A robust stress update algorithm for elastoplastic models without analytical derivation of the consistent tangent operator and loading/unloading estimation

Dechun Lu[a], Yaning Zhang[a], Xin Zhou[a]*, Cancan Su[a], Zhiwei Gao[b], Xiuli Du[a]

Abstract

A robust and concise implicit stress integration algorithm of elastoplastic models is presented. It does not require the loading/unloading estimation and analytical derivation operation for the stress update. First, the elastoplastic stress update problem is recast into an unconstrained minimization problem by utilizing the smooth function to bypass the loading/unloading estimation. Then, the object problem is solved by the line search method instead of the Newton method for better convergence. The consistent tangent operator is evaluated by the complex step derivative approximation without the subtraction cancellation error, which provides the quadratic convergence rate of global iteration. The rationality of the numerical consistent tangent operator is validated by the one obtained by the analytical derivation. A recently developed non-orthogonal elastoplastic (NEP) clay model is implemented using the new algorithm. The algorithm is confirmed through comparing the numerical solution and the analytical one for a cavity expansion problem. The algorithm performance is assessed based on a series of geotechnical boundary value problems. It is found that the new algorithm is more robust than the one employed by ABAQUS. The source code of the model implementation can be downloaded from https://github.com/zhouxin615.

KEYWORDS: Stress update algorithm; Line search method; Complex step derivative approximation; Smooth function; Consistent tangent operator; Constitutive model

[a]Institute of Geotechnical and Underground Engineering, Beijing University of Technology, Beijing 100124, China;

[b]James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK.

*Corresponding author, E-mail address: zhouxin@emails.bjut.edu.cn

Nomenclature

$\mathbf{s}$, $\boldsymbol{\sigma}$	deviatoric stress tensor, stress tensor
q, p	generalized shear stress, hydrostatic pressure
p_{c}	yield surface size
$\boldsymbol{\varepsilon}$, $\boldsymbol{\gamma}$, ε_{v}	total and deviatoric strain tensors, total volume strain
$\boldsymbol{\varepsilon}^{\mathrm{p}}$, $\boldsymbol{\gamma}^{\mathrm{p}}$, $\varepsilon_{\mathrm{v}}^{\mathrm{p}}$	plastic strain tensor, deviatoric plastic strain tensor, plastic volume strain
ν	Poisson's ratio
K, G, E	bulk, shear, and Young's moduli
$\mathbf{D}$	elastic stiffness tensor
f	yield function
λ, κ	compression and swelling indexes in the e-ln$p$ plane
e_0, e_1	initial void ratio and one at $p = 1\,\mathrm{kPa}$
c_κ, c_{p}	$c_\kappa = (1+e_0)/\kappa$ and $c_{\mathrm{p}} = (1+e_0)/(\lambda-\kappa)$ for convenience in writing.
$\Delta\Pi$	vertical distance between the NCL and the CSL in the e-ln$p$ plane
M	slope of the critical state line in triaxial compression conditions
N	shape parameter of the elliptical yield curve
μ	fractional order
$\mathbf{1}$, $\mathbf{I}$	second-order and fourth-order unit tensors
$\mathbf{I}^{\mathrm{vol}}$, $\mathbf{I}^{\mathrm{sym}}$	volumetric and symmetric parts of $\mathbf{I}$
h	perturbation value
$\mathrm{d}\phi$	plastic multiplier
ψ	merit function
c_{d}, β	dimensional parameter, smoothing parameter in the smoothing function
ρ, ς	parameters of line search method
α, $\mathbf{d}$	size and direction of search step

1. INTRODUCTION

The stress update problem of elastoplastic models is an initial value problem of the ordinary differential equations (ODEs) constrained by inequalities. The ODEs are usually transformed into algebraic equations to solve based on the explicit[1,2] or implicit[3-6] integral schemes. The implicit algorithm requires the Jacobian matrix in the local stress update iteration, which can be difficult to derive, especially for sophisticated soil models. But it is still preferred because it preserves the quadratic convergence rate of global iteration[7-10].

The most popular implicit stress updating algorithm may be the return-mapping algorithm where the operator splitting technique addresses the inequality constraints and the Newton method[11] solves the nonlinear equations[12]. This computational paradigm is also followed by the cutting plane algorithm[13] and the semi-implicit algorithm[14] and has been used widely in the numerical implementation of advanced soil models[15,16]. But it is found that the iterations may not converge for the Newton method when the initial value is far from the final solution or the problem is highly nonlinear due to complex model formulations[17,18]. The loading/unloading estimation of the operator splitting technique also makes the stress update procedure more cumbersome. Therefore, attempts have been made to improve the efficiency of the implicit stress integration method. For instance, one can use the smoothing function to replace the inequality constraints[19-23] or the penalty function[24] to bypass loading/unloading judgment. The nonlinear equations can be solved by the homotopy method[25], the line search method[18,19], or the trust region method [19] instead of the Newton method. These three methods can achieve better convergence. The line search method, however, is a more cost-effective manner from the perspective of conciseness. Compared with the Newton method, it only adds a one-dimensional nonlinear problem about the optimal step size in search, since the trust region method requires optimizing the multidimensional search direction, and the homotopy method needs to solve homotopy equations to obtain a better initial value. Some contributions worthy of attention in this field can be found in the literature[18,26,21,27,28], which initially focused primarily on constitutive models for metal materials. Theoretically, these methods should also have great potential in implementing advanced soil models with complex formulations and deserves further study.

In the implicit model implementation, the derivation operation is required to determine the Jacobian matrix and the consistent tangent operator. The former is used for the solution of local nonlinear stress integral equations and the latter is used for global equilibrium iterations. The analytical derivatives of constitutive equations can be obtained easily for some simple cases, e.g., the Mises model, the Mohr-Coulomb model, and others. For elastoplastic soil models with highly nonlinear characteristics[29], however, the analytical derivation operation, especially for the consistent tangent operator, has become an increasingly cumbersome and even impossible task. Numerical differentiation may be preferred to analytical derivation because it avoids tedious algebraic work and is easy to implement[30]. There are three practical numerical differentiation methods: the finite difference method[30], the complex step derivative approximation (CSDA)[31], and the Hyper-dual step derivative approximation (HDSDA)[32]. The essence of these methods is to expand the object function on different types of number axes and truncate the higher-order term of the Taylor series to obtain the desired derivative term. The calculation results of HDSDA are almost equivalent to those of analytical derivation, but a lot of function overloading and operator overloading are required to define the operation rules of the Hyper-dual number[33]. For the finite difference method, there are two kinds of numerical errors, namely the truncation error and the rounding error dominated by subtraction cancellation error. The former can be reduced effectively with a small perturbation value (i.e., differential step size), but the latter will increase with the decrease of the perturbation value. It is often a prerequisite for the successful application of the finite difference method to determine an optimal perturbation value[34]. There is no subtraction cancellation error for the CSDA due to the absence of subtraction operation. On the other hand, the operation rules of the complex number have been added to mainstream programming languages. Therefore, the CSDA makes it possible for the concise and robust implementation of constitutive models.

This study aims to propose a robust and concise stress update algorithm to reduce the complexity in the implicit numerical implementation of advanced elastoplastic models and to improve its computational efficiency. The root of complexity is that the implicit algorithm needs to calculate the Jacobian matrix of nonlinear equations and consistent tangent stiffness, in which tedious derivative operations are required for complex elastoplastic models. Therefore, the proposed algorithm uses the CSDA method with high precision to obtain numerical derivatives instead of analytical derivatives. The loading/unloading estimation for the elastoplastic stress update problem is bypassed by using smooth functions. On the other hand, the main reason for limiting the computational efficiency of implicit algorithms is that the Newton method requires a small load step to ensure the convergence of the solution under strong nonlinear conditions. In the proposed algorithm, the line search strategy will be used to improve the computational efficiency of the algorithm, in which a larger load increment step is allowed. This paper is organized as follows: Section 2 gives the implicit integral scheme of the NEP clay model by the Backward Euler method. In Section 3, the complete stress update procedure is given. Section 4 is devoted to the determination of consistent tangent operator from the analytical and numerical perspectives, where different numerical schemes are discussed and compared. In Section 5, the robustness and accuracy of model implementation are assessed and validated by a series of boundary problems.

2. NON-ORTHOGONAL ELASTOPLASTIC (NEP) CLAY MODEL

The elastoplastic models with the non-orthogonal flow rule have piqued an increasing interest in modelling the mechanical behaviour of geomaterials in recent years, in which some salient material properties (e.g., the dilatancy[35,36], strain hardening/softening[37], and state-dependence[38]) can be captured by the fractional derivative. A potential function is not required because the direction of plastic flow is given by the fractional gradient of the yield function[39]. Though these models show excellent predictive capability for different soils[40,41], no research has been done on the numerical implementation. A NEP clay model established by Liang et al.[35] is employed for the algorithm validation because the consistent tangent operator and Jacobian matrix of this model can be analytically derived due to its relative simplicity.

2.1 Brief review of the model concept

The NEP clay model is developed based on the modified Cam-clay (MCC) model[42]. The basic equations of both models are presented in Table 1. They have the same elastic law and hardening law. The elastic stiffness matrix is expressed as

$$\mathbf{D} = K\mathbf{1}\otimes\mathbf{1} + 2Kr\left(\mathbf{I} - \frac{1}{3}\mathbf{1}\otimes\mathbf{1}\right) \tag{1}$$

where $\mathbf{I}$ and $\mathbf{1}$ denote the fourth-order and second-order unit tensors, respectively. $r = 1.5(1-2\nu)/(1+\nu)$ and ν is Poisson's ratio. The bulk modulus K depends on the hydrostatic pressure p:

$$K = c_\kappa p \tag{2}$$

where $c_\kappa = (1+e_0)/\kappa$, e_0 denotes the initial void ratio. The parameters κ and λ in Table 1 are the swelling and compression indexes of soil in isotropic consolidation conditions, respectively. The calibration method for κ and λ is presented in Section 2.2.

In the NEP clay model, the fractional gradient $\left(\frac{\partial^\mu f}{\partial p^\mu}\frac{\partial p}{\partial \boldsymbol{\sigma}} + \frac{\partial^\mu f}{\partial q^\mu}\frac{\partial q}{\partial \boldsymbol{\sigma}}\right)$ of the yield function f is used as the plastic flow direction. q and μ represent the generalized shear stress and fractional order, respectively. In general, the Riemann-Liouville fractional derivative operator is employed. $\frac{\partial^\mu f}{\partial p^\mu}$ and $\frac{\partial^\mu f}{\partial q^\mu}$ are expressed as follows:

$$\frac{\partial^\mu f}{\partial p^\mu} = \frac{q^2}{p^\mu N^2 \Gamma(1-\mu)} + \frac{2p^{2-\mu}}{\Gamma(3-\mu)} - \frac{p_\mathrm{c} p^{1-\mu}}{\Gamma(2-\mu)} \tag{3}$$

$$\frac{\partial^\mu f}{\partial q^\mu} = \frac{2q^{2-\mu}}{N^2\Gamma(3-\mu)} + \frac{p(p-p_\mathrm{c})}{q^\mu \Gamma(1-\mu)} \tag{4}$$

where $\Gamma(\cdot)$ is the gamma function, p_c is the yield surface size. More details can refer to the literature[35,36,43].

The NEP clay model uses a different yield function (See Table 1) in which the ratio of vertical and horizontal axes of the elliptic yield curve at the meridian plane is defined by the parameter N. Based on the volume change condition $\mathrm{d}\varepsilon_{\mathrm{v}}^{\mathrm{p}} = 0$ at the critical state and Eq. (3), the relationship between parameters μ and N is obtained by:

$$N = M\sqrt{2-\mu} \tag{5}$$

where the parameter M (= q/p) is the critical state stress ratio in triaxial compression conditions. It should be emphasized that the shape of the yield curve is controlled by the parameter N, however, in the NEP clay model, N is not an independent material parameter and can be determined by μ and M (Eq. (5)). Therefore, for a given parameter M, the change of μ value will cause the change of N, leading to the change of the shape of the elliptical yield curves, as shown in Fig. 1 (a). From the perspective of model performance, the stiffness and dilatancy behaviour for different clays can be captured by an appropriate μ value as shown in Fig. 1 (b).

Table 1 Basic evaluation equations of two models

Basic equations	MCC model	NEP clay model
Hooke's law	$\mathrm{d}\boldsymbol{\sigma} = \mathbf{D} : \left(\mathrm{d}\boldsymbol{\varepsilon} - \mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}\right)$	$\mathrm{d}\boldsymbol{\sigma} = \mathbf{D} : \left(\mathrm{d}\boldsymbol{\varepsilon} - \mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}\right)$
Hardening law	$\mathrm{d}p_{\mathrm{c}} = \dfrac{1+e_0}{\lambda-\kappa} p_{\mathrm{c}} \mathrm{d}\varepsilon_{\mathrm{v}}^{\mathrm{p}}$	$\mathrm{d}p_{\mathrm{c}} = \dfrac{1+e_0}{\lambda-\kappa} p_{\mathrm{c}} \mathrm{d}\varepsilon_{\mathrm{v}}^{\mathrm{p}}$
Flow rule	$\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}} = \mathrm{d}\phi\left(\dfrac{\partial f}{\partial p}\dfrac{\partial p}{\partial \boldsymbol{\sigma}} + \dfrac{\partial f}{\partial q}\dfrac{\partial q}{\partial \boldsymbol{\sigma}}\right)$	$\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}} = \mathrm{d}\phi\left(\dfrac{\partial^{\mu} f}{\partial p^{\mu}}\dfrac{\partial p}{\partial \boldsymbol{\sigma}} + \dfrac{\partial^{\mu} f}{\partial q^{\mu}}\dfrac{\partial q}{\partial \boldsymbol{\sigma}}\right)$
Yield function	$f = q^2 + M^2\left(p^2 - p_{\mathrm{c}} p\right)$	$f = q^2 + N^2\left(p^2 - p_{\mathrm{c}} p\right)$
Karush-Kuhn-Tucker conditions	$\mathrm{d}\phi \geq 0,\ f \leq 0,\ \mathrm{d}\phi f = 0$	$\mathrm{d}\phi \geq 0,\ f \leq 0,\ \mathrm{d}\phi f = 0$
Model parameters	$M,\ \lambda,\ \kappa,\ \nu$	$M,\ \mu,\ \lambda,\ \kappa,\ \nu$

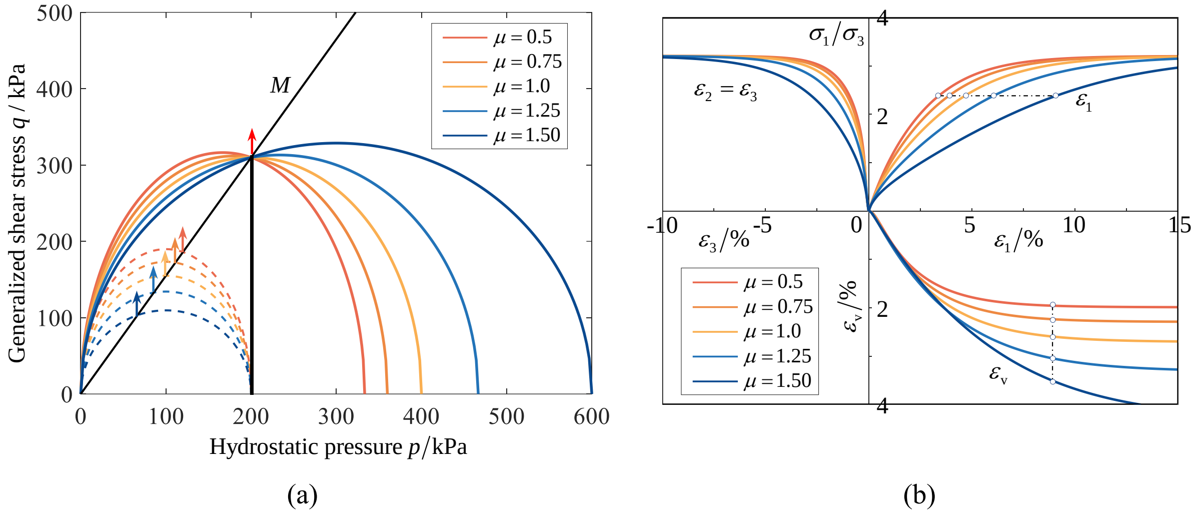


Fig. 1 NEP clay model: (a) yield curves; (b) stress-strain curves.

2.2 Parameter calibration and model validation

It can be seen from Table 1 that the NEP clay model has 5 material parameters, i.e., M, μ, λ, κ and ν, among which the calibration methods of parameters M, λ, κ and ν can refer to the MCC model. Only one more material parameter μ is added for the NEP clay model. Although the literature[35] has provided the detailed parameter calibration method of the NEP clay model, a brief review is necessary for the calibration method to facilitate the numerical application of the NEP clay model. First, the parameter $M = 6\sin\varphi/(3-\sin\varphi)$ can be determined by the internal friction angle φ in triaxial compression conditions. In the isotropic consolidation compression test, the swelling index κ and compression index λ can be determined by the slopes of swelling line (SWL: $e = e_S - \kappa \ln p$) and normally consolidated line (NCL: $e = e_N - \lambda \ln p$) in the e-ln$p$ plane, as shown in Fig. 2 (a). For the parameter calibration of μ, it is necessary to measure the vertical distance $\Delta\Pi$ between the critical state line (CSL) and the NCL in e-ln$p$ space. It can be seen from Fig. 2 that $\Delta\Pi$ is exactly equal to the plastic void ratio Δe^{p} caused by the triaxial compression path $A_0 A_0'$:

$$\Delta\Pi = (\lambda - \kappa)\ln\left(\frac{p_x}{p_0}\right) \tag{6}$$

where p_x/p_0 can be obtained by substituting stress point (p_0, Mp_0) into the yield function:

$$\frac{p_x}{p_0} = \frac{M^2}{N^2} + 1 \tag{7}$$

Substituting Eqs. (5) and (7) into Eq. (6), the parameter μ can be determined by:

$$\mu = 2 - \frac{1}{\exp\left(\frac{\Delta\Pi}{\lambda - \kappa}\right) - 1} \tag{8}$$

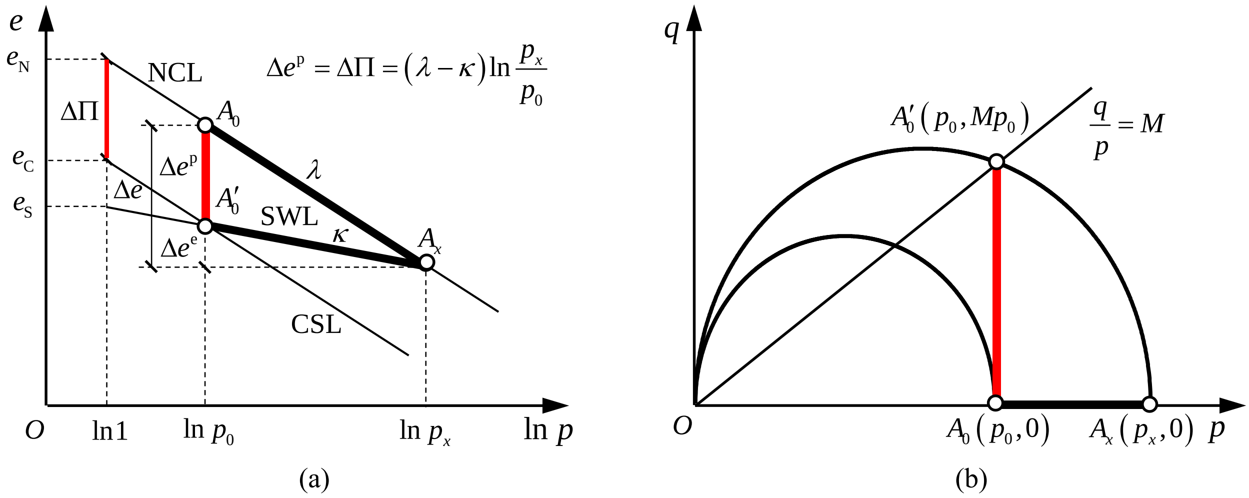
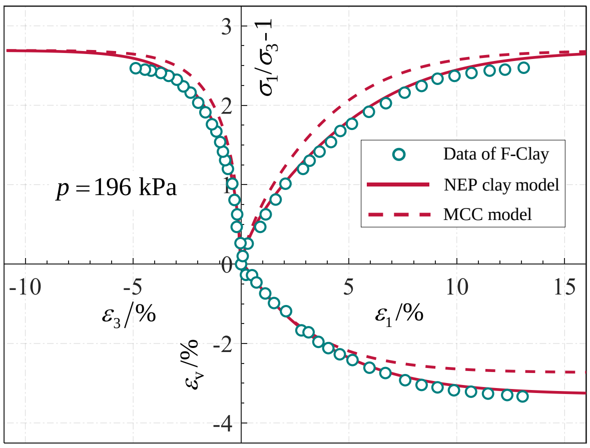


Fig. 2 Determination of parameter μ: (a) e-ln$p$ plane; (b) p-q plane.

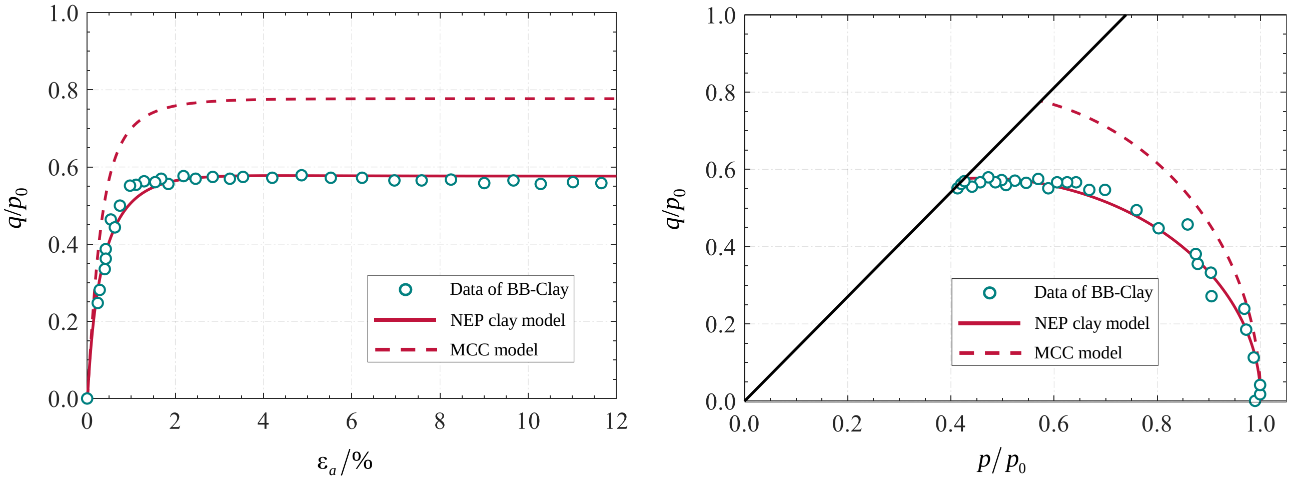
In what follows, the drained triaxial compression test of Fujinomori clay (F-clay) reported in literature[44] and the undrained triaxial compression test of Boston blue clay (BB-clay) reported in literature[45] are used to demonstrate the performance of NEP clay model. The material parameters are determined by the test data provided in the literature and the parameter calibration method mentioned above, as presented in Table 2. The test data of F-clay and the predicted curves of NEP clay model ($\mu = 1.23$) are illustrated in Fig. 3 (a), in which the results predicted by the MCC model ($\mu = 1.0$) are also presented. The prediction results from these two models finally reach the same stress ratio because the M-value for the two models is the same. However, the NEP clay model better describes the stress-strain behaviours of clay before the critical state, and reflects the deformation characteristics of soil with different stiffness by introducing fractional order μ. Fig. 3 (b) shows the model predictions and test data under undrained conditions, where ε_a is the axial strain. Comparing with the MCC model, the NEP clay model can more reasonably capture the strength and deformation characteristics of BB-clay under undrained conditions. The reason is that under undrained conditions, the stress path predicted by the model is influenced by the pore pressure which is closely associated with the volume change of soil, as shown in Fig. 3 (c). The NEP clay model can more reasonably describe the dilatancy law of soil by selecting a suitable μ-value.

Table 2 Material parameters of the NEP clay model

Material parameters	$\lambda/(1+e_0)$	$\kappa/(1+e_0)$	ν	M	μ
F-clay	0.0444	0.0047	0.3	1.36	1.23
BB-clay	0.0883	0.0173	0.1	1.35	1.47



(a)



b) (c)

Fig. 3 Model validation for: (a) F-clay; (b) stress-strain curve of BB-clay; (c) stress path of BB-clay.

2.3 Stress integral equations of the model

Table 1 presents the basic equations of the model defined in the form of the ODEs. In the model implementation, the ODEs need to be discretized into the algebraic equations for the time interval $\left[t_n,\ t_{n+1}\right]$. Based on the Backward Euler method, the control equations of NEP clay model are given by:

$$\begin{cases} \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \bar{\mathbf{D}}\mathbf{c}\left[\Delta\boldsymbol{\varepsilon}_{n+1} - \Delta\phi_{n+1}\left(\dfrac{\partial^\mu f_{n+1}}{\partial p_{n+1}^\mu}\dfrac{\partial p_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}} + \dfrac{\partial^\mu f_{n+1}}{\partial q_{n+1}^\mu}\dfrac{\partial q_{n+1}}{\partial \boldsymbol{\sigma}_{n+1}}\right)\right] \\ p_{\mathrm{c},\,n+1} = p_{\mathrm{c},\,n}\exp\left(c_{\mathrm{p}}\Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\,n+1}\right) \\ \Delta\phi_{n+1} \geq 0,\ f_{n+1} \leq 0,\ \Delta\phi_{n+1} f_{n+1} = 0 \end{cases} \tag{9}$$

where $c_{\mathrm{p}} = \left(1+e_0\right)/\left(\lambda-\kappa\right)$. $\bar{\mathbf{D}} = \mathbf{D}\left(\bar{K},\bar{G}\right)$ is the secant elastic stiffness tensor. The secant bulk modulus $\bar{K}$ is:

$$\bar{K} = \frac{p_n}{\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\ n+1}}\left[\exp\left(c_\kappa \Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\ n+1}\right)-1\right] \tag{10}$$

where $\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\ n+1}$ represents elastic volume strain increment and $\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\ n+1} = \Delta\varepsilon_{\mathrm{v},\ n+1} - \Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\ n+1}$. Eq. (9) contains 8 equality equations and 2 inequality constraints. The number of stress integral equations can be simplified to reduce the difficulty of the solution. The stress tensor $\boldsymbol{\sigma}_{n+1}$ can be decomposed into its isotropic part and deviatoric part:

$$\boldsymbol{\sigma}_{n+1} = p_{n+1}\mathbf{1} + \mathbf{s}_{n+1} \tag{11}$$

where $\mathbf{s}_{n+1}$ is the deviatoric stress tensor. p_{n+1} and $\mathbf{s}_{n+1}$ can be expressed as follows:

$$p_{n+1} = \frac{1}{3}\boldsymbol{\sigma}_{n+1} : \mathbf{1} = p_n + \bar{K}\left(\Delta\varepsilon_{\mathrm{v},\,n+1} - \Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\,n+1}\right) \tag{12}$$

$$\mathbf{s}_{n+1} = \mathbf{s}_n + 2\bar{G}(\Delta\boldsymbol{\gamma}_{n+1} - \Delta\boldsymbol{\gamma}^{\mathrm{p}}_{n+1}) \tag{13}$$

where $\bar{G} = r\bar{K}$ represents the secant shear modulus, $\Delta\boldsymbol{\gamma}_{n+1}$ and $\Delta\boldsymbol{\gamma}^{\mathrm{p}}_{n+1}$ denote the total deviatoric strain increment and its plastic part. $\Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\,n+1}$ and $\Delta\boldsymbol{\gamma}^{\mathrm{p}}_{n+1}$ are expressed by:

$$\Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\,n+1} = \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} : \mathbf{1} = \Delta\phi_{n+1}\left[\frac{q^2_{n+1}}{p^\mu_{n+1}N^2\ \Gamma(1-\mu)} + \frac{2p^{2-\mu}_{n+1}}{\Gamma\left(3-\mu\right)} - \frac{p_{\mathrm{c},\,n+1}p^{1-\mu}_{n+1}}{\Gamma\left(2-\mu\right)}\right] \tag{14}$$

$$\Delta\boldsymbol{\gamma}^{\mathrm{p}}_{n+1} = \mathrm{P} : \Delta\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} = 3\Delta\phi_{n+1}\mathbf{s}_{n+1}\left[\frac{q^{1-\mu}_{n+1}}{N^2\Gamma\left(3-\mu\right)} + \frac{p_{n+1}\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2q^{1+\mu}_{n+1}\Gamma\left(1-\mu\right)}\right] \tag{15}$$

where the fourth-order projection tensor $\mathbf{P}$ is determined by $\mathbf{P} = \mathbf{I} - \mathbf{1} \otimes \mathbf{1}/3$. Substituting Eq. (15) into Eq. (13), one can get the expression for $\mathbf{s}_{n+1}$:

$$\mathbf{s}_{n+1} = \frac{\mathbf{s}_{\mathbf{n}} + 2\bar{G}\boldsymbol{\Delta\gamma}_{\mathbf{n+1}}}{1+c} \tag{16}$$

where

$$c = 6\bar{G}\Delta\phi_{n+1}\left[\frac{q_{n+1}^{1-\mu}}{N^2\ \Gamma(3-\mu)} + \frac{p_{n+1}\left(p_{n+1} - p_{\mathrm{c,\,n+1}}\right)}{2\ q^{1+\mu}\Gamma(1-\mu)}\right] \tag{17}$$

Substituting Eq. (14) into Eq.(12) and considering Eq. (16), the update formulas of p_{n+1} and q_{n+1} can be obtained to replace that of $\boldsymbol{\sigma}_{n+1}$ in Eq. (9).

$$p_{n+1} = p_n \exp\left\{c_\kappa \Delta\varepsilon_{\mathrm{v},\,n+1} - c_\kappa \Delta\phi_{n+1}\left[\frac{q_{n+1}^2}{p_{n+1}^{\mu} N^2\ \Gamma(1-\mu)} + \frac{2p_{n+1}^{2-\mu}}{\Gamma(3-\mu)} - \frac{p_{\mathrm{c},\,n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)}\right]\right\} \tag{18}$$

$$q_{n+1} = \sqrt{\frac{3}{2}}\frac{\left\|\mathbf{s}_n + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|}{1+c} \tag{19}$$

Finally, the implicit stress integral equations of the NEP clay model can be simplified in the following form:

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix} = \begin{Bmatrix} p_{n+1} - p_n \exp\left(c_\kappa \Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}\right) = 0 \\ q_{n+1} - \sqrt{\frac{3}{2}}\frac{\left\|\mathbf{s}_n + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|}{1+c} = 0 \\ p_{\mathrm{c},\,n+1} - p_{\mathrm{c},\,n} \exp\left(c_{\mathrm{p}} \Delta\varepsilon^{\mathrm{p}}_{\mathrm{v},\,n+1}\right) = 0 \\ \Delta\phi_{n+1} \geq 0,\ f_{n+1} \leq 0,\ \Delta\phi_{n+1} f_{n+1} = 0 \end{Bmatrix} \tag{20}$$

where Eq. (20) contains only 4 equalities. Comparing with Eq. (9) containing 8 equalities, the number of nonlinear equations is significantly reduced.

3. UNCONSTRAINED IMPLICIT STRESS UPDATE BASED ON THE LINE SEARCH METHOD

Eq. (20) is non-smooth due to the existence of *KKT* conditions, i.e., Eq. $(20)_4$, where the inequality constraints mean that the nonlinear equations cannot be solved directly. In the operator splitting technique, the "*elastic prediction*", i.e., $\boldsymbol{\sigma}_{n+1}^{trial} = \boldsymbol{\sigma}_n + \bar{\mathbf{D}}\left(\bar{K}, \bar{G}\right) : \Delta\boldsymbol{\varepsilon}_{n+1}$, is conducted to estimate loading and unloading states of material to address the inequality constraints. If $\boldsymbol{\sigma}_{n+1}^{trial}$ is within the current yield surface, i.e., $f\left(\boldsymbol{\sigma}_{n+1}^{trial}\right) < 0$, the stress update follows the elastic Hooke's law and no plastic strain occurs in this step. On the other hand, if $\boldsymbol{\sigma}_{n+1}^{trial}$ exceeds the current yield surface, i.e., $f\left(\boldsymbol{\sigma}_{n+1}^{trial}\right)>0$, Eq. (20) is solved by using $f_{n+1}=0$ instead of Eq. $(20)_4$. The stress gradually iterates back from $\boldsymbol{\sigma}_{n+1}^{trial}$ to the true stress point. The solving process is also known as the "plastic correction". The loading/unloading estimation is required at each increment step, which increases the complexity of the model implementation. To this end, the *KKT conditions* in Eq. $(20)_4$ are replaced equivalently by the *Fischer-Burmeister* smooth function[19].

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix} = \begin{Bmatrix} p_{n+1} - p_n \exp\left(c_\kappa \Delta\varepsilon_{\mathrm{v},\, n+1}^{\mathrm{e}}\right) = 0 \\ q_{n+1} - \sqrt{\dfrac{3}{2}} \dfrac{\left\| \mathbf{s}_n + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1} \right\|}{1+c} = 0 \\ p_{\mathrm{c},\, n+1} - p_{\mathrm{c},\, n} \exp\left(c_{\mathrm{p}} \Delta\varepsilon_{\mathrm{v},\, n+1}^{\mathrm{p}}\right) = 0 \\ \sqrt{\left(c_{\mathrm{d}}\Delta\phi_{n+1}\right)^2 + f_{n+1}^2 + 2\beta} - c_{\mathrm{d}}\Delta\phi_{n+1} + f_{n+1} = 0 \end{Bmatrix} \tag{21}$$

where $c_{\mathrm{d}} = \left\| \boldsymbol{\sigma}_n + \bar{\mathbf{D}}\left(\bar{K}, \bar{G}\right) : \Delta\boldsymbol{\varepsilon}_{n+1} \right\|^3$ and $\beta = 0.5 FTOL^2$ are the parameters in the smoothing function Eq. $(21)_4$. $FTOL$ is the allowable error for judging the convergence of solutions of nonlinear equations. Fig. 4 shows that the smooth curve will gradually approximate *KKT conditions* as the parameter β decreases. There is no need for the loading/unloading estimation in solving Eq. (21). The calculation results exactly satisfy the *KKT* conditions when the solution of Eq. (21) converges.

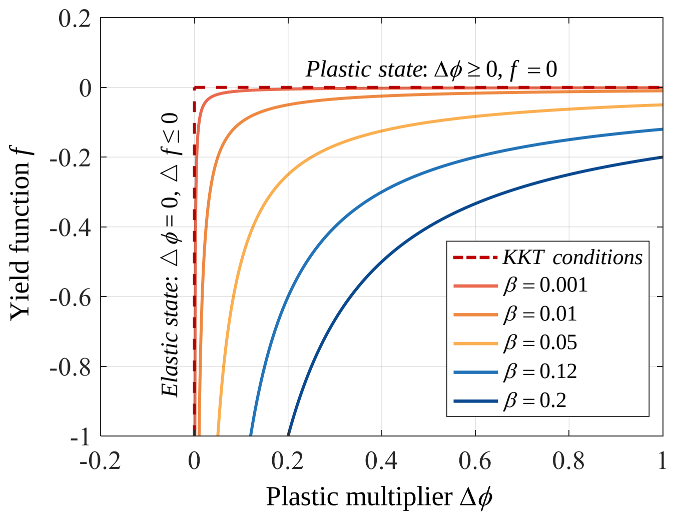


Fig. 4 *Fischer-Burmeister* smooth function.

By utilizing the smooth function, the elastoplastic stress update problem can be recast into a minimization problem:

$$\min \quad \psi\left(\{\mathbf{x}\}_{n+1}\right)=\frac{1}{2}\{\mathbf{f}(\mathbf{x})\}_{n+1}^{\ T}\{\mathbf{f}(\mathbf{x})\}_{n+1} \tag{22}$$

The decline of the merit function ψ can be achieved by the iterative search in multi-dimensional space:

$$\{\mathbf{x}\}^{k+1}=\{\mathbf{x}\}^{k}+\alpha^{k}\{\mathbf{d}\}^{k} \tag{23}$$

where $\{\mathbf{d}\}$ denotes the search direction, which is usually determined by the Newton direction $\{\mathbf{d}\}^{k}=-\left[\nabla\mathbf{f}(\mathbf{x})\right]_{k}^{-1}\{\mathbf{f}(\mathbf{x})\}^{k}$ to provide the quadratic convergence rate of local stress update iteration, where *k* denotes the iteration number in the local stress update. $\nabla\mathbf{f}(\mathbf{x})$ is the Jacobian matrix of nonlinear equations defined by Eq. (20), which can be calculated by the numerical differentiation. Appendix A also provides the analytical expression of $\nabla\mathbf{f}(\mathbf{x})$. α is the step size. The most essential task for the line search technique is to optimize the step size α^{k} to achieve the maximum benefit of minimizing ψ^{k} under a given search direction $\{\mathbf{d}\}^{k}$, which will further produce a one-dimensional sub problem to find α^{k}.

$$\min \quad \psi\left(\{\mathbf{x}+\alpha\mathbf{d}\}_{n+1}^{k}\right) \tag{24}$$

However, the exact minimization of Eq. (24) may be computationally expensive and is usually unnecessary. α^{k} is thus updated by a more practical iterative formula with *Goldstein's condition*:

$$\begin{cases} \mathit{Accept}\ \alpha_{j}^{k}\ \mathit{and\ exit} \qquad \mathit{IF}\ \ \psi\left(\alpha_{j}^{k}\right)<\left(1-2\rho\alpha_{j}^{k}\right)\psi(0) \\ \alpha_{j+1}^{k}=\dfrac{\psi(0)}{\psi(0)+2\psi\left(\alpha_{j}^{k}\right)} \qquad\qquad \mathit{ELSE} \end{cases} \tag{25}$$

where the initial value of α is set to 1. The updated step size α_{j+1}^{k} needs to be greater than a minimum value to avoid too small a benefit:

$$\alpha_{j+1}^{k}=\max\left\{\varsigma\alpha_{j}^{k},\frac{\psi(0)}{\psi(0)+2\psi\left(\alpha_{j}^{k}\right)}\right\} \tag{26}$$

where the algorithm parameters ρ and ς are recommended as 10^{-4} and 0.1[46]. Eqs. (25) and (26) essentially provide an inexact line search strategy, in which the step size for a given descent direction $\{\mathbf{d}\}^{k}$ is not a value that minimizes $\psi\left(\{\mathbf{x}+\alpha\mathbf{d}\}_{n+1}^{k}\right)$, but an acceptable range, as shown in Fig. 5. Finally, the complete stress update procedure of the NEP clay model is demonstrated in Fig. 6.

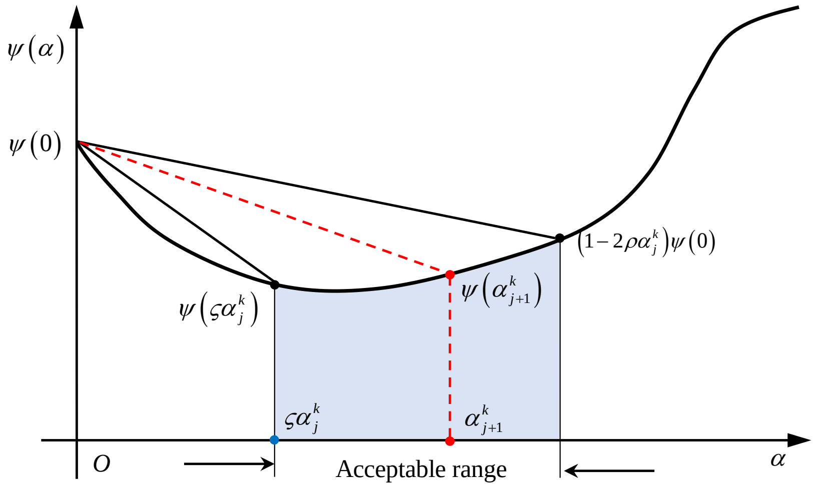


Fig. 5 Inexact line search method.

Input : $\boldsymbol{\sigma}_n,\ \Delta\boldsymbol{\varepsilon}_{n+1},\ p_{c,\,n}$

Compute $\boldsymbol{\sigma}_{n+1}^{trial}$ as the initial point of iteration and set $FTOL$

Set $k=0$, $\left\{p_{n+1}^0,\ q_{n+1}^0,\ p_{c,\,n+1}^0,\ \Delta\phi_{n+1}^0\right\}=\left\{p_{n+1}^{trial},\ q_{n+1}^{trial},\ p_{c,\,n},\ 0\right\}$ and $\beta = FTOL^2/2$, $c_d=\left\|\boldsymbol{\sigma}_{n+1}^{trial}\right\|^3$

Set $\{\mathbf{x}\}_{n+1}^0=\left\{p_{n+1}^0,\ q_{n+1}^0,\ p_{c,\,n+1}^0,\ \Delta\phi_{n+1}^0\right\}$ and compute $\psi^0=\psi\left(\{\mathbf{x}\}_{n+1}^0\right)$

do while $\left\|\mathbf{f}\left(\{\mathbf{x}\}^k\right)\right\|\le FTOL$ and $k\le k_{\max}$

Compute $\{\mathbf{x}^*\}_{n+1}^{k+1}$ using the line search method

$\{\mathbf{d}\}^k=-\left[\nabla\mathbf{f}(\mathbf{x})\right]_k^{-1}\{\mathbf{f}\}_{n+1}^k$, $j=0$, $\alpha_j^k=1$

$\psi(0)=\dfrac{1}{2}\left\|\{\mathbf{f}(\mathbf{x})\}_n\right\|^2$, $\psi\left(\alpha_j^k\right)=\dfrac{1}{2}\left\|\left\{\mathbf{f}\left(\mathbf{x}+\alpha_j^k\mathbf{d}\right)\right\}_{n+1}^k\right\|^2$

do while $\psi\left(\alpha_j^k\right)\ge\left(1-2\rho\alpha_j^k\right)\psi(0)$ and $j\le j_{\max}$

$$\alpha_{j+1}^k=\max\left\{\varsigma\alpha_j^k,\ \frac{\psi(0)}{\psi(0)+\psi\left(\alpha_j^k\right)}\right\}$$

$$\psi\left(\alpha_{j+1}^k\right)=\frac{1}{2}\left\|\left\{\mathbf{f}\left(\mathbf{x}+\alpha_{j+1}^k\mathbf{d}\right)\right\}_{n+1}^k\right\|^2$$

$j=j+1$

end do

$\{\mathbf{x}\}_{n+1}^{k+1}=\{\mathbf{x}\}_{n+1}^k+\alpha_j^k\{\mathbf{d}\}^k$

Set $k=k+1$

end do

State update and compute $\boldsymbol{\sigma}_{n+1}$

$p_{n+1}=p_{n+1}^k,\ q_{n+1}=q_{n+1}^k$, $p_{c,\,n+1}=p_{c,\,n+1}^k,\ \Delta\phi_{n+1}=\Delta\phi_{n+1}^k$

$\mathbf{s}_{n+1}=\dfrac{\mathbf{s}_n+2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}}{1+c}$, $\boldsymbol{\sigma}_{n+1}=p_{n+1}\mathbf{1}+\mathbf{s}_{n+1}$

Compute the consistent tangent operator $\partial\boldsymbol{\sigma}_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$

Output : $\boldsymbol{\sigma}_{n+1},\ p_{c,\,n+1}$, and $\partial\boldsymbol{\sigma}_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$

Fig. 6 Algorithm flow of a numerical implementation of the NEP clay model.

4. CONSISTENT TANGENT OPERATOR

The consistent tangent operator of NEP clay model is given by the analytical derivation and the numerical differentiation, respectively. The latter can be easily extended to the implicit calculation of other non-orthogonal models or elastoplastic models.

4.1 Analytical evaluation

The consistent tangent operator can be expressed by:

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = \mathbf{1} \otimes \frac{\partial p_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\partial \mathbf{s}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \tag{27}$$

Taking the derivatives of both Eq. (12) and Eq. (16) over $\boldsymbol{\varepsilon}_{n+1}$, one can obtain:

$$\frac{\partial p_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = a_1 K_{n+1}^{*} \mathbf{1} + a_2 K_{n+1}^{*} \frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + a_3 K_{n+1}^{*} \frac{\partial q_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \tag{28}$$

$$\begin{aligned} \frac{\partial \mathbf{s}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} &= 2a_8 \mathbf{R} \mathbf{:} \left(\bar{G}\mathbf{P} + a_1 a_9 \boldsymbol{\Delta\gamma}_{n+1} \otimes \mathbf{1} + a_2 a_9 \boldsymbol{\Delta\gamma}_{n+1} \otimes \frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) \\ &-2\sqrt{6} a_8 q_{n+1} \mathbf{R} : \hat{\mathbf{n}} \otimes \left[a_1 a_7 a_9 \Delta\phi_{n+1} \mathbf{1} + a_7 \left(a_2 a_9 \Delta\phi_{n+1} + \bar{G} \right) \frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + \frac{\bar{G} \Delta\phi_{n+1} K_{n+1}^{*}}{2 q_{n+1}^{1+\mu} \Gamma(1-\mu)} \left(c_1 \mathbf{1} + c_2 \frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right) \right] \end{aligned} \tag{29}$$

where $K_{n+1}^{*} = c_{\kappa} p_n \exp\left(c_{\kappa} \Delta\varepsilon_{\mathrm{v},\, n+1}^{\mathrm{e}} \right)$.

$$\begin{aligned} \mathbf{R} = \Bigg\{ \mathbf{I} + 6 a_8 q_{n+1} \Delta\varphi_{n+1} \Bigg[\bar{G} \frac{2 q_{n+1}^{-\mu} + N^2 \left(\mu^2 - \mu - 2 \right) \left(p_{n+1}^2 - p_{n+1} p_{\mathrm{c},\, n+1} \right) q_{n+1}^{-2-\mu}}{2N^2 (2-\mu) \Gamma(1-\mu)} \\ + \frac{\bar{G} K_{n+1}^{*} \left(2 p_{n+1} - p_{\mathrm{c},n+1} \right) a_3 - \bar{G} K_{n+1}^{*} p_{n+1} a_6}{2 q_{n+1}^{1+\mu} \Gamma(1-\mu)} + a_3 a_7 a_9 \Bigg] \hat{\mathbf{n}} \otimes \tilde{\mathbf{n}} - \sqrt{6} a_3 a_8 a_9 \boldsymbol{\Delta\gamma}_{\mathbf{n+1}} \otimes \tilde{\mathbf{n}} \Bigg\}^{-1} \end{aligned} \tag{30}$$

$$
\begin{cases}
a = 1 + K^*_{n+1}\Delta\phi_{n+1}\left(\dfrac{2p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{\mu q_{n+1}^2}{p_{n+1}^{1+\mu}N^2\ \Gamma(1-\mu)} - \dfrac{p_{\mathrm{c},\,n+1}}{p_{n+1}^{\mu}\Gamma(1-\mu)}\right) + \dfrac{p_{\mathrm{c},\,n+1}c_{\mathrm{p}}\Delta\phi_{n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} \\
a_1 = \left(1 + \dfrac{p_{\mathrm{c},\,n+1}c_{\mathrm{p}}\Delta\phi_{n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)}\right)\Big/ a \\
a_2 = \left(\dfrac{p_{\mathrm{c},\,n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{q_{n+1}^2}{p_{n+1}^{\mu}N^2\ \Gamma(1-\mu)} - \dfrac{2p_{n+1}^{2-\mu}}{\Gamma(3-\mu)}\right)\Big/ a \\
a_3 = \dfrac{-2\Delta\phi_{n+1}q_{n+1}}{a p_{n+1}^{\mu}N^2\ \Gamma(1-\mu)} \\
a_4 = p_{\mathrm{c},\,n+1}c_{\mathrm{p}}\Delta\phi_{n+1}\left(\dfrac{2p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{\mu q_{n+1}^2}{p_{n+1}^{1+\mu}N^2\ \Gamma(1-\mu)} - \dfrac{p_{\mathrm{c},\,\mathrm{n}+1}}{p_{n+1}^{\mu}\Gamma(1-\mu)}\right)\Big/ a \\
a_5 = p_{\mathrm{c},\,n+1}c_{\mathrm{p}}\left(\dfrac{q_{n+1}^2}{p_{n+1}^{\mu}N^2\ \Gamma(1-\mu)} + \dfrac{2p_{n+1}^{2-\mu}}{\Gamma(3-\mu)} - \dfrac{p_{\mathrm{c},\,n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)}\right)\Big/\left(aK^*_{n+1}\right) \\
a_6 = \left(2p_{\mathrm{c},\,n+1}c_{\mathrm{p}}\Delta\phi_{n+1}q_{n+1}\right)\Big/\left[p_{n+1}^{\mu}N^2\ K^*_{n+1}a\,\Gamma(1-\mu)\right] \\
a_7 = \dfrac{q_{n+1}^{1-\mu}}{N^2\ \Gamma(3-\mu)} + \dfrac{p_{n+1}^2 - p_{n+1}p_{\mathrm{c},\,n+1}}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)} \\
a_8 = \left(1 + 6\bar{G}\Delta\phi_{n+1}\ a_7\right)^{-1} \\
a_9 = r\left(K^* - \bar{K}\right)\Big/\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1} \\
c_1 = a_1(2p_{n+1} - p_{\mathrm{c},\,n+1}) - p_{n+1}a_4 \\
c_2 = a_2(2p_{n+1} - p_{\mathrm{c},\,n+1}) - p_{n+1}a_5
\end{cases}
\tag{31}
$$

$\dfrac{\partial p_{\mathrm{c},\,n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$ can be derived by taking Eq. $(9)_3$ with respect to $\boldsymbol{\varepsilon}_{n+1}$:

$$
\frac{\partial p_{\mathrm{c},\,n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = a_4 K^*_{n+1}\ \mathbf{1} + \ a_5 K^*_{n+1}\frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + a_6 K^*_{n+1}\frac{\partial q_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}
\tag{32}
$$

There are four unknowns (i.e., $\partial p_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$, $\partial \mathbf{s}_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$, $\partial p_{\mathrm{c},\,n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$, and $\partial\Delta\phi_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$) in Eqs. (28), (29), and (32). An additional constraint is needed to close the equations involving unknowns. The total differential of Eq. $(20)_4$ can yield:

$$
\frac{\partial f_4}{\partial f}\left[\frac{3\mathbf{s}_{n+1}}{M^2} : \frac{\partial \mathbf{s}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + \left(2p_{n+1} - p_{\mathrm{c},\,n+1}\right)\frac{\partial p_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} - p_{n+1}\frac{\partial p_{\mathrm{c},\,n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}\right] + \frac{\partial f_4}{\partial \Delta\phi_{n+1}}\frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = 0
\tag{33}
$$

From Eqs. (28), (29), (32), and (33), $\dfrac{\partial p_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$, $\dfrac{\partial \mathbf{s}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$, $\dfrac{\partial p_{\mathrm{c},\,n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$, and $\dfrac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$ can be derived as follows:

$$
\begin{aligned}
\frac{\partial p_{n+1}}{\partial \varepsilon_{n+1}} &= \left(a_1 + a_2 b_1\right)K^*_{n+1}\mathbf{1} + \left(a_2 b_2 + \sqrt{6}a_3 a_8 \bar{G}\right)K^*_{n+1}\tilde{\mathbf{n}} : \mathbf{R} : \mathbf{P} + \sqrt{6}a_3 a_8 a_9 K^*_{n+1}\left(a_1 + a_2 b_1\right)\tilde{\mathbf{n}} : \mathbf{R} : \Delta\boldsymbol{\gamma}_{n+1} \otimes \mathbf{1} \\
&+ \sqrt{6}a_2 a_3 a_8 a_9 b_2 K^*_{n+1}\tilde{\mathbf{n}} : \mathbf{R} : \boldsymbol{\Delta\gamma}_{n+1} \otimes \tilde{\mathbf{n}} : \mathbf{R} : \mathbf{P} - 6a_3 a_8 b_4 q_{n+1}K^*_{n+1}\tilde{\mathbf{n}} : \mathbf{R} : \hat{\mathbf{n}} \otimes \mathbf{1} - 6a_3 a_8 b_2 b_5 q_{n+1}K^*_{n+1}\tilde{\mathbf{n}} : \mathbf{R} : \hat{\mathbf{n}} \otimes \tilde{\mathbf{n}} : \mathbf{R} : \mathbf{P}
\end{aligned}
\tag{34}
$$

$$\frac{\partial \mathbf{s}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = 2a_8\bar{G}\mathbf{R}:\mathbf{P} + 2a_8a_9\left(a_1 + a_2b_1\right)\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\mathbf{1}$$
$$+2a_2a_8a_9b_2\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} - 2\sqrt{6}a_8b_4q_{n+1}\mathbf{R}:\hat{\mathbf{n}}\otimes\mathbf{1} - 2\sqrt{6}a_8b_2b_5q_{n+1}\mathbf{R}:\hat{\mathbf{n}}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} \tag{35}$$

$$\frac{\partial p_{\mathrm{c},\,n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = a_4K_{n+1}^{*}\,\mathbf{1} + a_5K_{n+1}^{*}\frac{\partial \Delta\phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} + a_6K_{n+1}^{*}\frac{\partial q_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \tag{36}$$

$$\frac{\partial \phi_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = b_1\mathbf{1} + b_2\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} \tag{37}$$

where the coefficients

$$\begin{cases} b = 2a_2a_8a_9b_3\tilde{\mathbf{n}}:\mathbf{R}:\Delta\boldsymbol{\gamma}_{\mathbf{n+1}}\dfrac{\partial f_4}{\partial f} - 2\sqrt{6}a_8b_3q_{n+1}\left\{a_2a_7a_9\Delta\phi_{n+1} + a_7\bar{G} + \dfrac{\bar{G}\Delta\phi_{n+1}K_{n+1}^{*}c_2}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)}\right\}\tilde{\mathbf{n}}:\mathbf{R}:\hat{\mathbf{n}}\dfrac{\partial f_4}{\partial f} + K_{n+1}^{*}c_2\dfrac{\partial f_4}{\partial f} + \dfrac{\partial f}{\partial \Delta\phi_{n+1}} \\ b_1 = -\dfrac{\partial f_4}{\partial f}\left[2a_1a_8a_9b_3\tilde{\mathbf{n}}:\mathbf{R}:\Delta\boldsymbol{\gamma}_{\mathbf{n+1}} - 2\sqrt{6}a_8b_3q_{n+1}\tilde{\mathbf{n}}:\mathbf{R}:\hat{\mathbf{n}}\left(a_1a_7a_9\Delta\phi_{n+1} + \dfrac{\bar{G}\Delta\phi_{n+1}K_{n+1}^{*}c_1}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)}\right) + K_{n+1}^{*}c_1\right]\Big/ b \\ b_2 = -\left(2a_8b_3\bar{G}\dfrac{\partial f_4}{\partial f}\right)\Big/ b \\ b_3 = \sqrt{6}\dfrac{q}{N^2} + \sqrt{\dfrac{3}{2}}K_{n+1}^{*}\left[\left(2p_{n+1} - p_{\mathrm{c},\,n+1}\right)a_3 - p_{n+1}a_6\right] \\ b_4 = a_1a_7a_9\Delta\phi_{n+1} + a_2a_7a_9b_1\Delta\phi_{n+1} + a_7b_1\bar{G} + \dfrac{\bar{G}\Delta\phi_{n+1}K_{n+1}^{*}\left(c_1 + b_1c_2\right)}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)} \\ b_5 = a_2a_7a_9\Delta\phi_{n+1} + a_7\bar{G} + \dfrac{\bar{G}\Delta\phi_{n+1}K_{n+1}^{*}c_2}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)} \end{cases} \tag{38}$$

Substituting Eqs. (34) and (35) into Eq. (27), the consistent tangent operator is obtained analytically as follows:

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = \left(a_1 + a_2b_1\right)K_{n+1}^{*}\mathbf{1}\otimes\mathbf{1} + \left(a_2b_2 + \sqrt{6}a_3a_8\bar{G}\right)K_{n+1}^{*}\mathbf{1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} + \sqrt{6}a_3a_8a_9\left(a_1 + a_2b_1\right)K_{n+1}^{*}\mathbf{1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\mathbf{1}$$
$$+\sqrt{6}a_2a_3a_8a_9b_2K_{n+1}^{*}\mathbf{1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} - 6a_3a_8b_4q_{n+1}K_{n+1}^{*}\mathbf{1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\hat{\mathbf{n}}\otimes\mathbf{1} - 6a_3a_8b_2b_5q_{n+1}K_{n+1}^{*}\mathbf{1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\hat{\mathbf{n}}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P}$$
$$+2\bar{G}a_8\mathbf{R}:\mathbf{P} + 2a_8a_9\left(a_1 + a_2b_1\right)\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\mathbf{1} + 2a_2a_8a_9b_2\mathbf{R}:\Delta\boldsymbol{\gamma}_{n+1}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} - 6\sqrt{\frac{2}{3}}a_8b_4q_{n+1}\mathbf{R}:\hat{\mathbf{n}}\otimes\mathbf{1} - 6\sqrt{\frac{2}{3}}a_8b_2b_5q_{n+1}\mathbf{R}:\hat{\mathbf{n}}\otimes\tilde{\mathbf{n}}:\mathbf{R}:\mathbf{P} \tag{39}$$

where $\dfrac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}}$ will degenerate into that of the MCC model presented in the literature [19] in the case of $\mu = 1$.

4.2 Numerical evaluation

Section 4.1 has provided the analytic consistent tangent operator. It can be observed that it is a cumbersome task to derive analytically the consistent tangent operator for the elastoplastic model with highly nonlinear characteristics. The verbose and complex expressions also make programming and code debugging more difficult. Therefore, the numerical evaluation is recommended from the perspective of implementation difficulty. In what follow, the CSDA is used to evaluate the derivatives of stress integral equations. As a comparison, the central difference method (CDM) and forward difference method (FDM) are also presented.

In the FDM, the derivative of $f\left(x\right)$ at the interesting point *x* is obtained by the Taylor expansion of $f\left(x+h\right)$ on the real number axis:

$$f\left(x+h\right)=f\left(x\right)+f'\left(x\right)h+\frac{f''\left(x\right)h^{2}}{2!}+\frac{f'''\left(x\right)h^{3}}{3!}+\ldots \tag{40}$$

where *h* denotes a smaller perturbation value. Assuming the truncation error terms can be neglected, one can yield:

$$f'\left(x\right)=\frac{f\left(x+h\right)-f\left(x\right)}{h}+O\left(h\right) \tag{41}$$

where $O\left(h\right)$ indicates that the FDM has first-order accuracy. Following a similar procedure. The Taylor expansion of $f\left(x-h\right)$ on the real number axis can yield:

$$f\left(x-h\right)=f\left(x\right)-f'\left(x\right)h+\frac{f''\left(x\right)h^{2}}{2!}-\frac{f'''\left(x\right)h^{3}}{3!}+\ldots \tag{42}$$

From Eqs. (40) and (42), the approximation of $f'\left(x\right)$ based on the CDM is obtained by:

$$f'\left(x\right)=\frac{f\left(x+h\right)-f\left(x-h\right)}{2h}+O\left(h^{2}\right) \tag{43}$$

There are two numerical errors in Eqs. (41) and (43). One is the truncation error which decreases with the decrease of *h*. The other is the rounding off error caused by representing real numbers with floating-point numbers of finite digits. It is worth emphasizing that the subtraction operation of two very close numbers will cause a significant subtractive cancellation error which is a special case of rounding off error and increases with the decrease of *h*. The error distribution of the finite difference method with the perturbation is shown in Fig. 7 (a). In the CSDA[47], the Taylor series expansion is conducted on both the real number and imaginary number axes:

$$f\left(x+h\right)=f\left(x\right)+f'\left(x\right)ih-\frac{f''\left(x\right)h^{2}}{2!}-\frac{f'''\left(x\right)ih^{3}}{3!}+\ldots \tag{44}$$

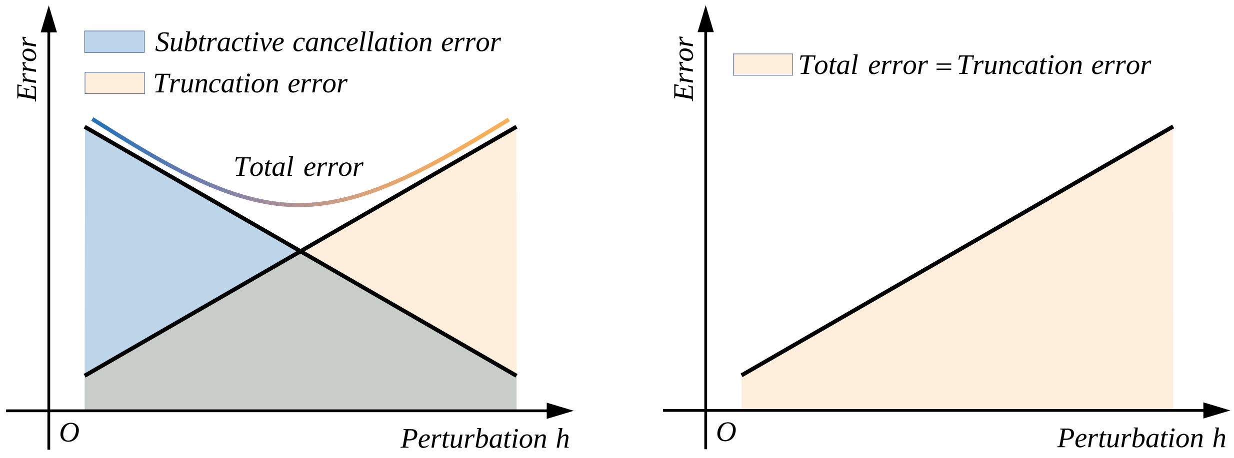
where i denotes the imaginary number ($i^{2}=-1$). The approximation formula of $f'\left(x\right)$ with second-order accuracy can be obtained by the division operation of the imaginary part of Eq. (44) as follows:

$$f'\left(x\right)=\frac{I\left[f\left(x+ih\right)\right]}{h}+O\left(h^{2}\right) \tag{45}$$

where $I\left[\cdot\right]$ is used to extract the imaginary part of the argument. The approximation formula in Eq. (45) can be easily extended to multi-dimensional cases as follows:

$$\frac{\partial f}{\partial x_{i}}=\frac{I\left[f\left(\mathbf{x}+ih_{i}\mathbf{e}_{i}\right)\right]}{h_{i}}+O\left(h_{i}^{2}\right) \tag{46}$$

where x_i denotes ith component of $\mathbf{x}$. $\mathbf{e}_i$ and h_i denote ith unit vector and the perturbation value in ith direction. It can be found that there is no subtraction operation in the CSDA. Therefore, there is no subtraction cancellation error and the rounding off error is bounded. On the other hand, the truncation error can be reduced by decreasing the perturbation value. In theory, there is no lower bound for the perturbation value in the CSDA[31]. Fig. 7 (b) demonstrates the error distribution of CSDA with the perturbation.



(a) (b)

Fig. 7 Errors change with the perturbation *h*: (a) FDM/CDM; (b) CSDA..

The derivative of $f(x)=\cos(x)/\left[1+\sin^2(x)\right]$ at $x=\pi/3$ is calculated as an example to further demonstrate the characteristics of three numerical schemes. The numerical examples are run on MATLAB 2020a. The range of the perturbation value *h* is [10^{-16}, 10^{-1}]. Fig. 8 (a) shows the change of relative total error with the perturbation value, in which the double-precision computation is conducted and the lower limit of relative total error is set to *eps* = 2.2204e-16 (double floating-point relative accuracy). It is clear that, at the beginning of decreasing perturbation, the decline rate of the relative total error of CSDA and CDM is approximately the same and faster than that of FDM since CSDA and CDM are second-order accuracy schemes while FDM is first-order accuracy scheme. With the further decrease of perturbation, the relative total errors of CDM and FDM begin to increase due to the presence of subtractive cancellation error while in the CSDA any perturbation value lower than 10^{-8} gives rise to a relative error near *eps*. A clearer contrast can be found in Fig. 8 (b), in which the precision of the variables is set to 100 bits. Therefore, it can be approximately considered that there is no rounding off error in the calculation results and the total error equals the truncation error.

The results depicted in Fig. 8 (b) show the decline rate of the relative truncation error of the three schemes is completely consistent with their accuracy orders of the numerical differentiation method. By subtracting the truncation error from the total error, one can obtain the change of the rounding off error dominated by the subtractive cancellation error with the perturbation, as shown in Fig. 8 (c). It is clear that the relative rounding off error of CSDA is maintained near *eps* owing to the lack of the subtractive cancellation error but the relative rounding off errors of CDM and FDM increase with decreasing perturbation. The CSDA provides a more robust numerical derivation scheme than the finite difference methods.

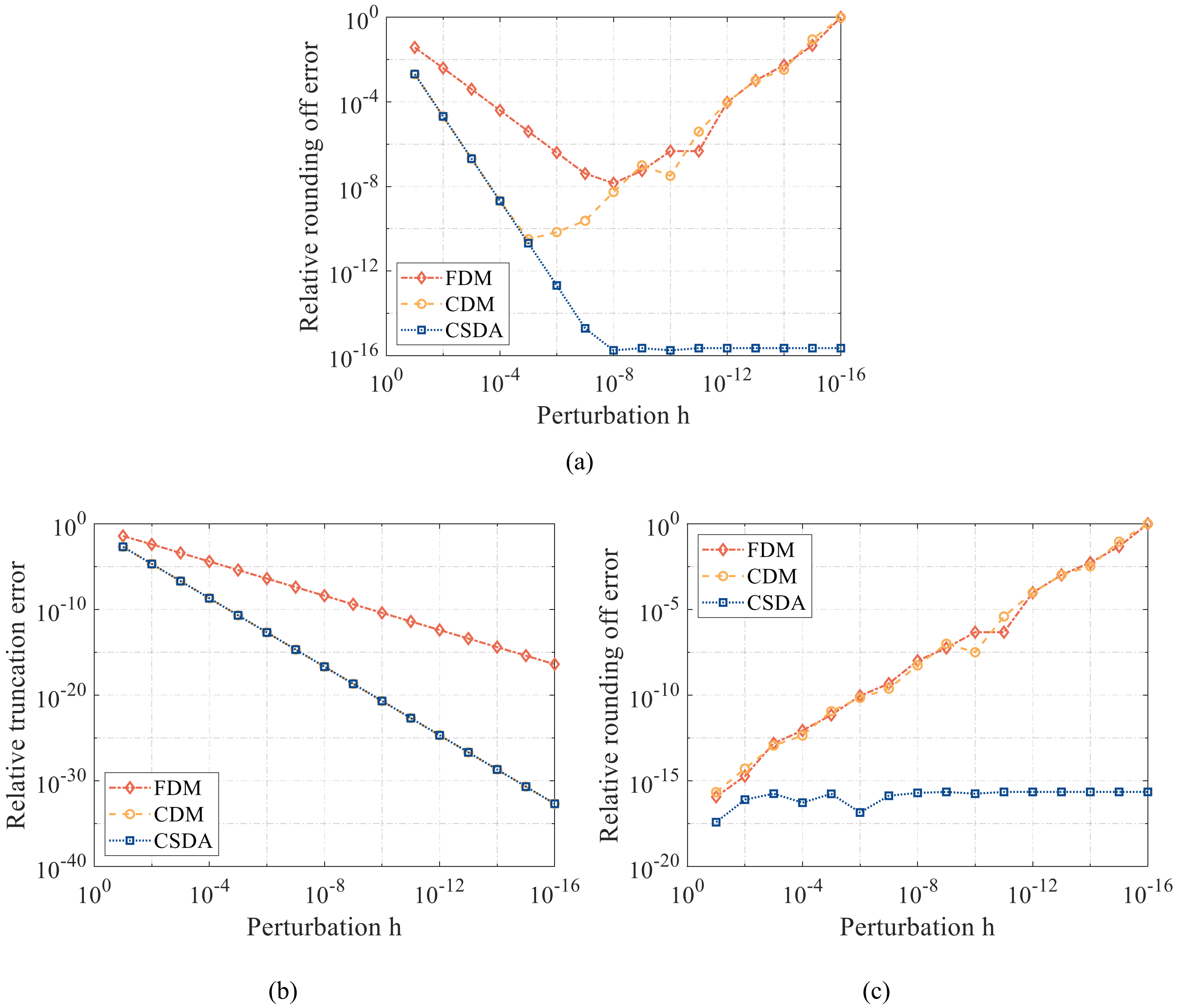


Fig. 8 Change of relative errors with the perturbation value: (a) relative total error; (b) relative truncation error; (c) relative rounding off error.

Based on the presented CSDA, the numerical Jacobian matrix of Eq. (20) can be easily obtained without cumbersome derivation. To numerically calculate the consistency tangent operator, however, the non-orthogonal stress integral equations usually need to be expressed in the following more general form:

$$\begin{Bmatrix} f_{\boldsymbol{\sigma}} \\ f_{p_\mathrm{c}} \\ f_{\Delta\phi} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n - \bar{\mathbf{D}} : \left(\Delta\boldsymbol{\varepsilon}_{n+1} - \Delta\boldsymbol{\varepsilon}^\mathrm{p}_{n+1} \right) \\ p_{\mathrm{c},\, n+1} - p_{\mathrm{c},\, n} \exp\left[c_\mathrm{p} \Delta\varepsilon^\mathrm{p}_{\mathrm{v},\, n+1} \right] \\ \sqrt{\left(c_\mathrm{d} \Delta\phi_{n+1} \right)^2 + f^2_{n+1} + 2\beta} - c_\mathrm{d}\Delta\phi_{n+1} + f_{n+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ 0 \\ 0 \end{Bmatrix} \tag{47}$$

Taking the total differential of Eq. (47) with independent variables $\boldsymbol{\sigma}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}$, $p_{\mathrm{c},\, n+1}$, and $\Delta\phi_{n+1}$, one can yield:

$$\begin{bmatrix} \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial \Delta\phi} \\ \dfrac{\partial f_{p_\mathrm{c}}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{p_\mathrm{c}}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{p_\mathrm{c}}}{\partial \Delta\phi} \\ \dfrac{\partial f_{\Delta\phi}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{\Delta\phi}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{\Delta\phi}}{\partial \Delta\phi} \end{bmatrix}_{n+1} \begin{Bmatrix} \mathrm{d}\boldsymbol{\sigma} \\ \mathrm{d}p_\mathrm{c} \\ \mathrm{d}\Delta\phi \end{Bmatrix}_{n+1} + \begin{bmatrix} \partial f_{\boldsymbol{\sigma}} / \partial \boldsymbol{\varepsilon} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathrm{d}\boldsymbol{\varepsilon} \\ 0 \\ 0 \end{Bmatrix}_{n+1} = \begin{Bmatrix} \mathbf{0} \\ 0 \\ 0 \end{Bmatrix} \tag{48}$$

where the Jacobian matrix of Eq. (47) can be solved by the numerical evaluation. After transposition and matrix inversion operations, one can obtain:

$$\begin{Bmatrix} \mathrm{d}\boldsymbol{\sigma} \\ \mathrm{d}p_\mathrm{c} \\ \mathrm{d}\Delta\phi \end{Bmatrix}_{n+1} = - \begin{bmatrix} \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{\boldsymbol{\sigma}}}{\partial \Delta\phi} \\ \dfrac{\partial f_{p_\mathrm{c}}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{p_\mathrm{c}}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{p_\mathrm{c}}}{\partial \Delta\phi} \\ \dfrac{\partial f_{\Delta\phi}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial f_{\Delta\phi}}{\partial p_\mathrm{c}} & \dfrac{\partial f_{\Delta\phi}}{\partial \Delta\phi} \end{bmatrix}^{-1}_{n+1} \begin{bmatrix} \partial f_{\boldsymbol{\sigma}} / \partial \boldsymbol{\varepsilon} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathrm{d}\boldsymbol{\varepsilon} \\ 0 \\ 0 \end{Bmatrix}_{n+1} = \left[\mathbf{A} \right] \begin{Bmatrix} \mathrm{d}\boldsymbol{\varepsilon} \\ 0 \\ 0 \end{Bmatrix}_{n+1} \tag{49}$$

where the consistent tangent operator is determined by extracting 6×6 upper-left block matrix of $\left[\mathbf{A} \right]$.

5. NUMERICAL VALIDATION

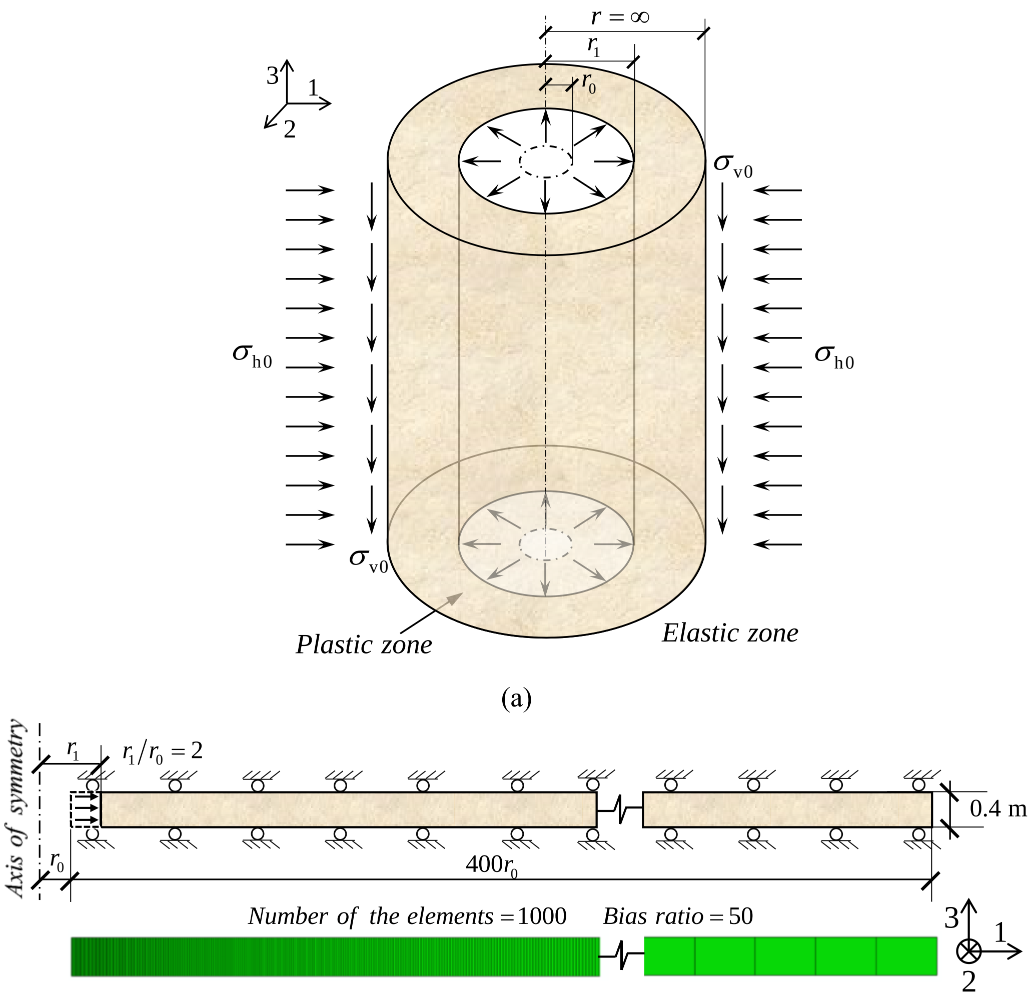
The NEP clay model can degenerate into the MCC model under the condition of $\mu = 1.0$. In this case, the accuracy of model implementation can be validated by the analytical solutions of the MCC model in a cylindrical cavity expansion problem. The robustness of model implementation can be also assessed by comparison with the MCC model that is available in the ABAQUS software. In what follows, the accuracy and performance of the presented model implementation are validated and evaluated. The influence of perturbation value on the numerical stability is also investigated. In particular, the ability of the model implementation to address the coupling geotechnical problem is also assessed by a pile foundation bearing capacity test under the undrained condition. These computations were made on Intel® Core(TM) i5-6200U processor 2.3 GHz processor running on a 64-bit Windows 10 operating system.

5.1 Cylindrical cavity expansion

The cylindrical cavity expansion is a typical axisymmetric problem, as shown in Fig. 9(a), and thus its mathematical description can be transformed into ODEs, which makes it possible to obtain the analytical solution to the problem. With the aid of scientific computing software Mathematica, Chen and Abousleiman have given the undrained[48] and drained[49] exact solutions of the MCC model in the case of cylindrical cavity expansion, which provides a valuable benchmark for the verification of the stress update results of the critical state models. Fig. 9(b) shows the simplified finite element model where the eight-node axisymmetric elements (CAX8) and the corresponding pore pressure elements (CAX8P) are employed respectively for the drained and undrained cases. It should be noted that the displacement of the right boundary in 1-direction is constrained in the undrained case and is free in the drained case. The initial state of soil and model parameters[48] are presented in Table 3 and Table 5. In particular, the permeability of soil and water weight are set to $2.3\times10^{-3}\ \mathrm{m/s}$ and $10\ \mathrm{kN/m^3}$ [50] for the undrained case, respectively. The total analysis time of $0.001\ \mathrm{s}$ is used to approximate undrained loading conditions. Fig. 10 shows the calculation results of the cylindrical cavity expansion problem under drained and undrained conditions, respectively. The over consolidation ratios (OCR) of examples is set to 10. It is clear that the numerical solution from the model implementation is in good agreement with the analytical solution.

Table 3 Initial state of soil in the cylindrical cavity expansion problem

OCR	σ'_{r0}	$\sigma'_{\theta 0}$	σ'_{z0}	p'_0	q'_0	e_0
10	144	144	72	120	72	0.802
Undrained case: permeability: $2.3\times10^{-3}\ \mathrm{m/s}$, water weight: $10\ \mathrm{kN/m^3}$.						



(b)

Fig. 9 Cylindrical cavity expansion example: (a) problem schematic; (b) simplified finite element model and mesh.

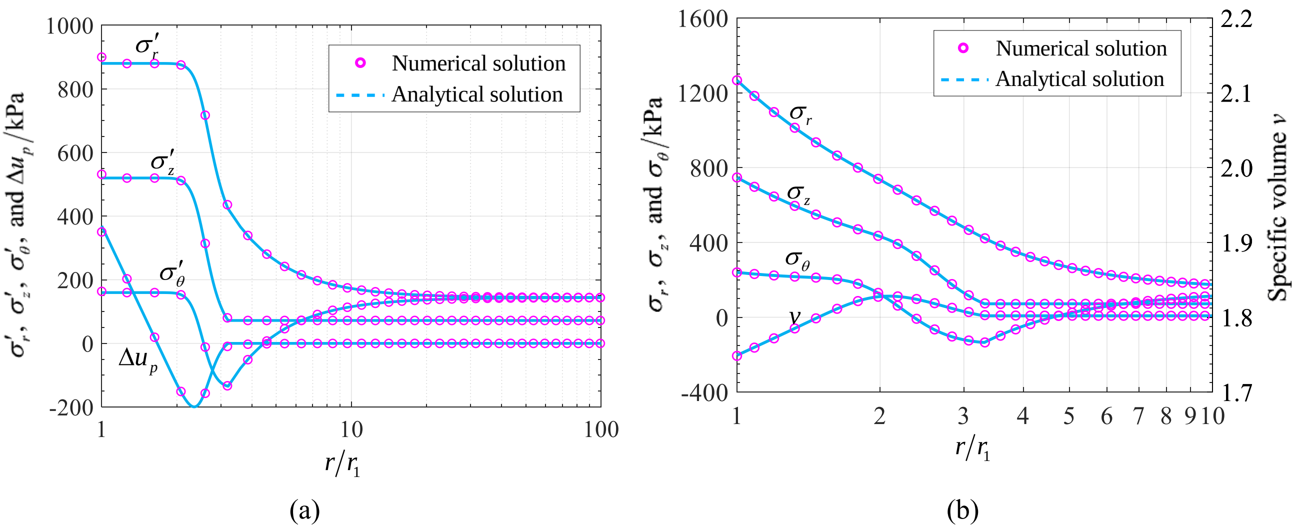
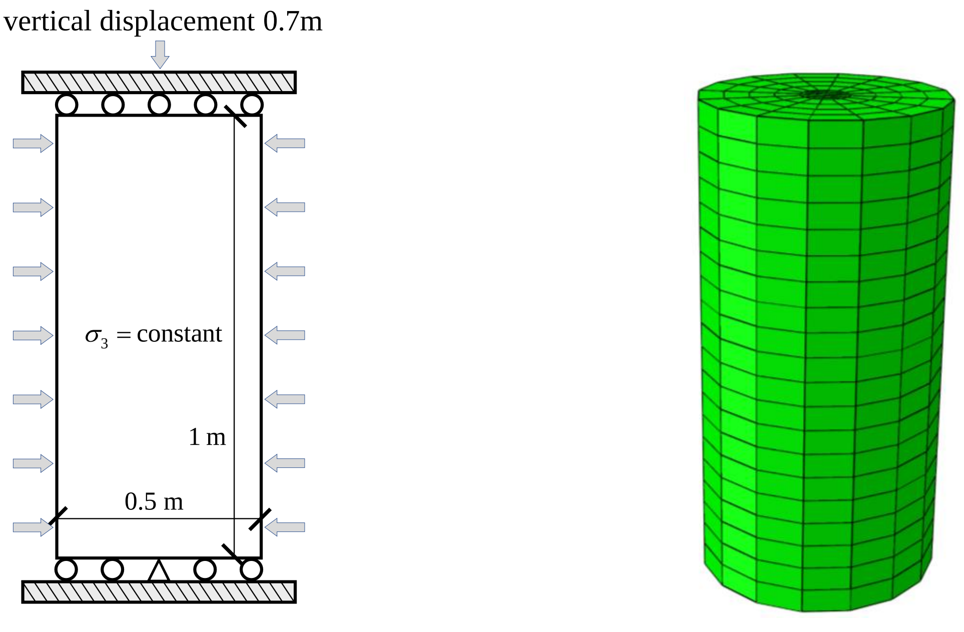


Fig. 10 Comparison between analytical and numerical solutions: (a) undrained condition; (b) drained condition.

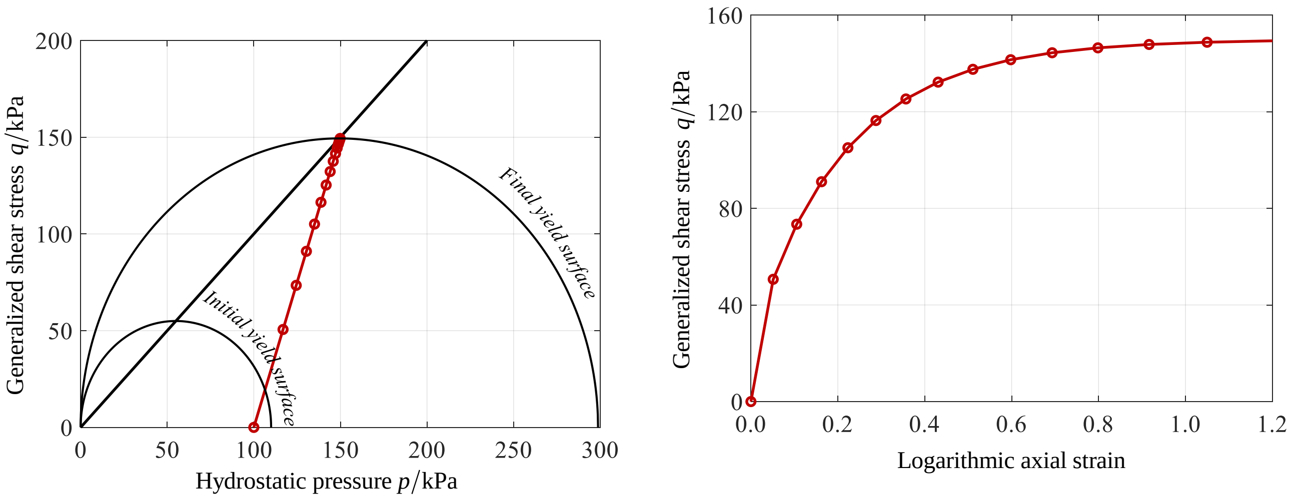
5.2 Conventional triaxial compression test

From the analysis results of Section 3.2, the computational accuracy of numerical differentiation methods depends heavily on the perturbation value. The inappropriate perturbation value may destroy the quadratic convergence when the consistent tangent operator is evaluated by numerical differentiation. In this subsection, the influence of perturbation value on the performance of numerical consistent tangent operator is assessed by an example of conventional triaxial compression demonstrated in Fig. 11. The model parameters of soil are tabulated in Table 5. The initial stress state of $\sigma_1 = \sigma_2 = \sigma_3 = 100$ kPa is employed. The vertical displacement of 0.7m is loaded on the top surface of cylinder with 14 equal incremental steps to ensure that the soil can reach the critical state. The generalized shear stress *vs.* axial strain and the stress path of the examples are depicted in Fig. 12, where $\mu = 1.0$ is considered. .



(a) (b)

Fig. 11 Summary of conventional triaxial compression test: (a) problem schematic; (b) finite element mesh.



(a) (b)

Fig. 12 Simulation result of conventional triaxial compression test: (a) stress path; (b) generalized shear stress *vs.* axial strain.

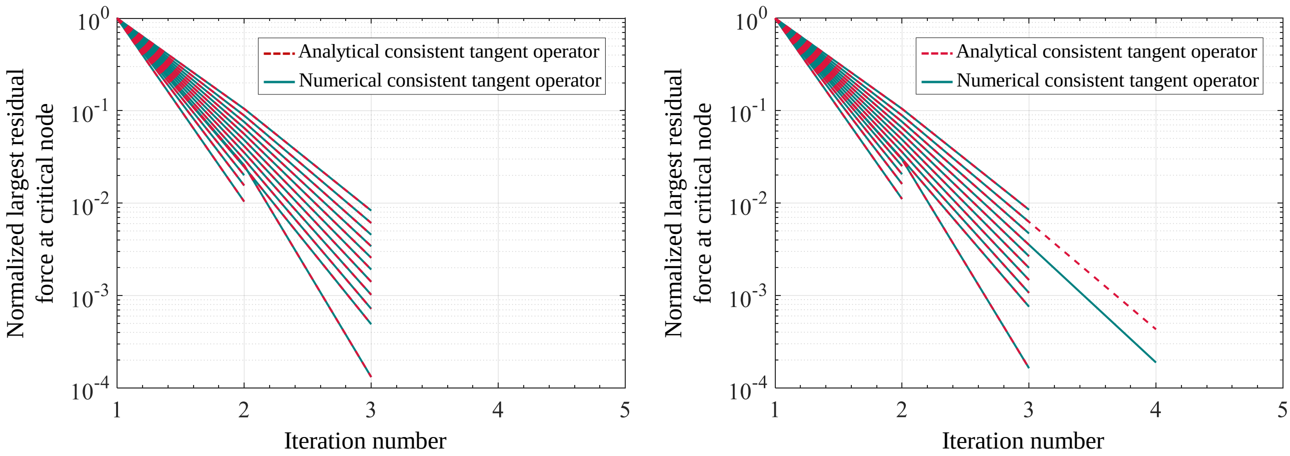
Table 4 reports the total number of global iterations for the 14 steps and CPU time required by the numerical consistent tangent operator with the different perturbations. It is worth emphasizing that the CPU time and global iterations are 38.1s and 39 for the analytical derivation case. In the case of $h \geq 10^{-2}$, the numerical consistent tangent operators obtained by three numerical methods require more global iteration steps than the analytical consistent tangent operator or have encountered failure in the global iteration, which shows that the truncation error caused by too large perturbation value has seriously distorted the numerical solution. In the case of $10^{-10} \leq h \leq 10^{-3}$, the analytical derivation and numerical differentiation schemes both have about the same amount of global iterations, which indicates that the numerical consistent tangent operator obtained by FDM, CDM, and CSDA all achieve quadratic convergence. With the further decrease of *h*, the CSDA still remains convergent, the global iterations of the other two difference methods increase again, and even the global calculation encounter failure when $h \leq 10^{-12}$ for the FDM and $h \leq 10^{-13}$ for CDM. The reason is that the increasing subtractive cancellation error caused by the decrease of *h* has resulted in the distortion of the numerical consistent tangent operator again, which further spoils the convergence of global iteration. From the results presented in Table 4, it is observed that CSDA is superior to other numerical differentiation methods in numerical stability. Finally, an additional case denoted by full CSDA is also presented in Table 4, where both the consistent tangent operator and the Jacobian matrix are evaluated by the CSDA.

Table 4 Computational overhead of different numerical schemes

h	Number of global iterations				CPU time / s			
	FDM	CDM	CSDA	Full CSDA	FDM	CDM	CSDA	Full CSAD
10^{0}	failure	failure	failure	failure	failure	failure	failure	failure
10^{-1}	56	56	failure	failure	98.3	100.1	failure	failure
10^{-2}	55	47	47	failure	95.4	86.4	84.3	failure
10^{-3}	36	38	39	failure	69.4	73.5	72.4	failure
10^{-4}	38	40	39	42	72.1	76.1	72.0	194.7
10^{-5}	39	39	39	40	73.0	74.9	71.9	166.0
10^{-6}	40	39	40	40	74.4	75.6	73.5	162.2
10^{-7}	40	39	42	39	74.4	75.5	76.8	159.9
10^{-8}	40	39	39	42	75.0	74.4	72.1	169.8
10^{-9}	39	40	39	41	73.8	75.8	72.2	165.0
10^{-10}	39	43	42	40	72.8	80.6	76.4	162.1
10^{-11}	47	42	40	39	86.6	79.4	74.4	159.5
10^{-12}	failure	58	39	40	failure	103.0	71.6	161.0
10^{-13}	failure	failure	39	40	failure	failure	71.9	161.9
10^{-14}	failure	failure	41	40	failure	failure	74.6	160.2
10^{-15}	failure	failure	40	39	failure	failure	73.8	156.6
10^{-16}	failure	failure	39	39	failure	failure	71.3	157.5

It can be found that the full CSDA consumes more CPU time than CSDA because the full CSDA involves more numerical evaluation of derivatives. In addition, the convergence of full CSDA is worse than that of CSDA when the perturbation value is large. The reason is that the truncation error of the numerical solution obtained by the full CSDA will not only influence the convergence of the global solution but also influence the convergence of local iteration by the Jacobian matrix. Whereas, the full CSDA will make the model implementation extremely simple because there is no need for any analytical derivative evaluation for both the consistent tangent operator and Jacobian matrix. In the practical application, the simple derivative terms in the two can be analytically derived to reduce the computational overhead.

In what follows, the convergence behaviour of the proposed algorithm on the global level is investigated in the cases of $\mu = 1.0$ and 0.9. Fig. 13 shows the changing law of logarithm normalized largest residual force with the global iteration number, where the numerical consistent tangent operator obtained by the CSDA and the analytical consistent tangent operator are compared. The global iterations number of each load step is almost less than 4 due to the global quadratic convergence of consistent tangent operator. In addition, the convergence behaviours of the numerical consistent tangent operator is almost the same as that of the analytical one, which shows that the proposed algorithm based on the CSDA can avoid tedious derivative operation while ensuring the global quadratic convergence.



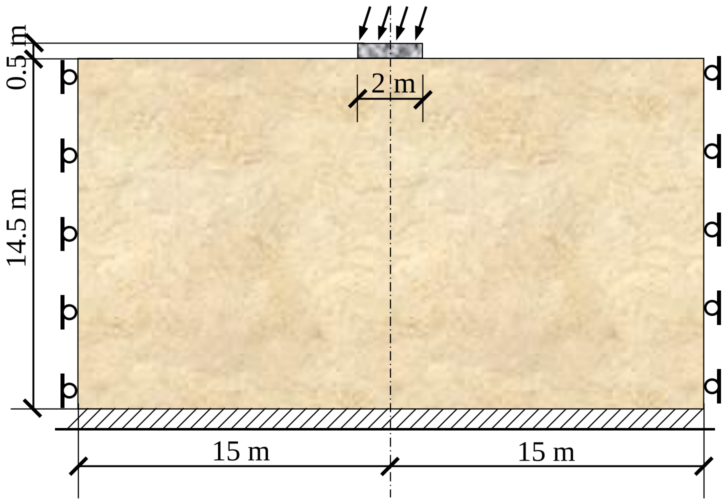
(a) (b)

Fig. 13 Convergence behaviour at the global equilibrium iteration: (a) $\mu = 1.0$; (b) $\mu = 0.9$.

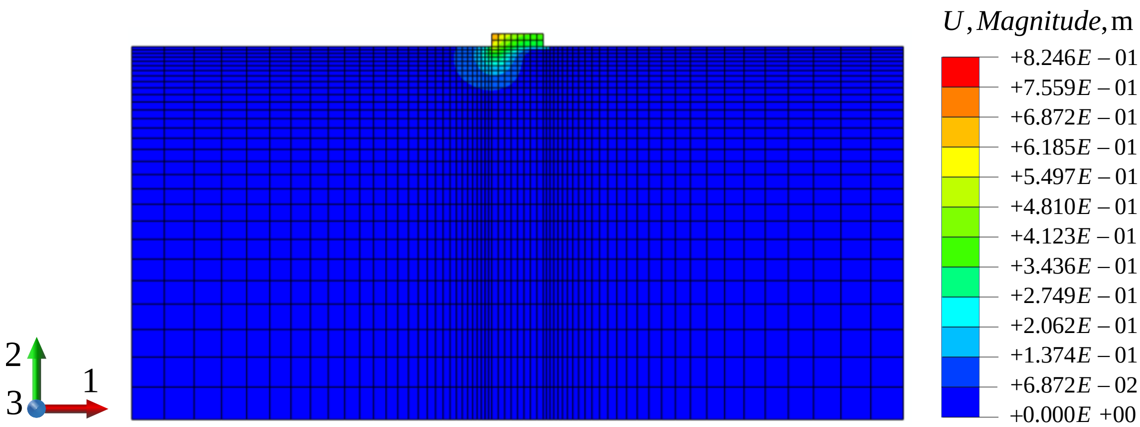
5.3 Strip foundation under inclined load

In what follows, the convergence of the model implementation under large load increment input is investigated by comparing it with the default MCC model of ABAQUS. The analytical Jacobian matrix and consistent tangent operator are used to objectively evaluate the gain from the line search method on the algorithm's convergence. The target example is a strip foundation under an inclined load. Fig. 14 presents boundary conditions and finite element mesh of example. The elastic model and NEP clay model are adopted for the strip foundation and soil, respectively.

$E = 20$ MPa and $\nu = 0.3$ are chosen for the elastic model. The total weight of soil is set to $19\ \mathrm{kN/m^3}$. The material parameters of NEP clay model[51] are presented in Table 5. The preload of top surface is 9kPa at the geostatic step. Then, the top surface of foundation is subjected to an inclined displacement load ($U_1 = $-0.8 m and $U_1 = $-0.2 m). The total analysis time of 1 s and the initial time increment size of 0.01 s are chosen. The size of subsequent time increments is determined by the ABAQUS default step control strategy.



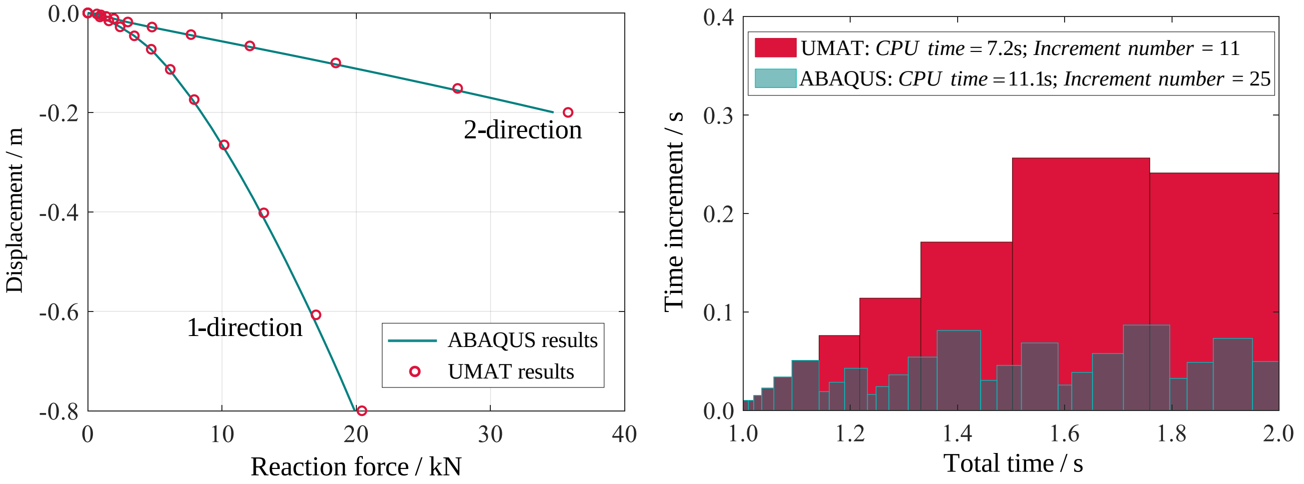
(a)



(b)

Fig. 14 Strip foundation: (a) model geometry; (b) finite element mesh.

The numerical results from the UMAT and the ABAQUS default algorithm are presented in Fig. 15. The UMAT uses less than half as much CPU time as the ABAQUS default algorithm on the premise that the simulation results are almost the same. The reason is that the line search method has a stronger convergence than the Newton method. Therefore, a larger load step input is allowed for the presented model implementation than the ABAQUS default algorithm.



(a) (b)

Fig. 15 Comparison between ABAQUS default algorithm and the presented numerical implementation with $\mu = 1.0$: (a) reaction force *vs.* displacement; (b) change in time increment.

Furthermore, the influence of fractional order μ on the mechanical response of strip foundation example are investigated, where the cases of μ = 0.3, 0.6, 1.0, 1.4, and 1.7 are considered, as shown in Fig. 16. During the initial loading period, the reaction force-displacement curves with the different μ–values almost coincide. With the increase of displacement load, the reaction force of the foundation top surface is smaller with a higher μ–value due to that the stiffness of clay decreases as μ increases. This means that the NEP clay model may provide an effective tool for numerical analysis of geotechnical problems of clay with different stiffness. On the other hand, the calculation results with the different μ–values also demonstrate that the proposed algorithm is not only applicable to MCC model (μ = 1.0), but also NEP clay model ($\mu \neq 1.0$).

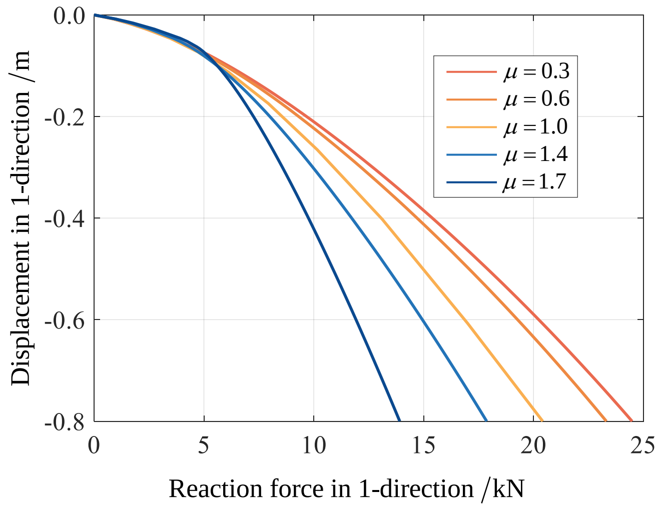


Fig. 16 Influence of parameter μ on simulation results

5.4 Pile foundation bearing capacity test

The last boundary problem is a pile load capacity in the undrained clay subsoil. In view of the symmetry of the problem, a quarter model as shown in Fig. 17 (a) is established to further explore the ability of the presented model implementation to address the 3D coupled problem. The bottom boundary of the analysis area is 1 times the pile diameter from the pile bottom, and the horizontal range is 20 times the pile diameter. The pile-soil interface is modelled by the frictional contact with a frictional coefficient of 0.25. The parameters of NEP clay model are presented in Table 5. The pile employs the linear elastic model with $E = 20$ GPa and $\nu = 0.2$. The 8 nodes brick pore pressure elements (C3D8P) is used to capture the pore pressure response of soil during pile penetration. The effective weight of soil $8\ \mathrm{kN/m^3}$ and permeability $3.6\times10^{-4}\ \mathrm{m/h}$ are used. The numerical simulation contains two analysis steps, i.e., the geostatic equilibrium and loading analysis steps. The vertical displacement of $0.05\ \mathrm{m}$ is loaded on the pile top. The initial time increment, maximum time increment, and total analysis time are set to $500\ \mathrm{s}$, $50\ \mathrm{s}$, and $3600\ \mathrm{s}$, respectively. Fig. 17 (b) depicts the pore pressure distribution result.

Fig. 18 (a) shows the reaction force-displacement curves from ABAQUS and UMAT. It is clear that when the NEP clay model degenerates into the MCC model (μ = 1), the simulation results of UMAT are almost identical to those of ABAQUS, which again verifies the current algorithm's correctness. The results under different μ–values are also presented in Fig. 18 (b). The difference of simulation results with different μ–values is very small. For the pile foundation problem, its bearing capacity mainly depends on the friction effect between the pile and soil and the undrained shear strength of soil. The change of parameter μ will not affect these two.

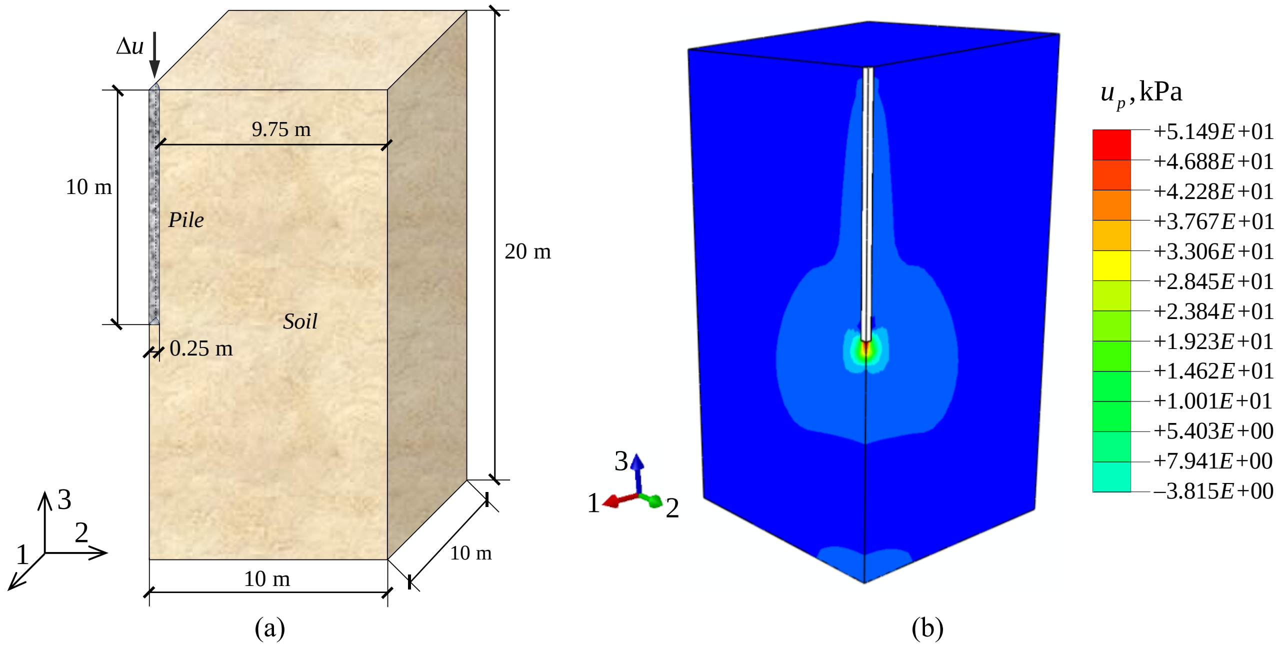


Fig. 17 Pile foundation: (a) model geometry; (b) simulation results.

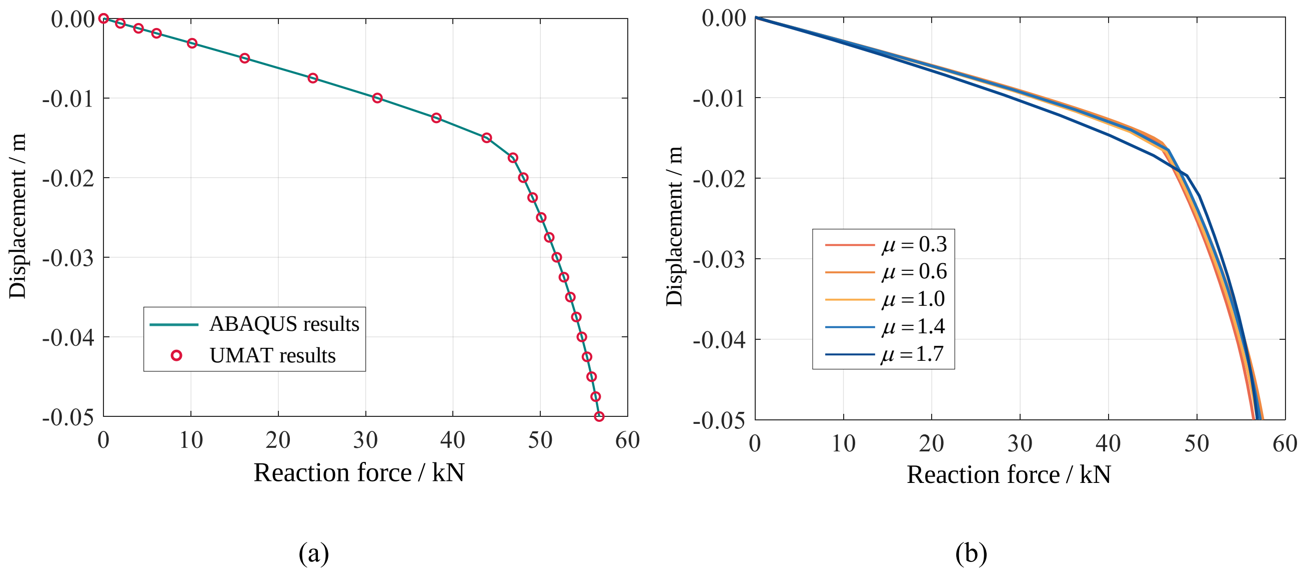


Fig. 18 Reaction force-displacement curves of pile foundation: (a) comparison between ABAQUS and UMAT; (b) influence of parameter μ on simulation results.

Table 5 Material parameters of NEP clay model used for boundary value problems

Boundary value problems	M	λ	κ	ν	e_1	μ
Cylindrical cavity expansion[48]	1.2	0.15	0.03	0.278	1.823	1.0
Conventional triaxial compression test	1.0	0.25	0.05	0.3	1.825	0.9/1.0
Square/Strip foundations[51]	0.898	0.25	0.05	0.3	1.6	0.3~1.7
Pile foundation[20]	1.2	0.2	0.04	0.35	2.0	0.3~1.7

6. CONCLUSION

This paper has proposed a robust and concise implicit stress update algorithm through the combination of smooth function, line search method, and CSDA to implement the NEP clay model. In the model implementation, the smooth function replaces inequality constraints of stress integral equations to eliminate the non-smoothness. The use of the line search method makes the proposed algorithm have a better convergence in large step calculation, even for strong nonlinear cases. The CSDA was used to numerically evaluate the Jacobian matrix used in the local iteration of the model and the consistent tangent operator used in the global iteration to provide quadratic convergence. The NEP clay model has been implemented into the ABAQUS through the new algorithm.

For the validation purpose, the performance of model implementation was assessed by four boundary value problems. In the cylindrical cavity expansion examples, the numerical predictions with the UMAT were in good agreement with the analytical solution, which verified the accuracy of the model implementation. Conventional triaxial compression examples under different perturbation values show that the CSDA has a better numerical robustness than the FDM and the CDM because it has no subtraction cancellation error. The strip foundation example under inclined load also indicated that the proposed algorithm has better convergence than the ABAQUS default algorithm and allows large step load calculation. As a potential application, the model implementation based on the proposed algorithm was used for the analysis of bearing capacity of the pile in the undrained clay subsoil, where the consistency of the results from the UMAT and the ABAQUS further verify the effectiveness of the model implementation in dealing with geotechnical problems.

The proposed algorithm is extremely attractive for the implicit implementation of the complex elastoplastic model since there is no need for cumbersome derivative evaluation and loading/unloading estimation. Users are only required to pay attention to the construction of implicit stress integral equations. Although the numerical differentiation requires more computational overhead than the analytical derivation for the determination of the Jacobian matrix and consistent tangent operator, this additional time consumption can be reduced by using the analytical derivation to obtain simple derivative terms.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Dechun Lu: Conceptualization, Methodology, Supervision, Project administration, Writing - Original draft preparation. **Yaning Zhang**: Data curation, Writing - Original draft preparation, Writing - Review & Editing, Visualization, Software, Formal analysis. **Xin Zhou**: Writing - Original draft preparation, Software, Writing - Review & Editing, Formal analysis, Supervision. **Cancan Su**: Conceptualization, Methodology, Formal analysis. **Zhiwei Gao**: Conceptualization, Methodology. **Xiuli Du**: Conceptualization, Methodology.

ACKNOWLEDGMENTS

Support for this study is provided by the National Natural Science Foundation of China (Grant Nos., 52025084 and 52008231) and China Postdoctoral Science Foundation (Grant Nos., 2022M721884).

DATA AVAILABILITY STATEMENT

The source codes of this study are available from https://github.com/zhouxin615.

APPENDIX A. DETAILS OF JACOBIAN MATRIX:

$$\begin{cases} f_{1,1} = 1+\Delta\phi_{n+1}K^{*}_{n+1}\left(\dfrac{2p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{\mu q_{n+1}^{2}}{N^{2}p_{n+1}^{1+\mu}\Gamma(1-\mu)} - \dfrac{p_{\mathrm{c},\,n+1}}{p_{n+1}^{\mu}\Gamma(1-\mu)}\right) \\ f_{1,2} = \dfrac{2q_{n+1}\Delta\phi_{n+1}K^{*}_{n+1}}{N^{2}p_{n+1}^{\mu}\Gamma(1-\mu)} \\ f_{1,3} = -\dfrac{p_{n+1}^{1-\mu}\Delta\phi_{n+1}K^{*}_{n+1}}{\Gamma(2-\mu)} \\ f_{1,4} = K^{*}_{n+1}\left(\dfrac{q_{\mathrm{n+1}}^{2}}{p_{n+1}^{\mu}N^{2}\Gamma(1-\mu)} + \dfrac{2p_{n+1}^{2-\mu}}{\Gamma(3-\mu)} - \dfrac{p_{\mathrm{c},\,n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)}\right) \end{cases} \tag{A1}$$

$$\begin{cases} f_{2,1} = -\sqrt{\dfrac{3}{2}}a_{8}\left\{2a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},n+1}}{\partial p_{n+1}}\hat{\mathbf{n}}:\Delta\boldsymbol{\gamma}_{n+1} - 6a_{8}\Delta\phi_{n+1}\left\|\mathbf{s}_{n} + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|\left[a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}}{\partial p_{n+1}}\left[\dfrac{q_{n+1}^{1-\mu}}{N^{2}\Gamma(3-\mu)} + \dfrac{p_{n+1}\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2\,q_{n+1}^{1+\mu}\Gamma(1-\mu)}\right] + \bar{G}\dfrac{\left(2p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2\,q_{n+1}^{1+\mu}\Gamma(1-\mu)}\right]\right\} \\ f_{2,2} = 1-\sqrt{\dfrac{3}{2}}a_{8}\left\{2a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v,n+1}}}{\partial q_{n+1}}\hat{\mathbf{n}}:\Delta\boldsymbol{\gamma}_{n+1} - 6a_{8}\Delta\phi_{n+1}\left\|\mathbf{s}_{n} + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|\left[a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}}{\partial q_{n+1}}\left[\dfrac{q_{n+1}^{1-\mu}}{N^{2}\Gamma(3-\mu)} + \dfrac{p_{n+1}\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2q_{n+1}^{1+\mu}\Gamma(1-\mu)}\right] + \bar{G}\left[\dfrac{1-\mu}{N^{2}q_{n+1}^{\mu}\,\Gamma(3-\mu)} + \dfrac{p_{n+1}(-\mu-1)\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2\,q_{n+1}^{2+\mu}\,\Gamma(1-\mu)}\right]\right]\right\} \\ f_{2,3} = -\sqrt{\dfrac{3}{2}}a_{8}\left\{2a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}}{\partial p_{\mathrm{c},\,n+1}}\hat{\mathbf{n}}:\Delta\boldsymbol{\gamma}_{n+1} - 6a_{8}\Delta\phi_{n+1}\left\|\mathbf{s}_{n} + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|\left[a_{9}\left[\dfrac{q_{n+1}^{1-\mu}}{N^{2}\Gamma(3-\mu)} + \dfrac{p_{n+1}\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2\,q_{n+1}^{\mu+1}\Gamma(1-\mu)}\right]\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}}{\partial p_{\mathrm{c},\,n+1}} - \dfrac{\bar{G}p_{n+1}}{2\,q_{n+1}^{\mu+1}\,\Gamma(1-\mu)}\right]\right\} \\ f_{2,4} = -\sqrt{\dfrac{3}{2}}a_{8}\left\{2a_{9}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v,n+1}}}{\partial\Delta\phi_{n+1}}\hat{\mathbf{n}}:\Delta\boldsymbol{\gamma}_{n+1} - 6a_{8}\left\|\mathbf{s}_{n} + 2\bar{G}\Delta\boldsymbol{\gamma}_{n+1}\right\|\left[\dfrac{q_{n+1}^{1-\mu}}{N^{2}\Gamma(3-\mu)} + \dfrac{p_{n+1}\left(p_{n+1}-p_{\mathrm{c},\,n+1}\right)}{2q_{n+1}^{\mu+1}\Gamma(1-\mu)}\right]\left[\bar{G} + a_{9}\Delta\phi_{n+1}\dfrac{\partial\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v,n+1}}}{\partial\phi_{n+1}}\right]\right\} \end{cases} \tag{A2}$$

where $\hat{\mathbf{n}} = \dfrac{\mathbf{s}_{n+1}}{\left\|\mathbf{s}_{n+1}\right\|}$, $\bar{G} = \bar{K}r = r\dfrac{p_{n}}{\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1}}\left[\exp(c_{\kappa}\Delta\varepsilon^{\mathrm{e}}_{\mathrm{v},\,n+1})-1\right]$.

$$\begin{cases} f_{3,1} = -c_{\mathrm{p}}\Delta\phi_{n+1}p_{\mathrm{c}}^{*}\left[\dfrac{2p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{\mu q^{2}}{p_{n+1}^{1+\mu}N^{2}\Gamma(1-\mu)} - \dfrac{p_{\mathrm{c},\,n+1}}{p_{n+1}^{\mu}\Gamma(1-\mu)}\right] \\ f_{3,2} = -\dfrac{2p_{\mathrm{c}}^{*}c_{\mathrm{p}}\Delta\phi_{n+1}q_{n+1}}{p_{n+1}^{\mu}N^{2}\Gamma(1-\mu)} \\ f_{3,3} = 1+\dfrac{p_{\mathrm{c}}^{*}c_{\mathrm{p}}\Delta\phi_{n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} \\ f_{3,4} = p_{\mathrm{c}}^{*}c_{\mathrm{p}}\left[\dfrac{p_{\mathrm{c},\,n+1}p_{n+1}^{1-\mu}}{\Gamma(2-\mu)} - \dfrac{q_{n+1}^{2}}{N^{2}p_{n+1}^{\mu}\,\Gamma(1-\mu)} - \dfrac{2p_{n+1}^{2-\mu}}{\Gamma(3-\mu)}\right] \end{cases} \tag{A3}$$

where $p_{\mathrm{c}}^{*} = \left(p_{\mathrm{c}}\right)_{n}\exp\left(c_{\mathrm{p}}\Delta\phi_{n+1}\dfrac{\partial^{\mu}f_{n+1}}{\partial p_{n+1}^{\mu}}\right)$

$$\begin{cases} f_{4,1} = \dfrac{\partial f_{4}}{\partial p_{n+1}} = \chi_{0}\left(2p_{n+1}-p_{c,\,n+1}\right) \\ f_{4,2} = \dfrac{\partial f_{4}}{\partial q_{n+1}} = \chi_{0}\dfrac{2q_{n+1}}{N^{2}} \\ f_{4,3} = \dfrac{\partial f_{4}}{\partial p_{\mathrm{c},\,n+1}} = -\chi_{0}p_{n+1} \\ f_{4,4} = \dfrac{\partial f_{4}}{\partial\Delta\phi_{n+1}} = \chi_{1} \end{cases} \tag{A4}$$

where $\chi_{0} = \dfrac{\partial f_{4}}{\partial f_{n+1}} = \dfrac{f_{n+1}}{\sqrt{\left(c_{\mathrm{d}}\Delta\phi_{n+1}\right)^{2} + f_{n+1}^{2} + 2\beta}} + 1$ and $\chi_{1} = \dfrac{\partial f_{4}}{\partial\Delta\phi_{n+1}} = \dfrac{c_{\mathrm{d}}^{2}\Delta\phi_{n+1}}{\sqrt{\left(c_{\mathrm{d}}\Delta\phi_{n+1}\right)^{2} + f_{n+1}^{2} + 2\beta}} - c_{\mathrm{d}}$.

APPENDIX B. MATRIX REPRESENTATION OF TENSORS:

The matrix representation of second order includes:

$\mathbf{1} = \delta_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the orthonormal bases of second-order tensor. The Kronecker delta δ_{ij} can be expressed by:

$$\delta_{ij} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{B1}$$

$\partial f / \partial \boldsymbol{\sigma} = \partial f / \partial \sigma_{ij} \, \mathbf{e}_i \otimes \mathbf{e}_j$, where $\partial f / \partial \sigma_{ij}$ is expressed by:

$$\frac{\partial f}{\partial \sigma_{ij}} = \begin{bmatrix} \partial f / \partial \sigma_{11} & \partial f / \partial \sigma_{22} & \partial f / \partial \sigma_{33} & 2\partial f / \partial \sigma_{12} & 2\partial f / \partial \sigma_{23} & 2\partial f / \partial \sigma_{13} \end{bmatrix}^T \tag{B2}$$

$\Delta \boldsymbol{\sigma} = \Delta \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\Delta \sigma_{ij}$ can be expressed by:

$$\Delta \sigma_{ij} = \begin{bmatrix} \Delta \sigma_{11} & \Delta \sigma_{22} & \Delta \sigma_{33} & \Delta \sigma_{12} & \Delta \sigma_{23} & \Delta \sigma_{13} \end{bmatrix}^T \tag{B3}$$

$\Delta \boldsymbol{\varepsilon}_{n+1} = \Delta \varepsilon_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\Delta \varepsilon_{ij}$ can be expressed by:

$$\Delta \varepsilon_{ij} = \begin{bmatrix} \Delta \varepsilon_{11} & \Delta \varepsilon_{22} & \Delta \varepsilon_{33} & \Delta \varepsilon_{12} & \Delta \varepsilon_{23} & \Delta \varepsilon_{13} \end{bmatrix}^T \tag{B4}$$

$\Delta \boldsymbol{\gamma}_{n+1} = \Delta \gamma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\Delta \gamma_{ij} = \Delta \varepsilon_{ij} - \delta_{ij} \Delta \varepsilon_{\mathrm{v}} / 3$ can be expressed by:

$$\Delta \gamma_{ij} = \begin{bmatrix} \Delta \gamma_{11} & \Delta \gamma_{22} & \Delta \gamma_{33} & \Delta \gamma_{12} & \Delta \gamma_{23} & \Delta \gamma_{13} \end{bmatrix}^T \tag{B5}$$

$\hat{\mathbf{n}} = \hat{n}_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\hat{n}_{ij}$ can be expressed by:

$$\hat{n}_{ij} = \begin{bmatrix} \hat{n}_{11} & \hat{n}_{22} & \hat{n}_{33} & \hat{n}_{12} & \hat{n}_{23} & \hat{n}_{13} \end{bmatrix}^T = \frac{1}{\|\mathbf{s}\|} \begin{bmatrix} s_{11} & s_{22} & s_{33} & s_{12} & s_{23} & s_{13} \end{bmatrix}^T \tag{B6}$$

$\tilde{\mathbf{n}} = \tilde{n}_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, where $\tilde{n}_{ij}$ can be expressed by:

$$\tilde{n}_{ij} = \begin{bmatrix} \hat{n}_{11} & \hat{n}_{22} & \hat{n}_{33} & 2\hat{n}_{12} & 2\hat{n}_{23} & 2\hat{n}_{13} \end{bmatrix}^T \tag{B7}$$

The matrix representation of fourth order includes:

$\mathbf{I} = \mathbf{1} \otimes \mathbf{1} = \delta_{ij} \delta_{kl} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$, where $\mathbf{e}_i$, $\mathbf{e}_j$, $\mathbf{e}_k$, and $\mathbf{e}_l$ are the orthonormal bases of fourth-order tensor and $\delta_{ij} \delta_{kl}$ can be expressed by:

$$\delta_{ij}\delta_{kl} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B8}$$

$\mathbf{I}^{\mathrm{vol}} = \frac{1}{3}\mathbf{1}\otimes\mathbf{1} = \frac{1}{3}\delta_{ij}\delta_{kl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$, where $\frac{1}{3}\delta_{ij}\delta_{kl}$ can be expressed by:

$$\frac{1}{3}\delta_{ij}\delta_{kl} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B9}$$

$\mathbf{I}^{\mathrm{sym}} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$, where $\frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)$ can be expressed by:

$$\frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \end{bmatrix} \tag{B10}$$

$\bar{\mathbf{D}} = 3\mathbf{I}^{\mathrm{vol}}\left(\bar{K} - \frac{2}{3}\bar{G}\right) + 2\mathbf{I}^{\mathrm{sym}}\bar{G} = \bar{D}_{ijkl}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$, where $\bar{D}_{ijkl}$ can be expressed by:

$$\bar{D}_{ijkl} = \delta_{ij}\delta_{kl}\bar{K}\left(1 - \frac{2}{3}r\right) + \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)\bar{K}r = \bar{K}\begin{bmatrix} 1+\frac{1}{3}r & 1-\frac{2}{3}r & 1-\frac{2}{3}r & 0 & 0 & 0 \\ 1-\frac{2}{3}r & 1+\frac{1}{3}r & 1-\frac{2}{3}r & 0 & 0 & 0 \\ 1-\frac{2}{3}r & 1-\frac{2}{3}r & 1+\frac{1}{3}r & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 0 & 0 \\ 0 & 0 & 0 & 0 & r & 0 \\ 0 & 0 & 0 & 0 & 0 & r \end{bmatrix} \tag{B11}$$

$\mathbf{P} = \left(\delta_{ik}\delta_{jl} - \frac{1}{3}\delta_{ij}\delta_{kl}\right)\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$ where $\delta_{ik}\delta_{jl} - \frac{1}{3}\delta_{ij}\delta_{kl}$ can be expressed by:

$$\bar{D}_{ijkl} = \delta_{ij}\delta_{kl}\bar{K}\left(1-\frac{2}{3}r\right) + \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)\bar{K}r = \bar{K}\begin{bmatrix} 1+\frac{1}{3}r & 1-\frac{2}{3}r & 1-\frac{2}{3}r & 0 & 0 & 0 \\ 1-\frac{2}{3}r & 1+\frac{1}{3}r & 1-\frac{2}{3}r & 0 & 0 & 0 \\ 1-\frac{2}{3}r & 1-\frac{2}{3}r & 1+\frac{1}{3}r & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 0 & 0 \\ 0 & 0 & 0 & 0 & r & 0 \\ 0 & 0 & 0 & 0 & 0 & r \end{bmatrix} \tag{B12}$$

$$\delta_{ik}\delta_{jl} - \frac{1}{3}\delta_{ij}\delta_{kl} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} \tag{B13}$$

References:

1. Zhao JD, Sheng DC, Rouainia M, Sloan SW. Explicit stress integration of complex soil models. *Int J Numer Anal Met*. 2005;29(12):1209-1229.

2. Gao ZW, Li X, Lu DC. Nonlocal regularization of an anisotropic critical state model for sand. *Acta Geotech*. 2021;17(2):427-439.

3. Borja RI. Cam-Clay plasticity, Part II: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor. *Comput Method Appl M*. 1991;88:225-240.

4. Borja RI, Lee SR. Cam-clay plasticity, part 1: implicit integration of elasto-plastic constitutive relations. *Comput Method Appl M*. 1990;78:49-72.

5. De Borst R, Heeres OM. A unified approach to the implicit integration of standard, non-standard and viscous plasticity

models. *Int J Numer Anal Met*. 2002;26(11):1059-1070.

6. Heeres OM, De Borst R. Implicit Integration of Non-Standard Plasticity Models. Application of Numerical Methods to Geotechnical Problems: *Springer*; 1998. p. 483-492.
7. Huang JS, Griffiths DV. Return mapping algorithms and stress predictors for failure analysis in geomechanics. *J Eng Mech*. 2009;135(4):276-284.
8. Huang JS, Griffiths DV. Observations on return mapping algorithms for piecewise linear yield criteria. *Int J Geomech*. 2008;8(4):253-265.
9. Duretz T, Souche A, Borst R, Le Pourhiet L. The Benefits of Using a Consistent Tangent Operator for Viscoelastoplastic Computations in Geodynamics. *Geochemistry, Geophysics, Geosystems*. 2018;19(12):4904-4924.
10. Panteghini A, Lagioia R. A fully convex reformulation of the original Matsuoka－Nakai failure criterion and its implicit numerically efficient integration algorithm. *Int J Numer Anal Met*. 2014;38(6):593-614.
11. Potts DM, Cui WJ, Zdravković L. A coupled THM finite element formulation for unsaturated soils and a strategy for its nonlinear solution. *Comput Geotech*. 2021;136:104221.
12. Ortiz M, Simo JC. An analysis of a new class of integration algorithms for elastoplastic constitutive relations. *Int J Numer Meth Eng*. 1986;23(3):353-366.
13. Simo JC, Ortiz M. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. *Comput Method Appl M*. 1985;49(2):221-245.
14. Moran B, Ortiz M, Shih CF. Formulation of implicit finite element methods for multiplicative finite deformation plasticity. *Int J Numer Meth Eng*. 1990;29(3):483-514.
15. Petalas AL, Dafalias YF. Implicit integration of incrementally non-linear, zero- elastic range, bounding surface plasticity. *Comput Geotech*. 2019;112:386-402.

16. Lloret-Cabot M, Sheng DC. Assessing the accuracy and efficiency of different order implicit and explicit integration schemes. *Comput Geotech*. 2022;141:104531.

17. Brannon RM, Leelavanichkul S. A multi-stage return algorithm for solving the classical damage component of constitutive models for rocks, ceramics, and other rock-like media. *Int J Fracture*. 2010;163:133-149.

18. Scherzinger WM. A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method. *Comput Method Appl M*. 2017;317:526-553.

19. Zhou X, Lu DC, Su CC, Gao ZW, Du XL. An unconstrained stress updating algorithm with the line search method for elastoplastic soil models. *Comput Geotech*. 2022;143:104592.

20. Zhou X, Lu DC, Zhang YN, Du XL, Rabczuk T. An open-source unconstrained stress updating algorithm for the modified Cam-clay model. *Comput Method Appl M*. 2022;390:114356.

21. Areias P, Rabczuk T. Smooth finite strain plasticity with non-local pressure support. *Int J Numer Meth Eng*. 2010;81(1):106-134.

22. Areias P, Rabczuk T, César De Sá J. Semi-implicit finite strain constitutive integration of porous plasticity models. *Finite Elem Anal Des*. 2015;104:41-55.

23. Areias P, Rabczuk T, César De Sá J, Alves JL. Semi-implicit finite strain constitutive integration and mixed strain/stress control based on intermediate configurations. *Eng Struct*. 2016;124:344-360.

24. Krabbenhoft K, Lyamin AV. Computational Cam clay plasticity using second-order cone programming. *Comput Method Appl M*. 2012;209-212:239-249.

25. Geng DJ, Dai N, Guo PJ, Zhou SH, Di HG. Implicit numerical integration of highly nonlinear plasticity models. *Comput Geotech*. 2021;132:103961.

26. Shterenlikht A, Alexander NA. Levenberg-Marquardt vs Powell's dogleg method for Gurson-Tvergaard-Needleman

plasticity model. *Comput Method Appl M*. 2012;237-240:1-9.

27. Yoon SY, Lee SY, Barlat F. Numerical integration algorithm of updated homogeneous anisotropic hardening model through finite element framework. *Comput Method Appl M*. 2020;372:113449.

28. Choi H, Yoon JW. Stress integration-based on finite difference method and its application for anisotropic plasticity and distortional hardening under associated and non-associated flow rules. *Comput Method Appl M*. 2019;345:123-160.

29. Carow C, Rackwitz F. Comparison of implicit and explicit numerical integration schemes for a bounding surface soil model without elastic range. *Comput Geotech*. 2021;140:104206.

30. Pérez-Foguet A, Rodríguez-Ferran A, Huerta A. Numerical differentiation for local and global tangent operators in computational plasticity. *Comput Method Appl M*. 2000;189(1):277-296.

31. Kiran R, Khandelwal K. Complex step derivative approximation for numerical evaluation of tangent moduli. *Comput Struct*. 2014;140:1-13.

32. Tanaka M, Sasagawa T, Omote R, Fujikawa M, Balzani D, Schröder J. A highly accurate 1st-and 2nd-order differentiation scheme for hyperelastic material models based on hyper-dual numbers. *Comput Method Appl M*. 2015;283:22-45.

33. Fike JA. Multi-objective optimization using hyper-dual numbers. Palo Alto: *Stanford university*; 2013.

34. Balzani D, Gandhi A, Tanaka M, Schröder J. Numerical calculation of thermo-mechanical problems at large strains based on complex step derivative approximation of tangent stiffness matrices. *Comput Mech*. 2015;55(5):861-871.

35. Liang JY, Lu DC, Du XL, Wu W, Ma C. Non-orthogonal elastoplastic constitutive model for sand with dilatancy. *Comput Geotech*. 2020;118:103329.

36. Zhou X, Lu DC, Du XL, Wang GS, Meng FP. A 3D non-orthogonal plastic damage model for concrete. *Comput Method Appl M*. 2019;360:112716.

37. Liang JY, Lu DC, Zhou X, Du XL, Wu W. Non-orthogonal elastoplastic constitutive model with the critical state for clay. *Comput Geotech*. 2019;116:103200.

38. Sun YF, Gao YF, Zhu QZ. Fractional order plasticity modelling of state-dependent behaviour of granular soils without using plastic potential. *Int J Plasticity*. 2018;102:53-69.

39. Sun YF, Sumelka W, Gao YF, Nimbalkar S. Phenomenological fractional stress-dilatancy model for granular soil and soil-structure interface under monotonic and cyclic loads. *Acta Geotech*. 2021;16(10):3115-3132.

40. Qu PF, Zhu QZ, Zhao LY, Cao YJ. A micromechanics-based fractional frictional damage model for quasi-brittle rocks. *Comput Geotech*. 2021;139:104391.

41. Lu DC, Zhou X, Du XL, Wang GS. A 3D fractional elastoplastic constitutive model for concrete material. *Int J Solids Struct*. 2019;165:160-175.

42. Roscoe KH, Burland JB. On the generalized stress-strain behaviour of wet clay: Engineering Plasticity. Cambridge: *Cambridge University Press*; 1968.

43. Lu DC, Liang JY, Du XL, Ma C, Gao ZW. Fractional elastoplastic constitutive model for soils based on a novel 3D fractional plastic flow rule. *Comput Geotech*. 2019;105:277-290.

44. Nakai T, Matsuoka H. A generalized elastoplastic constitutive model for clay in three-dimensional stresses. *Soils Found*. 1986;26(3):81-98.

45. Sheahan T. An experimental study of the time-dependent undrained shear behavior of resedimented clay using automated stress path triaxial equipment. Cambridge: *Massachusetts Institute of Technology*; 1991.

46. Pérez-Foguet A, Armero F. On the formulation of closest‐point projection algorithms in elastoplasticity—part II: Globally convergent schemes. *Int J Numer Meth Eng*. 2002;53(2):331-374.

47. Lyness JN, Moler CB. Numerical differentiation of analytic functions. *Siam J Numer Anal*. 1967;4(2):202-210.

48. Chen SL, Abousleiman YN. Exact undrained elasto-plastic solution for cylindrical cavity expansion in modified Cam Clay soil. *Géotechnique*. 2012;62(5):447-456.

49. Chen SL, Abousleiman YN. Exact drained solution for cylindrical cavity expansion in modified Cam Clay soil. *Géotechnique*. 2013;63(6):510-517.

50. Liu K, Chen SL, Voyiadjis GZ. Integration of anisotropic modified Cam Clay model in finite element analysis: Formulation, validation, and application. *Comput Geotech*. 2019;116:103198.

51. Sheng DC, Sloan SW, Yu HS. Aspects of finite element implementation of critical state models. *Comput Mech*. 2000;26(2):185-196.