Parresia Research Limited, 41 Hassan Drive, Auckland 0614, New Zealand, www.parresianz.com

Parresia Research Report #PAR-10021867-1516.v1

Theory, verification, and validation of the ARENA constitutive model

for applications to high-rate loading of fully or partially saturated granular media

Author: **Biswajit Banerjee** **(b.banerjee.nz@gmail.com)**
Rebecca Brannon **(rebecca.brannon@utah.edu)**

Prepared for:

Public Dissemination

Parresia Research Limited
Wednesday 20th September, 2017

Contents

Abstract

The ARENA continuum-scale model for sand and/or clay under high-rate loading conditions is presented. Our scope is limited to adiabatic load/unload conditions in order to focus on model features that most crucial for simulations of buried explosives and similar phenomena that involve shock compression followed by free expansion (possibly with re-compression when ejecta impacts an object). Evidence is provided that such conditions fall in a realm for which there is no substantial difference between additive or multiplicative inelasticity approaches. The ARENA model is implemented in a Material Point Method (MPM) code and details of the implementation and algorithms are discussed. The model is verified and validated against split-Hopkinson pressure bar (SHPB) experimental data. Finally, the ARENA model is used to simulate explosions in soil contained in a rotating centrifuge. Potential advantages and shortcomings of the model are discussed and future research directions are identified.

1 Introduction

Numerous phenomenological models have been proposed in the scientific literature for predicting stresses and deformations in partially saturated granular and porous media.[1] These models are typically expressed in terms of either total or "effective" stresses (the stress experienced by the solid skeleton in the absence of pore fluids) (Hassanizadeh and Gray, 1990; Borja, 2004). Most studies deal with quasistatic or long-wavelength conditions for which the choice of stress measure in the momentum equations has been explored extensively (Bishop and Skinner, 1977; Gray and Schrefler, 2001; Borja, 2006; Gray and Schrefler, 2007; Coussy, 2007; Nikooee et al., 2012). It is less clear which stress measure is most appropriate for dynamics, mainly because the definition of total stress is ambiguous in the presence of relative motions of the pore fluids, although (as illustrated in Fig. 1) this issue is less concerning in extremely high-rate applications that suppress relative motion of fluid and solid constituents and therefore eliminate the need to simultaneously solve separate momentum equations for each constituent. Investment in multiscale modeling (*c.f.* (Reguerio et al., 2013)) seems necessary to identify appropriate forms for macroscale constitutive relations, regardless of whether for quasistatic or dynamic conditions.

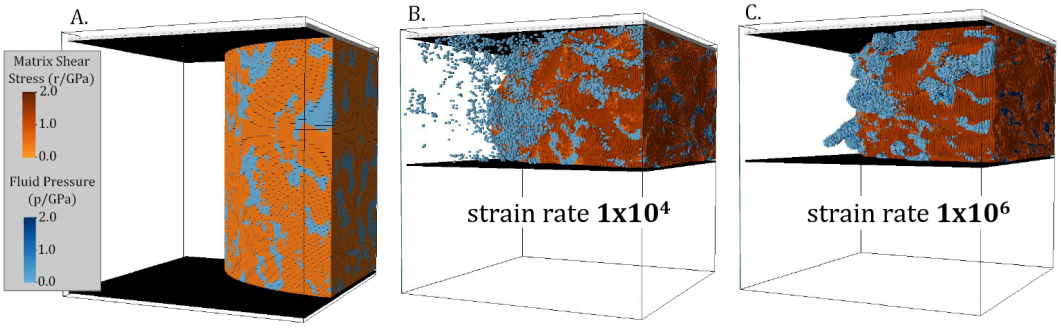


Figure 1 – *Mesoscale models of fluid-solid composites illustrate that fluid motion through open porosity is small at the high loading rates (and short time scales) typical of blast applications, thus justifying an assumption of zero permeability that eliminates the need to simultaneously solve multiple momentum balance equations for each constituent. (Homel, Guilkey, and Brannon, 2017)*

The first few sections of this paper provide equations governing fully or partially saturated sand. The theory is necessarily simplistic (idealized) in order to accommodate practical constraints (such as deployability into a massively parallel host code), so it must be understood that it includes only features that seem to be absolutely crucial for high-rate loading of granular media containing fluids. Model parameterization procedures are discussed, at which time it becomes clear that even feature-rich models fall short of matching experimental data across broad ranges of loading rates, scales, and other

[1] Examples can be found in the following publications and the references cited in them: Reguerio et al., 2013; Bishop and Skinner, 1977; Hassanizadeh and Gray, 1979; Gray and Hassanizadeh, 1989; Alonso, Gens, and Josa, 1990; Hassanizadeh and Gray, 1990; Gray and Hassanizadeh, 1991b; Svendsen and Hutter, 1995; Berg, 1995; Meroi, Schrefler, and Zienkiewicz, 1995; Gawin, Schrefler, and Galindo, 1996; Pietruszczak and Pande, 1996; Houlsby, 1997; Gray and Schrefler, 2001; Schrefler and Scotta, 2001; Rossi, 2002; Brown et al., 2007; Faria Santos, 2003; Georgiadis, 2003; Borja, 2004; Hamiel, Lyakhovsky, and Agnon, 2004; C. Li, Borja, and Regueiro, 2004; Schrefler and Pesavento, 2004; Lade, 2005; Borja, 2006; Santagiuliana and Schrefler, 2006; Coussy, 2007; Gray and Schrefler, 2007; Andrade and Ellison, 2008; Mica Grujicic et al., 2008; Kohler and Hofstetter, 2008; Arson and Gatmiri, 2009; Coussy and Brisard, 2009; Gray, Schrefler, and Pesavento, 2009; Passarotto et al., 2009; Vlahinić, Jennings, and J. J. Thomas, 2009; Borja and White, 2010; Coussy, Pereira, and Vaunat, 2010; Goren et al., 2010; Papastavrou and Steinmann, 2010; Dumont et al., 2011; Gajo, 2011; Higgins and Basu, 2011; Liu and Muraleetharan, 2011; Vlahinić, Jennings, Andrade, et al., 2011; G. Buscarnera and Einav, 2012; Grasley and Rajagopal, 2012; Mašín and Khalili, 2012; Nikooee et al., 2012; Uzuoka and Borja, 2012; Aharonov et al., 2013; Fuentes and Triantafyllidis, 2013; Madeo, Dell'Isola, and Darve, 2013; Nedjar, 2013; Giuseppe Buscarnera, 2014; Lakeland, Rechenmacher, and Ghanem, 2014; Sanavia et al., 2014; Song and Borja, 2014b; Wong and Mašín, 2014; Zhou and Sheng, 2015; Serpieri, Travascio, et al., 2015; Cao, Sanavia, and Schrefler, 2016; Le Pense, Arson, and Pouya, 2016; X. Li et al., 2016; Serpieri and Travascio, 2016.

application-specific conditions. High-rate loading, for example, typically suppresses many low-rate effects (creep, fluid-flow permeability, heat conduction, etc.) because the pertinent rheological phenomena do not have sufficient time to develop. High-rate problems furthermore might exhibit other phenomena (non-constant bulk modulus from concomitant larger deformations, rate-sensitivity of strength, etc.) that are essentially absent in low-rate small-deformation data. Hence, to legitimately use quasistatic data to help parameterize a high-rate model, these low-rate effects in the data must be "stripped out" or otherwise accounted for(e.g., to convert the quasistatic isothermal bulk modulus to the isentropic bulk modulus), and high-rate effects (like viscoplasticity and a nonlinear equation of state) must be added . Further restricting the model's usage to problems having approximately stationary reference stretch directions ensures little or no difference in the predictions of a model that uses a multiplicative deformation decomposition in comparison with one that adopts and additive decompositions of strain rates (Itskov, 2004) or velocity gradients (Rubin and Ichihara, 2010). A multiplicative model cannot be legitimately claimed to be to be superior to an additive one without showing their equivalence in conditions of stationary reference stretch directions and comparative inadequacy of the additive one otherwise. If stretch rotation is negligible in an application of interest, then there is no need to suffer the computational overhead. of a multiplicative model (not to mention their still poorly developed extension to problems involving additional mechanisms of inelasticity, such as damage, irreversible phase transformations, and associated dissipative heating pertinent to granular media). Some applicability limitations of idealized models are discussed. For example, it is argued that an additive decomposition of elastic and inelastic strain rates (or the velocity gradient) is an acceptable alternative to more complicated multiplicative decompositions as long as the application of interest meets a simple (typically satisfied) criterion about the character of material rotation.

Continuum constitutive models for granular media often suffer challenges of a solid, fluid, and gas. In a buried-explosive simulation, for example, the Jacobian might vary non-monotonically by orders of magnitude (from very stiff highly compressed states during the initial blast, to gas-like free expansion during ejecta propagation, finishing with recompression upon impacting an obstacle). Accordingly, several — and many still unsolved — challenges exist for implementing these models into momentum-solver host codes. A Lagrangian finite-element code cannot handle the large distortions, while an Eulerian approach typically corrupts fidelity of the constitutive model's evolving internal state variables. Advantages and shortcomings of some actively researched particle methods are therefore discussed. To alleviate these problems, many simulations in this paper use the open-source computational framework Uintah (Davison de St Germain et al., 2000), which was selected because of its well-supported option to solve the momentum equations via the Material Point Method (Sulsky, Chen, and Schreyer, 1994).

Homogenized models for granular media are of dubious value unless their implementation accounts for the fact that the aggregates (sand grains) are not sufficiently small to reach the continuum limit. As aleatory heterogeneity can't be neglected, any local and deterministic theory (like the one described herein) ought to be invoked within a host code using nonlocality and/or scale-dependent property variability to benefit both predictiveness and convergence (Kamrin and Bazant, 2007; Strack, Leavy, and Brannon, 2014). Aleatory uncertainty in the constitutive model is necessary (though not sufficient) to stimulate realistic localization and other forms of bifurcation that might arise from statistical variability in force chains between grains. Statistical variation also contributes to non-continuum inelastic scale effects. The paper concludes with a section summarizing an empirical scaling law for realistic soils, which has been recently touted as a potential means of predicting full-scale buried-explosive phenomena from observations of less-expensive smaller explosives in the high-gravity environment of a centrifuge (Reguerio et al., 2013).

This paper is organized to proceed through the following topics, some of which are covered only lightly in order to highlight areas for future research:

1. Governing equations for an idealized (and open-source) constitutive model (Arena) for partially or fully saturated soil subjected to high-rate loading including a high-rate Duvaut-Lions

model for viscoplastic effects.

2. Accounting for sub-scale (non-continuum) heterogeneity via scale-dependent aleatory uncertainty in the continuum constitutive model.
3. Large deformation considerations: multiplicative and additive decompositions of deformation and the appropriateness of additive decompositions in Arena .
4. The MPM and stress-update algorithms and numerical implementation challenges, both in advancing the constitutive state through time and in effectively using granular models in particle-based momentum solvers.
5. Model parameterization procedures, with examples of fitted data. These are based on theoretical and empirical scaling laws, which facilitate using inexpensive small-scale tests to predict results of relatively expensive large-scale experiments.
6. Exploration of the theoretical behavior of the Arena model and identification of issues that need further research.
7. Verification of the Arena model using single particle MPM simulations.
8. Validation of the Arena model using SHPB test data outside the range used in model parameterization.
9. MPM simulations of explosions in soils modeled with Arena .

2 The ARENA model

This paper describes a constitutive model called ARENA .[2] By limiting its scope to features that seem to be crucial for applications involving high-rate loading of partially saturated soil/sand, ARENA is reasonably robust in buried-explosive simulations. The Arena model solves a total stress[3] form of the balance of momentum. The constitutive relations include a nonlinear bulk modulus model in conjunction with a nonlinear Drucker-Prager limit surface with a compression cap that is fitted to crush-curve data. Building on work of Homel *et al.* (Homel, Guilkey, and Brannon, 2015) (which itself was specialized from the multipurpose KAYENTA geomechanics model (Brannon, Fuller, et al., 2015)), pore pressure is modeled as an isotropic contribution to backstress with a better-developed theory for partial saturation,[4] Features available in the more complicated KAYENTA model *(not ARENA)* include third-invariant dependence, kinematic hardening, advanced multi-purpose thermodynamic equations of state *etc.* , all of which are well documented elsewhere and could easily be restored into the ARENA model at a considerable cost in code maintenance and robustness for relatively little gain in comparison to the model features retained as essential for soil-blast simulations (where, for example, approximately adiabatic – not isentropic – conditions provide simplifying constraints to a general EOS formulation). ARENA is superior to KAYENTA in its support for fully and partially saturated states as well as its model for adjusting parameters based on initial sample preparation (*e.g.,* compaction through shaking), so these features are a primary focus of this paper .

2.1 Volume fraction, porosity, saturation, density

The ARENA model assumes a three-phase porous medium with constituents $\alpha = \{s, w, a\}$ where s is the solid skeleton, w is water, and a is air. Let $\mathbf{X}$ and $\mathbf{x}$ be the reference and current positions, respectively, of a point inside a body Ω. Each material point is considered to be the center of mass of a representative volume element (RVE) region (ω) of volume dv and mass dm.[5] The region ($\omega^\alpha \subset \omega$) occupied by phase α at each material point has a volume dv^α and mass dm^α such that

$$dv = dv^s + dv^w + dv^a \quad \text{and} \quad dm = dm^s + dm^w + dm^a \,. \tag{1}$$

The *volume fraction* (f^α) of phase α at each point is defined as

$$f^\alpha = \frac{dv^\alpha}{dv} \quad \text{so that} \quad \sum_\alpha f^\alpha = 1 \,. \tag{2}$$

[2] "Arena" is the Spanish word for "sand," referring to the model's origins in Southwestern USA states (Utah and New Mexico). It has been verified and regression tested against a fairly standard battery of analytical single-element tests (Kamojjala et al., 2015) and (as will be described) validated against data for a variety of sands at different moisture and initial states. It is available under open source (Davison de St Germain et al., 2000), allowing it to serve as a starting point to be revised to include any number of other model features (not part of ARENA) that might be needed for a given application.

[3] The phrase "total stress" refers to the continuum-scale stress that is the average of microscale stresses in the heterogeneous mixture of solid, fluid, and air.

[4] Not part of KAYENTA , these governing equations are based on work of Uzuoka and Borja (Uzuoka and Borja, 2012) (see also Borja, (2004), C. Li, Borja, and Regueiro, (2004), Borja, (2006), Borja and White, (2010), Uzuoka and Borja, (2012), Song and Borja, (2014b), and Song and Borja, (2014a)) which is, in turn, a simplification of the averaging theory proposed by Hassanizadeh and Gray (Hassanizadeh and Gray, 1979; Gray and Hassanizadeh, 1989; Hassanizadeh and Gray, 1990; Gray and Hassanizadeh, 1991b; Gray and Hassanizadeh, 1991a). The original equations allow for separate tracking of constituents in a mixture (e.g., solid, water, and air) that might move relative to each other as in a diffusive process, but our Fig. 1 shows that this complication is avoidable for high-rate applications because of "head loss" that prevents appreciable motion of fluid or gas relative to the solid granular constituent over the time range that is typical for shock-loading events.

[5] This classical assumption can significantly increase computational overhead if a distended (*i.e.,* highly expanded) domain impacts an obstacle after a blast event. Simple simulations of a material that has expanded to a a Jacobian exceeding 10 (essentially representing a cloud of "dust") can be readily confirmed to give inaccurate predictions of momentum and energy deposition rates at tractable grid discretizations. This problem is rooted in the fact that the notion of an RVE does not exist on this scale, where part of the material within the supposed RVE has recompacted upon impact while the remainder is still distended. Accounting for this extreme degree of heterogeneity in an RVE remains an as-yet unsolved problem that might well require host-code modifications such as enriched basis functions (Mo es, Dolbow, and Belytschko, 1999).

The volume fraction of the solid skeleton is

$$f^s = 1 - \phi \,, \quad \phi = \frac{dv^w + dv^a}{dv} \tag{3}$$

where the ϕ is the *porosity*. The volume fractions of water and air are

$$f^w = \phi S_w \,, \quad f^a = \phi S_a = \phi(1 - S_w) \,, \quad S_w = \frac{dv^w}{dv^w + dv^a} \tag{4}$$

where S_w is the *saturation*. The Biot parameter is a measure of the influence of porosity on the elastic bulk modulus:

$$B := 1 - \frac{\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)} \,, \tag{5}$$

where $\widetilde{K}^s$ is the bulk modulus of the porous solid and $\widetilde{p}^s$ is the pressure in the porous solid. Physical arguments leading to specific formulas used to compute Arena 's Biot parameter are detailed in this paper 's appendices, with this section listing the major final results.

Conditions of high-rate loading greatly simplify the field equations for mass, momentum, and energy. For example, phases move approximately together despite a microstructure consisting of an open porosity network. These field equations are further reduced with additional simplifying assumptions (such as pressure equilibrium between phases) and adiabatic conditions (justifying a mechanical model without ignoring dissipation or thermodynamic consistency). Of course, the phrase "mechanical model" merely implies that there is a thermodynamic constraint sufficiently restrictive to treat stress as determinable exclusively through knowledge of the deformation history. Even in this context, irreversible processes such as plastic flow and damage (and their associated dissipative heating) are included in principle. The Arena model is a bit more restrictive in that it does not account directly for temperature dependence of material properties (which can certainly be done as described in Kayenta documentation (Brannon, Fuller, et al., 2015)), and consequently Arena is designed for events involving one primary loading interval (*e.g.,* initial shock compression) followed by a secondary unloading to a significantly distended state (free flight of disaggregated particles), and only qualitatively reasonable support for recompaction upon impact with an obstacle.

Let the mixture mass density (ρ) and the mass density of each phase (ρ^α) be defined as

$$\rho = \frac{dm}{dv} \quad \text{and} \quad \rho^\alpha = \frac{dm^\alpha}{dv^\alpha} \,. \tag{6}$$

We define the *partial mass density* of each phase as

$$\langle \rho^\alpha \rangle = f^\alpha \rho^\alpha \quad \text{such that} \quad \langle \rho^s \rangle + \langle \rho^w \rangle + \langle \rho^a \rangle = \rho \,. \tag{7}$$

2.2 Material derivatives

The material time derivative of $\mathbf{f}^\alpha$ with respect to phase α is defined as

$$\frac{D^\alpha \mathbf{f}^\alpha}{Dt} = \frac{\partial \mathbf{f}^\alpha}{\partial t} + \nabla \mathbf{f}^\alpha \cdot \mathbf{v}^\alpha \tag{8}$$

where $\mathbf{v}^\alpha$ is the velocity of phase α. The relative material derivative of phase β with respect to phase α is

$$\frac{D^\alpha \mathbf{f}^\beta}{Dt} = \frac{D^\beta \mathbf{f}^\beta}{Dt} + \nabla \mathbf{f}^\beta \cdot (\mathbf{v}^\alpha - \mathbf{v}^\beta) = \frac{D^\beta \mathbf{f}^\beta}{Dt} + \nabla \mathbf{f}^\beta \cdot \mathbf{v}^{\alpha\beta} \,. \tag{9}$$

2.3 Mass balance

The phase mass balance equation for the RVE in the absence of mass exchange between the phases is

$$\frac{\partial}{\partial t}\left(\langle\rho^{\alpha}\rangle\right) + \nabla\cdot\left(\langle\rho^{\alpha}\rangle\mathbf{v}^{\alpha}\right) = 0\,. \tag{10}$$

Using the definition of the material time derivative, we can write the mass balance equations as

$$\frac{D^{\alpha}}{Dt}\left(\langle\rho^{\alpha}\rangle\right) + \langle\rho^{\alpha}\rangle\nabla\cdot\mathbf{v}^{\alpha} = 0\,. \tag{11}$$

The above equation can be expressed relative to phase β as

$$\frac{D^{\beta}}{Dt}\left(\langle\rho^{\alpha}\rangle\right) - \nabla\langle\rho^{\alpha}\rangle\cdot\mathbf{v}^{\beta\alpha} + \langle\rho^{\alpha}\rangle\nabla\cdot\mathbf{v}^{\alpha} = 0\,. \tag{12}$$

Similarly, the assumption of no relative motion between phases ultimately implies that no revisions are required in how a host code enforces mass conservation. At the constitutive level, the relative mass fractions remain constant, while the volume fractions vary depending assumptions of how the overall RVE stress is distributed to each phase.

2.3.1 Solid phase

For the solid phase,

$$\frac{D^{s}}{Dt}\left(\langle\rho^{s}\rangle\right) + \langle\rho^{s}\rangle\nabla\cdot\mathbf{v}^{s} = 0\,. \tag{13}$$

In terms of volume fractions, the above balance equation can be written as

$$\frac{D^{s}f^{s}}{Dt} + \frac{f^{s}}{\rho^{s}}\frac{D^{s}\rho^{s}}{Dt} + f^{s}\nabla\cdot\mathbf{v}^{s} = 0\,. \tag{14}$$

Let us assume the existence of a smooth function $\rho^{s} = \rho^{s}(p^{s})$, where p^{s} is the intrinsic pressure in the solid phase (actual force per unit area acting on the solid phase – without considering voids). We also define the the bulk modulus of the solid phase as

$$K_{s}(p^{s}) = -\frac{dp^{s}}{d\varepsilon_{v}^{e}} = -\frac{dp^{s}}{d\rho^{s}}\frac{d\rho^{s}}{d\varepsilon_{v}^{e}} = \rho^{s}\frac{dp^{s}}{d\rho^{s}} \tag{15}$$

where the elastic volumetric strain is defined as

$$\varepsilon_{v}^{e} = \ln\left(\frac{v^{s}}{v_{0}^{s}}\right) = -\ln\left(\frac{\rho^{s}}{\rho_{0}^{s}}\right) = \ln J^{s}\,. \tag{16}$$

Then

$$\frac{D^{s}f^{s}}{Dt} + \frac{f^{s}}{\rho^{s}}\frac{d\rho^{s}}{dp^{s}}\frac{D^{s}p^{s}}{Dt} + f^{s}\nabla\cdot\mathbf{v}^{s} = 0\,. \tag{17}$$

We can use the above definition to get

$$\frac{D^{s}f^{s}}{Dt} + \frac{f^{s}}{K_{s}(p^{s})}\frac{D^{s}p^{s}}{Dt} + f^{s}\nabla\cdot\mathbf{v}^{s} = 0\,. \tag{18}$$

Note that there is some ambiguity here about the conservation of intrinsic mass and the definition of intrinsic volume (v^{s}).

For our purposes, it is more convenient to work with a solid phase pressure that is defined over a RVE containing voids. Borja and coworkers define this pressure as

$$p^{s} = \widetilde{p}^{s}(\rho^{s}, f^{s}) \tag{19}$$

such that

$$\frac{D^s p^s}{Dt} = \frac{\partial \widetilde{p}^s}{\partial \rho^s}\frac{D^s \rho^s}{Dt} + \frac{\partial \widetilde{p}^s}{\partial f^s}\frac{D^s f^s}{Dt}. \tag{20}$$

Now,

$$\frac{D^s \varepsilon_v^e}{Dt} = -\frac{1}{\rho_s}\frac{D^s \rho^s}{Dt} = \frac{D^s}{Dt}(\ln J^s) = \frac{1}{J_s}\frac{D^s J^s}{Dt}. \tag{21}$$

Recall that

$$\frac{D^s J^s}{Dt} = J^s \nabla \cdot \mathbf{v}^s. \tag{22}$$

Therefore,

$$\frac{D^s \rho^s}{Dt} = -\rho^s \nabla \cdot \mathbf{v}^s \tag{23}$$

Substituting into (20) gives

$$\frac{D^s p^s}{Dt} = -\rho^s \frac{\partial \widetilde{p}^s}{\partial \rho^s}\nabla \cdot \mathbf{v}^s + \frac{\partial \widetilde{p}^s}{\partial f^s}\frac{D^s f^s}{Dt}. \tag{24}$$

Therefore,

$$\frac{D^s f^s}{Dt} = \frac{1}{\dfrac{\partial \widetilde{p}^s}{\partial f^s}}\left[\frac{D^s p^s}{Dt} + \rho^s \frac{\partial \widetilde{p}^s}{\partial \rho^s}\nabla \cdot \mathbf{v}^s\right]. \tag{25}$$

Substituting this into the mass balance equation (18) for the solid phase,

$$\frac{D^s p^s}{Dt} + \rho^s \frac{\partial \widetilde{p}^s}{\partial \rho^s}\nabla \cdot \mathbf{v}^s + \frac{f^s}{K_s(p^s)}\frac{\partial \widetilde{p}^s}{\partial f^s}\frac{D^s p^s}{Dt} + f^s \frac{\partial \widetilde{p}^s}{\partial f^s}\nabla \cdot \mathbf{v}^s = 0 \tag{26}$$

or

$$\left(1 + \frac{f^s}{K_s(p^s)}\frac{\partial \widetilde{p}^s}{\partial f^s}\right)\frac{D^s p^s}{Dt} + \left(\rho^s \frac{\partial \widetilde{p}^s}{\partial \rho^s} + f^s \frac{\partial \widetilde{p}^s}{\partial f^s}\right)\nabla \cdot \mathbf{v}^s = 0 \tag{27}$$

or,

$$\frac{D^s p^s}{Dt} = -\left(\frac{\rho^s \dfrac{\partial \widetilde{p}^s}{\partial \rho^s} + f^s \dfrac{\partial \widetilde{p}^s}{\partial f^s}}{1 + \dfrac{f^s}{K_s(p^s)}\dfrac{\partial \widetilde{p}^s}{\partial f^s}}\right)\nabla \cdot \mathbf{v}^s. \tag{28}$$

Let us define the bulk modulus of the solid matrix ($\widetilde{K}^s$) using

$$\frac{D^s \widetilde{p}^s}{Dt} = \frac{\partial \widetilde{p}^s}{\partial \varepsilon_v^e}\frac{D^s \varepsilon_v^e}{Dt} =: -\widetilde{K}^s(\widetilde{p}^s)\nabla \cdot \mathbf{v}^s \tag{29}$$

In terms of the intrinsic pressure in the solid, Borja's definition leads to

$$f^s \frac{D^s p^s}{Dt} = -\widetilde{K}^s(\widetilde{p}^s)\nabla \cdot \mathbf{v}^s. \tag{30}$$

The reason for this definition is not obvious and needs re-examination but leads to the definition of the matrix bulk modulus used by Borja:

$$\widetilde{K}^s := f^s\left(\frac{\rho^s \dfrac{\partial \widetilde{p}^s}{\partial \rho^s} + f^s \dfrac{\partial \widetilde{p}^s}{\partial f^s}}{1 + \dfrac{f^s}{K_s(p^s)}\dfrac{\partial \widetilde{p}^s}{\partial f^s}}\right). \tag{31}$$

If we substitute (30) into (18), we get

$$\frac{D^s f^s}{Dt} - \frac{\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)} \nabla \cdot \mathbf{v}^s + f^s \nabla \cdot \mathbf{v}^s = 0 \,. \tag{32}$$

The Biot parameter is a measure of the influence of porosity on the elastic bulk modulus:

$$B := 1 - \frac{\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)} \,, \tag{33}$$

where $\widetilde{K}^s$ is the bulk modulus of the porous solid and $\widetilde{p}^s$ is the pressure in the porous solid. Then, it can be argued that the solid mass balance equation can be written as

$$\frac{D^s f^s}{Dt} + (B - 1 + f^s) \nabla \cdot \mathbf{v}^s = 0 \,. \tag{34}$$

2.3.2 Water phase

The balance of mass for the water phase, relative to the solid phase, is

$$\frac{D^s}{Dt}(\langle \rho^w \rangle) - \nabla \langle \rho^w \rangle \cdot \mathbf{v}^{sw} + \langle \rho^w \rangle \nabla \cdot \mathbf{v}^w = 0 \tag{35}$$

where $\mathbf{v}^{sw} = \mathbf{v}^s - \mathbf{v}^w$. Expanded,

$$\frac{D^s f^w}{Dt} + \frac{f^w}{\rho^w} \frac{D^s \rho^w}{Dt} - \left[\frac{f^w}{\rho^w} \nabla \rho^w + \nabla f^w \right] \cdot \mathbf{v}^{sw} + f^w \nabla \cdot \mathbf{v}^w = 0 \,. \tag{36}$$

Once again we assume the existence of a smooth function $\rho^w = \rho^w(p^w)$ where p^w is the intrinsic water pressure and get

$$\frac{D^s f^w}{Dt} + \frac{f^w}{\rho^w} \frac{d\rho^w}{dp^w} \frac{D^s p^w}{Dt} - \left[\frac{f^w}{\rho^w} \frac{d\rho^w}{dp^w} \nabla p^w + \nabla f^w \right] \cdot \mathbf{v}^{sw} + f^w \nabla \cdot \mathbf{v}^w = 0 \,. \tag{37}$$

As was done for the solid phase, the bulk modulus of water is defined as

$$K_w(p^w) = \rho^w \frac{dp^w}{d\rho^w} \tag{38}$$

which leads to

$$\frac{D^s f^w}{Dt} + \frac{f^w}{K_w(p^w)} \frac{D^s p^w}{Dt} - \left[\frac{f^w}{K_w(p^w)} \nabla p^w + \nabla f^w \right] \cdot \mathbf{v}^{sw} + f^w \nabla \cdot \mathbf{v}^w = 0 \,. \tag{39}$$

Recalling from the definitions of porosity and saturation that

$$f^w = \phi S_w = (1 - f^s) S_w \tag{40}$$

and using the mass balance equation for the solid phase (18), we have

$$\frac{D^s f^w}{Dt} = -S_w \frac{D^s f^s}{Dt} + (1 - f^s) \frac{D^s S_w}{Dt} = S_w \left[\frac{f^s}{K_s(p^s)} \frac{D^s p^s}{Dt} + f^s \nabla \cdot \mathbf{v}^s \right] + (1 - f^s) \frac{D^s S_w}{Dt} \,. \tag{41}$$

Therefore the mass balance of water can be expressed as

$$(1 - f^s) \frac{D^s S_w}{Dt} + \frac{f^w}{K_w(p^w)} \frac{D^s p^w}{Dt} + \frac{S_w f^s}{K_s(p^s)} \frac{D^s p^s}{Dt} + S_w f^s \nabla \cdot \mathbf{v}^s - \left[\frac{f^w}{K_w(p^w)} \nabla p^w + \nabla f^w \right] \cdot \mathbf{v}^{sw} + f^w \nabla \cdot \mathbf{v}^w = 0 \,. \tag{42}$$

Recall from equation (30) that

$$f^s \frac{D^s p^s}{Dt} = -\widetilde{K}^s(\widetilde{p}^s)\nabla\cdot\mathbf{v}^s\,. \tag{43}$$

Substituting this into the mass balance equation for water gives

$$(1-f^s)\frac{D^s S_w}{Dt}+\frac{f^w}{K_w(p^w)}\frac{D^s p^w}{Dt}-\frac{S_w\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)}\nabla\cdot\mathbf{v}^s+S_w f^s\nabla\cdot\mathbf{v}^s-\left[\frac{f^w}{K_w(p^w)}\nabla p^w+\nabla f^w\right]\cdot\mathbf{v}^{sw}+f^w\nabla\cdot\mathbf{v}^w=0\,. \tag{44}$$

Using the definition

$$B := 1-\frac{\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)} \tag{45}$$

we have

$$(1-f^s)\frac{D^s S_w}{Dt}+\frac{f^w}{K_w(p^w)}\frac{D^s p^w}{Dt}+S_w(B-1+f^s)\nabla\cdot\mathbf{v}^s-\left[\frac{f^w}{K_w(p^w)}\nabla p^w+\nabla f^w\right]\cdot\mathbf{v}^{sw}+f^w\nabla\cdot\mathbf{v}^w=0\,. \tag{46}$$

Using

$$\nabla\cdot\mathbf{v}^w=-\nabla\cdot\mathbf{v}^{sw}+\nabla\cdot\mathbf{v}^s \tag{47}$$

and

$$S_w(B-1+f^s)+f^w=S_wB-S_w+S_w(1-\phi)+S_w\phi=S_wB \tag{48}$$

Using arguments similar to those used for the solid phase, equation (35) can be expressed as

$$\begin{aligned}(1-f^s)\frac{D^s S_w}{Dt}+\frac{f^w}{K_w(p^w)}\frac{D^s p^w}{Dt}+BS_w\nabla\cdot\mathbf{v}^s\\ -\left[\frac{f^w}{K_w(p^w)}\nabla p^w+\nabla f^w\right]\cdot\mathbf{v}^{sw}-f^w\nabla\cdot\mathbf{v}^{sw}=0\end{aligned} \tag{49}$$

where p^w is the intrinsic pressure in the water.

2.3.3 Air phase

For the air phase, relative to the solid phase, the mass balance equation is

$$\frac{D^s}{Dt}(\langle\rho^a\rangle)-\nabla\langle\rho^a\rangle\cdot\mathbf{v}^{sa}+\langle\rho^a\rangle\nabla\cdot\mathbf{v}^a=0 \tag{50}$$

where $\mathbf{v}^{sa}=\mathbf{v}^s-\mathbf{v}^a$.

In terms of volume fractions,

$$\frac{D^s f^a}{Dt}+\frac{f^a}{\rho^a}\frac{D^s\rho^a}{Dt}-\left[\frac{f^a}{\rho^a}\nabla\rho^a+\nabla f^a\right]\cdot\mathbf{v}^{sa}+f^a\nabla\cdot\mathbf{v}^a=0\,. \tag{51}$$

For the air phase, with

$$\rho^a=\rho^a(p^a)\quad\text{and}\quad K_a(p^a)=\rho^a\frac{dp^a}{d\rho^a} \tag{52}$$

we get

$$\frac{D^s f^a}{Dt}+\frac{f^a}{K_a(p^a)}\frac{D^s p^a}{Dt}-\left[\frac{f^a}{K_a(p^a)}\nabla p^a+\nabla f^a\right]\cdot\mathbf{v}^{sa}+f^a\nabla\cdot\mathbf{v}^a=0\,. \tag{53}$$

From the definitions of porosity and saturation,

$$f^a=\phi S_a=(1-f^s)(1-S_w)\,. \tag{54}$$

Using the procedure used for the water phase, the mass balance of air is

$$(1-f^s)\frac{D^sS_a}{Dt}+\frac{f^a}{K_a(p^a)}\frac{D^sp^a}{Dt}+\frac{S_af^s}{K_s(p^s)}\frac{D^sp^s}{Dt}+S_af^s\nabla\cdot\mathbf{v}^s-\left[\frac{f^a}{K_a(p^a)}\nabla p^a+\nabla f^a\right]\cdot\mathbf{v}^{sa}+f^a\nabla\cdot\mathbf{v}^a=0\,. \quad (55)$$

Expressed in terms of the bulk modulus, volume fraction, and intrinsic air pressure (p^a), equation (50) becomes

$$\begin{aligned}(1-f^s)\frac{D^sS_a}{Dt}+\frac{f^a}{K_a(p^a)}\frac{D^sp^a}{Dt}+BS_a\nabla\cdot\mathbf{v}^s\\ -\left[\frac{f^a}{K_a(p^a)}\nabla p^a+\nabla f^a\right]\cdot\mathbf{v}^{sa}-f^a\nabla\cdot\mathbf{v}^{sa}=0\,.\end{aligned} \quad (56)$$

2.4 Linear momentum balance

The linear momentum balance for the three phases in the absence of mass exchange between phases can be written as

$$\langle\rho^\alpha\rangle\frac{D^\alpha\mathbf{v}^\alpha}{Dt}=\nabla\cdot\boldsymbol{\sigma}^\alpha+\langle\rho^\alpha\rangle\mathbf{b}^\alpha+\mathbf{p}^\alpha+\mathbf{t}^\alpha\,,\qquad\sum_\alpha(\mathbf{p}^\alpha+\mathbf{t}^\alpha)=\mathbf{0} \quad (57)$$

where $\boldsymbol{\sigma}^\alpha$ is the average stress of phase α in the RVE, $\mathbf{b}^\alpha$ is the body force experienced by phase α, and $\mathbf{p}^\alpha$ is the interaction force term due to the motion of the phases and $\mathbf{t}^\alpha$ is the interaction force vector due to surface tractions, between phase α and the other phases. Also, the total stress is given by

$$\boldsymbol{\sigma}=\sum_\alpha\boldsymbol{\sigma}^\alpha\,. \quad (58)$$

Borja uses expressions for $\mathbf{p}^\alpha$ and $\mathbf{t}^\alpha$ similar (but not identical) to that derived by Haasanizadeh and Gray (Hassanizadeh and Gray, 1990) by the application of an averaged energy balance and a reduced entropy inequality to the RVE balance equations discussed in the previous appendix. These expressions are

$$\mathbf{t}^\alpha=p^\alpha\nabla f^\alpha\,,\quad\mathbf{p}^\alpha=-f^\alpha\boldsymbol{\mu}^\alpha\cdot(\mathbf{v}^\alpha-\mathbf{v}^s) \quad (59)$$

where $\boldsymbol{\mu}^\alpha$ is a material parameter related to the relative permeability of the phases.

2.4.1 Solid phase

The material time derivative of the velocity of the solid phase is

$$\mathbf{a}^s=\frac{D^s\mathbf{v}^s}{Dt}=\frac{\partial\mathbf{v}^s}{\partial t}+\nabla\mathbf{v}^s\cdot\mathbf{v}^s\,. \quad (60)$$

The momentum equation for the solid phase is

$$\langle\rho^s\rangle\mathbf{a}^s=\nabla\cdot\boldsymbol{\sigma}^s+\langle\rho^s\rangle\mathbf{b}^s+p^s\nabla f^s\,. \quad (61)$$

2.4.2 Water phase

The material time derivative of the velocity for the water phase is

$$\mathbf{a}^w=\frac{D^w\mathbf{v}^w}{Dt}=\frac{\partial\mathbf{v}^w}{\partial t}+\nabla\mathbf{v}^w\cdot\mathbf{v}^w\,. \quad (62)$$

If the solid phase acts as a reference, we have

$$\mathbf{a}^w=\frac{D^s\mathbf{v}^w}{Dt}-\nabla\mathbf{v}^w\cdot(\mathbf{v}^s-\mathbf{v}^w)\,. \quad (63)$$

Therefore, the momentum equation for the water phase is

$$\langle \rho^w \rangle \mathbf{a}^w = \nabla \cdot \boldsymbol{\sigma}^w + \langle \rho^w \rangle \mathbf{b}^w + p^w \nabla f^w - f^w \boldsymbol{\mu}^w \cdot (\mathbf{v}^w - \mathbf{v}^s) \,. \tag{64}$$

2.4.3 Air phase

The material time derivative for the air phase is

$$\mathbf{a}^a = \frac{D^w \mathbf{v}^a}{Dt} = \frac{\partial \mathbf{v}^a}{\partial t} + \nabla \mathbf{v}^a \cdot \mathbf{v}^a \,. \tag{65}$$

If the solid phase acts as a reference, we have

$$\mathbf{a}^a = \frac{D^s \mathbf{v}^a}{Dt} - \nabla \mathbf{v}^a \cdot (\mathbf{v}^s - \mathbf{v}^a) \tag{66}$$

and the momentum equation for the air phase is

$$\langle \rho^a \rangle \mathbf{a}^a = \nabla \cdot \boldsymbol{\sigma}^a + \langle \rho^a \rangle \mathbf{b}^a + \mathbf{t}^a + p^a \nabla f^a - f^a \boldsymbol{\mu}^a \cdot (\mathbf{v}^a - \mathbf{v}^s) \,. \tag{67}$$

2.4.4 Mixture

Adding the momentum equations of the three phases, and assuming that all three phases experience identical body forces, we have the momentum balance of the mixture

$$\langle \rho^s \rangle \mathbf{a}^s + \langle \rho^w \rangle \mathbf{a}^w + \langle \rho^a \rangle \mathbf{a}^a = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}^s \tag{68}$$

where

$$\rho = \sum_\alpha \langle \rho^\alpha \rangle \,. \tag{69}$$

2.5 Energy balance

The energy balance of the mixture is

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma}^s : \boldsymbol{d}^s - \boldsymbol{\sigma}^w : \boldsymbol{d}^w - \boldsymbol{\sigma}^a : \boldsymbol{d}^a + \nabla \cdot \mathbf{q} - \rho h = 0 \,. \tag{70}$$

We assume that the fluid stresses are isotropic, *i.e.*,

$$\boldsymbol{\sigma}^w = -\langle p^w \rangle \boldsymbol{I} = -f^w p^w \boldsymbol{I} \,, \quad \boldsymbol{\sigma}^a = -\langle p^a \rangle \boldsymbol{I} = -f^a p^a \boldsymbol{I} \,. \tag{71}$$

Then, we have

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma}^s : \boldsymbol{d}^s + f^w p^w \nabla \cdot \mathbf{v}^w + f^a p^a \nabla \cdot \mathbf{v}^a + \nabla \cdot \mathbf{q} - \rho h = 0 \,. \tag{72}$$

Recall that the total stress is

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^s + \boldsymbol{\sigma}^w + \boldsymbol{\sigma}^a = \boldsymbol{\sigma}^s - f^w p^w \boldsymbol{I} - f^a p^a \boldsymbol{I} \,. \tag{73}$$

Therefore we can write the mixture balance equation as

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma} : \boldsymbol{d}^s - f^w p^w \nabla \cdot \mathbf{v}^{sw} - f^a p^a \nabla \cdot \mathbf{v}^{sa} + \nabla \cdot \mathbf{q} - \rho h = 0 \,. \tag{74}$$

After multiplying by p^w and p^a, respectively, we can write (49) and (56) as

$$p^w(1-f^s)\frac{D^sS_w}{Dt}+\frac{p^wf^w}{K_w(p^w)}\frac{D^sp^w}{Dt}+Bp^wS_w\nabla\cdot\mathbf{v}^s- \frac{p^wf^w}{K_w(p^w)}\nabla p^w\cdot\mathbf{v}^{sw}+p^w\nabla f^w\cdot\mathbf{v}^{sw}=f^wp^w\nabla\cdot\mathbf{v}^{sw} \tag{75}$$

and

$$-p^a(1-f^s)\frac{D^sS_w}{Dt}+\frac{p^af^a}{K_a(p^a)}\frac{D^sp^a}{Dt}+Bp^aS_a\nabla\cdot\mathbf{v}^s- \frac{p^af^a}{K_a(p^a)}\nabla p^a\cdot\mathbf{v}^{sa}+p^a\nabla f^a\cdot\mathbf{v}^{sa}=f^ap^a\nabla\cdot\mathbf{v}^{sa}\,. \tag{76}$$

Adding these equations and substitution into the mixture energy balance equation, and noting that

$$\nabla\cdot\mathbf{v}^s=\boldsymbol{I}:\boldsymbol{d}^s \tag{77}$$

gives

$$\rho\frac{D^se}{Dt}-\boldsymbol{\sigma}:\boldsymbol{d}^s-B\left[\sum_{\alpha=\{w,a\}}p^\alpha S_\alpha\right]\boldsymbol{I}:\boldsymbol{d}^s-(p^w-p^a)(1-f^s)\frac{D^sS_w}{Dt}-\sum_{\alpha=\{w,a\}}\frac{p^\alpha f^\alpha}{K^\alpha(p^\alpha)}\frac{D^sp^\alpha}{Dt} +\sum_{\alpha=\{w,a\}}\left[\frac{p^\alpha f^\alpha}{K^\alpha(p^\alpha)}\nabla p^\alpha\cdot\mathbf{v}^{s\alpha}-p^\alpha\nabla f^\alpha\cdot\mathbf{v}^{s\alpha}\right]+\nabla\cdot\mathbf{q}-\rho h=0\,. \tag{78}$$

We can define the effective stress to be

$$\boldsymbol{\sigma}_{\text{eff}}:=\boldsymbol{\sigma}+B\left[\sum_{\alpha=\{w,a\}}p^\alpha S_\alpha\right]\boldsymbol{I} \tag{79}$$

to get the energy balance equation

$$\rho\frac{D^se}{Dt}-\boldsymbol{\sigma}_{\text{eff}}:\boldsymbol{d}^s-(p^w-p^a)(1-f^s)\frac{D^sS_w}{Dt}-\sum_{\alpha=\{w,a\}}\frac{p^\alpha f^\alpha}{K^\alpha(p^\alpha)}\frac{D^sp^\alpha}{Dt} +\sum_{\alpha=\{w,a\}}\left[\frac{p^\alpha f^\alpha}{K^\alpha(p^\alpha)}\nabla p^\alpha-p^\alpha\nabla f^\alpha\right]\cdot\mathbf{v}^{s\alpha}+\nabla\cdot\mathbf{q}-\rho h=0\,. \tag{80}$$

Thus we see that the effective stress is energy conjugate to the rate of deformation of the solid. We also find that the capillary stress is energy conjugate to the saturation.

2.6 High strain-rate balance equations

Assume that the water and the air phases do not move relative to the solid phase, i.e.,

$$\mathbf{v}^s=\mathbf{v}^w=\mathbf{v}^a\quad\text{and}\quad\mathbf{a}^s=\mathbf{a}^w=\mathbf{a}^a\,. \tag{81}$$

Then the mass balance equations of the three phases reduce to

$$\begin{aligned}&\frac{D^sf^s}{Dt}+(B-1+f^s)\nabla\cdot\mathbf{v}^s=0\\&(1-f^s)\frac{D^sS_w}{Dt}+\frac{f^w}{K_w(p^w)}\frac{D^sp^w}{Dt}+BS_w\nabla\cdot\mathbf{v}^s=0\\&-(1-f^s)\frac{D^sS_w}{Dt}+\frac{f^a}{K_a(p^a)}\frac{D^sp^a}{Dt}+B(1-S_w)\nabla\cdot\mathbf{v}^s=0\,.\end{aligned} \tag{82}$$

The momentum balance equation for the mixture is

$$\rho \mathbf{a}^s = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}^s \quad \text{where} \quad \rho = \sum_\alpha \langle \rho^\alpha \rangle \,. \tag{83}$$

The energy balance equation is

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma}_{\text{eff}} : \boldsymbol{d}^s - (p^w - p^a)(1 - f^s)\frac{D^s S_w}{Dt} - \sum_{\alpha=\{w,a\}} \frac{p^\alpha f^\alpha}{K^\alpha(p^\alpha)} \frac{D^s p^\alpha}{Dt} + \nabla \cdot \mathbf{q} - \rho h = 0 \tag{84}$$

where

$$\boldsymbol{\sigma}_{\text{eff}} = \boldsymbol{\sigma} + B[p^w S_w + p^a S_a]\boldsymbol{I} = \boldsymbol{\sigma} + B[(p^w - p^a)S_w + p^a]\boldsymbol{I} \tag{85}$$

and

$$B := 1 - \frac{\widetilde{K}^s(\widetilde{p}^s)}{K_s(p^s)} \,. \tag{86}$$

If we make the further assumption that at high strain-rates the intrinsic phase pressures are equal, $p^s = p^w = p^a$, the momentum balance equation for the mixture is

$$\rho \mathbf{a}^s = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}^s \quad \text{where} \quad \rho = \sum_\alpha \rho^\alpha \tag{87}$$

where ρ is the mass density of the mixture, $\mathbf{a}^s$ is the acceleration of the solid skeleton, $\boldsymbol{\sigma}$ is the total Cauchy stress in the mixture, $\mathbf{b}^s$ is the body force per mixture mass, and $\alpha = \{s, w, a\}$ are the three phases (solid, water, air). The effective stress is

$$\boldsymbol{\sigma}_{\text{eff}} = \boldsymbol{\sigma} + B\overline{p^w}\boldsymbol{I} = \boldsymbol{\sigma} - \boldsymbol{\alpha} \tag{88}$$

where $\overline{p^w} = -p^w$ is the pore pressure, and B is the Biot coefficient which is defined as

$$B := 1 - \frac{K_d(\overline{p^s})}{K_s(\overline{p^s})} \,. \tag{89}$$

Here p^s is the intrinsic pressure in the solid skeleton, K_d is the bulk modulus of the drained solid skeleton, K_s is the bulk modulus of the solid grains, and we assume that $\overline{p^s} = \overline{p^w}$. All unbarred quantities are positive in tension. For example, $\boldsymbol{\sigma}$ and p^w are positive in tension while $\overline{p^w}$ is positive in compression. Note, incidentally that the term $\boldsymbol{\alpha} := -B\overline{p^w}\boldsymbol{I}$ can be regarded as an *isotropic* backstress, requiring an evolution model, as with a conventional backstress.

We the assumption of phase pressure equilibrium, the mass balance equation equations can be written as (with the substitution $f^s = 1 - \phi$, where ϕ is the porosity)

$$\begin{aligned} &\frac{D^s \phi}{Dt} - (B - \phi)\nabla \cdot \mathbf{v}^s = 0 \\ &\left[\frac{S_w \phi}{K_w(p^w)} + \frac{(1 - S_w)\phi}{K_a(p^w)}\right]\frac{D^s p^w}{Dt} + B\nabla \cdot \mathbf{v}^s = 0 \,. \end{aligned} \tag{90}$$

As shown in Appendix A, the mass balance equations can be expressed as a rate equation for the pore pressure:

$$\frac{d\overline{p^w}}{d\overline{\varepsilon_v}} = \frac{1}{\mathcal{B}} \tag{91}$$

where $\overline{\varepsilon_v}$ is the total volumetric strain, and

$$\begin{aligned} \mathcal{B} := \frac{1}{(1 - S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + S_0 \exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})} \Bigg[&-\frac{(B - \phi)\phi}{B\phi_0}\left(\frac{S_w}{K_w} + \frac{1 - S_w}{K_a}\right) + \\ &\frac{1 - S_0}{K_a}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + \frac{S_0}{K_w}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\Bigg] \,. \end{aligned} \tag{92}$$

In the above expression, B is the Biot coefficient, ϕ is the current total porosity, ϕ_0 is the initial total porosity, S_w is the total saturation, S_0 is the initial total saturation, K_w is the pressure-dependent bulk modulus of water, K_a is the pressure-dependent bulk modulus of air, $\overline{\varepsilon_v^a}$ is the total strain in the air phase, and $\overline{\varepsilon_v^w}$ is the total strain in the water phase.

If we use the pressure equilibrium assumption, the energy balance equation becomes

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma}_{\text{eff}} : \boldsymbol{d}^s + \overline{p^w} \left[\frac{S_w \phi}{K_w(\overline{p^w})} - \frac{(1 - S_w)\phi}{K_a(\overline{p^w})} \right] \frac{D^s \overline{p^w}}{Dt} + \nabla \cdot \mathbf{q} - \rho h = 0 \tag{93}$$

where

$$\boldsymbol{\sigma}_{\text{eff}} = \boldsymbol{\sigma} + B p^w \boldsymbol{I} \tag{94}$$

and e is the internal energy density, $\boldsymbol{d}^s$ is the symmetric part of the velocity gradient in the solid skeleton, $\mathbf{q}$ is the heat flux, and h is a heat source. This equation may also be written in the alternative form (Hassanizadeh and Gray, 1990)

$$\rho \frac{D^s e}{Dt} - \boldsymbol{\sigma} : \boldsymbol{d}^s + \nabla \cdot \mathbf{q} - \rho h = 0 \,. \tag{95}$$

We do not solve the energy equation directly in the ARENA model but do consider the energy conjugacy of $\boldsymbol{\sigma}$ and $\boldsymbol{d}^s$.

2.7 Constitutive model

We follow the approach used by Homel, Guilkey, and Brannon, (2015) where the total stress is used in the constitutive model instead of the effective stress. In rate form, the constitutive model is[6]

$$\begin{aligned} \dot{\boldsymbol{\sigma}} &= \dot{\boldsymbol{\sigma}}_{\text{eff}} + \dot{\boldsymbol{\alpha}} = \dot{\boldsymbol{\sigma}}_{\text{eff}} - \dot{B}\overline{p^w}\boldsymbol{I} - B\dot{\overline{p^w}}\boldsymbol{I} \\ &= g_1(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s) - g_2(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s)\overline{p^w} - g_3(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s)B \end{aligned} \tag{96}$$

where g_1, g_2, and g_3 are assumed constitutive relations that can be expressed as

$$\begin{aligned} g_1(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s) &= \dot{\boldsymbol{\sigma}}_{\text{eff}} = \mathbf{C}^{\text{pe}}_{\text{eff}} : \boldsymbol{d}^s \\ g_2(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s) &= \dot{B} = -\frac{1}{K_s}\dot{\tilde{K}}_s + \frac{K_d}{K_s^2}\dot{K}_s = -\frac{1}{K_s \mathcal{B}}\left[\frac{\partial K_d}{\partial \overline{p^w}} - \frac{K_d}{K_s}\frac{\partial K_s}{\partial \overline{p^w}}\right] \operatorname{tr}(\boldsymbol{d}^s) \\ g_3(\boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{d}^s) &= \dot{\overline{p^w}} = \frac{1}{\mathcal{B}} \operatorname{tr}(\boldsymbol{d}^s) \end{aligned} \tag{97}$$

In the above equations, $\mathbf{C}^{\text{pe}}_{\text{eff}}(\boldsymbol{\sigma}, \boldsymbol{\eta})$ is an effective elastic-plastic tangent modulus, $K_d(\boldsymbol{\sigma}, \boldsymbol{\eta})$ and $K_s(\boldsymbol{\sigma}, \boldsymbol{\eta})$ are the bulk moduli that contribute to the Biot parameter, and $\mathcal{B}(\boldsymbol{\sigma}, \boldsymbol{\eta})$ is defined in equation (92)[7] .

As discussed in Section 3, we assume that

$$\boldsymbol{d}^s = \boldsymbol{d}^{\text{e}} + \boldsymbol{d}^{\text{p}} = \dot{\boldsymbol{\varepsilon}}^{\text{e}} + \dot{\boldsymbol{\varepsilon}}^{\text{p}} \tag{98}$$

[6] This model is applied in the "unrotated" configuration to satisfy the principle of material frame indifference. The applications of interest are assumed to have negligible rotation of *reference* stretch directions, thus making the symmetric part of the velocity gradient, $\boldsymbol{d}$, a very good approximation to the rate of Hencky strain and hence (in this approximation) conjugate to Cauchy stress $\boldsymbol{\sigma}$. More general models based on a multiplicative decomposition of the deformation should (at least for an initially isotropic medium) reduce to this form in the case of non-rotating reference stretch directions, and therefore multiplicative models are worthwhile only if there exists sufficient validation data to distinguish them from this simpler and more robust subclass.

[7] Equation (96) suggests that the bulk modulus may also have to be treated as an internal variable if the backstress is treated as an internal variable.

where $\boldsymbol{\varepsilon}^{\mathrm{e}}$ is an elastic strain and $\boldsymbol{\varepsilon}^{\mathrm{p}}$ is a plastic strain.[8] In addition to plastic strain, ARENA tracks the following internal variables ($\boldsymbol{\eta}$)for the partially saturated soil model :

$$\boldsymbol{\eta} = \{\overline{p_{\mathrm{e}}^{w}}, \overline{p_{\mathrm{p}}^{w}}, B^{\mathrm{e}}, B^{\mathrm{p}}, \phi^{\mathrm{e}}, \phi^{\mathrm{p}}, S_{w}^{\mathrm{e}}, S_{w}^{\mathrm{p}}, \overline{X}^{\mathrm{e}}, \overline{X}^{\mathrm{p}}\} \tag{99}$$

where $\overline{p_{\mathrm{e}}^{w}}$ is the part of the pore pressure that goes to zero after a load that leads to plastic deformation is removed, $\overline{p_{\mathrm{p}}^{w}} = \overline{p^{w}} - \overline{p_{\mathrm{e}}^{w}}$ is the unrecoverable pore pressure which remains nonzero after removal of the load, B^{e}, B^{p} are the the Biot parameter values associated with elastic and inelastic deformation, ϕ^{e} is the recoverable porosity when the load is removed, ϕ^{p} is the unrecoverable or "unloaded" porosity, S_{w}^{e} is the recoverable saturation, S_{w}^{p} is the unrecoverable saturation, and $\overline{X}^{\mathrm{e}}$, $\overline{X}^{\mathrm{p}}$ are the hydrostatic compressive strengths associated with elastic and inelastic deformations, respectively.

2.7.1 Elasticity model

The ARENA models allows for the possibility that during elastic deformation the elastic behavior of the soil can be coupled to the plastic behavior. This seems to be an essential feature to include because, for example, crushing out pore space must affect not only strength but also stiffness. Recognizing that stress depends not only on elastic strain but also on evolving internal variables, the rate equation for elastic behavior can be written as (see Brannon, (2007))

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}^{\mathrm{e}} : \boldsymbol{d}^{\mathrm{e}} - \dot{\lambda}\boldsymbol{Z} \tag{100}$$

where the purely elastic stiffness tensor, $\mathbf{C}^{\mathrm{e}}(\boldsymbol{\sigma}, \overline{p_{\mathrm{e}}^{w}}, B^{\mathrm{e}}, \phi^{\mathrm{e}}, S_{w}^{\mathrm{e}}, \overline{X}^{\mathrm{e}})$, is defined as

$$\begin{aligned}\mathbf{C}^{\mathrm{e}} : \boldsymbol{d}^{\mathrm{e}} &= \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} : \boldsymbol{d}^{\mathrm{e}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{p_{\mathrm{e}}^{w}}} \dot{\overline{p_{\mathrm{e}}^{w}}} + \frac{\partial \boldsymbol{\sigma}}{\partial B^{\mathrm{e}}} \dot{B}^{\mathrm{e}} + \frac{\partial \boldsymbol{\sigma}}{\partial \phi^{\mathrm{e}}} \dot{\phi}^{\mathrm{e}} + \frac{\partial \boldsymbol{\sigma}}{\partial S_{w}^{\mathrm{e}}} \dot{S}_{w}^{\mathrm{e}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{X}^{\mathrm{e}}} \dot{\overline{X}}^{\mathrm{e}} \\ &= \left[\frac{\partial \boldsymbol{\sigma}_{\mathrm{eff}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{p_{\mathrm{e}}^{w}}} \otimes \frac{\partial \overline{p_{\mathrm{e}}^{w}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial B^{\mathrm{e}}} \otimes \frac{\partial B^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \phi^{\mathrm{e}}} \otimes \frac{\partial \phi^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial S_{w}^{\mathrm{e}}} \otimes \frac{\partial S_{w}^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{X}^{\mathrm{e}}} \otimes \frac{\partial \overline{X}^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} \right] : \boldsymbol{d}^{\mathrm{e}}\end{aligned} \tag{101}$$

or

$$\mathbf{C}^{\mathrm{e}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{p_{\mathrm{e}}^{w}}} \otimes \frac{\partial \overline{p_{\mathrm{e}}^{w}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial B^{\mathrm{e}}} \otimes \frac{\partial B^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \phi^{\mathrm{e}}} \otimes \frac{\partial \phi^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial S_{w}^{\mathrm{e}}} \otimes \frac{\partial S_{w}^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{X}^{\mathrm{e}}} \otimes \frac{\partial \overline{X}^{\mathrm{e}}}{\partial \boldsymbol{\varepsilon}^{\mathrm{e}}} . \tag{102}$$

where $\boldsymbol{\varepsilon}^{\mathrm{e}}$ is the elastic part of the strain that is energy conjugate to the *unrotated* Cauchy stress (see Norris, (2008) for possible strain measures). In the plastic coupling term in (100), $\dot{\lambda}$ is the plastic flow rate parameter (equal to the magnitude of the plastic strain rate when using normalized yield normal and flow tensors as advocated in Brannon and Leelavanichkul, (2010)), and $\boldsymbol{Z}(\boldsymbol{\sigma}, \overline{p_{\mathrm{p}}^{w}}, B^{\mathrm{p}}, \phi^{\mathrm{p}}, S_{w}^{\mathrm{p}}, \overline{X}^{\mathrm{p}}; \boldsymbol{\varepsilon}^{\mathrm{p}})$ is a rank-2 elastic-plastic coupling tensor given by

$$\boldsymbol{Z} = h_{p}\frac{\partial \boldsymbol{\sigma}}{\partial \overline{p_{\mathrm{p}}^{w}}} + h_{B}\frac{\partial \boldsymbol{\sigma}}{\partial B^{\mathrm{p}}} + h_{\phi}\frac{\partial \boldsymbol{\sigma}}{\partial \phi^{\mathrm{p}}} + h_{S_{w}}\frac{\partial \boldsymbol{\sigma}}{\partial S_{w}^{\mathrm{p}}} + h_{X}\frac{\partial \boldsymbol{\sigma}}{\partial \overline{X}^{\mathrm{p}}} \tag{103}$$

where

$$\dot{\overline{p_{\mathrm{p}}^{w}}} = \dot{\lambda}h_{p} , \quad \dot{B}^{\mathrm{p}} = \dot{\lambda}h_{B} , \quad \dot{\phi}^{\mathrm{p}} = \dot{\lambda}h_{\phi} , \quad \dot{S}_{w}^{\mathrm{p}} = \dot{\lambda}h_{S_{w}} , \quad \dot{\overline{X}}^{\mathrm{p}} = \dot{\lambda}h_{X} . \tag{104}$$

[8] As this theory is described in the context of an additive decomposition of strain rates, it carries with it an implicit potential limitation that reference stretch directions remain approximately stationary or that, by the time such rotations become large, the stress in the material is negligible due to disaggregation. These assumptions are quite reasonable in high-rate buried-explosive applications for which the model was designed. In this context, the unrotated symmetric part of the velocity gradient $\boldsymbol{d}$equals the rate of reference Hencky strain $\dot{\boldsymbol{\varepsilon}}$ and is conjugate to the unrotated Cauchy stress $\boldsymbol{\sigma}$. Because multiplicative decompositions of the deformation gradient should (for initially isotropic media) become additive in these conditions, any claim of their superiority must be backed with (1) demonstrated equivalence to simpler additive models if stretch directions are stationary and (2) compellingly better agreement with validation data when stretch directions rotate.

The "hardening" functions h_p, h_b, h_ϕ, h_{S_w}, and h_X require additional constitutive assumptions. For example, as explained in Brannon, (2007), a theoretical form for h_ϕ can be derived by approximating the solid matrix to be plastically incompressible. As explained in Brannon, Fuller, et al., (2015), the h_X function can be determined directly from the so-called "crush curve" for pressure vs. volumetric strain in hydrostatic loading.[9] During purely elastic loading, the coupling term is zero because $\dot{\lambda}$ is zero.

To simplify our calculations and the parameter calibration process, we assume that

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}^e} = \mathbf{C}^{\mathrm{e}}_{\sigma} = \left(\widehat{K} - \tfrac{2}{3}\widehat{G}\right) \boldsymbol{I} \otimes \boldsymbol{I} + 2\widehat{G}\, \mathbf{I}^{(s)} \tag{105}$$

where $\widehat{K}$ and $\widehat{G}$ are respectively the tangent bulk and shear moduli of the mixture (both of which are generally functions of $\boldsymbol{\sigma}$, $\overline{p^w}$, B, ϕ, and S_w), $\boldsymbol{I}$ is rank-2 identity tensor, and $\mathbf{I}^{(s)}$ is the minor-symmetric part of the rank-4 identity tensor. The bulk and shear moduli of the mixture are determined using the dry moduli as described below. As detailed in Fuller and Brannon, (2012), apparent pressure dependence of the shear modulus would be better justified by using a hyperelastic formulation (*c.f.* Borja, (2006)).

Bulk modulus - Dry soil: Data for dry sands can be fit reasonably well to the form[10]

$$\frac{\bar{p}_{\mathrm{eff}}}{K_s(\bar{p}_{\mathrm{eff}})} = b_0\, \overline{\varepsilon^e_v} + \frac{b_1(\overline{\varepsilon^e_v})^{b_4}}{b_2(\overline{\varepsilon^e_v})^{b_4} + b_3} \tag{106}$$

Here, $b_0 > 0$, $b_1 > 0$, $b_2 > 0$, $b_3 > 0$, and $b_4 > 1$ are fitting parameters, $\overline{\varepsilon^e_v}$ is the volumetric elastic strain in the matrix, K_s is the bulk modulus of the solid grain material, and

$$\bar{p}_{\mathrm{eff}} := -\tfrac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}_{\mathrm{eff}}) = -\tfrac{1}{3}\mathrm{tr}(\boldsymbol{\sigma} - \boldsymbol{\alpha})\,. \tag{107}$$

The bulk modulus of the solid grains is assumed to be given by

$$K_s(\overline{p^s}) = K_{s0} + n_s\,(\overline{p^s} - \overline{p^s_0}) \tag{108}$$

where K_{s0} and n_s are material properties, and $\overline{p^s_0}$ is a reference pressure.

The tangent bulk modulus of the dry soil is defined as

$$K_d(\bar{p}_{\mathrm{eff}}) := \frac{d\bar{p}_{\mathrm{eff}}}{d\overline{\varepsilon^e_v}}. \tag{109}$$

Then, using (106),

$$K_d(\bar{p}_{\mathrm{eff}}) = \frac{[K_s(\bar{p}_{\mathrm{eff}})]^2}{[K_s(\bar{p}_{\mathrm{eff}}) - n_s\bar{p}_{\mathrm{eff}}]}\left[b_0 + \frac{b_1 b_3 b_4(\overline{\varepsilon^e_v})^{b_4-1}}{\left[b_2(\overline{\varepsilon^e_v})^{b_4} + b_3\right]^2}\right]. \tag{110}$$

To express (110) in closed-form in terms of $\bar{p}$ we have to eliminate $\bar{\varepsilon}^e_v$. But a closed form expression for the volumetric elastic strain cannot be derived from the pressure model. So we find an approximate form of (106) by assuming $b_0 \to 0$, which is valid at moderate to large strains. Then, from (106) with $b_0 = 0$, we have

$$\overline{\varepsilon^e_v} \approx \left[\frac{b_3\bar{p}_{\mathrm{eff}}}{b_1 K_s(\bar{p}_{\mathrm{eff}}) - b_2\bar{p}_{\mathrm{eff}}}\right]^{1/b_4}. \tag{111}$$

[9] For high-rate applications, a crush curve that is measured in quasistatic conditions must be converted to a form that removes low-rate creep (and other effects from heat transfer, fluid seepage, *etc.*), that would not occur in dynamic loading. The constitutive model must be furthermore supplemented with viscoplasticity parameters needed to predict apparent strengthening (beyond hardening) that does pertain to slow loading.

[10] Even if the numerical model is revised to use tabular data for this type of function, this approximate form can potentially serve as an interpolation function that is more accurate than a piecewise-linear fit – especially for low-data situations and for extrapolation beyond available data.

Shear modulus - Dry soil: A non-constant shear modulus may be needed to fit experimental data and to prevent negative values of Poisson's ratio in some simulations. In those situations, a variable Poisson's ratio (ν) is defined as

$$\nu(\bar{p}_{\text{eff}}) = \nu_1 + \nu_2 \exp\left[-\frac{K_d(\bar{p}_{\text{eff}})}{K_s(\bar{p}_{\text{eff}})}\right] \tag{112}$$

where ν_1 and ν_2 are material parameters. The shear modulus is computed using the Poisson's ratio and the dry bulk modulus:

$$G_d(\bar{p}_{\text{eff}}) = \frac{3K_d(\bar{p}_{\text{eff}})(1-2\nu)}{2(1+\nu)}\,. \tag{113}$$

Elastic moduli - Partially saturated soil: Saturated and dry soils are assumed to have the same shear modulus:

$$\widehat{G}(\bar{p}_{\text{eff}}, \overline{p^w}, \overline{\varepsilon_v^p}, \phi, S_w) = G_d(\bar{p}_{\text{eff}})\,. \tag{114}$$

The tangent bulk modulus, $\widehat{K}$, of the partially saturated soil is computed using a variant of the Biot-Gassman model for fully saturated rocks that is valid for long wavelength displacements (Berryman and Milton, 1991; Berryman, 2006; Dvorkin et al., 1999). The Biot-Gassman model is based on the Biot equations of poroelasticity and assumes small strains and long wavelength disturbances relative to the pore size. In this model, the average stress tensor ($\boldsymbol{\sigma}$) of the saturated porous medium is given by

$$\boldsymbol{\sigma} = \left[(H-2G)\text{tr}(\boldsymbol{\varepsilon}) - C\zeta\right]\boldsymbol{I} + 2G\boldsymbol{\varepsilon} \tag{115}$$

and the fluid pressure ($\overline{p^w}$) is

$$\overline{p^w} = M\zeta - C\text{tr}(\boldsymbol{\varepsilon}) \tag{116}$$

where G is the shear modulus, $\boldsymbol{\varepsilon}$ is the strain in the mixture, C is the Biot cross coefficient, and

$$C = \frac{B}{\frac{B}{K_s} + \phi\left(\frac{1}{K_w} - \frac{1}{K_s}\right)}\,,\quad H = K_d + BC + \frac{4}{3}G\,,\quad \zeta = \phi\left[\text{tr}(\boldsymbol{\varepsilon}) - \varepsilon_v^w\right] \tag{117}$$

where

$$M = \frac{C}{B}\,,\quad B := 1 - \frac{K_d}{K_s} \tag{118}$$

and K_d is the bulk modulus of the drained porous frame, K_s is the bulk modulus of the solid grains, ϕ is the porosity, and ε_v^w is the volumetric strain in the fluid.

Using these, the fully saturated bulk modulus for a soil (with a single material in the solid matrix) is

$$\widehat{K} = K_d + B^2 D \tag{119}$$

where

$$\frac{1}{D} = \frac{B(1-B)}{K_d} + \phi\left(\frac{1}{K_w} - \frac{1}{K_s}\right)\,. \tag{120}$$

At partial saturation, we compute the pore fluid bulk modulus using a harmonic mean (lower bound) on the air and water bulk moduli (K_a, K_w):

$$\frac{1}{K_f(\overline{p^w})} = \frac{S_w}{K_w(\overline{p^w})} + \frac{1-S_w}{K_a(\overline{p^w})} \tag{121}$$

and get the expression for the bulk modulus that is used by the partially saturated Arena model:

$$\widehat{K}(\bar{p}_{\text{eff}}, \overline{p^w}, \overline{\varepsilon_v^p}, \phi, S_w) = K_d(\bar{p}_{\text{eff}}) + \frac{B^2(\bar{p}_{\text{eff}})}{\frac{B(\bar{p}_{\text{eff}})}{K_s(\bar{p}_{\text{eff}})} + \phi\left(\frac{1}{K_f(\overline{p^w})} - \frac{1}{K_s(\bar{p}_{\text{eff}})}\right)} \tag{122}$$

where K_d is the bulk modulus of the drained soil, K_f is the bulk modulus of the pore fluid, and K_s is the bulk modulus of the solid grains. For the situation where $B \approx 1$,

$$\widehat{K}(\bar{p}_{\text{eff}}, \overline{p^w}, \overline{\varepsilon_v^p}, \phi, S_w) = K_d(\bar{p}_{\text{eff}}) + \frac{K_f(\overline{p^w})}{\phi}. \tag{123}$$

Bulk modulus model: Solid matrix material The pressure $\bar{p}_s$ in the solid matrix is expressed as

$$\bar{p}_s = K_s \overline{\varepsilon_v^s}\,, \qquad \text{where} \qquad \overline{\varepsilon_v^s} := \ln\left(\frac{V_{s0}}{V_s}\right) \tag{124}$$

Here, K_s is the solid bulk modulus, $\overline{\varepsilon_v^s}$ is the volumetric strain, V_{s0} is the initial volume of the solid, and V_s is the current volume of the solid. The solid bulk modulus is assumed to modeled by the Murnaghan equation:

$$K_s(\bar{p}_s) = K_{s0} + n_s\,(\bar{p}_s - \bar{p}_{s0}) \tag{125}$$

where K_{s0} and n_s are material properties, and $\bar{p}_{s0}$ is a reference pressure.

Bulk modulus model: Pore water The equation of state for the pore water pressure $\bar{p}_w$ is

$$\bar{p}_w = K_w \overline{\varepsilon_v^w} + \bar{p}_0\,, \qquad \text{where} \qquad \overline{\varepsilon_v^w} := \ln\left(\frac{V_{w0}}{V_w}\right) \tag{126}$$

Here, K_w is the water bulk modulus, V_{w0} is the initial volume of water, V_w is the current volume of water, $\bar{p}_0$ is the initial water pressure, and $\bar{\varepsilon}_v^w$ is the volumetric strain in the water. We use the isothermal Murnaghan bulk modulus model for water:

$$K_w(\bar{p}_w) = K_{w0} + n_w\,(\bar{p}_w - \bar{p}_{w0}) \tag{127}$$

where K_{w0} and n_w are material properties, and $\bar{p}_{w0}$ is a reference pressure.

Bulk modulus model: Pore air The isentropic ideal gas equation of state for the pore air is

$$\bar{p}_a = \bar{p}_r\left[\exp(\gamma\,\overline{\varepsilon_v^a}) - 1\right]\,, \qquad \text{where} \qquad \overline{\varepsilon_v^a} := \ln\left(\frac{V_{a0}}{V_a}\right) \tag{128}$$

Here, a subscript "a" flags quantities for the air model analogous to those for the water model in Eq. (126), $\bar{p}_r$ is a reference pressure (101325 Pa) and $\gamma = 1.4$. This model provides dependence of air's bulk modulus of air that is needed to ensure reasonable trends in model predictions in the limit as porosity is crushed out:[11] (K_a) on the volumetric strain:

$$K_a = \frac{d\bar{p}_a}{d\overline{\varepsilon_v^a}} = \gamma\,\bar{p}_r\,\exp(\gamma\,\overline{\varepsilon_v^a}) = \gamma\,(\bar{p}_a + \bar{p}_r)\,. \tag{129}$$

2.7.2 Porosity, saturation, and volumetric strain

The saturation is defined as

$$S_w = \frac{v^w}{v^a + v^w} = 1 - S_a \qquad \Longrightarrow \qquad \frac{v^a}{v^w} = \frac{1 - S_w}{S_w} \tag{130}$$

where v^α is the volume occupied by phase α in the pore volume. Also, the porosity is defined as

$$\phi = \frac{v^a + v^w}{v^s + v^a + v^w} \qquad \Longrightarrow \qquad 1 - \phi = \frac{v^s}{v^s + v^a + v^w}\,. \tag{131}$$

[11] Recognizing that an ideal gas approximation is inappropriate in highly compressed states, this equation can be adopted as phenomenological with γ reinterpreted as an adjustable parameter.

The volumetric strain in each phase is defined as

$$\varepsilon_v^\alpha = \ln\left(\frac{v^\alpha}{v_0^\alpha}\right) \quad \text{where} \quad \alpha = \{s, w, a\} \tag{132}$$

where v_0^α is the initial volume of phase α. The total volumetric strain of the mixture is

$$\exp(\varepsilon_v) = (1 - S_0)\phi_0 \exp(\varepsilon_v^a) + S_0\phi_0 \exp(\varepsilon_v^w) + (1 - \phi_0)\exp(\varepsilon_v^s) \tag{133}$$

where S_0 is the initial saturation and ϕ_0 is the initial porosity. With the assumption that that the total volumetric strain can be additively decomposed into elastic and plastic parts, we have

$$\varepsilon_v = \varepsilon_v^{\mathrm{e}} + \varepsilon_v^{\mathrm{p}} . \tag{134}$$

Therefore,

$$\exp(\varepsilon_v^{\mathrm{p}}) = (1 - S_0)\phi_0 \frac{\exp(\varepsilon_v^a)}{\exp(\varepsilon_v^{\mathrm{e}})} + S_0\phi_0 \frac{\exp(\varepsilon_v^w)}{\exp(\varepsilon_v^{\mathrm{e}})} + (1 - \phi_0)\frac{\exp(\varepsilon_v^s)}{\exp(\varepsilon_v^{\mathrm{e}})} . \tag{135}$$

Saturation equation The saturation equation when $\overline{p^w} = \overline{p^a}$ is

$$S_w(\overline{p^w}) = \frac{\mathcal{C}}{1 - S_0 + \mathcal{C}} , \quad \mathcal{C} := S_0 \exp\left[\overline{\varepsilon_v^a}(\overline{p^w}) - \overline{\varepsilon_v^w}(\overline{p^w})\right] . \tag{136}$$

For a soil that has undergone compressive loading leading to plastic volumetric deformation, the continuity of the phases implies that, after unloading, the saturation is different from the initial value of saturation even when the hydraulic conductivity of the material is zero. Equation (136) then implies that the residual saturation is a function non-zero strains in the water and air phases. Because the air and the water are assumed to be perfectly elastic, the saturation is a function of a residual pressure which can be thought of as a "plastic" pore-pressure which is a function of the plastic volumetric strain.

One way of defining the "plastic" pore pressure can be that the unrecoverable part of the saturation is

$$S_w^{\mathrm{p}}(\overline{p_{\mathrm{p}}^w}) = \frac{\mathcal{C}^{\mathrm{p}}}{1 - S_0 + \mathcal{C}^{\mathrm{p}}} , \quad \mathcal{C}^{\mathrm{p}} := S_0 \exp\left[\overline{\varepsilon_v^a}(\overline{p_{\mathrm{p}}^w}) - \overline{\varepsilon_v^w}(\overline{p_{\mathrm{p}}^w})\right] . \tag{137}$$

Porosity equation Recall that the porosity is given by

$$1 - \phi = \frac{v^s}{v} \quad \Longrightarrow \quad \frac{1 - \phi}{1 - \phi_0} = \frac{v^s}{v_0^s}\frac{v_0}{v} = \exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^s}) = \frac{\exp(\varepsilon_v^s)}{\exp(\varepsilon_v)} . \tag{138}$$

If we know the total volumetric strain and the volumetric strains in the fluids, we can use equation (133) to write the above as

$$1 - \phi = \frac{\exp(\varepsilon_v) - (1 - S_0)\phi_0 \exp(\varepsilon_v^a) - S_0\phi_0 \exp(\varepsilon_v^w)}{\exp(\varepsilon_v)} \tag{139}$$

When the pore water and air pressure are equal to each other, and also equal to the intrinsic pressure in the solid grains), the porosity equation (139) can be expressed as

$$\phi(\overline{p^w}) = (1 - S_0)\phi_0 \exp\left[\overline{\varepsilon_v}(\overline{p^w}) - \overline{\varepsilon_v^a}(\overline{p^w})\right] + S_0\phi_0 \exp\left[\overline{\varepsilon_v}(\overline{p^w}) - \overline{\varepsilon_v^w}(\overline{p^w})\right] \tag{140}$$

The primary cause of the residual pressure in a soil that has undergone volumetric plastic deformation is the irreversible change in porosity. Let the unrecoverable part of the porosity (called the "unloaded" porosity by Brannon, (2007)) be ϕ^{p}. We could define the "unloaded" or "plastic" porosity as

$$\phi^{\mathrm{p}}(\overline{p_{\mathrm{p}}^w}) = (1 - S_0)\phi_0 \exp\left[\overline{\varepsilon_v}(\overline{p_{\mathrm{p}}^w}) - \overline{\varepsilon_v^a}(\overline{p_{\mathrm{p}}^w})\right] + S_0\phi_0 \exp\left[\overline{\varepsilon_v}(\overline{p_{\mathrm{p}}^w}) - \overline{\varepsilon_v^w}(\overline{p_{\mathrm{p}}^w})\right] \tag{141}$$

2.7.3 Rate-independent plasticity

For rate-independent plastic deformations we use the canonical phenomenological model that consists of a yield function (in our case, a limit surface), a flow rule and the associated consistency condition, and models of internal variable evolution.

Arena yield function/limit surface If the volumetric and deviatoric components of the total stress are

$$\bar{p} := -\tfrac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}) \quad \text{and} \quad \boldsymbol{s} := \boldsymbol{\sigma} + \bar{p}\boldsymbol{I} \tag{142}$$

we can define

$$\bar{p}_{\mathrm{eff}} := -\tfrac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}_{\mathrm{eff}}) = \bar{p} - B\overline{p^w} \tag{143}$$

and

$$J_2^{\mathrm{eff}} := \tfrac{1}{2}\boldsymbol{s}_{\mathrm{eff}} : \boldsymbol{s}_{\mathrm{eff}} \quad \text{where} \quad \boldsymbol{s}_{\mathrm{eff}} := \boldsymbol{\sigma}_{\mathrm{eff}} + \bar{p}_{\mathrm{eff}}\boldsymbol{I} = \boldsymbol{\sigma} + \bar{p}\boldsymbol{I} = \mathbf{s}\,. \tag{144}$$

Then the Arena yield function can be expressed as

$$f(\boldsymbol{\sigma}, B, \overline{p^w}, \overline{X}) = \beta\sqrt{J_2^{\mathrm{eff}}} - F_f(\bar{p}_{\mathrm{eff}})\, F_c(\bar{p}_{\mathrm{eff}}, \overline{X}) \tag{145}$$

where

$$F_f(\bar{p}_{\mathrm{eff}}) = a_1 - a_3 \exp[-3a_2\bar{p}_{\mathrm{eff}}] + 3a_4\bar{p}_{\mathrm{eff}} \tag{146}$$

and

$$F_c(\bar{p}_{\mathrm{eff}}, \overline{X}) = \begin{cases} 1 & \text{for} \quad 3\bar{p}_{\mathrm{eff}} \le \bar{\kappa} \\ \sqrt{1 - \left(\dfrac{3\bar{p}_{\mathrm{eff}} - \bar{\kappa}}{\overline{X}_{\mathrm{eff}} - \bar{\kappa}}\right)^2} & \text{for} \quad 3\bar{p}_{\mathrm{eff}} > \bar{\kappa}\,. \end{cases} \tag{147}$$

Here a_i are material parameters, $\overline{X}_{\mathrm{eff}}(\boldsymbol{\varepsilon}^{\mathrm{p}}, B, \overline{p^w}) = \overline{X} - 3B\overline{p^w}$ is the shifted form of the apparent hydrostatic compressive strength ($\overline{X}/3$) of the partially saturated material, and $\bar{\kappa}$ is the branch point at which the cap function F_c starts decreasing until it reaches the hydrostatic strength point ($\overline{X}$):

$$\bar{\kappa} = 3\bar{p}_{\mathrm{eff}}^{\mathrm{peak}} - (3\bar{p}_{\mathrm{eff}}^{\mathrm{peak}} - \overline{X}_{\mathrm{eff}})R_c \tag{148}$$

where $\bar{p}_{\mathrm{eff}}^{\mathrm{peak}}$ is the maximum hydrostatic tensile stress that the material can support and R_c is a cap ratio. Non-associativity is modeled using the parameter β that modifies $\sqrt{J_2^{\mathrm{eff}}}$.

Yield condition, flow rule, and consistency For continued plastic loading on the yield surface, the yield condition is

$$f(\boldsymbol{\sigma}, \overline{p_{\mathrm{e}}^w}, \overline{p_{\mathrm{p}}^w}, B^{\mathrm{e}}, B^{\mathrm{p}}, \phi^{\mathrm{e}}, \phi^{\mathrm{p}}, S_w^{\mathrm{e}}, S_w^{\mathrm{p}}, \overline{X}^{\mathrm{e}}, \overline{X}^{\mathrm{p}}; \boldsymbol{\varepsilon}^{\mathrm{p}}) = 0 \tag{149}$$

where $f(\dots)$ is the yield function. The plastic flow rule is

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \boldsymbol{d}^{\mathrm{p}} = \dot{\lambda}\boldsymbol{M} \tag{150}$$

where $\boldsymbol{M}$ is the unit tensor in the direction of the plastic rate of deformation. The consistency condition implies that for continued plastic loading

$$\dot{f}(\dots) = 0\,. \tag{151}$$

Assuming appropriate smoothness of $f(\dots)$, we have

$$\begin{aligned} &\frac{\partial f}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{\partial f}{\partial \overline{p_{\mathrm{e}}^w}}\dot{\overline{p_{\mathrm{e}}^w}} + \frac{\partial f}{\partial B^{\mathrm{e}}}\dot{B}^{\mathrm{e}} + \frac{\partial f}{\partial \phi^{\mathrm{e}}}\dot{\phi}^{\mathrm{e}} + \frac{\partial f}{\partial S_w^{\mathrm{e}}}\dot{S}_w^{\mathrm{e}} + \frac{\partial f}{\partial \overline{X}^{\mathrm{e}}}\dot{\overline{X}}^{\mathrm{e}} + \\ &\dot{\lambda}\left[h_p\frac{\partial f}{\partial \overline{p_{\mathrm{p}}^w}} + h_B\frac{\partial f}{\partial B^{\mathrm{p}}} + h_\phi\frac{\partial f}{\partial \phi^{\mathrm{p}}} + h_{S_w}\frac{\partial f}{\partial S_w^{\mathrm{p}}} + h_X\frac{\partial f}{\partial \overline{X}^{\mathrm{p}}}\right] = 0\,. \end{aligned} \tag{152}$$

The normal to the yield surface is

$$\boldsymbol{N} = \frac{\partial f}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial \overline{p}^{w}_{\mathrm{e}}}\frac{\partial \overline{p}^{w}_{\mathrm{e}}}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial B^{\mathrm{e}}}\frac{\partial B^{\mathrm{e}}}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial \phi^{\mathrm{e}}}\frac{\partial \phi^{\mathrm{e}}}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial S^{\mathrm{e}}_{w}}\frac{\partial S^{\mathrm{e}}_{w}}{\partial \boldsymbol{\sigma}} + \frac{\partial f}{\partial \overline{X}^{\mathrm{e}}}\frac{\partial \overline{X}^{\mathrm{e}}}{\partial \boldsymbol{\sigma}} \tag{153}$$

and the ensemble hardening modulus is

$$H = -\left[h_p\frac{\partial f}{\partial \overline{p}^{w}_{\mathrm{p}}} + h_B\frac{\partial f}{\partial B^{\mathrm{p}}} + h_\phi\frac{\partial f}{\partial \phi^{\mathrm{p}}} + h_{S_w}\frac{\partial f}{\partial S^{\mathrm{p}}_{w}} + h_X\frac{\partial f}{\partial \overline{X}^{\mathrm{p}}}\right]/\|\boldsymbol{N}\| \,. \tag{154}$$

The find the plastic flow rate parameter, we write the equation for coupled elasticity as

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}^{\mathrm{e}} : (\boldsymbol{d}^s - \boldsymbol{d}^{\mathrm{p}}) - \dot{\lambda}\boldsymbol{Z} \,. \tag{155}$$

After substituting the flow rule into the above relation and defining

$$\dot{\boldsymbol{\sigma}}_{\mathrm{trial}} := \mathbf{C}^{\mathrm{e}} : \boldsymbol{d}^s \quad \text{and} \quad \boldsymbol{P} := \mathbf{C}^{\mathrm{e}} : \boldsymbol{M} + \boldsymbol{Z} \,. \tag{156}$$

we get

$$\dot{\lambda} = \frac{\widehat{\boldsymbol{N}} : \dot{\boldsymbol{\sigma}}_{\mathrm{trial}}}{\widehat{\boldsymbol{N}} : \boldsymbol{P} + H} \,. \tag{157}$$

Therefore,

$$\dot{\boldsymbol{\sigma}} = \left[\mathbf{I}^{(s)} - \frac{\boldsymbol{P}\otimes\widehat{\boldsymbol{N}}}{\widehat{\boldsymbol{N}} : \boldsymbol{P} + H}\right] : \dot{\boldsymbol{\sigma}}_{\mathrm{trial}} =: (\mathbf{C}^{\mathrm{p}} : \mathbf{C}^{\mathrm{e}}) : \boldsymbol{d}^s = \mathbf{C}^{\mathrm{pe}} : \boldsymbol{d}^s \,. \tag{158}$$

The consistency condition equation (152) can be simplified to

$$\widehat{\boldsymbol{N}} : \dot{\boldsymbol{\sigma}} = \dot{\lambda} H \quad \text{where} \quad \widehat{\boldsymbol{N}} = \boldsymbol{N}/\|\boldsymbol{N}\| \,. \tag{159}$$

Internal variable evolution functions If we assume that

$$\overline{p}^{w}_{\mathrm{p}} \equiv \overline{p}^{w}_{\mathrm{p}}(\boldsymbol{\varepsilon}^{\mathrm{p}}) \,, \quad B^{\mathrm{p}} \equiv B^{\mathrm{p}}(\boldsymbol{\varepsilon}^{\mathrm{p}}) \,, \quad \phi^{\mathrm{p}} \equiv \phi^{\mathrm{p}}(\boldsymbol{\varepsilon}^{\mathrm{p}}) \,, \quad S^{\mathrm{p}}_{w} \equiv S^{\mathrm{p}}_{w}(\boldsymbol{\varepsilon}^{\mathrm{p}}) \,, \quad \overline{X}^{\mathrm{p}} \equiv \overline{X}^{\mathrm{p}}(\boldsymbol{\varepsilon}^{\mathrm{p}})$$

we can write

$$\begin{aligned} \dot{\overline{p}}^{w}_{\mathrm{p}} &= \frac{d\overline{p}^{w}_{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{d}^p \,, \quad \dot{B}^{\mathrm{p}} = \frac{dB^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{d}^p \,, \quad \dot{\phi}^{\mathrm{p}} = \frac{d\phi^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{d}^p \\ \dot{S}^{\mathrm{p}}_{w} &= \frac{dS^{\mathrm{p}}_{w}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{d}^p \,, \quad \dot{\overline{X}}^{\mathrm{p}} = \frac{d\overline{X}^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{d}^p \,. \end{aligned} \tag{160}$$

Using the flow rule (150), we have

$$\begin{aligned} \dot{\overline{p}}^{w}_{\mathrm{p}} &= \dot{\lambda}\frac{d\overline{p}^{w}_{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad \dot{B}^{\mathrm{p}} = \dot{\lambda}\frac{dB^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad \dot{\phi}^{\mathrm{p}} = \dot{\lambda}\frac{d\phi^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \\ \dot{S}^{\mathrm{p}}_{w} &= \dot{\lambda}\frac{dS^{\mathrm{p}}_{w}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad \dot{\overline{X}}^{\mathrm{p}} = \dot{\lambda}\frac{d\overline{X}^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,. \end{aligned} \tag{161}$$

Comparing the above with equations (104), we have

$$\begin{aligned} h_p &= \frac{d\overline{p}^{w}_{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad h_B = \frac{dB^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad h_\phi = \frac{d\phi^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \\ h_{S_w} &= \frac{dS^{\mathrm{p}}_{w}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,, \quad h_X = \frac{d\overline{X}^{\mathrm{p}}}{d\boldsymbol{\varepsilon}^{\mathrm{p}}} : \boldsymbol{M} \,. \end{aligned} \tag{162}$$

Noting that the derivative of the trace of a rank-2 tensor with respect to the tensor is equal to the rank-2 identity tensor, we have

$$\begin{aligned} h_p &= \frac{d\overline{p}^{w}_{\mathrm{p}}}{d\varepsilon^{\mathrm{p}}_{v}}\mathrm{tr}(\boldsymbol{M}) \,, \quad h_B = \frac{dB^{\mathrm{p}}}{d\varepsilon^{\mathrm{p}}_{v}}\mathrm{tr}(\boldsymbol{M}) \,, \quad h_\phi = \frac{d\phi^{\mathrm{p}}}{d\varepsilon^{\mathrm{p}}_{v}}\mathrm{tr}(\boldsymbol{M}) \\ h_{S_w} &= \frac{dS^{\mathrm{p}}_{w}}{d\varepsilon^{\mathrm{p}}_{v}}\mathrm{tr}(\boldsymbol{M}) \,, \quad h_X = \frac{d\overline{X}^{\mathrm{p}}}{d\varepsilon^{\mathrm{p}}_{v}}\mathrm{tr}(\boldsymbol{M}) \,. \end{aligned} \tag{163}$$

2.7.4 Evolution of internal variables

The internal variables that evolve include the hydrostatic compressive strength, the pore pressure, the saturation, the porosity, and potentially the Biot ratio.

Hydrostatic compressive strength evolution The drained hydrostatic compressive strength ($\overline{X}_d/3$) is found from the empirical drained material crush curve using

$$\overline{X}_d(\overline{\varepsilon_v^p}, \phi_0) - p_0 = p_1 \left[\frac{1-\exp(-p_3)}{1-\exp(-p_3+\overline{\varepsilon_v^p})} - 1 \right]^{1/p_2} \quad , \quad p_3 := -\ln(1-\phi_0) \tag{164}$$

where $\overline{\varepsilon_v^p} = -\mathrm{tr}(\boldsymbol{\varepsilon}^p)$ is the volumetric plastic strain, $\overline{X}_d/3$ is the drained hydrostatic compressive strength, p_0, p_1, p_2, p_3 are model parameters, and ϕ_0 is the initial porosity. The derivative of $\overline{X}_d$ is

$$\frac{d\overline{X}_d}{d\overline{\varepsilon_v^p}} = \frac{p_1[1-\exp(-p_3)]\exp(-p_3+\overline{\varepsilon_v^p})}{p_2[1-\exp(-p_3+\overline{\varepsilon_v^p})]^2} \left[\frac{1-\exp(-p_3)}{1-\exp(-p_3+\overline{\varepsilon_v^p})} - 1 \right]^{1/p_2-1} . \tag{165}$$

The effective hydrostatic compressive strength for a partially saturated material is expected to be different from the drained value (we do not have any direct experimental data which supports that conjecture). We follow the approach used by Grujicic (M. Grujicic et al., 2009) and assume a model of the form

$$\overline{X}_{\mathrm{eff}}(\overline{\varepsilon_v^p}, \phi_0, S_w) - p_0 = \left[(1-S_w) + p_1^{\mathrm{sat}} S_w\right](\overline{X}_d - p_0) \tag{166}$$

where $\overline{X} = \overline{X}_{\mathrm{eff}} + 3B\overline{p^w}$ and $p_1 \times p_1^{\mathrm{sat}}$ is the value of p_1 in a fully saturated material.

The rate form of the hydrostatic compressive strength equation (166) is

$$\dot{\overline{X}} = \dot{\overline{X}}_{\mathrm{eff}} + 3(\dot{B}\overline{p^w} + B\dot{\overline{p^w}}) = \dot{\overline{X^e}} + \dot{\overline{X^e}} \tag{167}$$

Details of the expressions are shown below

$$\begin{aligned}
\dot{\overline{X}} &= \dot{\overline{X}}_{\mathrm{eff}} + 3(\dot{B}\overline{p^w} + B\dot{\overline{p^w}}) \\
&= \left[(1-S_w) + p_1^{\mathrm{sat}} S_w\right]\dot{\overline{X}}_d + (p_1^{\mathrm{sat}} - 1)\dot{S}_w\overline{X}_d + 3(\dot{B}\overline{p^w} + B\dot{\overline{p^w}}) \\
&= \left[(1-S_w) + p_1^{\mathrm{sat}} S_w\right]\frac{d\overline{X}_d}{d\overline{\varepsilon_v^p}}\mathrm{tr}(\boldsymbol{d}^p) + (p_1^{\mathrm{sat}} - 1)\left(\dot{S}_w^e + \dot{S}_w^p\right)\overline{X}_d + 3[\dot{B}\overline{p^w} + B(\dot{\overline{p_p^w}} + \dot{\overline{p_e^w}})] \\
&= \left[(1-S_w) + p_1^{\mathrm{sat}} S_w\right]\frac{d\overline{X}_d}{d\overline{\varepsilon_v^p}}[\dot{\lambda}\mathrm{tr}(\boldsymbol{M})] + \left[\frac{\overline{X}_d(p_1^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}}\right]\left[\mathrm{tr}(\boldsymbol{d}^e) + \dot{\lambda}\mathrm{tr}(\boldsymbol{M})\right] + \\
&\quad 3\left[\frac{\overline{p^w}}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right)\left[\mathrm{tr}(\boldsymbol{d}^e) + \dot{\lambda}\mathrm{tr}(\boldsymbol{M})\right] + \frac{B}{\mathcal{B}}\mathrm{tr}(\boldsymbol{d}^e) + \frac{\dot{\lambda}B}{\mathcal{B}}\mathrm{tr}(\boldsymbol{M})\right] \\
&= \left[\frac{\overline{X}_d(p_1^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}} + \frac{3\overline{p^w}}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right) + \frac{3B}{\mathcal{B}}\right]\mathrm{tr}(\boldsymbol{d}^e) + \\
&\quad \dot{\lambda}\left[\left[(1-S_w) + p_1^{\mathrm{sat}} S_w\right]\frac{d\overline{X}_d}{d\overline{\varepsilon_v^p}} + \frac{\overline{X}_d(p_1^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}} + \frac{3\overline{p^w}}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right) + \frac{3B}{\mathcal{B}}\right]\mathrm{tr}(\boldsymbol{M}) .
\end{aligned} \tag{168}$$

We could define "elastic" and "plastic" parts such that $\dot{\overline{X}} = \dot{\overline{X^e}} + \dot{\overline{X^p}}$ where, we could define "elastic" and

"plastic" parts such that

$$\begin{aligned}\dot{\overline{X}}^{\mathrm{e}} &:= \left[\frac{\overline{X}_d(p_1^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}} + \frac{3\overline{p^w}}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right) + \frac{3B}{\mathcal{B}}\right]\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}})\\ \dot{\overline{X}}^{\mathrm{p}} &:= \dot{\lambda}\left[\left[(1-S_w)+p_1^{\mathrm{sat}}S_w\right]\frac{d\overline{X}_d}{d\overline{\varepsilon_v^{\mathrm{p}}}} + \right.\\ &\qquad \left.\frac{\overline{X}_d(p_1^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}} + \frac{3\overline{p^w}}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right) + \frac{3B}{\mathcal{B}}\right]\mathrm{tr}(\boldsymbol{M})\,.\end{aligned} \tag{169}$$

Pore pressure evolution As discussed earlier, an expression for the evolution of the pore pressure can be found from the mass balance equations:

$$\dot{\overline{p^w}} = \frac{1}{\mathcal{B}}\mathrm{tr}(\boldsymbol{d}^s) \tag{170}$$

where $\mathcal{B} \equiv \mathcal{B}(\overline{p^w}, B, \phi, S_w)$ is given in equation (92). If we express the above equation in terms of elastic and plastic components, we have

$$\dot{\overline{p_{\mathrm{e}}^w}} + \dot{\overline{p_{\mathrm{p}}^w}} = \frac{1}{\mathcal{B}}\left[\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}}) + \mathrm{tr}(\boldsymbol{d}^{\mathrm{p}})\right]\,. \tag{171}$$

We define the "elastic" and "plastic" pore pressure evolution as

$$\dot{\overline{p_{\mathrm{e}}^w}} := \frac{1}{\mathcal{B}}\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}}) \quad \text{and} \quad \dot{\overline{p_{\mathrm{p}}^w}} := \frac{1}{\mathcal{B}}\mathrm{tr}(\boldsymbol{d}^{\mathrm{p}}) = \frac{\dot{\lambda}}{\mathcal{B}}\mathrm{tr}(\boldsymbol{M})\,. \tag{172}$$

Saturation evolution From equation (136) we have

$$\begin{aligned}\dot{S}_w &= \frac{(1-S_0)\dot{\mathcal{C}}}{(1-S_0+\mathcal{C})^2} = \frac{1-S_0}{(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}}\dot{\overline{p^w}}\\ &= \frac{1-S_0}{(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}}\left(\dot{\overline{p_{\mathrm{e}}^w}} + \frac{\dot{\lambda}}{\mathcal{B}}\mathrm{tr}(\boldsymbol{M})\right)\,.\end{aligned} \tag{173}$$

We can express the above as $\dot{S}_w = \dot{S}_w^{\mathrm{e}} + \dot{S}_w^{\mathrm{p}}$, where the evolution equations for the "elastic" and "plastic" parts of the saturation are

$$\begin{aligned}\dot{S}_w^{\mathrm{e}} &:= \left[\frac{1-S_0}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}}\right]\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}})\\ \dot{S}_w^{\mathrm{p}} &:= \dot{\lambda}\left[\frac{1-S_0}{\mathcal{B}(1-S_0+\mathcal{C})^2}\frac{d\mathcal{C}}{d\overline{p^w}}\right]\mathrm{tr}(\boldsymbol{M})\,.\end{aligned} \tag{174}$$

Porosity evolution If we take the material time derivative of equation (140), we get

$$\dot{\phi} = (1-S_0)\phi_0\exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}\right)\left(\dot{\overline{\varepsilon_v}} - \frac{d\overline{\varepsilon_v^a}}{d\overline{p^w}}\dot{\overline{p^w}}\right) + S_0\phi_0\exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^w}\right)\left(\dot{\overline{\varepsilon_v}} - \frac{d\overline{\varepsilon_v^w}}{d\overline{p^w}}\dot{\overline{p^w}}\right)$$

or

$$\begin{aligned}\dot{\phi} = -\Bigg[&(1-S_0)\phi_0\exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}\right)\left(1 + \frac{1}{K_a\mathcal{B}}\right) + \\ &S_0\phi_0\exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^w}\right)\left(1 + \frac{1}{K_w\mathcal{B}}\right)\Bigg]\mathrm{tr}(\boldsymbol{d}^s)\,.\end{aligned} \tag{175}$$

We can express the above as $\dot{\phi} = \dot{\phi}^{\mathrm{e}} + \dot{\phi}^{\mathrm{p}}$, where

$$\begin{aligned}\dot{\phi}^{\mathrm{e}} &= -\left[(1-S_0)\phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}\right)\left(1+\frac{1}{K_a\mathcal{B}}\right) + \right.\\ &\qquad \left. S_0\phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^w}\right)\left(1+\frac{1}{K_w\mathcal{B}}\right)\right]\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}})\\ \dot{\phi}^{\mathrm{p}} &= -\dot{\lambda}\left[(1-S_0)\phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}\right)\left(1+\frac{1}{K_a\mathcal{B}}\right) + \right.\\ &\qquad \left. S_0\phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^w}\right)\left(1+\frac{1}{K_w\mathcal{B}}\right)\right]\mathrm{tr}(\boldsymbol{M})\,.\end{aligned} \tag{176}$$

Biot ratio evolution If we assume that $\overline{p^s} = \overline{p^w} = \overline{p^a}$, the material time derivative of the Biot ratio (89) is

$$\dot{B} := -\frac{1}{K_s}\left(\dot{\tilde{K}}_s - \frac{K_d}{K_s}\dot{K}_s\right) = -\frac{1}{K_s}\left(\frac{dK_d}{d\overline{p^w}} - \frac{K_d}{K_s}\frac{dK_s}{d\overline{p^w}}\right)\dot{\overline{p^w}}\,. \tag{177}$$

We can define "elastic" and "plastic" parts of the rate of the Biot ratio as $\dot{B} = \dot{B}^{\mathrm{e}} + \dot{B}^{\mathrm{p}}$ where

$$\begin{aligned}\dot{B}^{\mathrm{e}} &:= \frac{1}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right)\mathrm{tr}(\boldsymbol{d}^{\mathrm{e}})\\ \dot{B}^{\mathrm{p}} &:= \dot{\lambda}\left[\frac{1}{\mathcal{B}K_s}\left(\frac{K_d}{K_s}\frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}}\right)\right]\mathrm{tr}(\boldsymbol{M})\,.\end{aligned} \tag{178}$$

2.7.5 Density-dependence model

Unlike its predecessor models, ARENA accounts for the effect of sample preparation. Mechanical properties of sands and soils depend strongly on the initial density. Two otherwise identical sands of the same composition and grain morphology will, for example, exhibit significantly different behavior if one of them is shaken to compact the grains prior to subjecting it to a load.

Suppose that a sample's initial porosity ϕ_0 differs from the initial porosity ϕ_{ref} of the reference material that was used to calibrate the bulk modulus parameters and the crush curve. Following Pabst and Gregorova (Pabst and Gregorová, 2015), the need to re-calibrate elastic properties for each different initial porosity is avoided by using a modulus scaling factor K_{fac}

$$K_{\mathrm{fac}} = \exp\left[-\frac{\phi_0}{1-\phi_0} + \frac{\phi_{\mathrm{ref}}}{1-\phi_{\mathrm{ref}}}\right] \tag{179}$$

such that the bulk and shear moduli of the test material is

$$K_d \leftarrow K_{\mathrm{fac}}K_d\,, \quad \text{and} \quad G \leftarrow K_{\mathrm{fac}}G\,. \tag{180}$$

These scaled moduli are used in the partially saturated model.

The effect of sample compaction on in hydrostatic strength is modeled phenomenologically and calibrated using high density dry Mason sand SHPB tests. Specifically, examination of data for high-density dry Mason sand used in SHPB tests shows that the crush curve for the high-density sand is of the same shape as the low density sand except scaled in the pressure axis. For the crush formula adopted from Brannon, Fuller, et al., (2015), this scaling can be achieved by simply replacing the reference-density fitting parameter p_1 as follows:

$$p_1 \leftarrow p_1 \exp\left[\rho_{\mathrm{fac}}K_{\mathrm{fac}}(K_{\mathrm{fac}}-1)\right] \tag{181}$$

where ρ_{fac} is calibrated using high density uniaxial strain compression data and has a value between 1 and 10.

2.7.6 Damage model

ARENA uses the KAYENTA damage model (Brannon, Fuller, et al., 2015) to model the collapse of the limit surface with increasing damage. A scalar parameter $D \in [0,1]$ is used to modify the parameters that define the limit surface. These parameters are I_1^{peak}, f^{slope}, $\sqrt{J_2^{\text{coh}}}$, and y^{slope} where

$$\begin{aligned} I_1^{\text{peak}} &\leftarrow \alpha_1 \\ f^{\text{slope}} &\leftarrow \alpha_2 \\ a_4 &\leftarrow \alpha_3 = y^{\text{slope}} \\ a_1 &\leftarrow \alpha_4 = \sqrt{J_2^{\text{coh}}} \\ a_2 &\leftarrow \frac{\alpha_2 - \alpha_3}{\alpha_4 - \alpha_1\alpha_3} \\ a_3 &\leftarrow (\alpha_4 - \alpha_1\alpha_3)\exp(-a_2\alpha_1)\,. \end{aligned} \tag{182}$$

Let the input undamaged values of these parameters be

$$I_1^{\text{peak,intact}}\,,\quad f^{\text{slope,intact}}\,,\quad \sqrt{J_2^{\text{coh,intact}}}\,,\quad y^{\text{slope,intact}}\,. \tag{183}$$

Also, let the input fully damaged values of the four parameters be

$$I_1^{\text{peak,failed}}\,,\quad f^{\text{slope,failed}}\,,\quad \sqrt{J_2^{\text{coh,failed}}}\,,\quad y^{\text{slope,failed}}\,. \tag{184}$$

Then the current values of the four parameters are computed using linear interpolation between the intact and failed values:

$$\begin{aligned} I_1^{\text{peak}} &= (1-D)I_1^{\text{peak,intact}} + DI_1^{\text{peak,failed}} \\ f^{\text{slope}} &= (1-D)f^{\text{slope,intact}} + Df^{\text{slope,failed}} \\ \sqrt{J_2^{\text{coh}}} &= (1-D)\sqrt{J_2^{\text{coh,intact}}} + D\sqrt{J_2^{\text{coh,failed}}} \\ y^{\text{slope}} &= (1-D)y^{\text{slope,intact}} + Dy^{\text{slope,intact}}\,. \end{aligned} \tag{185}$$

Also, to incorporate variability and size effects during the failure process, the parameters are scaled by the material point volume using a relation of the form

$$\begin{aligned} I_1^{\text{peak}} &= I_1^{\text{peak,intact}}\left[\frac{v^{\text{expt}}\ln R}{v^{\text{elem}}\ln(1/2)}\right]^{1/m}\,, \quad f^{\text{slope}} = f^{\text{slope,intact}}\left[\frac{v^{\text{expt}}\ln R}{v^{\text{elem}}\ln(1/2)}\right]^{1/m} \\ \sqrt{J_2^{\text{coh}}} &= \sqrt{J_2^{\text{coh,intact}}}\left[\frac{v^{\text{expt}}\ln R}{v^{\text{elem}}\ln(1/2)}\right]^{1/m}\,, \quad y^{\text{slope}} = y^{\text{slope,intact}}\left[\frac{v^{\text{expt}}\ln R}{v^{\text{elem}}\ln(1/2)}\right]^{1/m} \end{aligned} \tag{186}$$

where $\{I_1^{\text{peak}}, f^{\text{slope}}, \sqrt{J_2^{\text{coh}}}, y^{\text{slope}}\}^{\text{intact}}$ are mean values from experiments performed on samples that have volume v^{expt}, v^{elem} is the volume of the material point, $R \sim \mathcal{U}[0,1]$ is a uniformly distributed random number, and m is a Weibull modulus.

At this stage we need a model for the evolution of D. KAYENTA uses a time-to-failure model that depends on two new input parameters: the time to failure (t^{fail}) and the failure speed (f^{speed}), and defines a "coherence" parameter (C) as

$$C := 1 - D = \frac{\exp\left[-f^{\text{speed}}\left(1 - \frac{t^{\text{grow}}}{t^{\text{fail}}}\right)\right]}{1 + \exp\left[-f^{\text{speed}}\left(1 - \frac{t^{\text{grow}}}{t^{\text{fail}}}\right)\right]} \tag{187}$$

where t^{grow} is state variable that increases only when the stress state is on the limit surface. The value of C is never allowed to increase.

2.7.7 Implementing parameter variability

A Weibull distribution is used for each parameter that is varied in the model. One can either use a mean based or a median based approach. In the ARENA model we use the median-based approach as discussed in Appendix C. Note that the form used in our model is

$$y = \left[\frac{v_{\text{expt}}}{v_{\text{elem}}}\right]^{1/a} \bar{y} \left[\frac{\ln x}{\ln(1/2)}\right]^{1/a} . \tag{188}$$

For our purposes, if we use the C++11 Weibull distribution generator, we can incorporate the volume scaling by just multiplying the scaling factor to the number generated, i.e.,

$$y = \left[\frac{v_{\text{expt}}}{v_{\text{elem}}}\right]^{1/m} \text{We}(\bar{y}, a, b, R) \tag{189}$$

where R is the uniformly distributed pseudo-random number in $[0,1]$ generated by the Mersenne twister algorithm.

2.7.8 Simplification of rate-independent model

Our implementation of the ARENA partially saturated soil model uses Homel's "consistency bisection" algorithm (Homel, Guilkey, and Brannon, 2015) to find the plastic strain direction and to update the internal state variables. A closest-point return algorithm in transformed stress space is used to project the trial stress state on to the yield surface. Because of the nonlinearities in the material models, it is easier to solve the problem by dividing the strain increment into substeps.

Recall from equation (156) and (102) that

$$\dot{\boldsymbol{\sigma}}_{\text{trial}} := \mathbf{C}^{\text{e}} : \boldsymbol{d}^s \tag{190}$$

where

$$\mathbf{C}^{\text{e}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{p_{\text{e}}^w}} \otimes \frac{\partial \overline{p_{\text{e}}^w}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial B^{\text{e}}} \otimes \frac{\partial B^{\text{e}}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \phi^{\text{e}}} \otimes \frac{\partial \phi^{\text{e}}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial S_w^{\text{e}}} \otimes \frac{\partial S_w^{\text{e}}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} + \frac{\partial \boldsymbol{\sigma}}{\partial \overline{X}^{\text{e}}} \otimes \frac{\partial \overline{X}^{\text{e}}}{\partial \boldsymbol{\varepsilon}^{\text{e}}} . \tag{191}$$

In the ARENA model we assume an isotropic elastic tangent modulus

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}^e} = \mathbf{C}_\sigma^{\text{e}} = \left(\widehat{K} - \tfrac{2}{3}\widehat{G}\right) \boldsymbol{I} \otimes \boldsymbol{I} + 2\widehat{G}\, \mathbf{I}^{(s)} \tag{192}$$

and also neglect all other quantities in the expression for $\mathbf{C}^{\text{e}}$, i.e., $\mathbf{C}^{\text{e}} = \mathbf{C}_\sigma^{\text{e}}$.

The update algorithm uses the standard predictor-corrector approach of hypoelastic-plasticity where a trial predictor stress is computed first and then a corrector return algorithm is used to locate the position of the correct stress on the yield surface. This approach requires that the trial stress ($\boldsymbol{\sigma}_{\text{trial}}$) is computed using the relation

$$\boldsymbol{\sigma}_{\text{trial}} = \boldsymbol{\sigma}^n + \mathbf{C}^{\text{e}} : (\boldsymbol{d}^s\, \Delta t) . \tag{193}$$

After the trial stress is computed, the timestep is subdivided into substeps based on the characteristic dimension of the yield surface relative to the magnitude of the trial stress increment ($\boldsymbol{\sigma}_{\text{trial}} - \boldsymbol{\sigma}^n$). The substep size is then recomputed by comparing the elastic bulk modulus $\widehat{K}(\boldsymbol{\sigma}_{\text{trial}}, \dots)$ with that at $\boldsymbol{\sigma}^n$ to make sure that the nonlinear elastic solution is accurate.

Recall that the parameters of the yield function, not all of which are independent, are a_1, a_2, a_3, a_4, R_c, β, $I_1^{\text{peak}} = 3p_{\text{eff}}^{\text{peak}}$. The code uses the parameters α_1 := `PEAKI1`, α_2 := `FSLOPE`, α_3 := `YSLOPE`, and α_4 :=

STREN in some places. The relationship between these sets of parameters is:

$$\begin{aligned} I_1^{\text{peak}} &\leftarrow \alpha_1 = \texttt{PEAKI1} \\ a_1 &\leftarrow \alpha_4 = \texttt{STREN} \\ a_2 &\leftarrow \frac{\alpha_2 - \alpha_3}{\alpha_4 - \alpha_1\alpha_3} \\ a_3 &\leftarrow (\alpha_4 - \alpha_1\alpha_3)\exp(-a_2\alpha_1) \\ a_4 &\leftarrow \alpha_3 = \texttt{YSLOPE}\,. \end{aligned} \tag{194}$$

The trial stress is computed on the basis of the total stress rather than the effective stress. Recall that the trial stress ($\boldsymbol{\sigma}_{\text{trial}}$) is computed using the relation

$$\boldsymbol{\sigma}_{\text{trial}} = \boldsymbol{\sigma}^n + \mathbf{C}^{\text{e}} : (\boldsymbol{d}^s\,\Delta t) =: \boldsymbol{\sigma}^n + \mathbf{C}^{\text{e}} : \Delta\boldsymbol{\varepsilon} \tag{195}$$

where

$$\mathbf{C}^{\text{e}} = \left(\widehat{K} - \tfrac{2}{3}\widehat{G}\right)\boldsymbol{I}\otimes\boldsymbol{I} + 2\widehat{G}\,\mathbf{I}^{(s)}\,. \tag{196}$$

Therefore

$$\boldsymbol{\sigma}_{\text{trial}} = \boldsymbol{\sigma}^n + K\text{tr}(\Delta\boldsymbol{\varepsilon})\boldsymbol{I} + 2G\big[\Delta\boldsymbol{\varepsilon} - \tfrac{1}{3}\text{tr}(\Delta\boldsymbol{\varepsilon})\boldsymbol{I}\big]\,. \tag{197}$$

To allow for nonlinear parameter variations, the algorithm breaks a trial loading step into subcycles. The algorithm below, determines the number of substeps based on the magnitude of the trial stress increment relative to the characteristic dimensions of the yield surface. Another comparison uses the value of the pressure dependent elastic properties at $\boldsymbol{\sigma}_{qs}^n$ and $\boldsymbol{\sigma}_{\text{trial}}$ and adjusts the number of substeps if there is a large change in elastic moduli. This ensures an accurate solution for nonlinear elasticity even with fully elastic loading.

The number of substeps based on the bulk modulus is given by

$$n_{\text{bulk}} = \left\lceil \frac{\|K^n - K^{\text{trial}}\|}{K^n} \right\rceil\,. \tag{198}$$

The number of substeps based on the size of the yield surface is computed as

$$n_{\text{yield}} = \left\lceil \frac{\epsilon\|\boldsymbol{\sigma}_{\text{trial}} - \boldsymbol{\sigma}_{qs}^n\|}{\ell_{\text{yield}}} \right\rceil \tag{199}$$

where ϵ is a constant, and the characteristic length of the yield surface is

$$\ell_{\text{yield}} = \text{MIN}\left(a_1, \frac{I_1^{\text{peak}} - X^n}{2}\right)\,. \tag{200}$$

The number of substeps used in the actual calculation is

$$n_{\text{sub}} = \text{MAX}\big(n_{\text{bulk}}, n_{\text{yield}}\big)\,. \tag{201}$$

The nonhardening return algorithm uses a transformed space (see Homel, Guilkey, and Brannon, (2015)) where the computation is carried out in special Lode coordinates (z_{eff}, r') where

$$z_{\text{eff}} := \frac{\text{tr}(\boldsymbol{\sigma} - \boldsymbol{\alpha})}{\sqrt{3}} \quad \text{and} \quad r' = \beta r\sqrt{\frac{3K}{2G}}\,, \quad r := \sqrt{2J_2}\,. \tag{202}$$

If the flow rule is non-associative, the yield surface parameter $\beta \neq 1$.

The closest point from the trial stress state to the yield surface in this transformed space, $(z_{\text{eff}}^{\text{close}}, r'_{\text{close}})$, is computed using a geometric algorithm. After the closest point has been found, the coordinates of the closest point are transformed back to the (z, r) coordinates and the stress state is reconstructed using

$$\boldsymbol{\sigma}^{\text{fixed}} = z^{\text{close}} \frac{\boldsymbol{I}}{\|\boldsymbol{I}\|} + r^{\text{close}} \frac{\boldsymbol{s}^{\text{trial}}}{\|\boldsymbol{s}^{\text{trial}}\|} \,. \tag{203}$$

Let the stress at the beginning of the load step be $\boldsymbol{\sigma}^k$ and let the trial stress be $\boldsymbol{\sigma}_{\text{trial}}$. Assume the yield surface is fixed and let the correct projection of the trial stress on to the fixed yield surface be $\boldsymbol{\sigma}_{\text{fixed}}^{k+1}$.

The increment of stress for the load step $(\Delta\boldsymbol{\sigma}_{\text{fixed}})$ is related to the elastic strain increment $(\Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}})$ by

$$\Delta\boldsymbol{\sigma}_{\text{fixed}} = \boldsymbol{\sigma}_{\text{fixed}}^{k+1} - \boldsymbol{\sigma}^k = \mathbf{C}_{\text{fixed}}^{\text{e}} : \Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}} \tag{204}$$

where $\mathbf{C}_{\text{fixed}}^{\text{e}}$ is a constant elastic modulus tensor. The elastic modulus tensor can be assumed to be an average value of the nonlinear tangent modulus for the load step. If we know $\mathbf{C}_{\text{fixed}}^{\text{e}}$, we can compute the elastic strain increment using

$$\Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}} = (\mathbf{C}_{\text{fixed}}^{\text{e}})^{-1} : \Delta\boldsymbol{\sigma}_{\text{fixed}} \,. \tag{205}$$

For the situation where $\mathbf{C}^{\text{e}}$ is given by equation (196), we have

$$(\mathbf{C}^{\text{e}})^{-1} = \left(\frac{1}{9K^k} - \frac{1}{6G^k}\right) \boldsymbol{I} \otimes \boldsymbol{I} + \frac{1}{2G^k} \mathbf{I}^{(s)} \,. \tag{206}$$

Therefore,

$$\begin{aligned} \Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}} &= \left(\frac{1}{9K^k} - \frac{1}{6G^k}\right) \text{tr}(\Delta\boldsymbol{\sigma}_{\text{fixed}}) \boldsymbol{I} + \frac{1}{2G^k} \Delta\boldsymbol{\sigma}_{\text{fixed}} \\ &= \frac{1}{3K^k} \left[\frac{1}{3}\text{tr}(\Delta\boldsymbol{\sigma}_{\text{fixed}}) \boldsymbol{I}\right] + \frac{1}{2G^k} \left[\Delta\boldsymbol{\sigma}_{\text{fixed}} - \frac{1}{3}\text{tr}(\Delta\boldsymbol{\sigma}_{\text{fixed}}) \boldsymbol{I}\right] \end{aligned} \tag{207}$$

or

$$\Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}} = \frac{1}{3K^k} \Delta\boldsymbol{\sigma}_{\text{fixed}}^{\text{iso}} + \frac{1}{2G^k} \Delta\boldsymbol{\sigma}_{\text{fixed}}^{\text{dev}} \,. \tag{208}$$

For a strain driven update algorithm, the total strain increment $\Delta\boldsymbol{\varepsilon}$ is known. Assuming that the total strain increment can be additively decomposed into an elastic and a plastic part, we can find the plastic strain increment $(\Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{p}})$ using

$$\Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{p}} = \Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{e}} \,. \tag{209}$$

Now, if we allow the yield surface to harden, the distance between the trial stress point and its projection on to the yield surface decreases compared to that for a fixed yield surface. If $\Delta\boldsymbol{\varepsilon}^{\text{p}}$ is the plastic strain increment for a hardening yield surface, we have

$$\Delta\boldsymbol{\varepsilon}^{\text{p}} < \Delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{p}} \tag{210}$$

where the inequality can be evaluated using an appropriate Euclidean norm. Note that this distance is proportional to the consistency parameter $\dot{\lambda}$.

In Homel's *fully saturated* version of the Arenisca model (Homel, Guilkey, and Brannon, 2015), the internal variables are the hydrostatic compressive strength (X) and the scalar isotropic backstress $(\boldsymbol{\alpha})$. These depend only on the *volumetric* plastic strain increment

$$\Delta\varepsilon_v^{\text{p}} = \text{tr}(\Delta\boldsymbol{\varepsilon}^{\text{p}}) \,. \tag{211}$$

Because $\Delta\varepsilon_v^{\text{p}} < \Delta\varepsilon_{v,\text{fixed}}^{\text{p}}$ we can define a parameter, $\eta \in (0, 1)$, such that

$$\eta := \frac{\Delta\varepsilon_v^{\text{p}}}{\Delta\varepsilon_{v,\text{fixed}}^{\text{p}}} \,. \tag{212}$$

Since the solution is bounded by the fixed yield surface, a *bisection algorithm can be used to find the parameter* η.

To use the same approach for the ARENA *partially saturated* model, the internal variables have to be of the form

$$\overline{p_{\mathrm{p}}^{w}} \equiv \overline{p_{\mathrm{p}}^{w}}(\varepsilon_v^{\mathrm{p}}) \,, \quad B^{\mathrm{p}} \equiv B^{\mathrm{p}}(\varepsilon_v^{\mathrm{p}}) \,, \quad \phi^{\mathrm{p}} \equiv \phi^{\mathrm{p}}(\varepsilon_v^{\mathrm{p}}) \,, \quad S_w^{\mathrm{p}} \equiv S_w^{\mathrm{p}}(\varepsilon_v^{\mathrm{p}}) \,, \quad \overline{X^{\mathrm{p}}} \equiv \overline{X}^{\mathrm{p}}(\varepsilon_v^{\mathrm{p}})$$

so that

$$\begin{aligned} \dot{\overline{p_{\mathrm{p}}^{w}}} &= \dot{\lambda} \frac{d\overline{p_{\mathrm{p}}^{w}}}{d\varepsilon_v^{\mathrm{p}}} \mathrm{tr}(\boldsymbol{M}) \,, \quad \dot{B}^{\mathrm{p}} = \dot{\lambda} \frac{dB^{\mathrm{p}}}{d\varepsilon_v^{\mathrm{p}}} \mathrm{tr}(\boldsymbol{M}) \,, \quad \dot{\phi}^{\mathrm{p}} = \dot{\lambda} \frac{d\phi^{\mathrm{p}}}{d\varepsilon_v^{\mathrm{p}}} \mathrm{tr}(\boldsymbol{M}) \\ \dot{S_w^{\mathrm{p}}} &= \dot{\lambda} \frac{dS_w^{\mathrm{p}}}{d\varepsilon_v^{\mathrm{p}}} \mathrm{tr}(\boldsymbol{M}) \,, \quad \dot{\overline{X^{\mathrm{p}}}} = \dot{\lambda} \frac{d\overline{X}^{\mathrm{p}}}{d\varepsilon_v^{\mathrm{p}}} \mathrm{tr}(\boldsymbol{M}) \,. \end{aligned} \tag{213}$$

Recall that

$$\dot{\overline{p_{\mathrm{p}}^{w}}} = \frac{1}{\mathcal{B}} \mathrm{tr}(\boldsymbol{d}^{\mathrm{p}}) = \frac{\dot{\lambda}}{\mathcal{B}} \mathrm{tr}(\boldsymbol{M}) \tag{214}$$

$$\dot{B}^{\mathrm{p}} = \dot{\lambda} \left[\frac{1}{\mathcal{B} K_s} \left(\frac{K_d}{K_s} \frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}} \right) \right] \mathrm{tr}(\boldsymbol{M}) \tag{215}$$

$$\dot{S}_w^{\mathrm{p}} = \dot{\lambda} \left[\frac{1 - S_0}{\mathcal{B}(1 - S_0 + \mathcal{C})^2} \frac{d\mathcal{C}}{d\overline{p^w}} \right] \mathrm{tr}(\boldsymbol{M}) \tag{216}$$

$$\dot{\phi}^{\mathrm{p}} = -\dot{\lambda} \left[(1 - S_0) \phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}\right) \left(1 + \frac{1}{K_a \mathcal{B}} \right) + S_0 \phi_0 \exp\left(\overline{\varepsilon_v} - \overline{\varepsilon_v^w}\right) \left(1 + \frac{1}{K_w \mathcal{B}} \right) \right] \mathrm{tr}(\boldsymbol{M}) \tag{217}$$

$$\dot{\overline{X^{\mathrm{p}}}} := \dot{\lambda} \left[\left[(1 - S_w) + p_1^{\mathrm{sat}} S_w \right] \frac{d\overline{X}_d}{d\overline{\varepsilon_v^{\mathrm{p}}}} + \frac{\overline{X}_d (p_1^{\mathrm{sat}} - 1)(1 - S_0)}{\mathcal{B}(1 - S_0 + \mathcal{C})^2} \frac{d\mathcal{C}}{d\overline{p^w}} + \frac{3\overline{p^w}}{\mathcal{B} K_s} \left(\frac{K_d}{K_s} \frac{dK_d}{d\overline{p^w}} - \frac{dK_s}{d\overline{p^w}} \right) + \frac{3B}{\mathcal{B}} \right] \mathrm{tr}(\boldsymbol{M}) \tag{218}$$

where

$$\begin{aligned} \mathcal{B} := & \frac{1}{(1 - S_0)\exp[\overline{\varepsilon_v} - \overline{\varepsilon_v^a}(\overline{p^w})] + S_0 \exp[\overline{\varepsilon_v} - \overline{\varepsilon_v^w}(\overline{p^w})]} \left[-\frac{(B - \phi)\phi}{B\phi_0} \left(\frac{S_w}{K_w} + \frac{1 - S_w}{K_a} \right) + \right. \\ & \left. \frac{1 - S_0}{K_a} \exp[\overline{\varepsilon_v} - \overline{\varepsilon_v^a}(\overline{p^w})] + \frac{S_0}{K_w} \exp[\overline{\varepsilon_v} - \overline{\varepsilon_v^w}(\overline{p^w})] \right] \\ \mathcal{C} := & S_0 \exp\left[\overline{\varepsilon_v^a}(\overline{p^w}) - \overline{\varepsilon_v^w}(\overline{p^w}) \right] . \end{aligned} \tag{219}$$

The evolution equations for the internal variables are consistent with Homel's assumptions (allowing for the use of a bisection algorithm) only if

$$\overline{p^w} = \overline{p_{\mathrm{p}}^{w}} \,, \quad B = B^{\mathrm{p}} \,, \quad \phi = \phi^{\mathrm{p}} \,, \quad S_w = S_w^{\mathrm{p}} \,, \quad \overline{X} = \overline{X}^{\mathrm{p}} \,. \tag{220}$$

This is a strong assumption and may violate the balance of mass. However, to keep the algorithm efficient, we make the assumptions in (220) and hence can use the bisection algorithm discussed later.

Now we can compare equations (213) and (214), and use the assumption that $B \approx 1$, to get the following

set of coupled ordinary differential equations for the internal variables.

$$\begin{aligned}
\frac{d\overline{p_{\mathrm{p}}^{w}}}{d\varepsilon_{v}^{\mathrm{p}}} &= \frac{1}{\mathcal{B}^{\mathrm{p}}} \\
\frac{d\phi^{\mathrm{p}}}{d\varepsilon_{v}^{\mathrm{p}}} &= (S_0-1)\phi_0 \exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}})]\left(1+\frac{1}{K_a(\overline{p_{\mathrm{p}}^{w}})\mathcal{B}^{\mathrm{p}}}\right) - S_0\phi_0 \exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})]\left(1+\frac{1}{K_w(\overline{p_{\mathrm{p}}^{w}})\mathcal{B}^{\mathrm{p}}}\right) \\
\frac{dS_{w}^{\mathrm{p}}}{d\varepsilon_{v}^{\mathrm{p}}} &= \frac{1-S_0}{\mathcal{B}^{\mathrm{p}}(1-S_0+\mathcal{C}^{\mathrm{p}})^2}\frac{d\mathcal{C}^{\mathrm{p}}}{d\overline{p_{\mathrm{p}}^{w}}} \\
\frac{d\overline{X}^{\mathrm{p}}}{d\varepsilon_{v}^{\mathrm{p}}} &= \left[(1-S_{w}^{\mathrm{p}}) + p_{1}^{\mathrm{sat}}S_{w}^{\mathrm{p}}\right]\frac{d\overline{X}_d}{d\overline{\varepsilon_{v}^{\mathrm{p}}}} + \frac{\overline{X}_d(p_{1}^{\mathrm{sat}}-1)(1-S_0)}{\mathcal{B}^{\mathrm{p}}(1-S_0+\mathcal{C}^{\mathrm{p}})^2}\frac{d\mathcal{C}^{\mathrm{p}}}{d\overline{p_{\mathrm{p}}^{w}}} + \frac{3}{\mathcal{B}^{\mathrm{p}}}
\end{aligned} \tag{221}$$

where

$$\begin{aligned}
\mathcal{B}^{\mathrm{p}} &:= \frac{1}{(1-S_0)\exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}})] + S_0\exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})]}\left[-\frac{(1-\phi^{\mathrm{p}})\phi^{\mathrm{p}}}{\phi_0}\left(\frac{S_{w}^{\mathrm{p}}}{K_w(\overline{p_{\mathrm{p}}^{w}})} + \frac{1-S_{w}^{\mathrm{p}}}{K_a(\overline{p_{\mathrm{p}}^{w}})}\right) + \right. \\
&\qquad \left. \frac{1-S_0}{K_a(\overline{p_{\mathrm{p}}^{w}})}\exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}})] + \frac{S_0}{K_w(\overline{p_{\mathrm{p}}^{w}})}\exp[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})]\right] \\
\mathcal{C}^{\mathrm{p}} &:= S_0 \exp\left[\overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}}) - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})\right], \quad \frac{d\mathcal{C}^{\mathrm{p}}}{d\overline{p_{\mathrm{p}}^{w}}} = \mathcal{C}^{\mathrm{p}}\left[\frac{1}{K_a(\overline{p_{\mathrm{p}}^{w}})} - \frac{1}{K_w(\overline{p_{\mathrm{p}}^{w}})}\right].
\end{aligned} \tag{222}$$

If we assume that the Biot coefficient is equal to 1, the vector of internal variables is $\boldsymbol{\eta} := \{\overline{p_{\mathrm{p}}^{w}}, \phi^{\mathrm{p}}, S_{w}^{\mathrm{p}}, \overline{X}^{\mathrm{p}}\}$, we can express the evolution of the internal variables as

$$\frac{d\boldsymbol{\eta}}{d\varepsilon_{v}^{\mathrm{p}}} = \mathbf{F}(\boldsymbol{\eta}, \varepsilon_{v}^{\mathrm{p}}). \tag{223}$$

We can use a forward Euler integration approach to update the internal variables:

$$\boldsymbol{\eta}^{n+1} = \boldsymbol{\eta}^{n} + \mathbf{F}[\boldsymbol{\eta}^{n}, (\varepsilon_{v}^{\mathrm{p}})^{n}]\Delta\varepsilon_{v}^{\mathrm{p}}. \tag{224}$$

However, since the hydrostatic compressive strength, porosity, and saturation can be computed in closed form, we use the following in our implementation

$$\begin{aligned}
\phi^{\mathrm{p}}(\overline{p_{\mathrm{p}}^{w}}) &= (1-S_0)\phi_0 \exp\left[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}})\right] + S_0\phi_0 \exp\left[\overline{\varepsilon_{v}^{\mathrm{p}}} - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})\right] \\
S_{w}^{\mathrm{p}}(\overline{p_{\mathrm{p}}^{w}}) &= \frac{\mathcal{C}^{\mathrm{p}}}{1-S_0+\mathcal{C}^{\mathrm{p}}}, \quad \mathcal{C}^{\mathrm{p}} := S_0 \exp\left[\overline{\varepsilon_{v}^{a}}(\overline{p_{\mathrm{p}}^{w}}) - \overline{\varepsilon_{v}^{w}}(\overline{p_{\mathrm{p}}^{w}})\right].
\end{aligned} \tag{225}$$

However, since the hydrostatic compressive strength, porosity, and saturation can be computed in closed form, we integrate only the expression for pore pressure evolution in our implementation.

2.8 Rate-dependent plasticity

Rate-dependent plastic effects are determined after the rate-independent solution has been found. An additive decomposition of the (unrotated) rate of deformation into elastic and viscoplastic parts is assumed, i.e.,

$$\boldsymbol{d}^s = \boldsymbol{d}^{\mathrm{e}} + \boldsymbol{d}^{\mathrm{vp}} . \tag{226}$$

The Perzyna formulation for the viscoplastic rate of deformation is used (Brannon, 2007):

$$\boldsymbol{d}^{\mathrm{vp}} = \left\langle \frac{f(\boldsymbol{\sigma},\boldsymbol{\eta})}{\tau} \right\rangle = \begin{cases} \dfrac{f(\boldsymbol{\sigma},\boldsymbol{\eta})}{\tau}\boldsymbol{M} & \text{if } f(\boldsymbol{\sigma},\boldsymbol{\eta}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{227}$$

where $f(\dots)$ is the yield function and τ is a relaxation time. Note that the above relation can be expressed in Duvaut-Lion form as

$$\boldsymbol{d}^{\mathrm{vp}} = \begin{cases} (\mathbf{C}^{\mathrm{e}})^{-1} : \left(\dfrac{\boldsymbol{\sigma} - \mathcal{P}\boldsymbol{\sigma}}{\tau} \right) & \text{if } f(\boldsymbol{\sigma},\boldsymbol{\eta}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{228}$$

where $\mathbf{C}^{\mathrm{e}}$ is the elastic stiffness tensor and $\mathcal{P}\boldsymbol{\sigma}$ is the closest point projection of the stress state on to the boundary of the region that bounds all possible elastic stress states. Clearly, $\mathcal{P}\boldsymbol{\sigma} = \boldsymbol{\sigma}_{qs}$, and for stress states outside the yield surface we have

$$\boldsymbol{d}^{\mathrm{vp}} = \frac{1}{\tau}\,(\mathbf{C}^{\mathrm{e}})^{-1} : \left(\boldsymbol{\sigma} - \boldsymbol{\sigma}_{qs}\right) . \tag{229}$$

Recall from equation (100) that

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}^{\mathrm{e}} : \boldsymbol{d}^{\mathrm{e}} - \dot{\lambda}\boldsymbol{Z} . \tag{230}$$

If we ignore the $\boldsymbol{Z}$ term and use equations (156) and (229), we have

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}^{\mathrm{e}} : (\boldsymbol{d}^s - \boldsymbol{d}^{\mathrm{vp}}) = \mathbf{C}^{\mathrm{e}} : \boldsymbol{d}^s - \frac{1}{\tau}\,(\boldsymbol{\sigma} - \boldsymbol{\sigma}_{qs}) = \dot{\boldsymbol{\sigma}}^{\mathrm{trial}} - \frac{1}{\tau}\,(\boldsymbol{\sigma} - \boldsymbol{\sigma}_{qs}) . \tag{231}$$

We can subtract the rate of the quasistatic stress from both sides to get an equation for the viscoplastic overstress contribution ($\boldsymbol{\sigma}_{\mathrm{over}} := \boldsymbol{\sigma} - \boldsymbol{\sigma}_{qs}$):

$$\dot{\boldsymbol{\sigma}} - \dot{\boldsymbol{\sigma}}_{qs} = \dot{\boldsymbol{\sigma}}^{\mathrm{trial}} - \dot{\boldsymbol{\sigma}}_{qs} - \frac{1}{\tau}\,(\boldsymbol{\sigma} - \boldsymbol{\sigma}_{qs}) \tag{232}$$

or

$$\dot{\boldsymbol{\sigma}}_{\mathrm{over}} = \left(\dot{\boldsymbol{\sigma}}^{\mathrm{trial}} - \dot{\boldsymbol{\sigma}}_{qs}\right) - \frac{1}{\tau}\,\boldsymbol{\sigma}_{\mathrm{over}} . \tag{233}$$

From equation (158) we have

$$\dot{\boldsymbol{\sigma}}_{qs} = \dot{\boldsymbol{\sigma}}_{\mathrm{trial}} - \frac{\boldsymbol{P} \otimes \widehat{\boldsymbol{N}}}{\widehat{\boldsymbol{N}} : \boldsymbol{P} + H} : \dot{\boldsymbol{\sigma}}_{\mathrm{trial}} . \tag{234}$$

Therefore, (233) can be written as

$$\dot{\boldsymbol{\sigma}}_{\mathrm{over}} = \frac{\boldsymbol{P} \otimes \widehat{\boldsymbol{N}}}{\widehat{\boldsymbol{N}} : \boldsymbol{P} + H} : \dot{\boldsymbol{\sigma}}_{\mathrm{trial}} - \frac{1}{\tau}\,\boldsymbol{\sigma}_{\mathrm{over}} =: \dot{\boldsymbol{\sigma}}_{\mathrm{proj}} - \frac{1}{\tau}\,\boldsymbol{\sigma}_{\mathrm{over}} \tag{235}$$

where

$$\dot{\boldsymbol{\sigma}}_{\mathrm{proj}} := \dot{\boldsymbol{\sigma}}^{\mathrm{trial}} - \dot{\boldsymbol{\sigma}}_{qs} = \frac{\boldsymbol{P} \otimes \widehat{\boldsymbol{N}}}{\widehat{\boldsymbol{N}} : \boldsymbol{P} + H} : \dot{\boldsymbol{\sigma}}_{\mathrm{trial}} . \tag{236}$$

The above ordinary differential equation has the solution

$$\boldsymbol{\sigma}_{\mathrm{over}}(t) = \frac{\int \exp\left[\int^t \frac{1}{\tau(t')}\,dt'\right] \dot{\boldsymbol{\sigma}}_{\mathrm{proj}}(t)\,dt + C}{\exp\left[\int^t \frac{1}{\tau(t')}\,dt'\right]} \tag{237}$$

where C is a constant. If we assume that τ is constant over a timestep, we have

$$\boldsymbol{\sigma}_{\text{over}}(t) = \frac{\int \exp\left[\frac{t}{\tau}\right] \dot{\boldsymbol{\sigma}}_{\text{proj}}(t)dt + C}{\exp\left[\frac{t}{\tau}\right]} \tag{238}$$

In addition, if we assume that $\dot{\boldsymbol{\sigma}}_{\text{proj}}$ is also constant over a timestep, we have

$$\boldsymbol{\sigma}_{\text{over}}(t) = \frac{\tau \exp\left[\frac{t}{\tau}\right] \dot{\boldsymbol{\sigma}}_{\text{proj}} + C}{\exp\left[\frac{t}{\tau}\right]}. \tag{239}$$

If the initial overstress at $t = t_0$ is $\boldsymbol{\sigma}_{\text{over}}(t_0)$, we have

$$C = \exp\left[\frac{t_0}{\tau}\right] \boldsymbol{\sigma}_{\text{over}}(t_0) - \tau \exp\left[\frac{t_0}{\tau}\right] \dot{\boldsymbol{\sigma}}_{\text{proj}}\,. \tag{240}$$

Therefore

$$\boldsymbol{\sigma}_{\text{over}}(t) = \tau\left[1 - \exp\left(\frac{t_0 - t}{\tau}\right)\right] \dot{\boldsymbol{\sigma}}_{\text{proj}} + \exp\left(\frac{t_0 - t}{\tau}\right) \boldsymbol{\sigma}_{\text{over}}(t_0)\,. \tag{241}$$

The model used for the material relaxation time in Arensica is

$$\tau = T_1\, \dot{\varepsilon}^{-T_2} \quad \text{where} \quad \dot{\varepsilon} = \|\boldsymbol{d}^s\| \tag{242}$$

and T_1, T_2 are material parameters.

Recall from (233) and (236) that

$$\dot{\boldsymbol{\sigma}}_{\text{over}} = \dot{\boldsymbol{\sigma}}_{\text{proj}} - \frac{1}{\tau}\boldsymbol{\sigma}_{\text{over}} \quad \text{where} \quad \dot{\boldsymbol{\sigma}}_{\text{over}} := \dot{\boldsymbol{\sigma}} - \dot{\boldsymbol{\sigma}}_{qs} \quad \text{and} \quad \dot{\boldsymbol{\sigma}}_{\text{proj}} := \dot{\boldsymbol{\sigma}}^{\text{trial}} - \dot{\boldsymbol{\sigma}}_{qs}\,. \tag{243}$$

and the exact solution (241) when τ and $\dot{\boldsymbol{\sigma}}_{\text{proj}}$ are constant:

$$\boldsymbol{\sigma}_{\text{over}}(t) = \tau\left[1 - \exp\left(\frac{t_0 - t}{\tau}\right)\right] \dot{\boldsymbol{\sigma}}_{\text{proj}} + \exp\left(\frac{t_0 - t}{\tau}\right) \boldsymbol{\sigma}_{\text{over}}(t_0)\,. \tag{244}$$

For a timestep $\Delta t = t_{n+1} - t_n$, we can write the above as

$$\boldsymbol{\sigma}_{\text{over}}(t_{n+1}) = \tau(t_n)\left[1 - \exp\left(\frac{t_n - t_{n+1}}{\tau(t_n)}\right)\right] \dot{\boldsymbol{\sigma}}_{\text{proj}}(t_n) + \exp\left(\frac{t_n - t_{n+1}}{\tau(t_n)}\right) \boldsymbol{\sigma}_{\text{over}}(t_n) \tag{245}$$

or

$$\boldsymbol{\sigma}^{n+1}_{\text{over}} = \left[\frac{1 - \exp\left(-\Delta t/\tau^n\right)}{\Delta t/\tau^n}\right] \left(\Delta t\, \dot{\boldsymbol{\sigma}}^n_{\text{proj}}\right) + \exp\left(-\Delta t/\tau^n\right) \boldsymbol{\sigma}^n_{\text{over}}\,. \tag{246}$$

If we approximate the rate of $\boldsymbol{\sigma}_{\text{proj}}$ via approximations of the rates of the trial and quasistatic stresses using

$$\dot{\boldsymbol{\sigma}}_{\text{proj}} \approx \frac{\boldsymbol{\sigma}^{\text{trial}}_{n+1} - \boldsymbol{\sigma}^{\text{trial}}_{n}}{\Delta t} - \frac{\boldsymbol{\sigma}^{n+1}_{qs} - \boldsymbol{\sigma}^{n}_{qs}}{\Delta t} \tag{247}$$

noting that

$$\boldsymbol{\sigma}^{\text{trial}}_{n} = \boldsymbol{\sigma}^n, \quad \boldsymbol{\sigma}^{\text{trial}}_{n+1} = \boldsymbol{\sigma}^n + (\mathbf{C}^{\text{e}} : \boldsymbol{d}^s)\Delta t\,, \tag{248}$$

and using the definition of $\boldsymbol{\sigma}_{\text{over}}$, we have

$$\boldsymbol{\sigma}^{n+1} - \boldsymbol{\sigma}^{n+1}_{qs} = \left[\frac{1 - \exp\left(-\Delta t/\tau^n\right)}{\Delta t/\tau^n}\right] \left[\left(\boldsymbol{\sigma}^{\text{trial}}_{n+1} - \boldsymbol{\sigma}^n\right) - \left(\boldsymbol{\sigma}^{n+1}_{qs} - \boldsymbol{\sigma}^n_{qs}\right)\right] + \exp\left(-\Delta t/\tau^n\right)\left(\boldsymbol{\sigma}^n - \boldsymbol{\sigma}^n_{qs}\right). \tag{249}$$

If we define

$$R_H := \frac{1 - \exp\left(-\Delta t/\tau^n\right)}{\Delta t/\tau^n} \quad \text{and} \quad r_h := \exp\left(-\Delta t/\tau^n\right) \tag{250}$$

we have the dynamic stress state

$$\boldsymbol{\sigma}^{n+1} = \boldsymbol{\sigma}_{qs}^{n+1} + \left[\left(\boldsymbol{\sigma}_{n+1}^{\text{trial}} - \boldsymbol{\sigma}^{n} \right) - \left(\boldsymbol{\sigma}_{qs}^{n+1} - \boldsymbol{\sigma}_{qs}^{n} \right) \right] R_H + \left(\boldsymbol{\sigma}^{n} - \boldsymbol{\sigma}_{qs}^{n} \right) r_h \,. \tag{251}$$

Our implementation does not consider rate-dependent updates of the internal variables. Equation 251 requires that we compute the trial stress for the step, but this is not known since the bulk modulus is evolving through the substeps. It would be necessary to to loop through the substeps to compute the trial stress assuming nonlinear elasticity, but instead we approximate the trial stress using the average of the elastic moduli at the start and end of the step.

3 Decompositions of deformation rates

A decomposition of the deformation gradient $\mathbf{F}$ into the composition of elastic and plastic parts,

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p , \tag{252}$$

is often hypothesized in finite-deformation plasticity, giving the rate form

$$\dot{\mathbf{F}} = \dot{\mathbf{F}}^e \cdot \mathbf{F}^p + \mathbf{F}^e \cdot \dot{\mathbf{F}}^p . \tag{253}$$

Therefore, the velocity gradient, $\mathbf{L} := \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$, decomposes additively as

$$\mathbf{L} = \mathbf{L}^e + \mathbf{L}^p , \tag{254}$$

where

$$\mathbf{L}^e := \dot{\mathbf{F}}^e \cdot \mathbf{F}^{e-1} \tag{255}$$

and

$$\mathbf{L}^p := \mathbf{F}^e \cdot (\dot{\mathbf{F}}^p \cdot \mathbf{F}^{p-1}) \cdot \mathbf{F}^{e-1} . \tag{256}$$

The approximation $\mathbf{L}^p \approx \dot{\mathbf{F}}^p \cdot \mathbf{F}^{p-1}$ might apply if it can be argued that elastic strains are small (so that $\mathbf{F}^e \approx \mathbf{I}$). Another line of reasoning (Itskov, 2004) pertains to the degree to which reference stretch directions rotate through the course of a simulation. If reference stretch directions of $\mathbf{F}$ are stationary, then (at least for initially isotropic media) it is reasonable to expect the same of the reference stretches of $\mathbf{F}^e$ and $\mathbf{F}^p$ so that the finite deformation decomposition becomes

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U}^e \cdot \mathbf{U}^p . \tag{257}$$

Furthermore, under these conditions, the reference stretch, $\mathbf{U}$, and its parts, $\mathbf{U}^e$ and $\mathbf{U}^p$, should be expected to share a common set of eigenvectors and hence commute, thus leading to no difference between the multiplicative and additive decompositions for this class of problem. An inexpensive run-time check of the degree to which this approximation applies requires monitoring the value of

$$\|\mathbf{Z}\| := \left\| \overline{\mathbf{D}} \cdot \mathbf{U} - \mathbf{U} \cdot \overline{\mathbf{D}} \right\| , \tag{258}$$

where $\overline{\mathbf{D}}$ is the so-called "unrotated rate of deformation," given by a polar-rotation pull-back operation on the symmetric part of the velocity gradient, $\overline{\mathbf{D}} := \mathbf{R}^T \cdot \mathrm{sym}\mathbf{L} \cdot \mathbf{R}$. We assert that rate of rotation of the reference stretch directions is zero if and only if the "Z-norm" is zero. Hence, choosing to adopt an additive strain rate (as well as using $\overline{\mathbf{D}}$ as an approximation of the rate of the Hencky strain) can be rigorously justified if it can be shown, as in Fig. 2, that a simulation of interest has a small Z-norm during important phases of material deformation. Otherwise, if this is not the case, then designers of a multiplicative model must demonstrate its undeniable superiority in comparison to additive decompositions in problems having rotation of the reference stretch directions. The models must, of course, be tuned to agree exactly with each other for model calibration scenarios (such as dynamic plate compaction) having stationary reference stretch directions.

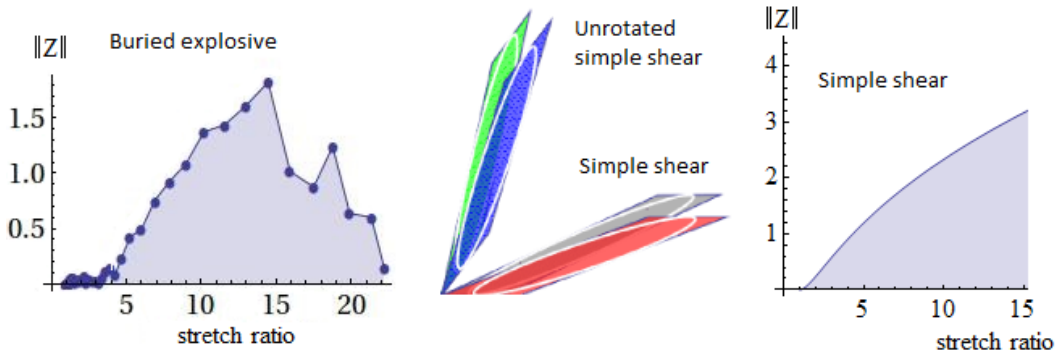


Figure 2 – *Up until a stretch ratio of ≈5 in a buried explosive simulation (well beyond the state for which anything but crude accuracy could be expected of any granular constitutive model), the small values of a run-time indicator,* $\|\mathbf{Z}\| = \|\mathbf{DV} - \mathbf{VD}\|$ *demonstrate that there is very little rotation of reference stretch directions. Values in simple shear are shown for comparison. The stretch ellipsoids in shear further indicate qualitative differences between the effective deformed states in a model that approximates Hencky strain to be the time integral of the symmetric part of the velocity gradient. The close proximity of these ellipsoids visually hint that model prediction errors are most likely rooted in simplified model physics, not errors associated with convenient approximations in the strain definitions.*

4 The MPM implementation used by ARENA

The spatial momentum equation is solved using the MPM algorithm while forward Euler time-stepping is use to integrate time derivatives. The pseudocode of the overall algorithm is given below. The main quantities of interest are:

- $t_{\max}$: The maximum time until which the simulation is to run.
- $t, \Delta t$: The current time ($t = t_n$) and the time step.
- $\mathbf{h}_g$: The grid spacing vector.
- m_p : The particle mass.
- V_p^n, V_p^{n+1} : The particle volume at $t = t_n$ and $t = t_{n+1}$.
- $\mathbf{x}_p^n, \mathbf{x}_p^{n+1}$: The particle position at $t = t_n$ and $t = t_{n+1}$.
- $\mathbf{u}_p^n, \mathbf{u}_p^{n+1}$: The particle displacement at $t = t_n$ and $t = t_{n+1}$.
- $\mathbf{v}_p^n, \mathbf{v}_p^{n+1}$: The particle velocity at $t = t_n$ and $t = t_{n+1}$.
- $\boldsymbol{\sigma}_p^n, \boldsymbol{\sigma}_p^{n+1}$: The particle Cauchy stress at time $t = t_n$ and $t = t_{n+1}$.
- $\boldsymbol{F}_p^n, \boldsymbol{F}_p^{n+1}$: The particle deformation gradient at time $t = t_n$ and $t = t_{n+1}$.

Algorithm 1 The algorithm

procedure RUN(inputUPSFile)
  $t_{\max}$, $\mathbf{h}_g$, `xmlProblemSpec`, `grid`, `globalState` ← READINPUTUPSFILE(inputUPSFile) ▷*Parse the input XML file (<filename>.ups), create the background grid, and set up a SIMULATIONSTATE.*
  `mpmFlags`, `prescribedDefGrad`, `particleBC`, `contactModel`, `constitutiveModel`, `defGradComputer`, `damageModel` ← PROBLEMSETUP(`xmlProblemSpec`, `grid`, `globalState`) ▷*Set up flags, the constitutive model, and the deformation gradient algorithm based on data in input file.*
  $t \leftarrow 0, n \leftarrow 0$
  $\mathbf{x}_p^n, \mathbf{u}_p^n, m_p, V_p^n, \mathbf{v}_p^n, \boldsymbol{\sigma}_p^n, \boldsymbol{F}_p^n$ ← INITIALIZE(`xmlProblemSpec`) ▷*Find the grid size and initialize particle variables based on geometry and other information in the input file.*
  isSuccess ← FALSE
  repeat
    $\Delta t \leftarrow$ COMPUTESTABLETIMESTEP($\mathbf{h}_g, \mathbf{v}_p$) ▷*Find a stable time increment based on grid size and velocity*
    $t \leftarrow t + \Delta t, n \leftarrow n + 1$ ▷*Update the time*
    isSuccess, $\mathbf{x}_p^{n+1}, \mathbf{u}_p^{n+1}, v_p^{n+1}, \mathbf{v}_p^{n+1}, \boldsymbol{\sigma}_p^{n+1}, \boldsymbol{F}_p^{n+1}$ ← TIMEADVANCE($\mathbf{h}_g, \mathbf{x}_p^n, \mathbf{u}_p^n, m_p, V_p^n, \mathbf{v}_p^n, \boldsymbol{\sigma}_p^n, \boldsymbol{F}_p^n$) ▷*Compute updated quantities*
    OUTPUTDATA($\mathbf{x}_p^{n+1}, \mathbf{u}_p^{n+1}, V_p^{n+1}, \mathbf{v}_p^{n+1}, \boldsymbol{\sigma}_p^{n+1}, \boldsymbol{F}_p^{n+1}$) ▷*Save the solution*
    $n \leftarrow n + 1$
  until $t \geq t_{\max}$
  return isSuccess
end procedure

4.1 Reading the input file

The process used to read the input file is identical to that given in the Uintah Developer Manual and will not be discussed here.

4.2 Problem setup

The overall structure of the problem setup code is given below. Details can be found in the code.

Algorithm 2 Problem setup

Require: `xmlProblemSpec`, `grid`, `globalState`

procedure PROBLEMSETUP(`xmlProblemSpec`, `grid`, `globalState`)
  `flags` ← READMPMFLAGS(`xmlProblemSpec`) ▷*Read the option flags that determine the details of the MPM algorithm to be used in the simulation.*
  if `flags`.prescribeDeformation = TRUE **then**
    `prescribedDefGrad` ← READPRESCRIBEDDEFORMATIONS(`flags`.prescribedFileName)
  end if
  `particleBC` ← CREATEMPMPHYSICALBC(`xmlProblemSpec`, `grid`, `flags`) ▷*Create the model used to apply pressures and forces directly to particles.*
  `contactModel` ← CREATECONTACTMODEL(`xmlProblemSpec`, `grid`, `flags`, `globalState`) ▷*Create the contact algorithm model used to compute interactions between objects.*
  `constitutiveModel` ← CREATECONSTITUTIVEMODELS(`xmlProblemSpec`, `grid`, `flags`, `globalState`) ▷*Create the constitutive models that are needed for the simulation.*
  `defGradComputer` ← CREATEDEFORMATIONGRADIENTCOMPUTER(`flags`, `globalState`) ▷*Create the model that will be used to compute velocity and deformation gradients.*
  if `flags`.doBasicDamage = TRUE **then**
    `damageModel` ← CREATEBASICDAMAGEMODEL(`flags`, `globalState`)
  end if
  return `flags`, `prescribedDefGrad`, `particleBC`, `contactModel`, `constitutiveModel`, `damageModel`, `defGradComputer`
end procedure

4.3 Initialization

An outline of the initialization process is described below. Specific details have been discussed in earlier reports. The new quantities introduced in this section are

- n_p : The number of particles used to discretize a body.
- $\mathbf{b}_p^n, \mathbf{b}_p^{n+1}$: The particle body force acceleration at $t = t_n$ and $t = t_{n+1}$.
- D_p^n, D_p^{n+1} : The particle damage parameter at $t = t_n$ and $t = t_{n+1}$.
- $\mathbf{f}_p^{\text{ext},n}, \mathbf{f}_p^{\text{ext},n+1}$: The particle external force at $t = t_n$ and $t = t_{n+1}$.

Algorithm 3 Initialization

Require: `xmlProblemSpec`, `defGradComputer`, `constitutiveModel`, `damageModel`, `particleBC`, `mpmFlags` `materialList`,

procedure INITIALIZE
  for `matl` **in** `materialList` **do**
    n_p[`matl`], $\mathbf{x}_p^0$[`matl`], $\mathbf{u}_p^0$[`matl`], m_p[`matl`], V_p^0[`matl`], $\mathbf{v}_p^0$[`matl`], $\mathbf{b}_p^0$[`matl`], $\mathbf{f}_p^{\text{ext},0}$[`matl`] ← `matl`.CREATEPARTICLES()
    $\boldsymbol{F}_p^0$[`matl`] ← `defGradComputer`.INITIALIZE(`matl`)
    $\boldsymbol{\sigma}_p^0$[`matl`] ← `constitutiveModel`.INITIALIZE(`matl`)
    D_p^0[`matl`] ← `damageModel`.INITIALIZE(`matl`)
  end for
  if `mpmFlags`.initializeStressWithBodyForce = TRUE **then**
    $\mathbf{b}_p^0$ ← INITIALIZEBODYFORCE()
    $\boldsymbol{\sigma}_p^0, \boldsymbol{F}_p^0$ ← INITIALIZESTRESSANDDEFGRADFROMBODYFORCE()

end if
if mpmFlags.applyParticleBCs = TRUE **then**
$\mathbf{f}_p^{\text{ext},0} \leftarrow$ particleBC.initializePressureBCs()
end if
return $n_p, \mathbf{x}_p^0, \mathbf{u}_p^0, m_p, V_p^0, \mathbf{v}_p^0, \mathbf{b}_p^0, \mathbf{f}_p^{\text{ext},0}, \boldsymbol{F}_p^0, \boldsymbol{\sigma}_p^0, D_p^0$
end procedure

4.4 Time advance

The operations performed during a timestep are shown in the pseudocode below.

Algorithm 4 The MPM time advance algorithm

procedure timeAdvance($\mathbf{h}_g, x_p^n, u_p^n, m_p, V_p^n, \mathbf{v}_p^n, \mathbf{f}_p^{\text{ext},n}, \boldsymbol{d}_p^n$)
$\mathbf{b}_p^n \leftarrow$ computeParticleBodyForce() ▷*Compute the body force term*
$\mathbf{f}_p^{\text{ext},n+1} \leftarrow$ applyExternalLoads() ▷*Apply external loads to the particles*
$m_g, V_g, \mathbf{v}_g, \mathbf{b}_g, \mathbf{f}_g^{\text{ext}} \leftarrow$ interpolateParticlesToGrid() ▷*Interpolate particle data to the grid*
exchangeMomentumInterpolated() ▷*Exchange momentum between bodies on grid.*
↪ *Not discussed in this report.*
$\mathbf{f}_g^{\text{int}}, \boldsymbol{\sigma}_g, \mathbf{v}_g \leftarrow$ computeInternalForce() ▷*Compute the internal force at the grid nodes*
$\mathbf{v}_g^\star, \mathbf{a}_g \leftarrow$ computeAndIntegrateAcceleration() ▷*Compute the grid velocity*
↪ *and grid acceleration*
exchangeMomentumIntegrated() ▷*Exchange momentum between bodies on grid*
↪ *using integrated values. Not discussed in this report.*
$\mathbf{v}_g^\star, \mathbf{a}_g \leftarrow$ setGridBoundaryConditions() ▷*Update the grid velocity and grid*
↪ *acceleration using the BCs*
$\boldsymbol{l}_p^n, \boldsymbol{F}_p^{n+1}, V_p^{n+1} \leftarrow$ computeDeformationGradient() ▷*Compute the velocity gradient*
↪ *and the deformation gradient*
$\boldsymbol{\sigma}_p^{n+1}, \boldsymbol{\eta}_p^{n+1} \leftarrow$ computeStressTensor() ▷*Compute the updated stress and*
↪ *internal variables (if any)*
$\boldsymbol{\sigma}_p^{n+1}, \boldsymbol{\eta}_p^{n+1}, \chi_p^{n+1}, D_p^{n+1} \leftarrow$ computeBasicDamage() ▷*Compute the damage parameter*
↪ *and update the stress and internal variables*
$\chi_p^{n+1}, D_p^{n+1} \leftarrow$ updateErosionParameter() ▷*Update the indicator variable that is used*
↪ *to delete particles at the end of a time step*
$V_p^{n+1}, \mathbf{u}_p^{n+1}, \mathbf{v}_p^{n+1}, \mathbf{x}_p^{n+1}, m_p, \mathbf{h}_p^{n+1} \leftarrow$ interpolateToParticlesAndUpdate() ▷*Update the*
↪ *particle variables after interpolating grid quantities to particles*
end procedure

The algorithms used for the above operations are discussed next.

4.4.1 Computing the body force

The body force consists of a gravitational term and, optionally, centrifugal and coriolis terms that are needed for simulations inside a rotating frame such as a centrifuge.

Algorithm 5 Computing the body force on particles

Require: $\mathbf{x}_p^n, \mathbf{v}_p^n$, materialList, particleList, mpmFlags
procedure computeParticleBodyForce
for matl **in** materialList **do**
if mpmFlags.rotatingCoordSystem = TRUE **then**
$\mathbf{g} \leftarrow$ mpmFlags.gravityAcceleration
$\mathbf{b}_p^n$[matl] $\leftarrow \mathbf{g}$

```
        else
            for part in particleList do
                g ← mpmFlags.gravityAcceleration
                x_rc ← mpmFlags.coordRotationCenter
                z_r ← mpmFlags.coordRotationAxis
                w ← mpmFlags.coordRotationSpeed
                ω ← w z_r ▷Compute angular velocity vector
                a_corolis ← 2ω × v_p^n[matl, part] ▷Compute Coriolis acceleration
                r ← x_p^n[matl, part] − x_rc
                a_centrifugal ← ω × ω × r ▷Compute the centrifugal body force acceleration
                b_p^n[matl,part] ← g − a_centrifugal − a_coriolis ▷Compute the body force acceleration
            end for
        end if
    end for
    return b_p^n
end procedure
```

4.4.2 Applying external loads

Note that the updated deformation gradient has not been computed yet at this stage and the particle force is applied based on the deformation gradient at the beginning of the timestep. The new quantities introduced in this section are:

- $\mathbf{h}_p^n$: The particle size matrix at time $t = t_n$.

Algorithm 6 Applying external loads to particles

```
Require: t_{n+1}, x_p^n, h_p^n, u_p^n, f_p^{ext,n}, F_p^n, materialList, particleList, mpmFlags, particleBC
procedure APPLYEXTERNALLOADS
    f_p ← 0
    if mpmFlags.useLoadCurves = TRUE then
        f_p ← particleBC.COMPUTEFORCEPERPARTICLE(t^{n+1}) ▷Compute the force per particle
          ↪ due to the applied pressure
    end if
    for matl in materialList do
        if mpmFlags.useLoadCurves = TRUE then
            for part in particleList do
                f_p^{ext,n+1}[matl,part] ← particleBC.GETFORCEVECTOR(t_{n+1}, x_p^n, h_p^n, u_p^n,
                    ↪ f_p, F_p^n) ▷Compute the applied force vector at each particle
            end for
        else
            f_p^{ext,n+1}[matl] ← f_p^{ext,n}[matl]
        end if
    end for
    return f_p^{ext,n+1}
end procedure
```

4.4.3 Interpolating particles to grid

The grid quantities computed during this procedure and not stored for the next timestep except for the purpose of visualization. The new quantities introduced in this section are

- m_g : The mass at a grid node.

- V_g : The volume at a grid node.
- $\mathbf{v}_g$: The velocity at a grid node.
- $\mathbf{f}_g^{\text{ext}}$: The external force at a grid node.
- $\mathbf{b}_g$: The body force at a grid node.

Algorithm 7 Interpolating particle data to background grid

Require: m_p, V_p^n, $\mathbf{x}_p^n$, $\mathbf{h}_p^n$, $\mathbf{b}_p^n$, $\mathbf{f}_p^{\text{ext},n+1}$, $\boldsymbol{F}_p^n$, materialList, particleList, gridNodeList mpmFlags, particleBC

procedure INTERPOLATEPARTICLESTOGRID
  interpolator ← CREATEINTERPOLATOR(mpmFlags) ▷*Create the interpolator and find number of grid nodes that can affect a particle*
  for matl **in** materialList **do**
    for part **in** particleList **do**
      n_{gp}, S_{gp} ← interpolator.FINDCELLSANDWEIGHTS($\mathbf{x}_p^n$, $\mathbf{h}_p^n$, $\boldsymbol{F}_p^n$) ▷*Find the node indices of the cells affecting the particle and the interpolation weights*
      $\mathbf{p}_p \leftarrow m_p[\text{matl}][\text{part}]\ \mathbf{v}_p^n[\text{matl}][\text{part}]$ ▷*Compute particle momentum*
      for node **in** n_{gp} **do**
        $m_g[\text{matl}][\text{node}] \leftarrow m_g[\text{matl}][\text{node}] + m_p[\text{matl}][\text{part}]\ S_{gp}[\text{node}]$
        $V_g[\text{matl}][\text{node}] \leftarrow V_g[\text{matl}][\text{node}] + V_p^n[\text{matl}][\text{part}]\ S_{gp}[\text{node}]$
        $\mathbf{v}_g[\text{matl}][\text{node}] \leftarrow \mathbf{v}_g[\text{matl}][\text{node}] + \mathbf{p}_p\ S_{gp}[\text{node}]$
        $\mathbf{f}_g^{\text{ext}}[\text{matl}][\text{node}] \leftarrow \mathbf{f}_g^{\text{ext}}[\text{matl}][\text{node}] + \mathbf{f}_p^{\text{ext},n+1}[\text{matl}][\text{part}]\ S_{gp}[\text{node}]$
        $\mathbf{b}_g[\text{node}] \leftarrow \mathbf{b}_g[\text{node}] + m_p[\text{matl}][\text{part}]\ \mathbf{b}_p^n[\text{matl}][\text{part}]\ S_{gp}[\text{node}]$
      end for
    end for
    for node **in** gridNodeList **do**
      $\mathbf{v}_g[\text{matl}][\text{node}] \leftarrow \mathbf{v}_g[\text{matl}][\text{node}]/m_g[\text{matl}][\text{node}]$
    end for
    $\mathbf{v}_g[\text{matl}] \leftarrow$ APPLYSYMMETRYVELOCITYBC($\mathbf{v}_g[\text{matl}]$) ▷*Apply any symmetry velocity BCs that may be applicable*
  end for
  return m_g, V_g, $\mathbf{v}_g$, $\mathbf{b}_g$, $\mathbf{f}_g^{\text{ext}}$
end procedure

4.4.4 Exchanging momentum using interpolated grid values

The exchange of momentum is carried out using a contact model. Details can be found in the Uintah Developers Manual.

4.4.5 Computing the internal force

This procedure computes the internal force at the grid nodes. The new quantities introduced in this section are

- n_{gp} : The number of grid nodes that are used to interpolate from particle to grid.
- S_{gp} : The nodal interpolation function evaluated at a particle
- $\boldsymbol{G}_{gp}$: The gradient of the nodal interpolation function evaluated at a particle
- $\boldsymbol{\sigma}_v$: A volume weighted grid node stress.
- $\mathbf{f}_g^{\text{int}}$: The internal force at a grid node.

Algorithm 8 Computing the internal force

Require: $\mathbf{h}_g$, V_g, V_p^n, $\mathbf{x}_p^n$, $\mathbf{h}_p^n$, $\boldsymbol{\sigma}_p^n$, $\boldsymbol{F}_p^n$, materialList, particleList, gridNodeList mpmFlags

1: **procedure** COMPUTEINTERNALFORCE
2: interpolator ← CREATEINTERPOLATOR(mpmFlags) ▷*Create the interpolator and find number of grid nodes that can affect a particle*
3: **for** matl **in** materialList **do**
4: **for** part **in** particleList **do**
5: $n_{gp}, S_{gp}, \mathbf{G}_{gp} \leftarrow$ interpolator.FINDCELLSANDWEIGHTSANDSHAPEDERVATIVES($\mathbf{x}_p^n, \mathbf{h}_p^n, \boldsymbol{F}_p^n$) ▷*Find the node indices of the cells affecting the particle and the interpolation weights and gradients*
6: $\boldsymbol{\sigma}_v \leftarrow V_p$[matl][part] $\boldsymbol{\sigma}_p^n$[matl][part]
7: **for** node **in** n_{gp} **do**
8: $\mathbf{f}_g^{\text{int}}$[matl][node] ← $\mathbf{f}_g^{\text{int}}$[matl][node] - ($\mathbf{G}_{gp}$[node]/$\mathbf{h}_g$) ·$\boldsymbol{\sigma}_p^n$[matl][part] V_p^n[part]
9: $\boldsymbol{\sigma}_g$[matl][node] ← $\boldsymbol{\sigma}_g$[matl][node] + $\boldsymbol{\sigma}_v$ S_{gp}[node]
10: **end for**
11: **end for**
12: **for** node **in** gridNodeList **do**
13: $\boldsymbol{\sigma}_g$[matl][node] ← $\boldsymbol{\sigma}_g$[matl][node]/V_g[matl][node]
14: **end for**
15: $\mathbf{v}_g$[matl] ← APPLYSYMMETRYTRACTIONBC() ▷*Apply any symmetry tractions BCs that may be applicable*
16: **end for**
17: **return** $\mathbf{f}_g^{\text{int}}, \boldsymbol{\sigma}_g, \mathbf{v}_g$
18: **end procedure**

4.4.6 Computing and integrating the acceleration

This procedure computes the accelerations at the grid nodes and integrates the grid accelerations using forward Euler to compute grid velocities. The new quantities introduced in this section are

- $\mathbf{a}_g$: The grid accelerations.
- $\mathbf{v}_g^\star$: The integrated grid velocities.

Algorithm 9 Computing and integrating the acceleration

Require: $\Delta t, m_g, \mathbf{f}_g^{\text{int}}, \mathbf{f}_g^{\text{ext}}, \mathbf{b}_g, \mathbf{v}_g$, materialList, gridNodeList, mpmFlags
1: **procedure** COMPUTEANDINTEGRATEACCELERATION
2: **for** matl **in** materialList **do**
3: **for** node **in** gridNodeList **do**
4: $\mathbf{a}_g$[matl][node] ← ($\mathbf{f}_g^{\text{int}}$[matl][node] + $\mathbf{f}_g^{\text{ext}}$[matl][node] + $\mathbf{b}_g$[matl][node])/m_g[matl][node]
5: $\mathbf{v}_g^\star$ ← $\mathbf{v}_g$[matl][node] + $\mathbf{a}_g$[matl][node] ∗Δt
6: **end for**
7: **end for**
8: **return** $\mathbf{v}_g^\star, \mathbf{a}_g$
9: **end procedure**

4.4.7 Exchanging momentum using integrated grid values

The exchange of momentum is carried out using a contact model. Details can be found in the Uintah Developers Manual.

4.4.8 Setting grid boundary conditions

Algorithm 10 Setting grid boundary conditions

Require: Δt, $\mathbf{a}_g$, $\mathbf{v}_g^\star$, $\mathbf{v}_g$, materialList, gridNodeList, mpmFlags
1: **procedure** SETGRIDBOUNDARYCONDITIONS
2: **for** matl **in** materialList **do**
3: $\mathbf{v}_g^\star$[matl] ← APPLYSYMMETRYVELOCITYBC($\mathbf{v}_g^\star$[matl])
4: **for** node **in** gridNodeList **do**
5: $\mathbf{a}_g$[matl][node] ← ($\mathbf{v}_g^\star$[matl][node] - $\mathbf{v}_g$[matl][node]) /Δt
6: **end for**
7: **end for**
8: **return** $\mathbf{v}_g^\star$, $\mathbf{a}_g$
9: **end procedure**

4.4.9 Computing the deformation gradient

The velocity gradient is computed using the integrated grid velocities and then used to compute the deformation gradient. The new quantities introduced in this section are

- $\Delta \boldsymbol{F}_p^n$: The increment of the particle deformation gradient.
- $\boldsymbol{l}_p^{n+1}$: The particle velocity gradient.
- ρ_0 : The initial mass density of the material.

Algorithm 11 Computing the velocity gradient and deformation gradient

Require: Δt, $\mathbf{x}_p^n$, m_p, V_p^n, $\mathbf{h}_p^n$, $\mathbf{v}_p^n$, $\boldsymbol{l}_p^n$, $\boldsymbol{F}_p^n$, $\mathbf{h}_g$, $\mathbf{v}_g$, $\mathbf{v}_g^\star$, ρ_0 materialList, gridNodeList, mpmFlags, velGradComputer
1: **procedure** COMPUTEDEFORMATIONGRADIENT
2: interpolator ← CREATEINTERPOLATOR(mpmFlags)
3: **for** matl **in** materialList **do**
4: **for** part **in** particleList **do**
5: $\boldsymbol{l}_p^{n+1}$[matl,part] ← velGradComputer.COMPUTEVELGRAD(interpolator, $\mathbf{h}_g$, $\mathbf{x}_p^n$[matl,part], ↪ $\mathbf{h}_p^n$[matl,part], $\boldsymbol{F}_p^n$[matl,part], $\mathbf{v}_g^\star$[matl]) ▷*Compute the velocity gradient*
6: $\boldsymbol{F}_p^{n+1}$[matl,part], $\Delta \boldsymbol{F}_p^{n+1}$ ← COMPUTEDEFORMATIONGRADIENTFROMVELOCITY($\boldsymbol{l}_p^n$[matl,part], ↪ $\boldsymbol{l}_p^{n+1}$[matl,part], $\boldsymbol{F}_p^n$[matl,part]) ▷*Compute the deformation gradient*
7: V_p^{n+1}[matl,part] ← m_p[matl,part]/ρ_0 ∗ det($\boldsymbol{F}_p^{n+1}$[matl,part])
8: **end for**
9: **end for**
10: **return** $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^{n+1}$, V_p^{n+1}
11: **end procedure**

Algorithm 12 Computing the deformation gradient using the velocity gradient

Require: Δt, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^n$, mpmFlags
1: **procedure** COMPUTEDEFORMATIONGRADIENTFROMVELOCITY
2: **if** mpmFlags.defGradAlgorithm = "first_order" **then**
3: $\boldsymbol{F}_p^{n+1}$, $\Delta \boldsymbol{F}_p^{n+1}$ ← SERIESUPDATECONSTANTVELGRAD(numTerms = 1, Δt, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^n$)
4: **else if** mpmFlags.defGradAlgorithm = "subcycle" **then**
5: $\boldsymbol{F}_p^{n+1}$, $\Delta \boldsymbol{F}_p^{n+1}$ ← SUBCYCLEUPDATECONSTANTVELGRAD(Δt, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^n$)
6: **else if** mpmFlags.defGradAlgorithm = "taylor_series" **then**
7: $\boldsymbol{F}_p^{n+1}$, $\Delta \boldsymbol{F}_p^{n+1}$ ← SERIESUPDATECONSTANTVELGRAD(numTerms = mpmFlags.numTaylorSeriesTerms, Δt, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^n$)
8: **else**
9: $\boldsymbol{F}_p^{n+1}$, $\Delta \boldsymbol{F}_p^{n+1}$ ← CAYLEYUPDATECONSTANTVELGRAD(Δt, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^n$)
10: **end if**

11: **return** $\boldsymbol{F}_p^{n+1}, \Delta\boldsymbol{F}_p^{n+1}$
12: **end procedure**

4.4.10 Computing the stress tensor

The stress tensor is compute by individual constitutive models. Details of the Arena partially saturated model are given later. The new quantities introduced in this section are

- $\boldsymbol{\eta}_p^n, \boldsymbol{\eta}_p^{n+1}$: The internal variables needed by the constitutive model.

Algorithm 13 Computing the stress tensor

Require: $\Delta t, \mathbf{x}_p^n, m_p, V_p^{n+1}, \mathbf{h}_p^n, \boldsymbol{l}_p^{n+1}, \boldsymbol{F}_p^{n+1}, \boldsymbol{\sigma}_p^n, \boldsymbol{\eta}_p^n, \rho_0$, `materialList`, `mpmFlags`, `constitutiveModel`
1: **procedure** COMPUTESTRESSTENSOR
2: **for** `matl` **in** `materialList` **do**
3: $\boldsymbol{\sigma}^{n+1}, \boldsymbol{\eta}_p^{n+1} \leftarrow$ `constitutiveModel[matl]`.COMPUTESTRESSTENSOR($\Delta t, \mathbf{x}_p^n, m_p, V_p^{n+1}, \mathbf{h}_p^n,$ $\boldsymbol{l}_p^{n+1}, \boldsymbol{F}_p^{n+1}, \boldsymbol{\sigma}_p^n, \boldsymbol{\eta}_p^n, \rho_0$, `mpmFlags`) ▷*Update the stress and any internal variables needed by the constitutive model*
4: **end for**
5: **return** $\boldsymbol{\sigma}_p^{n+1}, \boldsymbol{\eta}_p^{n+1}$
6: **end procedure**

4.4.11 Computing the basic damage parameter

The damage parameter is updated and the particle stress is modified in this procedure. The new quantities introduced in this section are

- $\varepsilon_p^{f,n}, \varepsilon_p^{f,n+1}$: The particle strain to failure at $t = T_n$ and $t = T_{n+1}$.
- χ_p^n, χ_p^{n+1} : An indicator function that identifies whether a particle has failed completely.
- $t_p^{\chi,n}, t_p^{\chi,n+1}$: The time to failure of a particle.
- D_p^n, D_p^{n+1} : A particle damage parameter that can be used to modify the stress.

Algorithm 14 Computing the damage parameter

Require: $t^{n+1}, V_p^{n+1}, \boldsymbol{F}_p^{n+1}, \boldsymbol{\sigma}_p^{n+1}, D_p^n, \varepsilon_p^{f,n}, \chi_p^n, t_p^{\chi,n}$, `materialList`, `mpmFlags`
1: **procedure** COMPUTEDAMAGE
2: **for** `matl` **in** `materialList` **do**
3: **for** `part` **in** `particleList` **do**
4: **if** `brittleDamage` = TRUE **then**
5: $\boldsymbol{\sigma}_p^{n+1}, \varepsilon_p^{f,n+1}, \chi_p^{n+1}, t_p^{\chi,n+1}, D_p^{n+1} \leftarrow$ UPDATEDAMAGEANDMODIFYSTRESS($V_p^{n+1}, \boldsymbol{F}_p^{n+1},$ $\boldsymbol{\sigma}_p^{n+1}, D_p^n, \varepsilon_p^{f,n}, \chi_p^n, t_p^{\chi,n}$) ▷*Update the damage parameters and stress*
6: **else**
7: $\boldsymbol{\sigma}_p^{n+1}, \varepsilon_p^{f,n+1}, \chi_p^{n+1}, t_p^{\chi,n+1} \leftarrow$ UPDATEFAILEDPARTICLESANDMODIFYSTRESS($V_p^{n+1}, \boldsymbol{F}_p^{n+1},$ $\boldsymbol{\sigma}_p^{n+1}, \varepsilon_p^{f,n}, \chi_p^n, t_p^{\chi,n}, t^{n+1}$) ▷*Update the failed particles and stress*
8: **end if**
9: **end for**
10: **end for**
11: **return** $\boldsymbol{\sigma}_p^{n+1}, \varepsilon_p^{f,n+1}, \chi_p^{n+1}, t_p^{\chi,n+1}, D_p^{n+1}$
12: **end procedure**

4.4.12 Updating the particle erosion parameter

The particle failure indicator function is updated in this procedure and used later for particle deletion if needed.

Algorithm 15 Updating the particle erosion parameter

Require: D_p^n, χ_p^n `materialList`, `mpmFlags`, `constitutiveModel`

procedure UPDATEEROSIONPARAMETER
  for `matl` **in** `materialList` **do**
    for `part` **in** `particleList` **do**
      if `matl.doBasicDamage` = TRUE **then**
        $\chi_p^{n+1} \leftarrow$ `damageModel.getLocalizationParameter()` ▷*Just get the indicator parameter for particles that will be eroded.*
      else
        $\chi_p^{n+1}, D_p^{n+1} \leftarrow$ `constitutiveModel[matl].getDamageParameter`(χ_p^n, D_p^n) ▷*Update the damage parameter in the constitutive model.*
      end if
    end for
  end for
  return χ_p^{n+1}, D_p^{n+1}
end procedure

4.4.13 Interpolating back to the particles and update

This is the final step at which the particle velocities and positions are updated and the grid is reset. Particle that are to be removed are dealt with in a subsequent `relocation` step.

Algorithm 16 Interpolating back to the particles and position update

Require: Δt, $\mathbf{a}_g$, $\mathbf{v}_g^\star$, $\mathbf{x}_p^n$, $\mathbf{v}_p^n$, $\mathbf{u}_p^n$, $\mathbf{h}_p^n$, χ_p^{n+1}, $\boldsymbol{F}_p^{n+1}$, V_p^{n+1}, `materialList`, `particleList`, `gridNodeList`, `mpmFlags`

procedure INTERPOLATETOPARTICLESANDUPDATE
  `interpolator` ← CREATEINTERPOLATOR(`mpmFlags`)
  for `matl` **in** `materialList` **do**
    $\mathbf{h}_p^{n+1} \leftarrow \mathbf{h}_p^n$
    for `part` **in** `particleList` **do**
      $n_{gp}, S_{gp} \leftarrow$ `interpolator.findCellsAndWeights`$(\mathbf{x}_p^n, \mathbf{h}_p^{n+1}, \boldsymbol{F}_p^{n+1})$
      $\mathbf{v} \leftarrow \mathbf{0}$, $\mathbf{a} \leftarrow \mathbf{0}$,
      for `node` **in** `gridNodeList` **do**
        $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{v}_g^\star$[node] $*$ S_{gp}[node] ▷*Update particle velocity*
        $\mathbf{a} \leftarrow \mathbf{a} + \mathbf{a}_g$[node] $*$ S_{gp}[node] ▷*Update particle acceleration*
      end for
      $\mathbf{x}_p^{n+1} \leftarrow \mathbf{x}_p^n + \mathbf{v} * \Delta t$ ▷*Update position*
      $\mathbf{u}_p^{n+1} \leftarrow \mathbf{u}_p^n + \mathbf{v} * \Delta t$ ▷*Update displacement*
      $\mathbf{v}_p^{n+1} \leftarrow \mathbf{v}_p^n + \mathbf{a} * \Delta t$ ▷*Update velocity*
    end for
  end for
  DELETEROGUEPARTICLES() ▷*Delete particles that are to be eroded.*
  return $V_p^{n+1}, \mathbf{u}_p^{n+1}, \mathbf{v}_p^{n+1}, \mathbf{x}_p^{n+1}, m_p, \mathbf{h}_p^{n+1}$
end procedure

5 Implementation of stress update in ARENA

Recall the stress tensor computation procedure calls the computeStressTensor routine that is specific to each constitutive model as can be seen in the pseudocode below.

Algorithm 17 Computing the stress tensor

Require: Δt, $\mathbf{x}_p^n$, m_p, V_p^{n+1}, $\mathbf{h}_p^n$, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^{n+1}$, $\boldsymbol{\sigma}_p^n$, $\boldsymbol{\eta}_p^n$, ρ_0, materialList, mpmFlags, constitutiveModel

procedure computeStressTensor
  for matl **in** materialList **do**
    $\boldsymbol{\sigma}_p^{n+1}, \boldsymbol{\eta}_p^{n+1} \leftarrow$ constitutiveModel[matl].computeStressTensor(Δt, $\mathbf{x}_p^n$, m_p, V_p^{n+1}, $\mathbf{h}_p^n$, $\boldsymbol{l}_p^{n+1}$, $\boldsymbol{F}_p^{n+1}$, $\boldsymbol{\sigma}_p^n$, $\boldsymbol{\eta}_p^n$, ρ_0, mpmFlags) ▷ *Update the stress and any internal variables needed by the constitutive model*
  end for
  return $\boldsymbol{\sigma}_p^{n+1}$, $\boldsymbol{\eta}_p^{n+1}$
end procedure

5.1 Initialization of the model

The model is initialized in two steps. In the first step, the constitutive model object is created followed by initialization of the stress (and the deformation gradient if needed). Here

- ϕ_0 : The initial porosity.
- S_0 : The initial porosity.
- $n_{\max}$: The maximum number of subcycles in the plastic return algorithm.
- $\varepsilon_{v,p}^{e,n}$: The elastic volumetric strain at a particle at $t = t_n$.
- $\boldsymbol{\sigma}_p^n$: The dynamic Cauchy stress at a particle at $t = t_n$.
- $\boldsymbol{\sigma}_{qs,p}^n$: The quasistatic Cauchy stress at a particle at $t = t_n$.
- $\boldsymbol{\sigma}^0$: The initial Cauchy stress at a particle.

Algorithm 18 Creating the Arena3PartiallySaturated constitutive model object

Require: mpmFlags, xmlProblemSpec

procedure createConstitutiveModel
  elasticityModel ← ElasticModuliModelFactory.create(xmlProblemSpec)
  yieldCondition ← YieldConditionFactory.create(xmlProblemSpec)
  $p_0, p_1, p_1^{\text{sat}}, p_2 \leftarrow$ readCrushCurveParameters(xmlProblemSpec)
  $\phi_0, S_0, \overline{p_0^w} \leftarrow$ readInitialPorosityAndSaturation(xmlProblemSpec)
  $n_{\max} \leftarrow$ readSubcyclingCharacteristicNumber(xmlProblemSpec)
  return elasticityModel, yieldCondition, ϕ_0, S_0, $\overline{p_0^w}$, $n_{\max}$, p_0, p_1, p_1^{sat}, p_2
end procedure

Algorithm 19 Initializing the Arena3PartiallySaturated particle variables

Require: ϕ_0, S_0, $\overline{p_0^w}$, particleList, V_p^0, m_p^0, $\mathbf{v}_p^0$, fluidParams, elasticityModel, yieldCondition

procedure initialize
  yieldCondition.initialize(particleList, V_p^0)
  yieldParams ← yieldCondition.getParameters()
  for part **in** particleList **do**
    ϕ_p^0[part] ← ϕ_0
    $S_{w,p}^0$[part] ← S_0
    χ_p^0[part] ← 0

$\varepsilon^{e,0}_{v,p}$[part] ← 0
$\boldsymbol{\sigma}^0_p$[part] ← (fluidParams.$\overline{p^w_0}$) $\boldsymbol{I}$
$\boldsymbol{\sigma}^0_{qs,p}$[part] ← $\boldsymbol{\sigma}^0$
end for
Δt ← computeStableTimestep(V^0_p, m_p, $\mathbf{v}^0_p$, elasticityModel)
return ϕ^0_p, $S^0_{w,p}$, χ^0_p, $\varepsilon^{e,0}_{v,p}$, $\boldsymbol{\sigma}^0_{qs,p}$, $\boldsymbol{\sigma}^0_p$, Δt
end procedure

5.2 Computing the stress and internal variables

The computeStressTensor routine in the partially saturated Arena model assumes that the Biot coefficient is $B = 1$, and has the following form. Here we introduce the new variables

- ϕ^n_p, ϕ^{n+1}_p : The porosity at $t = t_n$ and $t = t_{n+1}$.
- $S^n_{w,p}, S^{n+1}_{w,p}$: The saturation at $t = t_n$ and $t = t_{n+1}$.
- $a_{1,p}, a_{2,p}, a_{3,p}, a_{4,p}$: Yield condition parameters at each particle.
- $p^n_{3,p}$: The particle crush curve parameter p_3 at $t = t_n$.
- X^n_p : The particle hydrostatic compressive strength at $t = t_n$.
- κ^n_p : The yield function branch point at $t = t_n$.
- $\boldsymbol{\varepsilon}^{p,n}_p$: The particle plastic strain tensor at $t = t_n$.
- $\boldsymbol{\alpha}^n_p$: The particle backstress tensor at $t = t_n$.
- $\mathbf{d}^n$: The particle rate of deformation tensor at $t = t_n$.
- $\boldsymbol{R}^n, \boldsymbol{U}^n$: The particle rotation and stretch tensors at $t = t_n$.
- K^n, G^n : The particle bulk and shear modulus at $t = t_n$.

Algorithm 20 Computing the Arena partially saturated stress tensor

Require: Δt, $\mathbf{x}^n_p$, m_p, V^{n+1}_p, $\mathbf{h}^n_p$, $\boldsymbol{l}^{n+1}_p$, $\boldsymbol{F}^n_p$, $\boldsymbol{F}^{n+1}_p$, X^n_p, κ^n_p, $\varepsilon^n_{v,p}$, $p^n_{3,p}$, $\boldsymbol{\varepsilon}^{p,n}_p$, $\boldsymbol{\alpha}^n_p$ ϕ^n_p, $S^n_{w,p}$, χ^n_p, $\varepsilon^{e,n}_{v,p}$, $\boldsymbol{\sigma}^n_{qs,p}$, $\boldsymbol{\sigma}^n_p$, ρ_0, particleList, mpmFlags, elasticityModel, yieldCondition

procedure computeStressTensor
  yieldParams ← yieldCondition.getParameters()
  $a_{1,p}, a_{2,p}, a_{3,p}, a_{4,p}, I^{peak}_{1,p}, R_{c,p}$ ← yieldCondition.getLocalVariables()
    ↪ ▷*Yield condition parameters vary at each particle.*
    ↪ *Get the per-particle values of these parameters.*
  for part **in** particleList **do**
    χ^{n+1}_p[part] ← χ^n_p[part] ▷*Copy over failure indicator variable*
    $\boldsymbol{d}^{n+1}$ ← $[\boldsymbol{l}^{n+1}_p$[part] $+(\boldsymbol{l}^{n+1}_p$[part]$)^T]/2$ ▷*Compute rate of deformation*
    $\boldsymbol{R}^n, \boldsymbol{U}^n$ ← polarDecomposition($\boldsymbol{F}^n_p$[part]) ▷*Compute rotation and stretch tensors*
    $\boldsymbol{d}^{n+1}_{\text{unrot}} \leftarrow (\boldsymbol{R}^n)^T \cdot \boldsymbol{d}^{n+1} \cdot \boldsymbol{R}^n$ ▷*Unrotate the rate of deformation tensor*
    $\boldsymbol{\sigma}^n_{qs,\text{unrot}} \leftarrow (\boldsymbol{R}^n)^T \cdot \boldsymbol{\sigma}^n_{qs,p}$[part]$\cdot \boldsymbol{R}^n$ ▷*Unrotate the quasistatic stress*
    $\boldsymbol{\sigma}^n_{\text{unrot}} \leftarrow (\boldsymbol{R}^n)^T \cdot \boldsymbol{\sigma}^n_p$[part]$\cdot \boldsymbol{R}^n$ ▷*Unrotate the total stress*
    $K^n, G^n, \boldsymbol{s}^n, (\overline{p^w})^n, I^{\text{eff},n}_1, \sqrt{J^n_2}, r^n, z^n_{\text{eff}}, \varepsilon^{p,n}_v$ ←
      ↪ computeElasticProperties($\boldsymbol{\sigma}^n_{qs,\text{unrot}}$[part],
      ↪ ϕ^n_p[part], $S^n_{w,p}$[part], $\boldsymbol{\varepsilon}^{p,n}_p$[part], $\boldsymbol{\alpha}^n_p$[part], $p^n_{3,p}$[part])
      ↪ ▷*Compute elastic properties and stress invariants*
    isSuccess, $\boldsymbol{\sigma}^{n+1}_{qs,\text{unrot}}$, ϕ^{n+1}_{qs}, $S^{n+1}_{w,qs}$, X^{n+1}_p[part], $\boldsymbol{\alpha}^{n+1}_p$[part], $\boldsymbol{\varepsilon}^{p,n+1}_p$[part] ←
      ↪ rateIndependentPlasticUpdate(Δt, $\boldsymbol{d}^{n+1}_{\text{unrot}}$, K^n, G^n, $\boldsymbol{s}^n$, $(\overline{p^w})^n$, $I^{\text{eff},n}_1$, $\sqrt{J^n_2}$, r^n, z^n_{eff},
      ↪ $\varepsilon^{p,n}_v$, $\boldsymbol{\sigma}^n_{qs,\text{unrot}}$, ϕ^n_p[part], $S^n_{w,p}$[part], X^n_p[part], $\boldsymbol{\alpha}^n_p$[part], $\boldsymbol{\varepsilon}^{p,n}_p$[part], $p^n_{3,p}$[part],
      ↪ $a_{1,p}$[part], $a_{2,p}$[part], $a_{3,p}$[part], $a_{4,p}$[part])
      ↪ ▷*Compute updated quasistatic state using the consistency bisection algorithm*

```
        if isSuccess = FALSE then
            flagParticleForDeletion(part)
        end if
        σ^{n+1}_unrot ← rateDependentPlasticUpdate(Δt, d^{n+1}_unrot, σ^n_{qs,unrot}, σ^{n+1}_{qs,unrot}, φ^n_p[part], φ^{n+1}_qs,
          ↪ S^n_{w,p}[part], S^{n+1}_{w,qs}, X^n_p[part], X^{n+1}_p[part], α^n_p[part], α^{n+1}_p[part],
          ↪ ε^{p,n}_p[part], ε^{p,n+1}_p[part], p^n_{3,p}[part],
          ↪ a_{1,p}[part], a_{2,p}[part], a_{3,p}[part], a_{4,p}[part])
        R^{n+1}, U^{n+1} ← polarDecomposition(F^{n+1}_p[part]) ▷Compute rotation and stretch tensors
        σ^{n+1}_{p,qs}[part] ← R^{n+1} · σ^{n+1}_{qs,unrot}·(R^{n+1})^T ▷Rotate the quasistatic stress
        σ^{n+1}_p[part] ← R^{n+1} · σ^{n+1}_{unrot}·(R^{n+1})^T ▷Rotate the dynamic stress
    end for
    Δt^{n+1} ← computeStableTimestep(V^{n+1}_p, m_p, σ^{n+1}_p, σ^n_p, l^{n+1}_p, elasticityModel)
    return σ^{n+1}_p, σ^{n+1}_{p,qs}, φ^{n+1}_p, S^{n+1}_{w,p}, α^{n+1}_p, X^{n+1}_p, ε^{p,n+1}_p, Δt^{n+1}.
end procedure
```

5.2.1 Compute elastic properties

The pseudocode for the generic computeElasticProperties is listed below. The function has side effects beyond computing the elastic properties and should be used carefully. Note that the subscript p has been dropped for simplicity because all quantities are particle-based.

Algorithm 21 Computing the elastic properties

```
Require: σ, φ, S_w, ε^p, α, p_3, elasticityModel
procedure computeElasticProperties
    s, p̄^w, I_1^eff, √J_2, r, z_eff ← updateStressInvariants(σ, α) ▷Compute the deviatoric stress and the invariants of the input stress tensor.
    ε_v^p ← updateVolumetricPlasticStrain(ε^p) ▷Compute the volumetric plastic strain from the input plastic strain tensor.
    K, G ← elasticityModel.getCurrentElasticModuli(I_1^eff, p̄^w, ε_v^p, φ, S_w) ▷Compute the elastic moduli corresponding to the input state.
    if useDisaggregationAlgorithm = TRUE then
        scale = MAX(exp[−(p_3 + ε_v^p)], 10^−5)
        K ← K*scale , G ← G*scale
    end if
    return K, G, s, p̄^w, I_1^eff, √J_2, r, z_eff, ε_v^p
end procedure
```

Algorithm 22 Updating the stress invariants

```
procedure updateStressInvariants(σ, α)
    I_1 ← tr(σ)
    s ← σ − (I_1/3)I ▷Compute deviatoric stress
    p̄^w ← −tr(α)/3 ▷Compute pore pressure
    I_1^eff ← I_1 + 3p̄^w , √J_2 ← √((1/2)s : s) ▷Compute invariants of the effective stress
    r ← √(2J_2) , z_eff ← I_1^eff/√3 ▷Compute Lode coordinates of the effective stress
    return s, p̄^w, I_1^eff, √J_2, r, z_eff
end procedure
```

The elastic modulus computation procedures: The functions used to compute the moduli are listed in the pseudocode below.

Algorithm 23 Computing the current elastic moduli

Require: I_1^{eff}, $\overline{p^w}$, ε_v^{p}, ϕ, S_w

procedure GETCURRENTELASTICMODULI
  $\overline{I_1}^{\text{eff}} \leftarrow -I_1^{\text{eff}}$, $\overline{\varepsilon_v^{\text{p}}} \leftarrow -\varepsilon_v^{\text{p}}$,
  $K \leftarrow 0$, $G \leftarrow 0$
  if $S_w > 0$ **then**
    $K, G \leftarrow$ COMPUTEPARTIALSATURATEDMODULI($\overline{I_1}^{\text{eff}}$, $\overline{p^w}$, $\overline{\varepsilon_v^{\text{p}}}$, ϕ, S_w)
  else
    $K, G \leftarrow$ COMPUTEDRAINEDMODULI($\overline{I_1}^{\text{eff}}$, $\overline{\varepsilon_v^{\text{p}}}$)
  end if
  return K, G
end procedure

Algorithm 24 Computing the partially saturated elastic moduli

Require: K_{s0}, n_s, $\bar{p}_{s0}$, K_{w0}, n_w, $\bar{p}_{w0}$, γ, $\bar{p}_r$

procedure COMPUTEPARTIALSATURATEDMODULI($\overline{I_1}^{\text{eff}}$, $\overline{p^w}$, $\overline{\varepsilon_v^{\text{p}}}$, ϕ, S_w)
  if $\overline{I_1}^{\text{eff}} > 0$ **then**
    $\bar{p}^{\text{eff}} \leftarrow \overline{I_1}^{\text{eff}}/3$
    $K_s \leftarrow K_{s0} + n_s(\bar{p}^{\text{eff}} - \bar{p}_{s0})$
    $K_w \leftarrow K_{w0} + n_w(\overline{p^w} - \bar{p}_{w0})$
    $K_a \leftarrow \gamma(\overline{p^w} + \bar{p}_r)$
    $K_d, G \leftarrow$ COMPUTEDRAINEDMODULI($\overline{I_1}^{\text{eff}}$, $\overline{\varepsilon_v^{\text{p}}}$)
    $K_f \leftarrow 1/\left[S_w/K_w + (1 - S_w)/K_a\right]$ ▷*Bulk modulus of air + water mixture*
    numer $\leftarrow (1 - K_d/K_s)^2$
    denom $\leftarrow (1/K_s)\,(1 - K_d/K_s) + \phi\,(1/K_f - 1/K_s)$
    $K \leftarrow K_d +$ numer/denom ▷*Bulk modulus of partially saturated material (Biot-Gassman model)*
  else
    $K, G \leftarrow$ COMPUTEDRAINEDMODULI($\overline{I_1}$, $\overline{\varepsilon_v^{\text{p}}}$)
  end if
  return K, G
end procedure

Algorithm 25 Computing the drained elastic moduli

Require: K_{s0}, n_s, $\bar{p}_{s0}$, b_0, b_1, b_2, b_3, b_4, G_0, ν_1, ν_2

procedure COMPUTEDRAINEDMODULI($\overline{I_1}^{\text{eff}}$, $\overline{\varepsilon_v^{\text{p}}}$)
  if $\overline{I_1}^{\text{eff}} > 0$ **then**
    $\bar{p}^{\text{eff}} \leftarrow \overline{I_1}^{\text{eff}}/3$
    $K_s \leftarrow K_{s0} + n_s(\bar{p}^{\text{eff}} - \bar{p}_{s0})$
    $K_s^{\text{ratio}} \leftarrow K_s/(1 - n_s\bar{p}^{\text{eff}}/K_s)$
    $\varepsilon_v^{\text{e}} \leftarrow \text{POW}((b_3\bar{p}^{\text{eff}})/(b_1K_s - b_2\bar{p}^{\text{eff}}), (1/b_4))$;
    $y \leftarrow \text{POW}(\varepsilon_v^{\text{e}}, b_4)$
    $z \leftarrow b_2y + b_3$
    $K \leftarrow K_s^{\text{ratio}}[b_0 + (1/\varepsilon_v^{\text{e}})b_1b_3b_4y/z^2]$; ▷ *Compute compressive bulk modulus*
    $\nu = \nu_1 + \nu_2 \exp(-K/K_s)$
    $G \leftarrow G_0$
    if $\nu > 0$ **then**
      $G \leftarrow 1.5K\,(1 - 2\nu)/(1 + \nu)$ ▷*Update the shear modulus (if $\nu_1, \nu_2 > 0$)*
    end if

else
  $K \leftarrow b_0 K_{s0}$ ▷*Tensile bulk modulus = Bulk modulus at $p = 0$*
  $G \leftarrow G_0$ ▷*Tensile shear modulus*
end if
return K, G
end procedure

5.2.2 Rate-independent stress update

The pseudocode for the algorithm is given below. All quantities are particle-based and the subscript p has been dropped for convenience.

Algorithm 26 The rate-independent stress and internal variable update algorithm

Require: Δt, $\boldsymbol{d}^{n+1}$, K^n, G^n, $\boldsymbol{s}^n$, $(\overline{p^w})^n$, $I_1^{\text{eff},n}$, $\sqrt{J_2^n}$, r^n, z_{eff}^n, $\varepsilon_v^{\text{p},n}$, a_1, a_2, a_3, a_4, p_3^n, $\boldsymbol{\sigma}_{qs}^n$, ϕ^n, S_w^n, X^n, $\boldsymbol{\alpha}^n$, $\boldsymbol{\varepsilon}^{\text{p},n}$, $n_{\max}$, ϵ_{sub}, $\chi_{\max}$

procedure RATEINDEPENDENTPLASTICUPDATE
  $\boldsymbol{\sigma}_{\text{trial}} \leftarrow$ COMPUTETRIALSTRESS($\boldsymbol{\sigma}_{qs}^n$, K^n, G^n, $\boldsymbol{d}^{n+1}$, Δt) ▷*Compute trial stress*
  $\boldsymbol{\alpha}^{\text{trial}} \leftarrow \boldsymbol{\alpha}^n$, $p_3^{\text{trial}} \leftarrow p_3^n$, $\phi^{\text{trial}} \leftarrow \phi^n$, $S_w^{\text{trial}} \leftarrow S_w^n$,
    ↪ $X^{\text{trial}} \leftarrow X^n$, $\boldsymbol{\varepsilon}^{\text{p,trial}} \leftarrow \boldsymbol{\varepsilon}^{\text{p},n}$ ▷*Set all other trial quantities to the values*
    ↪ *at the beginning of the timestep*
  K^{trial}, G^{trial}, $\boldsymbol{s}^{\text{trial}}$, $(\overline{p^w})^{\text{trial}}$, $I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, $\varepsilon_v^{\text{p,trial}} \leftarrow$
    ↪ COMPUTEELASTICPROPERTIES($\boldsymbol{\sigma}_{\text{trial}}$, ϕ^{trial}, S_w^{trial}, $\boldsymbol{\varepsilon}^{\text{p,trial}}$, $\boldsymbol{\alpha}^{\text{trial}}$, p_3^{trial}) ▷*Update the trial*
    ↪ *values of the moduli and compute the invariants of the trial stress*
  $n_{\text{sub}} \leftarrow$ COMPUTESTEPDIVISIONS($n_{\max}$, ϵ_{sub}, K^n, K^{trial}, I_1^{peak}, a_1, X^n, $\boldsymbol{\sigma}_{qs}^n$, $\boldsymbol{\sigma}_{\text{trial}}$)
    ↪ ▷*Compute number of substeps used by the return algorithm*
  if $n_{\text{sub}} < 0$ **then**
    return isSuccess = FALSE
  end if
  $\delta t \leftarrow \dfrac{\Delta t}{n_{\text{sub}}}$ ▷*Substep timestep*
  $\chi \leftarrow 1$, $t_{\text{local}} \leftarrow 0$ ▷*Initialize substep multiplier and accumulated time increment*
  $\boldsymbol{\sigma}^k \leftarrow \boldsymbol{\sigma}_{qs}^n$, $\boldsymbol{\varepsilon}^{\text{p},k} \leftarrow \boldsymbol{\varepsilon}^{\text{p},n}$, $\phi^k \leftarrow \phi^n$, $S_w^k \leftarrow S_w^n$, $X^k \leftarrow X^n$, $\boldsymbol{\alpha}^k \leftarrow \boldsymbol{\alpha}^n$, $K^k \leftarrow K^n$, $G^k \leftarrow G^n$, $p_3^k \leftarrow p_3^n$,
    ↪ $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^n$, $(\overline{p^w})^k \leftarrow (\overline{p^w})^n$, $I_1^{\text{eff},k} \leftarrow I_1^{\text{eff},n}$, $\sqrt{J_2^k} \leftarrow \sqrt{J_2^n}$, $r^k \leftarrow r^n$, $z_{\text{eff}}^k \leftarrow z_{\text{eff}}^n$, $\varepsilon_v^{\text{p},k} \leftarrow \varepsilon_v^{\text{p},n}$
  repeat
    isSuccess, $\boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\varepsilon}^{\text{p},k+1}$, ϕ^{k+1}, S_w^{k+1}, X^{k+1}, $\boldsymbol{\alpha}^{k+1}$, K^{k+1}, G^{k+1}, $p_3^{k+1} \leftarrow$
      ↪ COMPUTESUBSTEP($\boldsymbol{\sigma}^k$, $\boldsymbol{\varepsilon}^{\text{p},k}$, ϕ^k, S_w^k, X^k, $\boldsymbol{\alpha}^k$, K^k, G^k, $\boldsymbol{s}^k$, $(\overline{p^w})^k$, $I_1^{\text{eff},k}$, $\sqrt{J_2^k}$, r^k, z_{eff}^k,
      ↪ $\varepsilon_v^{\text{p},k}$, p_3^k, $\boldsymbol{d}^{n+1}$, δt)
      ↪ ▷*Compute updated stress and internal variables for the current substep*
    if isSuccess = TRUE **then**
      $t_{\text{local}} \leftarrow t_{\text{local}} + \delta t$
      $\boldsymbol{\sigma}^k \leftarrow \boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\varepsilon}^{\text{p},k} \leftarrow \boldsymbol{\varepsilon}^{\text{p},k+1}$, $\phi^k \leftarrow \phi^{k+1}$, $S_w^k \leftarrow S_w^{k+1}$, $X^k \leftarrow X^{k+1}$, $\boldsymbol{\alpha}^k \leftarrow \boldsymbol{\alpha}^{k+1}$
      $K^k \leftarrow K^{k+1}$, $G^k \leftarrow G^{k+1}$, $p_3^k \leftarrow p_3^{k+1}$
    else
      $\delta t \leftarrow \delta t/2$ ▷*Halve the timestep*
      $\chi \leftarrow 2\chi$ ▷*Keep a count of how many times the timestep has been halved.*
      if $\chi > \chi_{\max}$ **then**
        return isSuccess = FALSE, $\boldsymbol{\sigma}^k$, ϕ^k, S_w^k, X^k, $\boldsymbol{\alpha}^k$, $\boldsymbol{\varepsilon}^{\text{p},k}$, K^k, G^k, p_3^k
          ↪ ▷*Algorithm has failed to converge*
      end if
    end if
  until $t_{\text{local}} \geq \Delta t$

$\boldsymbol{\sigma}_{qs}^{n+1} \leftarrow \boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\alpha}^{n+1} \leftarrow \boldsymbol{\alpha}^{k+1}$, $\boldsymbol{\varepsilon}^{\mathrm{p},n+1} \leftarrow \boldsymbol{\varepsilon}^{\mathrm{p},k+1}$, $\phi^{n+1} \leftarrow \phi^{k+1}$, $S_w^{n+1} \leftarrow S_w^{k+1}$, $X^{n+1} \leftarrow X^{k+1}$
$K^{n+1} \leftarrow K^{k+1}$, $G^{n+1} \leftarrow G^{k+1}$, $p_3^{n+1} \leftarrow p_3^{k+1}$
return isSuccess = TRUE, $\boldsymbol{\sigma}_{qs}^{n+1}$, ϕ^{n+1}, S_w^{n+1}, X^{n+1}, $\boldsymbol{\alpha}^{n+1}$, $\boldsymbol{\varepsilon}^{\mathrm{p},n+1}$, K^{n+1}, G^{n+1}, p_3^{n+1} ↪ ▷*Algorithm has converged*
end procedure

5.2.3 Computing the trial stress

The pseudocode of the trial stress algorithm is given below.

Algorithm 27 Computing the trial stress

procedure COMPUTETRIALSTRESS($\boldsymbol{\sigma}_{qs}^n$, K^n, G^n, $\boldsymbol{d}^{n+1}$, Δt)
  $\Delta\boldsymbol{\varepsilon} \leftarrow \boldsymbol{d}^{n+1}\Delta t$ ▷*Total strain increment*
  $\Delta\boldsymbol{\varepsilon}^{\mathrm{iso}} \leftarrow \frac{1}{3}\mathrm{tr}(\Delta\boldsymbol{\varepsilon})\boldsymbol{I}$
  $\Delta\boldsymbol{\varepsilon}^{\mathrm{dev}} \leftarrow \Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{iso}}$
  $\boldsymbol{\sigma}_{\mathrm{trial}} \leftarrow \boldsymbol{\sigma}_{qs}^n + 3K^n\Delta\boldsymbol{\varepsilon}^{\mathrm{iso}} + 2G^n\Delta\boldsymbol{\varepsilon}^{\mathrm{dev}}$
  return $\boldsymbol{\sigma}_{\mathrm{trial}}$
end procedure

5.2.4 Computing the number of subcycles in the return algorithm

Algorithm 28 Computing the number of subcycles

Require: $n_{\max}$, $\epsilon_{\mathrm{sub}} \leftarrow 10^{-4}$, K^n, K^{trial}, I_1^{peak}, a_1, X^n, $\boldsymbol{\sigma}_{qs}^n$, $\boldsymbol{\sigma}_{\mathrm{trial}}$
procedure COMPUTESTEPDIVISIONS
  $n_{\mathrm{bulk}} \leftarrow \lceil |K^n - K^{\mathrm{trial}}| / K^n \rceil$ ▷*Compute change in bulk modulus*
  $\Delta\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma}_{\mathrm{trial}} - \boldsymbol{\sigma}_{qs}^n$
  $L \leftarrow \frac{1}{2}(I_1^{\mathrm{peak}} - X^n)$
  if $a_1 > 0$ **then**
    $L \leftarrow \mathrm{MIN}(L, a_1)$
  end if
  $n_{\mathrm{yield}} \leftarrow \lceil \epsilon_{\mathrm{sub}} \times \|\Delta\boldsymbol{\sigma}\| / L \rceil$ ▷*Compute trial stress increment relative to yield surface size*
  $n_{\mathrm{sub}} \leftarrow \mathrm{MAX}(n_{\mathrm{bulk}}, n_{\mathrm{yield}})$ ▷n_{sub} *is the maximum of the two values*
  if $n_{\mathrm{sub}} > n_{\max}$ **then**
    $n_{\mathrm{sub}} \leftarrow -1$
  end if
  return n_{sub}
end procedure

5.2.5 Updating the stress for a substep: consistency bisection

This procedure computes the updated stress state for a substep that may be either elastic, plastic, or partially elastic. It uses Homel's consistency bisection and non-hardening return concepts (Homel, Guilkey, and Brannon, 2015).

Algorithm 29 Computing the stress and internal variable update for a substep

Require: $\boldsymbol{d}^{n+1}$, δt, $\boldsymbol{\sigma}^k$, $\boldsymbol{\varepsilon}^{\mathrm{p},k}$, ϕ^k, S_w^k, X^k, $\boldsymbol{\alpha}^k$, K^k, G^k, $\boldsymbol{s}^k$, $(\overline{p^w})^k$, $I_1^{\mathrm{eff},k}$, $\sqrt{J_2^k}$, r^k, z_{eff}^k, $\varepsilon_v^{\mathrm{p},k}$, p_3^k, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, `yieldCondition`
procedure COMPUTESUBSTEP
  $\delta\boldsymbol{\varepsilon} \leftarrow \boldsymbol{d}^{n+1}\delta t$ ▷*Compute strain increment*

$\boldsymbol{\sigma}_{\text{trial}} \leftarrow$ computeTrialStress($\boldsymbol{\sigma}^k$, K^k, G^k, $\boldsymbol{d}^{n+1}$, Δt) ▷*Compute substep trial stress*
$\boldsymbol{\alpha}^{\text{trial}} \leftarrow \boldsymbol{\alpha}^k$, $K^{\text{trial}} \leftarrow K^k$, $G^{\text{trial}} \leftarrow G^k$, $p_3^{\text{trial}} \leftarrow p_3^k$, $\phi^{\text{trial}} \leftarrow \phi^k$, $S_w^{\text{trial}} \leftarrow S_w^k$, $X^{\text{trial}} \leftarrow X^k$, $\boldsymbol{\varepsilon}^{\text{p,trial}} \leftarrow \boldsymbol{\varepsilon}^{\text{p},k}$ ▷*Set all other trial quantities to the values at the beginning of the substep*
K^{trial}, G^{trial}, $\boldsymbol{s}^{\text{trial}}$, $(\overline{p^w})^{\text{trial}}$, $I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, $\varepsilon_v^{\text{p,trial}} \leftarrow$ computeElasticProperties($\boldsymbol{\sigma}_{\text{trial}}$, ϕ^{trial}, S_w^{trial}, $\boldsymbol{\varepsilon}^{\text{p,trial}}$, $\boldsymbol{\alpha}^{\text{trial}}$, p_3^{trial}) ▷*Compute elastic moduli and stress invariants for the trial state*
isElastic ← `yieldCondition`.evalYieldCondition($I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, X^{trial}, $(\overline{p^w})^{\text{trial}}$, ϕ^{trial}, S_w^{trial}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β) ▷*Determine whether the trial stress is elastic or not*
if isElastic = TRUE **then**
  $\boldsymbol{\sigma}^{k+1} \leftarrow \boldsymbol{\sigma}_{\text{trial}}$, $\boldsymbol{\varepsilon}^{\text{p},k+1} \leftarrow \boldsymbol{\varepsilon}^{\text{p,trial}}$, $\phi^{k+1} \leftarrow \phi^{\text{trial}}$, $S_w^{k+1} \leftarrow S_w^{\text{trial}}$, $X^{k+1} \leftarrow X^{\text{trial}}$, $\boldsymbol{\alpha}^{k+1} \leftarrow \boldsymbol{\alpha}^{\text{trial}}$
  $K^{k+1} \leftarrow K^{\text{trial}}$, $G^{k+1} \leftarrow G^{\text{trial}}$, $p_3^{k+1} \leftarrow p_3^{\text{trial}}$ ▷*This is an elastic substep. Update the state to the trial value.*
  isSuccess = TRUE
  return isSuccess, $\boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\varepsilon}^{\text{p},k+1}$ ϕ^{k+1}, S_w^{k+1}, X^{k+1}, $\boldsymbol{\alpha}^{k+1}$, K^{k+1}, G^{k+1}, p_3^{k+1}
end if
$\boldsymbol{\sigma}_{\text{fixed}}$, $\delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{p}} \leftarrow$ nonHardeningReturn($\boldsymbol{\sigma}^k$, $\delta\boldsymbol{\varepsilon}$, X^k, K^k, G^k, $(\overline{p^w})^k$, $\boldsymbol{s}^{\text{trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β) ▷*Compute return to updated yield surface (no hardening)*
isSuccess, $\boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\varepsilon}^{\text{p},k+1}$, $\boldsymbol{\alpha}^{k+1}$, $(\overline{p^w})^{k+1}$, ϕ^{k+1}, S_w^{k+1}, X^{k+1}, K^{k+1}, G^{k+1}, $\boldsymbol{s}^{k+1}$, $(\overline{p^w})^{k+1}$, $I_1^{\text{eff},k+1}$, $\sqrt{J_2^{k+1}}$, r^{k+1}, z_{eff}^{k+1}, $\varepsilon_v^{\text{p},k+1} \leftarrow$ consistencyBisection($\delta\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^{\text{p},k}$, $\boldsymbol{\sigma}^k$, K^k, G^k, $(\overline{p^w})^k$, ϕ^k, S_w^k, X^k, $\boldsymbol{s}^{\text{trial}}$, $I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, $\varepsilon_v^{\text{p,trial}}$, p_3^{trial}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, $i_{\max}$, $j_{\max}$, $\boldsymbol{\sigma}_{\text{fixed}}$, $\delta\boldsymbol{\varepsilon}_{\text{fixed}}^{\text{p}}$) ▷*The bisection return algorithm to take care of yield surface hardening.*
if iSuccess = FALSE **then**
  return isSuccess, $\boldsymbol{\sigma}^k$, $\boldsymbol{\varepsilon}^{\text{p},k}$, ϕ^k, S_w^k, X^k, $\boldsymbol{\alpha}^k$, K^k, G^k, p_3^k
end if
return isSuccess, $\boldsymbol{\sigma}^{k+1}$, $\boldsymbol{\varepsilon}^{\text{p},k+1}$, ϕ^{k+1}, S_w^{k+1}, X^{k+1}, $\boldsymbol{\alpha}^{k+1}$, K^{k+1}, G^{k+1}, p_3^{k+1}
end procedure

5.2.6 The nonhardening return algorithm

The nonhardening return algorithm pseudocode is listed below:

Algorithm 30 Non-hardening return algorithm

Require: $\boldsymbol{\sigma}^k$, $\delta\boldsymbol{\varepsilon}$, X^k, K^k, G^k, $(\overline{p^w})^k$, $\boldsymbol{s}^{\text{trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, `yieldCondition`
procedure nonHardeningReturn
  $r'_{\text{trial}} \leftarrow \beta\, r^{\text{trial}} \sqrt{\dfrac{3K^k}{2G^k}}$ ▷*Transform the trial r coordinate*
  $X_{\text{eff}}^k \leftarrow X^k + 3(\overline{p^w})^k$
  $z_{\text{eff}}^{\text{close}}$, $r'_{\text{close}} \leftarrow$ `yieldCondition`.getClosestPoint(K^k, G^k, X_{eff}^k, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, $z_{\text{eff}}^{\text{trial}}$, r'_{trial})
  $I_1^{\text{close}} \leftarrow \sqrt{3} z_{\text{eff}}^{\text{close}} - 3(\overline{p^w})^k$, $\sqrt{J_2^{\text{close}}} \leftarrow \frac{1}{\beta}\sqrt{\frac{G^k}{3K^k}}\, r'_{\text{close}}$
  if $\sqrt{J_2^{\text{trial}}} > 0$ **then**
    $\boldsymbol{\sigma}^{\text{fixed}} = \frac{1}{3} I_1^{\text{close}} \boldsymbol{I} + \frac{\sqrt{J_2^{\text{close}}}}{\sqrt{J_2^{\text{trial}}}} \boldsymbol{s}^{\text{trial}}$ ▷*Compute updated total stress*
  else
    $\boldsymbol{\sigma}^{\text{fixed}} = \frac{1}{3} I_1^{\text{close}} \boldsymbol{I} + \boldsymbol{s}^{\text{trial}}$ ▷*Compute updated total stress when the trial stress is hydrostatic*
  end if
  $\delta\boldsymbol{\sigma}_{\text{fixed}} \leftarrow \boldsymbol{\sigma}^{\text{fixed}} - \boldsymbol{\sigma}^k$ ▷*Compute stress increment*

$\delta\boldsymbol{\sigma}^{\text{iso}}_{\text{fixed}} \leftarrow \frac{1}{3}\text{tr}(\delta\boldsymbol{\sigma}_{\text{fixed}})\boldsymbol{I}, \quad \delta\boldsymbol{\sigma}^{\text{dev}}_{\text{fixed}} \leftarrow \delta\boldsymbol{\sigma}_{\text{fixed}} - \delta\boldsymbol{\sigma}^{\text{iso}}_{\text{fixed}}$

$\delta\boldsymbol{\varepsilon}^{\text{p,fixed}} = \delta\boldsymbol{\varepsilon} - \frac{1}{3K^k}\,\delta\boldsymbol{\sigma}^{\text{iso}}_{\text{fixed}} - \frac{1}{2G^k}\,\delta\boldsymbol{\sigma}^{\text{dev}}_{\text{fixed}}$ ▷*Compute plastic strain increment*

return $\boldsymbol{\sigma}^{\text{fixed}}$, $\delta\boldsymbol{\varepsilon}^{\text{p,fixed}}$

end procedure

Finding the closest point in transformed space

Algorithm 31 Compute the closest point from the trial state to transformed non-hardening yield surface

Require: K^k, G^k, X^k_{eff}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, $z^{\text{trial}}_{\text{eff}}$, r'_{trial}

procedure GETCLOSESTPOINT

$n_{\text{poly}} \leftarrow 1000$

$\mathbf{x}_{\text{poly}} \leftarrow$ GETYIELDSURFACEPOINTSALL_RPRIMEZ(n_{poly}, K^k, G^k, X^k_{eff}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β)
▷*Get the polygon that represents the yield surface in z_{eff}-r' space.*

$\mathbf{x}_{\text{seg}} \leftarrow$ GETCLOSESTSEGMENTS($z^{\text{trial}}_{\text{eff}}$, r'_{trial}, $\mathbf{x}_{\text{poly}}$)
▷*Find two yield surface segments that are closest to the trial stress state.*

$\mathbf{x}_{\text{segpoly}} \leftarrow$ GETYIELDSURFACEPOINTSSEGMENT_RPRIMEZ(n_{poly}, K^k, G^k, X^k_{eff}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, $\mathbf{x}_{\text{seg}}[1]$, $\mathbf{x}_{\text{seg}}[3]$) ▷*Discretize the two closest segments.*

$\mathbf{x}_{\text{close}} \leftarrow$ FINDCLOSESTPOINT($z^{\text{trial}}_{\text{eff}}$, r'_{trial}, $\mathbf{x}_{\text{segpoly}}$)
▷*Find the closest point in the discretized segments to the trial stress state.*

return isSuccess = TRUE, $\mathbf{x}_{\text{close}}.z_{\text{eff}}$, $\mathbf{x}_{\text{close}}.r'$

end procedure

Finding the yield surface polygon in z_{eff}-r' space

Algorithm 32 Find points in a closed polygon that describes the yield surface in z_{eff}-r' space

Require: n_{poly}, K^k, G^k, X^k_{eff}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β

procedure GETYIELDSURFACEPOINTSALL_RPRIMEZ

$\kappa \leftarrow I_1^{\text{peak}} - R_c(I_1^{\text{peak}} - X^k_{\text{eff}})$ ▷*Compute κ.*

$I_1^{\text{eff}} \leftarrow$ LINSPACE(`from` $= X^k_{\text{eff}}$, `to` $= I_1^{\text{peak}}$, `points` $= n_{\text{poly}}$)
▷*Create an equally spaced set of I_1^{eff} values.*

for I_1 **in** I_1^{eff} **do**

$F_f = a_1 - a_3\exp(a_2 I_1) - a_4 I_1$; ▷*Compute F_f.*

$F_c^2 \leftarrow 1$

if $I_1 < \kappa$ **and** $X^k_{\text{eff}} < I_1$ **then**

$F_c^2 = 1 - \left[\frac{\kappa - I_1}{\kappa - X^k_{\text{eff}}}\right]^2$; ▷*Compute F_c.*

end if

$J_2 = F_f^2 F_c^2$ ▷*Compute J_2 and push into a vector*

end for

$z_{\text{eff}} \leftarrow I_1^{\text{eff}}/\sqrt{3}$, $r' \leftarrow \beta\sqrt{\frac{3K^k}{2G^k}}\sqrt{2J_2}$

$\mathbf{x}_{\text{poly}}.z_{\text{eff}} \leftarrow z_{\text{eff}} \cup \text{REVERSE}(z_{\text{eff}})$, $\mathbf{x}_{\text{poly}}.r' \leftarrow r' \cup \text{REVERSE}(-r')$
▷*Add the points on the negative r' side of the polygon*

$\mathbf{x}_{\text{poly}}[2n_{\text{poly}} + 1] \leftarrow \mathbf{x}_{\text{poly}}[1]$ ▷*Add the first point to close the polygon*

return $\mathbf{x}_{\text{poly}}$

end procedure

Locating the closest segments of the yield surface polygon in z_{eff}-r' space

Algorithm 33 Locate two closest segments of the yield surface polygon to the trial stress state

Require: $z_{\text{eff}}^{\text{trial}}$, r'_{trial}, $\mathbf{x}_{\text{poly}}$
1: **procedure** GETCLOSESTSEGMENTS
2: $n_{\text{poly}} \leftarrow \text{LENGTH}(\mathbf{x}_{\text{poly}})$ ▷*Get the number of points in the closed polygon*
3: $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}_{\text{poly}}[n_{\text{poly}} - 1]$ ▷*Get the second to last point in the closed polygon*
4: $i_{\text{next}} \leftarrow 2$, $\mathbf{x}_{\text{next}} \leftarrow \mathbf{x}_{\text{poly}}[i_{\text{next}}]$ ▷*Get the second point in the closed polygon*
5: $d^2_{\text{min}} \leftarrow$ DOUBLE_MAX
6: $\mathbf{x}_{\text{min}} \leftarrow \mathbf{0}$, $\mathbf{x}^{\text{min}}_{\text{prev}} \leftarrow \mathbf{0}$, $\mathbf{x}^{\text{min}}_{\text{next}} \leftarrow \mathbf{0}$
7: **for** $\mathbf{x}_{\text{cur}}$ **in** $\mathbf{x}_{\text{poly}}$ **do**
8: $d^2 \leftarrow \text{DISTANCESQ}(z_{\text{eff}}^{\text{trial}}, r'_{\text{trial}}, \mathbf{x}_{\text{cur}}.z_{\text{eff}}, \mathbf{x}_{\text{cur}}.r')$
9: **if** $d^2 < d^2_{\text{min}}$ **then**
10: $d^2_{\text{min}} \leftarrow d^2$
11: $\mathbf{x}_{\text{min}} \leftarrow \mathbf{x}_{\text{cur}}$, $\mathbf{x}^{\text{min}}_{\text{prev}} \leftarrow \mathbf{x}_{\text{prev}}$, $\mathbf{x}^{\text{min}}_{\text{next}} \leftarrow \mathbf{x}_{\text{next}}$
12: **end if**
13: $i_{\text{next}} \leftarrow i_{\text{next}} + 1$
14: **if** $i_{\text{next}} = n_{\text{poly}}$ **then**
15: **break** ▷*Since the polygon is closed, ignore the last point*
16: **end if**
17: $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}_{\text{cur}}$, $\mathbf{x}_{\text{next}} \leftarrow \mathbf{x}_{\text{poly}}[i_{\text{next}}]$
18: **end for**
19: $\mathbf{x}_{\text{seg}} \leftarrow \{\mathbf{x}^{\text{min}}_{\text{prev}}, \mathbf{x}_{\text{min}}, \mathbf{x}^{\text{min}}_{\text{next}}\}$
20: **return** $\mathbf{x}_{\text{seg}}$
21: **end procedure**

Finding the segments of the surface polygon in z_{eff}-r' space

Algorithm 34 Find points in an open segment that describes the yield surface in z_{eff}-r' space

Require: n_{poly}, K^k, G^k, X^k_{eff}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, $\mathbf{x}_{\text{start}}$, $\mathbf{x}_{\text{end}}$
1: **procedure** GETYIELDSURFACEPOINTSSEGMENT_RPRIMEZ
2: $\kappa \leftarrow I_1^{\text{peak}} - R_c(I_1^{\text{peak}} - X^k_{\text{eff}})$ ▷*Compute κ.*
3: $I_1^{\text{start}} \leftarrow \sqrt{3}\mathbf{x}_{\text{start}}.z_{\text{eff}}$, $I_1^{\text{end}} \leftarrow \sqrt{3}\mathbf{x}_{\text{end}}.z_{\text{eff}}$
4: $I_1^{\text{eff}} \leftarrow \text{LINSPACE}(\texttt{from} = I_1^{\text{start}}, \texttt{to} = I_1^{\text{end}}, \texttt{points} = n_{\text{poly}})$
↪ ▷*Create an equally spaced set of I_1^{eff} values.*
5: **for** I_1 **in** I_1^{eff} **do**
6: $F_f = a_1 - a_3 \exp(a_2 I_1) - a_4 I_1$; ▷*Compute F_f.*
7: $F_c^2 \leftarrow 1$
8: **if** $I_1 < \kappa$ **and** $X^k_{\text{eff}} < I_1$ **then**
9: $F_c^2 = 1 - \left[\frac{\kappa - I_1}{\kappa - X^k_{\text{eff}}}\right]^2$; ▷*Compute F_c.*
10: **end if**
11: $J_2 = F_f^2 F_c^2$ ▷*Compute J_2 and push into a vector*
12: **end for**
13: $z_{\text{eff}} \leftarrow I_1^{\text{eff}}/\sqrt{3}$, $r' \leftarrow \beta\sqrt{\frac{3K^k}{2G^k}}\sqrt{2J_2}$
14: $\mathbf{x}_{\text{segpoly}}.z_{\text{eff}} \leftarrow z_{\text{eff}} \cup \text{REVERSE}(z_{\text{eff}})$, $\mathbf{x}_{\text{segpoly}}.r' \leftarrow r' \cup \text{REVERSE}(-r')$
↪ ▷*Add the points on the negative r' side of the polygon*
15: **return** $\mathbf{x}_{\text{segpoly}}$
16: **end procedure**

Finding the closest point on yield surface segments in z_{eff}-r' space

Algorithm 35 Find the closest point from the trial stress state on the polyline describing the yield surface

Require: $\mathbf{x}_{\text{trial}}.z_{\text{eff}}$, $\mathbf{x}_{\text{trial}}.r'$, $\mathbf{x}_{\text{segpoly}}$

procedure FINDCLOSESTPOINT
  $i \leftarrow 0$
  for $\{\mathbf{x}_{\text{segstart}}, \mathbf{x}_{\text{segend}}\}$ **in** $\mathbf{x}_{\text{segpoly}}$ **do**
    $\mathbf{x}_{\text{seg}} \leftarrow \mathbf{x}_{\text{segend}} - \mathbf{x}_{\text{segstart}}$
    $\mathbf{x}_{\text{proj}} \leftarrow \mathbf{x}_{\text{trial}} - \mathbf{x}_{\text{segstart}}$
    $t \leftarrow \frac{\mathbf{x}_{\text{proj}} \cdot \mathbf{x}_{\text{seg}}}{\|\mathbf{x}_{\text{seg}}\|}$
    $i \leftarrow i + 1$
    if $t < 0$ **then**
      $\mathbf{x}[i] \leftarrow \mathbf{x}_{\text{segstart}}$
    else if $t > 1$ **then**
      $\mathbf{x}[i] \leftarrow \mathbf{x}_{\text{segend}}$
    else
      $\mathbf{x}[i] \leftarrow \mathbf{x}_{\text{segstart}} + t\,\mathbf{x}_{\text{seg}}$
    end if
  end for
  $d^2_{\text{min}} \leftarrow$ DOUBLE_MAX
  $\mathbf{x}_{\text{close}} \leftarrow \mathbf{0}$
  for $\mathbf{x}_i$ **in** $\mathbf{x}$ **do**
    $d^2 \leftarrow$ DISTANCESQ($\mathbf{x}_i$, $\mathbf{x}_{\text{trial}}$)
    if $d^2 < d^2_{\text{min}}$ **then**
      $d^2_{\text{min}} \leftarrow d^2$
      $\mathbf{x}_{\text{close}} \leftarrow \mathbf{x}_i$
    end if
  end for
  return $\mathbf{x}_{\text{close}}$
end procedure

5.2.7 Consistency bisection algorithm

Algorithm 36 The consistency bisection algorithm for partially saturated materials

Require: $\delta\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^{\text{p},k}$, $\boldsymbol{\sigma}^k$, K^k, G^k, $(\overline{p^w})^k$, ϕ^k, S^k_w, X^k, $\boldsymbol{s}^{\text{trial}}$, $I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, r^{trial}, $z_{\text{eff}}^{\text{trial}}$, $\varepsilon_v^{\text{p,trial}}$, p_3^{trial}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β, i_{max}, j_{max}, $\boldsymbol{\sigma}_{\text{fixed}}$, $\delta\boldsymbol{\varepsilon}^{\text{p}}_{\text{fixed}}$, `yieldCondition`

procedure CONSISTENCYBISECTION
  $\delta\varepsilon_v^{\text{p,fixed}} \leftarrow \text{tr}(\delta\boldsymbol{\varepsilon}^{\text{p}}_{\text{fixed}})$
  $i \leftarrow 1$
  $\eta^{\text{in}} \leftarrow 0$, $\eta^{\text{out}} \leftarrow 1$
  while ABS($\eta^{\text{out}} - \eta^{\text{in}}$) > TOLERANCE **do**
    $j \leftarrow 1$
    isElastic ← TRUE
    while isElastic = TRUE **do**
      $\eta^{\text{mid}} \leftarrow \frac{1}{2}(\eta^{\text{in}} + \eta^{\text{out}})$
      $\delta\varepsilon_v^{\text{p,mid}} \leftarrow \eta^{\text{mid}}\delta\varepsilon_v^{\text{p,fixed}}$
      $(\overline{p^w})^{\text{mid}}, \phi^{\text{mid}}, S_w^{\text{mid}}, X^{\text{mid}} \leftarrow$ COMPUTEINTERNALVARIABLES(K^k, G^k, $(\overline{p^w})^k$, ϕ^k, S_w^k, X^k, $\delta\varepsilon_v^{\text{p,mid}}$) ▷*Update the internal variables using the bisected increment of the volumetric plastic strain*
      isElastic ← `yieldCondition`.EVALYIELDCONDITION($I_1^{\text{eff,trial}}$, $\sqrt{J_2^{\text{trial}}}$, X^{mid}, $(\overline{p^w})^{\text{mid}}$, ϕ^{mid}, S_w^{mid}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, β)

↪ ▷*Determine whether the trial stress is elastic or not*
if isElastic = TRUE **then**
$\eta^{\text{out}} \leftarrow \eta^{\text{mid}}$ ▷*If the local trial state is inside the updated yield surface, the yield condition evaluates to "elastic". We need to reduce the size of the yield surface by decreasing the plastic strain increment.*
$j \leftarrow j + 1$
if $j \geq j_{\max}$ **then**
return isSuccess ← FALSE ▷*The bisection algorithm failed because of too many iterations.*
end if
end if
end while
$\boldsymbol{\sigma}^{\text{new}}_{\text{fixed}}, \delta\boldsymbol{\varepsilon}^{\text{p,new}}_{\text{fixed}} \leftarrow$ nonHardeningReturn$(\boldsymbol{\sigma}^k, \delta\boldsymbol{\varepsilon}, X^{\text{mid}}, K^k, G^k, (\overline{p^w})^{\text{mid}}, \boldsymbol{s}^{\text{trial}}, \sqrt{J_2^{\text{trial}}}, r^{\text{trial}}, z^{\text{trial}}_{\text{eff}}, a_1, a_2, a_3, a_4, I_1^{\text{peak}}, R_c, \beta)$
↪ ▷*Compute return to updated yield surface (no hardening)*
if sign$(\text{tr}(\boldsymbol{\sigma}_{\text{trial}} - \boldsymbol{\sigma}^{\text{new}}_{\text{fixed}}))$ ≠ sign$(\text{tr}(\boldsymbol{\sigma}_{\text{trial}} - \boldsymbol{\sigma}_{\text{fixed}}))$ **then**
$\eta^{\text{out}} \leftarrow \eta^{\text{mid}}$ ▷*Too much plastic strain*
continue
end if
if $\|\delta\boldsymbol{\varepsilon}^{\text{p,new}}_{\text{fixed}}\| > \eta^{\text{mid}} \|\delta\boldsymbol{\varepsilon}^{\text{p}}_{\text{fixed}}\|$ **then**
$\eta^{\text{in}} \leftarrow \eta^{\text{mid}}$ ▷*Too little plastic strain*
else
$\eta^{\text{out}} \leftarrow \eta^{\text{mid}}$ ▷*Too much plastic strain*
end if
$i \leftarrow i + 1$
if $i \geq i_{\max}$ **then**
return isSuccess ← FALSE ▷*Too many iterations*
end if
end while
$\delta\varepsilon^{\text{p},k+1}_{v,\text{fixed}} \leftarrow \text{tr}(\delta\boldsymbol{\varepsilon}^{\text{p},k+1}_{\text{fixed}})$
$(\overline{p^w})^{k+1}, \phi^{k+1}, S_w^{k+1}, X^{k+1} \leftarrow$ computeInternalVariables$(K^k, G^k, (\overline{p^w})^k, \phi^k, S_w^k, X^k, \delta\varepsilon^{\text{p},k+1}_{v,\text{fixed}})$ ▷*Update the internal variables using the bisected increment of the volumetric plastic strain*
$\boldsymbol{\sigma}^{k+1} \leftarrow \boldsymbol{\sigma}^{\text{new}}_{\text{fixed}}$, $\boldsymbol{\alpha}^{k+1} \leftarrow -(\overline{p^w})^{k+1}\boldsymbol{I}$, $p_3^{k+1} \leftarrow p_3^{\text{trial}}$
$\boldsymbol{\varepsilon}^{\text{p},k+1} = \boldsymbol{\varepsilon}^{\text{p},k} + \delta\boldsymbol{\varepsilon}^{\text{p},k+1}$ ▷*Update the plastic strain*
$K^{k+1}, G^{k+1}, \boldsymbol{s}^{k+1}, (\overline{p^w})^{k+1}, I_1^{\text{eff},k+1}, \sqrt{J_2^{k+1}}, r^{k+1}, z^{k+1}_{\text{eff}}, \varepsilon_v^{\text{p},k+1} \leftarrow$ computeElasticProperties$(\boldsymbol{\sigma}^{k+1}, \phi^{k+1}, S_w^{k+1}, \boldsymbol{\varepsilon}^{\text{p},k+1}, \boldsymbol{\alpha}^{k+1}, p_3^{k+1})$
↪ ▷*Compute elastic moduli and stress invariants for the new state*
return isSuccess ← TRUE, $\boldsymbol{\sigma}^{k+1}, \boldsymbol{\varepsilon}^{\text{p},k+1}, \boldsymbol{\alpha}^{k+1}, (\overline{p^w})^{k+1}, \phi^{k+1}, S_w^{k+1}, X^{k+1}, K^{k+1}, G^{k+1}, \boldsymbol{s}^{k+1}, (\overline{p^w})^{k+1}, I_1^{\text{eff},k+1}, \sqrt{J_2^{k+1}}, r^{k+1}, z^{k+1}_{\text{eff}}, \varepsilon_v^{\text{p},k+1}, p_3^{k+1}$
end procedure

Updating the internal variables

Algorithm 37 Updating the internal variables for partially saturated materials

Require: $\boldsymbol{\sigma}^k, \varepsilon_v^{\text{p},k}, K^k, G^k, (\overline{p^w})^k, \phi^k, S_w^k, X^k, \delta\varepsilon_v^{\text{p}}$, `fluidParams`, `crushParams`, `airModel`, `waterModel`
procedure computeInternalVariables
$\overline{\varepsilon_v^{\text{p},k}} \leftarrow -\varepsilon_v^{\text{p},k}$, $\delta\overline{\varepsilon_v^{\text{p}}} \leftarrow -\delta\varepsilon_v^{\text{p}}$
$\overline{p_0^w} \leftarrow$ `fluidParams`.$\overline{p_0^w}$, $S_0 \leftarrow$ `fluidParams`.S_0, $\phi_0 \leftarrow$ `fluidParams`.ϕ_0, $p_1^{\text{sat}} \leftarrow$ `crushParams`.p_1^{sat}
$K_a \leftarrow$ `airModel`.computeBulkModulus$((\overline{p^w})^k)$

$K_w \leftarrow$ `waterModel`.computeBulkModulus($(\overline{p^w})^k$)
$\varepsilon_v^{a,0} \leftarrow$ `airModel`.computeElasticVolumetricStrain($\overline{p_0^w}$)
$\varepsilon_v^a \leftarrow$ `airModel`.computeElasticVolumetricStrain($(\overline{p^w})^k$)
$\varepsilon_v^w \leftarrow$ `waterModel`.computeElasticVolumetricStrain($(\overline{p^w})^k$, $\overline{p_0^w}$)
$\overline{\varepsilon_v^a} \leftarrow -(\varepsilon_v^a - \varepsilon_v^{a,0})$, $\overline{\varepsilon_v^w} \leftarrow -\varepsilon_v^w$
$\mathcal{C}_p \leftarrow S_0 \exp(\overline{\varepsilon_v^a} - \overline{\varepsilon_v^w})$
$\mathcal{D}_p \leftarrow \frac{1-S_0}{(1-S_0+\mathcal{C}_p)^2}$
$\frac{d\mathcal{C}_p}{d\overline{p^w}} \leftarrow \mathcal{C}_p\left[\frac{1}{K_a} - \frac{1}{K_w}\right]$
$\mathcal{G}_a \leftarrow \exp(\overline{\varepsilon_v^{p,k}} - \overline{\varepsilon_v^a})$, $\mathcal{G}_w \leftarrow \exp(\overline{\varepsilon_v^{p,k}} - \overline{\varepsilon_v^w})$
$\mathcal{B}_p \leftarrow \frac{1}{(1-S_0)\,\mathcal{G}_a + S_0\,\mathcal{G}_w}\left[-\frac{(1-\phi^k)\phi^k}{\phi_0}\left(\frac{S_w^k}{K_w} + \frac{1-S_w^k}{K_a}\right) + \frac{1-S_0}{K_a}\,\mathcal{G}_a + \frac{S_0}{K_w}\,\mathcal{G}_w\right]$
$(\overline{p^w})^{k+1} \leftarrow \max\left[(\overline{p^w})^k + \frac{1}{\mathcal{B}_p}\,\delta\varepsilon_v^p, 0\right]$ ▷*Update the pore pressure making sure that pressure does not become negative during dilatative plastic deformations.*
$\overline{X}_d, \frac{d\overline{X}_d}{d\varepsilon_v^p} \leftarrow$ computeDrainedHydrostaticStrengthAndDeriv($\overline{\varepsilon_v^{p,k}}$) ▷*Compute the drained hydrostatic compressive strength and its derivative*
$\overline{X}^{k+1} = -X^k + \left[(1 - S_w^k + p_1^{sat}\,S_w^k)\,\frac{d\overline{X}_d}{d\varepsilon_v^p} + \overline{X}_d\,(p_1^{sat} - 1)\,\frac{\mathcal{D}_p}{\mathcal{B}_p}\,\frac{d\mathcal{C}_p}{d\overline{p^w}} + \frac{3}{\mathcal{B}_p}\right]\delta\varepsilon_v^p$ ▷*Update the hydrostatic compressive strength*
$X^{k+1} \leftarrow -\overline{X}^{k+1}$
$\overline{\varepsilon_v^{p,k+1}} \leftarrow \overline{\varepsilon_v^{p,k}} + \delta\overline{\varepsilon_v^p}$ ▷*Compute the updated volumetric plastic strain.*
$\varepsilon_v^{a,k+1} \leftarrow$ `airModel`.computeElasticVolumetricStrain($(\overline{p^w})^{k+1}$)
$\varepsilon_v^{w,k+1} \leftarrow$ `waterModel`.computeElasticVolumetricStrain($(\overline{p^w})^{k+1}$, $\overline{p_0^w}$)
$\overline{\varepsilon_v^{a,k+1}} \leftarrow -(\varepsilon_v^{a,k+1} - \varepsilon_v^{a,0})$, $\overline{\varepsilon_v^{w,k+1}} \leftarrow -\varepsilon_v^{w,k+1}$ ▷*The updated strains in the fluid phases.*
$\mathcal{C}_p^{k+1} \leftarrow S_0 \exp(\overline{\varepsilon_v^{a,k+1}} - \overline{\varepsilon_v^{w,k+1}})$
$S_w^{k+1} \leftarrow \frac{\mathcal{C}_p^{k+1}}{1-S_0+\mathcal{C}_p^{k+1}}$ ▷*Update the saturation*
$\mathcal{G}_a^{k+1} \leftarrow \exp(\overline{\varepsilon_v^{p,k+1}} - \overline{\varepsilon_v^{a,k+1}})$, $\mathcal{G}_w^{k+1} \leftarrow \exp(\overline{\varepsilon_v^{p,k+1}} - \overline{\varepsilon_v^{w,k+1}})$
$\phi^{k+1} \leftarrow (1 - S_0)\phi_0\mathcal{G}_a^{k+1} + S_0\phi_0\mathcal{G}_w^{k+1}$ ▷*Update the porosity*
return $(\overline{p^w})^{k+1}$, ϕ^{k+1}, S_w^{k+1}, X^{k+1}
end procedure

Algorithm 38 Computing the drained hydrostatic strength and its derivative

Require: $\overline{\varepsilon_v^{p,k}}$, `fluidParams`, `crushParams`
procedure computeDrainedHydrostaticStrengthAndDeriv
$\phi_0 \leftarrow$ `fluidParams`.ϕ_0
$p_0 \leftarrow$ `crushParams`.p_0, $p_1 \leftarrow$ `crushParams`.p_1, $p_1^{sat} \leftarrow$ `crushParams`.p_1^{sat}, $p_2 \leftarrow$ `crushParams`.p_2
$p_3 \leftarrow -\log(1 - \phi_0)$
$\overline{X}_d \leftarrow \max(p_0, 1000)$; ▷$\overline{X}_d$ *has a minimum value of 1000 pressure units*
$\frac{d\overline{X}_d}{d\varepsilon_v^p} \leftarrow 0$
if $\overline{\varepsilon_v^{p,k}} > 0$ **then**
$\phi_{temp} \leftarrow \exp(-p_3 + \overline{\varepsilon_v^{p,k}})$
$\phi \leftarrow 1 - \phi_{temp}$
$\bar{\xi} \leftarrow p_1\ \mathrm{pow}\left(\frac{\phi_0}{\phi} - 1, \frac{1}{p_2}\right)$
$\overline{X}_d \leftarrow \overline{X}_d + \bar{\xi}$

12: $\dfrac{d\overline{X}_d}{d\varepsilon_v^{\mathrm{p}}} \leftarrow \frac{1}{p_2}\frac{\phi_0}{\phi}\phi_{\mathrm{temp}}\dfrac{\bar{\xi}}{\phi\left(\frac{\phi_0}{\phi}-1\right)}$
13: **end if**
14: **return** $\overline{X}_d$, $\dfrac{d\overline{X}_d}{d\varepsilon_v^{\mathrm{p}}}$
15: **end procedure**

5.2.8 Rate-dependent plastic update

Algorithm 39 Computing the correction to the stress due to rate-dependent plasticity

Require: Δt, $\boldsymbol{d}^{n+1}$, $\boldsymbol{\sigma}^n$, K^n, G^n, ϕ^n, S_w^n, X^n, $\boldsymbol{\alpha}^n$, $\boldsymbol{\varepsilon}^{\mathrm{p},n}$, p_3^n, $\boldsymbol{\sigma}_{qs}^n$, $\boldsymbol{\sigma}_{qs}^{n+1}$, K^{n+1}, G^{n+1}, ϕ^{n+1}, S_w^{n+1}, X^{n+1}, $\boldsymbol{\alpha}^{n+1}$, $\boldsymbol{\varepsilon}^{\mathrm{p},n+1}$, p_3^{n+1}, a_1, a_2, a_3, a_4, I_1^{peak}, R_c, `yieldParams`

1: **procedure** RATEDEPENDENTPLASTICUPDATE
2: $T_1 \leftarrow$ `yieldParams`.T_1, $T_2 \leftarrow$ `yieldParams`.T_2
3: **if** $T_1 = 0$ **or** $T_2 = 0$ **then** ▷*Check if rate-dependent plasticity has been turned on*
4: **return** isRateDependent ← FALSE, $\boldsymbol{\sigma}_{qs}^{n+1}$
5: **end if**
6: $K_{\mathrm{dyn}} \leftarrow \frac{1}{2}(K^n + K^{n+1})$, $G_{\mathrm{dyn}} \leftarrow \frac{1}{2}(G^n + G^{n+1})$ ▷*Compute mid-step bulk and shear modulus*
7: $\Delta\boldsymbol{\varepsilon} \leftarrow \Delta t\, \boldsymbol{d}^{n+1}$
8: $\boldsymbol{\sigma}_{\mathrm{trial,dyn}} \leftarrow$ COMPUTETRIALSTRESS($\boldsymbol{\sigma}^n$, K_{dyn}, G_{dyn}, $\boldsymbol{d}^{n+1}$, Δt) ▷*Compute substep trial stress*
9: $\dot{\varepsilon} \leftarrow$ MAX($\|\boldsymbol{d}^{n+1}\|$, ABS_DOUBLE_MIN)
10: $\tau \leftarrow T_1$ POW($\dot{\varepsilon}$, T_2) ▷*The characteristic time is defined from the rate-dependence input parameters and the magnitude of the strain rate*
11: $r_h \leftarrow \exp\left(-\frac{\Delta t}{\tau}\right)$
12: $R_H \leftarrow \frac{1-r_h}{\frac{\Delta t}{\tau}}$
13: $\boldsymbol{\sigma}^{n+1} \leftarrow \boldsymbol{\sigma}_{qs}^{n+1} + \left[(\boldsymbol{\sigma}_{\mathrm{trial,dyn}} - \boldsymbol{\sigma}^n) - (\boldsymbol{\sigma}_{qs}^{n+1} - \boldsymbol{\sigma}_{qs}^n)\right] R_H + (\boldsymbol{\sigma}^n - \boldsymbol{\sigma}_{qs}^n)\, r_h$ ▷*Stress update*
14: **return** $\boldsymbol{\sigma}^{n+1}$, isRateDependent ← TRUE
15: **end procedure**

6 Parameter fitting

The ARENA model parameters are fit in eight stages:

1. Stage 1: Fit the bulk modulus model for the dry soil using low strain-rate hydrostatic compression data. Fit a shear modulus model assuming a constant Poisson's ratio.
2. Stage 2: Fit the hydrostatic compressive strength model (crush curve) for the dry soil using low strain-rate hydrostatic compression data
3. Stage 3: Fit the limit surface model for the dry soil using low strain-rate traixial compression data
4. Stage 4: Fit the rate dependence parameters for the dry soil using high rate split Hopkison pressure bar (SHPB) data
5. Stage 5: Fit the damage model for the dry soil using high rate SHPB data
6. Stage 6: Fit the density-dependence model for the dry soil using high rate SHPB data
7. Stage 7: Fit the fully saturated hydrostatic strength parameter for the saturated soil using SPHB data for partially saturated soil.
8. Stage 8: Fit the parameter variability model using low strain-rate triaxial compression data.

These steps presume the idealized case that all required data are available, spanning ranges expected in applications. As is common for complex constitutive models, some hand fitting of parameters might be needed along with and engineering judgment for data extrapolation.

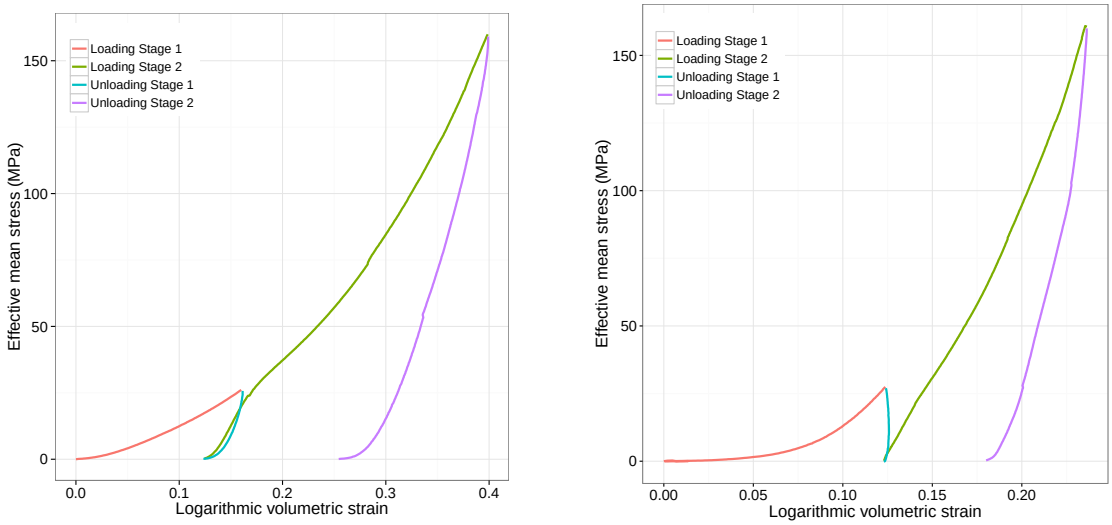
6.1 Data cleaning and processing

Experimental data are typically made available in spreadsheets that often contain metadata and test results (including computations) in a single sheet. These data should ideally be read, cleaned, and processed in a repeatable and systematic manner. We used the R programming language to process the data for parameter fitting.

The raw data also typically have several thousand data points per test. An optimization algorithm is typically used to fit the models to the data. To improve the efficiency of the optimization process, we resampled the data at 1000 equal intervals. The resampled data are essentially indistinguishable from the raw data at that resolution.

For the bulk modulus and crush curve models, loading and unloading segments must be extracted from hydrostatic stress-strain curves. To automate this process, central-differences based derivatives of the pressure-time curves were computed, and the points where the derivative of pressure changed sign were extracted and used to identify the segments corresponding to loading and unloading. These curves for dry Mason sand and Boulder clay are shown in Fig. 3. As mentioned in the introduction, creep and/or fluid seepage (especially pronounced in the clay data) must be removed (*e.g.,* by using a viscoelastic model) to obtain crush curve data appropriate for high-rate loading in which these effects are negligible. Ideally, quasistatic crush data should be taken at a variety of temperatures in order to convert these effectively isothermal crush curves to the adiabatic curves needed for high-rate loading – bringing rigor to this process is an excellent avenue for future research.

In general, the Levenberg-Marquardt algorithm does not produce adequate fits to the data. For example, parameters may be produced that lead to negative bulk moduli or yield functions that are not convex. To rectify this problem, a constrained optimization (minimization) algorithm is typically used to fit the data. The BFGS algorithm is a robust method that can be used to solve the minimization problem.



(a) Dry Mason sand.

(b) Dry Boulder clay.

Figure 3 – *Load-unload curves for sand and clay under hydrostatic compression.*

For example, when fitting the yield surface parameters, the minimization problem is posed as:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{argmin}} \quad & J(\boldsymbol{\theta}) = \tfrac{1}{2}\|\mathbf{f}(\boldsymbol{\theta};\mathbf{x}) - \mathbf{y}\|^2 \\ \text{such that} \quad & \boldsymbol{M}\cdot\boldsymbol{\theta} \geq \mathbf{0} \end{aligned} \tag{259}$$

where $\boldsymbol{\theta} = [a_1\ a_2\ a_3\ a_4]^T$, $\mathbf{x} = [I_1^{(1)}\ I_1^{(2)}\ \ldots\ I_1^{(m)}]^T$, $\mathbf{y} = [\sqrt{J_2}^{(1)}\ \sqrt{J_2}^{(2)}\ \ldots\ \sqrt{J_2}^{(m)}]^T$, and

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix}.$$

The function $\mathbf{f}(\boldsymbol{\theta};\mathbf{x})$ is the vector-valued yield function and $J(\boldsymbol{\theta})$ is the scalar-valued objective function. The gradients of the objective function with respect to the parameters are:

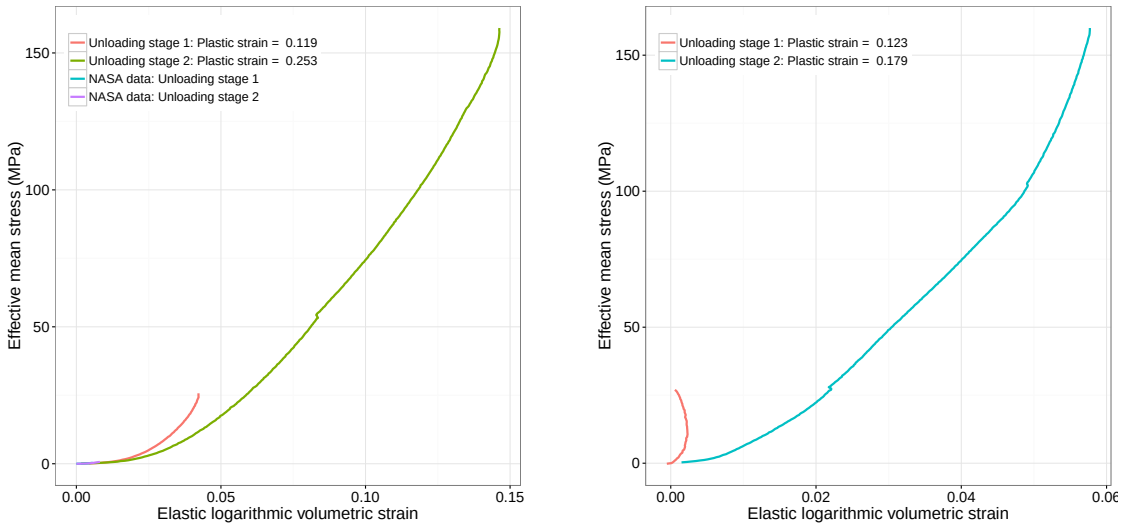
$$\nabla J = \frac{\partial J}{\partial \boldsymbol{\theta}} = [\mathbf{f}(\boldsymbol{\theta};\mathbf{x}) - \mathbf{y}]\cdot\frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}. \tag{260}$$

For fitting some of the parameters, such as those used to model rate-dependence, it is more convenient to use a package such as Dakota (Adams et al., 2009).

6.2 Stage 1: Bulk and shear modulus model parameters

After the unloading segments of the hydrostatic compression stress-strain curves have been identified as shown in Fig. 3, the elastic strain in the unloading segments has to be determined. The most straightforward way of doing this is to assume an additive decomposition of strain into elastic and plastic parts. The unrecoverable strain at zero stress is subtracted from the total strain to get the elastic volumetric strain. Plots of mean stress as a function of elastic volumetric strain are shown in Fig. 4.

The behavior observed in the figure can be modeled with a log-logistic curve that saturates at a value of $\bar{p} = K_s(\bar{p})$, where $\bar{p}$ is the pressure (mean stress) and K_s is the bulk modulus of the matrix solid



(a) Dry Mason sand.

(b) Dry Boulder clay.

Figure 4 – *The pressure as a function of the volumetric elastic strain for sand and clay. Note the creep that is apparent in the first unloading stage of the Boulder clay sample. Also note that the initial bulk moduli suggested by these data are very small.*

material (assumed to be granite in our case). The log-logistic function has the form

$$\frac{\bar{p}}{K_s(\bar{p})} = \frac{b_1}{b_2 + \left(\dfrac{\bar{\varepsilon}_v^e}{b_3}\right)^{-b_4}} \quad \text{where} \quad K_s(\bar{p}) = K_{s0} + n_s(\bar{p} - p_{s0}) \tag{261}$$

where $\bar{\varepsilon}_v^e$ is the elastic volumetric strain and $b_1 > 0, b_2 > 0, b_3 > 0, b_4 > 1$ are parameters that depend on the material. Note that the model (261) requires that $\bar{\varepsilon}_v^e > 0$ and is compressive. Equation (261) can be expressed as

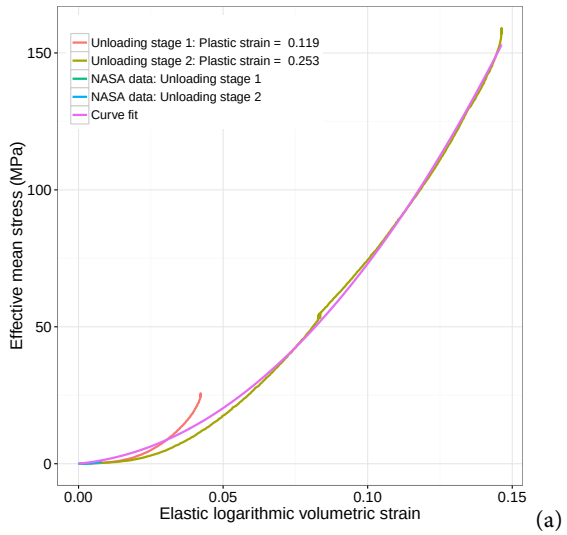
$$\frac{\bar{p}}{K_s(\bar{p})} = \frac{b_1}{b_2 + \left(\dfrac{b_3}{\bar{\varepsilon}_v^e}\right)^{b_4}} = \frac{b_1(\bar{\varepsilon}_v^e)^{b_4}}{b_2(\bar{\varepsilon}_v^e)^{b_4} + b_3^{b_4}} \tag{262}$$

Therefore, at $\bar{\varepsilon}_v^e = 0$ we have $\bar{p} = 0$. This form of the equation is more convenient for the computation of pressure because the singularity at zero strain can be avoided. For zero stress and tensile stress states, we can use the "near-zero" value of the bulk modulus given by this model. However, the bulk modulus predicted by this model is zero when $\bar{p} = 0$. We can rectify that problem by adding a linear term to the pressure model such that

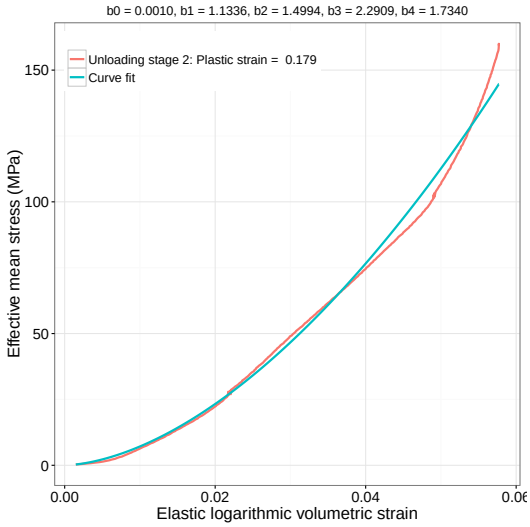
$$\frac{\bar{p}}{K_s(\bar{p})} = b_0\,\bar{\varepsilon}_v^e + \frac{b_1(\bar{\varepsilon}_v^e)^{b_4}}{b_2(\bar{\varepsilon}_v^e)^{b_4} + b_3} \tag{263}$$

where $b_0 > 0$. Dependence on plastic strain can be added to the model if needed. The disadvantage of this model is that there is no closed form expression for the volumetric elastic strain in terms of the mean stress (pressure).

Fig. 5 shows fits of the pressure model to the experimental data. The rapid increase in the bulk modulus at higher strains is deliberately not fitted closely because creep effects are dominant in this region. Also note that much larger volumetric strains are expected in numerical simulations than are reached in the experiments.



(a) Dry Mason sand.



(b) Dry Boulder clay.

Figure 5 – *Experimental data of pressure vs. elastic volumetric strain for sand and clay showing fits to the model in equation* (263).

The tangent bulk modulus is defined as

$$K(\bar{p}) := \frac{d\bar{p}}{d\bar{\varepsilon}_v^e}. \tag{264}$$

Then, from (263), we have

$$\left[\frac{1}{K_s} - \frac{n_s\bar{p}}{K_s^2}\right]\frac{d\bar{p}}{d\bar{\varepsilon}_v^e} = b_0 + \frac{b_1 b_3 b_4 (\bar{\varepsilon}_v^e)^{b_4-1}}{\left[b_2(\bar{\varepsilon}_v^e)^{b_4} + b_3\right]^2}. \tag{265}$$

Using definition (264), we have an expression for the bulk modulus of the soil:

$$K(\bar{p}) = \frac{K_s^2}{(K_s - n_s\bar{p})}\left[b_0 + \frac{b_1 b_3 b_4 (\bar{\varepsilon}_v^e)^{b_4-1}}{\left[b_2(\bar{\varepsilon}_v^e)^{b_4} + b_3\right]^2}\right]. \tag{266}$$

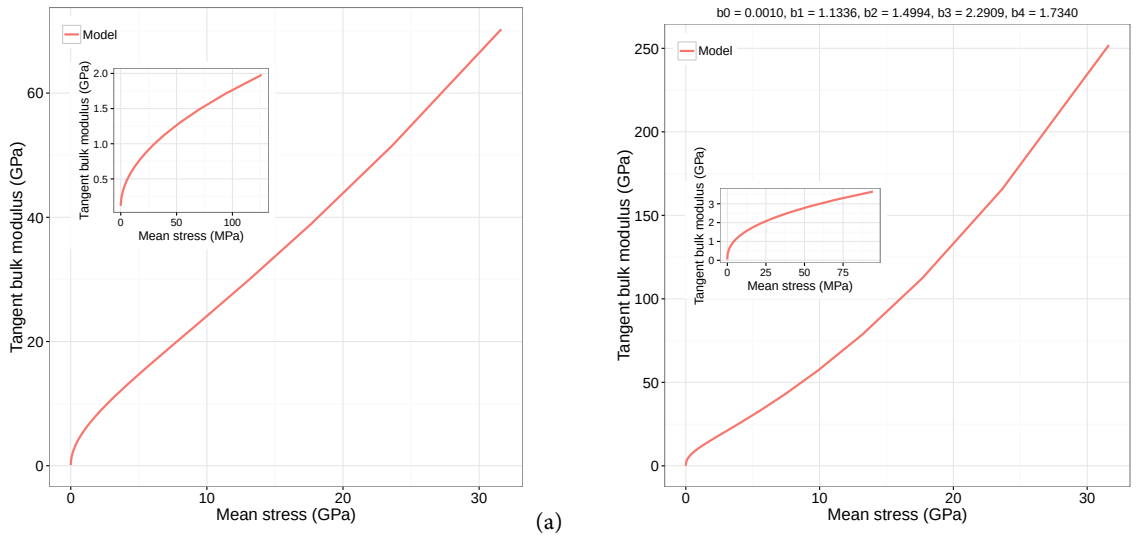
To express (266) in closed-form in terms of $\bar{p}$ we have to eliminate $\bar{\varepsilon}_v^e$. But a closed form expression for the volumetric elastic strain cannot be derived from the pressure model. So we find an approximate form of (263) by assuming $b_0 \to 0$. This approximation is valid at moderate to large strains. Then, from (263) with $b_0 = 0$, we have

$$\bar{\varepsilon}_v^e \approx \left[\frac{b_3\bar{p}}{b_1 K_s(\bar{p}) - b_2\bar{p}}\right]^{1/b_4} \tag{267}$$

and (266) can be expressed as

$$K(\bar{p}) = \frac{K_s^2}{(K_s - n_s\bar{p})}\left[b_0 + \frac{b_1 b_3 b_4 \left(\frac{b_3\bar{p}}{b_1 K_s(\bar{p}) - b_2\bar{p}}\right)^{1-1/b_4}}{\left[b_2\left(\frac{b_3\bar{p}}{b_1 K_s(\bar{p}) - b_2\bar{p}}\right) + b_3\right]^2}\right]. \tag{268}$$

The plots in Fig. 6 shows the tangent bulk moduli predicted by the model for Mason sand and Boulder clay. To confirm that the approximations introduced in equation (268) do not cause the predicted stress-strain behavior to deviate significantly from the experimental data, we can integrate the bulk modulus with respect to the volumetric strain to produce pressure vs. volumetric strain curves. The



(a) Dry Mason sand.

(b) Dry Boulder clay.

Figure 6 – *Predicted bulk modulus vs. pressure for dry Mason sand and Boulder clay using the model in* (268).

Table 1 – *Bulk modulus model parameters for Mason sand and Boulder clay*

Dry Mason sand

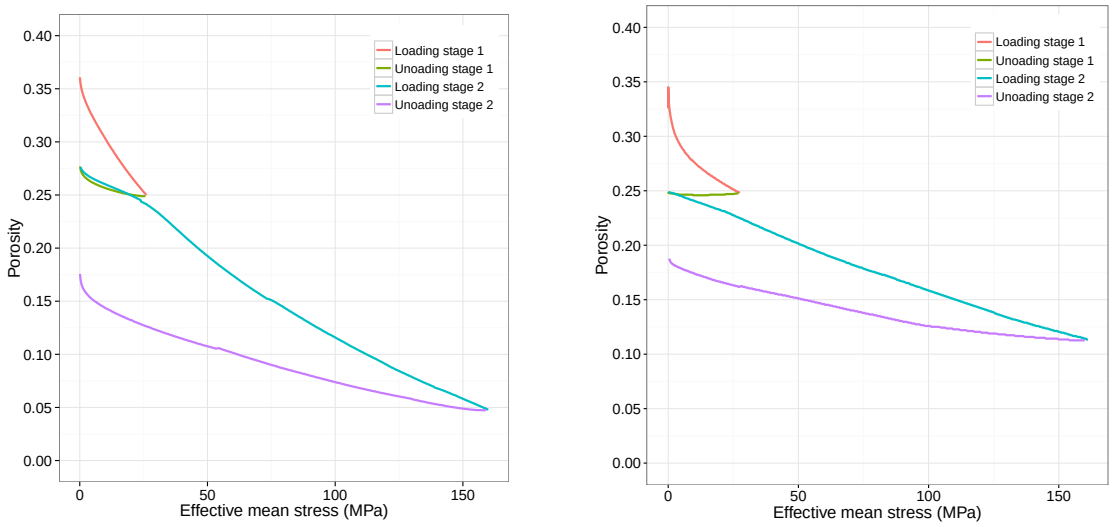
	b_0	b_1	b_2	b_3	b_4	p_{s0} (Pa)	K_{s0} (GPa)	n_s
Dry Mason sand	0.0029	0.4731	1.5057	2.5728	2.0799	101325	40	4
Dry Boulder clay	0.0010	1.1336	1.4994	2.2909	1.7340	101325	40	4

model reproduces the experimental stress-strain data for both Mason sand and Boulder clay reasonably well.

The bulk modulus model parameters for Mason sand and Boulder clay are shown in Table 1. For the shear modulus model, we assume a constant Poisson's ratio of 0.35 ($\nu_1 = 0.35$, $\nu_2 = 0$) for clay while for sand we use a Poisson's ratio that decreases from 0.35 to 0 ($\nu_1 = 0.35$, $\nu_2 = -0.35$) over the course of deformation.

6.3 Stage 2: Crush curve parameters

The experimental data typically contain a void ratio that can be converted into porosity. Plots of the porosity as a function of mean strain for the two *loading* steps in the hydrostatic compression test data for Mason sand and Boulder clay are shown in Fig. 7.



(a) Dry Mason sand.

(b) Dry Boulder clay.

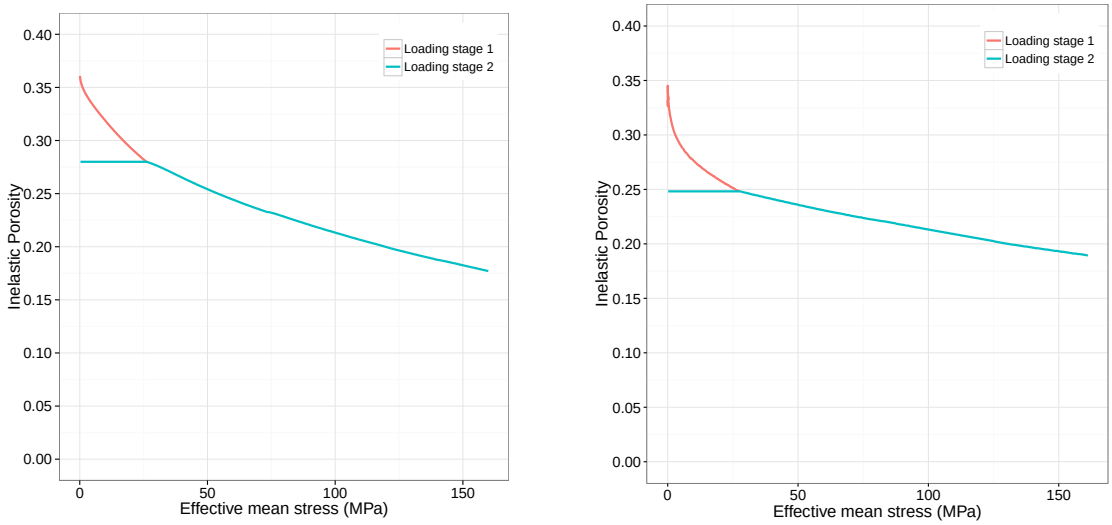
Figure 7 – *Porosity as a function of pressure for dry Mason sand and Boulder clay.*

From the figures we observe that:

1. The porosity is not constant at any point in the loading process in the first *loading* leg. This indicates that the *critical elastic limit pressure* beyond which there are changes in porosity in the sample is close to zero.
2. The porosity decreases during the first *unloading* leg, indicating that part of the change in porosity is elastic.
3. During the elastic stage of the second *loading* leg, the inelastic part of the change in porosity is again less than the total change in porosity.

For consistency with the simplifications we have made in the ARENA model, i.e. that the porosity depends only on the plastic volumetric strain, we have to remove the elastic part of the porosity from the total porosity before fitting the hydrostatic strength model. We use an approach similar to that used to determine volumetric plastic strains to compute the elastic and inelastic parts of the porosity during a loading step. Plots of the plastic part of the porosity as a function of pressure are shown in Fig. 8. These figures are called *crush curves*.

We assume (Brannon, Fuller, et al., 2015) the following relation between the rate of change of plastic



(a) Dry Mason sand.

(b) Dry Boulder clay.

Figure 8 – *Inelastic part of the porosity as a function of mean stress.*

porosity (Π) and the plastic volume strain (ε_v^p):

$$\frac{d\Pi}{dt} = (1-\Pi)\,\frac{d\varepsilon_v^p}{dt}. \tag{269}$$

In integral form, the above equation can be written as

$$\int_{\Pi_0}^{\Pi} \frac{1}{1-\Pi}\,\mathrm{d}\Pi = \int_0^{\varepsilon_v^p} \mathrm{d}\varepsilon_v^p\,. \tag{270}$$

Integration gives

$$\bar{\varepsilon}_v^p = \ln\left(\frac{1-\Pi}{1-\Pi_0}\right). \tag{271}$$

Alternatively, Let Π_0 be the initial porosity. Then the above equation has the solution

$$\Pi = 1-(1-\Pi_0)\exp(\bar{\varepsilon}_v^p)\,. \tag{272}$$

If we define p_3 such that

$$\exp(-p_3) := 1-\Pi_0 \tag{273}$$

we have

$$\Pi = 1-\exp(-p_3+\bar{\varepsilon}_v^p)\,. \tag{274}$$

Examination of the experimental data for dry Mason sand shows that the change in porosity is predicted very well by equation (272). The fit for Boulder clay is not as accurate, but the errors are typically less than 5%.

Next we relate the plastic porosity to the hydrostatic compressive strength ($\overline{X}$) to complete the crush curve. We assume the rational sigmoid form:

$$\Pi = \frac{\Pi_0}{1+p_1\overline{X}^{p_2}}. \tag{275}$$

where $\Pi_0 = 1-\exp(-p_3)$. Equating (274) and (275) gives us a relation between the plastic volumetric strain and the pressure along the hydrostat:

$$\frac{1-\exp(-p_3+\bar{\varepsilon}_v^p)}{1-\exp(-p_3)} = \frac{1}{1+p_1\overline{X}^{p_2}} \tag{276}$$

or

$$p_3 - \bar{\varepsilon}_v^p = -\ln\left[1 - \frac{1-\exp(-p_3)}{1+p_1\overline{X}^{p_2}}\right]. \tag{277}$$

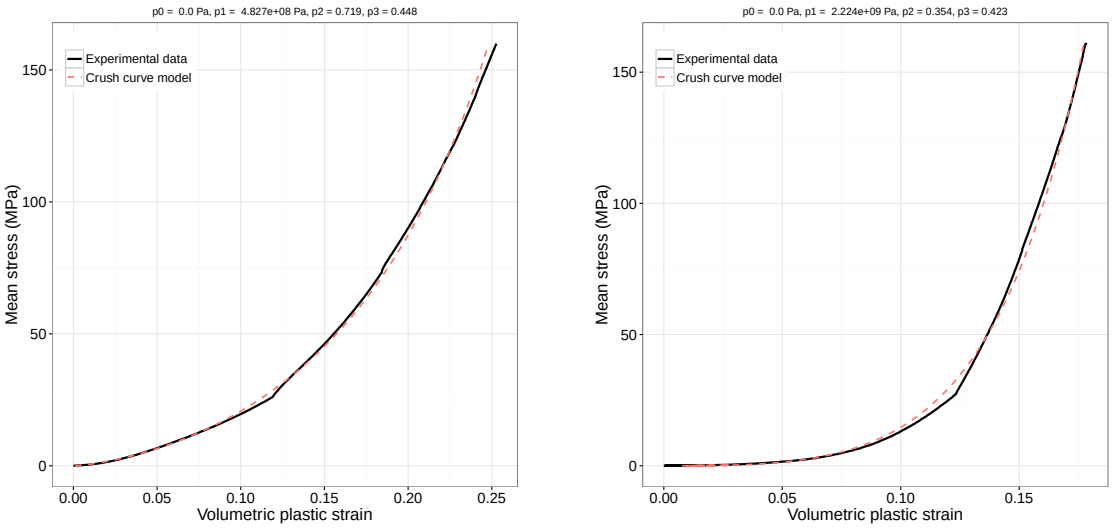
The inverse relationship is

$$\overline{X} = \left[\frac{1}{p_1}\left(\frac{1-\exp(-p_3)}{1-\exp(-p_3+\bar{\varepsilon}_v^p)} - 1\right)\right]^{1/p_2}. \tag{278}$$

For curve fitting purposes, we can use the right hand side of (276) to find the Jacobian needed by the minimization algorithm. The relevant derivatives are

$$\frac{dJ}{dp_1} = -\frac{\overline{X}^{p_2}}{\left[1+p_1\overline{X}^{p_2}\right]^2} \quad \text{and} \quad \frac{dJ}{dp_2} = -\frac{p_1\overline{X}^{p_2}\ln\overline{X}}{\left[1+p_1\overline{X}^{p_2}\right]^2}. \tag{279}$$

Fits to the experimental data using this model are quite accurate for the range of available data as can be seen in Fig. 9.



(a) Dry Mason sand.

(b) Dry Boulder clay.

Figure 9 – *Crush curve fits in mean stress vs. volumetric plastic strain space.*

The parameters fit to the dry Mason sand Boulder clay are listed below. in Table 2.

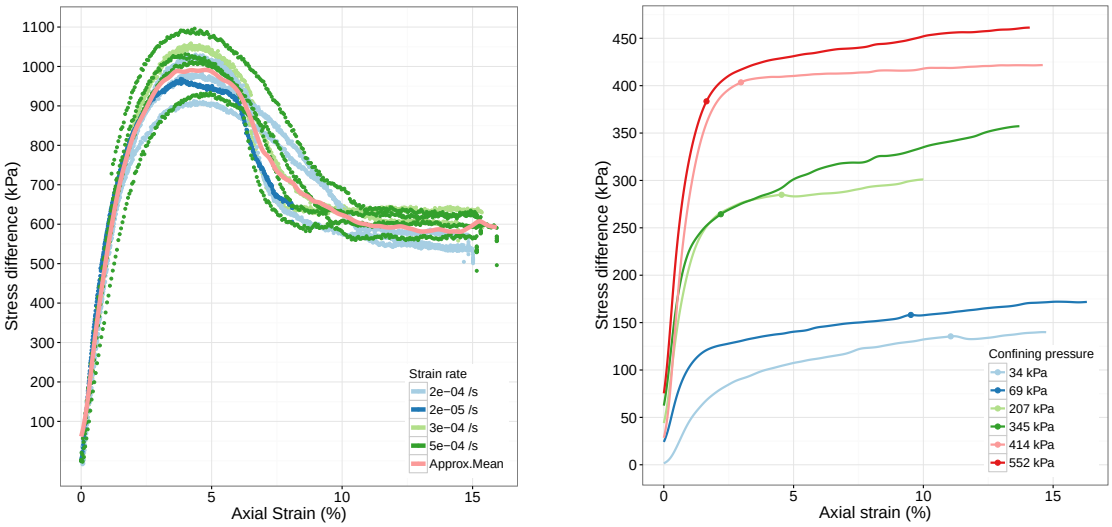
Table 2 – *Crush curve model parameters for dry Mason sand.*

	p_0 (Pa)	p_1 (MPa)	p_2	p_3
Dry Mason sand	0	482.7	0.719	0.448
Dry Boulder clay	0	2224	0.354	0.423

6.3.1 Stage 3: Yield function parameters

To determine the parameters for the quasistatic limit surface, we need triaxial stress data for a set of confining pressures. If such data are not available for a particular soil, we can use data for a comparable

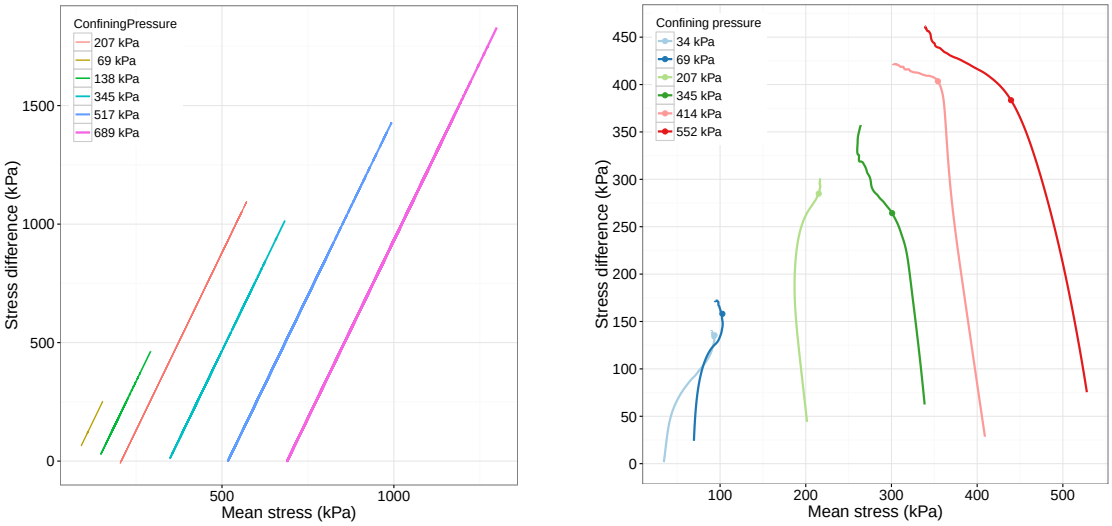
soil (which can be compared on the basis of density and grain size distribution). Typical plots of the stress difference as a function of the axial strain for consolidated sands are shown in Fig. 10(a). For Boulder clay, the equivalent plots are given in Fig. 10(b).



(a) Mason sand. (b) Boulder clay.

Figure 10 – *Difference between axial and radial stress as a function of axial stress from triaxial compression experiments.*

The data in Fig. 10 plotted in q-p space (stress difference vs. mean stress) is shown in Fig. 11. Note that the experimental data for dry Mason sand was collected only for a single confining pressure (207 kPa). These data have been augmented with data on a comparable sand from M. A. Thomas and Chitty, 2011. Also, note that some of the data for Boulder clay suggest that the limit surface may not have been reached.



(a) Mason sand. (b) Boulder clay.

Figure 11 – *Combined triaxial loading data for dry Mason sand (207 kPa) and a comparable sand from a different source (M. A. Thomas and Chitty, 2011) (left) and dry Boulder clay (right). Both sets of data are from consolidated undrained tests on nominally dry soils.*

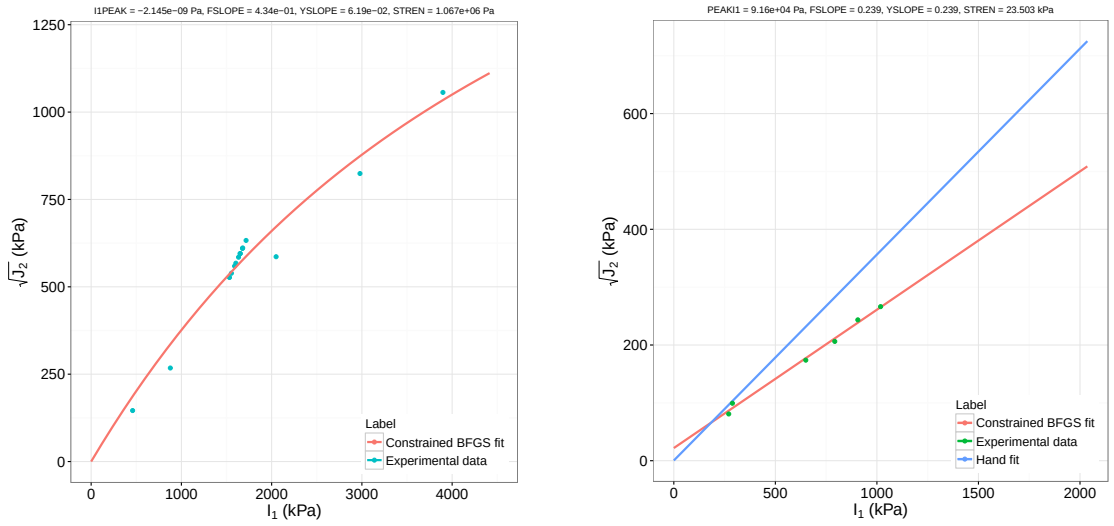
Recall that the ARENA yield function is defined as:

$$\sqrt{J_2} = [a_1 - a_3 \exp(-a_2 I_1)] + a_4 I_1 \tag{280}$$

where J_2 is the second invariant of the deviatoric stress and I_1 is the trace of the stress. We want to find the parameters a_1, a_2, a_3, and a_4. These parameters are related to I_1^{peak} (`PEAKI1`), f^{slope} (`FSLOPE`), $\sqrt{J_2^{\text{coh}}}$ (`STREN`), and y^{slope} (`YSLOPE`) by

$$\begin{aligned} &\alpha_1 := I_1^{\text{peak}}\,, \quad \alpha_2 := f^{\text{slope}}\,, \quad \alpha_3 := y^{\text{slope}}\,, \quad \alpha_4 := \sqrt{J_2^{\text{coh}}} \\ &a_1 = \alpha_4\,, \quad a_2 = \frac{\alpha_2 - \alpha_3}{\alpha_4 - \alpha_1 \alpha_3} \\ &a_3 = (\alpha_4 - \alpha_1 \alpha_3) \exp(-a_2 \alpha_1)\,, \quad a_4 = \alpha_3\,. \end{aligned} \tag{281}$$

We solve a constrained optimization problem to find the best fit to the experimental data in a least-squares sense. The limit surface is determined by the maximum values of the stress-difference that is achieved in the experiments. The fits to the resulting data points are shown in Fig. 12. Neither of the data sets in these plots is adequate for fitting a nonlinear limit surface. However, this lack of data is typical and the modeler has to take into consideration other experiments to make sure that the fit parameters are appropriate. For example, our attempts to simulate SHPB experiments with the fit parameters for the Boulder clay showed that the initial slope of the limit surface curve was too small, and a recursive fit to the SHPB provided a different set of parameters (shown in blue in the figure). Also, for numerical robustness, the value of I_1^{peak} is typically assumed to be greater than 1 kPa.



(a) Mason sand.

(b) Boulder clay.

Figure 12 – *Yield functions fitted to the combined triaxial loading data for sand and clay.*

We also need the cap ratio (R_c) and the non-associativity parameter (β) to define the limit surface. The cap ratio can be estimated using hydrostatic and uniaxial strain data. We use a value of 0.5 based on observations of uniaxial strain SHPB data. The non-associativity parameter can be determined after the rate-dependence parameters have been found by fitting experimental uniaxial strain loading curves. The parameters computed from the limit surface fitting process are listed in Table 3.

6.3.2 Stage 4: Rate-dependence parameters

Table 3 – *Yield condition parameters for dry Mason sand and Boulder clay*

	a_1 (kPa)	a_2 (kPa^{-1})	a_3 (kPa)	a_4	PEAKI1 (kPa)	FSLOPE	YSLOPE	STREN (kPa)	R_c	β
Dry Mason sand (hand fit)	556	5.49e-4	556	0.143	-1.23e-4 1	0.448 0.453	0.143 0.31	556 10000	0.5	2
Dry Boulder clay (hand fit)	23.5	2.77e-08	1600	0.239	91.6 1	0.239 0.356	0.239 0.355	23.5 17700	0.5	1

If the fitted rate-independent limit surface is used to simulate the response during a high-rate process such as a split Hopkinson bar test, the predicted stresses are much lower than their observed values. This common observation is the primary evidence motivating a high-rate component in the model.

To calibrate the ARENA model for high-rate processes we have to estimate the rate-dependence parameters T_1 and T_2.

We can use the loading-stage curves of axial and radial stress from SHPB tests to fit these parameters. The optimization problem is slight more complex in this case because the convex minimization problem now has the form

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{argmin}} \quad & J(\boldsymbol{\theta}) = \tfrac{1}{2}\left(\|\mathbf{f}_1(\boldsymbol{\theta};\mathbf{x}) - \mathbf{y}_1\|^2 + \|\mathbf{f}_2(\boldsymbol{\theta};\mathbf{x}) - \mathbf{y}_2\|^2\right) \\ \text{such that} \quad & \boldsymbol{M}\cdot\boldsymbol{\theta} \geq \mathbf{0} \end{aligned} \tag{282}$$

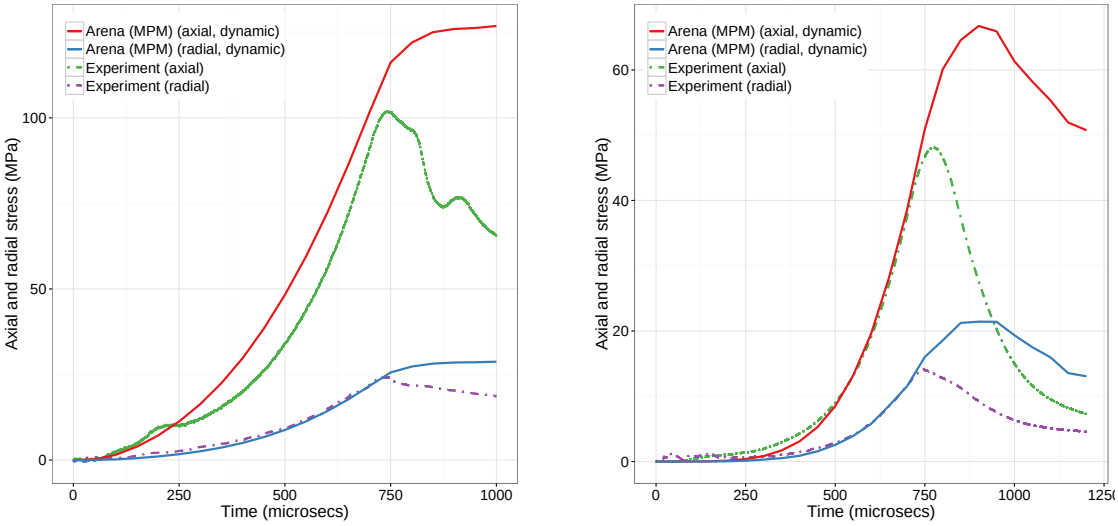
where $\boldsymbol{\theta} = \left[T_1\ T_2\right]^T$, $\mathbf{x} = \left[t^{(1)}\ t^{(2)}\ \ldots\ t^{(m)}\right]^T$, $\mathbf{y}_1 = \left[\sigma_{11}^{(1)}\ \sigma_{11}^{(2)}\ \ldots\ \sigma_{11}^{(m)}\right]^T$, $\mathbf{y}_2 = \left[\sigma_{22}^{(1)}\ \sigma_{22}^{(2)}\ \ldots\ \sigma_{22}^{(m)}\right]^T$, and $\boldsymbol{M}$ is the matrix defining the constraints on T_1 and T_2. Here, t is the time, σ_{11} is the axial stress, σ_{22} is the radial stress, and m is the number of samples. An alternative formulation of the problem is to remove the time-dependence and formulate the minimization problem in p-q space. However, some of the resolution of the objective function is lost if the time variable is removed from the objective function. Also note that the functions $\mathbf{f}_1$ and $\mathbf{f}_2$ are generated from simulations of a single particle with the material point method (MPM). The deformation of the particle is driven using deformation gradients computed from the experimental data. A two-level factorial design of experiments is used to narrow down the values of the parameters that are input as initial values to the optimization algorithm.

After the fit parameters have been found, we can used these for a single particle simulation. The predicted stress-time curves for the dry Mason sand calibration sample are shown in Fig. 13. The solid lines show the predicted stresses while the blue line show the experimental values. Fig. 14(a) and (b) show the fits to the data for dry Colorado Mason sand and dry Boulder clay. The rate-dependence parameters for the two materials are, for dry Colorado Mason sand, T_1 = 5.0e-5, T_2 = 0.5 and for dry Boulder clay, T_1 = 1.0e-6, T_2 = 0.5.

The ARENA model is essentially a specialized cap plasticity model (with enhancements for fluid effects and soil preparation), so its meridional yield profile is in the shape of an evolving teardrop. As depicted in Fig. 14, the movement of the stress state (blue dashed line) for experimental SHPB data is well captured with the ARENA model (red line); the sequence of evolving teardrop yield surfaces is controlled primarily by the cap hardening evolution equations.

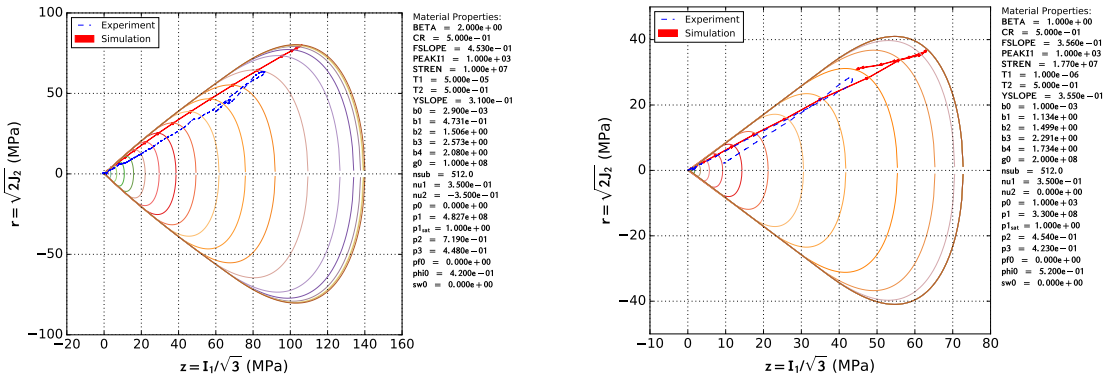
6.3.3 Stage 5: Damage model parameters

Damage model parameters can also be fit using the uniaxial-strain compression data from SHPB tests. However, there are two issues that have to be considered in this case. The first is that even though the region of interest is only the unloading part of the axial and radial stress-time curves, the damage model starts to affect the stress before the peak stress is attained. The second issue is that the ARENA model does not consider the possibility of a decrease in hydrostatic compressive strength after the peak stress has been reached. These factors preclude rigorously fitting of a curve by minimizing a convex objective function. In particular, ARENA will not predict the observed decrease in mean stress with increasing



(a) Dry Colorado Mason sand (b) Dry Boulder clay

Figure 13 – *Predicted and experimental stress-time curves for split-Hopkinson pressure bar uniaxial compression tests on dry sand and clay.*

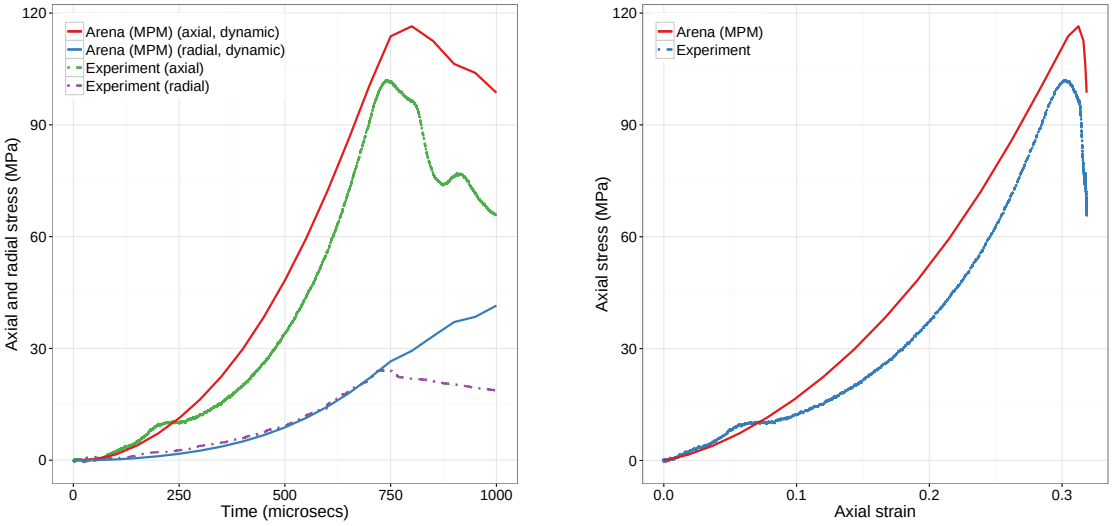


(a) Dry Colorado Mason sand. (b) Dry Boulder clay.

Figure 14 – *Dynamic stress in z-r-space – comparisons between predictions (solid) and experimental data (dashed) from split-Hopkinson pressure bar uniaxial compression tests.*

volumetric strain. Therefore, it is more convenient to use a design-of-experiments approach to find an estimate of the damage model parameters.

Fig. 15(a) compares the predicted stress-time curves (solid lines) and the experimental data (dashed lines) for dry Mason sand with an initial density of 1520 kg/m^3. Plots of the axial stress-strain curves and the same data are given in Fig. 15(b). The parameters used in the simulations were $f^{\text{speed}} = 15$ and $t_{\text{fail}} = 0.80\times10^{-3}$ s. The "failed" parameters of the yield surface were $I_1^{\text{peak,fail}} = 1.0\text{e-}5$ Pa, $f_{\text{slope}}^{\text{fail}} = 0.5$ f_{slope}, $y_{\text{slope}}^{\text{fail}} = 1.0\text{e-}5$, $\sqrt{J_2^{\text{coh,fail}}} = 0.5\sqrt{J_2^{\text{coh}}}$.



(a) Axial and radial stress vs. time

(b) Axial stress vs. axial strain.

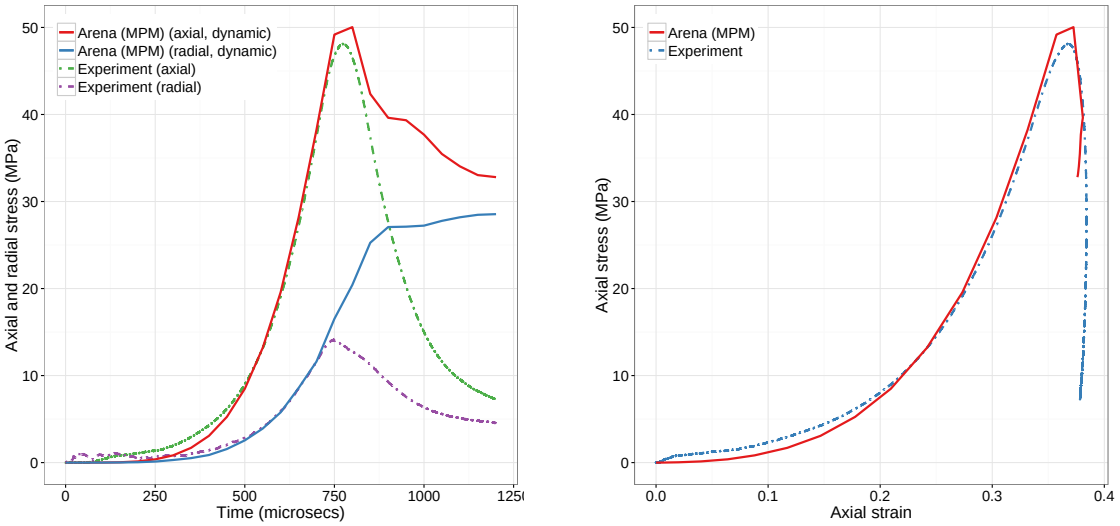
Figure 15 – *Dry Mason sand (a) stress-time curves predicted by the* Arena *model (with damage) and the experimental split-Hopkinson pressure bar data, and (b) axial stress vs. axial strain.*

Comparisons of the stress-time and stress-strain curves for a Boulder clay sample are shown in Fig. 16. The crush curve parameter p_1 had to be recalibrated to fit the experimental SHPB data. The new parameters are listed in Table 4. Though the initial slope of the stress-strain curves is underestimated by the model, the rapid increase in stress with time is predicted quite accuractely by Arena . The initial decrease in axial stress after the stress peak at 750 microseconds is also predicted by our model. However, the decrease in radial stress seen in the experiment is not seen in our simulation and the rapid drop in the axial stress is also not observed.

Table 4 – *Crush curve model parameters for Boulder clay.*

	p_0 (kPa)	p_1 (MPa)	p_1^{sat}	p_2	p_3
Dry Boulder clay	0	2224	0	0.354	0.423
Wet Boulder clay	1	330	-1	0.454	0.423

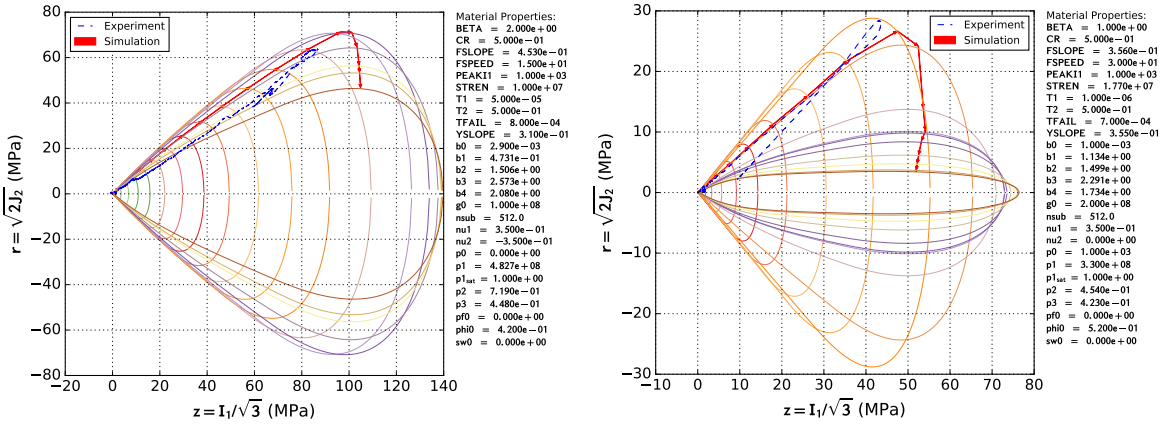
If we examine the yield surface evolution and the predicted stress paths in Fig. 17, we see that the slope of the load path is larger in the Arena simulation and, after failure, the stress path does not retrace the loading path. Instead, Arena predicts a load path that reflects the fact that the yield surface continues to grow along the hydrostatic axis. We are unsure whether the yield surface actually shrinks after failure in reality or whether the experimental data are reflecting an artifact of the SHPB test procedure. *It is*



(a) z vs. time (b) Axial stress vs. axial strain.

Figure 16 – *Dry Boulder clay (a) stress-time curves predicted by the ARENA model (with damage) and the experimental split-Hopkinson pressure bar data, and (b) axial stress vs. axial strain.*

unclear why the hydrostatic strength of the clay, even when fractured into a fine powder, should decrease as much as the experiments suggest since the material continues to be compressed.

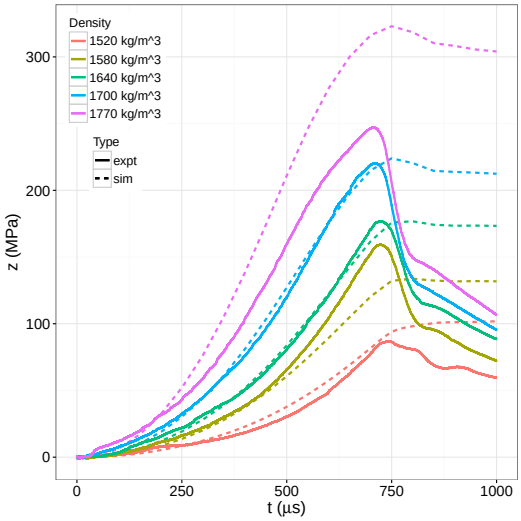


(a) Mason sand (b) Boulder clay

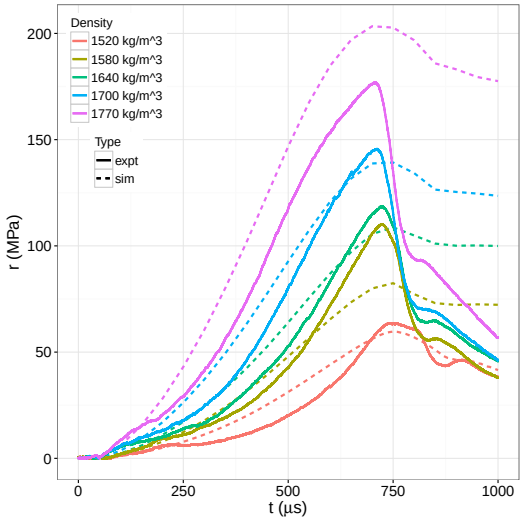
Figure 17 – *Comparison of ARENA model (with damage) and the experimental split-Hopkinson pressure bar data showing yield surface evolution and stress paths in z-r-space.*

6.3.4 Stage 6: Density-dependence parameters

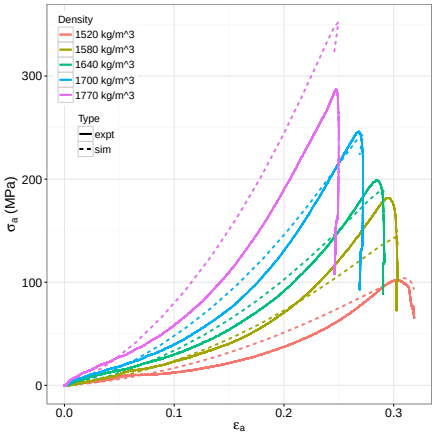
SHPB tests on dry Colorado Mason sand revealed a very strong dependence of the mechanical behavior of sands on initial density. To fit the ARENA density-dependence model, we choose uniaxial compressive strain SHPB data for Mason sand with initial densities of 1520 kg/m^3 and 1700 kg/m^3 and use the optimization procedure discussed earlier to find estimates of the model parameter ρ_{fac}. The reference porosity (ϕ_{ref}) is 0.42. We find that a fit parameter $\rho_{\text{fac}} = 5$ produces a reasonable fit to the experimental data as shown in Fig. 18. Predicted stress-strain curves and the stress path in z-r-space



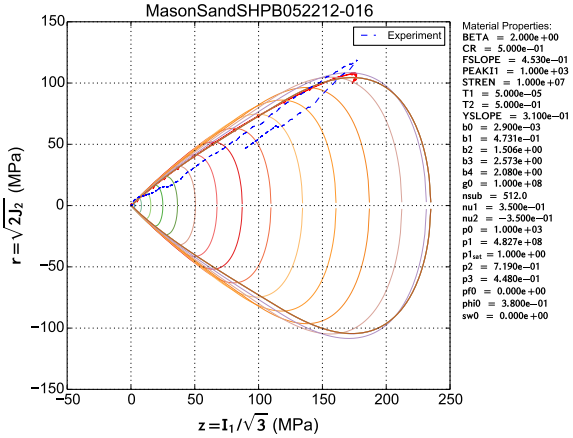
(a) z vs. time.



(b) r vs. time.



(c) Axial stress vs. axial strain.



(d) Mason sand density = 1640 kg/m^3

Figure 18 – *(a) Axial and (b) radial stress vs. time for sands of various initial densities: model vs. split-Hopkinson pressure bar experimental data on dry Mason sand, (c) axial stress vs. axial strain, (d) plot of stress path and yield surface in z-r space for a validation sample.*

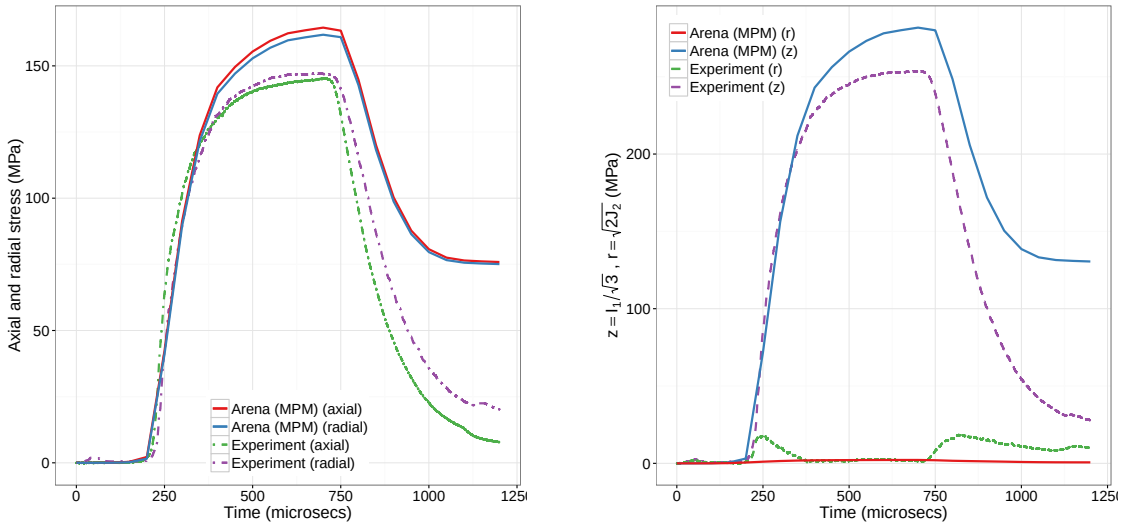
for a validation sample are shown in Fig. 18(c) and (d).

6.3.5 Stage 7: Fully saturated hydrostatic strength

For some soils, an additional parameter for the hydrostatic strength at full saturation (p_1^{sat}) has to be estimated. For instance, in the case of Colorado Mason sand, $p_1^{\text{sat}} = p_1$. However, we have found that a value of $p_1^{\text{sat}} > p_1$ is needed to fit the experimental SHPB data for saturated Boulder clay.

The Boulder clay sample used for the estimation of the parameter p_1^{sat} had a 40.8% water content by weight. The equations discussed in Appendix B were used to convert the weight of water into a saturation value. The resulting saturation was found to be greater than 1, indicating swelling of the clay. We assume in this section that the soil sample used for parameter estimation has a saturation of 90% ($S_w = 0.9$) after swelling, but that the porosity does not change significantly during this process. We also assume that tests on this sample can be used to estimate the fully saturated hydrostatic strength.

A design-of-experiments approach can be used to estimate a value of p_1^{sat}. In our case, $p_1^{\text{sat}} = 5$ produces the stress vs. time plots shown in Fig. 19. The predicted peak stresses are lower than those observed in the experiments. However, the shapes of the stress-time curves match experimental data remarkably well. We have also plotted the stresses in z-r-space in the figure. The z-r plots show that the z stress dominates the stress data and the deviatoric part is small enough to be negligible.



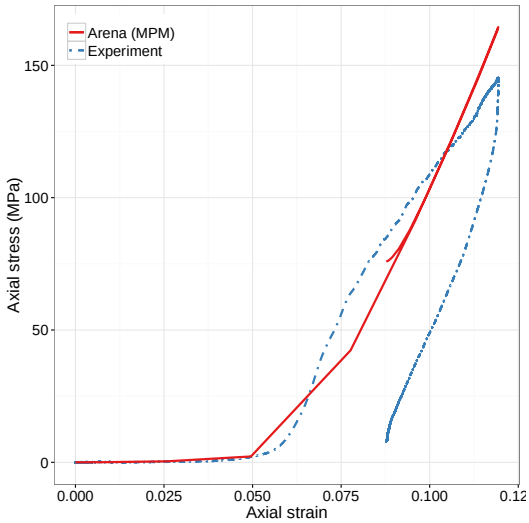
(a) Axial/radial stress vs. time

(c) z, r vs. time

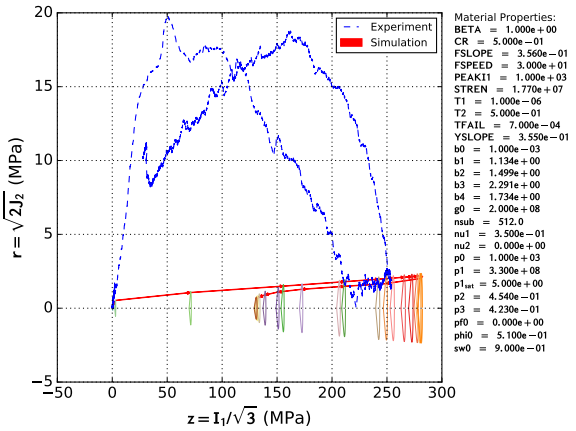
Figure 19 – *Comparisons of predicted axial, radial, mean, and deviatoric stress with experimental data from split-Hopkinson pressure bar uniaxial compression tests on 90% saturated Boulder clay.*

Plots of axial stress vs. axial strain and the stress path in z-r-space are given in Fig. 20. If we examine the stress-strain plot, we notice that the slope of the loading part of the curve is predicted reasonably well by Arena . However, the unloading curve appears to return along the loading path in the Arena simulation because of our choice of damage model. Clearly, the simple damage model used in Arena may not be performing adequately in this case even though it captures the unloading curves in Figure 19 reasonably well. If we look at the stress path in z-r stress space, the experiment appears to produce results that are not close to the Arena predictions. However, this is an artifact of the small values of the deviatoric stress in this experiment and probably indicates that the r-stresses are too small to be resolved by the instrumentation used in the SHPB experiments.

6.3.6 Stage 8: Yield parameter variability



(a) Axial stress vs. axial strain.



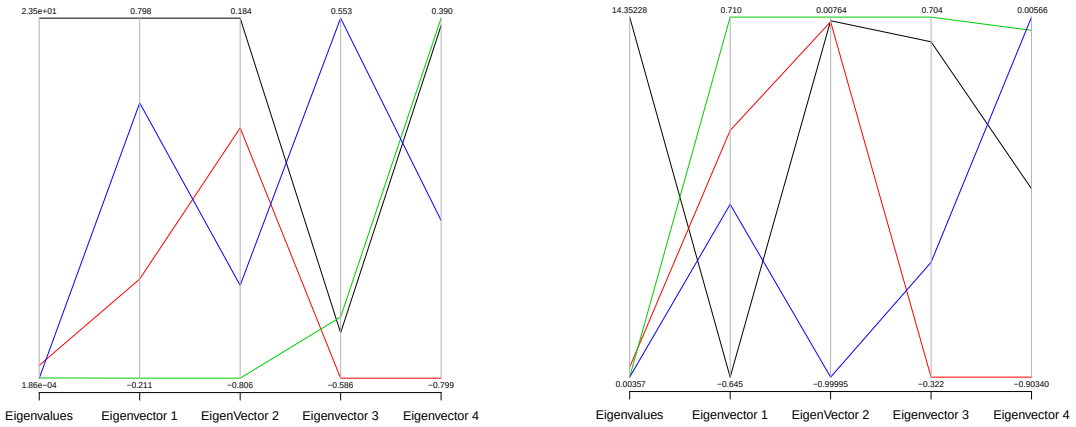
(b) Dynamic stress in $z - r$-space

Figure 20 – *Split-Hopkinson pressure bar: uniaxial: 90% saturated Boulder clay: yield surface.*

Statistical variation (as well as specimen scale effects) in yield parameter values from point-to-point in a soil should ideally be extracted from the parameter fitting process. This process has been applied successfully to modeling ceramics (Strack, Leavy, and Brannon, 2014), but experimental data is typically incomplete, implying that unique values of the fit parameters cannot be found(see Appendix E) . Lack of data, however, does not imply no *need* for data. Accordingly, any truly systematic experimental program should report not only median observations, but also repeatability of the data as quantified by statistically significant measurement of variance and scale effects.

During the parameter fitting process, we use on optimization algorithm that computes the Hessian on the objective function with respective to the fit parameters. The eigenvalues of these Hessian matrices can provide an indication of the confidence intervals that are appropriate for each parameter and whether a parameter is constrained at all.

Fig. 21 shows parallel coordinate plots of the eigen decomposition of the Hessians of the objective function at the fit points for the bulk modulus model. We can see from the plot that the eigenvalues of the Hessians for both materials are separated into two distinct sets (one high value and three low values) indicating that the parameters can be varied almost arbitrarily if we attempt to use the Hessian to compute confidence intervals. Alternate approaches are needed to determine the Weibull distribution parameters for the yield condition model parameters.



(a) Mason sand.

(b) Boulder clay.

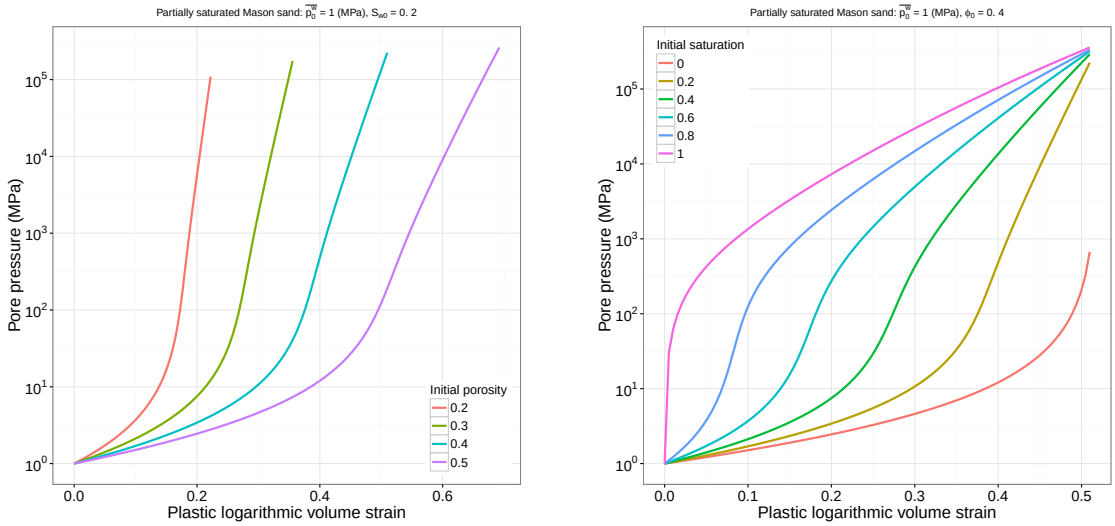
Figure 21 – *Parallel coordinate plots of the eigen decompositions of the Hessian of the objective function used to determine the parameters of the bulk modulus model.*

7 ARENA model behavior

This section illutrates how the internal variables in the model vary with changes in porosity and saturation. Analytical expressions for the hydrostatic strength, porosity, and saturation are used to compute the values predicted by the model. Since an analytical expression does not exist for the pore pressure, a forward Euler integration process (with a plastic strain increment of 0.001) is used to compute the pore pressure in the partially saturated soil model.

7.1 Pore pressure

Figure 22 shows the computed pore pressure as a function of the volumetric plastic strain. For a given initial porosity, the pore pressure increases with increasing volumetric plastic strain. The rate of increase of pore pressure is slow in the initial stages of compression but increases rapidly after an inflection point. A monotonic rate of increase is observed for small initial porosity values but the rate of increase is generally not monotonic. If we change the saturation but keep the initial porosity fixed, we observe that the pore pressure decreases as the degree of initial saturation is reduced. Note that even when the saturation is zero, a small pore pressure develops due to the compression of the air in the pores.



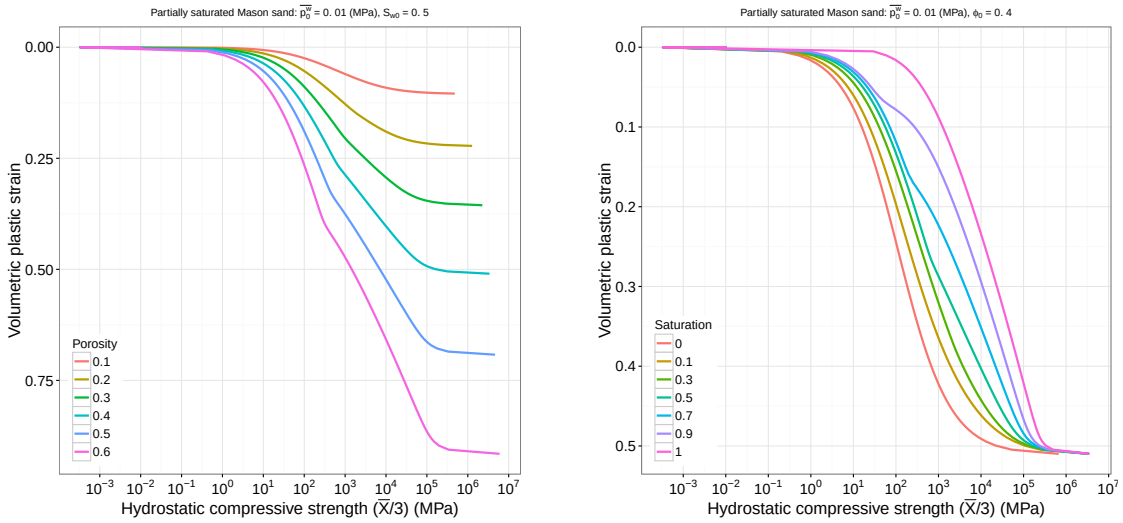
(a) Varying porosity (initial saturation is 0.2)

(b) Varying saturation (initial porosity is 0.4)

Figure 22 – *ARENA predictions of the pore pressure as a function of plastic volumetric strain.*

7.2 Hydrostatic compressive strength

The effect of varying porosity and saturation on the hydrostatic compressive strength is shown in Figure 23. The plots have been presented in the form commonly used for consolidation curves for soils (the volumetric plastic strain is a proxy for the void ratio). We observe that ARENA predicts that the hydrostatic compressive strength decreases with increasing porosity and the curves contain an inflection point where the slope of the plastic strain - compressive strength curve changes rapidly. ARENA also predicts that when the volumetric plastic strain exceeds a threshold (whose value depends on the initial porosity) the hydrostatic strength increases without any change in the plastic strain. If we keep the initial porosity constant and increase the saturation, we observe that the strain-stress curves shift to the right indicating an increase in strength. Both sets of plots in the figure are largely according to intuition. The inflection points in the curves are cause by the compression of the air in the voids and are absent if the material is either fully saturated or dry.

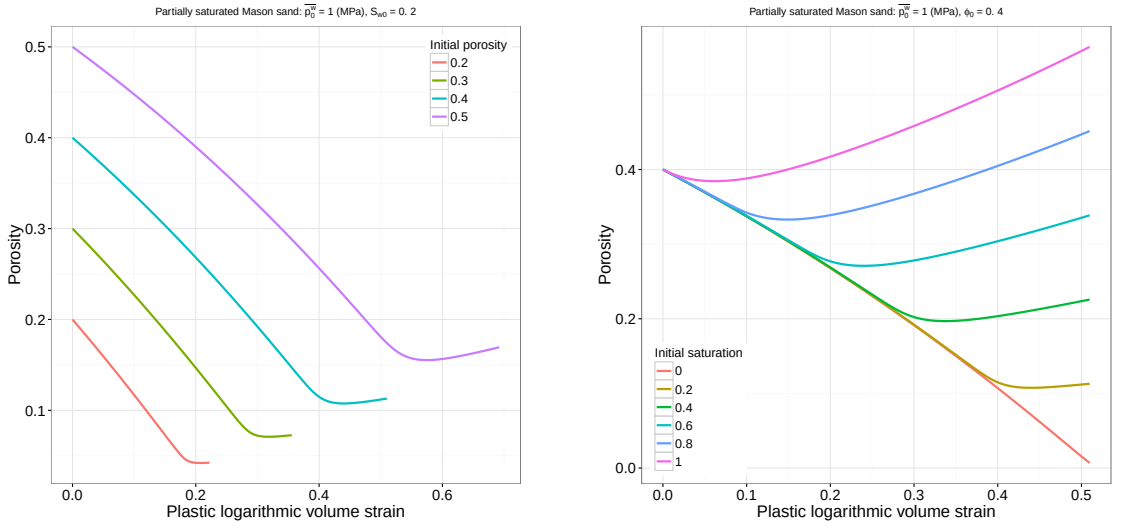


(a) Varying porosity (initial saturation is 0.5) (b) Varying saturation (initial porosity is 0.4)

Figure 23 – *ARENA soil hydrostatic compressive strength as a function of plastic volumetric strain.*

7.3 Porosity

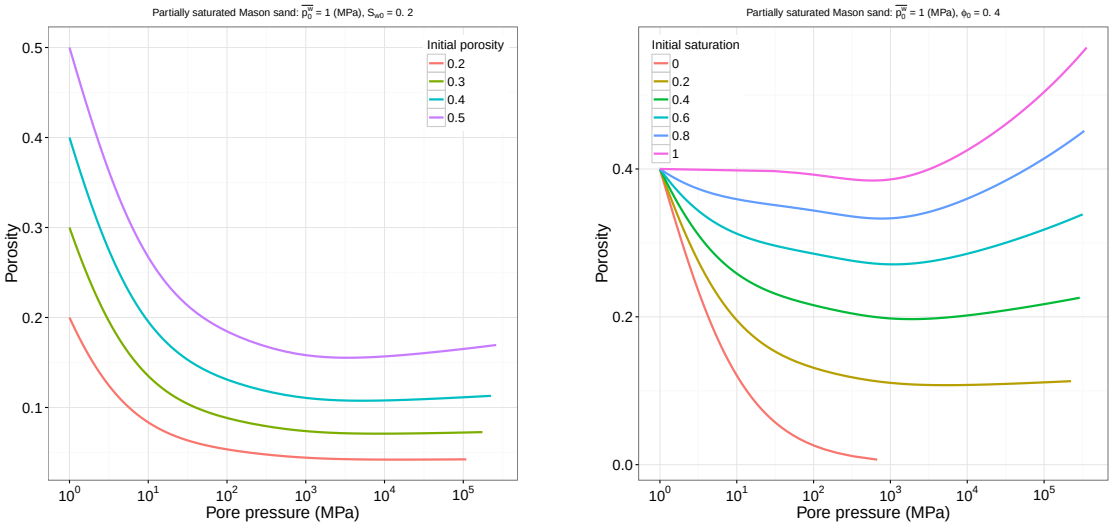
Figure 24 shows the porosity predicted by ARENA as a function of the volumetric plastic strain for various initial values of porosity and saturation. Notice that the porosity decreases as the material is compressed. However, beyond a compression threshold, the porosity starts increasing due to the presence of the pore fluids. This effect becomes clearer when we vary the saturation while keeping the initial porosity fixed. In that case we notice that if the initial saturation is zero, the material continues to be compressed without any increase in porosity. However, if the material is initially fully saturated, the porosity decreases momentarily and then starts increasing. It is unclear whether this is a physically observed effect or just an artifact of the assumptions we have made in the ARENA model.



(a) Varying porosity (initial saturation is 0.2) (b) Varying saturation (initial porosity is 0.4)

Figure 24 – *Porosity predicted by ARENA as a function of plastic volumetric strain.*

To explore the above behavior further and the determine the effect of pore pressure, we can plot the porosity as a function of pore pressure as shown in Figure 25. The pore pressure increases monotonically with the plastic volumetric strain, which is the expected behavior of the model.

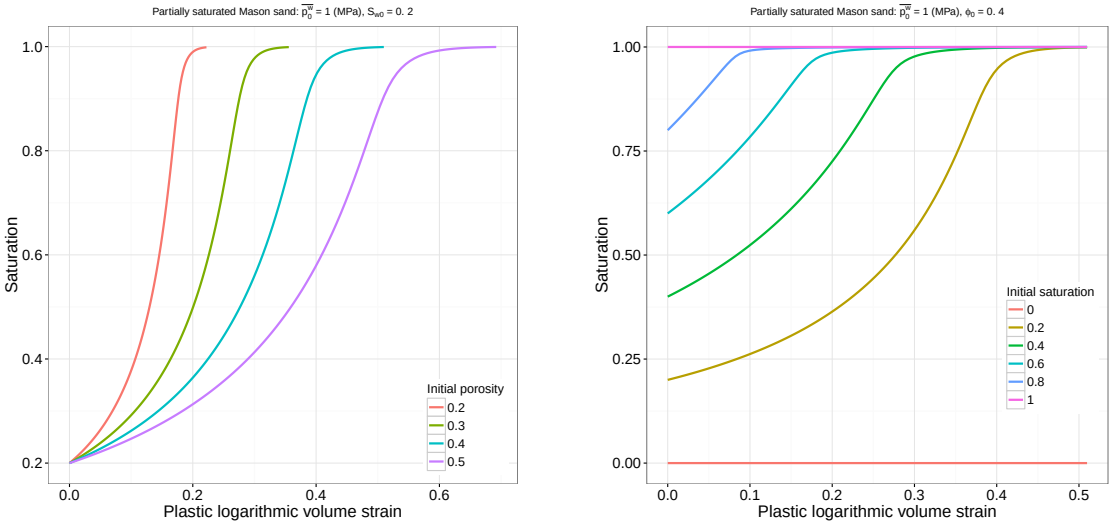


(a) Varying initial porosity (initial saturation is 0.2) (b) Varying initial saturation (initial porosity is 0.4)

Figure 25 – *Porosity predicted by ARENA as a function of pore pressure.*

7.4 Saturation

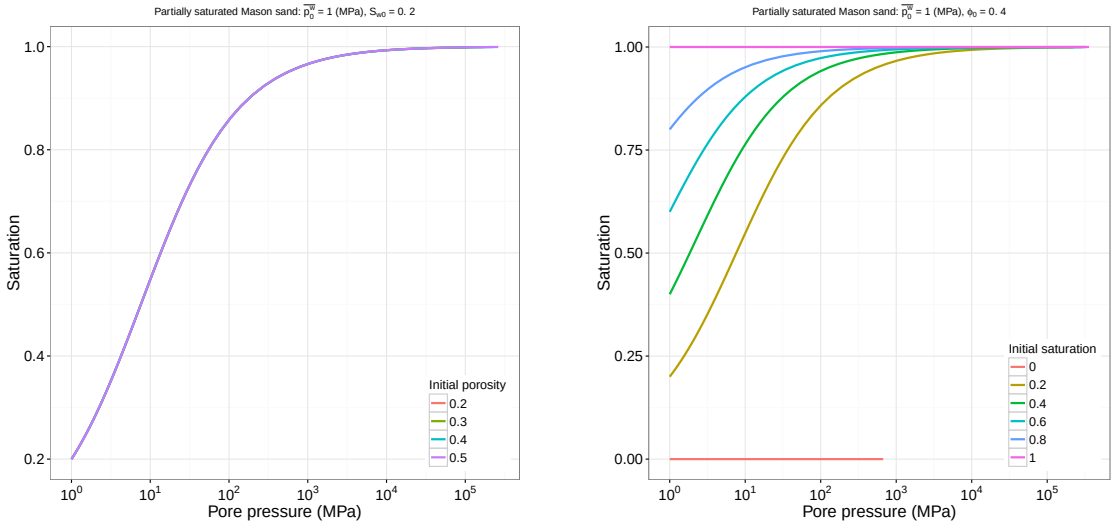
Plots of the saturation as a function of volumetric plastic strain are shown in Figure 26. The saturation increases rapidly until it reaches the limiting value of 1. When the initial saturation is 0 or 1, we do not observe any change in saturation with increasing compression.



(a) Varying initial porosity (initial saturation is 0.2) (b) Varying initial saturation (initial porosity is 0.4)

Figure 26 – *Saturation as a function of plastic volumetric strain (varying initial porosity).*

Alternatively, if we plot the saturation as a function of pore pressure, we get the behavior shown in Figure 27. If we start with the same initial saturation, the pore pressure vs. saturation curves do not change depending on the initial porosity. Also, the saturation rapidly reaches the limiting value of 1 as the pore pressure increases.



(a) Varying initial porosity

(b) Varying initial saturation

Figure 27 – *Saturation as a function of pore pressure (varying initial porosity).*

8 ARENA model verification with MPM

Typical verification tests for pressure-dependent nonlinear plasticity models for granular material include hydrostatic compression, uniaxial compression, uniaxial tension, and multiaxial strain loading paths, all of which are affine (single element) deformations. A momentum balance solver (in this case the material point method - MPM) and a complete constitutive model implementation for the momentum solver are used to compute predicted values of stresses, plastic and elastic strains, and internal variables in non-affine deformations. As summarized in Brannon, Fuller, et al., 2015, predicted values may then either be compared with model predictions from, for example, the bulk modulus/crush curve model, trend tests, or manufactured solutions (*c.f.* Kamojjala et al., 2015).

In the case of an Arena implementation, preliminary verification tests can be performed using the some of the tests suggested by Brannon and co-workers (Kamojjala et al., 2015). However, exact verification of the complete Arena model is nontrivial and manufactured solutions have not yet been developed for the model. In this section we discuss a few verification tests that, while not exact comparisons, provide confidence that the model has been implemented without major errors.

This section provides evidence of passing the following verification tests:

1. Hydrostatic compressive loading-unloading of a dry sand
2. Hydrostatic loading of a fully saturated sand
3. Uniaxial compressive loading of a dry sand
4. Uniaxial compressive loading of a fully saturated sand
5. Uniaxial tensile loading of a dry sand
6. Multiaxial loading-unloading of a dry sand

8.1 Hydrostatic compression loading and unloading: Dry sand

The hydrostatic loading-unloading simulations shown in this section show that our code is able to reproduce the bulk modulus and crush curve models in ARENA . Figure 28 shows the loading path in *z*-*r* stress space for hydrostatic loading followed by unloading.

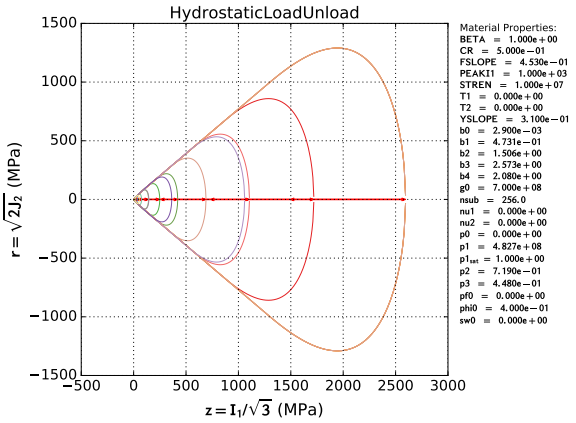
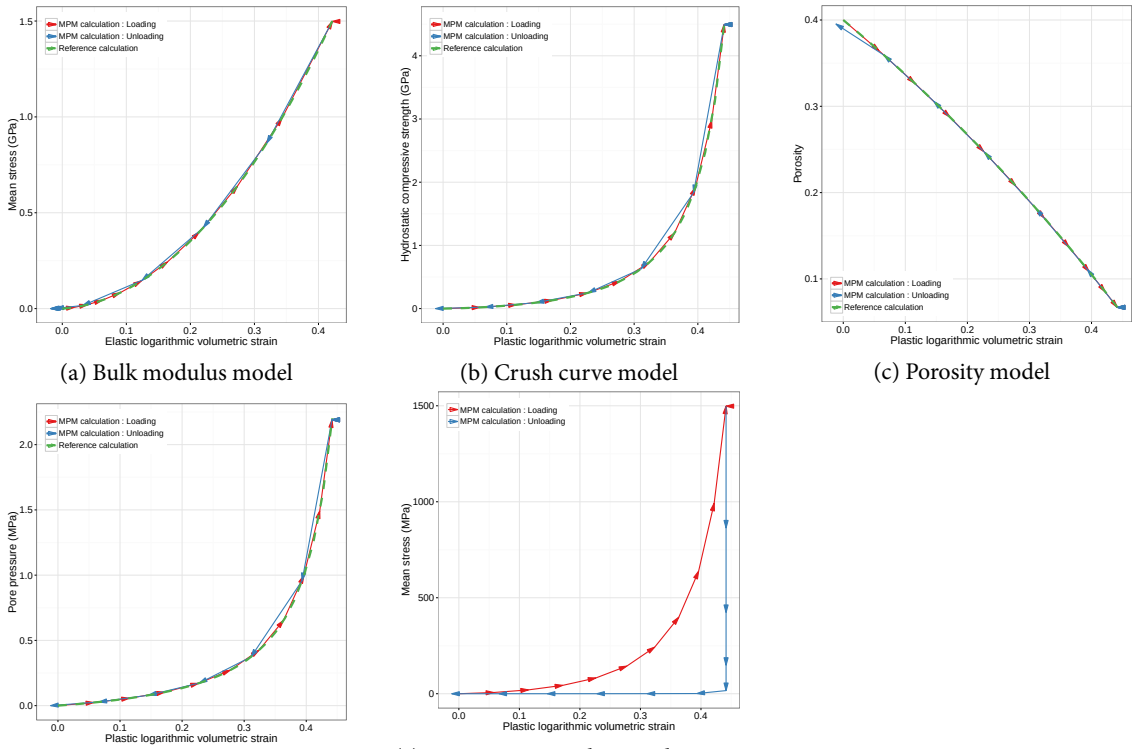


Figure 28 – *Verification of loading path during a hydrostatic load-unload cycle on a single particle of dry sand.*

The implementation of the bulk modulus model can be seen to be correct in Figure 29(a). Similarly, the crush curve and porosity models are also reproduced accurately by this verification test. The porosity does not recover completely during the unloading process because the plastic volumetric strain, by

design, reaches a dilatative state at the end of the cycle. These tests indicate that the implementation of the plasticity algorithm and model is reasonably accurate for dry soils.



(a) Bulk modulus model

(b) Crush curve model

(c) Porosity model

(d) Pore pressure model

(e) Mean stress vs. plastic volumetric strain

Figure 29 – *Verification of implementation of bulk modulus, crush curve, porosity and pore-pressure models in* ARENA *using single-particle hydrostatic loading-unloading of dry sand.*

8.2 Hydrostatic compression loading: Fully saturated sand

Results from the hydrostatic compression of a fully saturated single particle of ARENA sand are shown in this section. Figure 30 shows the evolution of the yield surface as compression proceeds. The yield surface translates and grows with increasing pore pressure.

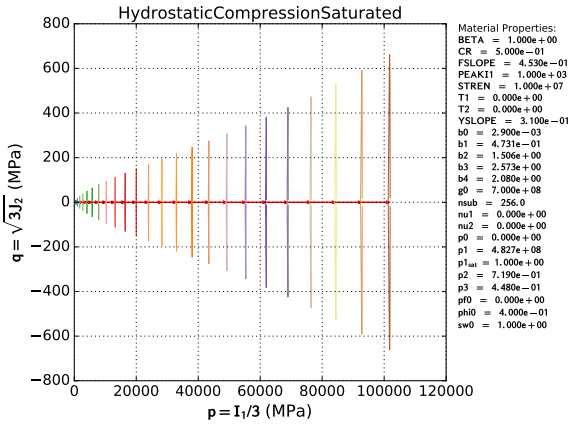
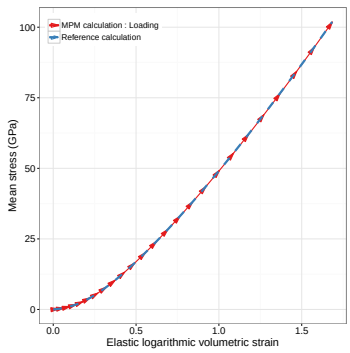
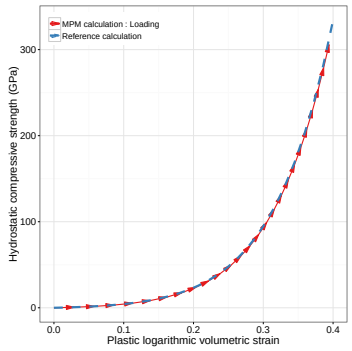


Figure 30 – *Evolution of the yield surface and the stress path in p-q space during hydrostatic compressive strain loading of a fully saturated particle of ARENA sand.*

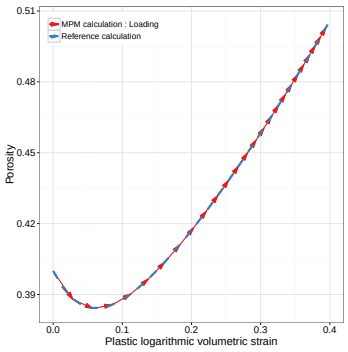
The bulk modulus and crush curve models can be verified by comparison with the plots in Figure 31(a) and (b). The porosity evolves according to the designed behavior in Figure 31(c) as does the pore pressure.



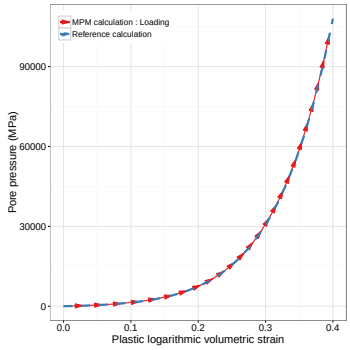
(a) Mean stress vs. Elastic vol. strain



(b) Hydrostatic strength vs. Plastic vol. strain



(a) Porosity vs. Plastic vol. strain



(a) Pore pressure vs. Plastic vol. strain

Figure 31 – *Verification of the bulk modulus, crush curve, porosity and pore pressure models in ARENA using a hydrostatic compression on saturated sand.*

8.3 Uniaxial compression loading: Dry sand

Because the change in direction of the load path upon first encountering the yield surface is not obvious in our previous simulations, we ran the simulation with a larger value of the initial hydrostatic compressive strength. Figure 32 shows the effect of this change on the stress path in $z - r$ space.

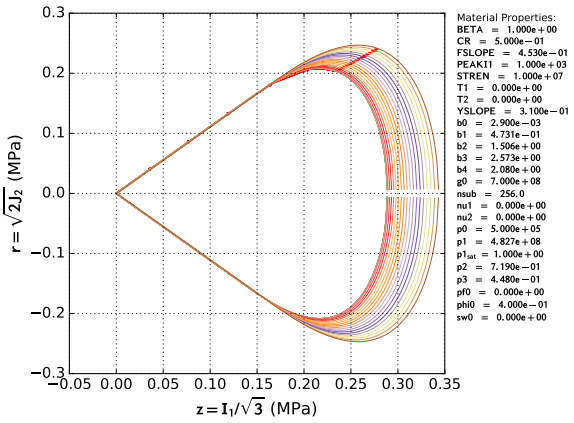
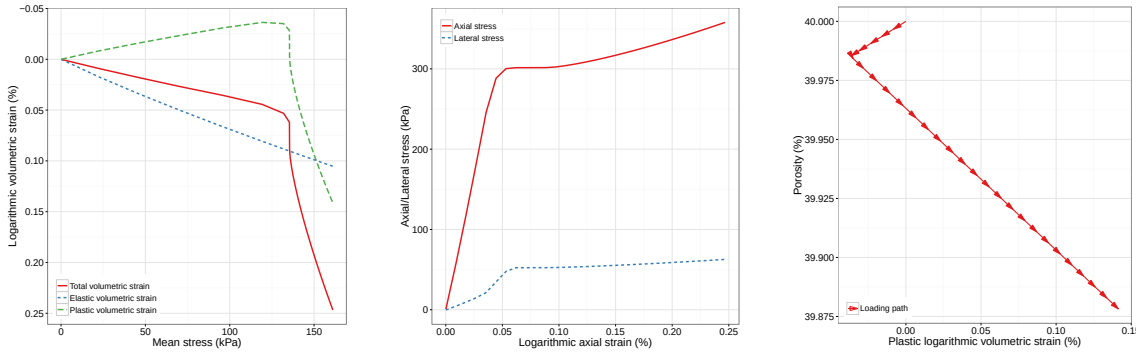


Figure 32 – *Yield surface evolution under uniaxial strain compression of a dry sand with a large p_0.*

Figure 33 shows the evolution of the mean and axial stresses, and porosity, as loading progresses. There is initially a dilation phase accompanied by a small decrease in porosity. It is not clear whether this behavior is accurate and comparison with an exact solution is needed to verify the accuracy of the ARENA implementation. The sand particle is then compressed and the porosity continues to decrease.

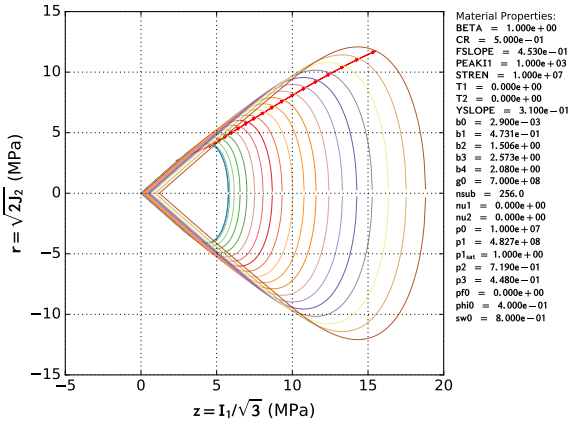


(a) Mean stress vs. volumetric strain (b) Axial/lateral stress vs. axial strain (c) Porosity vs. Plastic vol. strain

Figure 33 – *Uniaxial strain compression of a dry ARENA sand particle.*

8.4 Uniaxial compression loading: Partially saturated sand

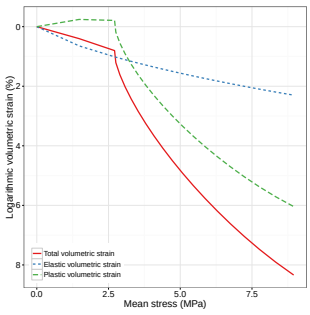
Figure 34 shows the stress path and the evolution of the yield surface when a single ARENA partially saturated Colorado Mason sand particle is compressed in uniaxial strain. The initial porosity is 0.4 and the initial saturation is 0.8. We observe that the yield surface translates with increasing pore pressure as the particle is compressed. This is the designed behavior of our model. Also note the characteristic kink shown by uniaxial strain compression in the load path in the figure.



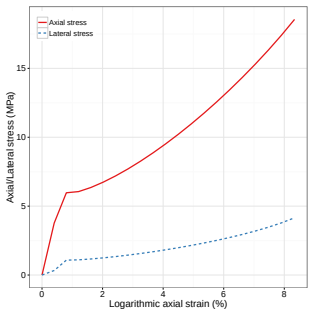
(a) Dynamic stress in z-r space

Figure 34 – *Uniaxial strain: fully saturated: pore pressure and strength*

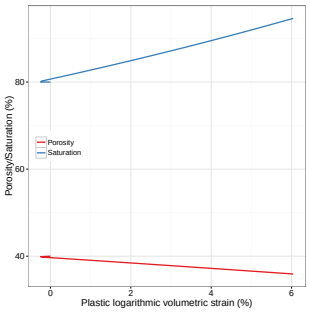
The consolidation curves and stress-strain curves in Figure 35(a) and (b) indicate that, as expected, there an initial dilatation period after which the material follows the crush curve. The porosity, saturation, and pore pressure curves in Figure 35(c) and (d) also exhibit the behavior expected of ARENA , indicating that the implementation is reasonably correct. Exact solutions can probably be computed for this test for more rigorous verification of the model.



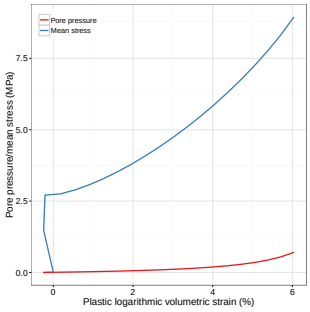
(a) Mean stress vs. Elastic vol. strain



(b) Mean stress vs. Plastic vol. strain



(c) Porosity vs. Plastic vol. strain

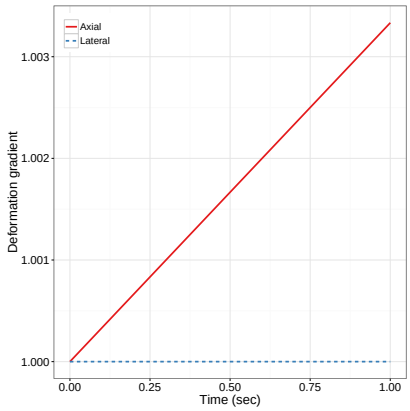


(d) Pore pressure vs. Plastic vol. strain

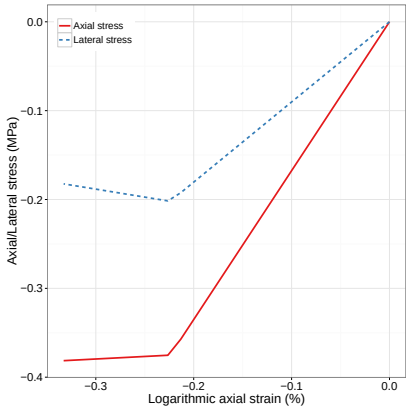
Figure 35 – *Uniaxial strain compression of a fully saturated ARENA particle.*

8.5 Uniaxial tensile strain loading: Dry sand

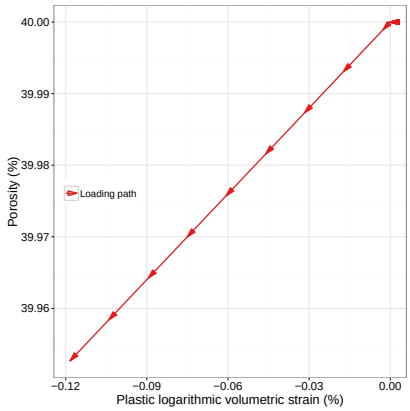
An uniaxial tensile stress produces the expected p-q ratio (see Appendix D and Figures 36(a), (b), (c), and (d) if the stress state is elastic. However, if the rate of deformation is large enough for the trial stress to be outside the yield surface, the plastic return point can be such that the value of p is compressive. Therefore, compressive stress states are possible even if the applied strain is purely uniaxial tension. This is a problem that can be resolved with a non-associative flow rule. Note that in our case small changes in the parameter β were not sufficient to make the stress state tensile in some of our uniaxial tensile strain simulations.



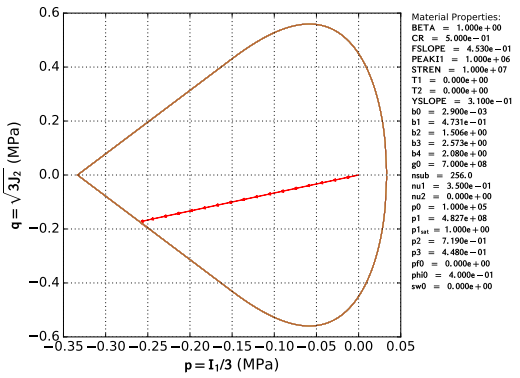
(a) Deformation gradient vs. time



(b) Stress vs. axial strain



(c) Porosity vs. plastic vol. strain

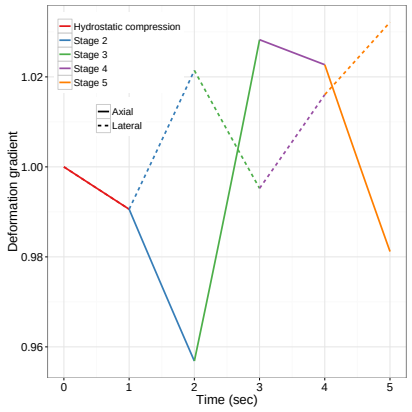


(d) Dynamic stress in p-q space

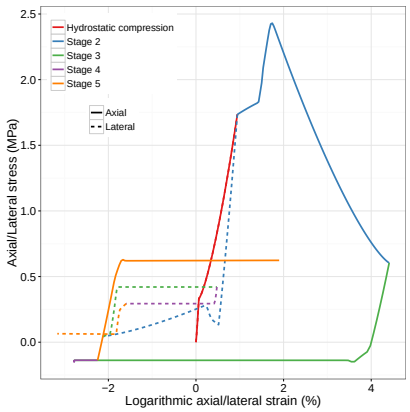
Figure 36 – *Uniaxial strain tension of a dry ARENA sand particle.*

8.6 Multiaxial strain loading-unloading: Dry sand

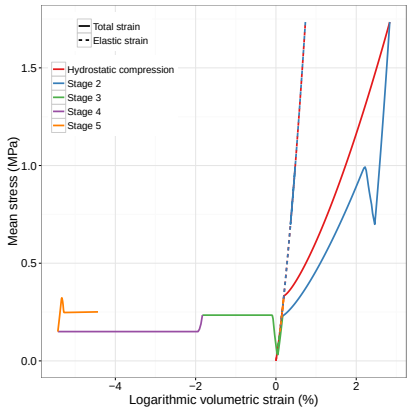
A multiaxial strain loading path was chosen next to explore the behavior of the algorithm during loading-unloading and during the traversal of the tension vertex. Shear strains were kept close to zero during the deformation. The results of a single particle simulation with this loading path are shown in Figure 37. The initial loading is along the hydrostatic axis and includes an elastic component followed by the extension of the yield surface along the crush curve. The stress is then reversed util it reaches the yield surface and moves along the surface back to an elastic state. The axial stress reaches a tensile state and then is reversed until it is compressive again. This simulation shows that the results produced by our ARENA implementation are reasonable.



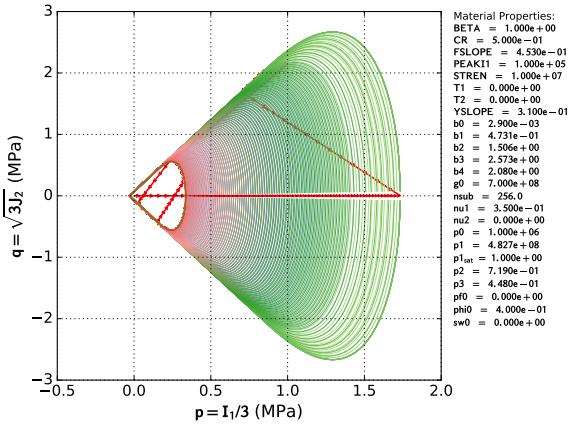
(a) Deformation gradient vs. time.



(b) Stress vs. strain.



(c) Mean stress vs. volumetric strain



(d) Dynamic stress in p-q space

Figure 37 – *Multiaxial strain loading-unloading of a single particle of dry ARENA sand.*

9 ARENA model validation: SHPB simulation vs. experiment

We summarize results from two validation cases here:

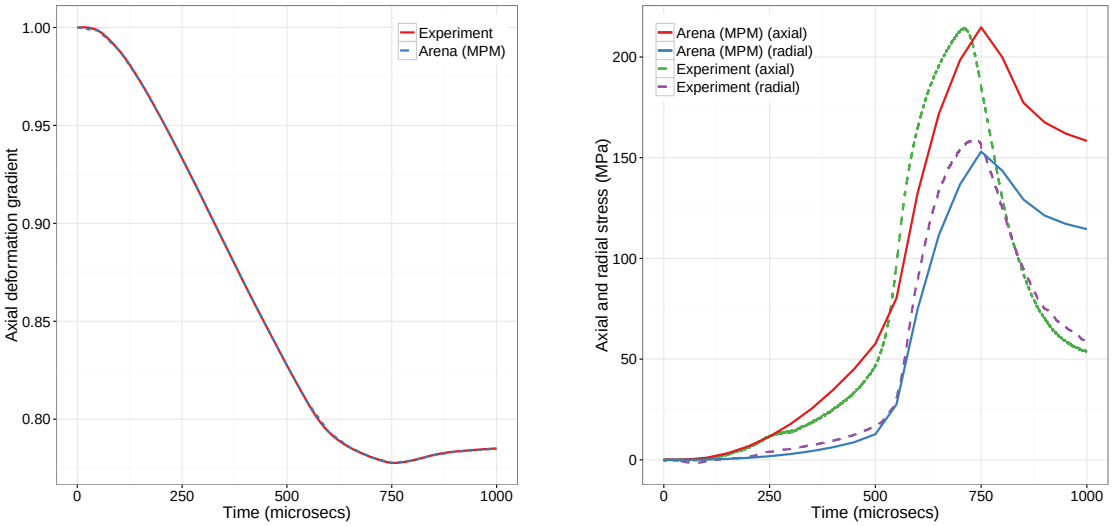
1. uniaxial strain SHPB on a Mason sand (dry density 1700 kg/m^3) containing 18% water by weight, and
2. uniaxial strain SHPB on a Boulder clay (dry density 1300 kg/m^3) containing 12.8% water by weight .

The ARENA model requires the initial density, porosity, and saturation as inputs. The dry density and water content are used to compute these quantities with the approach described in Appendix B.

9.1 SHPB - Wet Mason sand - 18% water by weight

We ran a single particle MPM simulation with dry Mason sand parameters after changing the saturation and mass density. The purpose of this test is to determine whether the ARENA model can predict the behaviour of partially saturated soils that were not used to calibrate the model.

The deformation gradient applied to the particle is shown in Figure 38(a). These data have been extracted from experimental SHPB data (sample 003, on 16 Aug 2012 at UT Dallas), on a partially saturated Mason sand containing 10% water by weight (initial density = 1800 kg/m^3, initial porosity = 0.4264, initial saturation = 0.6559). simulation, we can predict the behavior of partially saturated Mason sand remarkably well. The plots in this section show the predictions of our partially saturated soil model compared with experimental data for a sand containing 18% water by weight.



(a) Deformation gradient vs. time

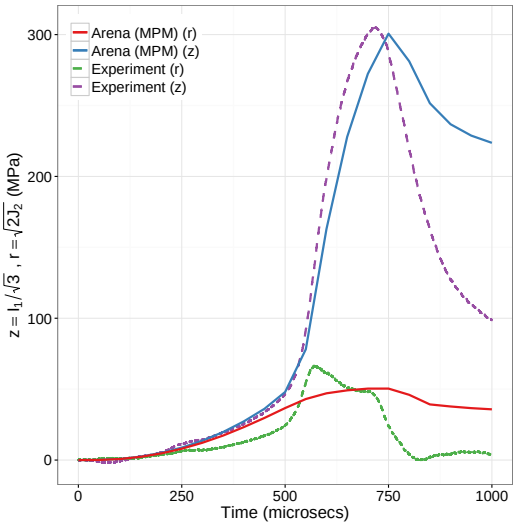
(b) Axial/radial stress vs. time

Figure 38 – *Split-Hopkinson pressure bar: uniaxial: wet Mason sand: 18.4% water by wt.: Time evolution.*

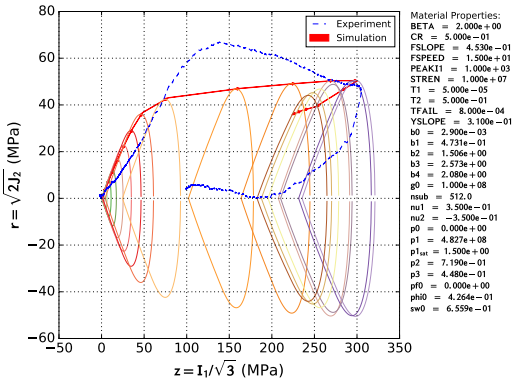
9.2 SHPB - 12.8% w/w Boulder clay - Sample 014

The ARENA model was calibrated using dry and almost fully saturated Boulder clay. In this section we observe how well the model performs for a clay containing 12.8% water by weight. The applied deformation gradient is shown in Figure 40(a) and the axial and radial stresses predicted by the model are given in Figure 40(b).

We observe that the predicted stresses are higher than those measured in the experiment. The peak stress does not change significantly with saturation if $p_1^{\text{sat}} = 1$. However, if the value of p_1^{sat} is increased,



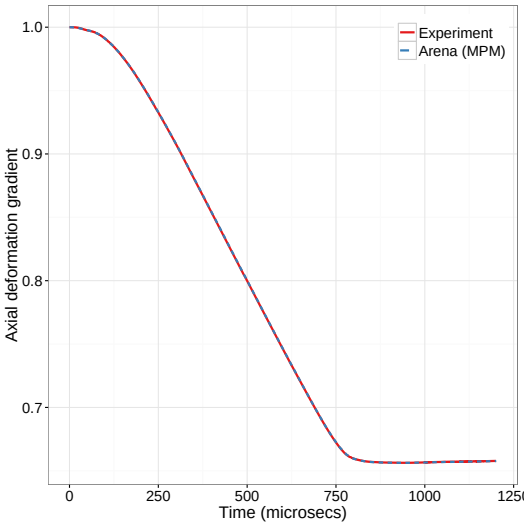
(a) z and r vs. time



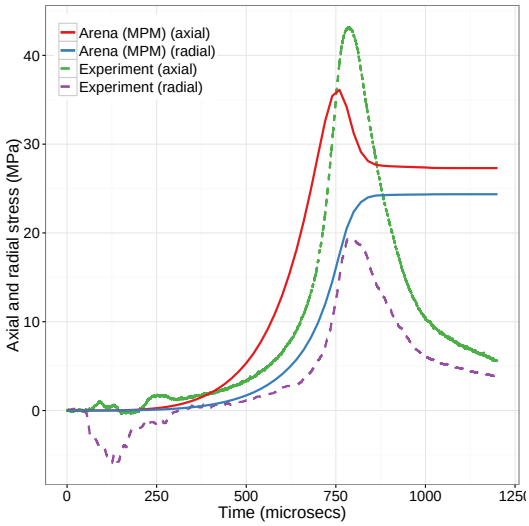
(b) Dynamic stress in $z - r$-space

Figure 39 – *Split-Hopkinson pressure bar: uniaxial: wet Mason sand: 18.4% water by wt.: z, r vs. time.*

the peak stress increases making the fit to experiment worse. The fit improves when $p_1^{\text{sat}} < 0$. We conjecture that the compressive strength decreases with saturation in this case because the clay grains can move relative to each other more easily when a small amount of water is added. This decreases the hydrostatic strength relative to that of the dry material. Beyond a certain level of saturation, this trend is reversed, as seen in the simulations used for parameter calibration.

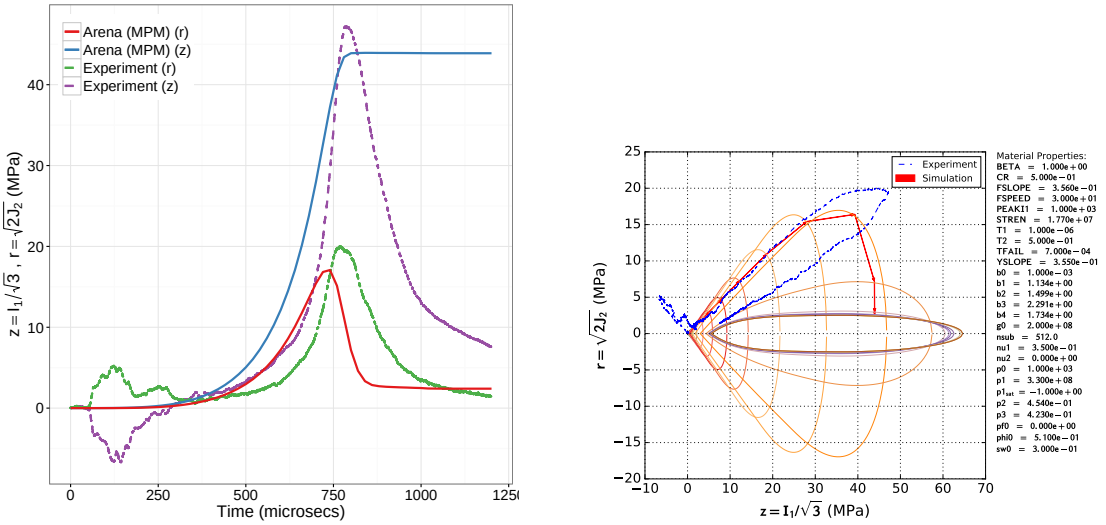


(a) Deformation gradient vs. time



(b) Axial/Radial stress vs. time

Figure 40 – *Split-Hopkinson pressure bar: uniaxial: 12.8% w/w partially saturated Boulder clay: Time evolution.*



(a) z and r vs. time

(b) Dynamic stress in $z - r$-space

Figure 41 – *Split-Hopkinson pressure bar: uniaxial: 12.8% w/w partially saturated Boulder clay: r, z vs. time.*

10 Simulations of explosions with ARENA

One of the main reasons for the development of the ARENA model is the simulation of explosions in soil. While explosions and their effects on soils are of scientific and engineering interest themselves, we are more interested in the effect of these explosions on objects on the surface. In this section we describe a few simulations of these effects. The actual explosive gases and pressure pulse are approximated, first by a piston and then by a pulse that has the shape measured in experiments. Since the experiments were performed in a centrifuge, we include the effect of the artificial gravity that is induced in the process.

10.1 An one-dimensional impact simulation

In this simulation, we approximate the explosive pulse by a piston. The piston impacts a soil sample that is modeled with ARENA after which the impact wave passes through the soil sample and initiates motion in a plate sitting on the free surface of the soil. We would like to examine whether the degree of saturation of the soil sample has any effect on the motion of the plate.

The entire simulation is performed under an artificial gravity of 10g. The piston is assumed to be made of steel and has a density of 8000 kg/m^3. It is modeled as a compressive Neo-Hookean material, with thickness 1 cm and initial velocity 30 m/s. The soil is modeled first using the properties of dry Boulder clay with an initial density of 1270 kg/m^3, initial porosity 0.52, and damage model activated. The soil sample is 3 cm thick and is initially at rest. After that the soil is replaced with wet Boulder clay that has an initial density of 1830 kg/m^3 and an initial saturation of 0.9. For the steel plate on the surface of the soil, we use the same properties as that of the piston. We assume that the steel plate is 1 cm thick and at rest 2 mm above the surface of the soil.

The domain is discretized into a grid of 1 mm spacing, with four particles per cell. To keep the problem approximately one-dimensional, only one MPM cell is used in the directions normal to the direction of motion of the piston. Four MPM particles are used to discretize each cell.

Figure 42 shows an animation of the simulation (needs Adobe Acrobat reader). In the figure, the impactor is colored dark blue at $t = 0$, the soil is blue, and the plate is yellow. The animation on the top represents the dry soil while that on the bottom of the figure is for the wet soil.

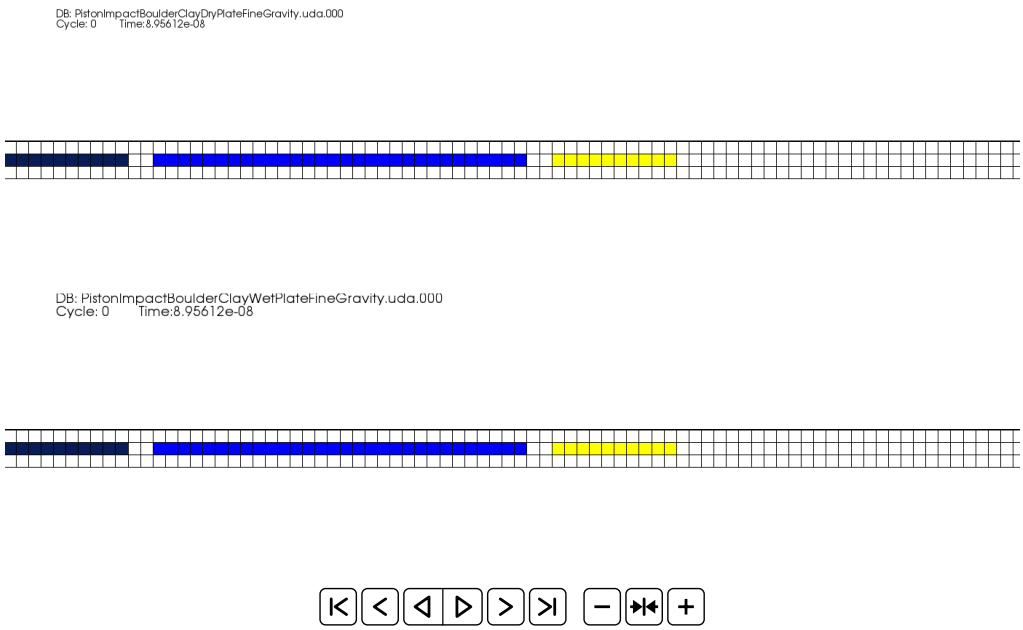
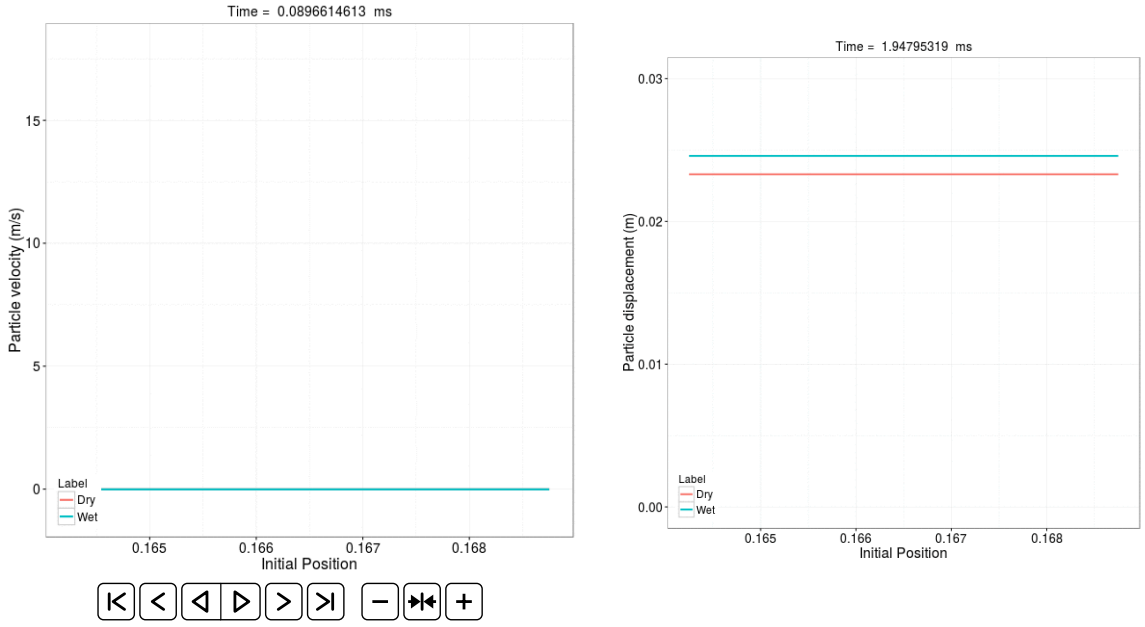


Figure 42 – *Animation of MPM simulation the behavior of a plate on the surface of a soil sample impacted by a piston.*

The plate on the soil surface is provided more momentum by the partially saturated soil and moves further than the plate on the dry soil. It is difficult to separate out the effect of mass density from that of the constitutive model on the momentum of the plate. However, we have estimated from preliminary comparisons with a stiff elastic model for the soil that the impact of mass density is about 60% on the initial velocity of the plate.

Figure 43(a) shows an animation of the velocities of the particles in the plate after the soil has transferred its momentum. The wet soil (blue) transmits its velocity to the plate much earlier than the dry soil. Note that the terminal velocities in the two cases reach approximately the same value after some time. The difference in the initial acceleration between the two plates is small. However, the wet soil causes a slightly larger displacement (Figure 43(b)) of the plate under these high gravity conditions and we speculate that it is this difference in displacement makes wet soil appear more destructive.



(a) Plate velocity.

(b) Plate displacement.

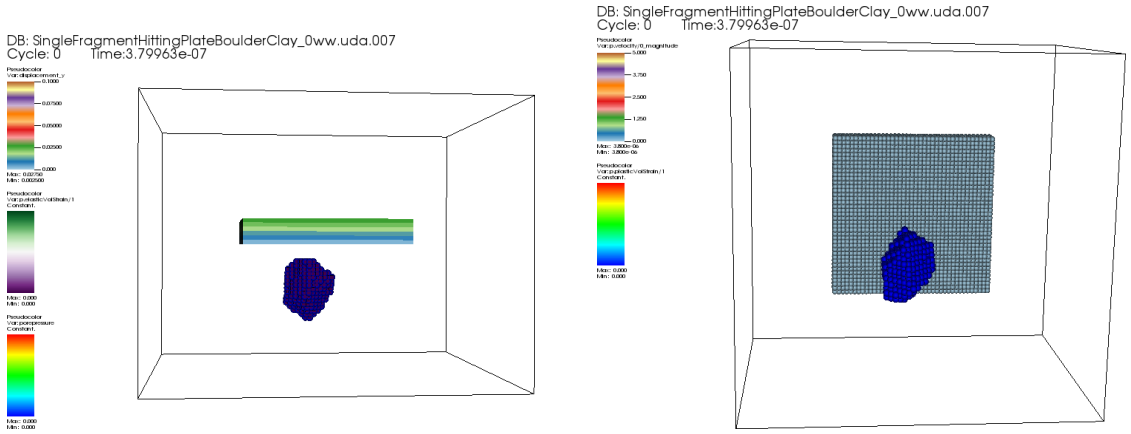
Figure 43 – *Animation of particle velocities and a snapshot of the displacements in the plate on the soil surface.*

10.2 A lump of clay impacting a hollow box

In this simulation we model the impact of a lump of Boulder clay on a hollow box under 1g gravity. The hollow box, which is initially at rest, is made of steel of density 7850 kg/m^3, bulk modulus 8.9 GPa, and shear modulus 3.52 GPa. The outer dimensions of the box are 20 cm × 20 cm × 3 cm and the wall thickness is 5 mm. We use the ARENA model for Boulder clay to model the clay lump and compare the dry and partially saturated cases. For clay, we use the properties listed in the previous section and assign an initial horizontal velocity of 20 m/s and a vertical velocity of 100 m/s.

The initial configuration of the simulation is shown in Figure 44. We have used the magnitude of the displacement of the box as its color while the soil lump shows the pore pressure as spheres and the elastic volumetric strain as glyphs that deform as a factor of the deformation gradient of each soil particle. The discretization is relatively coarse. There is an initial separation between the soil lump and the box.

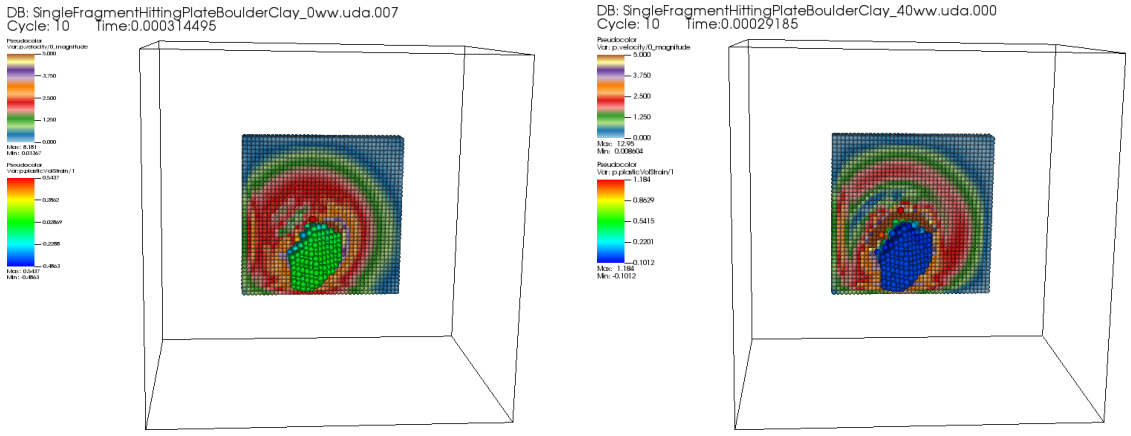
Soon after the soil lump impacts the box, we get the propagation of stress waves in the box as depicted in Figure 45. The figure shows the particle velocities in the box when impacted by dry clay (left) versus those when impacted by wet clay (right) and the plastic volumetric strain in the clay samples. The wet



(a) View from side. (b) View from bottom.

Figure 44 – *Initial configuration of the simulation of a lump of clay impacting a hollow box.*

clay exhibits significantly larger plastic volumetric strains than the dry clay at approximately the same time after impact (300 microsec). However, the velocities in the box are approximately the same in both cases.

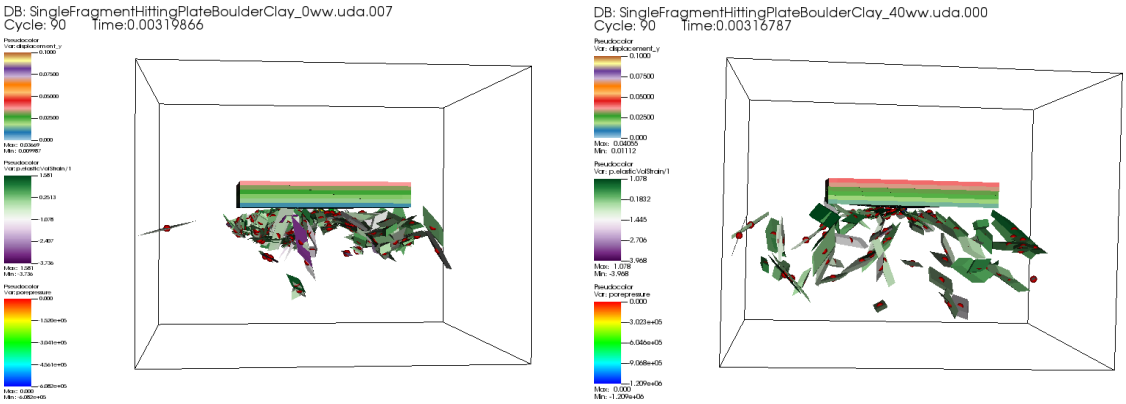


(a) Dry Boulder clay. (b) Wet Boulder clay.

Figure 45 – *Comparison of stress waves produced after initial impact.*

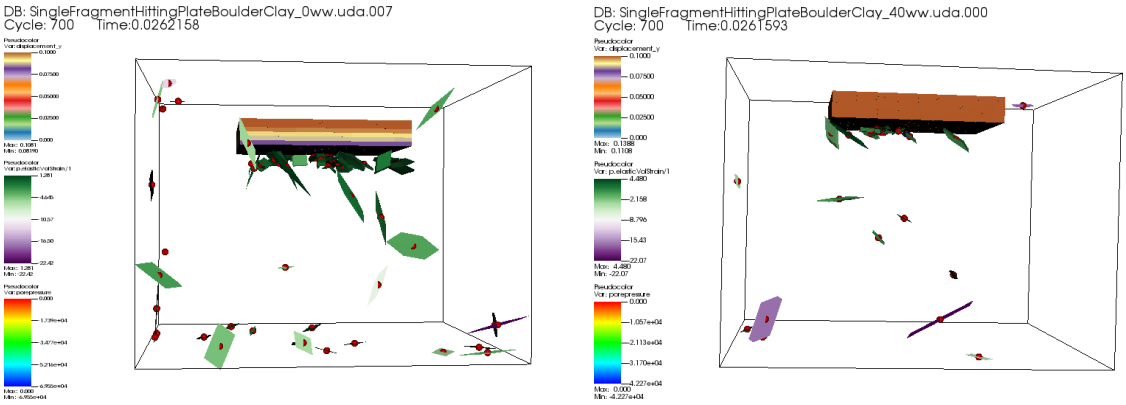
After 3.2 milliseconds, we see from Figure 46 that the lumps have started to break up and many particles show a significant amount of deformation as indicated by the flatness of the glyphs. Moreover, the box has not yet displaced significantly and both the dry and the wet sand produces approximately the same displacement in the box.

At 26 milliseconds, we observe that the wet clay has displaced the box further than the dry clay (Figure 47). This is similar to what we saw in the previous section. Also, many of the wet clay fragments have left the domain owing to their greater density and momentum. This observation suggests that the Arena model has to be modified to allow for water to exit the particles when the material is disaggregated. Typical partially saturated soil models that solve the mass balance equations for the phases without making the restrictive assumptions of Arena will also run into similar issues because the material loses continuity in simulations such that these.



(a) Dry Boulder clay. (b) Wet Boulder clay.

Figure 46 – *Comparison of deformed particles at t = 3.2 millisec .*



(a) Dry Boulder clay. (b) Wet Boulder clay.

Figure 47 – *Comparison of deformed particles at t = 26 millisec.*

10.3 An explosion in soil inside a centrifuge

Let us now examine the effects of explosions in soil contained in a box that is being rotated in a centrifuge. The configuration of the box and the soil is shown in Figure 48. The centrifuge rotates at 4 to 9 radians/s, producing artificial gravity loads between 10g and 40g in the soil. The soil is contained in a steel bucket that is isolated from the soil by an elastomeric DuxSeal layer and an aluminum plate with an elastomeric pad protect the bottom of the bucket from explosions. The explosive charge is placed between 1 cm and 9 cm from the surface of the soil and is usually centered in the bucket though it can be offset as shown in the figure to avoid shock focusing.

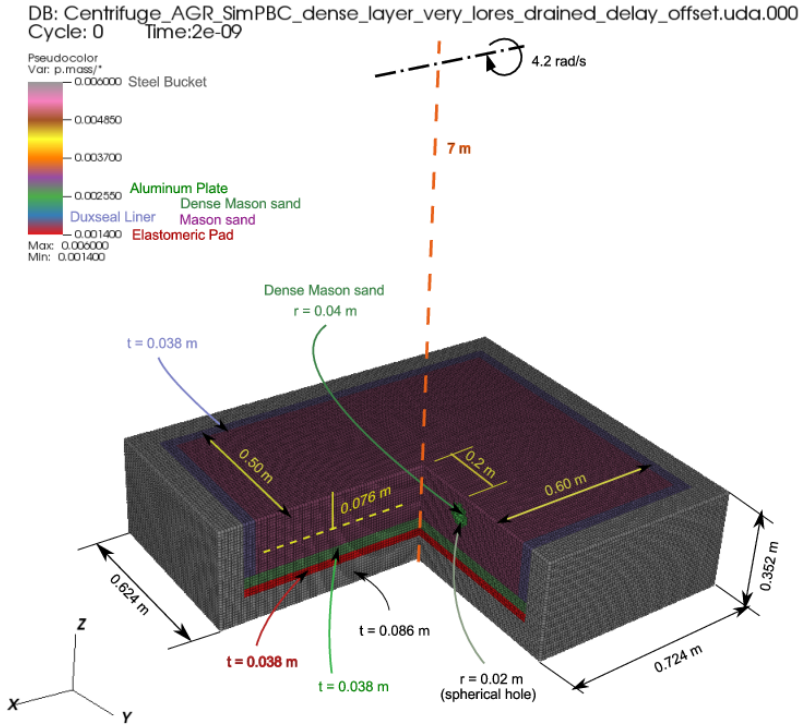
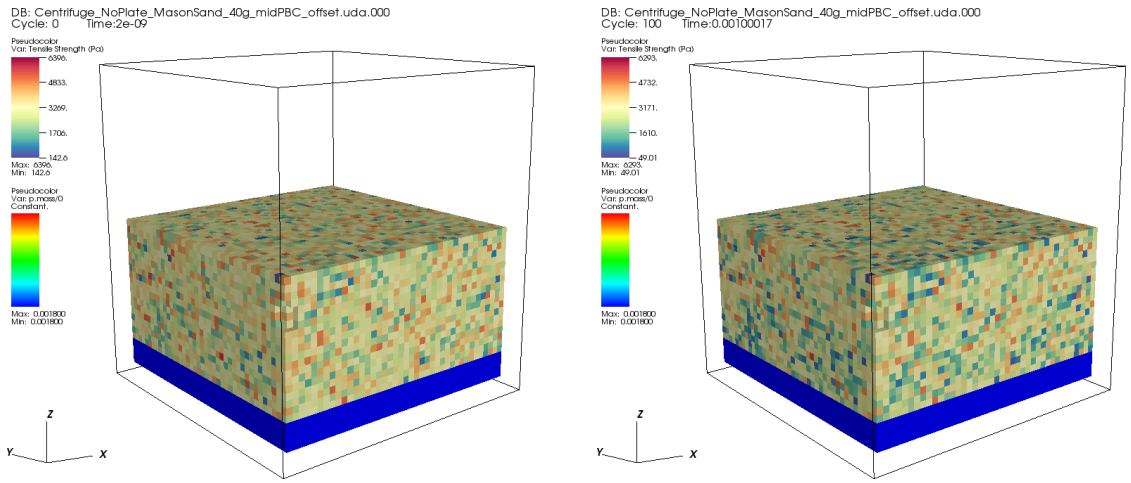


Figure 48 – *Geometry of the soil sample and box that is rotating in a centrifuge.*

10.3.1 The spin-up phase

To simulate the effect of the rotation of the centrifuge, each simulation is run through a spin-up phase where the centrifugal forces are computed. The spin-up simulation is run until the stresses in the soil have stabilized. Our approach described in Appendix C is used to vary the yield condition parameters of the soil. Figure 49(a) shows the distribution of the tensile strength (`PEAKIO/3`) in the soil at the beginning of a simulation. The material is Colorado Mason sand. Only a quarter of the simulated region is shown in the figure. For simplicity, the steel bucket and the DuxSeal layer have not been modeled. Only the aluminum plate below the soil is included and is shown in blue in the figure. After running the simulation for 1 millisecond, we reach the tensile strength distribution depicted in Figure 49(b). We would like to point out that the additional gravity load leads to small changes in the tensile strength distribution because the damage model is activated in the process. *Therefore, just the process of applying a gravity load of 40g is enough to cause inelastic deformations that lead to damage in the soil.*

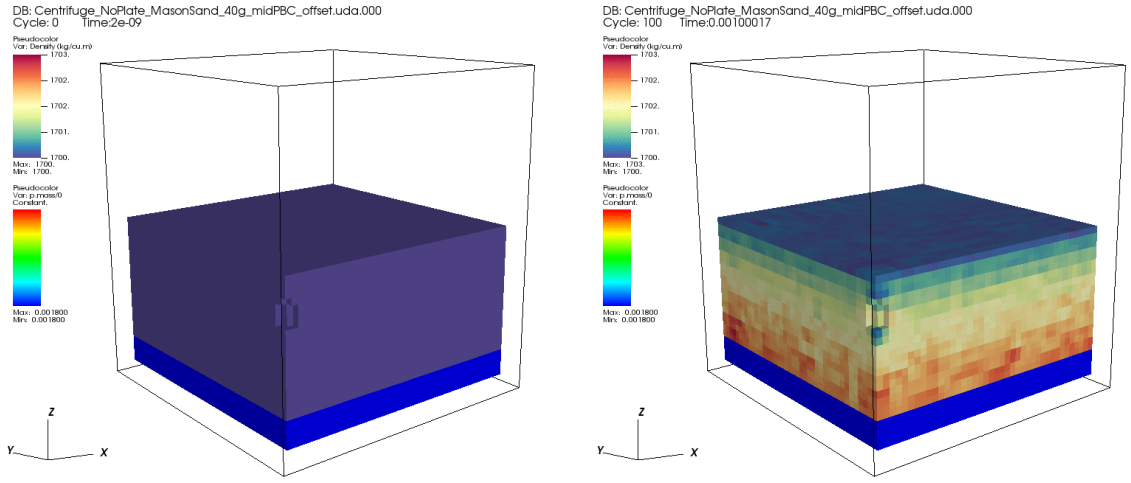
Due to the application of the 40g acceleration induced by the rotation of the centrifuge, the soil sample in the bucket is compressed. Figure 50 shows the initial and final densities of the sample during the spin-up phase. We observe that though the change in density is small, there is a clear layering effect induced by the spin-up process. The material is less dense near the surface. An effect due to the variability of the material is also observed from the simulation. Note that the initial density of the sand is



(a) Tensile strength at t = 0 ms (b) Tensile strength at t = 1 ms

Figure 49 – *The tensile strength distribution in a sample of Colorado Mason sand contained in a centrifuge producing 40g acceleration at the beginning (a) and end (b) of the spin-up phase. The process of spin-up is enough to induce plastic strains and activate the ARENA damage model.*

1700 kg/m^3 with a reference porosity of 0.42[12] and an initial porosity of 0.36. The material is dry.



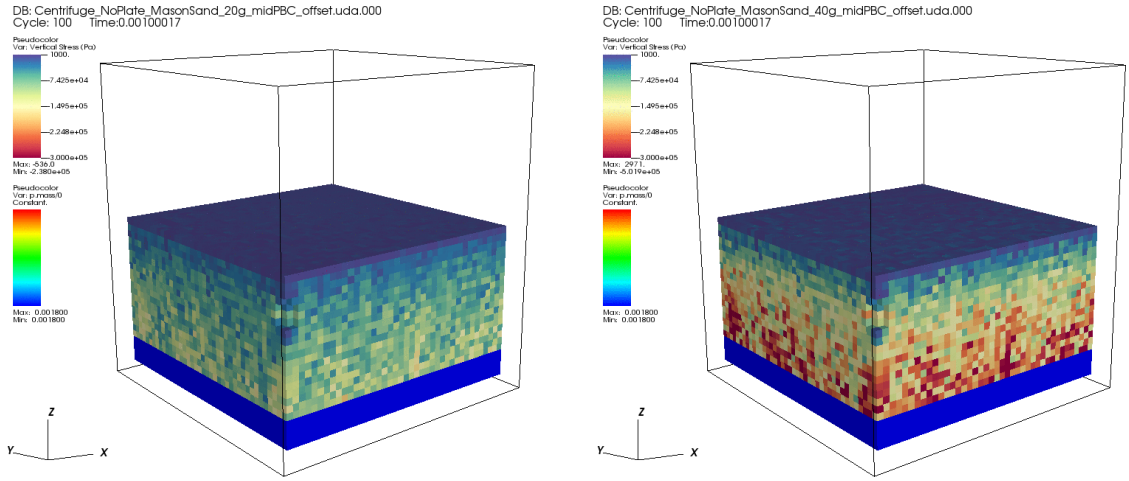
(a) Density at t = 0 ms (b) Density at t = 1 ms

Figure 50 – *The density distribution in the Colorado Mason sand sample at the beginning and end of the spin-up phase of the simulation.*

Figure 51 shows the vertical stress distribution in the soil at the end of the spin-up phase (1 millisec rotation). As expected, the stresses are smaller when a 20 g acceleration applied to the sample compared to when a 40 g acceleration is applied. Also, the vertical stress varies from zero at the surface to 150 kPa (at 20 g) to 300 kPa (at 40 g) at the bottom of the sample.

We can see from Figure 52 that the stress has stabilized at almost all depths after 1 ms. The plots in the figure show the evolution of the vertical stress in the soil at distances of 1 cm, 1.52 cm, 2.03 cm, 2.54 cm, 3.05 cm, and 3.56 cm from the point at which the explosive charge is placed. It can be observed that the vertical stress varies with location at the same depth due to the variability of the material. The

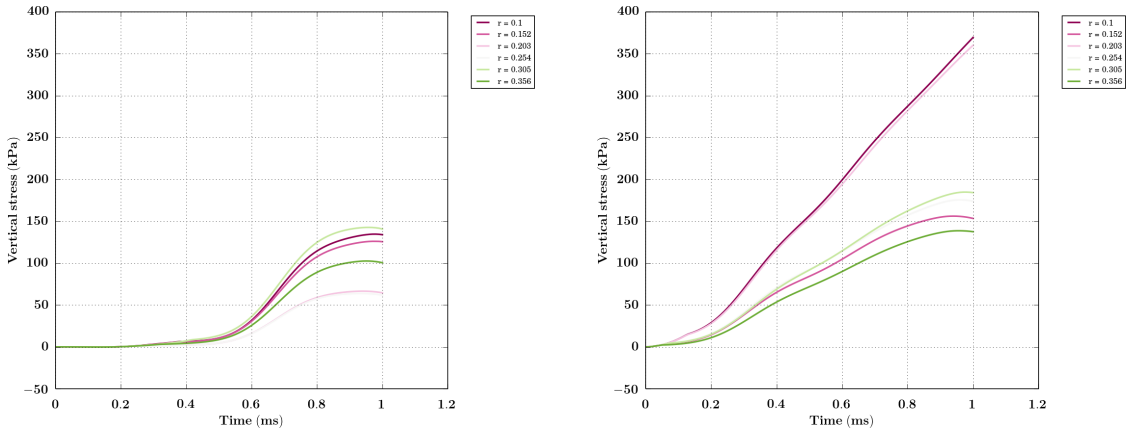
[12] Recall that the reference porosity is the porosity of dry Colorado Mason sand at the reference density of 1520 kg/m^3.



(a) Vertical stress at 20 g (b) Vertical stress at 40 g

Figure 51 – *The vertical stress in the dry Colorado Mason sand sample at the end of the spin-up phase for two different centrifuge rotation speeds.*

stress takes longer to settle own to a steady value at points closer to the spherical hole representing the charge in the horizontal plane but further in the vertical plane (in this case the charge is at a depth of 5.1 cm). It is not clear why that is the case.

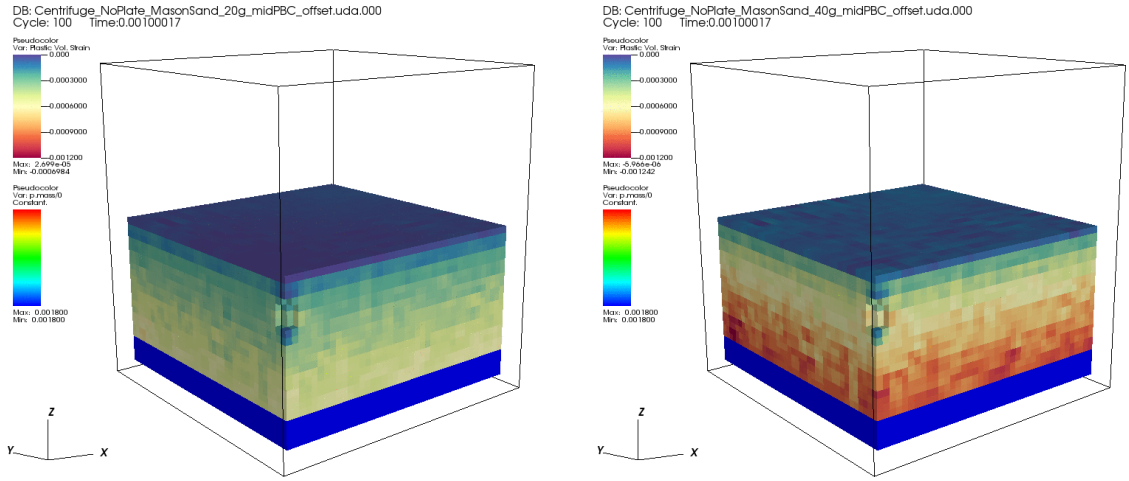


(a) Stress at depth of 5.1 cm (b) Stress at depth of 15.2 cm

Figure 52 – *Evolution of vertical stress with time during the spin-up phase of the simulation at two different depths from the surface.*

To examine whether there is a significant amount of plastic deformation during the spin-up phase, we plot the volumetric plastic strain in the soil at the end of that phase. Snapshots of the strain (shown in Figure 53) indicate that for a 20 g load, we get approximately 0.06% plastic strain at the bottom of the sample and at 40 g, 0.12% plastic strain.

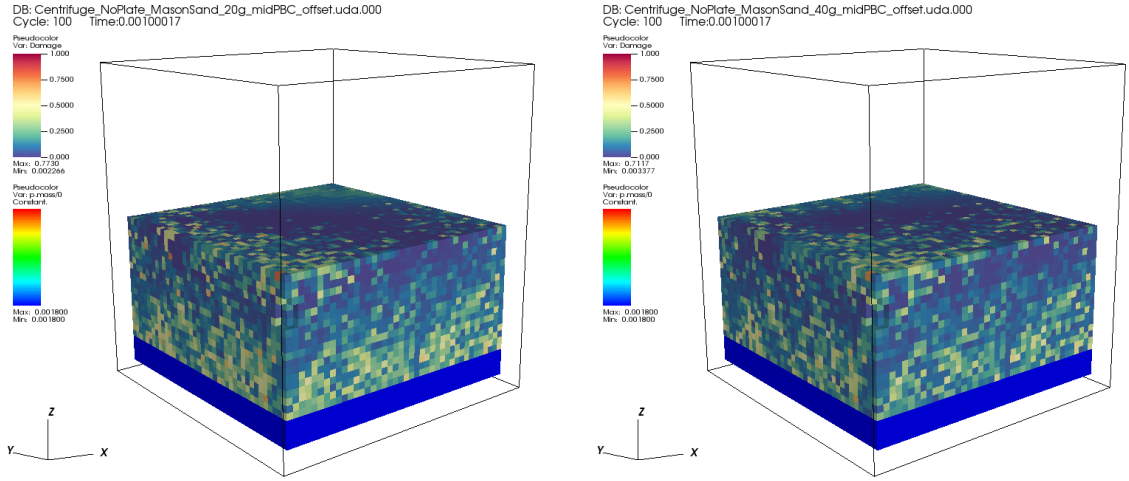
Interestingly, a significant amount of damage is observed during the spin-up phase particularly at a few isolated particles near the surface of the soil. This can be seen in Figure 54 both for the 20g and the 40g cases. We attribute these effects partly to boundary effects caused by the planes of symmetry in the simulation (anomalous tension) and partly to the Weibull modulus used to generate the distribution of yield parameters in the simulation. We also observe that both the 20g and the 40g simulation appear to predict the same amount of damage at the end of the spin-up phase even though the stresses and plastic strains are different. It is possible that this anomaly is caused by the MPM algorithm and further



(a) Vol. plastic strain at 20 g (b) Vol. plastic strain at 40 g

Figure 53 – *The volumetric plastic strain in a dry Colorado Mason sand sample inside a centrifuge at the end of the spin-up phase for two different centrifuge rotation speeds.*

research is needed to determine the cause.



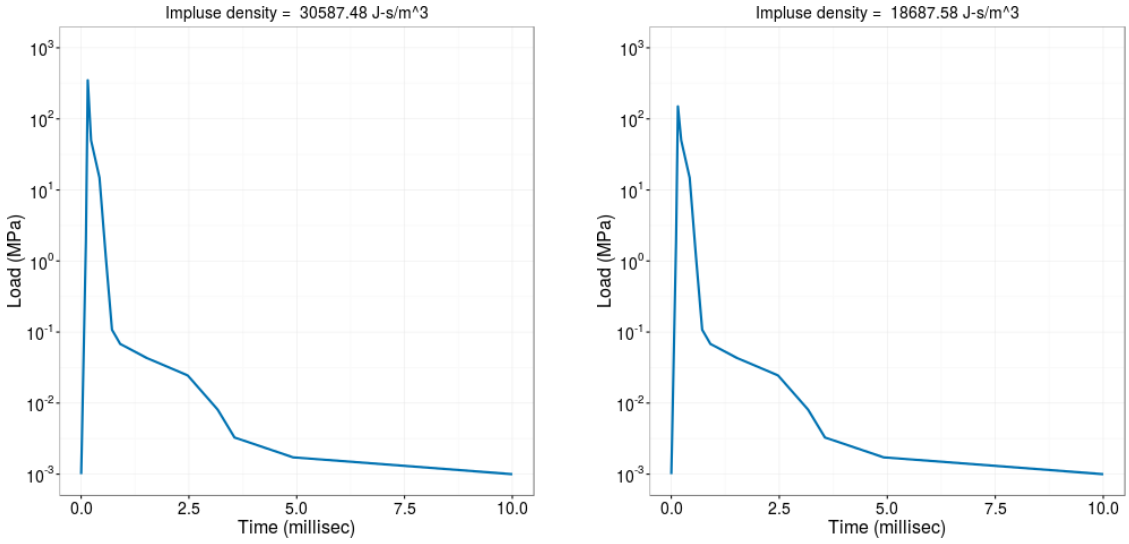
(a) Damage at 20 g (b) Damage at 40 g

Figure 54 – *The amount of damage induced in a dry Colorado Mason sand sample inside a centrifuge at the end of the spin-up phase for two different centrifuge rotation speeds.*

10.3.2 The explosion phase

At the end of the spin-up phase, stresses in the soil have reached a relatively steady state and the explosion simulation can proceed. Since we are not modeling the explosion directly using a model such as JWL++, we apply a pressure pulse to the surface of a spherical hole centered at a point that represents the center of mass of the explosive charge. The pressure pulse that is used in the simulations is either what we term "medium" (impulse density 30.6 kJ-s/m^3) or "medium-low" (impulse density 18.7 kJ-s/m^3). Figure 55 shows the shapes of the two pressure pulses. The medium-low pressure pulse produces initial surface displacements that are close to those observed in experiments. However, that pulse is not strong enough to break through the surface of the soil. We therefore, focus our attention

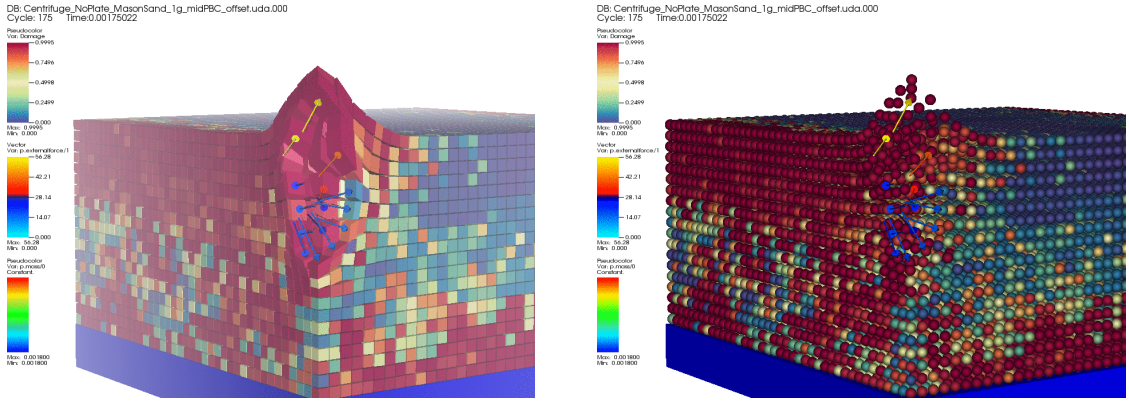
to the medium pulse in what follows.



(a) Medium pressure pulse (b) Medium-low pressure pulse

Figure 55 – *The pressure pulse used to simulate explosions in sand samples inside a centrifuge.*

The MPM particles in the simulation undergo considerable amounts of deformation due to the explosive load as can be seen in Figure 56(a) where the MPM particles are represented as cubic glyphs that are deformed. These glyphs make clear that we are, in fact, using a continuum description of the material rather than a particle description such as that used in discrete element methods. In some situations it is more convenient to plot the particles using spherical glyphs as shown in Figure 56(b). Spherical glyphs are used in most of the plots that follow.

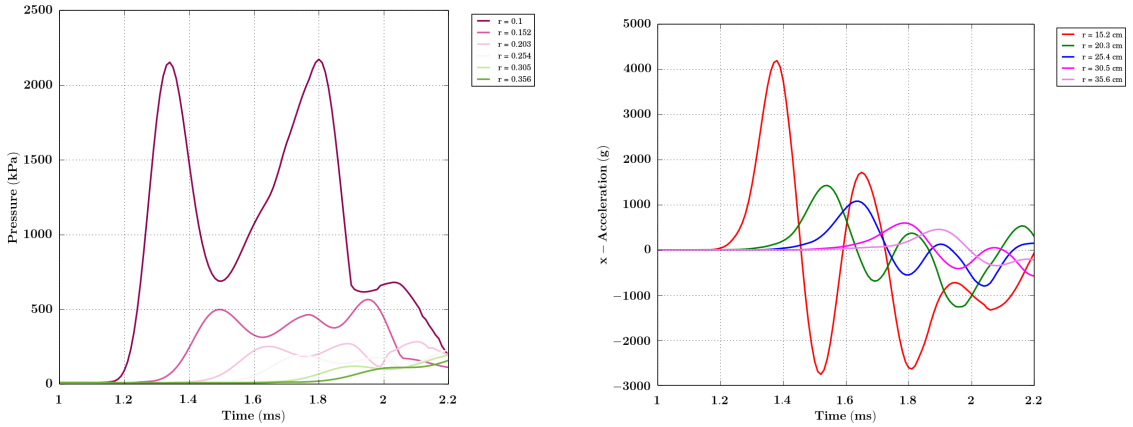


(a) Cubic glyphs showing damage (b) Spherical glyphs showing damage

Figure 56 – *Plots of damage and the applied forces in the soil particles during the explosion phase.*

In Figure 57(a) we plot the evolution of pressure at various distances from the explosive charge at a depth of 5.1 cm from the surface of the soil. Two peaks are observed in the pressure at a distance of 10 cm from the charge and the peak pressure is around 2 MPa. In centrifuge experiments performed under similar conditions only one peak is observed in the pressure after which the pulse decays rapidly. It appears that there is a significant amount of viscous damping in the sand that is not captured adequately by our model. A higher resolution study is needed to determine if these peaks are an artifact

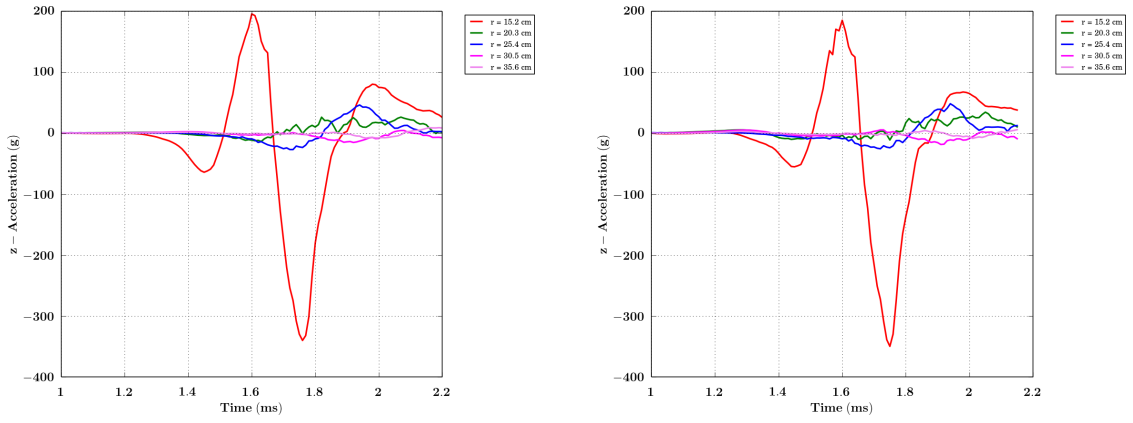
of the MPM grid size. The accelerations at points at various distances from the charge are shown in Figure 57(b). Experimental data suggest much more rapid damping than that observed in the figure.



(a) Soil pressures (b) Soil accelerations

Figure 57 – *Soil pressures and accelerations due to the medium explosion pulse inside a centrifuge producing a 20g load.*

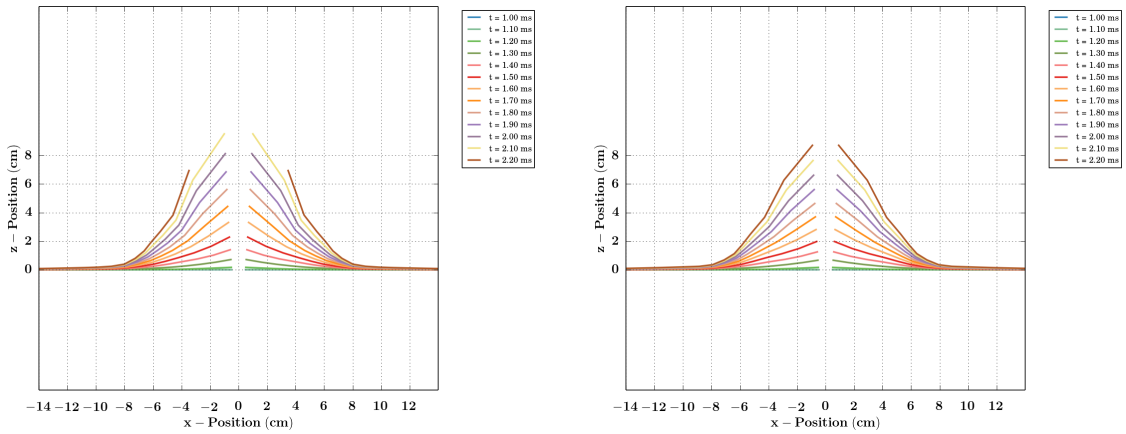
Accelerations at various locations on the surface of the aluminum plate below the soil at two different value of gravity are plotted in Figure 58. The effect of the g-values on these accelerations appears to be marginal.



(a) 20 g (b) 40 g

Figure 58

The bulge produced at the surface as a function of time after the explosion is an indicator of the effectiveness of the Arena model in simulating explosion scenarios. Though the explosion is not being modeled accurately, the shapes of the bulges shown in Figure 59 seem to suggest that the model is reasonably accurate when compared with experiment. The size of the bulge is smaller at larger g-values. However, if we compare the height of the bulge with experimental data, we observe that the predicted values are larger than observed. Reducing the pulse energy leads to better fits to experiment at the cost of failure to cause tensile failure in the soil model. These issues need further exploration before the Arena model can be used to model tension dominated situations.



(a) 1g (b) 20 g

Figure 59 – *Bulge evolution in dry Colorado Mason sand from an explosion in a centrifuge and two different g-load levels.*

10.3.3 Explosions with a target over partially saturated soil

Considering the fact that the explosive pulse and gases are not being modeled accurately in these simulations, the best we can do a comparative study rather than accurate predictions. With that in mind we will model the effect of water content on the behavior of Colorado Mason sand and Boulder clay in a centrifuge under 20g acceleration. These simulations include a steel target plate that is placed above the surface of the soil.

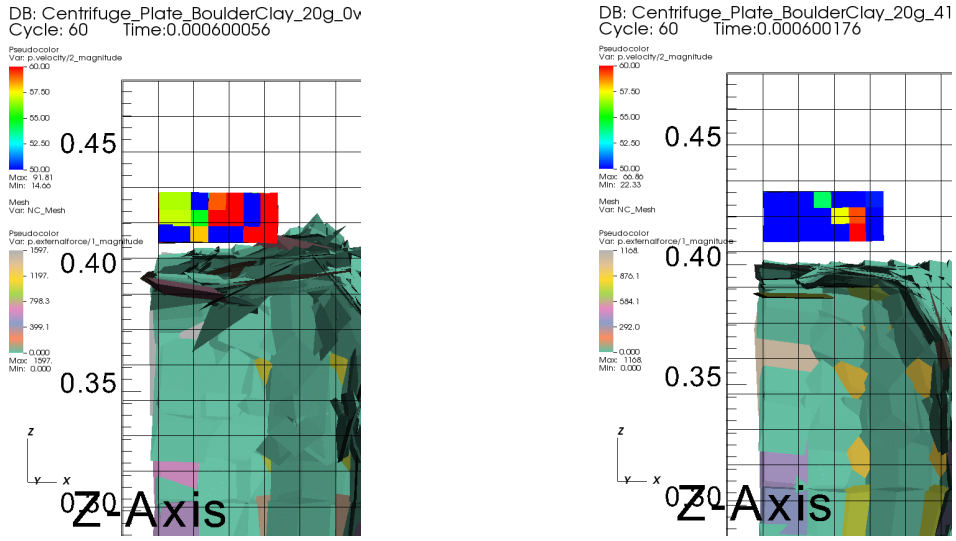
In the images from the simulations shown in Figure 60, the depth of burial of the explosive charge is 5.1 cm and the target is 5.1 cm above the soil surface. The MPM particles are displayed at cubic glyphs. We observe the extreme deformations of these particles and loss of continuity, particularly for the dry clay simulation. The wet clay deforms less and also displaces the target plate less than the dry clay. However, the wet clay imparts more momentum to the target plate. These simulations used the GIMP interpolation technique. We expect the CPDI method to provide improved continuity of the MPM Particles.

Let us now consider a more detailed simulation with Colorado Mason sand. As before the centrifuge applies an acceleration of 20g. The depth of burial is 5.1 cm and the target is 5.1 cm above the soil surface. Cubic glyphs make the simulation confusing and we use spherical glyphs instead. Figure 61 shows the explosion in the sand at a late stage in the simulations. The wet sand contains 10% water by weight. The plots for the dry and wet sands do not show much difference and are only indicative of the progress of the simulation.

However, if we extract the particle accelerations in the soil as plotted in Figure 62 we notice that the particle accelerations are larger in the dry sand while the particle pressures are nearly identical.

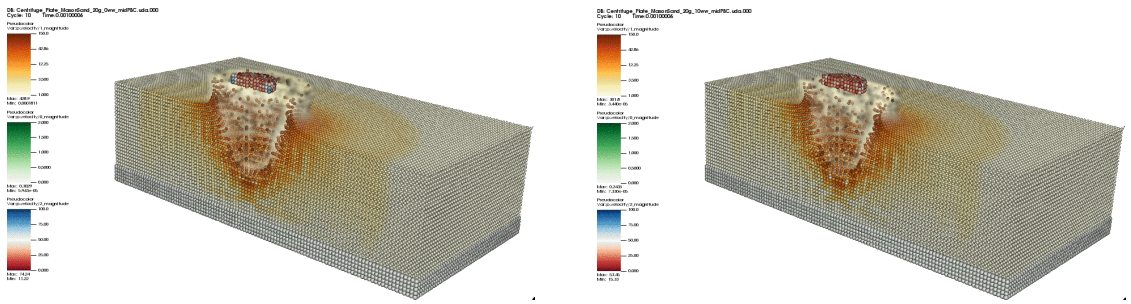
If we repeat the previous simulations with Boulder clay (dry and 41% water content), we observe the deformations shown in Figure 63 and the accelerations and pressures in Figure 64. Examining the deformations in Figure 63 we observe that the dry clay deforms much more above the surface than the wet clay. However, the stress pulse propagates further in the wet clay than in the dry clay. The pressure in the dry and wet clays (figures 64(c) and (d)) shows that the dissipation in the dry clay is faster but the speed of the pressure pulse is faster in the wet clay. Similar behavior is observed in the acceleration time-histories for the two cases. Interestingly, the peak accelerations are around 2 to 3 times higher in the wet clay compared to the dry clay.

It is instructive to compare the momentum imparted to the target and the corresponding impulse (the sum of the surface forces on the target multiplied by the time) for the sand and clay simulations



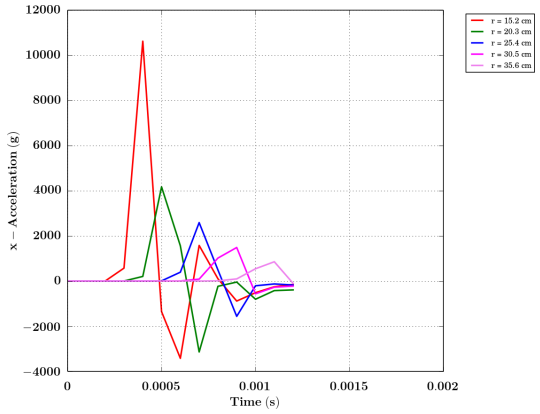
(a) Dry clay (b) Wet clay with 41% water by weight

Figure 60 – *Deformation of Boulder clay and displacement of the target plate in a centrifuge simulation at 20g. The charge is at a depth of 5.1 cm and the plate is 5.1 cm above the soil surface.*

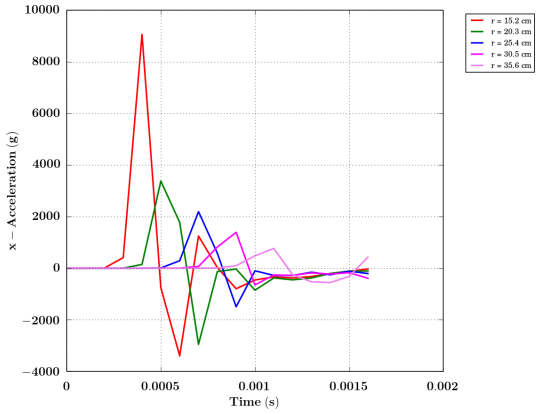


(a) Dry Mason sand (b) Wet Mason sand containing 10% water by weight

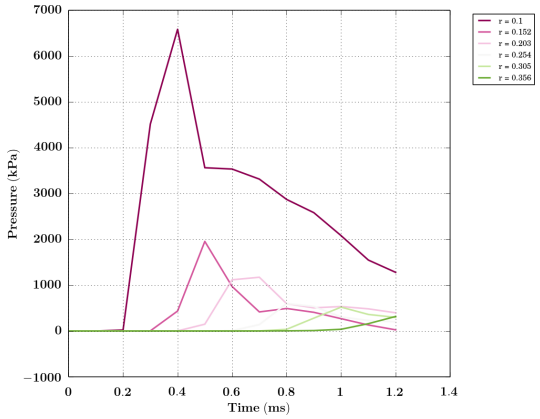
Figure 61 – *Explosion simulation with a plate over Mason sand inside a centrifuge at 20g.*



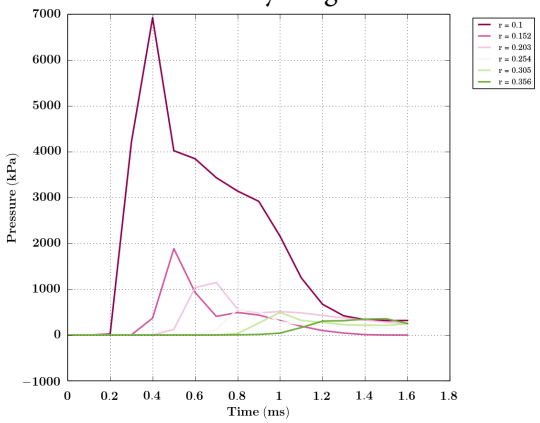
(a) Acceleration: Dry Mason sand



(b) Acceleration: Wet Mason sand containing 10% water by weight

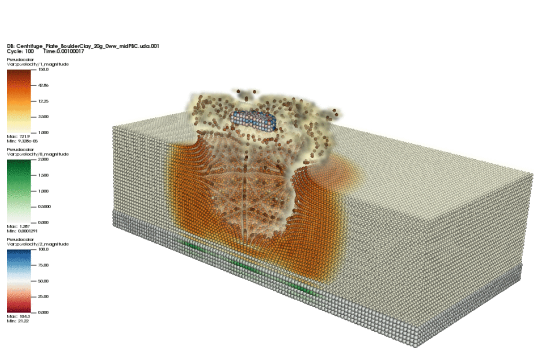


(c) Pressure: Dry Mason sand

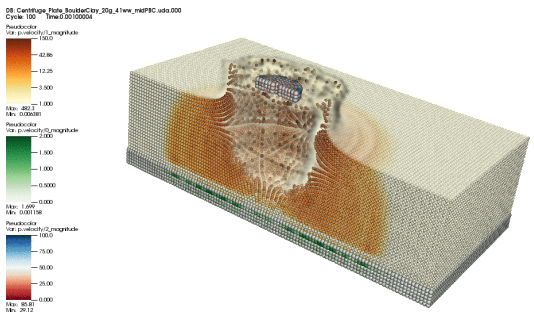


(d) Pressure: Wet Mason sand containing 10% water by weight

Figure 62 – *Accelerations and pressures in the soil caused by an explosion simulated inside Mason sand in a centrifuge bucket.*

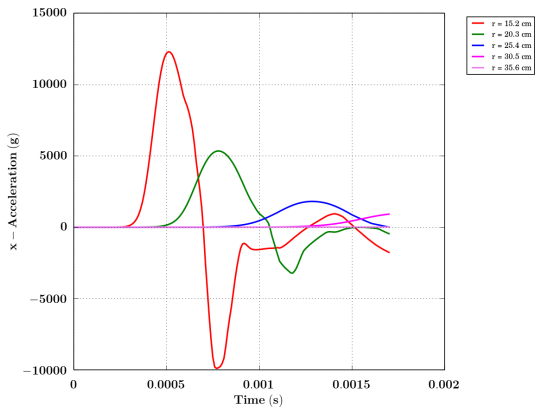


(a) Dry Boulder clay

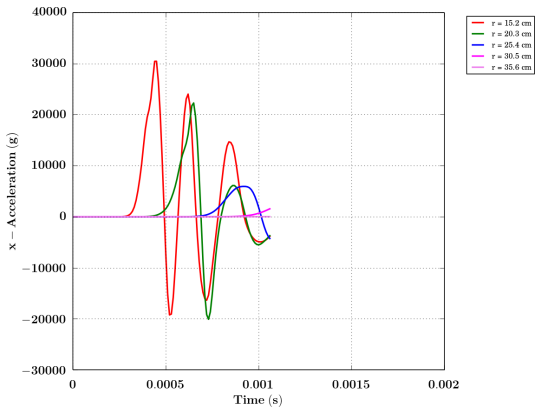


(b) Wet Boulder clay containing 41% water by weight

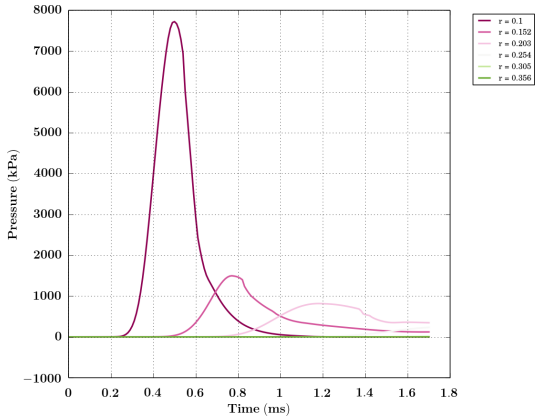
Figure 63 – *Boulder clay explosion simulation with a plate inside a centrifuge at 20g.*



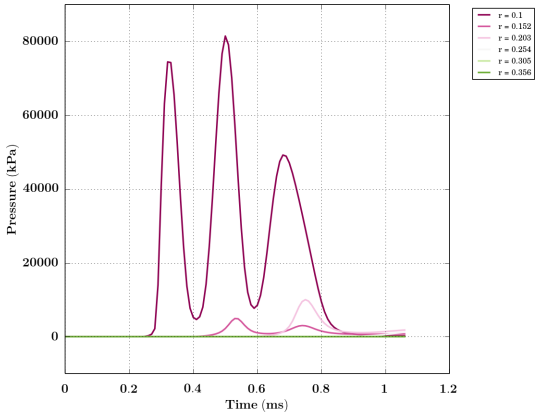
(a) Dry Boulder clay



(b) Wet Boulder clay containing 41% water by weight



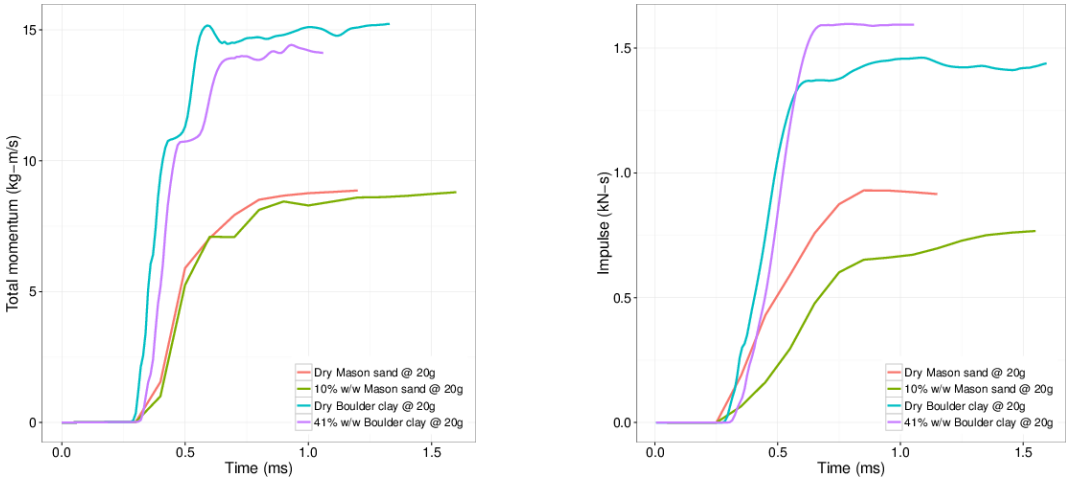
(c) Dry Boulder clay



(d) Wet Boulder clay containing 41% water by weight

Figure 64 – *Boulder clay accelerations and pressures caused by a simulated explosion in a centrifuge bucket.*

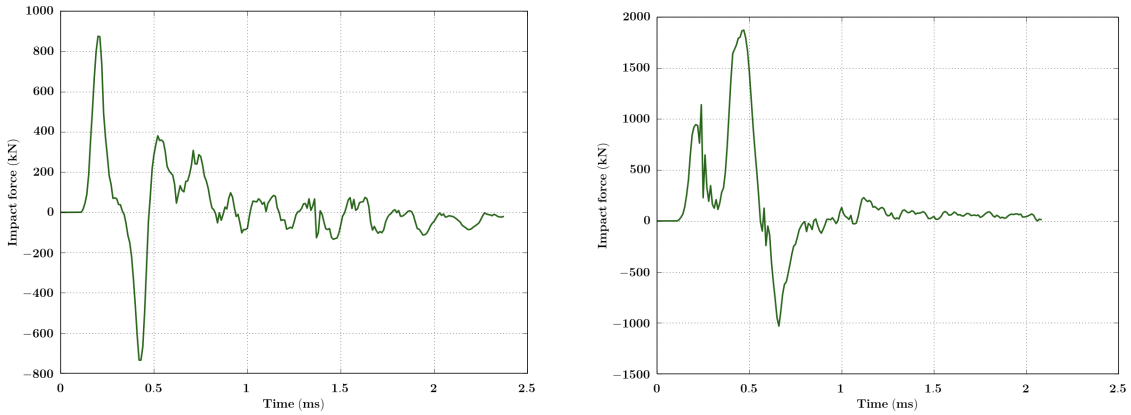
discussed above. Figure 65(a) and (b) show the evolution of the momentum of the target and the corresponding "impulse" values. The momentum imparted by the sand is lower than that imparted by the clay However, there appears to be no significant difference between dry and wet soil samples. On the other hand, if we examine the impulse, the dry sand appears to provide greater impulse that the wet sand and the converse holds for clay. We are of the opinion that the impulse value is less accurate than the momentum value but note that the impulse value is typically easier to measure in an experiment.



(a) Total target momentum (b) Target impulse

Figure 65 – *Momentum and impulse of the target cause by simulations explosions in Colorado Mason sand and Boulder clay.*

If, instead of a height of 5.1 cm, we place the target plate in contact with the soil surface we observe a larger peak impact force on the target due to the wet clay. This can be seen in the plots in Figure 66. There also seem to be two impact peaks for the wet Boulder clay which may possibly be seen in experiments.

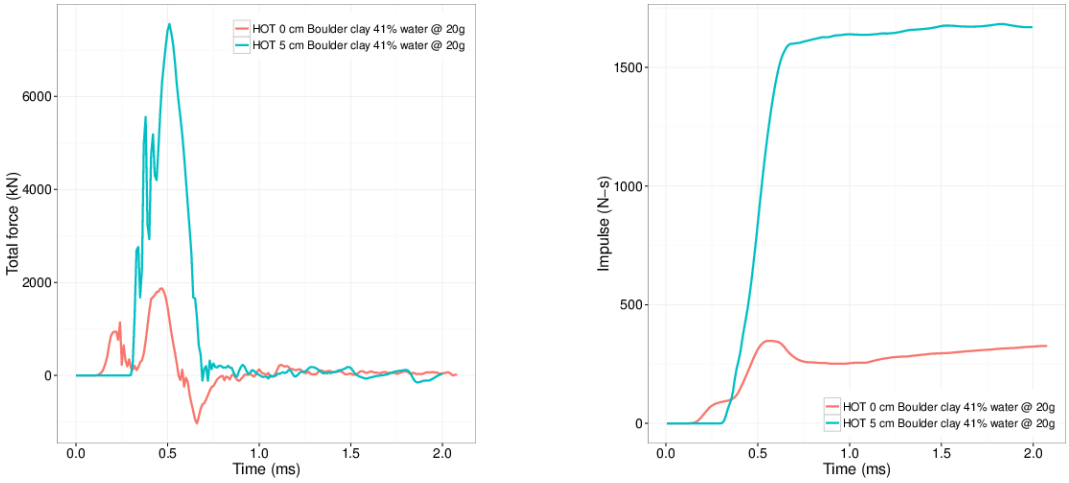


(a) Dry Boulder clay (b) Wet Boulder clay containing 41% water by weight

Figure 66 – *Impact forces on a target place on the surface of Boulder clay in a centrifuge bucket due at explosion at a depth of 5.1 cm.*

A comparison of the total force on the target and impulse for the simulation with Boulder clay containing 41% water (by weight) is shown in Figure 67. Counterintuitively, if the target is placed at the surface it experiences a smaller impulse than than if it is placed 5.1 cm above the surface. This observation,

if confirmed by experiments, would suggest that the position of the hull in a vehicle be lowered if the impact of a blast on a vehicle is to be reduced.



(a) Force on target (b) Impulse on target

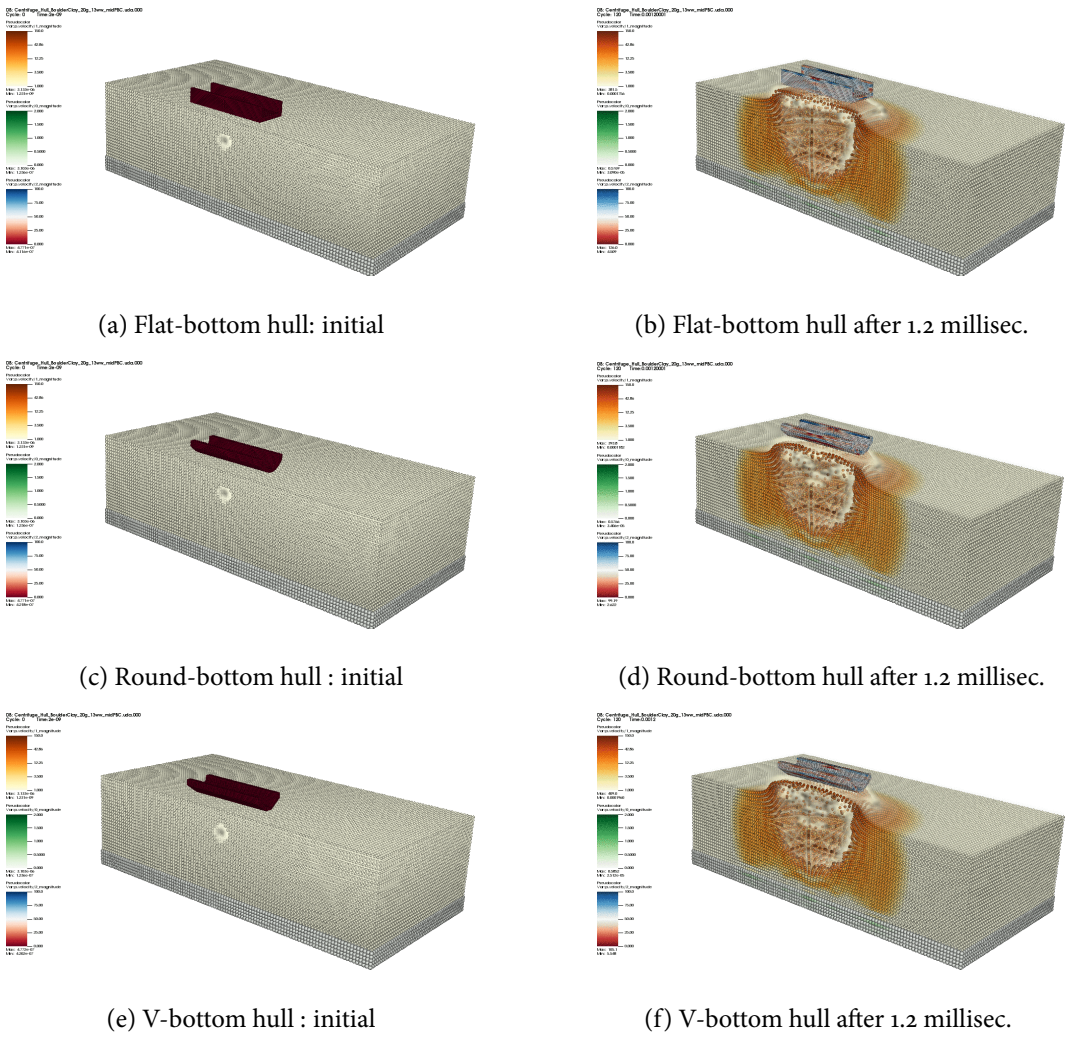
Figure 67 – *Comparison of force and impulse on the target for two heights of target (HOT).*

10.3.4 Explosion under a vehicle

Of particular interest in engineering design is the situation where an explosion occurs under a vehicle. Experimental studies have shown that a V-shaped hull in the undercarriage of a vehicle is effective at reducing the effect of the blast. The following simulations have been performed to find out whether the ARENA model also predicts similar results.

We continue simulating explosions under centrifuge loading at 20g. The soil is Boulder clay with 13% water by weight. Three hull shapes are tested: a plat hull, a round hull, and a V-shaped hull. The hulls are initially positioned a short distance from the surface of the soil. We have chosen the geometry of the hulls such that the volumes of the three hulls are identical. Figure 68 shows the initial configurations of the simulations and the deformed state a short time (1.2 millisec) after the explosion has been initiated. Nominally, there does not appear to be any difference between the the three cases.

However, if we examine the total momentum imparted to each hull, we notice that the V-shaped hull experiences a smaller velocity than the flat or the round hull (see Figure 69). In fact, the flat and the round hull have essentially the same momentum transferred to them by the explosion. This observation suggests that the ARENA model may have merit if used to predict the response of soils even under conditions that involve a significant amount of disaggregation.



(a) Flat-bottom hull: initial

(b) Flat-bottom hull after 1.2 millisec.

(c) Round-bottom hull : initial

(d) Round-bottom hull after 1.2 millisec.

(e) V-bottom hull : initial

(f) V-bottom hull after 1.2 millisec.

Figure 68 – *Initial and deformed configurations after an explosion at 20g in Boulder clay containing 13% water by weight.*

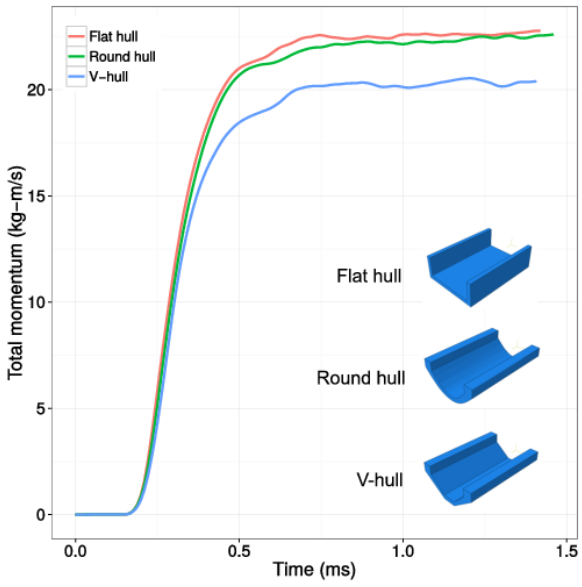


Figure 69 – *The momentum transferred to the three hull shapes from an explosion in Boulder clay containing 13% water by weight.*

11 Observations and discussion

Some observations about the performance of the ARENA partially saturated soil model are listed below.

- When the model is driven using pressures or plastic strains, the results are reasonable (see Section 7). Interesting features include a kink in the volumetric plastic strain vs. hydrostatic compressive strength curves and an increase in the plastic porosity after a plastic volumetric strain threshold is exceeded.
- Single particle hydrostatic, compressive, tensile, and multiple strain path tests were performed with the partially saturated model. The results are reasonable and have been verified where possible (see Section 8).
- In some uniaxial extension tests, the closest point projection on the yield surface can produce a compression mean stress (p) if the trial stress falls outside the yield surface in the first timestep. This can possibly be mitigated by choosing a β-nonassociativity parameter that produces the expected tensile stress .
- In Section 9, we use the parameters fitted to a dry Mason sand sample and the full machinery of the partially saturated soil model to predict the behavior of Mason sand containing 18.4% water by weight. The correspondence between our predictions and experimental data is remarkable.
- The damage model was fitted using a dry Mason sand SHPB test, and matched experiments reasonably well during the loading phase. However, the simulations did not capture the observed decrease in the radial stress with when the deformation gradient became constant after failure. Such a decrease would have been observed in the simulations if the hydrostatic strength in the ARENA model had been allowed to decrease after failure. We do not have any direct experimental evidence that the hydrostatic strength decreases with increasing confinement, so further study is warranted.
- The parameters fitted directly to the quasistatic data for Boulder clay did not produce a good match to the SHPB experiments on dry Boulder clay. We had to reduce the crush curve steepness to match the SHPB data for dry clay.
- Simulations of Boulder clay mixed with 40.8% water by weight suggested that the clay swells significantly (the porosity remains nearly constant, thereby reducing the saturation). If we use a saturation of 0.9 and a value of $p_1^{\text{sat}} = 5$, we can match the experimental data reasonably well. Addressing this problem requires using parameter values inconsistent with calibration, suggesting a need for further research.
- The fluctuations in the experimental values of r seen in the saturated Boulder clay area because small difference in axial and radial stresses are not resolved by the SHPB instrumentation system.
- In Section 9, we used our partially saturated soil model with a saturation of 30% and dry clay parameters to predict the SHPB response of a clay containing 12.8% water by weight. We were able to match the experiments reasonably only after using a value of $p_1^{\text{sat}} = -1$. This implies that the hydrostatic strength actually decreases relative to the dry clay at this level of saturation. We believe that the compressive strength decreases with saturation in this case because the clay grains can move relative to each other more easily when a small amount of water is added, thus decreasing the strength relative to that of the dry material. Beyond a certain level of saturation, this trend is reversed, as seen in the simulations with clay containing 40.8% water by weight. We recommend further research to investigate our hypothesis that compressive strength decreases with saturation because of a grain lubrication effect.
- From the piston impact simulations in Section 10, we observe that the wet soil provides more momentum to the target plate than the dry soil. Whether this is purely due to the higher density of the wet soil or is influenced by the elastoplastic soil model requires further research. We note that the terminal velocity of the target plate reaches approximately the same value for both wet and dry clay even though the displacement is larger for the wet clay.

- Simulations of a clay lump impacting a hollow steel box indicate that, if the clay is wet, larger volumetric plastic strains develop in the material than if it is dry. We also observe that wet clay particles appear to dissipate less energy and disaggregate after than dry clay. This is counterintuitive and needs further research. As was seen in the piston impact simulations, the wet clay displaces the target further than the dry clay - in line with experimental observations. The lump impact simulation clearly indicates the need for further research into disaggregation models for soil because the continuum assumption breaks down at large strains.
- The spin-up simulations that emulate the effect of centrifugal forces (see Section 10.3.1) indicate that the ARENA model predicts that the soil can experience inelastic strains and damage even before any explosion has occurred. Whether this is realistic can only be determined after further research. Interestingly, the stress in the material settles to a steady value a few milliseconds after the centrifugal load is applied.
- The explosion phase of the centrifuge simulations is simplified considerably if a pressure pulse is used to represent the explosive gases. However, this is a crude approximation and research is needed to determine if the JWL++ model can be used instead. Our simulations with JWL++ (not discussed in this report) indicate that there is a loss of symmetry due to inaccuracies in the MPM contact model used in Vaango.
- Experimental pressure pulse and acceleration data from explosions inside a centrifuge indicate that there is a significant amount of damping in the soil that is not reflected in our simulations (see Section 10.3.2). MPM with ARENA predicts reflections and sinusoidal waves that are not observed in experiments. That can be caused by the grid resolution and symmetry boundary conditions and suggests further research.
- Simulations of explosions under a target plate has been presented in Section 10.3.3. These simulations predict that particle accelerations are larger in dry sand than in wet sand. However, in wet clay we observe much higher accelerations than in dry clay. The cause of that difference between the response of Colorado Mason sand and Boulder clay is unclear and suggests further exploration.
- If we examine the momentum imparted to the target plate by sand and clay, we notice that clay imparts more momentum than sand. However, we notice no significant difference between the momentum transferred by dry and wet soils.
- If is also interesting to note that a target at a height of 5.1 cm above the soil surface experiences a larger impulse load than a target that is on the surface.
- Simulations of an explosion under various shaped hulls suggest that a V-shaped hull is marginally better than flat or round hulls in that less momentum is transferred from the soil to the hull (see Section 10.3.4)

References

Adams, B. M. et al. (2009). "DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user's manual". In: *Sandia National Laboratories, Tech. Rep. SAND2010-2183* (cit. on p. 64).

Aharonov, Einat et al. (2013). "Localization of Shear in Saturated Granular Media: Insights from a Multi-Scaled Granular-Fluid Model". In: *arXiv preprint arXiv:1310.4912* (cit. on p. 6).

Alonso, E. E, A. Gens, A. Josa, et al. (1990). "Constitutive model for partially saturated soils". In: *Géotechnique* 40.3, pp. 405–430 (cit. on p. 6).

Andrade, J. E. and K. C. Ellison (2008). "Evaluation of a predictive constitutive model for sands". In: *Journal of geotechnical and geoenvironmental engineering* 134.12, pp. 1825–1828 (cit. on p. 6).

Arson, C. and B. Gatmiri (2009). "A mixed damage model for unsaturated porous media". In: *Comptes Rendus Mécanique* 337.2, pp. 68–74 (cit. on p. 6).

Berg, Charles R. (1995). "A simple, effective-medium model for water saturation in porous rocks". In: *Geophysics* 60.4, pp. 1070–1080 (cit. on p. 6).

Berryman, J. G. (2006). "Effective medium theories for multicomponent poroelastic composites". In: *Journal of Engineering Mechanics* 132.5, pp. 519–531 (cit. on p. 22).

Berryman, J. G. and G. W. Milton (1991). "Exact results for generalized Gassmann's equations in composite porous media with two constituents". In: *Geophysics* 56.12, pp. 1950–1960 (cit. on p. 22).

Bishop, A. W. and A.E. Skinner (1977). "The influence of high pore-water pressure on the strength of cohesionless soils". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 284.1318, pp. 91–130 (cit. on p. 6).

Borja, R. I. (2004). "Cam-Clay plasticity. Part V: A mathematical framework for three-phase deformation and strain localization analyses of partially saturated porous media". In: *Computer Methods in Applied Mechanics and Engineering* 193.48, pp. 5301–5338 (cit. on pp. 6, 9).

– (2006). "On the mechanical energy and effective stress in saturated and unsaturated porous continua". In: *International Journal of Solids and Structures* 43.6, pp. 1764–1786 (cit. on pp. 6, 9, 21).

Borja, R. I. and J. A. White (2010). "Conservation laws for coupled hydromechanical processes in unsaturated porous media: Theory and implementation". In: *Mechanics of Unsaturated Geomaterials*, pp. 185–208 (cit. on pp. 6, 9).

Brannon, R. M. (2007). "Elements of phenomenological plasticity: geometrical insight, computational algorithms, and topics in shock physics". In: *ShockWave Science and Technology Reference Library*. Springer, pp. 225–274 (cit. on pp. 20, 21, 24, 36).

Brannon, R. M., T. J. Fuller, et al. (2015). "KAYENTA: Theory and User's Guide". In: *Sandia National Laboratories report SAND2015-0803* (cit. on pp. 9, 10, 21, 29, 30, 68, 85).

Brannon, R. M. and S. Leelavanichkul (2010). "A multi-stage return algorithm for solving the classical damage component of constitutive models for rock, ceramics, and other rock-like media". In: *International Journal of Fracture* (cit. on p. 20).

Brown, J.L. et al. (2007). "Dynamic compaction of sand". In: *SHOCK COMPRESSION OF CONDENSED MATTER-2007: Proceedings of the Conference of the American Physical Society Topical Group on Shock Compression of Condensed Matter*. Vol. 955. AIP Publishing, pp. 1363–1366 (cit. on p. 6).

Buscarnera, G. and I. Einav (2012). "The yielding of brittle unsaturated granular soils". In: *Géotechnique* 62.2, p. 147 (cit. on p. 6).

Buscarnera, Giuseppe (2014). "Uniqueness and existence in plasticity models for unsaturated soils". In: *Acta Geotechnica* 9.2, pp. 313–327 (cit. on p. 6).

Cao, T. D., L. Sanavia, and B. A. Schrefler (2016). "A thermo-hydro-mechanical model for multiphase geomaterials in dynamics with application to strain localization simulation". In: *International Journal for Numerical Methods in Engineering* (cit. on p. 6).

Coussy, O. (2007). "Revisiting the constitutive equations of unsaturated porous solids using a Lagrangian saturation concept". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 31.15, pp. 1675–1694 (cit. on p. 6).

Coussy, O. and S. Brisard (2009). "Prediction of drying shrinkage beyond the pore isodeformation assumption". In: *Journal of Mechanics of Materials and Structures* 4.2, pp. 263–279 (cit. on p. 6).
Coussy, O., J.-M. Pereira, and J. Vaunat (2010). "Revisiting the thermodynamics of hardening plasticity for unsaturated soils". In: *Computers and Geotechnics* 37.1, pp. 207–215 (cit. on p. 6).
Davison de St Germain, J et al. (2000). "Uintah: A massively parallel problem solving environment". In: *High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on*. This has a generic bibtex key (just Uintah) so that it can be edited to be whatever is currently considered the main reference for Uintah, not any particular publication. IEEE, pp. 33–41 (cit. on pp. 7, 9).
Dumont, Matthieu et al. (2011). "A thermo-hydro-mechanical model for unsaturated soils based on the effective stress concept". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 35.12, pp. 1299–1317 (cit. on p. 6).
Dvorkin, J. et al. (1999). "Identifying patchy saturation from well logs". In: *Geophysics* 64.6, pp. 1756–1759 (cit. on p. 22).
Faria Santos, S. H. (2003). "Mechanics and Thermodynamics of Mixtures with Continuous Diversity". PhD thesis. TU Darmstadt (cit. on p. 6).
Fuentes, W. and Th. Triantafyllidis (2013). "Hydro-mechanical hypoplastic models for unsaturated soils under isotropic stress conditions". In: *Computers and Geotechnics* 51, pp. 72–82 (cit. on p. 6).
Fuller, T. J. and R. M. Brannon (2012). "On the effects of deformation-induced anisotropy on classical elastic-plastic materials. ." In: *Int. J. Numerical and Analytical Methods in Geomechanics* 37.9, pp. 1079–1094 (cit. on p. 21).
Gajo, A. (2011). "Finite strain hyperelastoplastic modelling of saturated porous media with compressible constituents". In: *International Journal of Solids and Structures* 48.11, pp. 1738–1753 (cit. on p. 6).
Gawin, D., B. A. Schrefler, and M. Galindo (1996). "Thermo-hydro-mechanical analysis of partially saturated porous materials". In: *Engineering Computations* 13.7, pp. 113–143 (cit. on p. 6).
Georgiadis, K. (2003). "Development, implementation and application of partially saturated soil models in finite element analysis". PhD thesis. Imperial College (cit. on p. 6).
Goren, L et al. (2010). "Pore pressure evolution in deforming granular material: A general formulation and the infinitely stiff approximation". In: *Journal of Geophysical Research: Solid Earth* 115.B9 (cit. on p. 6).
Grasley, Z. C. and K. R. Rajagopal (2012). "Revisiting total, matric, and osmotic suction in partially saturated geomaterials". In: *Zeitschrift für angewandte Mathematik und Physik* 63.2, pp. 373–394 (cit. on p. 6).
Gray, W. G. and S. M. Hassanizadeh (1989). "Averaging theorems and averaged equations for transport of interface properties in multiphase systems". In: *International Journal of Multiphase Flow* 15.1, pp. 81–95 (cit. on pp. 6, 9).
– (1991a). "Paradoxes and realities in unsaturated flow theory". In: *Water Resour. Res* 27.8, pp. 1847–1854 (cit. on p. 9).
– (1991b). "Unsaturated flow theory including interfacial phenomena". In: *Water Resour. Res* 27.8, pp. 1855–1863 (cit. on pp. 6, 9).
Gray, W. G. and B. A. Schrefler (2001). "Thermodynamic approach to effective stress in partially saturated porous media". In: *European Journal of Mechanics-A/Solids* 20.4, pp. 521–538 (cit. on p. 6).
– (2007). "Analysis of the solid phase stress tensor in multiphase porous media". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 31.4, pp. 541–581 (cit. on p. 6).
Gray, W. G., B. A. Schrefler, and F. Pesavento (2009). "The solid phase stress tensor in porous media mechanics and the Hill–Mandel condition". In: *Journal of the Mechanics and Physics of Solids* 57.3, pp. 539–554 (cit. on p. 6).
Grujicic, Mica et al. (2008). "Parameterization of the porous-material model for sand with different levels of water saturation". In: *Soil Dynamics and Earthquake Engineering* 28.1, pp. 20–35 (cit. on p. 6).

Grujicic, M. et al. (2009). "Development, parameterization, and validation of a visco-plastic material model for sand with different levels of water saturation". In: *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials Design and Applications* 223.2, pp. 63–81 (cit. on p. 27).

Hamiel, Y., V. Lyakhovsky, and A. Agnon (2004). "Coupled evolution of damage and porosity in poroelastic media: theory and applications to deformation of porous rocks". In: *Geophysical Journal International* 156.3, pp. 701–713 (cit. on p. 6).

Hassanizadeh, S. M. and W. G. Gray (1979). "General conservation equations for multi-phase systems: 1. Averaging procedure". In: *Advances in Water Resources* 2, pp. 131–144 (cit. on pp. 6, 9).

– (1990). "Mechanics and thermodynamics of multiphase flow in porous media including interphase boundaries". In: *Advances in water resources* 13.4, pp. 169–186 (cit. on pp. 6, 9, 15, 19).

Higgins, W. and Dipanjan Basu (2011). *A HIGH STRAIN-RATE CONSTITUTIVE MODEL FOR SAND WITH APPLICATION IN FINITE ELEMENT ANALYSIS Internal Geotechnical Report 2011-4*. Tech. rep. University of Connecticut. URL: http://digitalcommons.uconn.edu/cee%5C_techreports/4/ (cit. on p. 6).

Homel, M. A., J. E. Guilkey, and R. M. Brannon (2015). "Continuum effective-stress approach for high-rate plastic deformation of fluid-saturated geomaterials with application to shaped-charge jet penetration". In: *Acta Mechanica* 227.2, pp. 279–310 (cit. on pp. 9, 19, 31–33, 55).

– (2017). "Mesoscale Validation of the Effective Stress approach for Modeling the Plastic Deformation of Fluid-Saturated Porous Materials". In: *Journal of Dynamic Behavior of Materials* (cit. on p. 6).

Houlsby, G.T. (1997). "The work input to an unsaturated granular material". In: *Géotechnique* 47.1, pp. 193–196 (cit. on p. 6).

Itskov, M. (2004). "On the applicability of generalized strain measures in large strain plasticity". In: *Mechanics Research Communications* (cit. on pp. 7, 39).

Kamojjala, K. C. et al. (2015). "Verification tests in solid mechanics". In: *Engineering with Computers* 2, pp. 193–213 (cit. on pp. 9, 85).

Kamrin, Ken and Martin Z. Bazant (2007). "Stochastic flow rule for granular materials". In: *Physical Review E* 75. File is Kamrin07.pdf, 1–28 (preprint) (cit. on p. 7).

Kohler, R. and G. Hofstetter (2008). "A cap model for partially saturated soils". In: *International Journal for Numerical and analytical methods in Geomechanics* 32.8, pp. 981–1004 (cit. on p. 6).

Lade, P. V. (2005). *Overview of constitutive models for soils* (cit. on p. 6).

Lakeland, Daniel L., Amy Rechenmacher, and Roger Ghanem (2014). "Towards a complete model of soil liquefaction: the importance of fluid flow and grain motion". In: *Proc. R. Soc. A* 470.2165, p. 20130453 (cit. on p. 6).

Le Pense, S., C. Arson, and A. Pouya (2016). "A fully coupled damage-plasticity model for unsaturated geomaterials accounting for the ductile–brittle transition in drying clayey soils". In: *International Journal of Solids and Structures* (cit. on p. 6).

Li, C., R. I. Borja, and R. A. Regueiro (2004). "Dynamics of porous media at finite strain". In: *Computer methods in applied mechanics and engineering* 193.36, pp. 3837–3870 (cit. on pp. 6, 9).

Li, X. et al. (2016). "Meso-hydro-mechanically informed effective stresses and effective pressures for saturated and unsaturated porous media". In: *European Journal of Mechanics-A/Solids* 59, pp. 24–36 (cit. on p. 6).

Liu, Chunyang and Kanthasamy K. Muraleetharan (2011). "Coupled hydro-mechanical elastoplastic constitutive model for unsaturated sands and silts. I: Formulation". In: *International Journal of Geomechanics* 12.3, pp. 239–247 (cit. on p. 6).

Madeo, Angela, Francesco Dell'Isola, and Félix Darve (2013). "A continuum model for deformable, second gradient porous media partially saturated with compressible fluids". In: *Journal of the Mechanics and Physics of Solids* 61.11, pp. 2196–2211 (cit. on p. 6).

Mašín, D. and N. Khalili (2012). "A thermo-mechanical model for variably saturated soils based on hypoplasticity". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 36.12, pp. 1461–1485 (cit. on p. 6).

Meroi, E. A., B. A. Schrefler, and O. C. Zienkiewicz (1995). "Large strain static and dynamic semisaturated soil behaviour". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 19.2, pp. 81–106 (cit. on p. 6).

Mo es, Nicolas, John Dolbow, and Ted Belytschko (1999). "A finite element method for crack growth without remeshing". In: *International Journal for Numerical Methods in Engineering* 46.1, pp. 131–150 (cit. on p. 9).

Nedjar, B. (2013). "Formulation of a nonlinear porosity law for fully saturated porous media at finite strains". In: *Journal of the Mechanics and Physics of Solids* 61.2, pp. 537–556 (cit. on p. 6).

Nikooee, Ehsan et al. (2012). "The effective stress in unsaturated soils: Insights from thermodynamics". In: *Unsaturated soils: Research and applications*. Springer, pp. 5–11 (cit. on p. 6).

Norris, A. N. (2008). "Eulerian conjugate stress and strain". In: *Journal of Mechanics of Materials and Structures* 3.2, pp. 243–260 (cit. on p. 20).

Pabst, W and E Gregorová (2015). "Critical Assessment 18: elastic and thermal properties of porous materials–rigorous bounds and cross-property relations". In: *Materials Science and Technology* 31.15, pp. 1801–1808 (cit. on p. 29).

Papastavrou, Areti and Paul Steinmann (2010). "On deformational and configurational poro-mechanics: dissipative versus non-dissipative modelling of two-phase solid/fluid mixtures". In: *Archive of Applied Mechanics* 80.9, pp. 969–984 (cit. on p. 6).

Passarotto, Mareva et al. (2009). *Coupled hydro-thermo-mechanical analysis of a deep radioactive waste disposal based on porous media mechanics*. Tech. rep. Universite Libre de Bruxelles, Belgium (cit. on p. 6).

Pietruszczak, S. and G.N. Pande (1996). "Constitutive relations for partially saturated soils containing gas inclusions". In: *Journal of geotechnical engineering* 122.1, pp. 50–59 (cit. on p. 6).

Reguerio, R. et al. (2013). "Dynamic Behavior of Materials Volume 1, Chapter: Chapter 42, Ed: B. Song". In: Springer, Vienna. Chap. ONR MURI Project on Soil Blast Modeling and Simulation, pp. 341–353 (cit. on pp. 6, 7).

Rossi, L. F. (2002). "A High Order Lagrangian Scheme for Flow Through". In: *Fluid Flow and Transport in Porous Media, Mathematical and Numerical Treatment: Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference on Fluid Flow and Transport in Porous Media, Mathematical and Numerical Treatment, June 17-21, 2001, Mount Holyoke College, South Hadley, Massachusetts*. Vol. 295. American Mathematical Soc., p. 433 (cit. on p. 6).

Rubin, M. B. and M. Ichihara (2010). "Rheological models for large deformations of elastic-viscoplastic materials". In: *Int. J. Eng. Sci.* 48, pp. 1534–1543 (cit. on p. 7).

Sanavia, L. et al. (2014). "A model for non-isothermal variably saturated porous media in dynamics". In: *International Center for Numerical Methods in Engineering* (cit. on p. 6).

Santagiuliana, R. and B. A. Schrefler (2006). "Enhancing the Bolzon–Schrefler–Zienkiewicz constitutive model for partially saturated soil". In: *Transport in Porous Media* 65.1, pp. 1–30 (cit. on p. 6).

Schrefler, B. A. and F. Pesavento (2004). "Multiphase flow in deforming porous material". In: *Computers and Geotechnics* 31.3, pp. 237–250 (cit. on p. 6).

Schrefler, B. A. and R. Scotta (2001). "A fully coupled dynamic model for two-phase fluid flow in deformable porous media". In: *Computer methods in applied mechanics and engineering* 190.24, pp. 3223–3246 (cit. on p. 6).

Serpieri, Roberto and Francesco Travascio (2016). *A medium-independent variational macroscopic theory of two-phase porous media–Part I: Derivation of governing equations and stress partitioning laws*. Tech. rep. CCSD. URL: https://hal.archives-ouvertes.fr/hal-01281182/ (cit. on p. 6).

Serpieri, Roberto, Francesco Travascio, et al. (2015). "Variationally consistent derivation of the stress partitioning law in saturated porous media". In: *International Journal of Solids and Structures* 56, pp. 235–247 (cit. on p. 6).

Song, X. and R. I. Borja (2014a). "Finite deformation and fluid flow in unsaturated soils with random heterogeneity". In: *Vadose Zone Journal* 13.5 (cit. on p. 9).

– (2014b). "Mathematical framework for unsaturated flow in the finite deformation range". In: *International Journal for Numerical Methods in Engineering* 97.9, pp. 658–682 (cit. on pp. 6, 9).

Strack, O. E., R. B. Leavy, and R. M. Brannon (2014). "Aleatory uncertainty and scale effects in computational damage models for failure and fragmentation". In: *International Journal for Numerical Methods in Engineering* (cit. on pp. 7, 79).

Sulsky, D., Z. Chen, and H. Schreyer (1994). "A particle method for history-dependent materials". In: *Computer Methods in Applied Mechanics and Engineering* 118, pp. 179–196 (cit. on p. 7).

Svendsen, B. and K. Hutter (1995). "On the thermodynamics of a mixture of isotropic materials with constraints". In: *International Journal of Engineering Science* 33.14, pp. 2021–2054 (cit. on p. 6).

Thomas, M. A. and D. E. Chitty (2011). *Constitutive Soil Properties for Mason Sand and Kennedy Space Center*. Tech. rep. NASA/CR-2011-217323. Applied Research Associates, Inc., Albuquerque, New Mexico. URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120000504.pdf (cit. on p. 71).

Uzuoka, R. and R. I. Borja (2012). "Dynamics of unsaturated poroelastic solids at finite strain". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 36.13, pp. 1535–1573 (cit. on pp. 6, 9).

Vlahinić, I., H. M. Jennings, J. E. Andrade, et al. (2011). "A novel and general form of effective stress in a partially saturated porous material: The influence of microstructure". In: *Mechanics of Materials* 43.1, pp. 25–35 (cit. on p. 6).

Vlahinić, I., H. M. Jennings, and J. J. Thomas (2009). "A constitutive model for drying of a partially saturated porous material". In: *Mechanics of Materials* 41.3, pp. 319–328 (cit. on p. 6).

Wong, K. S. and D. Mašín (2014). "Coupled hydro-mechanical model for partially saturated soils predicting small strain stiffness". In: *Computers and Geotechnics* 61, pp. 355–369 (cit. on p. 6).

Zhou, A. and D. Sheng (2015). "An advanced hydro-mechanical constitutive model for unsaturated soils with different initial densities". In: *Computers and Geotechnics* 63, pp. 46–66 (cit. on p. 6).

Appendices

A Pore pressure evolution from mass balance

Mass balance provides the following equation governing the rate of porosity ϕ associated with the solid skeleton velocity $\mathbf{v}^s$:

$$\begin{aligned} &\frac{D\phi}{Dt} - (B-\phi)\nabla\cdot\mathbf{v}^s = 0 \\ &\left[\frac{S_w\phi}{K_w(p^w)} + \frac{(1-S_w)\phi}{K_a(p^w)}\right]\frac{D\overline{p^w}}{Dt} + B\nabla\cdot\mathbf{v}^s = 0\,. \end{aligned} \tag{283}$$

Here, B is the Biot parameter and the scripts "w" refer to the water.Combining these two equations gives

$$\left[\frac{S_w\phi}{K_w(p^w)} + \frac{(1-S_w)\phi}{K_a(p^w)}\right]\frac{D\overline{p^w}}{Dt} + \left(\frac{B}{B-\phi}\right)\frac{D\phi}{Dt} = 0\,. \tag{284}$$

From equation (139),

$$\phi = (1-S_0)\phi_0\exp(\varepsilon_v^a - \varepsilon_v) + S_0\phi_0\exp(\varepsilon_v^w - \varepsilon_v) \tag{285}$$

Therefore, the material derivative of the porosity is

$$\frac{D\phi}{Dt} = (1-S_0)\phi_0\exp(\varepsilon_v^a - \varepsilon_v)\left(\frac{D\varepsilon_v^a}{Dt} - \frac{D\varepsilon_v}{Dt}\right) + S_0\phi_0\exp(\varepsilon_v^w - \varepsilon_v)\left(\frac{D\varepsilon_v^w}{Dt} - \frac{D\varepsilon_v}{Dt}\right) \tag{286}$$

Substituting (286) into (284) gives

$$\begin{aligned} &\left[\frac{S_w\phi}{K_w(p^w)} + \frac{(1-S_w)\phi}{K_a(p^w)}\right]\frac{D\overline{p^w}}{Dt} + \left(\frac{B\phi_0}{B-\phi}\right)\Bigg[(1-S_0)\exp(\varepsilon_v^a - \varepsilon_v)\left(\frac{D\varepsilon_v^a}{Dt} - \frac{D\varepsilon_v}{Dt}\right) + \\ &\qquad S_0\exp(\varepsilon_v^w - \varepsilon_v)\left(\frac{D\varepsilon_v^w}{Dt} - \frac{D\varepsilon_v}{Dt}\right)\Bigg] = 0\,. \end{aligned} \tag{287}$$

For equilibrated fluid pressures, $\overline{p^a} = \overline{p^w}$, and we can write the above using quantities that are positive in compression as

$$\begin{aligned} &\left[\frac{S_w\phi}{K_w} + \frac{(1-S_w)\phi}{K_a}\right]\frac{D\overline{p^w}}{Dt} - \left(\frac{B\phi_0}{B-\phi}\right)\left[(1-S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a})\frac{\partial\overline{\varepsilon_v^a}}{\partial\overline{p^w}} + S_0\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\frac{\partial\overline{\varepsilon_v^w}}{\partial\overline{p^w}}\right]\frac{D\overline{p^w}}{Dt} \\ &\quad = -\left(\frac{B\phi_0}{B-\phi}\right)\left[(1-S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + S_0\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\right]\frac{D\overline{\varepsilon_v}}{Dt} \end{aligned} \tag{288}$$

Noting that

$$\frac{\partial\overline{\varepsilon_v^a}}{\partial\overline{p^w}} = \frac{1}{K_a} \quad \text{and} \quad \frac{\partial\overline{\varepsilon_v^w}}{\partial\overline{p^w}} = \frac{1}{K_w} \tag{289}$$

we have

$$\begin{aligned} &\left[\frac{S_w\phi}{K_w} + \frac{(1-S_w)\phi}{K_a}\right]\frac{D\overline{p^w}}{Dt} - \left(\frac{B\phi_0}{B-\phi}\right)\left[\frac{1-S_0}{K_a}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + \frac{S_0}{K_w}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\right]\frac{D\overline{p^w}}{Dt} \\ &\quad = -\left(\frac{B\phi_0}{B-\phi}\right)\left[(1-S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + S_0\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\right]\frac{D\overline{\varepsilon_v}}{Dt} \end{aligned} \tag{290}$$

or

$$\begin{aligned} \frac{D\overline{\varepsilon_v}}{Dt} = \frac{1}{(1-S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + S_0\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})}\Bigg[&-\frac{(B-\phi)\phi}{B\phi_0}\left(\frac{S_w}{K_w} + \frac{1-S_w}{K_a}\right) + \\ &\frac{1-S_0}{K_a}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + \frac{S_0}{K_w}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\Bigg]\frac{D\overline{p^w}}{Dt} \end{aligned} \tag{291}$$

We can convert this equation into the form

$$\frac{d\overline{p^w}}{d\overline{\varepsilon_v}} = \frac{1}{\mathcal{B}} \tag{292}$$

where

$$\mathcal{B} := \frac{1}{(1-S_0)\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + S_0\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})}\left[-\frac{(B-\phi)\phi}{B\phi_0}\left(\frac{S_w}{K_w} + \frac{1-S_w}{K_a}\right) + \frac{1-S_0}{K_a}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^a}) + \frac{S_0}{K_w}\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\right]. \tag{293}$$

We can then find the total pore pressure using

$$\overline{p^w} = \int \frac{d\overline{p^w}}{d\overline{\varepsilon_v}}\mathrm{d}\overline{\varepsilon_v}\,. \tag{294}$$

For a fully saturated medium, $S_0 = 1$, we have

$$\mathcal{B} = \frac{1}{K_w\exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})}\left[-\frac{(B-\phi)\phi}{B\phi_0} + \exp(\overline{\varepsilon_v} - \overline{\varepsilon_v^w})\right]. \tag{295}$$

Note that a medium with zero initial porosity, $\phi_0 = 0$, cannot be modeled using our approach.

B Computing saturation from weight ratio

The experimental SHPB data provided by UT Dallas includes the dry sand density and the water content in the form of the ratio by weight of the water to the sand/soil. We have to convert that information into porosity, saturation, and mass density.

To do that we assume that the soil skeleton material has the density of quartz such that the mass (m) of the dry sand can be written as

$$m_{\text{drysand}} = m_{\text{air}} + m_{\text{quartz}} \,. \tag{296}$$

Expressed in terms of volume (v) and density (ρ),

$$\rho_{\text{drysand}} v_{\text{drysand}} = \rho_{\text{air}} v_{\text{air}} + \rho_{\text{quartz}} v_{\text{quartz}} \,. \tag{297}$$

If we neglect the density of air, we have

$$\rho_{\text{drysand}} v_{\text{drysand}} \approx \rho_{\text{quartz}} v_{\text{quartz}} \quad \Longrightarrow \quad \frac{\rho_{\text{drysand}}}{\rho_{\text{quartz}}} = \frac{v_{\text{quartz}}}{v_{\text{drysand}}} \,. \tag{298}$$

The porosity (ϕ) is defined as the ratio of the void volume to the total volume of the sand, i.e.,

$$\phi = \frac{v_{\text{air}}}{v_{\text{drysand}}} = 1 - \frac{v_{\text{quartz}}}{v_{\text{drysand}}} = 1 - \frac{\rho_{\text{drysand}}}{\rho_{\text{quartz}}} \,. \tag{299}$$

For partially saturated sand, the SHPB tests provide the ratio

$$\alpha = \frac{m_{\text{water}}}{m_{\text{drysand}}} = \frac{\rho_{\text{water}} v_{\text{water}}}{\rho_{\text{drysand}} v_{\text{drysand}}} \,. \tag{300}$$

The saturation is defined as

$$S_w = \frac{v_{\text{water}}}{v_{\text{air}} + v_{\text{water}}} \tag{301}$$

and the porosity of the wet soil is

$$\phi = \frac{v_{\text{air}} + v_{\text{water}}}{v_{\text{wetsand}}} \,. \tag{302}$$

Therefore, if the volume of the sand does not change due to the introduction of water,

$$S_w = \frac{1}{\phi} \frac{v_{\text{water}}}{v_{\text{wetsand}}} = \frac{1}{\phi} \frac{v_{\text{water}}}{v_{\text{drysand}}} \,. \tag{303}$$

Plugging the above relation into the expression for α gives us

$$S_w = \frac{\alpha}{\phi} \frac{\rho_{\text{drysand}}}{\rho_{\text{water}}} \,. \tag{304}$$

The mass of the wet sand is

$$m_{\text{wetsand}} = m_{\text{drysand}} + m_{\text{water}} \quad \Longrightarrow \quad \rho_{\text{wetsand}} v_{\text{wetsand}} = \rho_{\text{drysand}} v_{\text{drysand}} + \rho_{\text{water}} v_{\text{water}} \,. \tag{305}$$

Since $v_{\text{wetsand}} = v_{\text{drysand}}$, we have

$$\rho_{\text{wetsand}} = \rho_{\text{drysand}} + \rho_{\text{water}} \frac{v_{\text{water}}}{v_{\text{drysand}}} = \rho_{\text{drysand}} + \phi S_w \rho_{\text{water}} \,. \tag{306}$$

C Weibull parameter variability

The probability density function of random variable x that is Weibull distributed can be expressed as

$$\mathrm{We}(x) = \beta\alpha x^{\alpha-1}\exp(-\beta x^{\alpha}) \quad \text{for } x \geq 0\,. \tag{307}$$

The expression used in the C++11 standard implementation is

$$\mathrm{We}(x) = \frac{a}{b}\left(\frac{x}{b}\right)^{a-1}\exp\left[-\left(\frac{x}{b}\right)^{a}\right]. \tag{308}$$

The relationship between these two expressions is

$$\alpha \equiv a \quad \text{and} \quad \beta \equiv \frac{1}{b^a} \implies b = \beta^{-1/\alpha}\,. \tag{309}$$

The shape parameter is $a = \alpha > 0$ and the scale parameter is $b > 0$. The shape parameter a is also called the **Weibull modulus** in the content of material strength distribution.

The mean of the distribution is

$$\mathbb{E}(x) = b\Gamma\left(1+\frac{1}{a}\right) = \beta^{-1/\alpha}\Gamma\left(1+\frac{1}{\alpha}\right) \tag{310}$$

where Γ is the gamma function. If we assume that the expected value is better represented by the median, we have

$$\mathbb{E}(x) = b\left[\ln(2)\right]^{1/a}\,. \tag{311}$$

The generate the Weibull distribution for a random variable, we typically use a transformation from a uniformly distributed random variable. To find the transformation between two probability distributions $f(y)$ and $g(x)$, we use the fundamental relation

$$f(y) = g(x)\left|\frac{dx}{dy}\right| \tag{312}$$

where the absolute value of the Jacobian of the transformation is used to make sure that probabilities sum to 1. For the special case where the distribution $g(x), x \in \mathcal{U} \sim [0,1]$ is uniform, we have

$$f(y) = \left|\frac{dx}{dy}\right|\,. \tag{313}$$

Therefore,

$$x = \int_0^y f(z)\,dz\,. \tag{314}$$

For the Weibull distribution, the right hand side is the cumulative distribution function,

$$x = \int_0^y \mathrm{We}(z)\,dz = \int_0^y \beta\alpha z^{\alpha-1}\exp(-\beta z^{\alpha})\,dz = 1-\exp(-\beta y^{\alpha}) = 1-\exp\left[-\left(\frac{y}{b}\right)^{a}\right]. \tag{315}$$

This relation can be inverted to give the transformed uniformly distributed random number between 0 and 1:

$$y = \left[-\frac{1}{\beta}\ln(1-x)\right]^{1/\alpha} = b\left[-\ln(1-x)\right]^{1/a}\,. \tag{316}$$

For a random variable that has the mean $\mathbb{E}(y) \approx \bar{y}$, from (310), the scale parameter is

$$b = \frac{\mathbb{E}(x)}{\Gamma\left(1+\frac{1}{a}\right)} \approx \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)}\,. \tag{317}$$

Therefore, the Weibull-transformed uniformly distributed random variable can be written as

$$y = \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)}\left[-\ln(1-x)\right]^{1/a} . \tag{318}$$

At this stage one typically invokes the fact that if x is uniformly distributed then so is $1-x$ and we can simplify the computation by using

$$y = \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)}\left[-\ln(x)\right]^{1/a} . \tag{319}$$

Alternatively, we can assume that the sample median is a better approximation of the expected value and use equation (311) to compute the scale parameter:

$$b = \frac{\bar{y}}{\left[\ln(2)\right]^{1/a}} . \tag{320}$$

In that case we have

$$y = \bar{y}\left[-\frac{\ln(x)}{\ln(2)}\right]^{1/a} = \bar{y}\left[\frac{\ln(x)}{\ln(1/2)}\right]^{1/a} . \tag{321}$$

The existing implementation of the Weibull generator in Uintah uses the following approach. A uniformly distributed random number x is generated. This number is used to compute the quantity

$$F = \left[-\ln(x)\right]^{1/a} \tag{322}$$

where a is the Weibull modulus. Two other quantities are computed:

$$C = \left[\frac{v_{\text{expt}}}{v_{\text{elem}}}\right]^{1/m} \quad \text{and} \quad \eta = \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)} \tag{323}$$

where v_{expt} is a reference volume, v_{elem} is the particle volume, m is an exponent, and $\bar{y}$ is the mean value of the parameter (y) that is Weibull distributed. The value of y is computed using the product of F, C, and η, giving

$$y = \left[\frac{v_{\text{expt}}}{v_{\text{elem}}}\right]^{1/m} \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)}\left[-\ln(x)\right]^{1/a} \tag{324}$$

The code typically uses $m = a$ to get

$$y = \frac{\bar{y}}{\Gamma\left(1+\frac{1}{a}\right)}\left[-\frac{v_{\text{expt}}}{v_{\text{elem}}}\ln(x)\right]^{1/a} . \tag{325}$$

This expression is identical to equation (319) except for a size-effect factor.

D Verifying stress paths for uniaxial strain loading

Most of the simulations discussed in this report have been driven by uniaxial strain. We can check that the code is doing the right thing by comparing the slope of the loading path in p-q space for elastic states. This appendix discusses what we should expect for linearly elastic materials.

From linear elasticity,

$$\boldsymbol{\sigma} = \lambda \mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon} \tag{326}$$

where

$$\lambda := K - \tfrac{2}{3}G \quad \text{and} \quad \mu := G \implies K = \lambda + \tfrac{2}{3}\mu \tag{327}$$

and K, G are the bulk and shear modulus, respectively. Therefore,

$$\begin{aligned} p &:= \tfrac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}) = \left(\lambda + \tfrac{2}{3}\mu\right)\mathrm{tr}(\boldsymbol{\varepsilon}) \\ \boldsymbol{s} &:= \boldsymbol{\sigma} - p\boldsymbol{I} = \lambda\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon} - \left(\lambda + \tfrac{2}{3}\mu\right)\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I} = 2\mu\left[\boldsymbol{\varepsilon} - \tfrac{1}{3}\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I}\right]. \end{aligned} \tag{328}$$

From the second equation above, we can compute

$$q := \sqrt{3J_2} = \sqrt{\tfrac{3}{2}\boldsymbol{s}:\boldsymbol{s}} = \sqrt{\tfrac{3}{2}(2\mu)^2\left[\boldsymbol{\varepsilon}:\boldsymbol{\varepsilon} - \tfrac{1}{3}[\mathrm{tr}(\boldsymbol{\varepsilon})]^2\right]} = 2\mu\sqrt{\tfrac{3}{2}\left[\boldsymbol{\varepsilon}:\boldsymbol{\varepsilon} - \tfrac{1}{3}[\mathrm{tr}(\boldsymbol{\varepsilon})]^2\right]}. \tag{329}$$

For uniaxial strain in the 1-direction, $\mathrm{tr}(\boldsymbol{\varepsilon}) = \varepsilon_{11}$ and $\boldsymbol{\varepsilon}:\boldsymbol{\varepsilon} = \varepsilon_{11}^2$, and the above equations for p and q become

$$p = \left(\lambda + \tfrac{2}{3}\mu\right)\varepsilon_{11} = K\varepsilon_{11} \quad \text{and} \quad q = 2G\varepsilon_{11}. \tag{330}$$

Therefore, for uniaxial strain linear elastic deformations starting from zero strain, the slope of the loading path is

$$\frac{q}{p} = \frac{2G}{K} = \frac{2}{K}\frac{3K(1-2\nu)}{2(1+\nu)} = \frac{3(1-2\nu)}{1+\nu}. \tag{331}$$

For $\nu = 0.35$, the slope of the loading path is $\frac{2}{3}$ which can be used as a check of the algorithm.

E Parameter confidence intervals and Hessians

For model parameters that are independent and identically distributed, a rigorous way of determining parameters of the distributions that describe the variability of the material has been elusive. In this section, we describe one approach for finding the parameters of distributions.

Consider a linear Hooke's law constitutive model for which the user-input parameters are the bulk modulus (K) and the shear modulus (G). The input is the strain tensor ($\boldsymbol{\varepsilon}$) while the output is the stress tensor ($\boldsymbol{\sigma}$). If $\boldsymbol{I}$ is the identity tensor, we can define functions that give isotropic and deviatoric parts of a tensor

$$\text{iso}(\boldsymbol{A}) = \tfrac{1}{3}\text{tr}(\boldsymbol{A})\,\boldsymbol{I} \quad \text{and} \quad \text{dev}(\boldsymbol{A}) = \boldsymbol{A} - \text{iso}(\boldsymbol{A})\,. \tag{332}$$

Then Hooke's law can be defined as

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}, K, G) = 3K\,\text{iso}(\boldsymbol{\varepsilon}) + 2G\,\text{dev}(\boldsymbol{\varepsilon})\,. \tag{333}$$

Consider the special case of Hooke s law under axisymmetric loading, where ε_A is the axial component of strain, ε_L is the lateral component of strain, σ_A is the axial component of stress, and σ_L is the lateral component of stress. All other components of stress and strain are zero. The reduced Hooke's law for this special case of axisymmetric loading is

$$\begin{bmatrix}\sigma_A\\ \sigma_L\\ \sigma_L\end{bmatrix} = \begin{bmatrix}\frac{4}{3}G(\varepsilon_A-\varepsilon_L)+K(\varepsilon_A+2\varepsilon_L)\\ -\frac{2}{3}G(\varepsilon_A-\varepsilon_L)+K(\varepsilon_A+2\varepsilon_L)\\ -\frac{2}{3}G(\varepsilon_A-\varepsilon_L)+K(\varepsilon_A+2\varepsilon_L)\end{bmatrix}. \tag{334}$$

Further reducing these equations to the case of uniaxial stress (where $\sigma_L = 0$) gives

$$\sigma_A = \frac{9GK\varepsilon_A}{G+3K} \quad \text{and} \quad \varepsilon_L = \frac{(2G-3K)\varepsilon_A}{2(G+3K)}\,. \tag{335}$$

Consider uniaxial stress data for which the axial strain (ε_A) is controlled. The data set is considered to be complete if the measured data are axial stress (σ_A) and lateral strain (ε_L). Conversely, the data set is incomplete if one of the above two data sets is missing. For illustration, we will address the unfortunately common case in which experimentalists provide modelers with only axial stress-strain data, while failing to measure the lateral strain data.

The Young's modulus (E) and Poisson's ratio (ν) are defined as

$$E = \frac{\sigma_A}{\varepsilon_A} = \frac{9GK}{G+3K} \quad \text{and} \quad \nu = -\frac{\varepsilon_L}{\varepsilon_A} = -\frac{(2G-3K)}{2(G+3K)}\,. \tag{336}$$

To generate the exact data, we will used bulk and shear moduli for aluminum. These are $K_{\text{exact}} = 78$ GPa and $G_{\text{exact}} = 26$ GPa which imply that $E_{\text{exact}} = 70.2$ GPa and $\nu_{\text{exact}} = 7/20$.

We generate some synthetic experimental data for uniaxial stress loading for axial strain values of $\varepsilon_A = [1, 9, 14, 23, 28, 31, 35, 41, 45, 51]\times 10^{-5}$. The corresponding values of σ_A and ε_L are taken to be mean values. A normal distribution is used to generate variability in the synthetic data using a standard deviation of 650 kPa for stress and 0.0000033 for the lateral strain. The goal is to find values of the model parameters K and G that fit the synthetic data.

Let us define error measures as

$$\delta\sigma_A = \frac{\sigma_A}{\sigma_A^{\text{exact}}} - 1 \quad \text{and} \quad \delta\varepsilon_L = \frac{\varepsilon_L}{\varepsilon_L^{\text{exact}}} - 1 \tag{337}$$

where

$$\sigma_A^{\text{exact}} = \frac{9G_{\text{exact}}K_{\text{exact}}\varepsilon_A}{G_{\text{exact}}+3K_{\text{exact}}} \quad \text{and} \quad \varepsilon_L^{\text{exact}} = \frac{(2G_{\text{exact}}-3K_{\text{exact}})\varepsilon_A}{2(G_{\text{exact}}+3K_{\text{exact}})}\,. \tag{338}$$

Then the optimization problem to find the fitted values of K and G can be framed as as least-squares minimization problem where the objective function is

$$f := \frac{1}{n}\sum_{n}\left[(\delta\sigma_A)^2 + (\delta\varepsilon_L)^2\right] \tag{339}$$

where n is the number of observations.

In the following, the phrase "complete data" refers to having enough experimental data to find a unique best-fit set of values for K and G. We can use the exact data as a verification test to demonstrate that the optimization procedure does indeed give exact fits for K and G when the data set is complete (i.e., includes both axial stress data and lateral strain data) and when the data exactly fits the Hooke's law constitutive model.

Rather than optimizing in a parameter space (K, G), we will seek optimal values of dK and dG defined such that

$$K = K_{\text{fac}}(1 + dK) \quad \text{and} \quad G = G_{\text{fac}}(1 + dG) \tag{340}$$

Normally, if the exact values of K and G are not known, K_{fac} and G_{fac} are set to be numerals that are in the same order of magnitude as the expected exact solution. If, for example, the answer is expected to be on the order of 10^9 (i.e., GPa, as is typical for elastic moduli), then one would set $K_{\text{fac}} = G_{\text{fac}} = 10^9$ and then optimize on the dimensionless and much smaller dK and dG. In this case where we happen to know the exact solutions, let us set $K_{\text{fac}} = K_{\text{exact}}$ and $G_{\text{fac}} = G_{\text{exact}}$. This way, we know the exact optimal solution is $dK = 0$, $dG = 0$.

Let us define the Hessian matrix at the solution $(dK_{\text{fit}}, dG_{\text{fit}})$ as

$$\boldsymbol{H} := \begin{bmatrix} \frac{\partial^2 f}{\partial(dK)^2} & \frac{\partial^2 f}{\partial(dK)\partial(dG)} \\ \frac{\partial^2 f}{\partial(dG)\partial(dK)} & \frac{\partial^2 f}{\partial(dG)^2} \end{bmatrix}. \tag{341}$$

For the exact synthetic data, the Hessian matrix is

$$\boldsymbol{H} = \begin{bmatrix} 0.317551 & -0.117551 \\ -0.117551 & 1.91755 \end{bmatrix}. \tag{342}$$

The eigenvalues and eigenvectors (in columns) of $\boldsymbol{H}$ are

$$\lambda := \begin{bmatrix} 1.92614, 0.308961 \end{bmatrix} \quad \text{and} \quad \mathbf{v} := \begin{bmatrix} -0.0728827 & -0.997341 \\ 0.997341 & -0.0728827 \end{bmatrix}. \tag{343}$$

If we compute a Taylor series expansion of the objective function at the solution, we can confirm from contours of the Taylor series of the error function coincide with the plots of the actual error function. We can also conform that the directions of the eigenvectors of the Hessian matrix are aligned with the principal axes of the ellipse that represents a sample contour from the Taylor series for the objective function centered at the solution for the minimum.

Unlike the exact data case, the overall error to be minimized at values of dK and dG that are not zero is not zero when there is variability between samples . Hence, an optimized solution will not exactly match the exact solution in general. For the variable synthetic data, the minimization process yields the solution dK_{fit} = -0.113977 and dG_{fit} = -0.00106647 with a minimum error of 0.102031. The corresponding Hessian matrix is

$$\boldsymbol{H} := \begin{bmatrix} 0.522477 & -0.23128 \\ -0.23128 & 1.84879 \end{bmatrix} \tag{344}$$

with eigenvalues and eigenvectors

$$\lambda := \begin{bmatrix} 1.88797 & 0.483304 \end{bmatrix} \quad \text{and} \quad \mathbf{v} := \begin{bmatrix} -0.166997 & -0.985957 \\ 0.985957 & -0.166997 \end{bmatrix}. \tag{345}$$

Figure 70 shows the contour plot of the objective function, where the thick black ellipse is a sample contour from the Taylor series expansion of the objective function centered at the solution for the minimum. The straight red lines show the directions of the eigenvectors of the Hessian matrix.

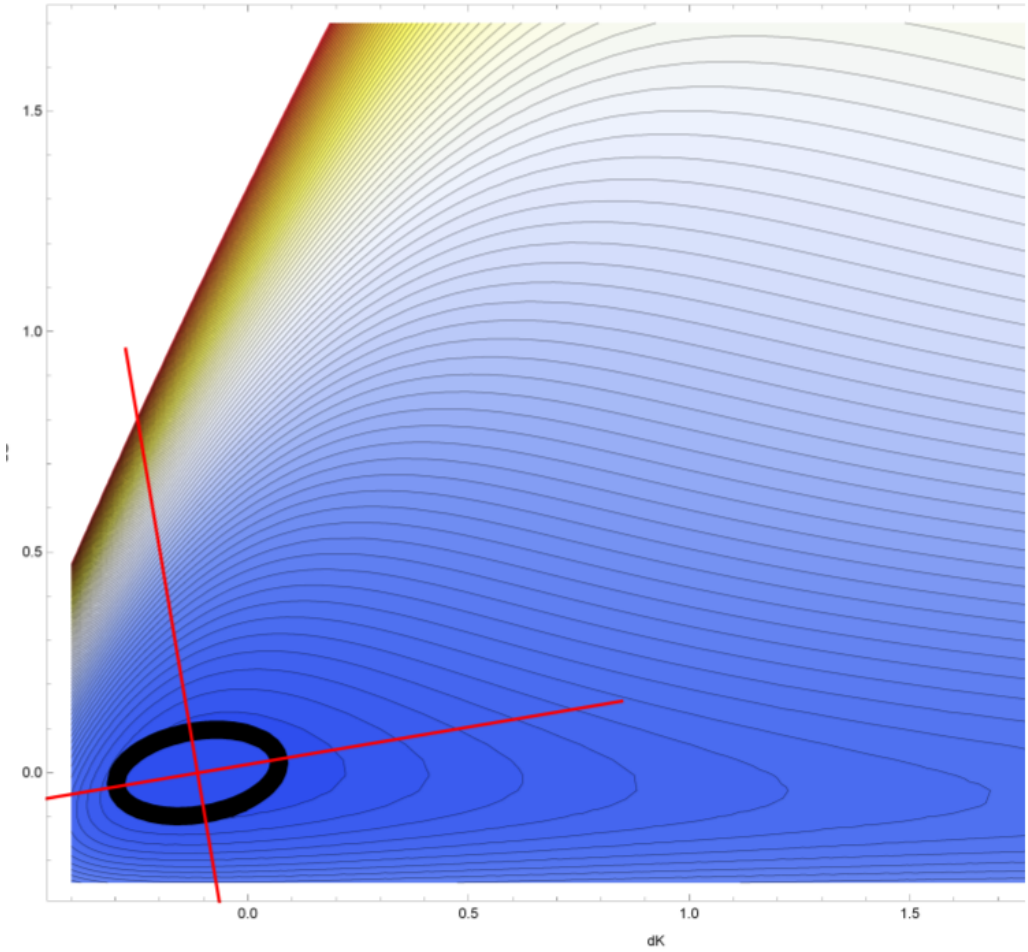


Figure 70 – *Objective function contours in dK-dG space showing a contour from the Taylor series around the optimum point.*

Now consider the situation where the data are incomplete (this is the situation that we observed for the Mason sand and Boulder clay data in this report). Let us assume that the only data available for fitting is the axial stress and strain data, while data for lateral strain are missing. Accordingly, an optimization will not be unique. Given only axial stress-strain data, the only thing that can be optimized is the fit to Young s modulus,

$$E = \frac{9GK}{3K + G} \,. \tag{346}$$

There are an infinite number of K and G values that can exactly match the slope of the axial stress-strain data. The optimizer will find only one acceptable K and G pair, and we need a way to determine the constraint for which other values are equally acceptable. When only the axial stress data are available, the overall error measure is taken to be the sum of squares of the axial stress errors:

$$f := \frac{1}{n} \sum_{n} \left[(\delta\sigma_A)^2 \right] \,. \tag{347}$$

If we try to fit the exact data, we find that the Hessian is

$$\boldsymbol{H} := \begin{bmatrix} 0.00191195 & 0.0612799 \\ 0.0612799 & 1.96408 \end{bmatrix} \tag{348}$$

with eigenvalues and eigenvectors

$$\lambda := \begin{bmatrix} 1.966 & 4.63095 \times 10^{-13} \end{bmatrix} \quad \text{and} \quad \mathbf{v} := \begin{bmatrix} 0.031185 & -0.999514 \\ 0.999514 & 0.031185 \end{bmatrix}. \tag{349}$$

As seen above, this time the Hessian matrix has a zero eigenvalue. That means that the provided data were insufficient to set values to all of the material properties. Thus, the optimized values (dK_{fit}, dG_{fit}) will be only one of an infinite number of equally good solutions. Because we are fitting only to axial stress-strain data, any bulk and shear modulus pair will minimize the error as long as that pair has the same Young's modulus (to optimally match the slope of the available data). The solution fits the incomplete data exactly.

For variable incomplete data, the above process also leads to a Hessian with a zero eigenvalue. Figure 71 shows that the contour plot of the objective function with the thick black ellipse is now a degenerate ellipse of infinite length in one direction. The slope of that ellipse is simply the plot of the constraint,

$$\frac{9GK}{3K+G} = E_{\text{exact}} \tag{350}$$

being a sample contour from the Taylor series for the objective function centered at the solution for the minimum. The red lines show the directions of the eigenvectors of the Hessian matrix. Because the data are incomplete, the ellipse direction associated with the zero eigenvalue of the Hessian matrix is infinitely long. That means any K and G value on that line will give equally accurate answers. The thick green line is the plot of

$$\frac{9GK}{3K+G} = \text{constant} = E_{\text{fit}}\,. \tag{351}$$

Note that the solution fits the incomplete data well, though not exactly.

These exercises using Hooke's law would be important to run when testing out any optimization soft ware. Of crucial importance is the situation for which there is not yet enough data to fit the model. That means that an optimized fit to the available data is actually not unique, and the eigenvectors of the Hessian matrix corresponding to zero eigenvalues tell us admissible directions in which the parameters may be tweaked without disrupting our previous fits. In this Hooke's law example, the fit to incomplete data allows tweaking the model parameters, K and G, in any direction for which Young's modulus does not change.

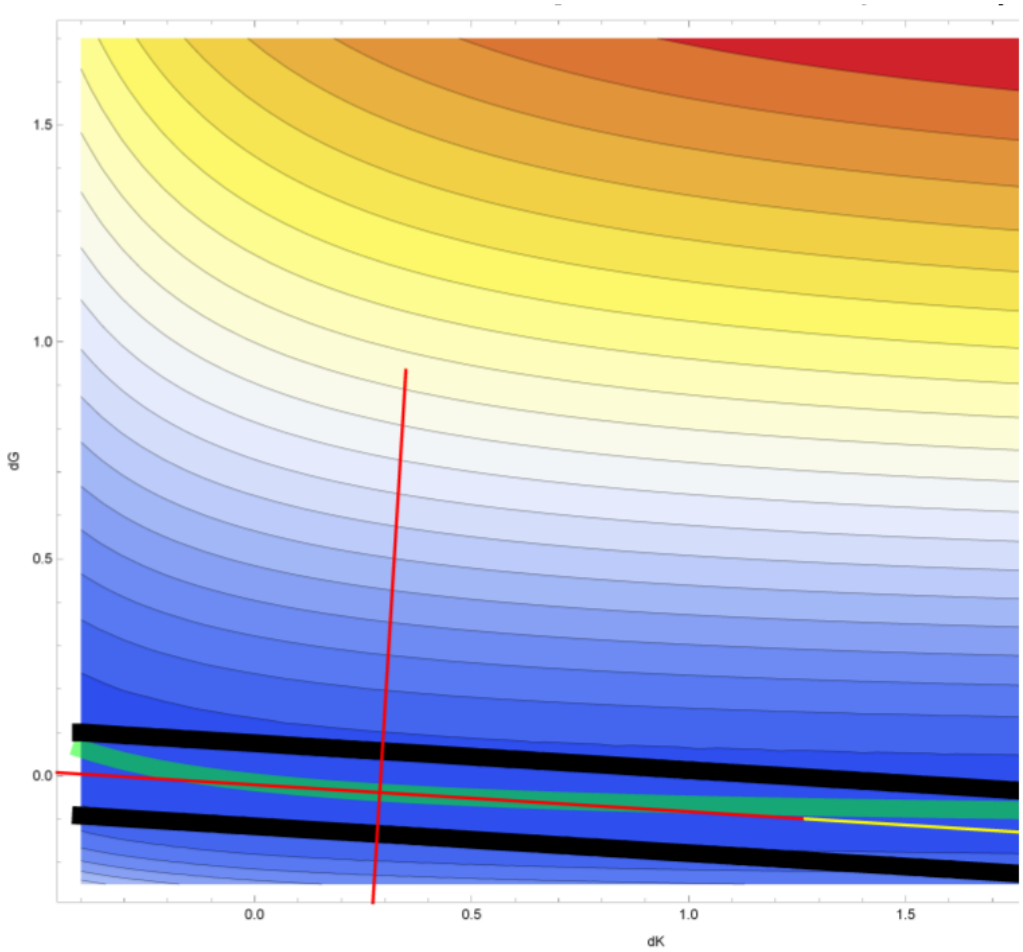


Figure 71 – *Objective function contours in dK-dG space showing a contour from the Taylor series around the optimum point when only the axial stress-axial strain data are available.*