Comput Mech (2008) 42:685–694
DOI 10.1007/s00466-008-0268-0

ORIGINAL PAPER

Non-convexity and stress-path dependency of unsaturated soil models

Daichao Sheng · Dorival M. Pedroso · Andrew J. Abbo

Received: 8 October 2007 / Accepted: 2 March 2008 / Published online: 21 March 2008

Abstract Yield surfaces for unsaturated soils are usually non-convex if the size of the yield surface has to increase with increasing suction. An expanding yield surface with increasing suction is crucial for modelling the volume collapse due to wetting. The non-convexity always exists at the transition between saturated and unsaturated states, irrespective of the stress variables used in the model. Some recent models for unsaturated soils also possess a stress path dependent hardening law. The non-convexity and stress-path dependency of the constitutive model make the implementation into finite element codes very challenging. This paper discusses aspects of stress integration schemes for non-convex and stress-path dependent models for unsaturated soils.

Keywords Unsaturated soils · Path dependent hardening · Non-convex yield surface · Differential algebraic system · Runge–Kutta method · Numerical integration · Stress update

1 Introduction

Constitutive models for unsaturated soils usually adopt a yield surface that expands with increasing suction in order to model the volume collapse during wetting [1]. The non-convexity of the yield surface thus exists at the transition between saturated and unsaturated state [18], irrespective of the stress variables used in the model (see Fig. 1 below). The only exception is the model by Wheeler et al. [19], where the size of the yield surface does not change with the modified suction.

D. Sheng (✉) · D. M. Pedroso · A. J. Abbo
Centre for Geotechnical and Material Modelling,
University of Newcastle, New Castle, NSW 2308, Australia
e-mail: daichao.sheng@newcastle.edu.au

Another feature of unsaturated soil models is the stress-path dependent hardening law. Such a feature is present in a recent model by Sheng et al. [11] and is illustrated in Fig. 2. If a slurry soil is dried from point A to B, the yield surface expands from $\bar{p}_0$ to $\bar{p}_{yB}$, with its shape remaining unchanged. If the unsaturated soil at point B is then compressed to point C, the yield surface expands and its shape changes as well (denoted by the solid curve $\bar{p}_{yC}$). However, if the slurry soil at point A is first compressed to point D and then dried to C, the yield surface would take a different shape (denoted by the dashed curve $\bar{p}_{yC}$). The essential reason for this stress path dependency is that a change in suction has a different effect on the plastic volumetric strain than a change in mean stress when the soil becomes unsaturated.

The non-convexity and stress-path dependent hardening laws of unsaturated soil models present difficulties in the implementation of these soil models into finite element codes. These codes are usually based on global equations of the equilibrium of momentum and the continuity of pore fluids (water and air). The degrees of freedom are displacements, pore water pressure and pore air pressure. Under natural ground conditions, the pore air pressure is often under static atmospheric pressure, which further reduces the global equations to two [12]. The suction is the difference between the pore water pressure and pore air pressure, and is simply the negative pore water pressure if the air pressure is assumed to be atmospheric.

Rate independent elastoplastic models can be represented by a set of differential and algebraic equations (DAE) [3,6,14]. These equations must be solved (integrated) for applied increments of stress, suction or strains. As discussed in Sheng et al. [12], the pair of strain and suction increments can be conveniently used for the stress update step in finite element codes. The pair of stress and suction increments is generally considered in smaller programs, known as

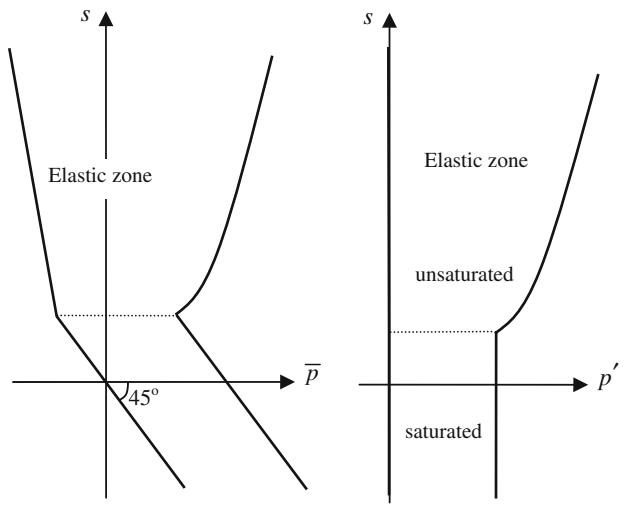

Fig. 1 Non-convexity of unsaturated soil models ($\bar{p}$: net mean stress; p': effective mean stress; s: suction)

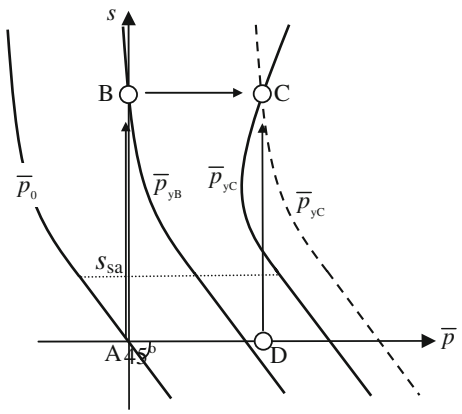

Fig. 2 Stress path dependency of the model by Sheng et al. (2006). s_{sa}: saturation suction

"constitutive drivers", that are useful for the verification of constitutive equations by means of the local integration of the DAE. These programs generally consider a pre-defined stress path and can also take advantage of Runge–Kutta methods.

In this paper, the accuracy of the proposed algorithm is verified in both ways: for constitutive equations driven by known strain and suction increments and for those driven by known stress and suction increments. The first one is based on a finite element analysis of a unit cube and the second one is carried out using a "constitutive driver" program. This methodology is very important in order to verify whether the trial stress/suction strategy adopted for the stress update algorithm in FEM leads to the same results obtained with the driver program. The latter results are considered the correct ones because, in the driver program, the stress/suction path is always known a priori. The comparison between FEM and local (driver) simulations allow us

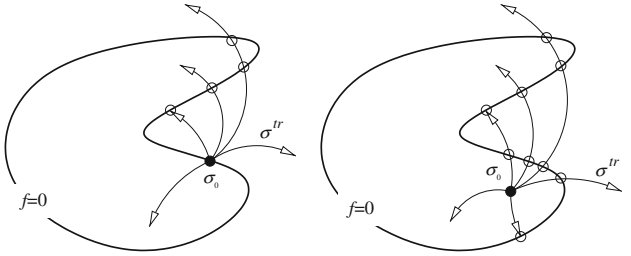

Fig. 3 Intersections between non-convex yield surface and elastic trial stress path

conclude that the algorithm proposed is accurate. In addition, due to the automatic substepping scheme, the solution is achieved efficiently. Overall, the algorithm is able to solve the relatively complicate system of equations of unsaturated soil models with stress path dependence and non-convexity with robustness and accuracy.

2 Non-convex yield surface

For given strain and suction increments, the current stress state and internal variables must be updated in accordance with the constitutive law. This update is generally carried out using numerical stress integration schemes. Both implicit and explicit schemes have been used to integrate elastoplastic models. Implicit schemes, where all gradients are estimated at an advanced stress state, cannot be used for elastoplastic models with non-convex yield surfaces, because the extrapolated gradients cannot be determined due to the uncertainty of whether an advanced position is inside or outside the yield surface. On the other hand, explicit schemes can proceed in an incremental fashion, but require the intersection between the current yield surface and an elastic trial stress path to be determined.

A key issue in integrating the incremental stress–strain relationships using an explicit method is to find the intersection between the elastic trial stress and the current yield surface. Fig. 3 illustrates some possible situations. The most complicated situation occurs when the yield surface is crossed three times. However, it is not possible to know a priori how many times the yield surface is crossed, because the size of the yield surface will change after the first intersection due to hardening. Therefore, for non-convex yield surfaces, the key task is to find the very first intersection for any possible path.

In order to determine whether the yield surface is crossed, a secant trial stress increment is computed, based on an elastic stress–suction–strain relationship. This elastic trial stress is given as follows:

$$\Delta\boldsymbol{\sigma}^{tr} = \boldsymbol{D}^e : \Delta\boldsymbol{\varepsilon} + \boldsymbol{W}^e \Delta s \tag{1}$$

where the stress is either the net stress or effective stress (depending on the model), $\boldsymbol{D}^e$ is the fourth order elastic stiffness tensor and $\boldsymbol{W}^e$ is a second order tensor defined according to a specific law for unsaturated soils; for example, the equations presented in Sheng et al. [11–13] may be adopted. For models of saturated soils, the term $\boldsymbol{W}^e \Delta s$ depends on the stress variables used. If the effective stress is used, the term $\boldsymbol{W}^e \Delta s$ becomes zero and can be disregarded. On the other hand, if the net stress is used, the term $\boldsymbol{W}^e \Delta s$ becomes $-\boldsymbol{I}\Delta u_w$, where $\boldsymbol{I}$ is the second order identity tensor and u_w the pore water pressure.

In (1), $\Delta\boldsymbol{\varepsilon}$ is the strain increment provided from the finite element routines prior to the computation of the residuals between internal and external forces. For unsaturated soils, the increment of suction Δs is also input for the stress-update algorithm. If the elastic modulus is linear, i.e. it is independent of the stresses, suction and internal variables, it is trivial to compute the elastic trial increment. Otherwise, for some non-linear relations, a secant analytical modulus may be considered.

Finding the intersection between the elastic trial stress increment and the current yield surface can be cast into the problem of finding multiple roots of a nonlinear equation. $f_\alpha(\alpha) = 0$. The roots (α) must be computed inside the interval [0, 1]. As this function involves the evaluation of the yield function along the strain and suction paths it is given as

$$f_\alpha(\alpha) = f(\boldsymbol{\sigma}_\alpha, s_\alpha, z_k) \tag{2}$$

where $f(\boldsymbol{\sigma}, s, z_k)$ is the yield function, z_k indicates a set of internal variables and the intermediate stress–suction states $\boldsymbol{\sigma}^\alpha$ and s^α are calculated according to

$$\boldsymbol{\sigma}^\alpha = \boldsymbol{\sigma}^\chi + \alpha\Delta\boldsymbol{\sigma}^{tr} \quad \text{and} \quad s^\alpha = s^c + \alpha\Delta s \tag{3}$$

in which $\boldsymbol{\sigma}^c$ and s^c are the current stress and suction states. Note that in (2) the internal variables z_k are kept constant during the solution for the intersection. These variables only change during hardening/softening when a portion of the trial stress–suction path is located outside the yield surface.

The technique proposed here follows the Kronecker–Picard (KP) formula [5] for the determination of the number of roots of a nonlinear equation. This number of roots is set to the near integral of the following real number:

$$N = \frac{-\gamma}{\pi}\int_a^b \frac{f_\alpha(x)h_\alpha(x) - g(x)^2}{f_\alpha(x)^2 + \gamma^2 g_\alpha(x)^2}dx + \frac{1}{\pi}\arctan\left\{\frac{\gamma\left([f_\alpha(a)g_\alpha(b) - f_\alpha(b)g_\alpha(a)]\right)}{f_\alpha(a)f_\alpha(b) + \gamma^2 g_\alpha(a)g_\alpha(b)}\right\} \tag{4}$$

The above formula requires that $f_\alpha(\alpha)$ must be continuously differentiable to the second order for values of α from a to b. In (4), g_α and h_α represent the first and second derivatives of the function f_α with respect to α, respectively, and γ is an arbitrary small positive constant. The parameter γ does not affect the number of roots computed with the KP formula [5], but a value of γ close to zero may increase the computation time [17]. For the unsaturated model under study, a value of $\gamma = 0.01$ lead to the correct number of roots for all paths, i.e. 100% accurate.

The first and second derivative of f_α with respect to α can be directly determined as follows:

$$g_\alpha(\alpha) = \frac{\partial f_\alpha}{\partial\alpha} = \frac{\partial f_\alpha}{\partial\boldsymbol{\sigma}_\alpha} : \frac{d\boldsymbol{\sigma}_\alpha}{d\alpha} + \frac{\partial f_\alpha}{\partial s_\alpha}\frac{ds_\alpha}{d\alpha} = \left.\frac{\partial f}{\partial\boldsymbol{\sigma}}\right|_\alpha : \Delta\boldsymbol{\sigma}^{tr} + \left.\frac{\partial f}{\partial s}\right|_\alpha \Delta s \tag{5}$$

$$h_\alpha(\alpha) = \frac{\partial^2 f_\alpha}{\partial\alpha^2} = \Delta\boldsymbol{\sigma}^{tr} : \left.\frac{\partial^2 f}{\partial\boldsymbol{\sigma}\,\partial\boldsymbol{\sigma}}\right|_\alpha : \Delta\boldsymbol{\sigma}^{tr} + 2\Delta\boldsymbol{\sigma}^{tr} : \left.\frac{\partial^2 f}{\partial\boldsymbol{\sigma}\,\partial s}\right|_\alpha \Delta s + \left.\frac{\partial^2 f}{\partial s^2}\right|_\alpha \Delta s^2 \tag{6}$$

The number of roots estimated according to (4) is used to divide the interval of α into subintervals until each subinterval contains at most one root. First, N is computed for the interval $[a, b]$. If N is larger than one, the interval $[a, b]$ is divided into two equal subintervals, $[a,\ (a+b)/2]$ and $[(a+b)/2,\ b]$. The number of roots for each subinterval is then computed and any subinterval that contains more than one root is further divided into two equal sub-subintervals. This process continues until each subinterval contains at most one root. As shown by Kavvadias et al. [5], the usage of equal-size intervals (equiprobable parts) is not much worse than an algorithm which would consider the statistical distribution of the roots inside $[a, b]$, such as the algorithm presented in Kavvadias et al. [5].

Once the roots are bracketed, the solution of each root can be found by using existing numerical methods such as the Newton–Raphson method. It should be noted that the Newton–Raphson method, although fast, may not converge in some circumstances because it does not constrain the solution to lie within specified bounds. Therefore, more advanced methods can be used here. For example, the Pegasus method used in Sloan et al. [16] is very robust and competitively fast. The method by Brent [2] provides another attractive alternative here. The Brent method does not use any derivative, does not require initial guesses and guarantees the convergence as long as the values of the function are computable within a given region containing a root. This characteristic of the Brent method is due to the combination of the bisection method, the secant method and inverse quadratic interpolation. Therefore, it has the reliability of the bisection method and the efficiency of the less reliable secant method or inverse quadratic interpolation.

The evaluation of the integral in (4) with the KP formula is generally not trivial and hence a numerical integration or

quadrature technique has to be utilised. For example, the Gauss–Legendre method [4] can be used here. In addition, for highly non-linear yield functions, an adaptive integration scheme may also have to be used. In the numerical examples presented in this paper, the adaptive integration scheme explained in Piessens et al. [9], implemented in the QAGS routines, is used. These routines which are based on the QUADPACK library, available at http://www.netlib.org, can efficiently perform the numerical integration even for functions with singularities.

3 Stress path dependency

The discussion in this section is limited to the SFG model by Sheng et al. [11]. In this model, the yield function is written as

$$f = q^2 - M^2 \cdot (\bar{p} - p_0(s)) \cdot \left(p_y(s, z_0, z_1) - \bar{p}\right) = 0 \quad (7)$$

where q is the deviatoric stress, M the slope of the critical state line, z_0 and z_1 are internal variables and p_0 and p_y are yield stresses given as follows:

$$p_0(s) = \begin{cases} k(s) & \text{if} s > s_{\text{sa}} \\ -s & \text{otherwise} \end{cases} \quad (8)$$

$$p_y(s, z_0, z_1) = \begin{cases} z_0 - s + \frac{z_0}{z_1}[s + k(s)] & \text{if } s > s_{\text{sa}} \\ z_0 - s & \text{otherwise} \end{cases} \quad (9)$$

where

$$k(s) = -s_{\text{sa}} - (1 + s_{\text{sa}}) \ln\left(\frac{1+s}{1+s_{\text{sa}}}\right) \quad (10)$$

The internal variable z_0 corresponds to the size of the yield surface for saturated conditions. The other internal variable z_1 is an auxiliary measure to the solution (integration) of the SFG model, and may be interpreted as a control on the shape of the yield surface. When it is smaller than z_0, the yield surface may be non-convex and the collapse due to wetting can be simulated.

The evolution for z_0 defines the hardening of the model. An isotropic hardening similar to the one used by the Cam clay model [10] is adopted. The evolution of z_1 is determined according to the stress-path, which is an interesting feature of SFG model, leading to a stress path dependent hardening law.

For elastoplastic behaviour, the suction–stress path can be measured according to the following expression (see Fig. 4):

$$\beta = \arctan\left(\frac{|\Delta \bar{p}|}{\Delta s}\right) \quad (11)$$

where $|\Delta \bar{p}|$ stands for the absolute value of a finite increment of the net mean stress, Δs is an finite increment of suction s. The computation of β is discussed at the end of Sect. 4.

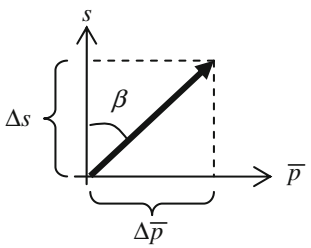

Fig. 4 β measure of the stress path

The evolution for z_0 is given by

$$\dot{z}_0 = \frac{z_0}{\lambda - \kappa} \dot{\boldsymbol{\varepsilon}}_v^p \quad (12)$$

The rate of change of the internal variable z_1 is given as a function of the rate of change of z_0:

$$\dot{z}_1 = c_{\text{path}} \times \dot{z}_0 \quad (13)$$

where c_{path} is a parameter reflecting the path-dependent hardening law. The evolution law for z_1 is coupled to that for z_0, but the main influence on the variation of z_1 is the stress path. Therefore, the stress path dependence of the model is observed in the different shapes the yield surface can have (measured by the z_1 variable). The basic requirements for the hardening law (Eq. 12) are:

- if $s > s_{\text{sa}}$
  - if $\dot{\bar{p}} > 0$ and $\dot{s} = 0$, the auxiliary internal variable z_1 must stay unchanged.
  - otherwise,
    - if $\dot{s} > 0$ and $\dot{\bar{p}} = 0$, z_1 must change at the same rate as z_0.
    - otherwise, z_1 must change at a rate proportional to the rate of z_0 so that the ratio z_1/z_0 stays constant.
- otherwise,
  - z_1 must change at a rate proportional to the rate of z_0 so that the ratio z_1/z_0 stays constant.

In this way, the behaviour of both normally consolidated and compacted materials can be captured by the system of equations. Any expression for c_{path} satisfying these requirements can be adopted. Here, we introduce the following expression:

$$c_{\text{path}} = (1 - \sin\beta)\left[1 - \left(1 - \frac{z_1}{z_0}\right)\frac{\beta}{\pi}\right] \quad (14)$$

and which is illustrated in Fig. 5; for instance, when $d\bar{p} = 0$ and $\beta = 0$, $c_{\text{path}} = 1$ and $\dot{z}_1 = \dot{z}_0$. When $ds = 0$ and $\beta = 90°$, $c_{\text{path}} = 0$ and $\dot{z}_1 = 0$. For wetting with $ds < 0$ and $\beta = 180°$, $c_{\text{path}} = z_1/z_0$ and $\dot{z}_1 = (z_1/z_0)\dot{z}_0$.

The stress–strain relationship may be derived from the above equations [11], leading to:

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D}^e : \dot{\boldsymbol{\varepsilon}} - \dot{\Lambda} \boldsymbol{D}^e : \frac{\partial f}{\partial \boldsymbol{\sigma}} + \boldsymbol{W}^e \dot{s} \quad (15)$$

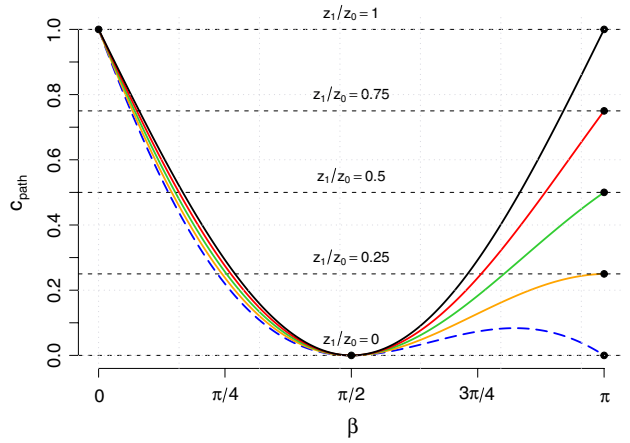

Fig. 5 Coefficient function c_{path} used in the definition of the path dependent hardening

and

$$\dot{z}_k = \dot{\Lambda} H_k \quad (k = 0 \text{ or } 1) \tag{16}$$

where

$$\dot{\Lambda} = \frac{\frac{\partial f}{\partial \boldsymbol{\sigma}} : \boldsymbol{D}^e : \dot{\boldsymbol{\varepsilon}} + \left(\frac{\partial f}{\partial \boldsymbol{\sigma}} : \boldsymbol{W}^e + \frac{\partial f}{\partial s}\right)\dot{s}}{\frac{\partial f}{\partial \boldsymbol{\sigma}} : \boldsymbol{D}^e : \frac{\partial f}{\partial \boldsymbol{\sigma}} - \frac{\partial f}{\partial z_0} H_0 - \frac{\partial f}{\partial z_1} H_1} \tag{17}$$

and

$$H_0 = \frac{z_0}{\lambda - \kappa}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}_{11}} + \frac{\partial f}{\partial \boldsymbol{\sigma}_{22}} + \frac{\partial f}{\partial \boldsymbol{\sigma}_{33}}\right), \quad H_1 = c_{\text{path}} H_0 \tag{18}$$

$\boldsymbol{D}^e$ and $\boldsymbol{W}^e$ can be found in Sheng et al. [11]. Equations (15) and (16) are used in the stress-update algorithm. For the implementation in a FEM code, the following equation is required as well:

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D}^{ep} : \dot{\boldsymbol{\varepsilon}} + \boldsymbol{W}^{ep} \dot{s} \tag{19}$$

where $\boldsymbol{D}^{ep}$ and $\boldsymbol{W}^{ep}$ are tangent modulus and are also presented in Sheng et al. [11].

4 Numerical integration

The integration of the constitutive relation is based on the Modified Euler method which is a particular case of an explicit Runge–Kutta method of second order. The embedded local error estimative is used to develop an automatic substepping scheme, which results in a very convenient and general numerical integrator for constitutive equations. Due to the non-convexity of the yield surface, the intersection-finding algorithm must be called for each sub-step (see [7]). The goal is to integrate the following incremental relationships (see e.g. [7,12]):

$$\begin{aligned} &\Delta\boldsymbol{\varepsilon} = C^e : \Delta\boldsymbol{\sigma} + \Delta\Lambda \boldsymbol{r} + \boldsymbol{w}^e \Delta s \quad \text{with} \\ &\Delta\Lambda = \frac{1}{h^p}(\boldsymbol{V} : \Delta\boldsymbol{\sigma} + S\Delta s) \end{aligned} \tag{20}$$

or

$$\begin{aligned} &\Delta\boldsymbol{\sigma} = \boldsymbol{D}^e : \Delta\boldsymbol{\varepsilon} - \Delta\Lambda \boldsymbol{D}^e : r + \boldsymbol{W}^e \Delta s \quad \text{with} \\ &\Delta\Lambda = \frac{1}{\Phi}\left[\boldsymbol{V} : \boldsymbol{D}^e : \Delta\boldsymbol{\varepsilon} + (\boldsymbol{V} : \boldsymbol{W}^e + S)\Delta s\right] \end{aligned} \tag{21}$$

where the gradients $\boldsymbol{V}$, S, and y_k are:

$$\boldsymbol{V} = \frac{\partial f}{\partial \boldsymbol{\sigma}}, \quad S = \frac{\partial f}{\partial s}, \quad \text{and} \quad y_k = \frac{\partial f}{\partial z_k} \tag{22}$$

and the hardening constants h^p and $\boldsymbol{\Phi}$ are:

$$h^p = -y_0 H_0 - y_1 H_1 \tag{23}$$

and

$$\boldsymbol{\Phi} = \boldsymbol{V} : \boldsymbol{D}^e : \boldsymbol{r} + h^p \tag{24}$$

The solution of (20) is usually computed via driver programs for given stress/suction paths and the solution of (21) is required in finite element codes during the stress-update step.

An explicit second-order accurate algorithm can be devised based on one first-order accurate FE stress-update and another of second order accuracy (named Modified–Euler). The first update is written as:

$$\begin{aligned} &\boldsymbol{\varepsilon}^{\text{FE}} = \boldsymbol{\varepsilon}^c + \Delta\boldsymbol{\varepsilon}^1 \quad \text{or} \quad \boldsymbol{\sigma}^{\text{FE}} = \boldsymbol{\sigma}^c + \Delta\boldsymbol{\sigma}^1 \quad \text{and} \\ &s^{\text{FE}} = s^c + \Delta s^1 \end{aligned} \tag{25}$$

where

$$\begin{aligned} \Delta\boldsymbol{\varepsilon}^1 = {} & \boldsymbol{C}^e(\boldsymbol{\sigma}^c, s^c) : \Delta\boldsymbol{\sigma} + \left\langle \Delta\Lambda(\boldsymbol{\sigma}^c, s^c, z_k^c) : \boldsymbol{r}(\boldsymbol{\sigma}^c, s^c) \right\rangle \\ & + \boldsymbol{w}^e(\boldsymbol{\sigma}^c, s^c)\Delta s \end{aligned} \tag{26}$$

or

$$\begin{aligned} \Delta\boldsymbol{\sigma}^1 = {} & \boldsymbol{D}^e(\boldsymbol{\sigma}^c, s^c) : \Delta\boldsymbol{\varepsilon} \\ & - \left\langle \Delta\Lambda(\boldsymbol{\sigma}^c, s^c, z_k^c)\boldsymbol{D}^e(\boldsymbol{\sigma}^c, s^c) : \boldsymbol{r}(\boldsymbol{\sigma}^c, s^c) \right\rangle \\ & + \boldsymbol{W}^e(\boldsymbol{\sigma}^c, s^c)\Delta s \end{aligned} \tag{27}$$

for which afterwards the stiffness $\boldsymbol{C}^e$ and $\boldsymbol{w}^e$ or $\boldsymbol{D}^e$ and $\boldsymbol{W}^e$ are computed a second time at the FE state obtained from (25). In the above equations, $\boldsymbol{\sigma}^c$, s^c and z_k^c stand for the current (net) stress, suction and internal variables. The FE updates are then used to calculate an intermediate increment, denoted by $\Delta\boldsymbol{\varepsilon}^2$ (in the driver program) or $\Delta\boldsymbol{\sigma}^2$ (in the finite element program) and defined according to

$$\begin{aligned} \Delta\boldsymbol{\varepsilon}^2 = {} & C^e(\boldsymbol{\sigma}^{\text{FE}}, s^{\text{FE}}) : \Delta\boldsymbol{\sigma} \\ & + \left\langle \Delta\Lambda(\boldsymbol{\sigma}^{\text{FE}}, s^{\text{FE}}, z_k^{\text{FE}}) : \boldsymbol{r}(\boldsymbol{\sigma}^{\text{FE}}, s^{\text{FE}}) \right\rangle \\ & + \boldsymbol{w}^e(\boldsymbol{\sigma}^{\text{FE}}, s^{\text{FE}})\Delta s \end{aligned} \tag{28}$$

or

$$\Delta\boldsymbol{\sigma}^2 = \boldsymbol{D}^e(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}}) : \Delta\boldsymbol{\varepsilon} - \Big\langle \Delta\Lambda(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}}, z_k^{\mathrm{FE}})\boldsymbol{D}^e(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}}) : \times \boldsymbol{r}(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}})\Big\rangle + \boldsymbol{W}^e(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}})\Delta s \tag{29}$$

The second-order accurate update (Modified Euler, ME) is thus calculated as follows

$$\boldsymbol{\varepsilon}^{\mathrm{ME}} = \boldsymbol{\varepsilon}^c + \frac{1}{2}(\Delta\boldsymbol{\varepsilon}^1 + \Delta\boldsymbol{\varepsilon}^2) \quad \text{or}$$
$$\boldsymbol{\sigma}^{\mathrm{ME}} = \boldsymbol{\sigma}^c + \frac{1}{2}(\Delta\boldsymbol{\sigma}^1 + \Delta\boldsymbol{\sigma}^2) \tag{30}$$

This method has an embedded error estimative (see e.g. [7]):

$$\mathrm{error}_{\mathrm{driver}} = \frac{\left\|\boldsymbol{\varepsilon}^{\mathrm{ME}} - \boldsymbol{\varepsilon}^{\mathrm{FE}}\right\|_2}{1 + \left\|\boldsymbol{\varepsilon}^{\mathrm{ME}}\right\|_2} \quad \text{or}$$
$$\mathrm{error}_{\mathrm{FEM}} = \frac{\left\|\boldsymbol{\sigma}^{\mathrm{ME}} - \boldsymbol{\sigma}^{\mathrm{FE}}\right\|_2}{1 + \left\|\boldsymbol{\sigma}^{\mathrm{ME}}\right\|_2} \tag{31}$$

The local error estimative, well known in Runge–Kutta embedded methods, is a powerful tool to devise algorithms with automatic determination of subincrements. If the local error is smaller than a prescribed tolerance (STOL), the stress update is accepted, otherwise, the stress/strain and suction increments are re-divided into smaller subincrements and the procedure is repeated, disregarding the recent updated values.

As the hardening law is non-linear and depends on the current state as well, the internal variables must be updated using the same strategy presented before, at the same time as the strains/stresses are updated. Nonetheless, the update of the internal variables must be done only when elastoplastic behaviour is detected (loading). In this way, the first estimative (Forward Euler) is given by

$$z_k^{\mathrm{FE}} = z_k^c + \Delta z_k^1 \tag{32}$$

where, according to hardening law,

$$\Delta z_k^1 = H_k(\boldsymbol{\sigma}^c, s^c, z_k^c)\Delta\Lambda(\boldsymbol{\sigma}^c, s^c, z_k^c) \tag{33}$$

in which both H_k and $\Delta\Lambda$ are computed at the current state. The intermediary state is obtained as follows

$$z_k^{\mathrm{ME}} = z_k^c + \frac{1}{2}(\Delta z_k^1 + \Delta z_k^2) \tag{34}$$

where

$$\Delta z_k^2 = H_k(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}}, z_k^{\mathrm{FE}})\Delta\Lambda(\boldsymbol{\sigma}^{\mathrm{FE}}, s^{\mathrm{FE}}, z_k^{\mathrm{FE}}) \tag{35}$$

Therefore, the local error during loading is defined as follows:

$$\mathrm{error}_{\mathrm{driver}} = \max\left(\frac{\left\|\boldsymbol{\varepsilon}^{\mathrm{ME}} - \boldsymbol{\varepsilon}^{\mathrm{FE}}\right\|_2}{1 + \left\|\boldsymbol{\varepsilon}^{\mathrm{ME}}\right\|_2}, \frac{\left|z_k^{\mathrm{ME}} - z_k^{\mathrm{FE}}\right|}{1 + \left|z_k^{\mathrm{ME}}\right|}\right) \tag{36}$$

Input: $\bar{\sigma}$, s, z_k, ε, Δs, and $\Delta\bar{\sigma}$ (driver) or $\Delta\varepsilon$ (FEM) ($\bar{\sigma}$ *stands for net stress*)

Output: $\Delta\varepsilon$ (driver) or $\Delta\bar{\sigma}$ (FEM)

Initialize: $T = 0$ and $\Delta T = 0.001$ *! Initial pseudo-time value and initial pseudo-time increment*

for (k **in** 1…max number of substeps)

if ($T \geq 1$) **then return** $\Delta\varepsilon$ or $\Delta\bar{\sigma}$

Substep increments: $\Delta\bar{\sigma}^T = \Delta\bar{\sigma}\Delta T$ (driver) or $\Delta\varepsilon^T = \Delta\varepsilon\Delta T$ (FEM) and $\Delta s^T = \Delta s\Delta T$

Elastic incr. ($\Delta\varepsilon^1$ or $\Delta\bar{\sigma}^1$): Eqs. (25) or (26) without plastic comp. and $\Delta z_k^1 = 0$

Find intersections and resize increments in case there are intersections

Loading/unloading decision:

if ($f_\alpha(\alpha_{checkpoint}) < 0$) **then** *IsElastic* = *TRUE* **else** *IsElastic* = *FALSE*

if ("driver") **then** Calculate $\Delta\bar{p}_\beta = (\Delta\bar{\sigma}_1^T + \Delta\bar{\sigma}_2^T + \Delta\bar{\sigma}_3^T)/3$

else if ($k = 1$) **then** Calculate $\Delta\bar{p}_\beta = (\Delta\bar{\sigma}_1^1 + \Delta\bar{\sigma}_2^1 + \Delta\bar{\sigma}_3^1)/3$

Calculate β using $\Delta\bar{p}_\beta$ and Δs^T

if (**not** *IsElastic*) **then**

Calculate gradients and hardening:

Input: $v, \bar{\sigma}, z_k, s, \beta$ *! Specific volume, net stress, internal variables, suction, path direction*

Output: $\boldsymbol{r}, V, y_k, S, H_k, h^p$ *! Flow dir., df / d$\bar{\sigma}$, df / dz$_k$, df / ds, hard. mod., hard. coef.*

Calculate $\Delta\Lambda$ (Eqs. (19) or (20)) and add the plastic component according to:

$\Delta\varepsilon^1 = \Delta\varepsilon^1 + \Delta\Lambda\boldsymbol{r}$ or $\Delta\sigma^1 = \Delta\sigma^1 - \Delta\Lambda\boldsymbol{D}^e : \boldsymbol{r}$ and $\Delta z_k^1 = \Delta\Lambda H_k$

end if

Set Forward Euler state according to Eqs. (24)

El. int. incr. ($\Delta\varepsilon^2$ or $\Delta\bar{\sigma}^2$): Eqs. (27) or (28) without the plastic comp. and $\Delta z_k^2 = 0$

if (**not** *sElastic*) **then**

Calculate gradients and hardening:

Calculate $\Delta\Lambda$ (Eqs. (19) or (20)) and add the plastic component according to:

$\Delta\varepsilon^2 = \Delta\varepsilon^2 + \Delta\Lambda\boldsymbol{r}$ or $\Delta\sigma^2 = \Delta\sigma^2 - \Delta\Lambda\boldsymbol{D}^e : \boldsymbol{r}$ and $\Delta z_k^2 = \Delta\Lambda H_k$

end if

Set Modified Euler state according to Eqs. (29)

Calculate local error estimative with Eqs. (35) or (36)

Calculate step multiplier: $m = (STOL / error)^{1/2}$

if ($error \leq STOL$) **then** *! step accepted*

Update: $T = T + \Delta T$, $s = s^{FE}$, and $z_k = z_k^{ME}$

$v = (1 - \Delta\varepsilon_1^{ME} - \Delta\varepsilon_2^{ME} - \Delta\varepsilon_3^{ME})v$ or $v = (1 - \Delta\varepsilon_1^T - \Delta\varepsilon_2^T - \Delta\varepsilon_3^T)v$

if ("driver") **then** $\bar{\sigma} = \bar{\sigma}^1$ and $\varepsilon = \varepsilon^E$

else then $\varepsilon = \varepsilon + \Delta\varepsilon^T$, $\sigma = \sigma^{ME}$ and $\Delta\bar{p}_\beta = (\Delta\bar{\sigma}_1^{ME} + \Delta\bar{\sigma}_2^{ME} + \Delta\bar{\sigma}_3^{ME})/3$

Correct any drift

if ($m > 10$) **then** $m \leftarrow 10$ *! upper bound step size change*

else *! step rejected*

if ($m < 0.01$) **then** $m \leftarrow 0.01$ *! lower bound step size change*

end if

$\Delta T \leftarrow m\Delta T$ *! next increment size*

if ($\Delta T > 1 - T$) **then** $\Delta T \leftarrow 1 - T$ *! last step*

end for

stop ('Stress-update did not converge')

Fig. 6 Stress update algorithm for driver programs and finite element codes

or

$$\mathrm{error}_{\mathrm{FEM}} = \max\left(\frac{\left\|\boldsymbol{\sigma}^{\mathrm{ME}} - \boldsymbol{\sigma}^{\mathrm{FE}}\right\|_2}{1 + \left\|\boldsymbol{\sigma}^{\mathrm{ME}}\right\|_2}, \frac{\left|z_k^{\mathrm{ME}} - z_k^{\mathrm{FE}}\right|}{1 + \left|z_k^{\mathrm{ME}}\right|}\right) \tag{37}$$

After a successful update, the drift correction must assure that the current state lies inside the yield surface with a certain tolerance. For the driver program, this correction changes

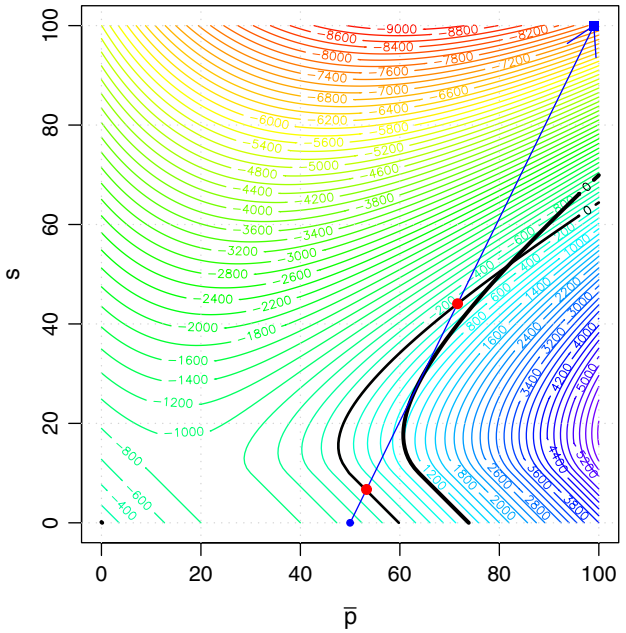

Fig. 7 Yield surface of SFG model and a stress–suction path with increasing suction. The initial state is inside the initial yield surface. After the first intersection, hardening takes place and the yield surface advances to the new position as shown

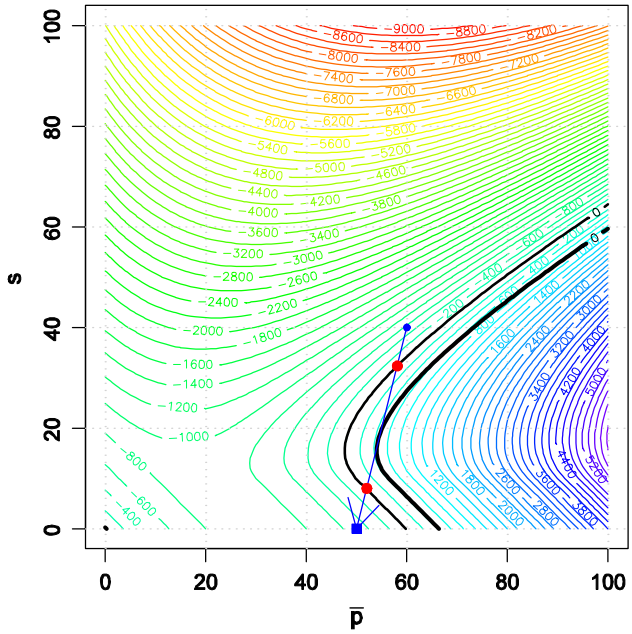

Fig. 8 Yield surface of SFG model and a stress–suction path with decreasing suction

both strain and internal variables and keeps the stress and suction increments unchanged. For the stress update stage in a Finite Element program, the correction must change the stresses and internal variables, but keep the strain and suction increments unchanged. Detailed explanation on this correction can be found elsewhere [8,9,16].

During the calculation of (33) and (35) for the two evaluations of the hardening modulus (H_k), the stress path must be known, where β is a measure of the direction of the stress path. For the driver program, as the stress/suction increments are always input for the integration, the β variable can be directly computed. However, during the stress update stage in finite element programs, a trial value of β has to be used because only the strain/suction path is known a priori.

One possible solution is the consideration of a the trial stress increment Δp_β for the evaluation of a trial direction β inside the main loop of the Runge–Kutta scheme and, then, compute the moduli H_k. The computation of the trial increment Δp_β must be made after the subdivision of the increments, but only for the first substep, because it is more accurate to update this trial increment only for those increments that leads to an acceptable local error. Therefore, the update of the trial increment Δp_β is done at the same time of that for the strain, stress, suction and internal variables (see algorithm in Fig. 6 and details concerning, for example, the numeric constants, in [7,12,16]).

Note that the suction increment is always known a priori, for both the driver and FE programs, and, hence, the β variable can be calculated for the Δp_β and Δs sub-increments. As the embedded RK method automatically accounts for the non-linearity of the system of equations (DAS) of the constitutive model, then the path direction will also be automatically adjusted. Therefore, as the numerical integration proceeds, the trial stress/suction path direction will be close to the correct path direction.

5 Simulations

We first demonstrate the numerical solutions of the intersection and the stress updates for specific stress paths. Figs. 7 and 8 show two examples where the initial stress/suction state is inside the yield surface and an intersection must be determined. In these two cases, only the first intersection is needed. The second intersection actually never happens due to hardening (inside the updated yield surface) and hence is irrelevant. However, this intersection is necessary for the loading/unloading decision. In conventional plasticity, the decision of loading/unloading is made based on the scalar product of the yield surface gradient and the trial stress increment.

Table 1 Parameters used in simulation

λ	κ	v_o [a]	ϕ	G (kPa)	s_{sa} (kPa)
0.1	0.02	1.7 (ABB′CD)	25°	100	10 (ABB′CD)
		3 (others)			100 (others)

[a] Initial specific volume at initial mean stress and suction

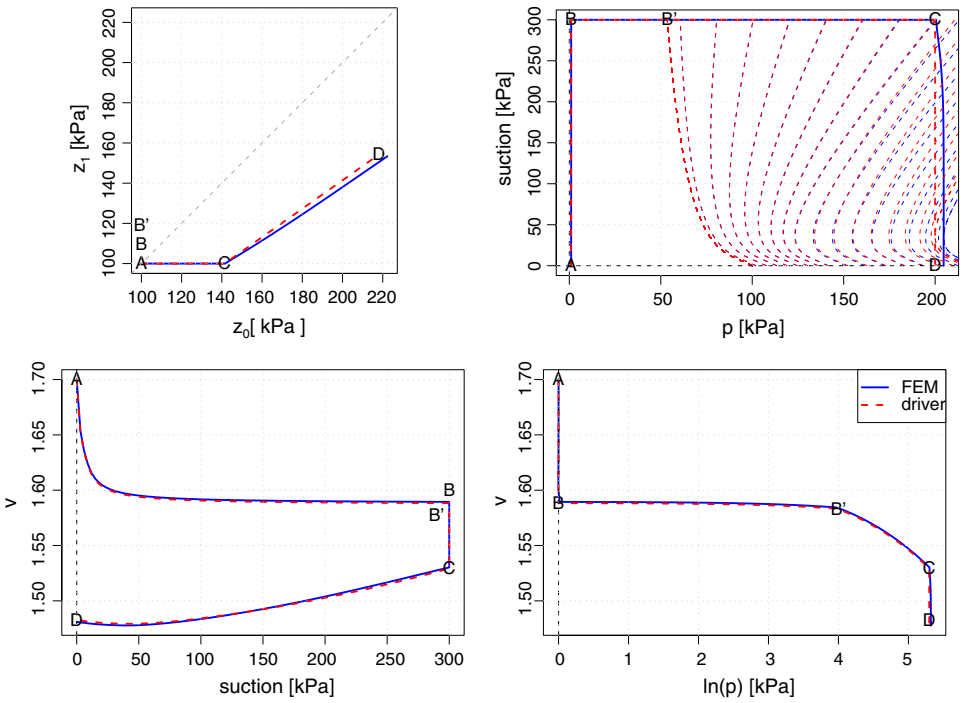

Fig. 9 Results for stress path ABB′CD

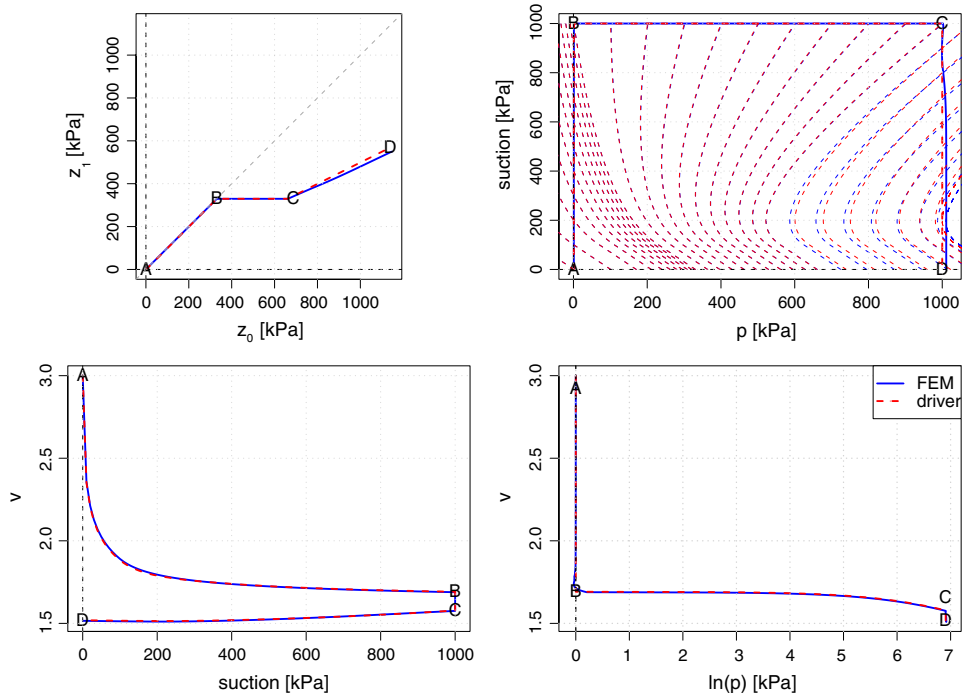

Fig. 10 Results for stress path ABCD

A positive means loading, a negative unloading and a zero neutral loading. However, these rules do not apply when the yield surface is non-convex. The second intersection is then used to decide if the stress increment is loading or unloading. The final yield surfaces are tangent to the stress paths. The material properties are listed in Table 1. It may be concluded that the algorithm performs very well in finding the appropriate intersection points.

Four different stress/suction paths are studied here to demonstrate the effectiveness of the proposed algorithms in tackling the stress path dependency and the non-convexity problems. These paths are denoted as ABB′CD, ABCD, ADCD and AFD and are shown in Figs. 9, 10, 11 and 12, respectively. In each figure, three plots are presented: the suction/mean net stress path and the corresponding yield surface evolution; the specific volume–suction relationship; and the

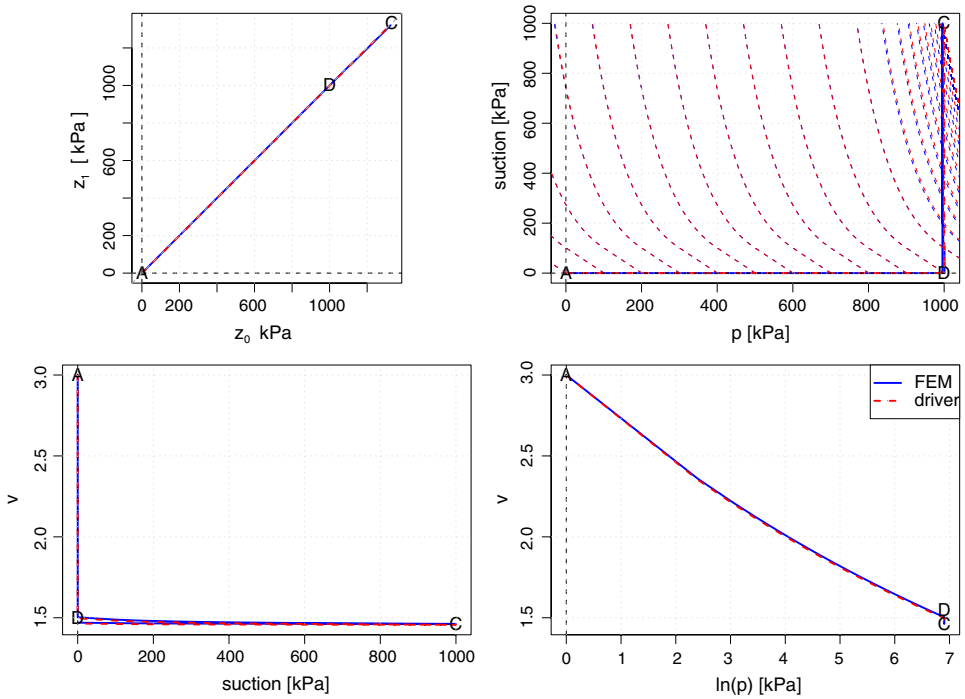

Fig. 11 Results for stress path ADCD

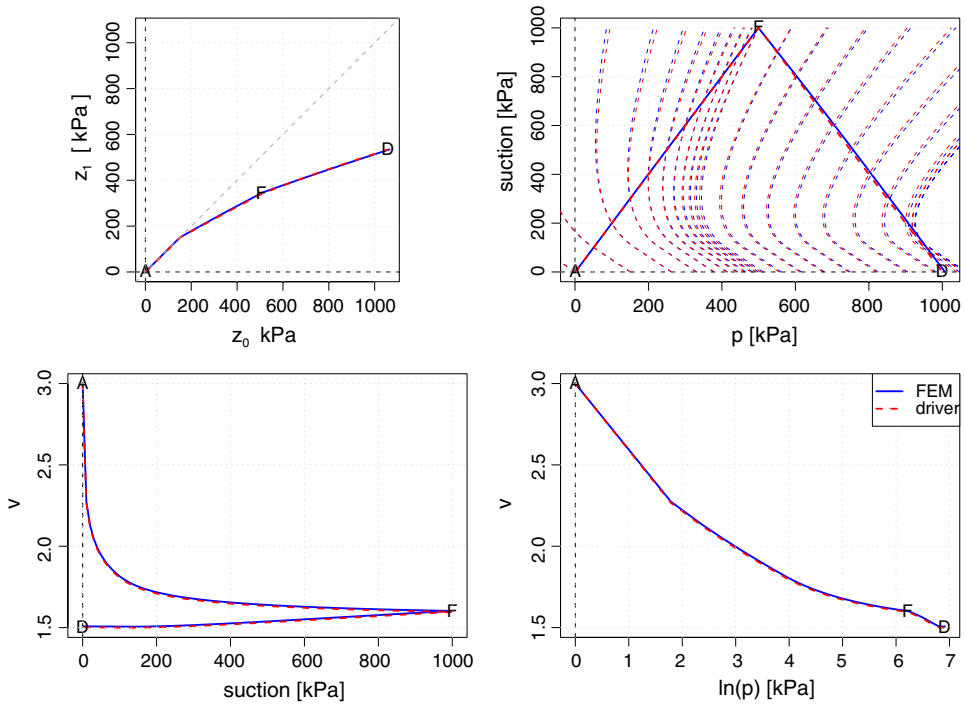

Fig. 12 Results for stress path AFD

specific volume–net mean stress relationship. The material parameters are the same as before and given in Table 1.

Test ABB′CD represents an over consolidated clay subjected to an increase in suction, followed by an increase of mean stress and decrease of suction. Tests ABCD and ADCD represent a slurry soil and are useful to check the stress/suction path dependency predicted by the SFG model. In test ABCD, the suction is increased firstly and then the mean stress is increased, followed by a decrease in suction. Test ADCD does the opposite: increase mean stress first and then increase (and decrease) the suction. Therefore, comparing the results between ABCD and ADCD tests, it is possible to observe that the predicted behaviour is different, according to the path, due to the different shapes that the yield surface can exhibit, even thought the initial and final states are the same.

The test AFD is useful to check the behaviour predicted by the SFG model, considering the path dependent hardening

introduced here. In this case, the intermediate values calculated with (13) are used, since this test is set for combined increments of mean net stress and suction.

From the results illustrated in Figs. 9, 10, 11 and 12, it is possible to conclude that the methods proposed here can readily deal with the stress-path dependency and the non-convexity of the yield surface in unsaturated soil models with robustness and accuracy. The efficiency of the stress-update algorithm, which is basically a second-order explicit Runge–Kutta method, must also be reasonable good, as it is well reported in the literature [12,15,16].

6 Conclusions

A simple method to account for the stress-path dependency during the stress update of an unsaturated soil model has been introduced. The method is based on the incorporation of a trial stress/suction increment into a second order explicit scheme. The non-convexity of the yield surface has also been considered by means of an explicit stress integration algorithm. This algorithm uses a recursive scheme to find all intersections that may arise during the stress update. The key step is the computation of the number of roots, which is done with the aid of the Kronecker–Picard formula. The only requirement for this method is that the yield function must be piecewise differentiable to the second order along the stress/suction secant path.

References

1. Alonso EE, Gens A, Josa A (1990) A constitutive model for partially saturated soils. Géotechnique 40(3):405–430
2. Brent RP (1971) An algorithm with guaranteed convergence for finding a zero of a function. Comp J 14:422–425
3. Büttner J, Simeon B (2002) Runge–Kutta methods in elastoplasticity. Appl Numer Math 41:443–458
4. Forsythe G, Malcolm M, Moler C (1990) Computer methods for mathematical computations. MIR, Moscou
5. Kavvadias DJ, Makri FS, Vrahatis MN (1999) Locating and computing arbitrarily distributed zeros. SIAM J Sci Comput 21(3):954–969
6. Papadopoulos P, Taylor RL (1994) On the application of multi-step integration methods to infinitesimal elastoplasticity. Int J Numer Methods Eng 37:3169–3184
7. Pedroso DM, Sheng D, Sloan SW (2007) Stress update algorithm for elastoplastic models with nonconvex yield surfaces. Int J Numer Methods Eng (submitted)
8. Potts DM, Gens A (1985) A critical assessment of methods of correcting for drift from the yield surface in elasto-plastic finite element analysis. Int J Numer Anal Methods Geomech 9:149–159
9. Piessens R, Doncker-Kapenga ED, Uberhuber C, Kahaner D (1983) Quadpack: a subroutine package for automatic integration. Springer, New York
10. Schofield AN, Wroth CP (1968) Critical state soil mechanics. McGraw-Hill, London
11. Sheng D, Fredlund DG, Gens A (2008) A new modelling approach for unsaturated soils using independent stress variables. Can Geotech J 45(5) (in press)
12. Sheng D, Sloan SW, Gens A, Smith DW (2003) Finite element formulation and algorithms for unsaturated soils: part I. Theory. Int J Numer Anal Methods Geomech 27:745–765
13. Sheng D, Sloan SW, Gens A (2004) A constitutive model for unsaturated soils: thermomechanical and computational aspects. Comput Mech 33(6):453–465
14. Shi P, Babuska I (1997) Analysis and computation of a cyclic plasticity model by aid of Ddassl. Comput Mech 19:380–385
15. Sloan SW (1987) Substepping schemes for the numerical integration of elastoplastic stress–strain relations. Int J Numer Methods Eng 24:893–911
16. Sloan SW, Abbo AJ, Sheng D (2001) Refined explicit integration of elastoplastic models with automatic error control. Eng Comput 18:121–154
17. Vrahatis MN, Ragos O, Skiniotis T, Zafiropoulos FA, Graspa TN (1995) RFSFNS: a portable package for the numerical determination of the number and the calculation of roots of Bessel functions. Comp Phys Commun 92:252–266
18. Wheeler SJ, Gallipoli D, Karstunen M (2002) Comments on use of the Barcelona basic model for unsaturated soils. Int J Numer Anal Methods Geomech 26:1561–1571
19. Wheeler SJ, Sharma RS, Buisson MSR (2003) Coupling of hydraulic hysteresis and stress–strain behaviour in unsaturated soils. Geotechnique 53(1):41–54

Comput Mech (2008) 42:695
DOI 10.1007/s00466-008-0284-0

ERRATUM

Non-convexity and stress-path dependency of unsaturated soil models

Daichao Sheng · Dorival M. Pedroso · Andrew J. Abbo

Published online: 22 April 2008

Erratum to: Comput Mech
DOI 10.1007/s00466-008-0268-0

The original version of this article unfortunately contained a mistake. The presentation of Fig. 7 was incorrect. The corrected figure is given below.

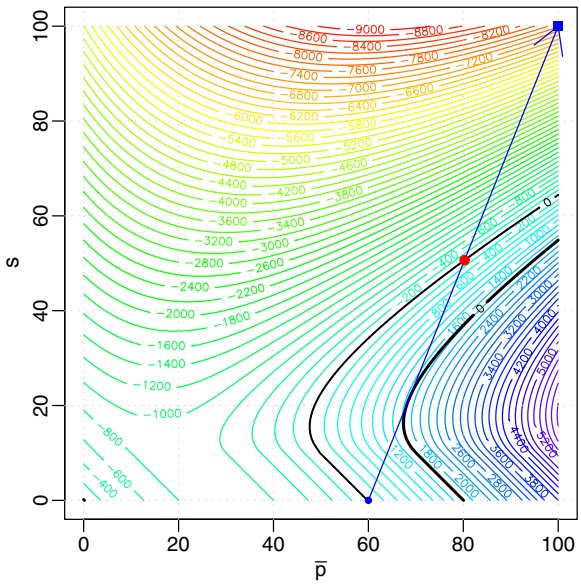

Fig. 7 Yield surface of SFG model and a stress–suction path with increasing suction. The initial state is inside the initial yield surface. After the first intersection, hardening takes place and the yield surface advances to the new position as shown

The online version of the original article can be found under doi:10.1007/s00466-008-0268-0.

D. Sheng (✉) · D. M. Pedroso · A. J. Abbo
Centre for Geotechnical and Material Modelling,
University of New Castle, New Castle, NSW 2308, Australia
e-mail: daichao.sheng@newcastle.edu.au