\tau)\mathrm{d}\tau\,. \tag{18}$$

These processes are automatically controlled through the bounding surface framework. Thus, different stages of cyclic loading that could be classified as “first loading”, “unloading” and “reloading” are addressed with a single set of constitutive equations. The only case which needs special treatment is the instantaneous loading direction reversal described in Section 2.7.2. This enables the model to capture effects of stress and strain history on the soil's behaviour in cyclic loading very efficiently.

Additionally, the model is able to simulate effects of non-proportional loading and fabric anisotropy if further state variables and plastic mechanisms are incorporated (Li and Dafalias, 2004; Gao and Zhao, 2015). This is omitted here, because it complicates the analytical formulation considerably, whereas the paper focuses on numerical implementation.

3. Numerical implementation

3.1. Algorithmic setup

Constitutive equations for path-dependent materials usually are of the rate-type and thus provide a relationship between infinitesimal increments of stresses and strains. This also holds for the model by Li (2002). As has already been noted in Section 1, such equations must be integrated over finite time or strain increments to obtain stress point algorithms. The general setup for that is defined subsequently.

Let $[t_n, t_{n+1}]$ be a given time interval. For $t = t_n$, effective stress $\boldsymbol{\sigma}_n$ and internal state variables $\boldsymbol{q}_n$ are known as initial conditions. The evolution of $\boldsymbol{\sigma}$ and $\boldsymbol{q}$ during the time increment $\Delta t = t_{n+1} - t_n$ is governed by a set of rate equations of the general form

$$\begin{aligned} \dot{\boldsymbol{\sigma}} &= g(\boldsymbol{\sigma}, \boldsymbol{q}, \dot{\boldsymbol{\varepsilon}})\,, \\ \dot{\boldsymbol{q}} &= h(\boldsymbol{\sigma}, \boldsymbol{q}, \dot{\boldsymbol{\varepsilon}})\,, \end{aligned} \tag{19}$$

in which dependency of all variables on time t has been suppressed. The symbols g and h represent generic constitutive functions. The actual evolution equations that Li (2002) proposed to use in place of g and h are summarised here in Table 1.

The strain rate $\dot{\boldsymbol{\varepsilon}}$ is given and constant for $t \in [t_n, t_{n+1}]$. Therefore, the update of stresses and state variables from t_n to t_{n+1} is solely driven by the strain increment

$$\Delta\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n = \dot{\boldsymbol{\varepsilon}}\Delta t\,. \tag{20}$$

Based on these assumptions, the problem to be solved by any stress point algorithm is: Find stresses and state variables at $t = t_{n+1}$ through numerical approximation of

$$\begin{aligned} \boldsymbol{\sigma}_{n+1} &= \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}\,, \quad \Delta\boldsymbol{\sigma} := \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{\sigma}}(t)\mathrm{d}t\,, \\ \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + \Delta\boldsymbol{q}\,, \quad \Delta\boldsymbol{q} := \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{q}}(t)\mathrm{d}t\,. \end{aligned} \tag{21}$$

Numerous methods for the solution of (21) were developed. Explicit methods approximate the integrals based on known quantities, usually the ones at $t = t_n$. Implicit methods evaluate the rate equations for $t = t_{n+1}$ at the end of the time increment (Simo and Hughes, 1998; Neto et al., 2008). Here, one algorithm out of each of the two categories is constructed.

Table 1: Evolution equations for stress and internal state variables

Var.	Description	Evolution	Eq.
$\boldsymbol{\sigma}$	Effective stress	$\dot{\boldsymbol{\sigma}} = \mathbf{c}^{\mathrm{ep}} : \dot{\boldsymbol{\varepsilon}}$	(16)
e	Void ratio	$-\dot{e} = (1+e)\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}})$	(B.8)
λ_1	Accum. pl. strain	$\dot{\lambda}_1 = \boldsymbol{\Theta} : \dot{\boldsymbol{\varepsilon}}$	(15)
H_1	Size of cone	$\dot{H}_1 = (A/p)\bar{K}_{\mathrm{p1}}\dot{\lambda}_1$	(6)
H_2	Position of cap	$\dot{H}_2 = \bar{K}_{\mathrm{p2}}\dot{\lambda}_2$	(8)
$\boldsymbol{\alpha}$	Projection centre	See Section 2.7.2	
β	Projection centre	See Section 2.7.2	

3.2. Implicit stress point algorithm

3.2.1. Backward Euler update

First of all, the implicit Backward Euler method is adopted for the numerical solution of the general problem (21). Its fundamental idea is to compute the increments of stresses and internal state variables by

$$\begin{aligned} \Delta\boldsymbol{\sigma} &= g(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \dot{\boldsymbol{\varepsilon}})\Delta t\,, \\ \Delta\boldsymbol{q} &= h(\boldsymbol{\sigma}_{n+1}, \boldsymbol{q}_{n+1}, \dot{\boldsymbol{\varepsilon}})\Delta t\,. \end{aligned} \tag{22}$$

That means, the constitutive functions g and h are evaluated for the updated configuration of stresses and internal state variables, which is unknown in advance. Please note that usually neither $\dot{\boldsymbol{\varepsilon}}$ nor Δt are considered in rate-independent plasticity. Instead, only their product $\Delta\boldsymbol{\varepsilon}$ is taken into account by the stress point algorithm.

In conventional plasticity, the basic equations (22) are further constrained because the yield criterion is to be obeyed at t_{n+1}. This is naturally accounted for by implicit return mapping procedures. In contrast, explicit time integration is prone to give numerical solutions which violate the yield criterion at t_{n+1}. This can be avoided by techniques like for example "consistent correction" of stresses and state variables (Sloan et al., 2001) or "event location and solving the corresponding index 2 problem“ (Fellin and Ostermann, 2014). However, such enhancements complicate the implementation of explicit integrators and increase their computational costs.

For the class of models considered here, there is no yield criterion to obey, which eliminates the advantage of implicit time integration described above. Still, the Backward Euler method should be more stable than explicit integration (Belytschko et al., 2014). The test calculations presented in Section 4 will show whether this is of significance here.

To derive an algorithm from the basic concept (22), the approach of de Borst and Heeres (2002) is adopted. Consequently, the Backward Euler stress increment is computed from the elastic stress-strain relation (2) and the plastic strains (4) at t_{n+1}. This yields the discrete equation

$$\Delta\boldsymbol{\sigma} = \mathbf{c}^{\mathrm{e}}_{n+1} : \left(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p1}}_{n+1} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p2}}_{n+1}\right) , \tag{23}$$

in which the plastic strain increments are

$$\begin{aligned} \Delta\boldsymbol{\varepsilon}^{\mathrm{p1}}_{n+1} &= \Delta\lambda_1 \left(\bar{\boldsymbol{n}}_{n+1} + \sqrt{2/27} D_{1,n+1}\boldsymbol{I}\right) , \\ \Delta\boldsymbol{\varepsilon}^{\mathrm{p2}}_{n+1} &= \Delta\lambda_2 \left(\bar{\boldsymbol{m}}_{n+1} + \sqrt{2/27} D_{2,n+1}\boldsymbol{I}\right) . \end{aligned} \tag{24}$$

For the cone's state variables, the idea of (22) can directly be applied to the rate equations listed in Table 1. Thus, the Backward Euler increments are

$$\begin{aligned} \Delta\lambda_1 &= \boldsymbol{\Theta}_{n+1} : \Delta\boldsymbol{\varepsilon} , \\ \Delta H_1 &= \frac{A_{n+1}}{p_{n+1}} \bar{K}_{\mathrm{p1},n+1} \Delta\lambda_1 . \end{aligned} \tag{25}$$

The incremental loading index of the cap $\Delta\lambda_2$ is not needed to update the state variables. Therefore, one could substitute the equation for $\Delta\lambda_2$ into (23). It turned out, however, that this leads to a poor convergence rate. Similar findings were reported in (de Borst and Heeres, 2002). Therefore, the discrete version

$$\Delta\lambda_2 = \boldsymbol{Z}_{n+1} : \Delta\boldsymbol{\varepsilon} \tag{26}$$

of rate equation (15) for $\dot{\lambda}_2$ is included in the Backward Euler algorithm to be developed here.

All members of Equations (23), (25) and (26) with the subscript $(\bullet)_{n+1}$ are non-linear functions of $\boldsymbol{\sigma}_{n+1}$ and $\boldsymbol{q}_{n+1}$. An iterative solution for this problem is developed in Section 3.2.2. The remaining state variables of Table 1 are treated differently from λ_1 and H_1. This is discussed in the following paragraphs.

Equation (B.8) for the void ratio e can be analytically integrated in Backward Euler fashion, so

$$e_{n+1} = \frac{e_n - \mathrm{tr}(\Delta\boldsymbol{\varepsilon})}{1 + \mathrm{tr}(\Delta\boldsymbol{\varepsilon})} . \tag{27}$$

According to the hardening rule (8), the implicit update of H_2 should be

$$H_{2,n+1} = H_{2,n} + \bar{K}_{\mathrm{p2},n+1} \Delta\lambda_2 . \tag{28}$$

Instead, the simplified update rule

$$H_{2,n+1} = \begin{cases} p_n & \text{if } p_{n+1} \leq p_n \\ p_{n+1} & \text{if } p_{n+1} > p_n \end{cases} \tag{29}$$

can be used because (B.20) essentially means that H_2 is the largest pressure that has been experienced throughout the loading history. This makes $H_{2,n+1}$ a dependent variable that can be calculated directly as soon as some estimate for $\boldsymbol{\sigma}_{n+1}$ is at hand.

The projection centres $\boldsymbol{\alpha}$ and β evolve stepwise, as described in Section 2.4. This is implemented as

$$\begin{aligned} \boldsymbol{\alpha}_{n+1} &= \begin{cases} \boldsymbol{\alpha}_n & \text{if } \Delta\lambda_1 \geq 0 \\ \boldsymbol{r}_n & \text{if } \Delta\lambda_1 < 0 \end{cases} , \\ \beta_{n+1} &= \begin{cases} \beta_n & \text{if } \left(\Delta p (p_n - \beta_n)\right) \geq 0 \\ p_n & \text{if } \left(\Delta p (p_n - \beta_n)\right) < 0 \end{cases} . \end{aligned} \tag{30}$$

3.2.2. Iterative solution

In order to calculate the Backward Euler increments (23), (25) and (26) iteratively, the equations are rearranged into the residual form

$$\begin{aligned} \boldsymbol{\emptyset} &= \Delta\boldsymbol{\sigma} - \mathbf{c}^{\mathrm{e}}_{n+1} : \left(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p1}}_{n+1} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p2}}_{n+1}\right) , \\ 0 &= \Delta\lambda_1 - \boldsymbol{\Theta}_{n+1} : \Delta\boldsymbol{\varepsilon} , \\ 0 &= \Delta H_1 - \frac{A_{n+1}}{p_{n+1}} \bar{K}_{\mathrm{p1},n+1} \Delta\lambda_1 , \\ 0 &= \Delta\lambda_2 - \boldsymbol{Z}_{n+1} : \Delta\boldsymbol{\varepsilon} . \end{aligned} \tag{31}$$

All unknowns of (31) are cast into the vector

$$\boldsymbol{u} := \begin{bmatrix} \Delta\boldsymbol{\sigma} & \Delta\lambda_1 & \Delta H_1 & \Delta\lambda_2 \end{bmatrix}^{\mathrm{T}} . \tag{32}$$

From the right-hand side of (31), the vector of residuals

$$\boldsymbol{R}(\boldsymbol{u}) := \begin{bmatrix} \Delta\boldsymbol{\sigma} - \mathbf{c}^{\mathrm{e}}_{n+1} : \left(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p1}}_{n+1} - \Delta\boldsymbol{\varepsilon}^{\mathrm{p2}}_{n+1}\right) \\ \Delta\lambda_1 - \boldsymbol{\Theta}_{n+1} : \Delta\boldsymbol{\varepsilon} \\ \Delta H_1 - (A_{n+1}/p_{n+1}) \bar{K}_{\mathrm{p1},n+1} \Delta\lambda_1 \\ \Delta\lambda_2 - \boldsymbol{Z}_{n+1} : \Delta\boldsymbol{\varepsilon} \end{bmatrix} \tag{33}$$

is defined.

The problem of solving (31) is then reformulated as finding the vector $\boldsymbol{u}_0$ for which

$$\boldsymbol{R}(\boldsymbol{u} = \boldsymbol{u}_0) = \boldsymbol{\emptyset}\,. \tag{34}$$

The task is accomplished by Newton's method. As initial guess for the unknowns, $\boldsymbol{u}_0^{i=0} = \boldsymbol{\emptyset}$ is assumed. This was proposed by Petalas and Dafalias (2019) and is equivalent to a Forward Euler trial step.

During the $(i+1)$th iteration, the function $\boldsymbol{R}(\boldsymbol{u})$ is linearised at $\boldsymbol{u}_0^i$ by

$$\mathrm{LIN}(\boldsymbol{R}; \boldsymbol{u} - \boldsymbol{u}_0^i) = \boldsymbol{R}(\boldsymbol{u}_0^i) + \boldsymbol{J}(\boldsymbol{u}_0^i) \cdot \left(\boldsymbol{u} - \boldsymbol{u}_0^i\right)\,, \tag{35}$$

in which the Jacobian matrix is

$$\boldsymbol{J}(\boldsymbol{u}_0^i) := \left.\frac{\partial \boldsymbol{R}(\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{u}_0^i}\,, \quad J_{lk} = \frac{\partial R_l}{\partial u_k}\,. \tag{36}$$

Let $\boldsymbol{u}_0^{i+1}$ be the zero of the linear equation (35). It constitutes an improved estimate for $\boldsymbol{u}_0$ and is found by solving

$$\Delta \boldsymbol{u}_0^i = \left(\boldsymbol{J}^i\right)^{-1} \cdot \left(-\boldsymbol{R}^i\right), \tag{37}$$

in which $\Delta \boldsymbol{u}_0^i := \boldsymbol{u}_0^{i+1} - \boldsymbol{u}_0^i$, $\boldsymbol{J}^i := \boldsymbol{J}(\boldsymbol{u}_0^i)$ and $\boldsymbol{R}^i := \boldsymbol{R}(\boldsymbol{u}_0^i)$. The iterative solution is then updated to

$$\boldsymbol{u}_0^{i+1} = \boldsymbol{u}_0^i + \Delta \boldsymbol{u}_0^i \tag{38}$$

and the new residual $\boldsymbol{R}^{i+1} = \boldsymbol{R}(\boldsymbol{u}_0^{i+1})$ is computed.

The iteration is repeated until

$$\|\boldsymbol{R}^{i+1}\| \leq \mathit{ITOL} \quad \text{and} \quad \|\Delta \boldsymbol{u}_0^i\| \leq \mathit{ITOL}\,, \tag{39}$$

with the tolerance ITOL being an user input to the algorithm. If convergence is not achieved within the maximum number of iterations $i_{\max}$, the input strain increment $\Delta\boldsymbol{\varepsilon}$ is split into substeps

$$\Delta\boldsymbol{\varepsilon}_{\mathrm{sub}} = \Delta T \Delta\boldsymbol{\varepsilon}\,. \tag{40}$$

In a first attempt, $\Delta T = 0.5$ is chosen for the increment of the unitless substepping pseudo-time $T \in [0, 1]$, yielding to substeps. The iteration is then restarted for the first subincrement. Repeated substepping may reduce ΔT to its pre-defined lower limit $\Delta T_{\min}$. In that case, the solution is accepted and a warning message is issued to the calling routine.

It is important to note that the convergence criterion (39) does not monitor the accuracy of the updated stresses and state variables. Instead, it measures to what extend (34) is satisfied by $\boldsymbol{u}_0^{i+1}$ and how much the solution has changed during the current iteration. The initiation of substepping through the convergence criterion improves the robustness of the method, but it is not suited to enhance the accuracy. That would require a method for estimating the local numerical error, as is presented for the explicit integrator in Section 3.3. In the context of implicit time integration an error estimate can be constructed by a Richardson extrapolation of the fully implicit Backward Euler scheme (22). Based on this, a semi-implicit method with error control can be derived (Fellin et al., 2009).

The complete implicit stress point algorithm is summarised in Algorithm 1. It relies on the numerical approximation of the Jacobian discussed in the following section.

Algorithm 1: Implicit stress point algorithm with substepping for the constitutive model by Li (2002), based on the method by de Borst and Heeres (2002)

```
Input: $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, $\Delta\boldsymbol{\varepsilon}$, $ITOL$, $i_{\max}$, $\Delta T_{\min}$
Output: $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{q}_{n+1}$
Initialize $T = 0$, $\Delta T = 1$, $\boldsymbol{\sigma}_T = \boldsymbol{\sigma}_n$, $\boldsymbol{q}_T = \boldsymbol{q}_n$
while $T < 1.0$ do
    compute strain subincrement $\Delta\boldsymbol{\varepsilon}_{\mathrm{sub}} = \Delta T \Delta\boldsymbol{\varepsilon}$
    initialize $i = 0$ and $\boldsymbol{u}_0^{i=0} = \boldsymbol{\emptyset}$
    compute $\boldsymbol{R}(\boldsymbol{u}_0^i)$ according to (33)
    update $\boldsymbol{\alpha}$ and $\beta$ according to (30)
    while $i < i_{\max}$ do
        compute Jacobian $\boldsymbol{J}^i = \boldsymbol{J}(\boldsymbol{u}_0^i)$ (Algorithm 2)
        solve $\Delta\boldsymbol{u}_0^i = \left(\boldsymbol{J}^i\right)^{-1} \cdot \left(-\boldsymbol{R}^i\right)$
        update $\boldsymbol{u}_0^{i+1} = \boldsymbol{u}^i + \Delta\boldsymbol{u}_0^i$
        compute $\boldsymbol{R}^{i+1} = \boldsymbol{R}(\boldsymbol{u}_0^{i+1})$ according to (33)
        update $i = i + 1$
        Check convergence criterion (39)
        if convergence criterion met then exit
    end while
    if convergence failed and $\Delta T > \Delta T_{\min}$ then
        reject substep
        restart with reduced increment $\Delta T = \Delta T/2$
    else
        update $\boldsymbol{\sigma}_{T+\Delta T}$ and $\boldsymbol{q}_{T+\Delta T}$ from $\boldsymbol{u}_0^i$
        update $T = T + \Delta T$
    end if
end while
Exit with $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_T$ and $\boldsymbol{q}_{n+1} = \boldsymbol{q}_T$
```

3.2.3. Computation of the local Jacobian

This section is intended to present a method for calculating the Jacobian $\boldsymbol{J} = \partial\boldsymbol{R}/\partial\boldsymbol{u}$ defined in (36). According to (32) and (33), the residual $\boldsymbol{R}$ depends on the components of $\boldsymbol{u}$ directly as well as through the discrete constitutive functions for $\mathbf{c}_{n+1}^{ep}$, $\boldsymbol{\Theta}_{n+1}$, A_{n+1} and $\bar{K}_{\mathrm{p1},n+1}$. These functions are formulated in terms of the image stress ratio $\bar{\boldsymbol{r}}$ and its invariants $\bar{R}$ and $\bar{\theta}$, as is clear from Appendix B. This poses a particular problem, because $\bar{\boldsymbol{r}}_{n+1}$ is calculated from $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}$ and $H_{1,n+1} = H_n + \Delta H_1$ by the iterative mapping procedure discussed in Section 2.4, which cannot be described by a differentiable equation. Therefore, $\bar{\boldsymbol{r}}$ cannot be differentiated with respect to the unknowns $\Delta\boldsymbol{\sigma}$ and ΔH_1, and an analytical expression for $\boldsymbol{J}$ cannot be derived. Instead, the partial derivates in (36) are obtained through numerical differentiation after Pérez-Foguet et al. (2000a,b).

Algorithm 2: Approximate computation of the Jacobian using the method by Pérez-Foguet et al. (2000a)

```
Input: $\boldsymbol{u}^i$, $\boldsymbol{R}(\boldsymbol{u}^i)$, $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, $\Delta\boldsymbol{\varepsilon}$, relative stepsize $h_\mathrm{r}$
Output: $J_{lk} = \partial R_l/\partial u_k$ approximated at $\boldsymbol{u}^i$
for $k = 1$ to $p + 2$ do
    compute stepsize $h = h_\mathrm{r} \max(|u^i_k|, 1.0)$
    compute $\boldsymbol{R}(\boldsymbol{u}^i + h\boldsymbol{e}_k)$ according to (33) (also
      requires $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$ and $\Delta\boldsymbol{\varepsilon}$)
    for $l = 1$ to $p + 3$ do compute $J_{lk}$ by (43)
end for
```

To outline the basic idea of the procedure, let $f(\boldsymbol{v})$ be a scalar function of some vector-valued variable $\boldsymbol{v}$ with components v_j. Then, the partial derivates of f at a given point $\boldsymbol{v}_0$ can be approximated by a forward difference scheme as

$$\left.\frac{\partial f(v_j)}{\partial v_j}\right|_{\boldsymbol{v}=\boldsymbol{v}_0} = \frac{f(\boldsymbol{v}_0 + h_j\boldsymbol{e}_j) - f(\boldsymbol{v}_0)}{h_j} + \mathcal{O}(h_j)\,. \qquad (41)$$

The variables $\boldsymbol{e}_j$ and h_j are the unit vector and the step size for the j-th component of $\boldsymbol{v}$, respectively. The higher order terms $\mathcal{O}(h_j)$ are not calculated, but included here to indicate that the approximation is first order accurate.

With regard to the component $v_{0,j}$ of $\boldsymbol{v}_0$ the step size is computed through

$$h_j = h_\mathrm{r} \max(|v_{0,j}|, 1.0)\,. \qquad (42)$$

The relative step size h_r is a control parameter that has to be pre-defined. In previous works, the approximation (41) proved to be efficient and robust for a wide range of values for h_r. Here, $h_\mathrm{r} = 1 \times 10^{-6}$ is chosen because that value yielded good results in all numerical experiments of Pérez-Foguet et al. (2000a).

The one-dimensional formula (41) can be applied to the vector-valued $\boldsymbol{R}$ if f is replaced by one component R_l. Consequently, the numerical approximation for $\boldsymbol{J}$ at $\boldsymbol{u}^i$ is

$$J_{lk} = \left.\frac{\partial R_l}{\partial u_k}\right|_{\boldsymbol{u}=\boldsymbol{u}^i} = \frac{R_l(\boldsymbol{u}^i + h_k\boldsymbol{e}_k) - R_l(\boldsymbol{u}^i)}{h_k}\,. \qquad (43)$$

The basic steps which are necessary to compute (43) are listed in Algorithm 2. The computational costs are scaled by the number of components of $\boldsymbol{R}$ and $\boldsymbol{u}$. According to (32), the length of these two vectors is $p + 3$, in which

$$p = \begin{cases} 4 & \text{for plane strain and axisymmetry} \\ 6 & \text{for 3D} \end{cases} \qquad (44)$$

is the length of stress and strain vectors in Voigt notation.

3.3. Explicit stress point algorithm

3.3.1. Basic concepts

The second stress point algorithm to be developed solves the problem (21) through the explicit method proposed by Sloan (1987); Sloan et al. (2001). Accordingly, stress and state variables are updated by

$$\begin{aligned} \tilde{\boldsymbol{\sigma}}_{n+1} &= \boldsymbol{\sigma}_n + \tfrac{1}{2}(\Delta\boldsymbol{\sigma}_1 + \Delta\boldsymbol{\sigma}_2)\,, \\ \tilde{\boldsymbol{q}}_{n+1} &= \boldsymbol{q}_n + \tfrac{1}{2}(\Delta\boldsymbol{q}_1 + \Delta\boldsymbol{q}_2)\,. \end{aligned} \qquad (45)$$

The first group of increments is computed by the Forward Euler method, so

$$\begin{aligned} \Delta\boldsymbol{\sigma}_1 &= g(\boldsymbol{\sigma}_n, \boldsymbol{q}_n, \Delta\boldsymbol{\varepsilon})\,, \\ \Delta\boldsymbol{q}_1 &= h(\boldsymbol{\sigma}_n, \boldsymbol{q}_n, \Delta\boldsymbol{\varepsilon})\,, \end{aligned} \qquad (46)$$

in which g and h are the generic constitutive functions introduced with (19). The second group of increments is

$$\begin{aligned} \Delta\boldsymbol{\sigma}_2 &= g(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_1, \boldsymbol{q}_n + \Delta\boldsymbol{q}_1, \Delta\boldsymbol{\varepsilon}), \\ \Delta\boldsymbol{q}_2 &= h(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_1, \boldsymbol{q}_n + \Delta\boldsymbol{q}_1, \Delta\boldsymbol{\varepsilon})\,. \end{aligned} \qquad (47)$$

The approximation (45) is frequently called “Modified Euler Scheme”, but from a theoretical point of view it represents a second order Runge-Kutta method (Tamagnini et al., 2000).

The relative numerical error of the explicit approximation is estimated from

$$R_\mathrm{rel} = \max\left(\frac{\|\Delta\boldsymbol{\sigma}_2 - \Delta\boldsymbol{\sigma}_1\|}{\|\tilde{\boldsymbol{\sigma}}_{n+1}\|}, \frac{\|\Delta\boldsymbol{q}_2 - \Delta\boldsymbol{q}_1\|}{\|\tilde{\boldsymbol{q}}_{n+1}\|}\right)\,. \qquad (48)$$

An error threshold is pre-defined as $STOL$. If the criterion

$$R_\mathrm{rel} \leq STOL \qquad (49)$$

is not met, the input strain increment $\Delta\varepsilon$ is split into substeps $\Delta\boldsymbol{\varepsilon}_\mathrm{sub}$ as shown in (40) and the update procedure is repeated for the subincrements. Otherwise, the solutions (45) are accepted as final values for $\boldsymbol{\sigma}_{n+1}$ and $\boldsymbol{q}_{n+1}$. This technique is commonly referred to as *automatic error control.*

The normalisation of R_rel by $\|\tilde{\boldsymbol{\sigma}}_{n+1}\|$ and $\|\tilde{\boldsymbol{q}}_{n+1}\|$ in (48) allows to treat different levels of stresses and state variables with the same tolerance. A drawback of the approach is that numerical singularities arise if one of the norms in the denominators tends to zero. If that is of relevance, as for example in boundary value problems with a free surface and no preloading, the mixture of relative and absolute errors proposed in (Fellin et al., 2009) could be employed.

The explicit procedure is summarized in Algorithm 3. All calculation steps from (Sloan et al., 2001) which are related to the yield surface have been omitted because the constitutive model is of zero elastic range type. This simplifies the implementation significantly, but might affect the accuracy because the evolution of stress and state variables is restricted less strongly.

3.3.2. Computation of substep size

In Algorithm 3, the size of local substeps is defined by the increment ΔT of the pseudo-time $T \in [0, 1]$, analogous to Algorithm 1. Depending on whether or not the criterion

Algorithm 3: Explicit stress point algorithm with automatic error control for the model by Li (2002), based on the method by Sloan et al. (2001)

```
**Input:** $\boldsymbol{\sigma}_n$, $\boldsymbol{q}_n$, $\Delta\boldsymbol{\varepsilon}$, $STOL$
**Output:** $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{q}_{n+1}$
Initialize $T = 0$, $\Delta T = 1$, $\boldsymbol{\sigma}_T = \boldsymbol{\sigma}_n$, $\boldsymbol{q}_T = \boldsymbol{q}_n$
**while** $T < 1$ **do**
    compute strain subincrement $\Delta\boldsymbol{\varepsilon}_{\mathrm{sub}} = \Delta T \Delta\boldsymbol{\varepsilon}$
    **for** $k = 1$ **to** $2$ **do**
        compute
            $\Delta\boldsymbol{\sigma}_k = g(\tilde{\boldsymbol{\sigma}}_k, \tilde{\boldsymbol{q}}_k, \Delta\boldsymbol{\varepsilon}_{\mathrm{sub}})$
            $\Delta\boldsymbol{q}_k = h(\tilde{\boldsymbol{\sigma}}_k, \tilde{\boldsymbol{q}}_k, \Delta\boldsymbol{\varepsilon}_{\mathrm{sub}})$
        where
            $\tilde{\boldsymbol{\sigma}}_1 = \boldsymbol{\sigma}_T \, , \quad \tilde{\boldsymbol{q}}_1 = \boldsymbol{q}_T$
            $\tilde{\boldsymbol{\sigma}}_2 = \boldsymbol{\sigma}_T + \Delta\boldsymbol{\sigma}_1 \, , \quad \tilde{\boldsymbol{q}}_2 = \boldsymbol{q}_T + \Delta\boldsymbol{q}_1$
    **end for**
    update $\boldsymbol{\alpha}$ and $\beta$ according to (30)
    update stress and material state according to
        $\tilde{\boldsymbol{\sigma}}_{T+\Delta T} = \boldsymbol{\sigma}_T + \frac{1}{2}(\Delta\boldsymbol{\sigma}_1 + \Delta\boldsymbol{\sigma}_2)$
        $\tilde{\boldsymbol{q}}_{T+\Delta T} = \boldsymbol{q}_T + \frac{1}{2}(\Delta\boldsymbol{q}_1 + \Delta\boldsymbol{q}_2)$
    Determine relative error $R_{\mathrm{rel}}(T + \Delta T)$ by (48)
    **if** $R_{\mathrm{rel}}(T + \Delta T) > STOL$ **then**
        reject substep
        decrease $\Delta T$ according to (50)
        go back to the beginning of the while loop
    **else** successful substep
        update
            $\boldsymbol{\sigma}_{T+\Delta T} = \tilde{\boldsymbol{\sigma}}_{T+\Delta T}$
            $\boldsymbol{q}_{T+\Delta T} = \tilde{\boldsymbol{\alpha}}_{T+\Delta T}$
            $T = T + \Delta T$
        increase $\Delta T$ according to (50)
    **end if**
**end while**
Exit with $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_T$ and $\boldsymbol{q}_{n+1} = \boldsymbol{q}_T$
```

(49) has been met, ΔT may increase or decrease after each substep.

Sloan et al. (2001) deduced guidelines for the computation of ΔT from numerical experiments. These are meant to guarantee optimal performance of the algorithm and thus are adopted here. Accordingly, the new increment size is determined at the end of substep k as

$$\Delta T_{k+1} = q \Delta T_k \, . \tag{50}$$

The step factor is set to

$$q = 0.9\sqrt{STOL/R_{\mathrm{rel}}(T_k)} \tag{51}$$

and constrained to lie within the range

$$0.1 \leq q \leq 1.1 \, . \tag{52}$$

4. Results and discussion

4.1. Criteria and methods for the test calculations

In subsequent sections, the implicit stress point algorithm and the explicit one are compared by means of example calculations. The results are assessed with respect to the algorithms' stability, accuracy and efficiency. These criteria are defined as follows (Tamagnini et al., 2000; Neto et al., 2008).

Stability. A stable algorithm ensures that variations of the solution which result from perturbations of the initial conditions of a calculation step are bounded within that step. In an unstable solution, on the other hand, the variations propagate through subsequent steps and may even be amplified, which then leads to erroneous results.

Accuracy. A numerical solution is regarded accurate if the difference to an exact solution is small. The exact solution usually is computed by means of numerical methods as well, but with load increments which are small enough to yield very high accuracy.

Efficiency. Algorithm A is regarded more efficient than algorithm B if A takes less computational costs than B to produce a solution with a given level of accuracy.

In general, the performance of integration algorithms with regard to the criteria defined above mostly depends on three factors: 1) the initial state conditions, 2) the magnitude of input strain increments and 3) the strain increment direction. The impact of the first two factors is investigated here through example calculations at stress point level with custom Fortran programs and *Incremental Driver* (Niemunis, 2017). To furthermore account for strain increments of different directions and to assess the algorithms' overall efficiency, a boundary value problem was solved by finite element simulations with *ANSYS® Mechanical™ APDL 17.0.*

Both ANSYS® and Incremental Driver require their constitutive routines to provide algorithmic stiffness operators for the Jacobians of the global Newton iteration. For that purpose, the stress update algorithms implemented

Table 2: Constitutive parameters for Toyoura Sand (Li, 2002)

Elasticity	CSL	Cone	Cap
$G_0 = 125$	$M = 1.25$	$d_1 = 0.41$	$d_2 = 1.0$
$\nu = 0.25$	$e_\Gamma = 0.934$	$m = 3.5$	$h_4 = 3.5$
	$\lambda_c = 0.019$	$h_1 = 3.15$	$a = 1.0$
	$\xi = 0.7$	$h_2 = 3.05$	
	$c = 0.75$	$h_3 = 2.2$	
		$n = 1.1$	

here use the continuum tangent $\mathbf{c}^{\mathrm{ep}}$ computed last in the course of the stress integration procedure. This is the simplest possible choice because it does not require additional computations. More elaborate approaches are discussed in Section 5.

Last but not least, all results presented subsequently were obtained using the constitutive parameters listed in Table 2.

4.2. Verification

Before investigating the algorithms' performance, it is necessary to verify that the constitutive routines are consistent with the underlying rate equations. To accomplish that, results from numerical simulations of element tests are compared to reference results from (Li, 2002). For the benefit of a clear presentation, only the explicit algorithm without local substepping is used. However, the source code for the constitutive functions is shared between the implicit and the explicit algorithm. Therefore, both methods yield the same results if load increments are small enough to render numerical errors negligible.

First, the cap mechanism is tested by means of isotropic and oedometric compression with un- and reloading. The loads are applied in steps of 1 kPa. The results obtained here are compared to those of the reference in Figure 3. They match very closely. This proves that the constitutive functions of the cap were implemented correctly.

Second, constant volume simple shear was simulated to verify those constitutive functions which are related to the cone bounding surface and account for changes in stress ratio. Cyclic loading with a shear stress amplitude of 60 kPa was applied in 120 steps per cycle. The results are shown in Figure 4. Accordingly, the stress path obtained here is very similar to the one presented in (Li, 2002). In detail, however, the curves differ slightly. On the other hand, the strain evolution computed here agrees very well with the reference.

When interpreting Figure 4, one has to bear in mind that the loading is stress controlled and that the successive build-up of excess pore pressure reduces the incremental stiffness severely. Because of that, the results strongly depend on the load step size as well as on the choice of numerical methods for the global iteration and the local algorithmic stress update. These computational parameters are unknown for the reference results, which makes it very difficult to exactly reproduce the results. In light of that, the small deviations in the stress path in Figure 4 can be accepted. The implementation of the constitutive functions therefore is considered successfully verified.

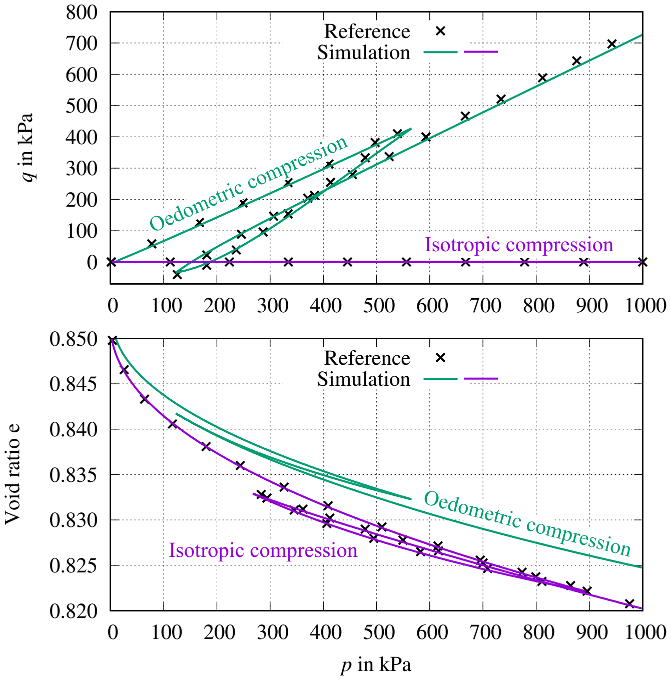


Figure 3: Verification of the cap functions through simulation of isotropic and oedometric compression; $e_0 = 0.85$; the reference is (Li, 2002); reference results for e in oedometric compression are not available

4.3. Performance at stress point level

4.3.1. Stability

This section investigates how the stability of Algorithms 1 and 3 is affected by the input strain increment size. For that purpose, constant volume triaxial compression was simulated up to a total equivalent shear strain of $\varepsilon_q = 5\,\%$ with different values for the increment $\Delta\varepsilon_q$.

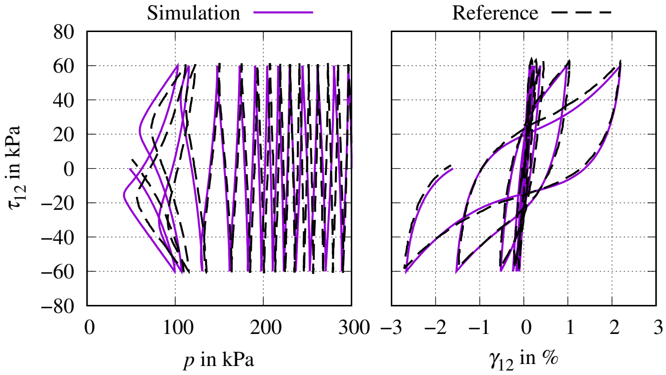


Figure 4: Verification of the constitutive functions through simulation of constant volume simple shear; $p_0 = 300\,\mathrm{kPa}$ and $e_0 = 0.73$; the reference is (Li, 2002); $\gamma_{12} = 2\varepsilon_{12}$ is the engineering shear strain

The initial state parameters are $e_0 = 0.93$ and $p_0 = 100\,\mathrm{kPa}$. They represent a soil specimen that exhibits low stiffness and shear strength as well as post-peak softening in undrained conditions. This creates a demanding test case for the numerical procedures.

An error tolerance of $STOL = 10^{-4}$ is pre-defined for the explicit substepping algorithm. This is considered a conservative value in most situations (Sloan, 1987). With regard to the convergence tolerance of the implicit iteration $ITOL$, no clear guidelines are available. Previous research publications reported values between 10^{-6} and 10^{-12} (Pérez-Foguet et al., 2000a; Tamagnini et al., 2002; de Borst and Heeres, 2002; Petalas and Dafalias, 2019). However, $ITOL = 10^{-6}$ already appears to be a fairly strict tolerance for practical applications. Therefore, this value is adopted as default here.

The stress evolution, as computed by both algorithms with different $\Delta\varepsilon_q$, is presented in Figure 5. Additionally, curves obtained by explicit Forward Euler integration are presented. This is done to provide an example for an integrator which is known to be unstable for large strain increments. The Forward Euler method is obtained from Algorithm 3 by omitting the error control and the Runge-Kutta update.

The outputs of all integration methods in Figure 5 tend towards limit curves when $\Delta\varepsilon_q$ decreases. This implies that each of the three methods yields reasonable results if the step size is sufficiently small.

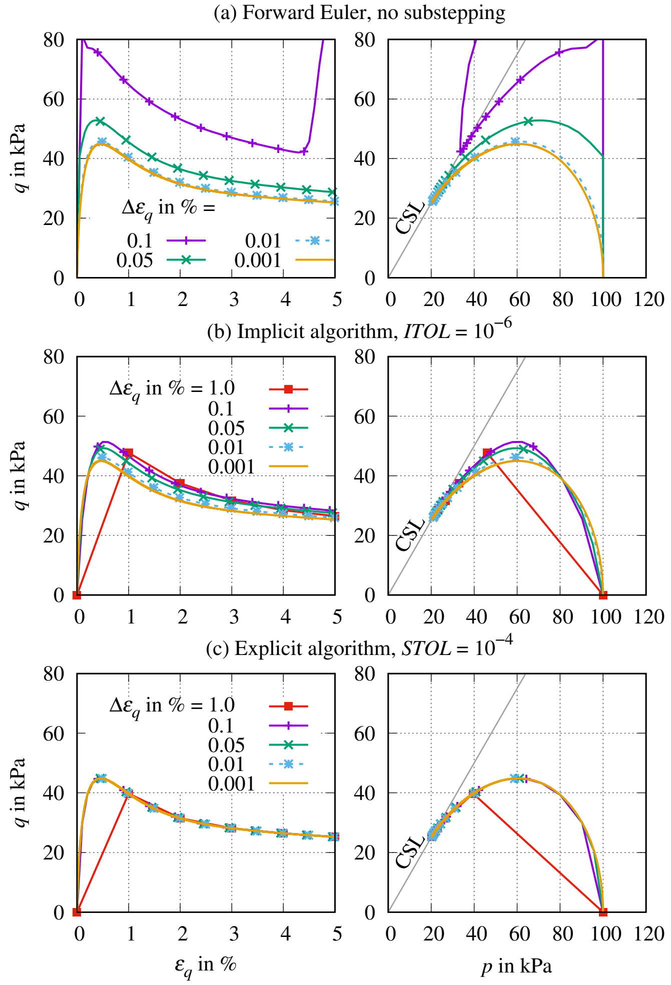
The curves of the Forward Euler method in Figure 5 (a) are significantly affected by $\Delta\varepsilon_q$. For $\Delta\varepsilon_q = 0.1\,\%$, the stress even becomes physically implausible in two aspects: 1) The peak strength is greatly overestimated. 2) The material state drifts away from the Critical State Line for $\varepsilon_q > 4.5\,\%$. The second phenomenon clearly contradicts the theoretical framework of the constitutive model and the more accurate solutions. Most likely, it results from the property of explicit integrators to be unstable for steps larger than a critical size.

The stress curves of the implicit algorithm in Figure 5 (b) continuously change with decreasing $\Delta\varepsilon_q$ until they converge to the limit curve. Nevertheless, the output remains plausible even for $\Delta\varepsilon_q = 1.0\,\%$. Thus, the algorithm is regarded stable for large strain increments.

In Figure 5 (c), all curves which were calculated by the explicit algorithm with automatic error control match very closely. Even the first step of the computation with a large input strain increment of $\Delta\varepsilon_q = 1.0\,\%$ precisely arrives at the stress path obtained with $\Delta\varepsilon_q = 0.001\,\%$. The stability of the explicit substepping algorithm therefore appears to be almost insensitive to changes in $\Delta\varepsilon_q$.

4.3.2. Effect of stress point tolerances on stability

The same test case as in Section 4.3.1 is employed. This time, the global strain increment is kept constant at $\Delta\varepsilon_q = 0.1\,\%$ while the implicit tolerance $ITOL$ and the explicit tolerance $STOL$ are varied. Additionally, reference curves are presented, which were computed by the


(a) Forward Euler, no substepping

(b) Implicit algorithm, $ITOL = 10^{-6}$

(c) Explicit algorithm, $STOL = 10^{-4}$

Figure 5: Influence of strain increment size on simulation results for constant volume triaxial compression ($p_0 = 100\,\mathrm{kPa}$, $e_0 = 0.93$)

explicit algorithm with $STOL = 10^{-9}$ along with a global equivalent shear strain increment of $\Delta\varepsilon_q = 0.001\,\%$ and consequently are highly accurate.

According to Figure 6 (a), the following results are obtained with the implicit integration scheme:

- With decreasing $ITOL$, the stress output converges to a curve which clearly differs from the reference solution.
- The stress remains physically plausible even for unusually large tolerances such as $ITOL = 10$.
- No substepping was needed to obtain the solutions.
- The number of iterations n_{iter} grows mildly when $ITOL$ is decreased, e. g. maximum n_{iter} is 2 for $ITOL = 20$ and 6 for $ITOL = 10^{-8}$.

The outcome of the test calculations with the explicit algorithm is presented in Figure 6 (b). Accordingly, the most obvious effects of changes to $STOL$ are:

- For $STOL \geq 10^{-1}$, the stress becomes erroneous, similar to Forward Euler integration, see Figure 5 (a).
- With decreasing $STOL$, the output converges to the reference solution.
- The number of local substeps n_{subst} grows rapidly with decreasing $STOL$, e. g. maximum n_{subst} is 50 for $STOL = 10^{-4}$ and 5000 for $STOL = 10^{-8}$.

The results of the current section and the previous one are interpreted as follows. Both the implicit and the explicit algorithm are stable for large strain increments. The explicit method ensures stability by means of substepping. Consequently, the output tends to become unreliable if the error tolerance $STOL$ is chosen too large and the number of substeps therefore is too small. This is consistent with the fundamental property of explicit numerical approximations that stability is ensured only for input steps smaller than a critical size (Belytschko et al., 2014). On the other hand, the implicit algorithm appears to be inherently stable because of the underlying Backward Euler concept, which is enforced by the local iteration. Furthermore, the implicit method proved to be robust for a wide range of values for the convergence tolerance $ITOL$.

4.3.3. Numerical error for single time increments

The remaining parts of Section 4.3 focus on the accuracy of the stress point algorithms. For that purpose, single time integration steps are considered, in which an input strain increment $\Delta\boldsymbol{\varepsilon}$ drives an initial state given by $(\boldsymbol{\sigma}_0, \boldsymbol{q}_0)$ to a new configuration $(\boldsymbol{\sigma}_1, \boldsymbol{q}_1)$. The loading is of constant volume triaxial compression type. Its magnitude is described by the equivalent strain increment $\Delta\varepsilon_q$.

The accuracy of the updated stress is measured by means of the relative error

$$E_\sigma = \frac{\|\boldsymbol{\sigma}_{\text{ref}} - \boldsymbol{\sigma}_1\|}{\|\boldsymbol{\sigma}_{\text{ref}}\|}\,, \tag{53}$$

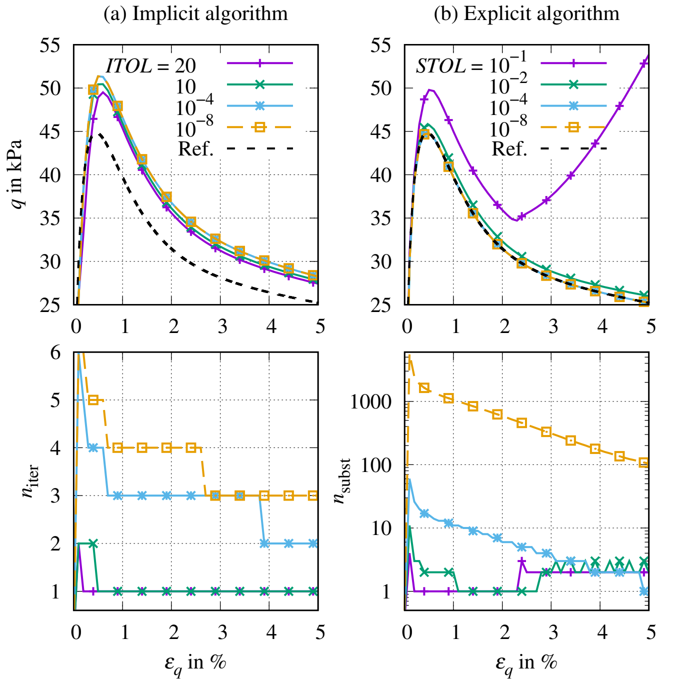


Figure 6: Influence of tolerances $ITOL$ and $STOL$ for the test case presented in Figure 5 with global load step size kept fix at $\Delta\varepsilon_q = 0.1\,\%$; all load steps in (a) converged successfully without local substepping

in which $\boldsymbol{\sigma}^{\text{ref}}$ is a reference solution with very high accuracy (Tamagnini et al., 2000; Mira et al., 2009). The reference solutions were computed here with the explicit Algorithm 3. In order to ensure very high accuracy, the input strain $\Delta\varepsilon_q$ was globally subdivided into steps of $10^{-4}\,\%$ and a very strict error tolerance of $STOL = 10^{-9}$ was chosen.

4.3.4. Accuracy for different strain increment magnitudes

The methodology described in the previous section was used to investigate how the equivalent strain increment magnitude $\Delta\varepsilon_q$ affects the accuracy of the stress point algorithms. The initial parameters from Section 4.3.1 were adopted, i. e. isotropic initial stress, $p_0 = 100\,\text{kPa}$ and $e_0 = 0.93$.

Figure 7 illustrates the evolution of E_σ with increasing $\Delta\varepsilon_q$. Results generated by both integration schemes with different values for their tolerances are shown in log-log-plots. It turns out that all graphs are approximately linear for $\Delta\varepsilon_q$ approaching zero. This is consistent with the results of analytical studies on the accuracy of numerical approximations (Neto et al., 2008).

The initial gradients of the error curves in Figure 7 are supposed to be $m + 1$, in which m is the order of accuracy of the numerical approximation (Lloret-Cabot et al., 2016). The Backward Euler approximation (22), that Algorithm 1 relies on, is of order $m = 1$. The Runge-Kutta approximation (45) of the explicit method with error control is second order accurate, i. e. $m = 2$. Therefore, the gradients of E_σ should be 2 and 3, respectively. To verify this, exponential functions were curve-fitted to the computation results in Figure 7. These functions match the data very well. This confirms that the stress point algorithms work as expected and that the numerical error analysis yields plausible results.

As $\Delta\varepsilon_q$ increases beyond $10^{-3}\,\%$, the curves in Figure 7 which were produced by the implicit algorithm slowly flatten. This implies that the implicit integrator can handle even very large strain increments. For $\Delta\varepsilon_q > 10^{-1}\,\%$, the error stabilises on a high level with $1\,\% < E_\sigma < 15\,\%$. This probably is too imprecise for practical applications with many successive calculation steps. However, the authors' experience suggests that global time increments usually are chosen so that $\Delta\varepsilon_q \leq 10^{-1}\,\%$ anyway. Otherwise, global solution procedures do not converge properly due to the strongly non-linear material behavior.

In addition to that, Figure 7 indicates that the value of the implicit convergence criterion $ITOL$ barely affects the error induced by a single strain increment. In contrast, Figure 6 clearly demonstrates that a complete test path evolves if $ITOL$ is changed. Probably, this discrepancy is due to the effect that sequential calculations with many consecutive steps amplify phenomena which appear negligible if merely one integration step is considered.

The results of the explicit method in Figure 7 are strongly affected by the value of the error tolerance $STOL$. For large $STOL$, the error criterion (49) is always met and thus no substepping takes place. In that case, the error E_σ grows linearly with the strain increment size in a log-log plot. This is illustrated by the curve labelled as *No substepping* in Figure 7, which was obtained with $STOL = 1.0$.

In contrast, all curves in Figure 7 that were obtained with the explicit method and $STOL \leq 10^{-2}$ leave the linear domain with a sharp bend. This bend is located at the smallest value of $\Delta\varepsilon_q$ which still causes the automatic error control to subdivide the input strain increment into substeps. It is evident from Figure 7 that the smaller $STOL$ is chosen, the smaller is the value for $\Delta\varepsilon_q$ which initiates substepping. Moreover, the evolution of E_σ is capped at a level of approximately $STOL \times 100\,\%$ for very large $\Delta\varepsilon_q$. These observations clearly prove that the automatic error control mechanism of the explicit algorithm works as intended.

The quantitative difference between the curves of the two integrators in Figure 7 is discussed in the next section.

4.3.5. Impact of initial state on numerical error

The calculations discussed in the previous section were repeated with four different sets of initial state parameters, termed A, B, C and D. The parameters were adopted from different points of the test path for $\Delta\varepsilon_q = 0.001\,\%$ in Figure 5 (c). They represent basic types of material behaviour that are encountered when a virgin sand sample is sheared: isotropic state (A), pre-peak hardening (B), post-peak softening (C) and shearing towards the Critical State Line (D). See Table 3 for further details.

The results produced by the implicit and the explicit algorithm are presented in Figure 8. It turns out that the numerical errors for a given $\Delta\varepsilon_q$ at the four initial states in general are related by

$$E_{\sigma,\mathrm{A}} > E_{\sigma,\mathrm{B}} > E_{\sigma,\mathrm{C}} > E_{\sigma,\mathrm{D}}\,. \tag{54}$$

Similarly, the absolute value of the equivalent stress increment $|\Delta q|$ induced by a given $\Delta\varepsilon_q$ is largest at A and decreases towards B, C and D. This follows from the slope of stress evolution, which, according to Table 3 and Figure 5 (c), is steepest at point A and flattens towards B, C and D. These findings imply that initial state parameters which cause small stress changes generate better accuracy than parameters that lead to large stress changes. This complies with the common observation that the accuracy strongly depends on the strain increment magnitude for a given initial state.

Another important finding is evident from Figure 8 as well as from Figure 7. Accordingly, the implicit method is less accurate than the explicit one for almost the entire range of computational parameters considered here. The only exception is the small portion of Figure 7, where $\Delta\varepsilon_q > 3\,\%$. For such unusually large input strains, the explicit method with a rather loose error tolerance of $STOL = 10^{-2}$ yields an error of similar magnitude as the implicit integrator.

Apparently, the considerable difference in accuracy between the two stress point algorithms is due to two reasons.

First, the implicit method is of lesser accuracy order, as has already been mentioned in Section 4.3.4. From a theoretical point of view, this does not necessarily provide information about the absolute values of numerical errors (Neto et al., 2008). Nevertheless, it is considered a relevant factor here.

Second, the explicit method systematically uses substepping to produce solutions which meet the pre-defined error tolerance $STOL$. In theory, this allows to reduce the numerical error down to levels where machine precision and rounding errors are predominant. No similar mechanism is included in the implicit stress point algorithm, as noted in Section 3.2.2. The accuracy of the implicit procedure therefore is conceptually limited by the magnitude of input strain increments.

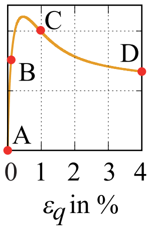
4.3.6. Convergence behaviour of the implicit scheme

Section 4.3.7 is concerned with the efficiency of the stress point algorithms. In case of implicit time integration, the efficiency depends crucially on the convergence speed of the local iteration. Newton's method should converge at quadratic rate if the Jacobian is computed sufficiently accurate. That means the magnitudes of the iteration increment and the vector of residuals decrease with the iteration number as quadratic functions The current section should explore whether this holds for the implicit procedure which is outlined in Algorithm 1.

The example calculations presented here are based on the same initial states as in the previous section. For

Table 3: Non-zero initial state parameters for calculation of E_σ; adopted from solution for $\Delta\varepsilon_q = 0.001\,\%$ in Figure 5 (c)

	A	B	C	D
ε_q in %	0.0	0.1	1.0	4.0
p in kPa	100	90.3	39.8	21.3
q in kPa	0.0	29.9	40.3	26.4
e	0.93	0.93	0.93	0.93
λ_1 in %	0	0.08	1.15	4.84
H_1	0	0.33	1.00	1.24
H_2	100	100	100	100
β	0.0	100	100	100



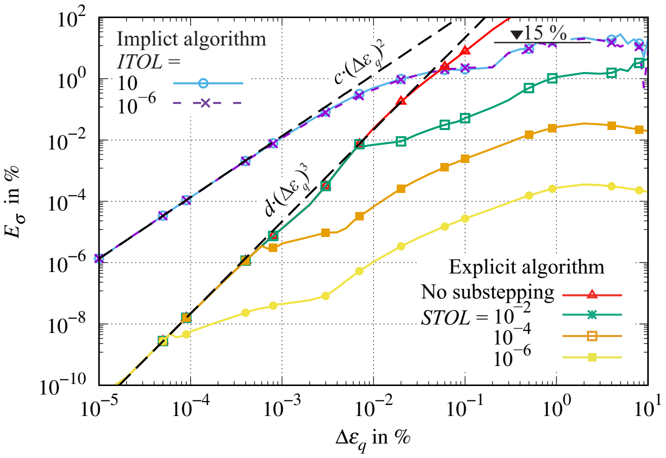
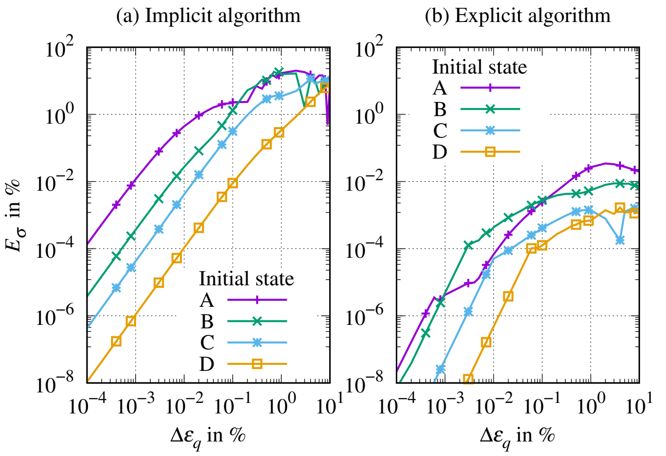


Figure 7: Numerical error E_σ for single increments of constant volume triaxial compression with magnitude $\Delta\varepsilon_q$; initial State A, see Table 3; parameters c and d were curve-fitted to the results obtained with $ITOL = 10^{-6}$ and $STOL = 10^{-4}$ in the range of $\Delta\varepsilon_q \leq 10^{-4}\,\%$; the curve labelled as *No substepping* was obtained with $STOL = 1.0$


(a) Implicit algorithm

(b) Explicit algorithm

Figure 8: Impact of initial state on E_σ for a single increment of constant volume triaxial compression with magnitude $\Delta\varepsilon_q$ ($ITOL = 10^{-6}$, $STOL = 10^{-4}$, see Table 3 for the definitions of A, B, C, D)

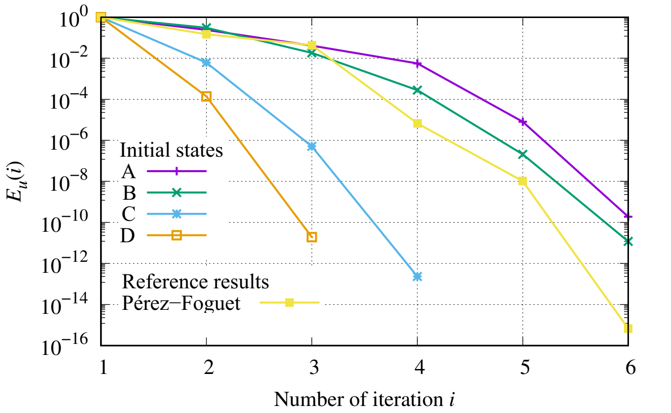


Figure 9: Convergence behaviour of the implicit algorithm with $ITOL = 10^{-6}$ for single increments of constant volume triaxial compression with $\Delta\varepsilon_q = 0.1\,\%$; Initial states adopted from Tab. 3; Reference results from (Pérez-Foguet et al., 2000b, Fig. 5) were obtained with $ITOL = 10^{-14}$

each of them, a single increment of constant volume triaxial compression with $\Delta\varepsilon_q = 0.1\,\%$ was integrated by the implicit algorithm. The evolution of the convergence parameters was monitored. It turned out that the iteration increment $\Delta\boldsymbol{u}$ converged slightly slower than the residual $\boldsymbol{R}$. Therefore, the convergence behaviour in each iteration i is assessed here with respect to the normalised iteration increment magnitude

$$E_u(i) = \frac{\|\Delta\boldsymbol{u}(i)\|}{\|\Delta\boldsymbol{u}(i=1)\|}\,. \tag{55}$$

The evolution of E_u with i for initial states A, B, C and D is presented in Figure 9. Accordingly, the fastest convergence is obtained for state D, the slowest for A. This is consistent with the effect of the initial state parameters observed in the previous section.

Furthermore, Figure 9 contains an example curve from the work by Pérez-Foguet et al. (2000b). It has been included for reference because it is said to represent quadratic convergence and was obtained with the same numerical approximation for the local Jacobian that is used here. Apparently, this reference curve is very similar to the results for initial states A, B, C and D. This proves that the numerical approximation for the Jacobian was implemented correctly and that the iterative procedure of Algorithm 1 can converge at quadratic rate.

4.3.7. Efficiency

In this section, the efficiency of Algorithms 1 and 3 is assessed in the manner of (Tamagnini et al., 2000). To this end, the stress response to constant volume triaxial compression with a total strain of $\varepsilon_q = 0.1\,\%$ is computed. The numerical error E_σ of the updated stress is varied systematically. The computational costs for a given E_σ are measured in terms of n_{eval}, which is the total number of evaluations of the constitutive functions during the integration procedure.

Table 4: Testing programme for assessing the algorithms' efficiency with regard to a constant volume triaxial compression load of magnitude $\varepsilon_q = 0.1\,\%$ ($\Delta\varepsilon_q$ is the global load step size)

	Implicit algorithm	Explicit algorithm
	$ITOL = 10^{-6}$	$\Delta\varepsilon_q = 0.1\,\%$
1.	$\Delta\varepsilon_q = 1 \times 10^{-1}\,\%$	$STOL = 10^{0}$
2.	$2 \times 10^{-2}\,\%$	10^{-1}
3.	$1 \times 10^{-2}\,\%$	10^{-2}
4.	$1 \times 10^{-3}\,\%$	10^{-3}
5.	$2 \times 10^{-4}\,\%$	10^{-4}
6.	$1 \times 10^{-4}\,\%$	10^{-5}
7.	$5 \times 10^{-5}\,\%$	10^{-6}

The previous sections have demonstrated that the accuracy of the explicit method is largely pre-defined by the error tolerance $STOL$. On the other hand, the implicit method's accuracy was shown to strongly depend on the global strain increment size $\Delta\varepsilon_q$. Both results are direct consequences of the basic concepts on which the algorithms rely. To account for this, the testing programme for the current section is organised as follows. The explicit integrator was invoked with seven different $STOL$, whereas for the implicit integrator seven different $\Delta\varepsilon_q$ were employed. Furthermore, a default value of $ITOL = 10^{-6}$ was adopted for the implicit convergence tolerance, as in previous sections. See Table 4 for more details of the setup.

In Figure 10, n_{eval} is plotted against E_σ for both integration methods and different initial states. Efficiency increases from the top right corner to the bottom left corner. According to the results presented in the diagram, initial state parameter affect the efficiency of the implicit and the explicit method in a similar way. Initial state A consumes most resources to achieve a certain degree of accuracy. If the initial state parameters change from set A to D via B and C, the required computational costs decrease. Analogous to Section 4.3.5, this implies that both algorithms are challenged most by initial states where the slope of stress evolution is steep.

Moreover, the results presented in Figure 10 reveal that for a given E_σ, the number n_{eval} induced by the implicit stress update is at least 10 times the n_{eval} from the explicit update. The largest difference which occurs even corresponds to a factor of roughly 1000. This indicates that the explicit stress point algorithm with error control is more efficient than the implicit Backward Euler algorithm. The reasons for that are further discussed in Section 4.5.

Additional numerical experiments showed that the efficiency of the implicit method improves only very slightly if the convergence tolerance $ITOL$ is increased above its default value. Hence, no results with regard to that are presented here.

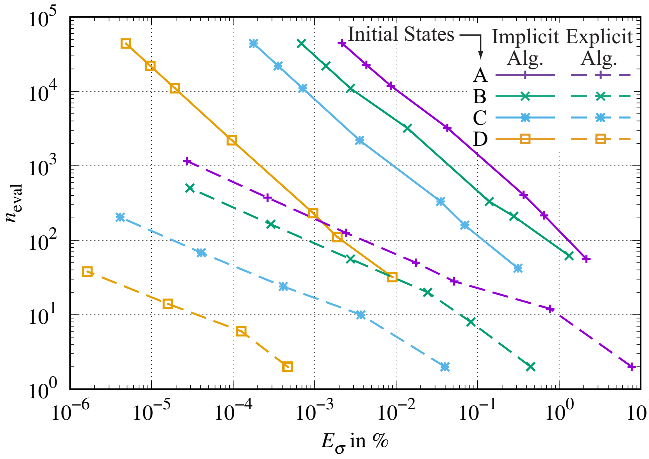


Figure 10: Impact of initial state on numerical efficiency for constant volume triaxial compression with $\varepsilon_q = 0.1\,\%$; $ITOL = 10^{-6}$, Initial states adopted from Tab. 3

4.4. Performance with respect to a boundary value problem

4.4.1. Problem statement

Section 4.4 addresses the algorithms' performance with regard to a fully fledged finite element simulation. The initial boundary value problem to be solved is the displacement controlled loading of a rigid, rough strip footing on weightless, drained soil. This example has served as benchmark in many works on numerical methods, for example in (Potts and Zdravković, 2001).

In order to study accuracy and efficiency similarly to Section 4.3, the following computational parameters were varied systematically: 1) stress point algorithm invoked at the finite element integration points, 2) initial conditions, 3) size of global load increments.

4.4.2. Finite Element model

The finite element model is displayed in Figure 11. Accordingly, only half of the problem was modelled by making use of symmetry. Furthermore, plane strain conditions were assumed to hold.

The soil was discretized by four-noded rectangles. Such elements may overestimate bearing capacity due to volumetric locking. Nevertheless, they were employed here out of two reasons. First, the problem is purely synthetic and the absolute value of bearing capacity therefore is of no relevance. Second, higher-order elements which are less prone to locking induce convergence problems of the global Newton procedure when large mesh distortions occur, as according to Figure 11 is the case here.

Since the footing is assumed to be perfectly rigid and rough, it can be modelled by appropriate boundary conditions instead of dedicated finite elements. Thus, a uniform vertical displacement u was applied to the nodes of the ground surface along the footing, while their relative horizontal movement was hindered. This is implied in Figure 11, which also shows the boundary conditions for other parts of the model.

Uniform initial conditions were applied to the complete mesh. The initial void ratio is e_0, the initial stress is hydrostatic and the initial mean effective pressure is p_0.

4.4.3. Reference solutions for evolution of footing load

Numerical solutions with high accuracy were computed to serve as reference for the quantification of numerical errors. The explicit Algorithm 3 was employed for that, with a very strict error tolerance of $STOL = 10^{-9}$. The maximum vertical displacement $u_{\max} = 0.4\,\text{m}$ was applied in 10 000 load steps. This number is sufficiently large compared to the maximum value used in the actual computations, which is $n_{\text{steps}} = 1000$.

Initial void ratio and initial pressure were varied systematically, based on a reference set with $e_0 = 0.80$ and $p_0 = 0.1\,\text{MPa}$. At each calculation step k with total displacement u_k, the footing load $Q(u_k)$ was computed from the sum of vertical nodal forces along the footing. The resulting load displacement curves are displayed in Figure 12. With increasing soil density and increasing initial stress, the curves become steeper and the maximum footing load grows. This illustrates that the impact of e_0 and p_0 on the footing's behaviour can be simulated in a plausible way. The results are further used in the following sections.

4.4.4. Effect of tolerances on accuracy and efficiency

In order to obtain numerical solutions with different levels of accuracy, the number of global load steps for the maximum settlement $u_{\max} = 0.4\,\text{m}$ was varied. Specifically, the testing programme includes calculations with $n_{\text{steps}} = 40/80/200/400/1000$.

All simulations were run on an Intel® Core$^{\text{TM}}$ I5-8250U machine with 8 GB of memory, requesting two logical cores in Shared Memory mode. The CPU times which were required to simulate a complete loading process are used to quantify the computational costs.

The accuracy of the solutions is determined as follows. Let Q_{exa} be a reference footing load according to Section 4.4.3 and Q some approximate solution obtained with n_{steps} load steps. Then the overall accuracy of the approximate load displacement curve $Q(u_k)$ can be quantified through the relative error

$$E_Q = \sqrt{\sum_{k=1}^{n_{\text{steps}}} \frac{\Big(Q_{\text{exa}}(u_k) - Q(u_k)\Big)^2}{\Big(Q_{\text{exa}}(u_k)\Big)^2}}\,. \tag{56}$$

In Figure 13, computational costs are plotted against E_Q for simulations with the implicit and the explicit integrator and different values for $ITOL$ and $STOL$. The results are:

- Reducing $ITOL$ does not appear to improve accuracy considerably but increases the computational costs.
- Reducing $STOL$ improves accuracy while the CPU time increases relatively little.
- The calculations with the explicit algorithm are most efficient with values of n_{steps} between 80 and 200.
- Calculations with the explicit stress point algorithm are more efficient than calculations with the implicit algorithm.

Table 5: Exemplary data for points in Figure 13

Stress point algorithm	E_Q in %	CPU time in s
Implicit, $ITOL = 1.0$	0.33	60.9
Explicit, $STOL = 10^{-3}$	0.34	11.5
		Ratio of CPU times = 5.3
Implicit, $ITOL = 1.0$	0.07	281.8
Explicit, $STOL = 10^{-4}$	0.08	14.9
		Ratio of CPU times = 18.9

Exemplary results from Figure 13 have been compiled in Table 5. Accordingly, CPU time with the implicit algorithm is larger than with the explicit algorithm at a given level of accuracy by a factor between 5 and 19. This holds for a convergence tolerance of $ITOL = 1.0$, which is an unusually large value. For smaller tolerances, the efficiency obtained with the implicit stress point algorithm is even worse.

The results obtained here are in agreement with those reported for stress point calculations in Section 4.3.7. There is only one exception. For single strain increments, the convergence tolerance $ITOL$ influenced the implicit algorithm's efficiency very little. In contrast, a considerable effect was observed in solving the initial value problem. This is due to the fact that the large numbers of integration points and load steps which are employed for the finite element simulations amplify any phenomena which occur at the integration points.

4.4.5. Effect of initial state on accuracy and efficiency

Furthermore, the influence of different initial void ratios e_0 and initial pressures p_0 on the overall efficiency was investigated. The results, as presented in Figure 14 and Figure 15, indicate that e_0 and p_0 do not significantly influence accuracy and computational costs of the simulations. At first glance, this seems to contradict the results obtained from the numerical experiments at stress point level in Section 4.3.7. However, there is a plausible explanation.

In Section 4.3.7, accuracy and efficiency have been investigated for single time integration steps. The magnitude and the direction of the stress increment computed during these time steps change drastically with the initial conditions. This, in turn, affects the integrators' efficiency, as has already been discussed.

In contrast to that, the accuracy of the finite element simulations was measured with respect to complete load displacement curves. These curves do not change fundamentally across the range of initial conditions considered

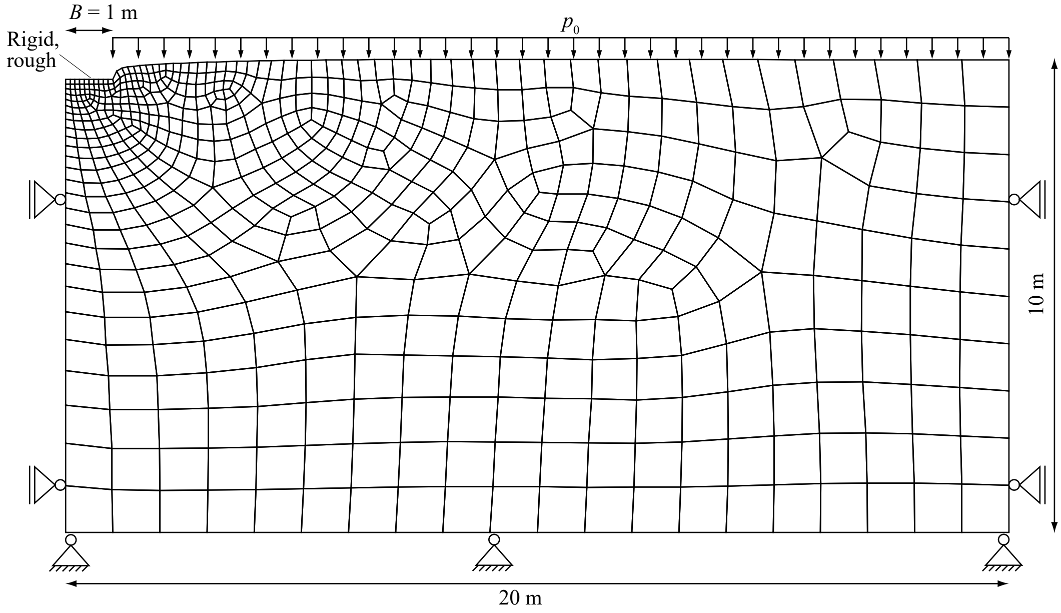


Figure 11: Finite element model, boundary conditions and deformed mesh at the final displacement $u_{\max} = 0.4\,\mathrm{m}$

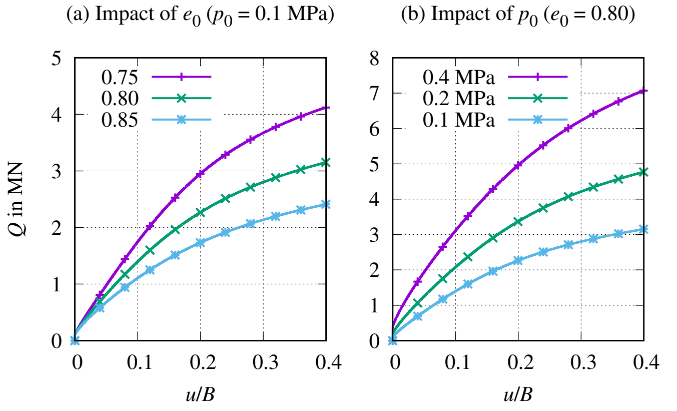


Figure 12: Reference load displacement curves of the strip footing for different initial conditions; obtained with explicit stress point Algorithm 3 ($STOL = 10^{-9}$, $n_{\mathrm{steps}} = 10\,000$)

here, as shown in Figure 12. That is why the efficiency of the numerical procedures in solving the strip footing problem is little affected by changes in e_0 and p_0.

4.5. Analysis of causes for the algorithms' performance

Bottom line, the finite element simulations clearly show that the explicit stress point algorithm is more efficient than the implicit one. A similar finding has already been described for the stress point calculations in Section 4.3.7. So the question arises, what are the causes of that result?

Presumably, the properties of the underlying rate equations play a crucial role. According to computational mathematics, there are two classes of differential equations, namely non-stiff problems and stiff problems. Non-stiff problems can be identified by the fact that they are most efficiently solved by explicit numerical methods. In

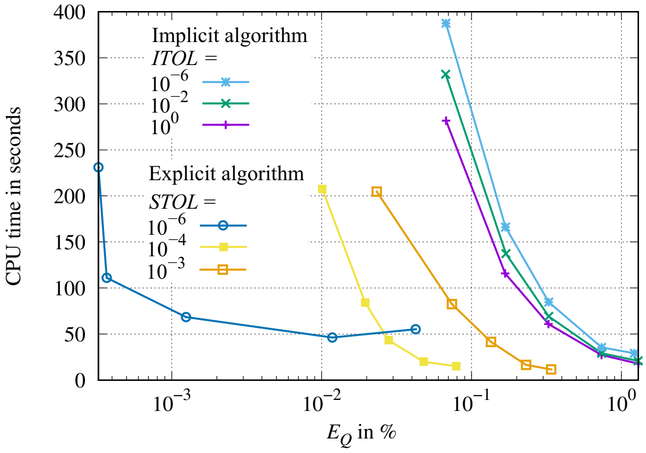


Figure 13: Effect of stress integration tolerances $ITOL$ and $STOL$ on numerical error and computational effort for the strip footing ($e_0 = 0.80$, $p_0 = 0.1\,\mathrm{MPa}$)

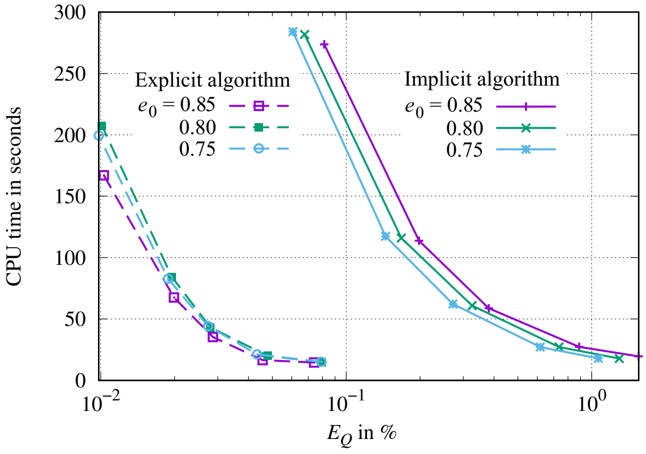


Figure 14: Effect of initial void ratio on numerical error and computation time for the strip footing ($STOL = 10^{-4}$, $ITOL = 1.0$)

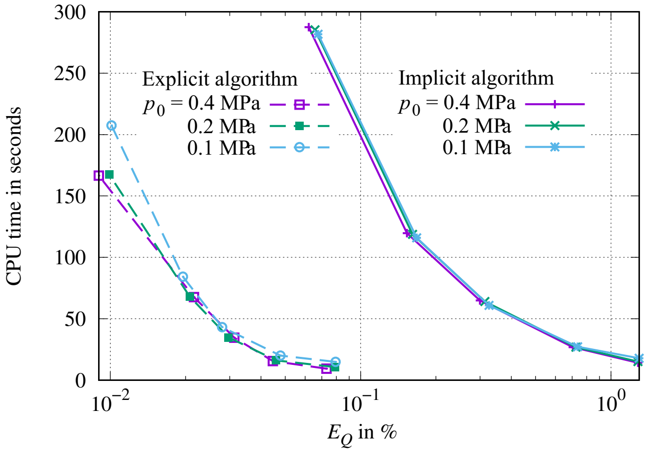


Figure 15: Effect of initial mean pressure on numerical error and computation time for the strip footing ($STOL = 10^{-4}$, $ITOL = 1.0$)

contrast, stiff equations demand implicit approximations, because these allow larger step sizes in situations where stability issues are encountered. Explicit methods could handle such situations only by means of very small time steps, which leads to inadequate efficiency (Hairer and Wanner, 1996).

Publications on soil models seldom seem to take into account the mathematical stiffness of the equations. Only for Karlsruhe Hypoplasticity some reports on that are available. They imply that the constitutive model behaves either stiff or non-stiff, depending on the initial state and the type of loading (Fellin et al., 2009, 2010; Fellin and Ostermann, 2014). Consequently, there could also be cases in which the constitutive model by Li (2002) is stiff enough for the efficiency of the explicit integrator to deteriorate significantly.

However, the references cited above further report for Karlsruhe Hypoplasticity that in typical geotechnical boundary value problems most of the elements behave non-stiff. Because of that, explicit time integration with error control turned out to yield superior efficiency overall. This is consistent with the results obtained here for the constitutive model by Li (2002). In light of that it appears plausible that the mathematical stiffness properties of the differential equations have contributed to the explicit method being more efficient than the implicit one in the present work.

Furthermore, there are some conceptual differences between the two algorithms presented here that apparently give the explicit approach an advantage:

- The accuracy order of the implicit Backward Euler method is smaller than that of the explicit Runge-Kutta approximation.
- The automatic error control of the explicit algorithm is very effective in boundary value problems where only small parts of the model require a fine discretization in time.
- The implicit integrator does not include any mechanism to effectively control the local numerical error.
- The local implicit iteration is more costly than the explicit substepping procedure in terms of computational effort.
- The numerical approximation of the Jacobian for the implicit iteration causes additional overhead, although it is said to require only slightly more computational costs than analytical derivatives (Pérez-Foguet et al., 2000a).

5. Conclusions

Within this paper, an implicit and an explicit integration algorithm for a bounding surface soil model without elastic range have been presented. Because the soil is assumed to be instantaneously plastic, no yield criterion had to be enforced by return mapping or drift correction procedures. The Jacobian of the local implicit iteration was approximated numerically, which proved to be straightforward to implement and sufficiently accurate to achieve quadratic convergence.

The stability as well as the accuracy of the implicit and the explicit algorithm have been investigated through numerical experiments at stress point level. The results are presented in Sections 4.3.1 to 4.3.5. They are summarised here as follows.

- Both methods are stable for equivalent strain increments of up to $\Delta\varepsilon_q = 1\,\%$.
- Even without return mapping, Backward Euler time integration still is more reliable than Forward Euler integration for large input strain increments.
- The accuracy of the explicit integrator is efficiently controlled by the automatic substepping procedure; the local numerical error does not exceed the predefined limit, even if input strain increments are very large.
- Due to the lack of a local error control, the accuracy of the implicit method is strictly limited by the global load step size; consequently, large strain steps yield large numerical errors.

Accordingly, the accuracy and stability attributes of the stress point algorithms are consistent with what could be expected from the properties of the underlying numerical methods. More interesting, then, is to assess the algorithms' efficiency at a given accuracy. To this end, further example calculations at stress point level and a comprehensive series of finite element simulations were performed. The results have been presented in Sections 4.3.7, 4.4.4 and 4.4.5. They clearly prove that the explicit algorithm is more efficient than the implicit one in terms of computational costs. The reasons for that have been analysed in Section 4.5 .

As stated in the introduction, this paper is intended to reveal which type of numerical integration scheme should be adopted for plasticity models without elastic range. This question can conclusively be answered in light of the findings summarised above: For practical applications, the explicit method with automatic error control should be adopted rather than implicit Backward Euler time integration.

One interesting point has not been investigated thoroughly. In this study, continuum tangents were used for the Jacobian of the global solution procedures, as noted in Section 4.1. This is simple to implement and saves computational resources at the integration point level, but may prevent the global solution from converging at its optimal rate. Faster convergence is said to be achieved if tangent stiffness operators are used which are consistent with the algorithmic stress update (Simo and Hughes, 1998; Neto et al., 2008). Unfortunately, the algorithms developed here do not allow to derive analytic expressions for consistent stiffness operators. First, due to the nature of the constitutive model, as discussed in Section 3.2.3. Second, because local substepping is used. However, it seems possible to obtain consistent tangents for combinations of complex soil models and substepping schemes from numerical approximations. This was demonstrated in (Pérez-Foguet et al., 2001) by employing numerical differentiation and in (Fellin and Ostermann, 2002), based on variational equations. Future research should explore whether such approaches can be applied to the problem investigated here and if this is more efficient than simply using continuum tangents.

Acknowledgement

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), grants SA 310/27-1 and SA 310/27-2. This support is gratefully acknowledged.

The authors wish to thank Prof. X. S. Li for initiating the work with the bounding surface model for sands and for contributing extensively to the long-term scientific exchange between Technische Universität Berlin and The Hong Kong University of Science and Technology.

The comments of the anonymous reviewers were very insightful and helped a lot to improve the manuscript. This is also gratefully acknowledged.

Appendix A. List of operators and symbol definitions

$(\boldsymbol{A}\cdot\boldsymbol{B})_{ij} = A_{ik}B_{kj}$ – Single contraction
$\boldsymbol{A}:\boldsymbol{B} = A_{ij}B_{ij}$ – Double contraction
$(\boldsymbol{A}\otimes\boldsymbol{B})_{ijkl} = A_{ij}B_{kl}$ – Tensor product
$\mathrm{tr}(\boldsymbol{A}) = A_{kk}$ – Trace
δ_{ij} – Kronecker delta
$\boldsymbol{I}$ – Second order unit tensor
$\boldsymbol{\emptyset}$ – Zero tensor
$\boldsymbol{A}_{\mathrm{dev}} = \boldsymbol{A} - (1/3)\,\mathrm{tr}(\boldsymbol{A})\boldsymbol{I}$ – Deviatoric tensor
$\|\boldsymbol{A}\| = \sqrt{\boldsymbol{A}:\boldsymbol{A}}$ – Euclidean norm
$\dot{\boldsymbol{A}}$ – Material time derivative
$\langle x\rangle = (x+|x|)/2$ – Macaulay brackets
$p = (1/3)\,\mathrm{tr}(\boldsymbol{\sigma})$ – Mean effective stress
$q = \sqrt{3/2}\|\boldsymbol{\sigma}_{\mathrm{dev}}\|$ – Equivalent shear stress
$\boldsymbol{r} = (\boldsymbol{\sigma} - p\boldsymbol{I})/p$ – Stress ratio tensor
$R = \sqrt{3/2}\|\boldsymbol{r}\|$ – Stress ratio invariant
$\theta = (1/3)\arcsin\left(-9r_{ij}r_{jk}r_{ki}/(2R^3)\right)$ – Lode angle
$\varepsilon_q = \sqrt{2/3}\|\boldsymbol{\varepsilon}_{\mathrm{dev}}\|$ – Equivalent shear strain

Appendix B. Differential constitutive equations

All formulas which were adopted from (Li, 2002) and have not been introduced in Section 2 are recalled here for completeness.

Variables with a superposed bar (e. g. $\bar{K}_{p1}$, $\bar{\boldsymbol{r}}$) are associated with a bounding surface. Variables without bar (e. g. $K_{\mathrm{p}1}$, $\boldsymbol{r}$) refer to the actual stress point.

Elastic stiffness:

$$(\mathbf{c}^{\mathrm{e}})_{ijkl} = K\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \tfrac{2}{3}\delta_{ij}\delta_{kl}) \qquad \text{(B.1)}$$

$$G = G_0 \frac{(2.97 - e)^2}{1 + e}\sqrt{p p_{\mathrm{a}}} \qquad \text{(B.2)}$$

$$K = G\frac{2\,(1+\nu)}{3\,(1-2\nu)} \qquad \text{(B.3)}$$

Elastoplastic stiffness, see also (15) and (16):

$$\begin{aligned}\mathbf{c}^{\mathrm{ep}} = \mathbf{c}^{\mathrm{e}} - \mathbf{c}^{\mathrm{e}} : \Big[\Big(\sqrt{2/27}D_1\boldsymbol{I} + \bar{\boldsymbol{n}}\Big) \otimes \boldsymbol{\Theta} \\ - \Big(\sqrt{2/27}D_2\boldsymbol{I} + \bar{\boldsymbol{m}}\Big) \otimes \boldsymbol{Z}\Big]\end{aligned} \qquad \text{(B.4)}$$

$$\boldsymbol{\Theta} = \frac{2G\bar{\boldsymbol{n}} - K\left(\bar{\boldsymbol{n}} : \boldsymbol{r} + B\right)\boldsymbol{I}}{2G - \sqrt{2/3}KD_1\left(\bar{\boldsymbol{n}} : \boldsymbol{r} + B\right) + K_{\mathrm{p}1}} \qquad \text{(B.5)}$$

$$\boldsymbol{Z} = \frac{K\boldsymbol{I} - \sqrt{2/3}KD_1\boldsymbol{\Theta}}{\sqrt{2/3}KD_2 + K_{\mathrm{p}2}} \qquad \text{(B.6)}$$

$$B = \frac{2G\bar{\boldsymbol{n}} : \bar{\boldsymbol{m}} - \sqrt{2/3}KD_2\bar{\boldsymbol{n}} : \boldsymbol{r}}{\sqrt{2/3}KD_2 + K_{\mathrm{p}2}} \qquad \text{(B.7)}$$

Evolution of void ratio (Muir Wood, 1990, p. 12):

$$-\dot{e} = (1+e)\,\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}) \qquad \text{(B.8)}$$

Unit flow directions:

$$\bar{\boldsymbol{n}} = \frac{1}{A}\left(\frac{\partial F_1}{\partial \bar{\boldsymbol{r}}}\right)_{\mathrm{dev}}, \quad \bar{\boldsymbol{m}} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \qquad \text{(B.9)}$$

$$A = \left\|\left(\frac{\partial F_1}{\partial \bar{\boldsymbol{r}}}\right)_{\mathrm{dev}}\right\| \qquad \text{(B.10)}$$

$$\text{(B.11)}$$

Shear mode interpolation function:

$$g(\bar{\theta}) = \frac{\sqrt{(1+c^2)^2 + 4c(1-c^2)\sin 3\bar{\theta}} - (1+c^2)}{2(1-c)\sin 3\bar{\theta}} \qquad \text{(B.12)}$$

$$\frac{\partial g(\bar{\theta})}{\partial \sin 3\bar{\theta}} = \frac{c(1+c)}{\sin 3\bar{\theta}\sqrt{(1+c^2)^2 + 4c(1-c^2)\sin 3\bar{\theta}}} - \frac{g(\bar{\theta})}{\sin 3\bar{\theta}} \qquad \text{(B.13)}$$

Normal to cone bounding surface:

$$\begin{aligned}\frac{\partial F_1}{\partial \bar{r}_{ij}} = \frac{3}{2\bar{R}^2 g^2(\bar{\theta})}\Bigg\{\left[\bar{R}g(\bar{\theta}) + 3\bar{R}\sin 3\bar{\theta}\frac{\partial g(\bar{\theta})}{\partial \sin 3\bar{\theta}}\right]\bar{r}_{ij} \\ + 9\frac{\partial g(\bar{\theta})}{\partial \sin 3\bar{\theta}}\bar{r}_{im}\bar{r}_{jm}\Bigg\}\end{aligned} \qquad \text{(B.14)}$$

Plastic modulus of cone:

$$\bar{K}_{\mathrm{p}1} = \frac{Gh}{\bar{R}}\left[M_{\mathrm{c}}g(\bar{\theta})\exp(-n\psi) - \bar{R}\right] \qquad \text{(B.15)}$$

$$K_{\mathrm{p}1} = \frac{Gh}{\bar{R}}\left[M_{\mathrm{c}}g(\bar{\theta})\exp(-n\psi)\left(\frac{\bar{\rho}_1}{\rho_1}\right) - \bar{R}\right] \qquad \text{(B.16)}$$

$$h = (h_1 - h_2 e)\left\{\left(\frac{\rho_1}{\bar{\rho}_1}\right)^{10} + h_3 f(\lambda_1)[1 - \left(\frac{\rho_1}{\bar{\rho}_1}\right)^{10}]\right\} \qquad \text{(B.17)}$$

$$f(\lambda_1) = \frac{1 - 0.01}{\sqrt{(1 - \frac{\lambda_1}{0.005})^2 + \frac{\lambda_1}{0.02}}} + 0.01 \qquad \text{(B.18)}$$

Plastic modulus of cap:

$$K_{\mathrm{p}2} = Gh_4\left(\frac{M_{\mathrm{c}}g(\theta)}{R}\right)\left(\frac{\bar{\rho}_2}{\rho_2}\right)^a \operatorname{sgn}\dot{p} \qquad \text{(B.19)}$$

$$\bar{K}_{\mathrm{p}2} = \begin{cases} K_{\mathrm{p}2} & \text{if } p = \bar{p} \text{ and } \dot{p} > 0 \\ 0 & \text{otherwise}\end{cases} \qquad \text{(B.20)}$$

Dilatancy functions:

$$D_1 = \frac{d_1}{M_{\mathrm{c}}g(\theta)}\left[M_{\mathrm{c}}g(\theta)\exp(m\psi)\sqrt{\frac{\bar{\rho}_1}{\rho_1}} - R\right] \qquad \text{(B.21)}$$

$$D_2 = d_2\left\langle\frac{M_{\mathrm{c}}g(\theta)}{R} - 1\right\rangle \operatorname{sgn}\dot{p} \qquad \text{(B.22)}$$

References

Andrianopoulos, K.I., Papadimitriou, A.G., Bouckovalas, G.D., 2010. Bounding surface plasticity model for the seismic liquefaction analysis of geostructures. Soil Dyn. Earthq. Eng. 30, 895–911. https://doi.org/10.1016/j.soildyn.2010.04.001.

Aubram, D., 2017. Notes on rate equations in nonlinear continuum mechanics. http://arxiv.org/abs/1709.10048v2.

Been, K., Jefferies, M.G., 1985. A state parameter for sands. Géotechnique 35, 99–112. https://doi.org/10.1680/geot.1985.35.2.99.

Belytschko, T., Liu, W.K., Moran, B., Elkhodhary, K.I., 2014. Nonlinear Finite Elements for Continua and Structures, 2nd Edition. John Wiley & Sons Ltd.

Carow, C., Rackwitz, F., Savidis, S., 2017. Über ein elastoplastisches Stoffgesetz für zyklisch beanspruchten Sand. Bautechnik 94, 604–612. https://doi.org/10.1002/bate.201700042. Postprint available at https://doi.org/10.14279/depositonce-9298.

Dafalias, Y.F., 1986. Bounding Surface Plasticity. I: Mathematical Foundation and Hypoplasticity. J. Eng. Mech. 112, 966–987. https://doi.org/10.1061/(ASCE)0733-9399(1986)112:9(966).

Dafalias, Y.F., Taiebat, M., 2016. SANISAND-z: zero elastic range sand plasticity model. Géotechnique 66, 999–1013. https://doi.org/10.1680/jgeot.15.P.271.

de Borst, R., Heeres, O.M., 2002. A unified approach to the implicit integration of standard, non-standard and viscous plasticity models. Int. J. Numer. Anal. Methods Geomech. 26, 1059–1070. https://doi.org/10.1002/nag.234.

Ding, Y., Huang, W., Sheng, D., Sloan, S.W., 2015. Numerical study on finite element implementation of hypoplastic models. Comput. Geotech. 68, 78–90. https://doi.org/10.1016/j.compgeo.2015.04.003.

Fellin, W., Mittendorfer, M., Ostermann, A., 2009. Adaptive integration of constitutive rate equations. Comput. Geotech. 36, 698–708. https://doi.org/10.1016/j.compgeo.2008.11.006.

Fellin, W., Mittendorfer, M., Ostermann, A., 2010. Adaptive integration of hypoplasticity, in: Benz, T., Nordal, S. (Eds.), Numerical Methods in Geotechnical Engineering, CRC Press. pp. 15–20.

Fellin, W., Ostermann, A., 2002. Consistent tangent operators for constitutive rate equations. Int. J. Numer. Anal. Methods Geomech. 26, 1213–1233. https://doi.org/10.1002/nag.242.

Fellin, W., Ostermann, A., 2014. Constitutive Models in Finite Element Codes, in: Hofstetter, G. (Ed.), Computational Engineering. Springer, pp. 1–42. https://doi.org/10.1007/978-3-319-05933-4_1.

Gao, Z., Zhao, J., 2015. Constitutive Modeling of Anisotropic Sand Behavior in Monotonic and Cyclic Loading. J. Eng. Mech. 141, 04015017. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000907.

Hairer, E., Wanner, G., 1996. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Number 14 in Springer Series in Computational Mathematics, Springer. https://doi.org/10.1007/978-3-642-05221-7.

Jefferies, M.G., 1993. Nor-Sand: a simple critical state model for sand. Geotechniqué 43, 91–103. https://doi.org/10.1680/geot.1993.43.1.91.

Kan, M.E., Taiebat, H.A., 2014. On implementation of bounding surface plasticity models with no overshooting effect in solving boundary value problems. Comput. Geotech. 55, 103–116. https://doi.org/10.1016/j.compgeo.2013.08.006.

Khalili, N., Habte, M.A., Valliappan, S., 2005. A bounding surface plasticity model for cyclic loading of granular soils. Internat. J. Numer. Methods Engrg. 63, 1939–1960. https://doi.org/10.1002/nme.1351.

Kolymbas, D., 1991. An outline of hypoplasticity. Arch. Appl. Mech. 61, 141–151. https://doi.org/10.1007/BF00788048.

Li, X.S., 2002. A sand model with state-dependent dilatancy. Géotechnique 52, 173–186. https://doi.org/10.1680/geot.2002.52.3.173.

Li, X.S., Dafalias, Y.F., 2000. Dilatancy for cohesionless soils. Géotechnique 50, 449–460. https://doi.org/10.1680/geot.2000.50.4.449.

Li, X.S., Dafalias, Y.F., 2004. A constitutive framework for anisotropic sand including non-proportional loading. Géotechnique 54, 41–55. https://doi.org/10.1680/geot.2004.54.1.41.

Li, X.S., Wang, Y., 1998. Linear Representation of Steady-State Line. J. Geotech. Geoenviron. Eng. 124, 1215–1217. https://doi.org/10.1061/(ASCE)1090-0241(1998)124:12(1215).

Lloret-Cabot, M., Sloan, S.W., Sheng, D., Abbo, A.J., 2016. Error behaviour in explicit integration algorithms with automatic substepping. Internat. J. Numer. Methods Engrg. 108, 1030–1053. https://doi.org/10.1002/nme.5245.

Manzari, M.T., Dafalias, Y.F., 1997. A critical state two-surface plasticity model for sands. Géotechnique 47, 255–272. https://doi.org/10.1680/geot.1997.47.2.255.

Mira, P., Tonni, L., Pastor, M., Fernández Merodo, J.A.F., 2009. A generalized midpoint algorithm for the integration of a generalized plasticity model for sands. Internat. J. Numer. Methods Engrg. 77, 1201–1223. https://doi.org/10.1002/nme.2445.

Muir Wood, D., 1990. Soil Behaviour and Critical State Soil Mechanics. Cambridge University Press.

Muir Wood, D., Belkheir, K., Liu, D.F., 1994. Strain softening and state parameter for sand modelling. Géotechnique 44, 335–339. https://doi.org/10.1680/geot.1994.44.2.335.

Neto, E., Perić, D., Owen, D.R.J., 2008. Computational Methods for Plasticity : Theory and Applications. Wiley. https://doi.org/10.1002/9780470694626.

Niemunis, A., 2017. IncrementalDriver. Fortran source code of a program for the simulation of element tests at stress point level. Version: 12/2017. https://soilmodels.com/idriver/.

Niemunis, A., Herle, I., 1997. Hypoplastic model for cohesionless soils with elastic strain range. Mech. Cohes.-frict. Mat. 2, 279–299. https://doi.org/10.1002/(SICI)1099-1484(199710)2:4<279::AID-CFM29>3.0.CO;2-8.

Pastor, M., Zienkiewicz, O.C., Chan, A.H.C., 1990. Generalized plasticity and the modelling of soil behaviour. Int. J. Numer. Anal. Methods Geomech. 14, 151–190. https://doi.org/10.1002/nag.1610140302.

Pérez-Foguet, A., Rodríguez-Ferran, A., Huerta, A., 2000a. Numerical differentiation for local and global tangent operators in computational plasticity. Comput. Methods Appl. Mech. Engrg. 189, 277–296. https://doi.org/10.1016/S0045-7825(99)00296-0.

Pérez-Foguet, A., Rodríguez-Ferran, A., Huerta, A., 2000b. Numerical differentiation for non-trivial consistent tangent matrices: an application to the MRS-Lade model. Internat. J. Numer. Methods Engrg. 48, 159–184. https://doi.org/10.1002/(SICI)1097-0207(20000520)48:2<159::AID-NME871>3.0.CO;2-Y.

Pérez-Foguet, A., Rodríguez-Ferran, A., Huerta, A., 2001. Consistent tangent matrices for substepping schemes. Comput. Methods Appl. Mech. Engrg. 190, 4627–4647. https://doi.org/10.1016/S0045-7825(00)00336-4.

Petalas, A.L., Dafalias, Y.F., 2019. Implicit integration of incrementally non-linear, zero- elastic range, bounding surface plasticity. Comput. Geotech. 112, 386–402. https://doi.org/10.1016/j.compgeo.2019.04.009.

Potts, D.M., Zdravković, L., 2001. Finite element analysis in geotechnical engineering. Application. volume 2. Thomas Telford. https://doi.org/10.1680/feaigea.27831.

Richart, F.E., Woods, R.D., Hall, J.R., 1970. Vibrations of Soils and Foundations. Civil Engineering and Engineering Mechanics Series, Prentice-Hall.

Sadeghian, S., Namin, M.L., 2013. Using state parameter to improve numerical prediction of a generalized plasticity constitutive model. Computers & Geosciences 51, 255–268. https://doi.org/10.1016/j.cageo.2012.06.025.

Shi, Z., Finno, R.J., Buscarnera, G., 2018. A hybrid plastic flow rule for cyclically loaded clay. Comput. Geotech. 101, 65–79. https://doi.org/10.1016/j.compgeo.2018.04.018.

Shu, S., Muhunthan, B., Li, X.S., 2011. Numerical Simulation of the Influence of Initial State of Sand on Element Tests and Micropile Performance. Int. J. Geomech. 11, 370–380. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000095.

Simo, J.C., Hughes, T.J.R., 1998. Computational Inelasticity. volume 7 of *Interdisciplinary Applied Mathematics*. Springer. https://doi.org/10.1007/b98904.

Sloan, S.W., 1987. Substepping schemes for the numerical integration of elastoplastic stress-strain relations. Internat. J. Numer. Methods Engrg. 24, 893–911. https://doi.org/10.1002/nme.1620240505.

Sloan, S.W., Abbo, A.J., Sheng, D., 2001. Refined explicit integration of elastoplastic models with automatic error control. Eng. Comput. 18, 121–194. https://doi.org/10.1108/02644400110365842.

Tamagnini, C., Castellanza, R., Nova, R., 2002. A Generalized Backward Euler algorithm for the numerical integration of an isotropic hardening elastoplastic model for mechanical and chemical degradation of bonded geomaterials. Int. J. Numer. Anal. Methods Geomech. 26, 963–1004. https://doi.org/10.1002/nag.231.

Tamagnini, C., Viggiani, G., Chambon, R., Desrues, J., 2000. Evaluation of different strategies for the integration of hypoplastic constitutive equations: Application to the CLoE model. Mech. Cohes.-frict. Mat. 5, 263–289. https://doi.org/10.1002/(SICI)1099-1484(200005)5:4<263::AID-CFM93>3.0.CO;2-Q.

von Wolffersdorff, P.A., 1996. A hypoplastic relation for granular materials with a predefined limit state surface. Mech. Cohes.-frict. Mat. 1, 251–271. https://doi.org/10.1002/(SICI)1099-1484(199607)1:3<251::AID-CFM13>3.0.CO;2-3.

Wang, G., Xie, Y., 2014. Modified Bounding Surface Hypoplasticity Model for Sands under Cyclic Loading. J. Eng. Mech. 140, 91–101. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000654.

Wang, S., Wu, W., Peng, C., He, X., Cui, D., 2018. Numerical integration and FE implementation of a hypoplastic constitutive model. Acta Geotech. 13, 1265–1281. https://doi.org/10.1007/s11440-018-0684-z.

Wang, Z., Dafalias, Y.F., Shen, C., 1990. Bounding Surface Hypoplasticity Model for Sand. J. Eng. Mech. 116, 983–1001. https://doi.org/10.1061/(ASCE)0733-9399(1990)116:5(983).