Finite-Volume Method Implementation of the Modified Cam-Clay Constitutive Model

Ting Bao[1]; Hanlong Liu[2]; Wengang Zhang[3]; Changbing Qin[4]; Xiangwei Fang[5]; and Zhen "Leo" Liu[6]

Abstract: This study addresses the challenge of numerically simulating nonlinear elastoplastic behavior in solid mechanics via the finite-volume method (FVM) that has not been traditionally dominant in the field. The study presents the implementation of a new FVM code tailored for nonlinear elastoplasticity of the modified Cam-clay model via OpenFOAM (version 2.3.0), an open-source C++ library primarily used for computational fluid dynamics, where OpenFOAM's capabilities are extended to, unlike conventional fluid mechanics applications, model solid mechanics problems. This implementation utilizes an implicit–explicit split strategy to effectively handle nonlinearity and displacement coupling inherent in solid mechanics. Additionally, a novel single-cell correction procedure is proposed to, when plasticity occurs, adjust stress and plastic strain incrementally. Our results, validated against benchmark tests, demonstrate a good alignment with expected outcomes and show notable computational time savings compared to the traditional FEMs by Abaqus (version 6.14) simulations for the cases examined here. The insights gained in this study reveal the potential of FVMs not only for elastoplasticity modeling but also for more complex multiphysics simulations involving coupled fluid and solid dynamics. This capability positions FVMs as a promising tool for addressing a wide range of problems in geotechnical and subsurface engineering, highlighting the innovative integration of methodologies across disciplinary boundaries. **DOI: 10.1061/IJGNAI.GMENG-9842**. © *2025 American Society of Civil Engineers*.

Author keywords: Elastoplasticity; Finite-volume method; Implicit–explicit split discretization; OpenFOAM.

Introduction

Numerical analyses play a significant role in evaluating the stability and safety of geotechnical and subsurface engineering in both the design and in-service stages. This is particularly in the case and valuable when the nonlinear mechanical behavior of geomaterials needs to be taken care of; nevertheless, analytical solutions for the problem are impossible to obtain, or field tests are difficult to conduct and/or the cost for conducting such tests is very demanding and expensive. Some examples are dynamic earthquake issues (Andrade and Borja 2006; Sevim 2011; Fatahi and Tabatabaiefar 2014), tunnel damage prevention evaluations (Lai et al. 1998; Addenbrooke and Potts 2001; Anastasopoulos et al. 2007), foundation-bearing capacity and slope stability analyses (Eid 2013; Franza and Sheil 2021; Gao et al. 2022), and assessments and applications of biogeochenics (Fauriel and Laloui 2012; Martinez et al. 2014; Wang et al. 2023a, b; Bai et al. 2024, in press; Mao et al. 2024; Yazdani et al. 2024).

The modified Cam-clay model (MCC) is one of the elastoplastic models and can characterize the nonlinear stress–strain behavior of cohesive soils subjected to the three-dimensional state of stress (Borja and Kavazaniian 1985). The MCC was developed and modified by Roscoe and Burland (1968) in terms of the original Cam-clay model (Roscoe et al. 1958) for having an ellipse for the yield surface instead of a log arc in the original form. One main problem for a log arc shape is that the yield locus at the largest value of the mean effective stress is not differentiable, which would cause stability issues in numerical analyses. Apparent advantages of the MCC include its simplicity and capability to characterize the mechanical behavior of soils realistically (Perić 2006) and that the MCC model requires a small number of material properties easily obtained from traditional laboratory tests and also contains features such as pressure sensitivity and hardening/softening responses (Borja and Lee 1990). Due to these merits, the MCC has been broadly employed to analyze the soil mechanical behavior and the associated foundation deformation or slope failure in geotechnical engineering (Borja and Lee 1990; Borja and Tamagnini 1998; Taiebat et al. 2011; El Kamash and Han 2014; Heidarzadeh 2021).

In the literature, the method to numerically implement MCC, first of all, comes to the FEM due to its popularity in computational solid mechanics. A prominent example is a series of work of Borja and his colleagues on FEM numerical formulations and modeling of soil MCC-based plasticity, including implicit integration (Borja and Lee 1990), an improved stress integration algorithm accounting for nonlinear elasticity (Borja 1991), an extension of the infinitesimal strains to finite strains (Borja and Tamagnini 1998), and an anisotropic bounding surface model with nonlinear hyperelasticity and ellipsoidal loading functions (Borja et al. 2001). Also, the MCC has been implemented in numerical codes like FLAC^{3D} (version 7.0) (Itasca 2009) using the finite-difference method (FDM). For instance, using FLAC^{3D}, El Kamash and Han (2014) showed that the verified MCC-based model could provide reasonable predictions for assessing damages caused by widening an

[1]School of Civil Engineering, Chongqing Univ., Chongqing 400044, China. Email: ting.bao@cqu.edu.cn
[2]School of Civil Engineering, Chongqing Univ., Chongqing 400044, China (corresponding author). Email: cehliu@cqu.edu.cn
[3]School of Civil Engineering, Chongqing Univ., Chongqing 400044, China. Email: cheungwg@126.com
[4]School of Civil Engineering, Chongqing Univ., Chongqing 400044, China. Email: qinchangbing@cqu.edu.cn
[5]School of Civil Engineering, Chongqing Univ., Chongqing 400044, China. Email: fangxiangwei1975@163.com
[6]Dept. of Civil and Environmental Engineering, Univ. of Virginia, Charlottesville 22904, VA. Email: leoliu@virginia.edu

Note. This manuscript was submitted on October 25, 2023; approved on October 21, 2024; published online on April 22, 2025. Discussion period open until September 22, 2025; separate discussions must be submitted for individual papers. This paper is part of the ***International Journal of Geomechanics***, © ASCE, ISSN 1532-3641.

existing column-supported embankment over soft soils and found that the maximum settlement occurs at the base of the widened portion of the embankment.

Other than FEMs and FDMs, the finite-volume method (FVM) is a well-known and the most prevalent technique in the field of computational fluid dynamics (CFD). The comparison between FDMs and FVMs is relatively few. However, in general, the FDM is based on the differential form of governing equations at each nodal point, where each derivative is approximated in terms of a Taylor series expansion. The FVM, however, discretizes the domain into finite volumes based on the integral form of governing equations. Simple schemes of FVMs could reduce to FDMs (Aleksendric and Carlone 2015). As for FVMs and FEMs, three distinctions between them appear and might make FVMs appealing: (1) FVMs directly discretize the strong integral form of governing equations, which is different from FEMs that employ the weak form to cast the strong form of governing equation before discretization; (2) FVMs usually differ from FEMs without the requirement of shape functions, and in FVMs, different shapes of finite-volume cells (i.e., elements in FEM) are discretized in the same manner; and (3) the FVM is, in general, relatively easy to use for model numerical implementations because the FVM theory is based on balancing forces acting on a volume cell. This simplicity stands in contrast to FEMs, which entail more complex mathematical frameworks.

Given the FVM benefits mentioned previously, the use of FVMs has increasingly gained momentum in the field of computational solid mechanics rather than only in CFD. The pioneering work is Demirdžic et al. (1988), who numerically simulated thermomechanical responses of the welded workpiece. Many studies then have emerged to explore FVMs for solid mechanics, and Cardiff and Demirdžić (2021) comprehensively summarized the progress of the last 30-year work in this regard. It turns out that FVMs have been successfully employed in viscoelasticity (Demirdžić et al. 2005), elastoplasticity (Maneeratana 2000), thermo-hydroelasticity (Demirdžić et al. 2000), poroelasticity and poro-elastoplasticity (Tang et al. 2015; Asadollahi 2017), fracture mechanics (Cardiff et al. 2015), finite strains and rotations (Bijelonja et al. 2005; Cardiff et al. 2012a), and many others. At some point, FVMs have some advantages of modeling solid mechanics problems. For example, a block-coupled finite-volume methodology could be more efficient in both computer memory and computation cost than Abaqus (version 6.11-1) FEMs for linear elasticity with unstructured meshes (Cardiff et al. 2016). Additionally, the FVM solver for contact mechanics exhibits a flexible ability to maintain contact convergences with larger values of the scaling factor, whereas small values are typically used in FEM simulations (Cardiff et al. 2012b). So, the success in various FVM applications for solid mechanics above has shown the intrinsic and good ability of FVMs to deal with nonlinear mechanics problems. However, despite the achievements from the aforementioned work, efforts are still needed to develop new FVM-based numerical codes, specifically those for soil and rock mechanics for geotechnical engineering, and to the best of the authors' knowledge, no research is reported on the FVM methodology for the MCC.

This study thus presents the implementation of a new numerical FVM-based code for the MCC as an example of model demonstration for elastoplasticity. This FVM-based implementation methodology is generally applicable to any material model besides MCC, contributing to the field of computational solid mechanics using an alternative, FVMs. The implementation is achieved using OpenFOAM (version 2.3.0), which is the popular open-source C++ library originally developed by Weller et al. (1998) and has been widely used for solid mechanics mentioned above and multiphysics simulations (Greenshields et al. 1999; Cardiff et al. 2015; Tang et al. 2015; Tuković et al. 2018; Bao and Liu 2019; Bao et al. 2020a, b). This could make this FVM solver public to be freely used and to be feasibly coupled with pore-water pressure and temperature, etc., in the same FVM framework for multiphysics simulations. In the implementation, the implicit–explicit split strategy used by Demirdžic et al. (1988) and Tang et al. (2015) is adopted here to handle the nonlinearity and displacement coupling in the MCC. However, a novel approach, i.e., a single-cell correction procedure, is proposed to correct the incremental stress and plastic strain when plasticity occurs. The FVM implementation of the MCC into OpenFOAM is verified and tested against triaxial compression tests and a foundation-bearing capacity problem.

The organization of this paper is as follows. The "Model Theory" section presents the essential assumptions and concepts of the MCC for the FVM implementation. In the "Numerical Method" section, the FVM methodology is detailed, including governing equations, FVM discretization, solution algorithm and workflow, and analytical solutions used for the MCC implementation verification. The "Numerical Test Examples" section presents the application of the implementation to triaxial compression tests and a foundation-bearing capacity problem including a verification analysis. A discussion is presented in the "Discussion" section regarding insights and a comparison analysis with Abaqus FEM simulations. A follow-up is conclusions and recommendations for future efforts.

Model Theory

The soil elastoplastic model adopted in this paper is the MCC. As its detailed derivation can be found in the published work (Roscoe et al. 1958; Roscoe and Burland 1968), this section presents some crucial assumptions and concepts of the MCC below.

Three variables in MCC are used to characterize the state of a soil specimen, i.e., effective mean stress p', deviatoric stress q, and specific volume V_s. p' and q can be computed by the effective principal stresses σ'_1, σ'_2, and σ'_3 via

$$\begin{cases} p' = -\dfrac{\sigma'_1 + \sigma'_2 + \sigma'_3}{3} \\ q = \dfrac{\sqrt{(\sigma'_1 - \sigma'_2)^2 + (\sigma'_2 - \sigma'_3)^2 + (\sigma'_3 - \sigma'_1)^2}}{\sqrt{2}} \end{cases} \tag{1}$$

Here and throughout, strain and stress are both taken to be negative in compression. This sign convention has been adopted here to align with the default settings in OpenFOAM, although a positive convention is often used in soil mechanics. A negative sign is thus added to p', rendering its sign positive. The specific volume V_s is calculated in terms of the void ratio e via $V_s = 1 + e$. The critical state where the soil distorts at a constant state of stress without a volume change is described by the critical state line (CSL). Fig. 1(a) shows the CSL in the p'–q plane and the MCC elliptic yield curve. The slope of the CSL is defined as $M = q/p'$, which is one of the soil characteristics. In the yield curve, p'_0 is the largest value of the mean effective stress, also known as the preconsolidation pressure, in determining the size of the yield surface. The MCC yield function is formulated by

$$Y = M^2 p'^2 - M^2 p'_0 p' + q^2 \tag{2}$$

The CSL in the $V_s - \ln p'$ space is parallel to the normal compression line (NCL) shown in Fig. 1(b). So, both lines have the

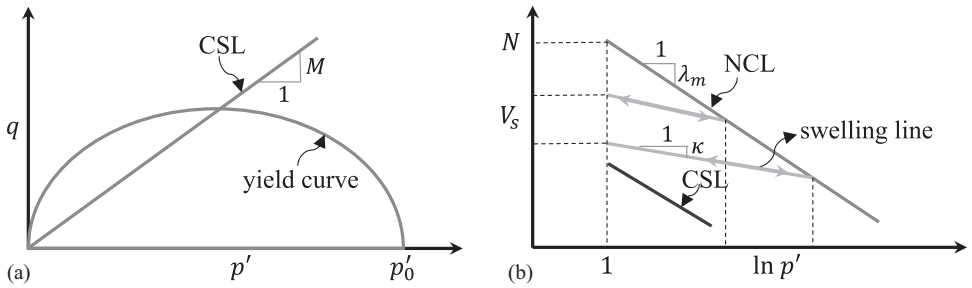

Fig. 1. Soil mechanical behavior for the MCC: (a) MCC yield surface; and (b) NCL and swelling line under compression. (Reprinted with permission from Springer Nature: Springer, *Acta Geotechnica*, "Performance of constitutive models in predicting behavior of remolded clay," L. S. Bryson and A. Salehian, © 2011.)

same slope λ_m. Using λ_m and V_s, the NCL can be calculated by

$$V_s = N_m - \lambda_m \ln p' \tag{3}$$

where N_m = specific volume at unit mean effective stress $p' = 1$ kPa. Under compression for drained conditions, the swelling line with the slope κ characterizes the unloading–reloading behavior of soils [Fig. 1(b)]. So, in addition to M, λ_m, N_m, and κ are also characteristic properties of soils. Those four properties are soil-dependent and can be determined by triaxial compression tests (Silvestri and Abou-Samra 2011; Dai and Qin 2013). Note that M can also be determined by $M = 6 \sin \varphi'/(3 - \sin \varphi')$ if the friction angle at the critical state φ' is given (Silvestri and Abou-Samra 2011).

There are three more parameters and/or assumptions needed to illustrate for constitutive modeling of the MCC. The first is the assumption of hardening and softening responses, depending on the location of the yielding occurrence. The softening behavior is exhibited if yielding occurs to the left side of the point at which the CSL intersects a yield surface in Fig. 1(a) (i.e., shrinking). The hardening behavior, however, appears if yielding occurs to the right side of that point (i.e., expanding). Such hardening is mainly discussed in this paper.

The second is the overconsolidation ratio (OCR) defined as $\text{OCR} = p'_0/p'$ to know the state where the maximum stress level is previously experienced. So, OCR can be determined when p'_0 is given. OCR greater than, equal to, and less than 1, respectively, represents an over-, normal-, and underconsolidated state at which soils have experienced in comparison with the present stress level p'.

The last is the determination of which soil elastic parameter to be a constant for MCC constitutive modeling. Usually, three elastic parameters are utilized, i.e., shear modulus G, bulk modulus K, and Poisson's ratio v. For the MCC, K is not constant but changes with p', computed by

$$K = \frac{V_s p'}{\kappa} \tag{4}$$

So, MCC constitutive modeling calculates K via Eq. (4) without a prior specification. Yet, either G or v needs to be specified as a constant to compute the other one via

$$\begin{cases} v = \dfrac{3K - 2G}{2G + 6K}, & \text{given } G \\ G = \dfrac{3K(1 - 2v)}{2(1 + v)}, & \text{given } v \end{cases} \tag{5}$$

For example, if v is given to be a constant, G is no longer a constant and is calculated via Eq. (5). Each as a constant is fine for MCC constitutive modeling, but this study adopts v as a constant for illustration in the numerical implementation.

Numerical Method

Governing Equation and FVM Implementation

For a solid body element, the momentum equation for steady-state conditions without the body force is governed by (Dym and Shames 1973)

$$\nabla \cdot \boldsymbol{\sigma} = \nabla \cdot (\boldsymbol{\sigma}' - p_w \mathbf{I}) = \mathbf{0} \tag{6}$$

where $\boldsymbol{\sigma}$ = total stress tensor; p_w = pore fluid pressure; ∇ = divergence operator; and $\mathbf{I}$ = identity tensor (note stress is negative in compression). For a drained condition mainly discussed here, $\boldsymbol{\sigma} = \boldsymbol{\sigma}'$ is true, and we thus utilized $\boldsymbol{\sigma}$ and ignored the superscript ($'$) for all variables in the following for simplicity.

The constitutive relation between stress and strain for linear isotropic elasticity can be expressed in terms of the incremental form

$$d\boldsymbol{\sigma} = \mathbf{D}_e [d\boldsymbol{\varepsilon}_t - d\boldsymbol{\varepsilon}_p] \tag{7}$$

where $\mathbf{D}_e$ = fourth-order stiffness matrix (note that $\mathbf{D}_e$ is formulated by Lame's constants μ and λ in OpenFOAM for linear elasticity, where μ is the shear modulus and λ is the first Lame constant. Here and throughout, $\mu = G$ holds true, meaning that both have the same shear modulus. Since the use of μ is conventional in FVM, the following adopts μ for illustration); $d\boldsymbol{\varepsilon}_t$ is the total strain, and $d\boldsymbol{\varepsilon}_p$ is the plastic strain. Employing the small-strain assumption, $d\boldsymbol{\varepsilon}_t$ can be computed using the incremental displacement $d\mathbf{D}$ (Dym and Shames 1973)

$$d\boldsymbol{\varepsilon}_t = \frac{1}{2}[\nabla(d\mathbf{D}) + \nabla(d\mathbf{D})^T] \tag{8}$$

where ∇ = gradient operator. Substituting Eqs. (7) and (8) into Eq. (6), we obtain the steady-state governing equation for the MCC with $\mathbf{D}_e$ expressed by μ and λ (Greenshields 2015)

$$\nabla \cdot \{\underbrace{[\mu \nabla(d\mathbf{D})^T + \lambda \mathbf{I} tr(\nabla d\mathbf{D}) + \mu \nabla(d\mathbf{D})]}_{\text{total strain part}} - \underbrace{[2\mu(d\boldsymbol{\varepsilon}_p) + \lambda \mathbf{I} tr(d\boldsymbol{\varepsilon}_p)]}_{\text{plastic strain part}}\} = \mathbf{0} \tag{9}$$

where $tr()$ = trace operator. The three components associated with $d\mathbf{D}$ in Eq. (9) are coupled with each other. To deal with this in FVM simulations, the segregated solution approach is usually employed, in which there is only one unknown field per equation and each system of equations is solved individually and in sequence (Versteeg and Malalasekera 2007). Here, this segregated approach is applied in a way that the total strain part in Eq. (9) is rewritten by arranging $\mu\nabla(d\mathbf{D})$ and $\lambda tr(\nabla d\mathbf{D})$ in the form shown in Eq. (10) to have both explicit and implicit terms [plastic strain part for nonlinearity is also treated explicitly], also called an implicit–explicit split

strategy detailed later in the "FVM Discretization" section

$$\underbrace{\nabla\cdot[(2\mu+\lambda)\nabla(d\mathbf{D})]}_{\text{implicit}}=\overbrace{\underbrace{\nabla\cdot[2\mu(d\boldsymbol{\varepsilon}_p)+\lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]}_{\text{explicit}}}^{\text{plastic}}-\underbrace{\nabla\cdot[\mu\nabla(d\mathbf{D})^T+\lambda\mathbf{I}tr(\nabla d\mathbf{D})-(\mu+\lambda)\nabla(d\mathbf{D})]}_{\text{explicit}} \tag{10}$$

This arrangement could well improve the convergence of the FVM global iterative solution procedure, as has been done by Tang et al. (2015) by balancing the explicit and implicit terms associated with $d\mathbf{D}$.

The boundary condition prescribed on a boundary of a computational domain could be traction, displacement, or pressure in a fixed or varying type. The treatment of displacement conditions is straightforward, yet the fulfillment of the momentum balance on boundary patches (i.e., surfaces) in OpenFOAM, if traction or pressure is applied, should be guaranteed because an external force or pressure applied on boundary patches will be treated as contributions into the source term (Tang et al. 2015). Since $d\mathbf{D}$ is computed in FVM simulations via Eq. (10), traction force and pressure conditions need to be converted into a form formulated in terms of $d\mathbf{D}$ derived from $d\mathbf{T}=d\boldsymbol{\sigma}\cdot\mathbf{n}$ (Greenshields 2015)

$$\nabla(d\mathbf{D})\cdot\mathbf{n}=\frac{d\mathbf{T}-dp_{bc}\cdot\mathbf{n}}{2\mu+\lambda}+\frac{[2\mu(d\boldsymbol{\varepsilon}_p)+\lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]\cdot\mathbf{n}}{2\mu+\lambda}-\frac{[\mu\nabla(d\mathbf{D})^T+\lambda\mathbf{I}tr(\nabla d\mathbf{D})-(\mu+\lambda)\nabla(d\mathbf{D})]\cdot\mathbf{n}}{2\mu+\lambda} \tag{11}$$

where $d\mathbf{T}=$ incremental traction; $dp_{bc}=$ incremental pressure; $d\boldsymbol{\sigma}$ comprises one implicit term and two explicit terms derived from Eq. (10) excluding the divergence operation; and $\mathbf{n}=$ outward-pointing surface/patch unit normal vector on the boundary. Eq. (11) ensures that any traction or pressure prescribed on boundary patches will be internally transformed into a displacement term. Note that only one traction or pressure input is, in general, needed for boundary conditions in FVM simulations. For instance, $d\mathbf{T}$ is a zero tensor if dp_{bc} is defined on a boundary patch. It is also worth mentioning that the displacement coupling and nonlinearity in Eq. (11) will be treated and solved using the implicit–explicit split strategy as well.

FVM Discretization

The FVM requires the discretization of time, space, and model governing equations ready for conducting simulations. For the MCC, since the steady-state condition is considered, time discretization is avoidable here, but the time in FVM is discretized into a finite number of time increments in governing equations to be solved using schemes like the Euler implicit method.

There are several space discretization methods available in the literature, such as the cell-centered approach (Demirdžic et al. 1988; LeVeque 2002) and the vertex-centered approach (Fryer et al. 1991; Xia et al. 2007). Here, we adopt the cell-centered approach, and Fig. 2 shows a typical control volume Ω_P consisting of the computational point P in the cell center, face f and its area vector $\mathbf{S}_f$, center point N_c of a neighboring cell that shares the same face f, distance vector $\mathbf{d}_f$ between P and N_c through f, and positional vector $\mathbf{x}_p$. This space discretization fashion is the same for any polyhedral cell shape of Ω_P such as tetrahedral and hexahedral with no difference. It should be noted that in some cases with nonorthogonality, for example, the conventional five-point stencil FVM scheme in solving two-dimensional (2D) problems may

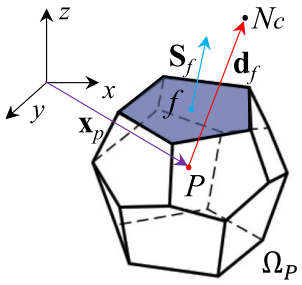

Fig. 2. General polyhedral control volume. (Reprinted with permission from Springer Nature: Springer, *Archives of Computational Methods in Engineering*, "Thirty years of the finite volume method for solid mechanics," P. Cardiff and I. Demirdžić, © 2021.)

cause some issues of nonconvergence; therefore, the impact on the FVM accuracy caused by nonorthogonality with some stencil FVM schemes and unstructured meshes like tetrahedra should be carefully assessed to add nonorthogonality or skewness corrections and other treatments in FVM calculations (Moraes et al. 2013). These, however, are not considered and discussed here. In this study, structured meshes are utilized to discretize the FVM computational domain.

The discretization of Eq. (10) applied to each finite-volume cell and Eq. (11) applied on each boundary patch is the volume integral over Ω_P and then converted to the surface integral using Gauss's theorem (Katz 1979). The surface integral for Eq. (10) [the same form for Eq. (11)] is formulated by

$$\overbrace{\underbrace{\oint_{\mathbf{S}}\mathbf{n}\cdot[(2\mu+\lambda)\nabla(d\mathbf{D})]dS}_{\text{implicit}}}^{\text{total}}=\overbrace{\underbrace{\oint_{\mathbf{S}}\mathbf{n}\cdot[2\mu(d\boldsymbol{\varepsilon}_p)+\lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]dS}_{\text{explicit}}}^{\text{plastic}}-\overbrace{\underbrace{\oint_{\mathbf{S}}\mathbf{n}\cdot[\mu\nabla(d\mathbf{D})^T+\lambda\mathbf{I}tr(\nabla d\mathbf{D})-(\mu+\lambda)\nabla(d\mathbf{D})]\,dS}_{\text{explicit}}}^{\text{total}} \tag{12}$$

Because both implicit and explicit parts are involved, there is a slight difference between implicit discretization and explicit discretization as in the following for isotropic and homogeneous conditions considered here.

For the explicit discretization, two explicit terms in Eq. (12) are approximated via the following equations, respectively, by converting the closed surface integral of Eq. (12) into a sum of linear variations of the values across each cell face:

$$\oint_{\mathbf{S}}\mathbf{n}\cdot[2\mu(d\boldsymbol{\varepsilon}_p)+\lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]dS=\sum_{f=1}^{NFaces}\mathbf{n}_f\cdot[2\mu(d\boldsymbol{\varepsilon}_p)_f+\lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)_f]S_f \tag{13}$$

$$\oint_{\mathbf{S}}\mathbf{n}\cdot[\mu\nabla(d\mathbf{D})^T+\lambda\mathbf{I}tr(\nabla d\mathbf{D})-(\mu+\lambda)\nabla(d\mathbf{D})]dS=\sum_{f=1}^{NFaces}\mathbf{n}_f\cdot[\mu[\nabla(d\mathbf{D})]_f^T+\lambda\mathbf{I}tr[(\nabla d\mathbf{D})]_f-(\mu+\lambda)[\nabla(d\mathbf{D})]_f]S_f \tag{14}$$

where $[(\nabla(d\mathbf{D})]_f$ and $(d\boldsymbol{\varepsilon}_p)_f$ at the surface centroid are the two key components that correlate with two neighboring cells. The central differencing scheme with a second-order accuracy is employed to calculate these two components in terms of cell centroid values available from the last iteration using (Greenshields 2015)

$$[(\nabla(d\mathbf{D})]_f = f_x[(\nabla(d\mathbf{D})]_P + (1-f_x)[\nabla(d\mathbf{D})]_N \tag{15}$$

$$(d\boldsymbol{\varepsilon}_p)_f = f_x(d\boldsymbol{\varepsilon}_p)_P + (1-f_x)(d\boldsymbol{\varepsilon}_p)_N \tag{16}$$

where f_x = interpolation weight within 0 and 1, computed by

$$f_x = \frac{|x_{N_f} - x_f|}{|(x_f - x_P) + (x_{N_f} - x_f)|} \tag{17}$$

Substituting Eqs. (15) and (16) into Eqs. (13) and (14), respectively, we obtain the discretized forms for the two explicit terms in Eq. (10) as follows:

$$\begin{aligned}\oint_S \mathbf{n}\cdot[2\mu(d\boldsymbol{\varepsilon}_p) + \lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]dS &= \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot\{2\mu[f_x(d\boldsymbol{\varepsilon}_p)_P + (1-f_x)(d\boldsymbol{\varepsilon}_p)_N\}S_f \\ &+ \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot\{\lambda\mathbf{I}[f_x tr(d\boldsymbol{\varepsilon}_p)_P + (1-f_x)tr(d\boldsymbol{\varepsilon}_p)_N\}S_f\end{aligned} \tag{18}$$

$$\begin{aligned}\oint_S \mathbf{n}\cdot[\mu\nabla(d\mathbf{D})^T + \lambda\mathbf{I}tr(\nabla d\mathbf{D}) - (\mu+\lambda)\nabla(d\mathbf{D})]\,dS &= \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot[\mu\{f_x[\nabla(d\mathbf{D})]_P^T + (1-f_x)[\nabla(d\mathbf{D})]_N^T\}]S_f \\ &+ \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot[\lambda\mathbf{I}\{f_x tr[\nabla(d\mathbf{D})]_P + (1-f_x)tr[\nabla(d\mathbf{D})]_N\}]S_f \\ &+ \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot[-(\mu+\lambda)\{f_x[\nabla(d\mathbf{D})]_P + (1-f_x)[\nabla(d\mathbf{D})]_N\}]S_f\end{aligned} \tag{19}$$

As for the implicit discretization of the implicit term in Eq. (10), no interpolation like Eq. (15) is needed to compute $[\nabla(d\mathbf{D})]_f$, but instead, $[\nabla(d\mathbf{D})]_f$ is calculated via the assumption of a linear variation across f using

$$\begin{aligned}\oint_S \mathbf{n}\cdot[(2\mu+\lambda)\nabla(d\mathbf{D})]dS &= \sum_{f=1}^{NFaces} \mathbf{n}_f\cdot(2\mu+\lambda)[\nabla(d\mathbf{D})]_f S_f \\ &= \sum_{f=1}^{NFaces} (2\mu+\lambda)|\mathbf{n}_f|\frac{(d\mathbf{D})_N - (d\mathbf{D})_P}{|d_f|}S_f\end{aligned} \tag{20}$$

It is worth noting that the traction boundary condition is discretized into $\nabla(d\mathbf{D})$ on boundary patches, as mentioned previously, explicitly treated as contributions to the source term in the FVM iterative solution procedure. The treatment of boundary conditions utilizes the same implicit–explicit split strategy, so Eq. (11) can be rewritten in the following converted manner for this strategy to update the traction boundary iteratively, formulated by

$$\overbrace{\underbrace{(2\mu+\lambda)[\nabla(d\mathbf{D})]_{bc}\cdot\mathbf{n}_{bc}}_{\text{implicit}}}^{\text{total}} = \overbrace{(d\mathbf{T})_{bc} - dp_{bc}\cdot\mathbf{n}_{bc}}^{\text{tractionandpressure}} + \overbrace{\underbrace{[2\mu(d\boldsymbol{\varepsilon}_p) + \lambda\mathbf{I}tr(d\boldsymbol{\varepsilon}_p)]\cdot\mathbf{n}}_{\text{explicit}}}^{\text{plastic}}$$
$$- \overbrace{\underbrace{[\mu\nabla(d\mathbf{D})^T + \lambda\mathbf{I}tr(\nabla d\mathbf{D}) - (\mu+\lambda)\nabla(d\mathbf{D})]\cdot\mathbf{n}}_{\text{explicit}}}^{\text{total}} \tag{21}$$

Solution Algorithm and Workflow

The "Governing Equation and FVM Implementation" section outlines the governing equations and variables pertinent to the MCC model. In the "FVM Discretization" section, the FVM is employed to discretize these equations, transforming them into solvable algebraic forms suitable for numerical computation. This section elaborates on the solution algorithm and workflow in OpenFOAM, explaining how the discretized equations are implemented and solved computationally using the specific approach and methodologies of OpenFOAM. Key variables addressed in the workflow include incremental displacements and derived strain and stress, which are discretized across the computational grid using customized numerical techniques.

The segregated approach introduced previously is the solution algorithm used here to solve the discretized form of Eq. (10) with specified boundary conditions. This approach temporarily segregates three parts of Eq. (10) to be solved sequentially within an iterative sequence, implemented in OpenFOAM shown below, instead of creating one and a big matrix equation for the entire system.

Because the implicit and explicit terms are solved individually, as shown in Fig. 3, the Geometric Agglomerated Algebraic Multigrid method with the Gauss–Seidel scheme as a smoother (Golub and Van Loan 2013) is used to provide the necessary coupling for $d\mathbf{D}$ during FVM calculations, where the solution tolerance for $d\mathbf{D}$ is set to 10^{-6}.

It is evident that solving Eq. (10) eventually yields a solution for $d\mathbf{D}$. The total displacement is updated by a sum of the current converged $d\mathbf{D}_n$ and the previous total displacement $\mathbf{D}_{n-1}$, and so is the total stress $\boldsymbol{\sigma}$, where $d\boldsymbol{\sigma}$ is calculated internally using $\nabla(d\mathbf{D})$ (see sigmaD in Fig. 3). The solution workflow for updating $\mathbf{D}$ and $\boldsymbol{\sigma}$ is shown in Fig. 4. Important remarks on $\mathbf{D}$ and $\boldsymbol{\sigma}$ updates are as follows:

Remark 1. The first solution for $d\boldsymbol{\sigma}_n$ in Step 2 is an elastic stress guess using the soil elastic parameters μ and λ without the plastic strain, namely, $(d\boldsymbol{\varepsilon}_p)_n = \mathbf{0}$ in Step 1. In this step, the initial Young's modulus E_i and Poisson's ratio v are used to calculate μ and λ.

Remark 2. In Step 3, the yield function is evaluated by Eq. (2) for all finite-volume cells and boundary patches by accessing them separately in OpenFOAM using *forAll(variable,celli)* and *forAll(variable.boundaryField(),patchi)* in terms of the principal stresses. In other words, both $d\boldsymbol{\sigma}$ and $d\boldsymbol{\varepsilon}_t$ calculated from Step 2 needed to be transformed into the principal space first, and then, we used them to calculate p and q via Eq. (1) and update K and G via Eqs. (4) and (5).

Remark 3. OpenFOAM has preimplemented *Foam::eigen* for calculating eigenvalues and eigenvectors. This function, however, always gave complex eigenvalues in OpenFOAM (version 2.3.0) calculations for cases we considered. So, a new function for computing eigenvalues and eigenvectors was implemented to solve this issue.

Remark 4. When plasticity occurs, a novel approach, i.e., single-cell correction procedure, is proposed in Step 4 to correct $d\boldsymbol{\sigma}$

```
fvVectorMatrix dDEqn
(
   fvm::laplacian(2*mu + lambda, dD, "laplacian(DdD,dD)") // implicit
part
 ==
   fvc::div
   (
        2*mu * deps_p + lambda * I * tr(deps_p), // explicit plastic
strain
        "div(sigmaP)"
   )
   - divSigmaExp
);
{
   volTensorField graddD(fvc::grad(dD));
   sigmaD = mu*twoSymm(graddD) + (lambda*I)*tr(graddD);

   divSigmaExp = fvc::div  // divSigmaExp: explicit part
   (
        sigmaD - (2*mu + lambda)*graddD,
        "div(sigmaD)"
   );
}
```

Fig. 3. FVM code for illustration of the *segregated approach* implementation and sigmaD calculation. (Note that in OpenFOAM, *fvm::* and *fvc::* represent the implicit and explicit discretization operations, respectively, *Laplacian* $=\nabla^2$, *div* $=\nabla\cdot$, and *grad* $=\nabla$.)

and $d\boldsymbol{\varepsilon}_p$, in which a new elastoplastic stiffness matrix $\mathbf{D}_{ep}$ is constructed to calculate $d\boldsymbol{\sigma}$. The idea is that we can calculate $d\boldsymbol{\sigma}$ via two different ways expressed as

$$d\boldsymbol{\sigma} = \mathbf{D}_e[d\boldsymbol{\varepsilon}_t - d\boldsymbol{\varepsilon}_p] = \mathbf{D}_{ep} d\boldsymbol{\varepsilon}_t \tag{22}$$

The former [i.e., Eq. (7)] in Eq. (22) is generally used in OpenFOAM, but in Step 4, we used the latter to build $\mathbf{D}_{ep}$ in terms of $\mathbf{D}_e$ and the yield function Y via

$$\mathbf{D}_{ep} = \mathbf{D}_e - h\frac{\left(\mathbf{D}_e\dfrac{\partial Y}{\partial \boldsymbol{\sigma}}\right)\left(\left[\dfrac{\partial Y}{\partial \boldsymbol{\sigma}}\right]^T \mathbf{D}_e\right)}{\left(\left[\dfrac{\partial Y}{\partial \boldsymbol{\sigma}}\right]^T \mathbf{D}_e \dfrac{\partial Y}{\partial \boldsymbol{\sigma}}\right) - \mathbf{K}_p} \tag{23}$$

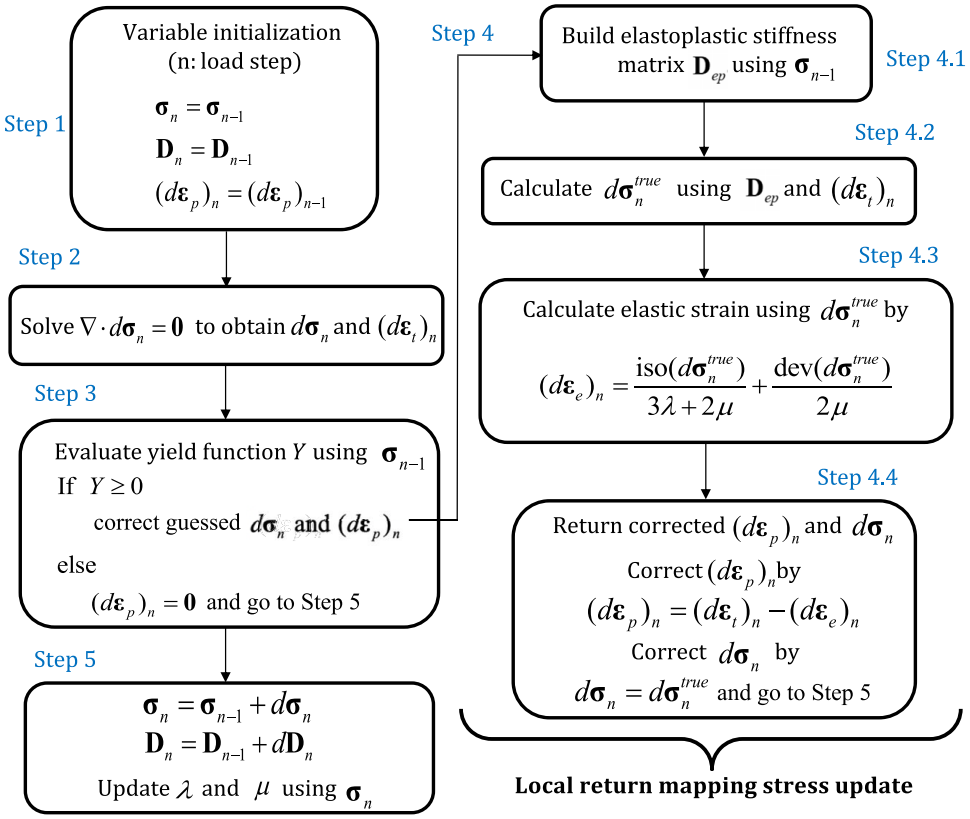

Fig. 4. Solution workflow for the MCC. (Note that the load step in OpenFOAM here basically denotes the number of incremental load iterations to be simulated.)

Int. J. Geomech., 2025, 25(7): 04025122

where $\mathbf{K}_p = [\text{var},\text{var},\text{var},0,0,0]^T$; $\text{var} = -pp_0(1+e_i)/1-\kappa$; e_i = initial void ratio; and h = Heaviside step function ($h = 1$ if plastic; otherwise, $h = 0$). $\mathbf{D}_e$ in Eq. (23) is constructed in terms of varying K and G and constant v.

Remark 5. Using the principal strains, the volumetric and deviatoric strains can be calculated by $\varepsilon_v = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ and (if ε_1 is in the loading direction) $\varepsilon_q = 2(\varepsilon_1 - \varepsilon_3)/3$, respectively, for a triaxial compression test.

Remark 6. Because $d\boldsymbol{\sigma}^{\text{true}}$ is calculated using the principal stresses, this $d\boldsymbol{\sigma}^{\text{true}}$ should be returned back to the general space based on the preserved eigenvectors in Step 4.4.

Remark 7. After the solution procedure has obtained the corrected $d\boldsymbol{\sigma}$ and $d\boldsymbol{\varepsilon}_p$ in Step 4.4 but before it goes to Step 5, p, q and V_s are updated using the corrected total stress via Eqs. (1) and (24). Then, updated p and V_s are used to update K, G, and λ accordingly. In this situation, the material properties thus log one load step behind the corresponding computation, but this error is small if the loading step is small

$$V_s = N_m - \lambda_m \ln p_0 + \kappa \ln (p_0/p) \tag{24}$$

The aforementioned MCC solution procedure has been implemented as a newly developed solver called MCCFoam in OpenFOAM. The verification and application of this solver are detailed in the "Numerical Test Examples" section.

MCC Analytical Solution

The verification of the MCC implementation is an essential step to examine the performance of the new MCCFoam solver developed in this study. MCC analytical solutions will be a benchmark used to evaluate the accuracy of this solver. The MCC analytical solution for drained conditions adopted here comes from Perić (2006) for the reduced form without one invariant θ from the three-invariant Cam-clay model. The following shows two equations for the total volumetric strain ε_v and the deviatoric strain ε_q utilized below, and more details can be found in the cited reference previously:

$$\varepsilon_v = \varepsilon_{v,i} + \frac{1}{V_{s,i}} \ln \left[\left(\frac{p}{p_i} \right)^{-\lambda_m} \left(\frac{M^2 + (q/p)^2}{M^2 + (q_i/p_i)^2} \right)^{\kappa - \lambda_m} \right] \tag{25}$$

$$\varepsilon_q = \varepsilon_{q,i} + \frac{1}{V_{s,i}} \left\{ \ln \left[\left(\frac{M - q/p}{M - q_i/p_i} \right)^{3(\lambda_m - \kappa)/M(M-3)} \left(\frac{M + q/p}{M + q_i/p_i} \right)^{3(\lambda_m - \kappa)/M(M+3)} \left(\frac{3 - q/p}{3 - q_i/p_i} \right)^{[6(\lambda_m - \kappa)/(9-M^2)] - [2\kappa(1+v)/3(1-2v)]} \right] \right\} - \frac{2(\lambda_m - \kappa)}{M V_{s,i}} \left[\arctan\left(\frac{q/p}{M} \right) - \arctan\left(\frac{q_i/p_i}{M} \right) \right] \tag{26}$$

where subscript i = initial condition of a given variable.

Numerical Test Examples

Verification against Analytical Solution

Two verification examples are presented in this section for two clays with different MCC parameters and sample shapes (i.e., cube and cylinder) to assess the performance of the MCCFoam solver. The first example is carried out using London clay with a cubic shape detailed in Silvestri and Abou-Samra (2011). As shown in Fig. 5, the clay specimen is numerically discretized into 125 finite-volume cells with an equal-edge size of 4 mm to simulate a triaxial compression test. MCC parameters for this clay and boundary conditions prescribed in the FVM simulation are tabulated in Table 1. It is a reminder that we use the zero pressure condition here because the solver computes the incremental displacement rather than the total one.

A comparison between FVM predictions and analytical solutions computed by Eq. (26) for q versus ε_q and p versus ε_a is shown in Fig. 6, where the relative error is also plotted. In this comparison, the cell shown in Fig. 5 is randomly selected from the domain as an example to obtain FVM simulation results. Two loading increments applied on the $+Y$ boundary are evaluated to test the loading impact. It is seen in Fig. 6 that the FVM simulation results agree very well with the analytical solutions for the two loading increments analyzed here.

The second example is conducted using the Fujian soft marine clay introduced by Dai and Qin (2013). The clay specimen was shaped into a cylinder 80 mm in height and 39.1 mm in diameter

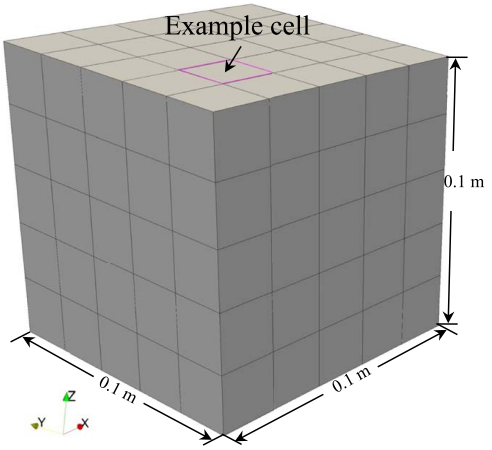

Fig. 5. Cubic clay specimen used in the simulation.

Table 1. MCC parameters for the London clay and boundary conditions used in the simulation

Parameter	Value	Boundary	Boundary condition
p_0 (kPa)	206	$+X$	Zero pressure and $d\mathbf{D}$
λ_m	0.161	$-X$	Zero pressure and $d\mathbf{D}$
κ	0.062	$+Y$	Pressure increment
v	0.3	$-Y$	Zero $d\mathbf{D}$
E_i (MPa)	7.975	$+Z$	Zero pressure and $d\mathbf{D}$
M	0.888	$-Z$	Zero pressure and $d\mathbf{D}$
N_m	2.858	—	—
e_i	1.0002	—	—

Note: The soil parameters are referenced from Silvestri and Abou-Samra (2011)

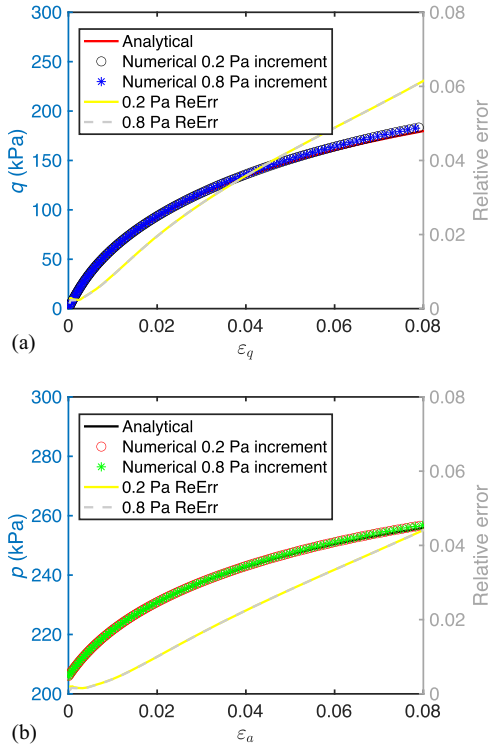

Fig. 6. FVM predictions and analytical solutions: (a) q versus ε_q; and (b) p versus ε_a.

[Fig. 7(a)]. The MCC parameters were experimentally determined, and their values are detailed in Table 2. Here, we simulate one of the consolidated-drained triaxial tests in Dai and Qin (2013) with a confining pressure of 200 kPa. In the simulation, the specimen discretization yields 18,000 finite-volume cells with an edge size of approximately 2 mm for each. A 0.388 kPa pressure increment is prescribed on the top surface of the specimen until 8 mm

Table 2. MCC parameters for the Fujian soft marine clay

Parameter	Value
p_0 (kPa)	200
λ_m	0.1297
κ	0.0322
V	0.3
E_i (MPa)	16.83
M	1.344
N_m	2.9452
e_i	1.258

Note: The parameters are referenced from Dai and Qin (2013).

deformation is finally induced in the loading direction, as shown in Fig. 7(b).

Fig. 8 plots the comparison between FVM results and analytical solutions for q versus ε_v. Similarly, an example cell shown in Fig. 7(a) is chosen for FVM predictions. We can see that the FVM predictions overlap the analytical solutions completely. The comparisons in Figs. 6 and 8 indicate the high accuracy of the MCCFoam solver developed in this study to simulate the soil elastoplastic behavior via the FVM.

Application in Bearing Capacity of Strip Footing

This section demonstrates the application of the MCCFoam solver to analyze the load–deflection behavior of a flexible strip foundation on clay detailed in Borja and Tamagnini (1998) and Heidarzadeh (2021). Briefly, a uniformly distributed load is applied gradually over a half width of 2 m on a 5-m-thick and 20-m-long layer of soft clay (Fig. 9). The clay is assumed to be homogeneous and isotropic and underlies the strip footing with a rough and rigid boundary at the base. The MCC parameters for this clay are detailed in Table 3 together with the boundary conditions applied on the FVM boundary patches illustrated in Fig. 9. Specifically, the left uses the symmetry boundary type as it is a symmetrical plane. Zero traction/pressure and

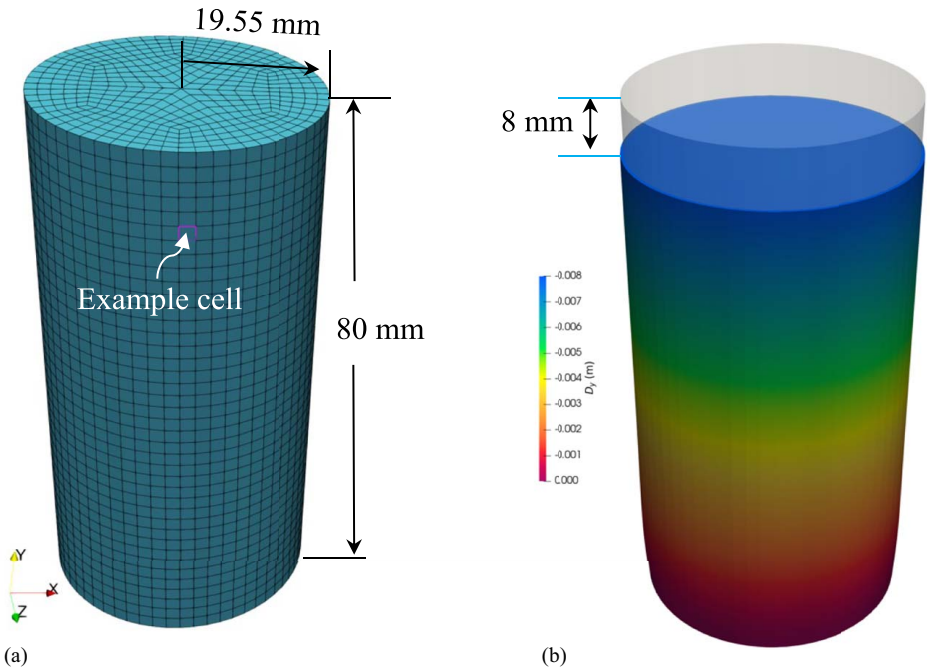

Fig. 7. Fujian soft marine clay: (a) configuration of the cylindrical specimen used in the simulation; and (b) deformation after shearing.

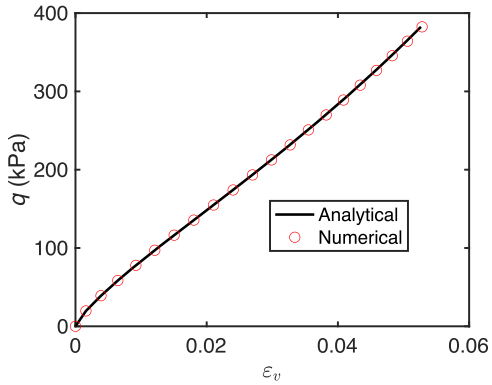

Fig. 8. FVM predictions and analytical solutions for q versus ε_v.

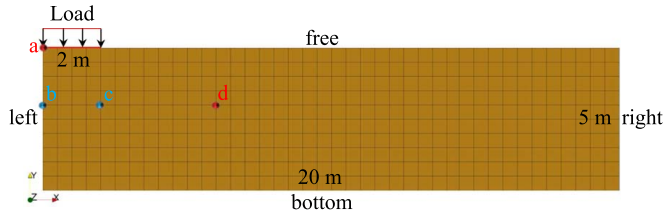

Fig. 9. 2D FVM finite-volume cell layout for a strip footing problem.

Table 3. Parameters and boundary conditions for 2D strip footing simulation

Parameter	Value	Boundary	Boundary condition
p_0 (kPa)	20	Left	Symmetry
λ_m	0.13	Right	Zero $d\mathbf{D}$
κ	0.018	Free	Zero traction and $d\mathbf{D}$
V	0.3	Load	Pressure increment
K_i (MPa)	1.9	Bottom	Zero $d\mathbf{D}$
E_i (MPa)	2.28	Front	Empty
M	1.05	Back	Empty
N_m	2.1	—	—
e_i	0.7106	—	—

Note: The soil parameters are referenced from Borja and Tamagnini (1998) and Heidarzadeh (2021).

displacement are specified on the free surface as it is the ground surface. The empty boundary type is applied on the front and back which are the two planes perpendicular to the z-axis. Because a 2D case similar to that in Borja and Tamagnini (1998) is considered here, the FVM calculation, therefore, does not involve the front and back planes.

The soil domain is discretized into 400 finite-volume cells with 0.5×0.5 m for each. $\boldsymbol{\sigma} = 20$ kPa is uniformly assumed as the initial stress of the clay domain. Here, a load of 20 kPa is applied on the load surface and is simulated as the vertical stress gradually increasing from 0 to 20 kPa by an equal increment of 0.4 kPa. During this process, the MCC constitutive model is utilized for evaluating the bearing capacity of the clay that is assumed to be fully saturated.

Fig. 10 shows the vertical deformation profile when the load reaches its maximum value of 20 kPa, where D_y deformation is observed. For example, Points a, b, and c marked in Fig. 9 have the maximum D_y deformation of -0.144, -0.074, and -0.04 m, respectively, as shown in Fig. 11(a). Because of the vertical load from the strip footing, σ_y at locations just underneath the strip footing (e.g., Points a, b, and c) decreases gradually with increasing load [Fig. 11(b)], meaning that the soil gets more compressive at Points a, b, and c. However, σ_y at Point d tends to increase a bit after the loading, which means that the soil at this point becomes less compressive. This is due to the free load at the top of Point d, so the squeezed soil from the lateral due to the strip footing loading results in an uplift in some soils adjacent to the loading area (see Fig. 10, where some soils have a positive D_y value), causing the soil at Point d, for example, to become slightly less compressive.

Significant deformation will occur if the load is increased to, for example, 50 kPa. The small-strain assumption then does not hold in this situation, and the finite-strain numerical approach is required. Although not a focus of this study, it is worth noting that if the finite-strain assumption is used in the MCC, the strip footing simulation would predict smaller D_y deformation than the small-strain assumption assumed here. This is mainly because the finite-strain solution will yield more significant D_x deformation, especially on the ground surface (Borja and Tamagnini 1998).

Discussion

Computational Efficiency Analysis

For any numerical approach, two important aspects that we care about are prediction accuracy and computational efficiency. The results in the "Numerical Test Examples" section have demonstrated the good accuracy of the MCCFoam solver against the analytical solutions for the cases examined previously. Another importance of using the FVM is to assess its computational efficiency relative

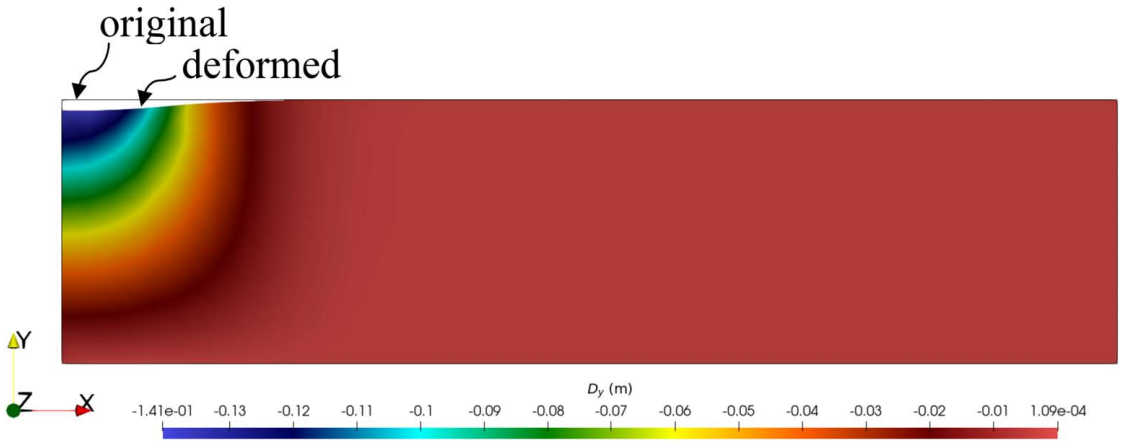

Fig. 10. Vertical deformation at Load = 20 kPa in the FVM simulation.

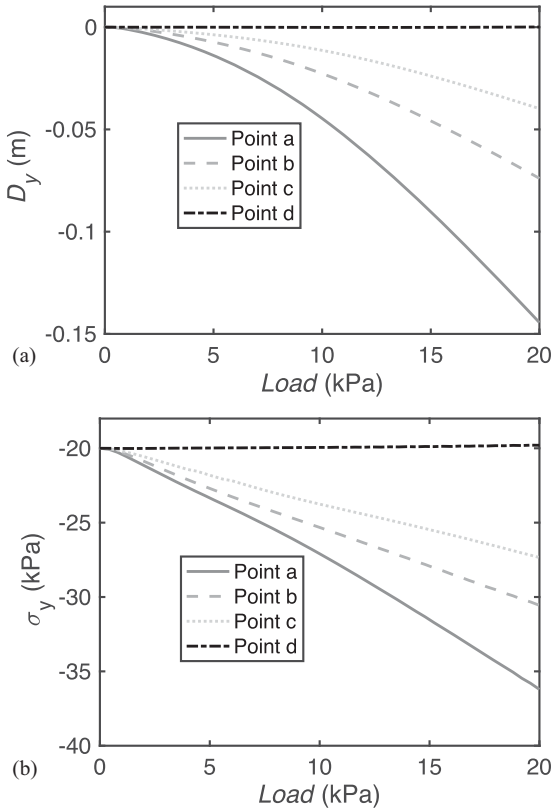

Fig. 11. FVM results change with gradually increasing load: (a) D_y versus Load; and (b) σ_y versus Load.

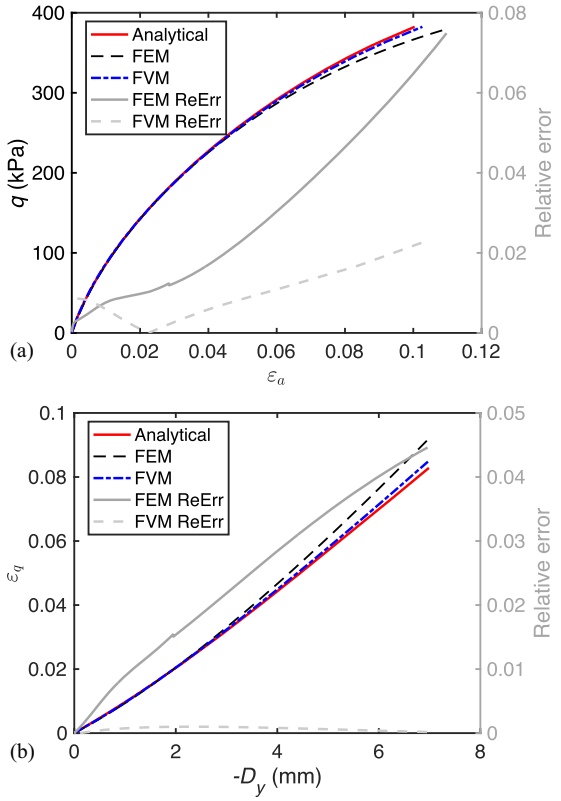

Fig. 12. Comparison of FVM and FEM predictions for a triaxial compression test: (a) ε_a versus q; and (b) $-D_y$ versus ε_q.

to other numerical methods. For this purpose, we compared FVM predictions with FEM results via a popular commercial code, Abaqus, for the triaxial compression test using the C3D8 element (i.e., the second example in the "Verification against Analytical Solution" section). The Abaqus simulation uses the direct sparse solver, while the OpenFOAM simulation uses the geometric agglomerated algebraic multigrid solver. Both simulations use the fixed load step, have almost the identical number of cells/elements (FVM uses a few more; Table 5), and are conducted using one core on a laptop computer with 12th Gen Intel I Core I i7-12700 2.10 GHz and 32.0 GB RAM.

As shown in Fig. 12, where the analytical solutions are also included for a better comparison, FVM and FEM results are close to each other. By comparison, FVM predictions surprisingly have a better agreement than FEM results in contrast to analytical solutions for the case discussed here because both Fig. 12 and Table 4 results for relative and absolute errors show that FVM predictions have smaller errors and higher accuracy. This might be because the FEM and FVM implementations for the MCC are very different in solution algorithms and calculation procedures. It is, however, not completely fair to conclude that FVM predictions are more accurate than FEM predictions because the published work (Perre and Passard 1995; Fang et al. 2002; Cardiff et al. 2016) has shown that both FVM and FEM predictions are closely comparable without significant differences.

For the consumed time, the computational cost between the two numerical methods is compared in Table 5 for three cases with different mesh sizes. As can be seen, the Abaqus FEM scenario has used more time for running a total time of 100 s. The OpenFOAM FVM scenario is less expensive in computation and has obtained an increase of over 98% in efficiency compared to the Abaqus FEM scenario for all three mesh size cases considered here. In addition to the comparison here for single core computing, the results from the strip foundation problem on the pore-elastoplastic soil (Tang et al. 2015) have also indicated that parallel computing using OpenFOAM FVM has taken 17 min for the drained case and 33 min for the undrained case, while these computational times are much less than 15 and 6 h for the two cases taken by Abaqus FEM parallel computing, respectively. All of these do not mean that the FVM is more accurate or faster than the FEM in terms of simulation time and accuracy, but based on the results obtained here, the analysis suggests that besides the FEM, the FVM is a valuable alternative numerical method with the desired computational efficiency and thus deserved to be used for numerical analyses of engineering problems.

FVM Application Potential for Geomechanics Modeling

This study has implied the potential of employing the FVM to solve many more linear and/or nonlinear geomechanics problems (the MCC illustrated as an example here) instead of the FEM that is always the first choice for many decades. One example is that based upon the numerical implementation framework presented here, the MCC implementation would be feasibly extended to the three-invariant model for elastoplasticity (Alawaji et al. 1992) by adding one more stress invariant θ other than p and q and also

Table 4. Relative and absolute errors compared to analytical solutions

	Max. relative error			Max. absolute error		
Method	ε_q (%)	ε_v (%)	ε_a (%)	ε_q	ε_v	ε_a
Abaqus FEM	8.11	4.46	7.49	0.0069	0.0023	0.0076
OpenFOAM FVM	2.67	0.33	2.26	0.0022	0.0002	0.0023

Table 5. Comparison of computational cost for FEM and FVM

Method	Mesh size (mm)	No. cell/element	Load step (MPa)	Total time (s)	Wall clock time (s)	Increase in efficiency to Abaqus (%)
FEM	10	144	0.1	100	156	—
	5	1,312			311	
	2	17,680			4,800	
FVM	10	160	0.1	100	1	99.4
	5	1,322			5	98.4
	2	18,000			83	98.3

extended to the Barcelona basic model for unsaturated-soil elastoplasticity (Alonso et al. 1990) and MCC-based poro-elastoplasticity with the consideration of suction and pore pressure coupling, respectively.

Besides that, using the FVM has advantages in performing multiphysics simulations involving geomechanics coupling. For example, fluid–solid interaction problems need to couple two fields of solid and fluid for analyses of many engineering problems. The existing methods, in most cases, rely on the coupling of either FEM–FVM, FDM–FVM, or other approaches (Rutqvist 2011; Rutqvist et al. 2011; To et al. 2020; Asadi and Ataie-Ashtiani 2021; Bao et al. 2021a, b; Bao and Burghardt 2022; Wang et al. 2022; Yang et al. 2024), where the FVM is solving fluid and the FEM/FDM is solving solid. This way of coupling treats fluid and solid governing equations in different implementation frameworks, which is thus difficult to program and may frequently cause convergence issues during information transfer and/or if a strongly nonlinear coupling exists in equations. However, utilizing the FVM could allow both fluid and solid to be performed and solved in the same FVM framework. This framework could straightforwardly examine fluid–solid equations, easily debug the program, and is unnecessary to know theories of two numerical methods well at the same time.

Conclusions

This paper presents a new numerical code for modeling nonlinear elastoplasticity based on the FVM. The MCC, chosen as an example for materials such as soft soils, has been implemented into OpenFOAM. The MCC implementation has been examined by examples including analytical solutions and FEM/FDM results for triaxial compression tests and bearing capacity of strip footing problems. The accuracy of our MCC implementation has been shown through a good agreement between FVM predictions and benchmarks. The comparison analysis with Abaqus FEM simulations also shows that the MCC solver developed based on the FVM is more computationally efficient in saving time and cost.

Take advantage of the open-source platform, OpenFOAM whose FVM classes and functions are free to use, the FVM-based methodology presented in this study has revealed the application potential of using the FVM as a viable alternative for multiphysics simulations of many geotechnical and subsurface engineering problems. This study is the first attempt of our research at looking into FVM-based modeling of nonlinear elastoplasticity, where drained conditions are involved only and the segregated solution approach is employed to consider the computational expense. Further research is, of course, needed to extend the current study. Examples of future work include integrating the kinematic condition of incompressibility into the current FVM-based numerical framework to model the MCC behavior under undrained conditions, considering soil anisotropy, dilation, over-, and underconsolidation conditions, analyzing the scalability of the FVM implementation concerning memory usage and runtime efficiency for field-scale applications, and developing a block-coupled approach in terms of recent advances in FVM to improve the poor convergence that the segregated solution approach may experience if a strong coupling exists in equations.

Data Availability Statement

All data sets generated or used during the study appear in the published article.

Acknowledgments

The authors acknowledge the financial support from the National Natural Science Foundation of China (No. 52378326) and the Chongqing Talents Program (No. cstc2021ycjh-bgzxm0051).

Notation

The following symbols are used in this paper:

$\mathbf{D}_e$ = elastic stiffness tensor;
$\mathbf{D}_{ep}$ = elastoplastic stiffness tensor;
$d\mathbf{D}$ = incremental displacement;
dp_{bc} = incremental pressure;
$d\mathbf{T}$ = incremental traction;
$d\boldsymbol{\varepsilon}_p$ = incremental plastic strain;
$d\boldsymbol{\varepsilon}_t$ = incremental total strain;
$d\boldsymbol{\sigma}^{\text{true}}$ = true incremental stress after correction;
E = Young's modulus;
e = void ratio;
e_i = initial void ratio;
G = shear modulus;
h = Heaviside step parameter;
$\mathbf{I}$ = identity tensor;
K = bulk modulus;
M = CSL slope;
N = specific volume;
N_m = specific volume at unit mean stress;
n = current load step;
$n-1$ = previous load step;
p = mean stress;
p_w = pore fluid pressure;
p_0 = preconsolidation pressure;
q = deviatoric stress;
V_s = specific volume;
v = Poisson's ratio;
ε_q = deviatoric strain;
ε_v = total volumetric strain;
κ = slope of the swelling line;
λ = Lame's constant;
λ_m = NCL slope;

μ = G directly;
$\boldsymbol{\sigma}$ = total stress tensor; and
φ' = friction angle at the critical state.

References

Addenbrooke, T., and D. Potts. 2001. "Twin tunnel interaction: Surface and subsurface effects." *Int. J. Geomech.* 1 (2): 249–271. https://doi.org/10.1061/(ASCE)1532-3641(2001)1:2(249).

Alawaji, H., K. Runesson, S. Sture, and K. Axelsson. 1992. "Implicit integration in soil plasticity under mixed control for drained and undrained response." *Int. J. Numer. Anal. Methods Geomech.* 16 (10): 737–756. https://doi.org/10.1002/nag.1610161004.

Aleksendric, D., and P. Carlone. 2015. "Soft computing techniques." In *Soft computing in the design and manufacturing of composite materials: Applications to brake friction and thermoset matrix composites*, edited by Aleksendric and Carlone, 39–60. Amsterdam, The Netherlands: Woodhead.

Alonso, E. E., A. Gens, and A. Josa. 1990. "A constitutive model for partially saturated soils." *Géotechnique* 40 (3): 405–430. https://doi.org/10.1680/geot.1990.40.3.405.

Anastasopoulos, I., N. Gerolymos, V. Drosos, R. Kourkoulis, T. Georgarakos, and G. Gazetas. 2007. "Nonlinear response of deep immersed tunnel to strong seismic shaking." *J. Geotech. Geoenviron. Eng.* 133 (9): 1067–1090. https://doi.org/10.1061/(ASCE)1090-0241(2007)133:9(1067).

Andrade, J. E., and R. I. Borja. 2006. "Quantifying sensitivity of local site response models to statistical variations in soil properties." *Acta Geotech.* 1 (1): 3–14. https://doi.org/10.1007/s11440-005-0002-4.

Asadi, R., and B. Ataie-Ashtiani. 2021. "Hybrid finite volume-finite element methods for hydro-mechanical analysis in highly heterogeneous porous media." *Comput. Geotech.* 132: 103996. https://doi.org/10.1016/j.compgeo.2020.103996.

Asadollahi, M. 2017. "Finite volume method for poroelasticity." Master's thesis, Dept. of Geoscience & Engineering, Delft Univ. of Technology.

Bai, S., K. Li, and C. Li. 2024, in press. "Triaxial compression test of MICP sand column and simulation of failure process." *Biogeotechnics* 100071. https://doi.org/10.1016/j.bgtech.2024.100071.

Bao, T., and J. Burghardt. 2022. "A Bayesian approach for in-situ stress prediction and uncertainty quantification for subsurface engineering." *Rock Mech. Rock Eng.* 55: 4531–4548. https://doi.org/10.1007/s00603-022-02857-0.

Bao, T., J. Burghardt, V. Gupta, E. Edelman, B. McPherson, and M. White. 2021a. "Experimental workflow to estimate model parameters for evaluating long term viscoelastic response of CO_2 storage caprocks." *Int. J. Rock Mech. Min. Sci.* 146: 104796. https://doi.org/10.1016/j.ijrmms.2021.104796.

Bao, T., J. Burghardt, V. Gupta, and M. White. 2021b. "Impact of time-dependent deformation on geomechanical risk for geologic carbon storage." *Int. J. Rock Mech. Min. Sci.* 148: 104940. https://doi.org/10.1016/j.ijrmms.2021.104940.

Bao, T., H. Cao, Y. Qin, G. Jiang, and Z. L. Liu. 2020a. "Critical insights into thermohaline stratification for geothermal energy recovery from flooded mines with mine water." *J. Cleaner Prod.* 273: 122989. https://doi.org/10.1016/j.jclepro.2020.122989.

Bao, T., S. Liu, Y. Qin, and Z. L. Liu. 2020b. "3D modeling of coupled soil heat and moisture transport beneath a surface fire." *Int. J. Heat Mass Transfer* 149: 119163. https://doi.org/10.1016/j.ijheatmasstransfer.2019.119163.

Bao, T., and Z. L. Liu. 2019. "Geothermal energy from flooded mines: Modeling of transient energy recovery with thermohaline stratification." *Energy Convers. Manage.* 199: 111956. https://doi.org/10.1016/j.enconman.2019.111956.

Bijelonja, I., I. Demirdzic, and S. Muzaferija. 2005. "A finite volume method for large strain analysis of incompressible hyperelastic materials." *Int. J. Numer. Methods Eng.* 64 (12): 1594–1609. https://doi.org/10.1002/nme.1413.

Borja, R. I. 1991. "Cam-clay plasticity, part II: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor." *Comput. Methods Appl. Mech. Eng.* 88 (2): 225–240. https://doi.org/10.1016/0045-7825(91)90256-6.

Borja, R. I., and E. Kavazaniian. 1985. "A constitutive model for the stress–strain–time behaviour of 'wet' clays." *Géotechnique* 35 (3): 283–298. https://doi.org/10.1680/geot.1985.35.3.283.

Borja, R. I., and S. R. Lee. 1990. "Cam-clay plasticity, part 1: Implicit integration of elasto-plastic constitutive relations." *Comput. Methods Appl. Mech. Eng.* 78 (1): 49–72. https://doi.org/10.1016/0045-7825(90)90152-C.

Borja, R. I., C.-H. Lin, and F. Montáns. 2001. "Cam-clay plasticity, part IV: Implicit integration of anisotropic bounding surface model with nonlinear hyperelasticity and ellipsoidal loading function." *Comput. Methods Appl. Mech. Eng.* 190 (26–27): 3293–3323. https://doi.org/10.1016/S0045-7825(00)00301-7.

Borja, R. I., and C. Tamagnini. 1998. "Cam-clay plasticity part iii: Extension of the infinitesimal model to include finite strains." *Comput. Methods Appl. Mech. Eng.* 155: 73–95. https://doi.org/10.1016/S0045-7825(97)00141-2.

Bryson, L. S., and A. Salehian. 2011. "Performance of constitutive models in predicting behavior of remolded clay." *Acta Geotech.* 6 (3): 143–154. https://doi.org/10.1007/s11440-011-0144-5.

Cardiff, P., and I. Demirdžić. 2021. "Thirty years of the finite volume method for solid mechanics." *Arch. Comput. Methods Eng.* 28 (5): 3721–3780. https://doi.org/10.1007/s11831-020-09523-0.

Cardiff, P., A. Ivanković, A. Karać, and Ž. Tukovic. 2012a. "Development of a finite volume based structural solver for large rotation of non-orthogonal meshes." In *Proc., 7th OpenFOAM Workshop*. Darmstadt, Germany: The OpenFOAM® Workshop Committee.

Cardiff, P., A. Karac, and A. Ivankovic. 2012b. "Development of a finite volume contact solver based on the penalty method." *Comput. Mater. Sci.* 64: 283–284. https://doi.org/10.1016/j.commatsci.2012.03.011.

Cardiff, P., R. Manchanda, E. C. Bryant, D. Lee, and M. M. Sharma. 2015. "A finite volume method for the simulation of hydraulic fractures." In *Proc., Joint Symp. of Irish Mechanics Society and Irish Society for Scientific & Engineering Computation*. Dublin, Ireland: University College Dublin.

Cardiff, P., Ž. Tuković, H. Jasak, and A. Ivanković. 2016. "A block-coupled finite volume methodology for linear elasticity and unstructured meshes." *Comput. Struct.* 175: 100–122. https://doi.org/10.1016/j.compstruc.2016.07.004.

Dai, Z.-h., and Z.-z. Qin. 2013. "Numerical and theoretical verification of modified cam-clay model and discussion on its problems." *J. Cent. South Univ.* 20 (11): 3305–3313. https://doi.org/10.1007/s11771-013-1854-7.

Demirdžić, I., E. Džaferović, and A. Ivanković. 2005. "Finite-volume approach to thermoviscoelasticity." *Numer. Heat Transfer, Part B* 47 (3): 213–237. https://doi.org/10.1080/10407790590901675.

Demirdžić, I., I. Horman, and D. Martinovic. 2000. "Finite volume analysis of stress and deformation in hygro-thermo-elastic orthotropic body." *Comput. Methods Appl. Mech. Eng.* 190 (8–10): 1221–1232. https://doi.org/10.1016/S0045-7825(99)00476-4.

Demirdžic, I., P. Martinovic, and A. Ivankovic. 1988. "Numerical simulation of thermal deformation in welded workpiece." *Zavarivanje* 31 (5): 209–219.

Dym, C. L., and I. H. Shames. 1973. *Solid mechanics*. Claremont, CA: Springer.

Eid, H. T. 2013. "Bearing capacity and settlement of skirted shallow foundations on sand." *Int. J. Geomech.* 13 (5): 645–652. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000237.

El Kamash, W., and J. Han. 2014. "Displacements of column-supported embankments over soft clay after widening considering soil consolidation and column layout: Numerical analysis." *Soils Found.* 54 (6): 1054–1069. https://doi.org/10.1016/j.sandf.2014.11.002.

Fang, Q., T. Tsuchiya, and T. Yamamoto. 2002. "Finite difference, finite element and finite volume methods applied to two-point boundary value problems." *J. Comput. Appl. Math.* 139 (1): 9–19. https://doi.org/10.1016/S0377-0427(01)00392-2.

Fatahi, B., and S. H. R. Tabatabaiefar. 2014. "Fully nonlinear versus equivalent linear computation method for seismic analysis of midrise buildings on soft soils." *Int. J. Geomech.* 14 (4): 04014016. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000354.

Fauriel, S., and L. Laloui. 2012. "A bio-chemo-hydro-mechanical model for microbially induced calcite precipitation in soils." *Comput. Geotech.* 46: 104–120. https://doi.org/10.1016/j.compgeo.2012.05.017.
Franza, A., and B. Sheil. 2021. "Pile groups under vertical and inclined eccentric loads: Elastoplastic modelling for performance based design." *Comput. Geotech.* 135: 104092. https://doi.org/10.1016/j.compgeo.2021.104092.
Fryer, Y. D., C. Bailey, M. Cross, and C.-H. Lai. 1991. "A control volume procedure for solving the elastic stress-strain equations on an unstructured mesh." *Appl. Math. Modell.* 15 (11–12): 639–645. https://doi.org/10.1016/S0307-904X(09)81010-X.
Gao, X., W.-p. Tian, J. Li, H. Qi, Z. Zhang, and S. Li. 2022. "Force and deformation response analysis of dual structure slope excavation and support." *Geomatics Nat. Hazards Risk* 13 (1): 501–537. https://doi.org/10.1080/19475705.2022.2037738.
Golub, G. H., and C. F. Van Loan. 2013. *Matrix computations*. Baltimore: JHU Press.
Greenshields, C., H. Weller, and A. Ivankovic. 1999. "The finite volume method for coupled fluid flow and stress analysis." *Comp. Model. Simul. Eng.* 4 (3): 213–218.
Greenshields, C. J. 2015. *OpenFOAM programmer's guide*. London, UK: OpenFOAM Foundation.
Heidarzadeh, H. 2021. "Evaluation of modified cam-clay constitutive model in flac and its development by fish programming." *Eur. J. Environ. Civ. Eng.* 25 (2): 226–244. https://doi.org/10.1080/19648189.2018.1521752.
Itasca. 2009. *Fast Lagrangian analysis of continua in 3 dimensions, user's guide*. Minneapolis: Itasca.
Katz, V. J. 1979. "The history of Stokes' theorem." *Math. Mag.* 52 (3): 146–156. https://doi.org/10.2307/2690275.
Lai, Y. M., Z. W. Wu, Y. L. Zhu, and L. N. Zhu. 1998. "Nonlinear analysis for the coupled problem of temperature, seepage and stress fields in cold-region tunnels." *Tunnelling Underground Space Technol.* 13 (4): 435–440. https://doi.org/10.1016/S0886-7798(98)00086-8.
LeVeque, R. J. 2002. *Finite volume methods for hyperbolic problems*. Cambridge, UK: Cambridge University Press.
Maneeratana, K. 2000. "Development of the finite volume method for non-linear structural applications." Ph.D. thesis, Dept. of Civil and Environmental Engineering, Imperial College London.
Mao, Z., X. Ma, Y. Liu, M. Geng, Y. Tian, J. Sun, and Z. Yang. 2024. "Study on time effect and prediction model of shear strength of root-soil complex under dry-wet cycle." *Biogeotechnics* 2: 100079. https://doi.org/10.1016/j.bgtech.2024.100079.
Martinez, B. C., J. T. DeJong, and T. R. Ginn. 2014. "Bio-geochemical reactive transport modeling of microbial induced calcite precipitation to predict the treatment of sand in one-dimensional flow." *Comput. Geotech.* 58: 1–13. https://doi.org/10.1016/j.compgeo.2014.01.013.
Moraes, A., P. Lage, G. Cunha, and L. da Silva. 2013. "Analysis of the non-orthogonality correction of finite volume discretization on unstructured meshes." In *Proc., 22nd Int. Congress of Mechanical Engineering*. Ribeirao Preto, Brasil: COBEM.
Perić, D. 2006. "Analytical solutions for a three-invariant Cam clay model subjected to drained loading histories." *Int. J. Numer. Anal. Methods Geomech.* 30 (5): 363–387. https://doi.org/10.1002/nag.482.
Perre, P., and J. Passard. 1995. "A control-volume procedure compared with the finite-element method for calculating stress and strain during wood drying." *Drying Technol.* 13 (3): 635–660. https://doi.org/10.1080/07373939508916978.
Roscoe, K., and J. Burland. 1968. "On the generalized stress-strain behaviour of wet clay." In *Proc., Engineering Plasticity Conf.*, 535–609. Cambridge, UK: Cambridge University Press.
Roscoe, K. H., A. N. Schofield, and C. P. Wroth. 1958. "On the yielding of soils." *Géotechnique* 8 (1): 22–53. https://doi.org/10.1680/geot.1958.8.1.22.
Rutqvist, J. 2011. "Status of the tough-flac simulator and recent applications related to coupled fluid flow and crustal deformations." *Comput. Geosci.* 37 (6): 739–750. https://doi.org/10.1016/j.cageo.2010.08.006.
Rutqvist, J., Y. Ijiri, and H. Yamamoto. 2011. "Implementation of the Barcelona basic model into tough–flac for simulations of the geomechanical behavior of unsaturated soils." *Comput. Geosci.* 37 (6): 751–762. https://doi.org/10.1016/j.cageo.2010.10.011.
Sevim, B. 2011. "Nonlinear earthquake behaviour of highway tunnels." *Nat. Hazards Earth Syst. Sci.* 11 (10): 2755–2763. https://doi.org/10.5194/nhess-11-2755-2011.
Silvestri, V., and G. Abou-Samra. 2011. "Application of the exact constitutive relationship of modified cam clay to the undrained expansion of a spherical cavity." *Int. J. Numer. Anal. Methods Geomech.* 35 (1): 53–66. https://doi.org/10.1002/nag.892.
Taiebat, M., A. M. Kaynia, and Y. F. Dafalias. 2011. "Application of an anisotropic constitutive model for structured clay to seismic slope stability." *J. Geotech. Geoenviron. Eng.* 137 (5): 492–504. https://doi.org/10.1061/(ASCE)GT.1943-5606.0000458.
Tang, T., O. Hededal, and P. Cardiff. 2015. "On finite volume method implementation of poro-elasto-plasticity soil model." *Int. J. Numer. Anal. Methods Geomech.* 39 (13): 1410–1430. https://doi.org/10.1002/nag.2361.
To, P., D. Agius, and L. Cussen. 2020. "Influence of relative density of the granular base soil on filter performance." *Acta Geotech.* 15 (12): 3621–3627. https://doi.org/10.1007/s11440-020-01064-x.
Tuković, Ž., A. Karač, P. Cardiff, H. Jasak, and A. Ivanković. 2018. "OpenFOAM finite volume solver for fluid-solid interaction." *Trans. FAMENA* 42 (3): 1–31. https://doi.org/10.21278/TOF.42301.
Versteeg, H., and W. Malalasekera. 2007. *An introduction to computational fluid dynamics: The finite volume method*. Upper Saddle River, NJ: Pearson Prentice Hall.
Wang, H., L. Li, J. Li, and D. Sun. 2022. "A simple stress correction method for explicit integration algorithm of elastoplastic constitutive models and its application to advanced anisotropic S-CLAY1 model." *Comput. Geotech.* 148: 104817. https://doi.org/10.1016/j.compgeo.2022.104817.
Wang, K., S. Wu, and J. Chu. 2023a. "Mitigation of soil liquefaction using microbial technology: An overview." *Biogeotechnics* 1: 100005. https://doi.org/10.1016/j.bgtech.2023.100005.
Wang, Y., C. Konstantinou, S. Tang, and H. Chen. 2023b. "Applications of microbial-induced carbonate precipitation: A state-of-the-art review." *Biogeotechnics* 1: 100008. https://doi.org/10.1016/j.bgtech.2023.100008.
Weller, H. G., G. Tabor, H. Jasak, and C. Fureby. 1998. "A tensorial approach to computational continuum mechanics using object-oriented techniques." *Comput. Phys.* 12 (6): 620–631. https://doi.org/10.1063/1.168744.
Xia, G. H., Y. Zhao, J. H. Yeo, and X. Lv. 2007. "A 3D implicit unstructured-grid finite volume method for structural dynamics." *Comput. Mech.* 40 (2): 299–312. https://doi.org/10.1007/s00466-006-0100-7.
Yang, C., F. Zhu, and J. Zhao. 2024. "Coupled total-and semi-Lagrangian peridynamics for modelling fluid-driven fracturing in solids." *Comput. Methods Appl. Mech. Eng.* 419: 116580. https://doi.org/10.1016/j.cma.2023.116580.
Yazdani, F., H. Sadeghi, P. AliPanahi, M. Gholami, and A. K. Leung. 2024. "Evaluation of plant growth and spacing effects on bioengineered slopes subjected to rainfall." *Biogeotechnics* 2: 100080. https://doi.org/10.1016/j.bgtech.2024.100080.