Hierarchical Multiscale Modeling to Inform Continuum Constitutive Models of Soils

by

Erik Jensen

B.A., University of Massachusetts Amherst, 2010

M.S., University of Colorado Boulder, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Civil, Environmental, and Architectural Engineering

2017

This thesis entitled:
Hierarchical Multiscale Modeling to Inform Continuum Constitutive Models of Soils
written by Erik Jensen
has been approved for the Department of Civil, Environmental, and Architectural Engineering

Richard Regueiro

Ronald Pak

Jeong-Hoon Song

Rebecca Brannon

Hongbing Lu

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Jensen, Erik (Ph.D., Geotechnical Engineering)

Hierarchical Multiscale Modeling to Inform Continuum Constitutive Models of Soils

Thesis directed by Associate Professor Richard Regueiro

The behavior of soils is fundamentally the combined interactions between billions and billions of individual particles and particle matrices. For computational tractability and despite this fact, many soil models assume the material is a continuum essentially averaging the inter-particle interactions to predict the behavior of the bulk material. In recent years, multiscale modeling techniques have been developed to reintroduce the effect of the grain-scale interactions to help model situations where the continuum assumption fails, such as shear band development in triaxial tests or soil disaggregation under blast loading. One of such multiscale models called hierarchical upscaling, or global-local analysis, model the problem domain using typical continuum-based approaches, but replace the continuum constitutive model with grain-scale models that represent the microstructure at the given point. The deformation of the material is solved via the continuum-scale boundary value problem, which is then passed to the grain-scale as boundary conditions on these representative volume elements (RVEs). The RVEs are then allowed to deform, and the resulting stress is passed back to the continuum scale.

Three hierarchical multiscale models were developed to model a series of experimental results on dry Colorado Mason Sand. In all three models, the grain-scale was modeled using the discrete element method through a code called ellip3D. Ellip3D models of dry Colorado Mason Sand were integrated with three continuum models: a one dimensional, finite strain finite element model with full dynamics (1D FEM-DEM model), a one dimensional material point method implementation (1D MPM-DEM model), and a two dimensional, small strain, axisymmetric finite element model (2Daxi FEM-DEM model). The one dimensional models were used to model a split Hopkinson pressure bar test and the two dimensional model was used to model two triaxial tests. With the inclusion of a particle fracture model, the results of the split Hopkinson pressure bar experiment were reasonably well replicated; however, the 2Daxi FEM-DEM model over-predicts the stress response of the triaxial tests. Regardless, to the author's knowledge, the results herein presented are the first attempt to directly compare FEM-DEM global-local analysis models to experimental data. Also, the MPM-DEM implementation is a novel extension of other FEM-DEM models to a meshfree numerical methods.

Dedication

To my wife, Jillian, for graciously agreeing to leave behind family, friends, and lifelong home to follow me to Colorado. These past seven years have been amazing. And to my daughter, Lily, for always smiling.

Acknowledgements

I would like to gratefully acknowledge the U.S. Department of Defense Office of Naval Research for funding this research through MURI grant N00014-11-1-0691, and the Department of Civil, Environmental, and Architectural Engineering at the University of Colorado Boulder for partially supporting this work through a Doctoral Assistantship for Completion of Dissertation.

There have been many people who have provided their expertise, help, and support to this research. Boning Zhang provided countless hours of work and troubleshooting support regarding ellip3D. Chris Kung at the Department of Defense High Performance Computing (HPC) Productivity Enhancement, Technology, Transfer, and Training (PETTT) program rewrote my MatLab code in C++ and taught me how to parallelize the new code for use on the Department of Defense's supercomputers saving me hundreds of hours of work. I also received excellent suggestions and advice from my committee members, group members, and the many researchers working on the Soil Blast MURI Project. All of their input is greatly appreciated.

Finally, I wouldn't be writing this if it weren't for my advisor, Richard Regueiro. Thank you for this opportunity, your continued support and advise, and convincing me to stay at CU for my doctorate. It's difficult to put into words what the past seven years have meant to me, so instead, I'll look forward and hope that this is only the beginning of a long research partnership and lasting friendship.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

Funded by an Office of Naval Research (ONR), United States Department of Defense, Multidisciplinary University Research Initiative (MURI) grant, the primary goal of the Soil Blast MURI Project was to use state-of-the-art multiphase and multiscale computational modeling tools to simulate partially saturated soil failure under buried explosive loading. Concurrent high-strain-rate experimentation using split Hopkinson pressure bars, shock tubes, and modified triaxial cells conducted on a variety of soils as well as geotechnical centrifuge blast experiments were conducted to be used for parameter calibration and potential validation of the numerical model. In addition, the soils used for the experimentation were a sand and clay extracted from the ground rather than the more homogeneous "laboratory" soils typically used in geotechnical research in an effort to capture the behavior of a "nature" soil or soil mixture.

As depicted on figure 1.1, the goal of the Soil Blast MURI Project was to develop a two-scale numerical model. The mesoscale (sand-grain-scale) was to be a combination of four coupled modeling methods: discrete element method (DEM) with added grain crushing for the sand particles, peridynamics (PD) to model clay deformation and fracture, smoothed particle hydrodynamics (SPH) for the free liquid water, and computational fluid dynamics (CFD) for the background air. This complicated mesoscale model was coupled to a macroscale continuum-based material point method (MPM) developed in Uintah (`http://uintah.utah.edu/`) using a hierarchical upscaling procedure. Within the MPM model, a new high-strain-rate, elasto-plastic soil constitutive model (Arena) developed for the project was implemented. Arena also includes a void disaggregation

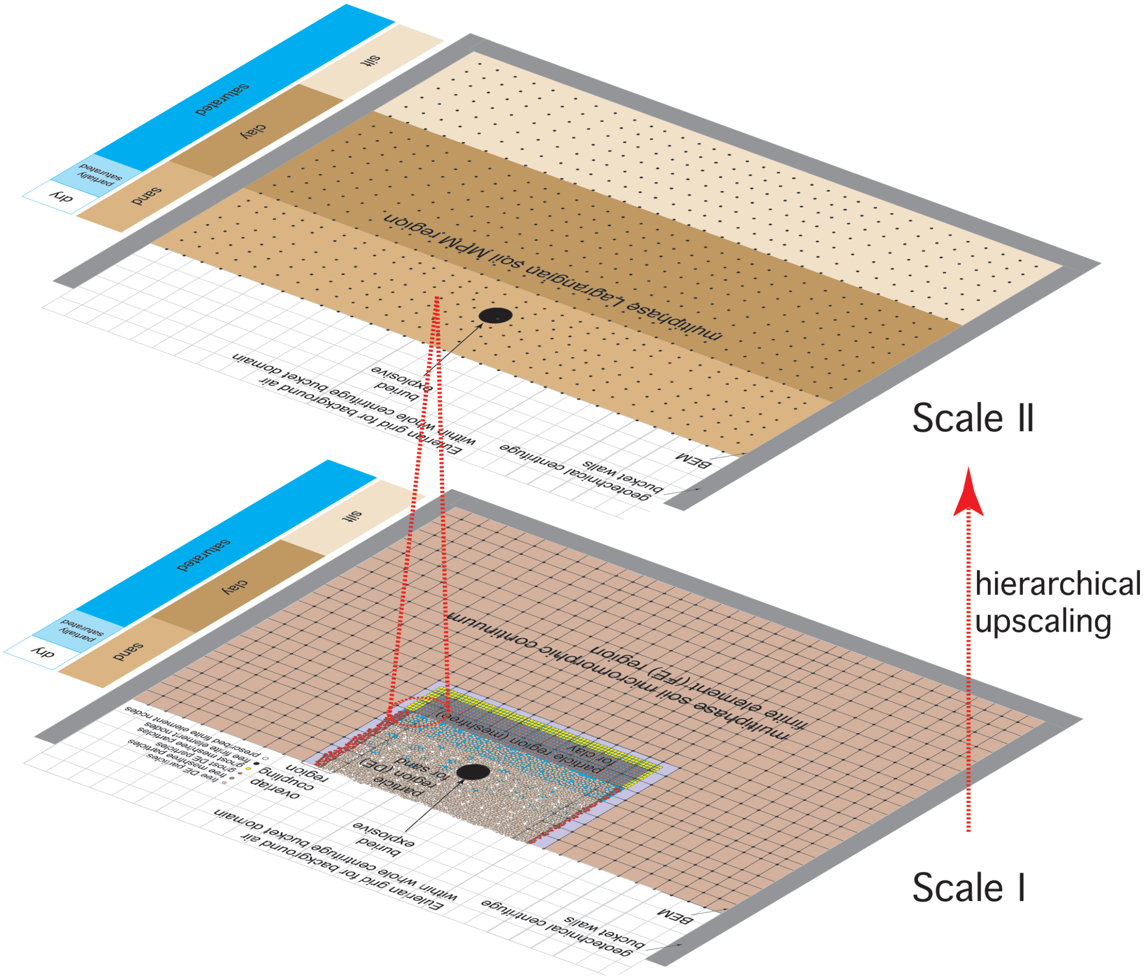

dry
partially saturated
saturated
sand
clay
silt
multiphase Lagrangian soil MPM region
buried explosive
Eulerian grid for background air within whole centrifuge bucket domain
BEM
geotechnical centrifuge bucket walls
Scale II
hierarchical upscaling
multiphase soil micromorphic continuum finite element (FE) region
particle region (meshfree) for clay
particle (DE) region for sand
overlap coupling region
Scale I

Figure 1.1: General Interaction Concept Between Scale I (Sand-grain-scale) and Scale II (Macro Continuum) Depicted Through the Cross-section of a Centrifuge Bucket Buried Explosive in Soil Experimental Set-up

algorithm, the effect of temperature changes on the material properties, and Duvaut-Lions viscoplasticity (Brannon et al., 2014).

At the conclusion of the project, many of the original goals were met. Both the sand and clay were exhaustively tested creating a database of experimental results for the soils under a wide array of loading paths, initial conditions, drainage conditions, and strain rates. Characterization of the soil structure at micrometer resolution was done using synchrotron microcomputed tomography (SMT) (Druckrey and Alshibli, 2014), and hundreds of geotechnical centrifuge blast experiments were run to directly model field-scale explosions in the soils (through scaling relationships). On the numerical modeling side, the macroscale constitutive model (Arena) implemented in the MPM framework of Uintah was completed and is available for use, and all of the individual components of the fully coupled sand-grain-scale model (DEM, PD, SPH, and CFD) were developed with testing ongoing. The DEM and CFD codes were fully coupled and extensively tested, and initial coupling of the DEM-PD and DEM-SPH models was completed. Using a subset of the experimental results, simplified versions of the macroscale constitutive model, and the mesoscale modeling efforts, this work focuses on the hierarchical multiscale modeling algorithms that were used for part of the modeling component of the overall project. Although a fully coupled, two-scale numerical model connecting the full sand-grain-scale and macroscale was not completed, the work outlined in this report presents a clear path towards connecting the two scales using hierarchical multiscale modeling.

Chapter 2 of this report contains a literature review outlining the history of hierarchical multiscale model in granular materials beginning with mathematical asymptotic homogenization theory and ending with modern global-local homogenization methods. Chapter 3 describes a simplified version of the Arena soil model that was developed in an ABAQUS user subroutine. The simplified model was calibrated for the sand and clay against data from the experimental effort of the Soil Blast MURI Project. Chapter 4 details the development of a one dimensional, uniaxial finite strain hierarchical multiscale model that was used to model experimental results from a split Hopkinson pressure bar experiment on a dry specimen of the sand. Chapter 5 extends the one dimensional model to use the material point method in place of the original FEM model, and chapter 6 extends the FEM-DEM model of chapter 4 to a two dimensional, axisymmetric hierarchical multiscale model at small strain. Chapter 7 discusses current conclusions of the research as well as potential future additions and improvements.

Chapter 2

Literature Review

Multiscale modeling is a broad term describing all modeling efforts where the behavior of a material is viewed through more than one length or time scale. Often in the case of undergraduate level mechanics, materials are assumed to be homogeneous blocks and their behavior described by simplistic constitutive models. In reality, even materials for which the homogeneous assumption is good have heterogeneities if one zooms in close enough. In fact, it has been demonstrated rigorously that when considering the heterogeneities that cause localized fracture in ceramics, a single continuum finite element would need to be kilometers in size for the element to be considered homogeneous (Huq et al., 2015). For other materials, such as geomaterials, the heterogeneities are much larger, more prominent, and can have a significant impact on the aggregate material's behavior. Multiscale modeling of geomaterials and, in particular, soils means recognizing that the behavior of a soil continuum is actually the combined behavior of billions of particle-to-particle and particle-to-matrix interactions often combined with the presence of liquid water, and in certain circumstances, the behavior of a subset of these contacts can have profound influence on the overall behavior of the soil. For example, even in the simplest of geotechnical engineering tests such as unconfined compression and dry triaxial compression testing, strain localizations through the formation of shear bands dominate the behavior of the soil at failure. Also under explosive loading, the loss of inter-particle contacts as the soil particles disaggregate is the grain-scale phenomenon that determines the macroscale behavior of the soil. Hierarchical multiscale modeling techniques can be used to study these types of phenomena without the need to make any a priori assumptions about where these localizations will occur.

Discussions of hierarchical multiscale modeling in geomaterials generally begin with an overview of mathematical asymptotic homogenization theory. Primarily developed in the 1970s and 1980s with the goal of analyzing composite materials for use in aircraft, as structural elements, and even understanding human tissue, the work defines a class of problems for which the macroscale behavior is analyzed through studying a representative section of the material's microstructure. These spatially repetitive or periodic subsections were coined as representative volume elements (RVEs) by Hill (1963). In homogenization theory, the stress and displacement fields are expanded asymptotically about their macroscopic values in order to derive boundary value problems for both scales. Variational principles are used to link the two scales together. Many books documenting the full mathematical framework and numerous examples of mathematical homogenization theory have been written, namely Bensoussan et al. (1978), Sanchez-Palencia (1980), Nemat-Nasser and Hori (1993), but the general idea is summarized below using the notation of Ghosh et al. (1996), which is based on the work by Bensoussan et al. (1978), DeVries et al. (1989), and Guedes and Kikuchi (1990).

Considering a body Ω with external tractions ($\boldsymbol{t}$), applied displacements ($\boldsymbol{g}$), and body forces ($\boldsymbol{b}$), the characteristic length of an RVE representing the microstructure of the body is assumed to be much smaller than the characteristic length of the overall body (see figure 2.1). The field variables, namely stress and deformation, are taken to not only vary over each macroscopic point ($\boldsymbol{x}$), but also in the nearby neighborhood of $\boldsymbol{x}$. One can enlarge the microstructure at $\boldsymbol{x}$ through a scaling parameter (ϵ) to introduce two coordinate systems related through ϵ. This allows a modeler to formulate multiple balance equations that are coupled through an integral averaging operator but solved separately. The field variables are then functions of both $\boldsymbol{x}$ and $\boldsymbol{y} = {}^{\boldsymbol{x}}/_{\epsilon}$. Also, the microstructure is assumed to be a matrix of periodic RVEs at each $\boldsymbol{x}$, so $\boldsymbol{y}$ is assumed to be a periodic function (Ghosh et al., 1996). As explained by Guedes and Kikuchi (1990), the material properties are then periodic functions of ϵ, and this periodicity is small as compared with the relative size of the macrostructure, which enables the computation of equivalent properties as the cell size

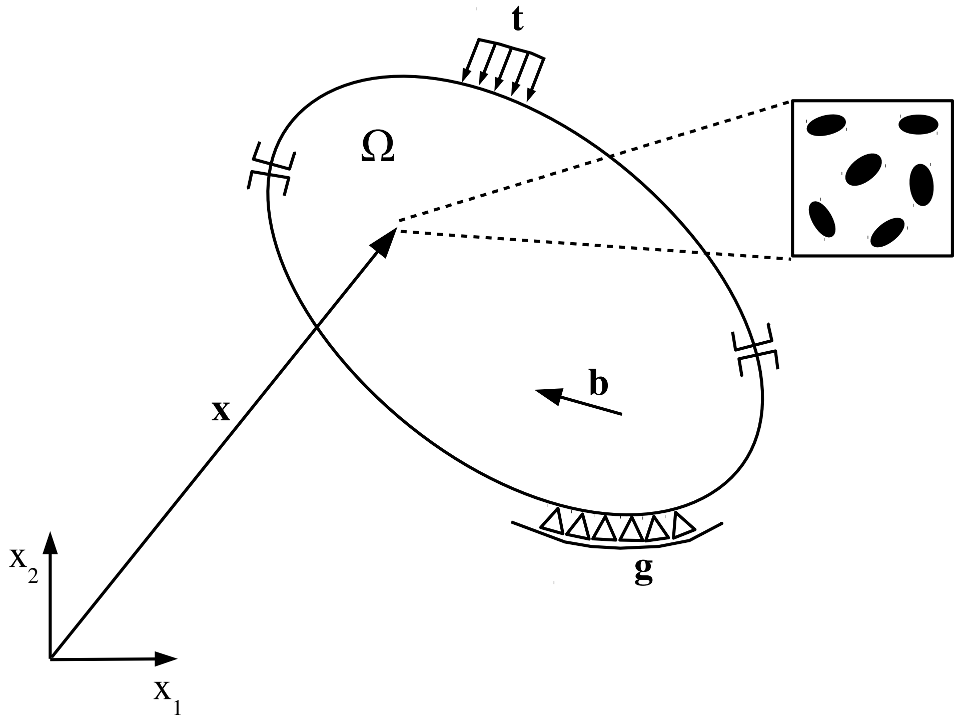


Figure 2.1: Geometry for Mathematical Asymptotic Homogenization Discussion

is reduced to zero. Only the microscale RVEs are considered periodic, not the entire macroscale domain. Different macro-points $\boldsymbol{x}$ can contain different microstructures, but the microstructures at each macro-point are periodic as seen in figure 2.2. This is called "local periodicity."

Before continuing, it is instructive to discuss the concept of a representative volume element more deeply. As aforementioned, the term was coined by Hill in 1963, and he defined a "representative volume" as one that "(a) is structurally entirely typical of the whole mixture on average, and (b) contains a sufficient number of inclusions for the apparent overall moduli to be effectively independent of the surface values of traction and displacement." In his paper, he was studying the elastic behavior of solid materials containing heterogeneous inclusions within the material, so the selected volume should be large enough to not only "look like", but also "behave like" the material it's representing. Over the years since, other researchers have developed working definitions for RVEs, such as Drugan and Willis (1996), who said, "[RVEs are] the smallest material volume

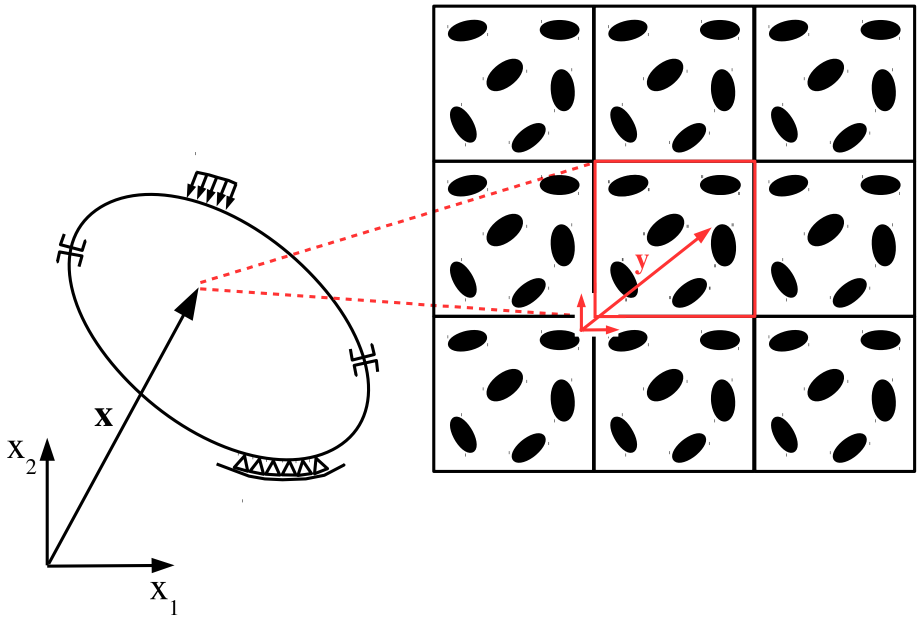


Figure 2.2: Local Periodicity

element of the composite for which the usual spatially constant (overall modulus) macroscopic constitutive representation is a sufficiently accurate model to represent mean constitutive response." Nearly all definitions one can find use similar language to create descriptions of what constitutes an RVE.

In practice, an RVE is often found to be representative through the literal testing of different sized volumes of material. One expects the behavior of "too small" a sample to vary relative to other samples of the same size, so as the sample is taken to be larger and larger, the variability in behavior from sample to sample decreases. When the variability in constitutive response is negligible as the sample continues to get larger, a representative volume is taken to have been found. This procedure can be done both experimentally or numerically depending on the application (Clendennen and Romero, 2013). For example, Kaufman (2012) studied RVEs of discrete element method (DEM) particle assemblies numerically to find the smallest RVE that reliably reproduced triaxial test results. Finding the smallest possible RVE is of particular importance in DEM modeling as large particle assemblies can be computationally expensive to the point of intractability. Also, the discovery of an RVE helps justify the assumption of local periodicity. If the behavior of the aggregate material is modeled well by an RVE, a conglomerate of the same spatially repetitive RVE should also model the material well.

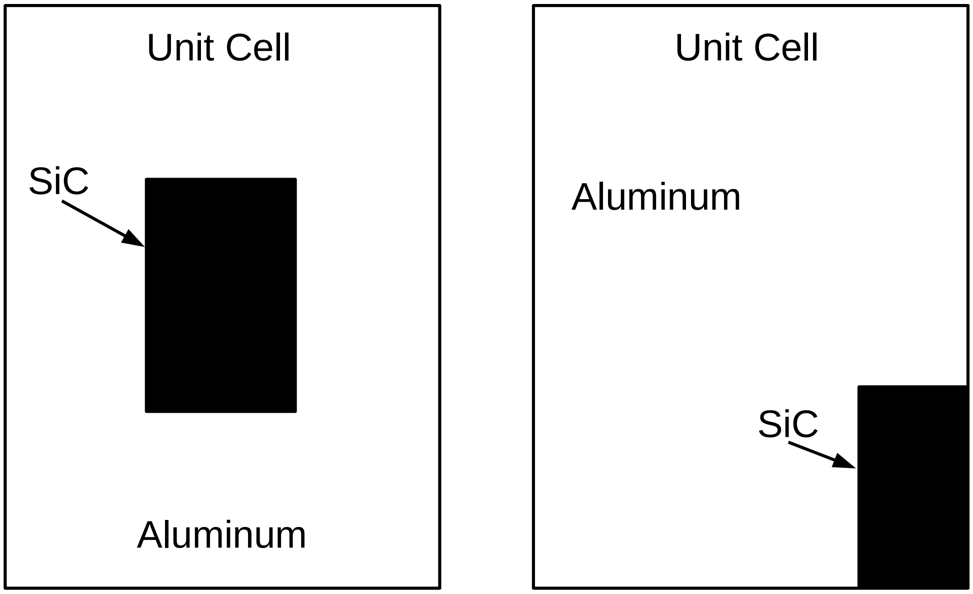
Homogenization theory is based on the asymptotic expansion of the field variables about ϵ. Asymptotic expansion is a general class of series expansions used to approximate more complicated functions. The famous Taylor series, for example, is a convergent asymptotic expansion. In the case of homogenization theory, we expand the field variable solutions around ϵ. At each macro-point $\boldsymbol{x}$, there exists a periodic microstructure such that the microstress ($\sigma_{ij}^{\text{micro}}(x_i, y_i)$) approximates the stress field over the period of the microstructure. Therefore,

$$\lim_{\epsilon \to 0} \left[\Sigma_{ij}(x_i) - \sigma_{ij}^{\text{micro}}(x_i, {}^{x_i}\!/\!_{\epsilon}) \right] = 0 \tag{2.1}$$

as ϵ goes to zero, which shows that $\sigma_{ij}^{\text{micro}}$ is an approximation of the macrostress (Σ_{ij}) when ϵ is small. Therefore, given an applied strain and constitutive assumptions of the RVE, an approximation of the stress solution $\boldsymbol{\sigma}^{\text{micro}}$ as a function of both $\boldsymbol{x}$ and $\boldsymbol{y}$ can be found from which the global Cauchy stress of the material can be derived (DeVries et al., 1989).

There are numerous examples from the engineering science, applied mathematics, and materials science literature of asymptotic homogenization theory being used to describe the behavior of materials. For example, Guedes and Kikuchi (1990) studied the behavior of linear elastic composite materials, specifically aluminum with boron fibers woven into the microstructure, for use in aircraft. They primarily focused on estimating the error and convergence rates of these types of multiscale simulations and methods to improve the method's accuracy via an adaptive finite element method. DeVries et al. (1989) and Fish et al. (1999) both extended the multiscale analysis to include microscale damage for use in layered composite materials and ceramics, respectively. Toledano and Murakami (1987) added dynamics and a basic mixture model to study particular composites (glass/epoxy and steel/PMMA composites). They compared their model to experiments studying harmonic wave propagation in material samples with good model agreement. Hollister and Kikuchi (1992) studied porous, linear elastic composite materials using both homogenization theory and a standard finite element model. They found that their finite element model converged to the homogenization theory solution as they refined the spatial discretization. Although these models provide local stress and strain fields and good homogenized material properties, they were originally restricted to linear elasticity or other simple constitutive equations (Kouznetsova, 2002).

Partially due to computation restrictions, unit cell multiscale models grew out of homogenization theory in the late 1980s. Rather than assuming only local periodicity of the microstructures, unit cell models assume global and local periodicity - i.e. the same RVE, called a "unit cell" in these cases, is assumed to represent the microstructure throughout the macrostructural domain. Focus can then be on solving one boundary value on the RVE to calculate the microstructural stress and displacement fields. Homogenization techniques can then be used to extrapolate those results to the global domain. In particular, many researchers focused on calculating homogenized material constants.



(a) Christman et al. (1989)

(b) Tvergaard (1990)

Figure 2.3: Comparison Between RVE Selection for Al-SiC Whisker Composite Unit Cell Models

Christman et al. (1989) and Tvergaard (1990) used similar unit cell models to simulate the behavior of Al-SiC whisker composites. Both works were focused on studying the particular failure mechanisms of the metal alloy, but differed in their RVE selection. Both used axisymmetric finite element models to discretize their Al-SiC microstructure, but Christman et al. (1989) placed a SiC inclusion encased in Al (figure 2.3(a)); whereas, Tvergaard (1990) had the SiC inclusion running halfway up the outside edge of the domain (figure 2.3(b)). They then compared the results of their competing models to the same uniaxial tensile tests performed on the metal alloy. In general, Tvergaard (1990) modeled the stress-strain behavior of the metal alloy more closely by more accurately modeling the generated shear stresses between the ceramic and metal, but the primary result is that due to the global periodicity assumption, RVE selection is paramount in creating predictive unit cell models.

Bao et al. (1991) extended the axisymmetric unit cell idea to model elastoplasticity, creep, and elliptical inclusions in Al-SiC composites. They also studied the effects of RVE selection, but didn't compare their results to any data. Brockenbrough et al. (1991) developed another unit cell model to study the packing and shape of boron fibers in aluminum. Rather than an axisymmetric model, Brockenbrough et al. (1991) developed two dimensional cross-sections of Al-B composites with the unit cells containing up to 30 separate boron inclusions. They compared some of their results to experimental data with reasonably good agreement. Nakamura and Suresh (1993) further extended the Brockenbrough et al. (1991) model to include residual stresses caused by heating the material during production. Lastly, van der Sluis et al. (1999) and van der Sluis (2001) took the unit cell idea and focused on using information from the microscale simulations to inform macroscale constitutive models. They modeled a perforated metal plate in uniaxial tension, shear, biaxial tension, and cyclic tension-compression testing using a Perznya type viscoplasticity model for a unit cell containing one perforation. Using boundary conditions imposed by the deformation of the macroscale domain, the unit cell was used to provide homogenized material constants to a second, macroscale version of the viscoplasticity model. Results from the homogenized medium were compared with a traditional finite element simulation of the perforated metal with good agreement between the models. Although all of these researchers studied the effect of their selected RVEs on their simulation results and one team, Brockenbrough et al., randomly selected the position of the boron inclusions in their RVEs, none of the researchers used a series of randomly generated RVEs to populate a statistical distribution of possible homogenized material parameters.

In the late 1990s and 2000s, multiscale computational homogenization, or global-local analysis, became a popular tool for studying heterogeneous materials. Also built on principles of mathematical asymptotic homogenization theory, global-local analysis can be traced back to Suquet (1985) who derived the thermodynamics of such analyses. Suquet also described the four basic steps of global-local analysis.

(1) Define an RVE that is "small enough" to fully capture the heterogeneities of the material's microstructure, but "large enough" to also represent the global behavior.

(2) Define the macro variables from micro variables using volume averaging relationships. In other words, given a micro-field, what is the average of that micro-field over the RVE?

(3) Using the averaging relationships, microscopic constitutive laws, equilibrium equations, and geometric assumptions - e.g. local periodicity, derive the micro variables from known macro variables through a localization procedure. Typically, a macro strain is applied to micro-scale boundary value problems via a localization procedure.

(4) Now that the micro-to-macro (localization) and macro-to-micro (averaging) relationships have been defined, complete the homogenization procedure to calculate the unknown macro fields.

Building on this work, many researchers developed FE^2 multiscale computational homogenization models. FE^2 is a reference to using the finite element method for both the macro and microscale spatial discretization. In general, a global FEM model is created representing the problem domain. Each integration point within the global FEM model is assigned an RVE representing the microstructure of the material at that point that is also spatially discretized using an FEM model.

From the global balance of linear momentum, the macroscale deformation gradient ($\boldsymbol{F}$) is used to define boundary conditions on the RVEs, which then cause each RVE to deform. The microscale stress and displacement fields are then averaged over the volume of the RVE and homogenized over the global domain to construct the macroscale stress ($\boldsymbol{\Sigma}$) and displacement fields. Figure 2.4 depicts the general idea of multiscale computational homogenization models. As summarized by Smit et al. (1998), the fundamental assumption of global-local analyses is that the averaged micro stress and displacement fields over an RVE are equal to the macro stress and displacement fields at that macroscale location.

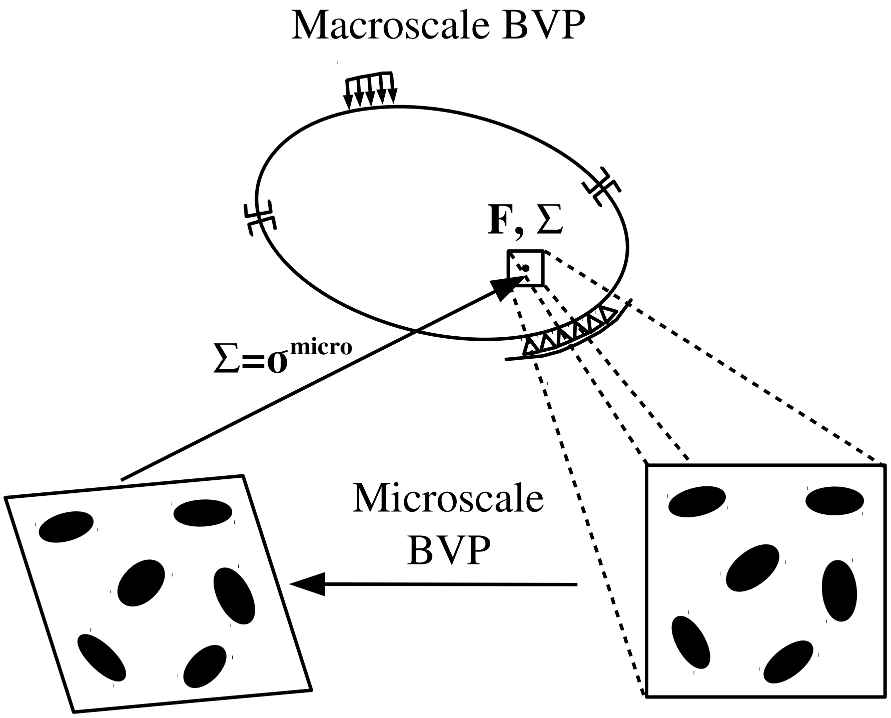


Figure 2.4: General Idea of FE^2 Multiscale Computational Homogenization Models

Ghosh et al. (1995) and Ghosh et al. (1996) created a two dimensional FE^2 model using the Voronoi cell method to generate the microscale discretization of their RVEs. Given the volume fraction of voids in a perforated metal, random voids were generated in the microstructure, and a Dirichlet tessellation was used to create a mesh around the voids such that each element only contained one. An elastoplastic constitutive model was developed for the metal, and the model was implemented in an ABAQUS user material subroutine (UMAT). ABAQUS UMATs allow a modeler to write his or her own constitutive subroutines, which is essentially what the RVE becomes in global-local analyses. ABAQUS provides the user with the strain calculated from solving the macroscale balance of momentum, and in turn, the UMAT provides ABAQUS with the averaged stress in the RVE at each integration point. Ghosh et al. (1996) verified their model against a unit cell model, equivalent continuum model, and analytical solutions concluding they could accurately predict the stress-strain response of the material. The authors also conducted analyses of different configurations of the microstructure, particularly varying the volume fraction and geometry of the voids, concluding that the both the macro and micro stress fields varied considerably given the different configurations. Ghosh et al. (2001) updated their original model to include an adaptive methodology that calls the global-local analysis routines only if the global FEM model senses the onset of strain localization by tracking regions with locally large gradients of macroscopic variables such as stress, strain, or strain energy.

Smit et al. (1998) also worked with two dimensional FE^2 models for perforated metals, but extended the global-local analysis to include finite strain elasto-viscoplasticity. Similarly, Miehe et al. (1999a) and Miehe et al. (1999b) developed a two dimensional, finite strain, elastoplasticity FE^2 model for polycrystalline materials, and they compared their results to experimental data with good overall agreement. Feyel and Chaboche (2000) studied SiC/Ti composite metals for use in airplane turbines using a two dimensional FE^2 model. They also parallelized their code using the finite element tearing and reconnect (FETI) method. Feyel and Chaboche (2000) incorporated a global Newton-Raphson algorithm and perturbation method to calculate a homogenized consistent tangent to aid with convergence issues resulting from the nonlinearity imposed by solving each macro integration point with its own microscale RVE.

In addition to studying the mathematics behind the fundamental assumption that the averaged micro stress field over an RVE is equal to the macro stress field at the macroscale location (so called “two scale convergence”), Terada and Kikuchi (2001) also implemented a global Newton-

Raphson algorithm in an FE^2 model to show the difference between a full Newton-Raphson iteration and a simplified "staggered" method. Kouznetsova (2002) extended the global-local analysis framework to include second order gradients that could capture large spatial variations in the deformation gradient. Work on FE^2 models has continued through the 2010s and was summarized nicely in the meta-analysis of the literature by Geers et al. (2010). Geers et al. also outlined current improvements on the technique to include continuous-discontinuous homogenization using X-FEM, thermomechanics and other multiphysics problems, extensions to shell elements, and extensions to include cohesive zones.

To this point, the discussion of hierarchical multiscale modeling history has focused on composite metals primarily used in structural, mechanical, and aerospace engineering. In these works, the microscale was most often represented through an FEM model that calculated the micro stress field through solving a boundary value problem. In granular materials, the microscale is most often represented by the discrete element method (DEM) (Cundall and Strack, 1979), which tracts the movement of each individual particle. Without getting into the history of granular mechanics, the primary work on the multiscale modeling of geomaterials through hierarchical multiscale modeling for civil and geotechnical engineering applications began with deriving methods for calculating average stress and strain from granular assemblies. In particular, Christoffersen et al. (1981) derived the volume averaged Cauchy stress from a summation over the particle-to-particle contact forces. Expanding on this work, Bagi (1996) formulated a method to calculate the strain by overlaying a mesh whose nodes were located at each particle's centroids in a granular assembly. By tracking the deformation of the mesh, Bagi was able to derive an averaged strain. More recently, Miehe and Dettmar (2004) extended the stress average calculations to include finite stress measures, and Zhang and Regueiro (2015) extended the strain calculations to include finite strain measures.

Similar to the aforementioned unit cell models, a class of multiscale models for granular materials focus on deriving information about a global domain through studying the particle-to-particle contact of DEM RVEs. Borja and Wren (1995) and Wren and Borja (1997) derived macroscale elastoplastic constitutive model parameters through the particle contacts of a two dimensional, circular particle DEM assembly. They then ran displacement (Borja and Wren, 1995) and stress (Wren and Borja, 1997) driven quasi-static simple shear tests for different RVE configurations and were able to predict the onset of strain localization - i.e. shear banding.

Ehlers et al. (2003) and D'Addetta et al. (2004) posited that in traditional FEM analysis strain localization causes the boundary value problem to become ill-posed requiring regularization in order to be solved. The regularization can be done "naturally" through use of micropolar continuum models, or Cosserat materials as they call them, so they derived homogenization methods to determine the macroscale stress and couple stress of the enriched continuum from DEM assemblies. Ehlers et al. (2003) used two dimensional, convex polygon particles for their DEM assemblies, and also studied RVE size to help determine Suquet's (1985) optimal size. They developed a heuristic that claims an RVE size with a characteristic length of approximately 5x the average particle diameter will approximate the material well. D'Addetta et al. (2004) used Voronoi polygons for their DEM assemblies and simulated cohesionless biaxial tests and cohesive "Brazillian" tests. Li et al. (2010a) and Li et al. (2010b) also focused on homogenization of granular assemblies to micropolar continua, but with an emphasis on capturing particle rotations.

In recent years, FEM-DEM multiscale computational homogenization techniques (hierarchical multiscale models) for granular materials have grown from the FE^2 models for metal composites. Rather than FEM modeling the microscale as in FE^2 models, the microscale RVE is replaced with a representative DEM particle assembly in FEM-DEM models. In the language of mathematical asymptotic homogenization theory for periodic media as explained by Kaneko et al. (2003), the general idea for these types of models is as follows and can also be seen pictorially described on figure 2.5.

(1) Solve the macroscale boundary value problem to calculate the macroscopic strain through the deformation gradient ($\boldsymbol{F}$).

(2) Apply boundary conditions to the microscale DEM RVE given the macroscopic strain.

(3) Allow the DEM RVE to deform given the boundary conditions. The averaged microscopic

stress ($\sigma_{ij}^{\text{micro}}$) is calculated directly from the DEM simulation.

(4) It is then assumed that $\sigma_{ij}^{\text{micro}}$ is equal to the macroscopic stress (Σ_{ij}) at the given macro point ($\Sigma_{ij} = \sigma_{ij}^{\text{micro}}$).

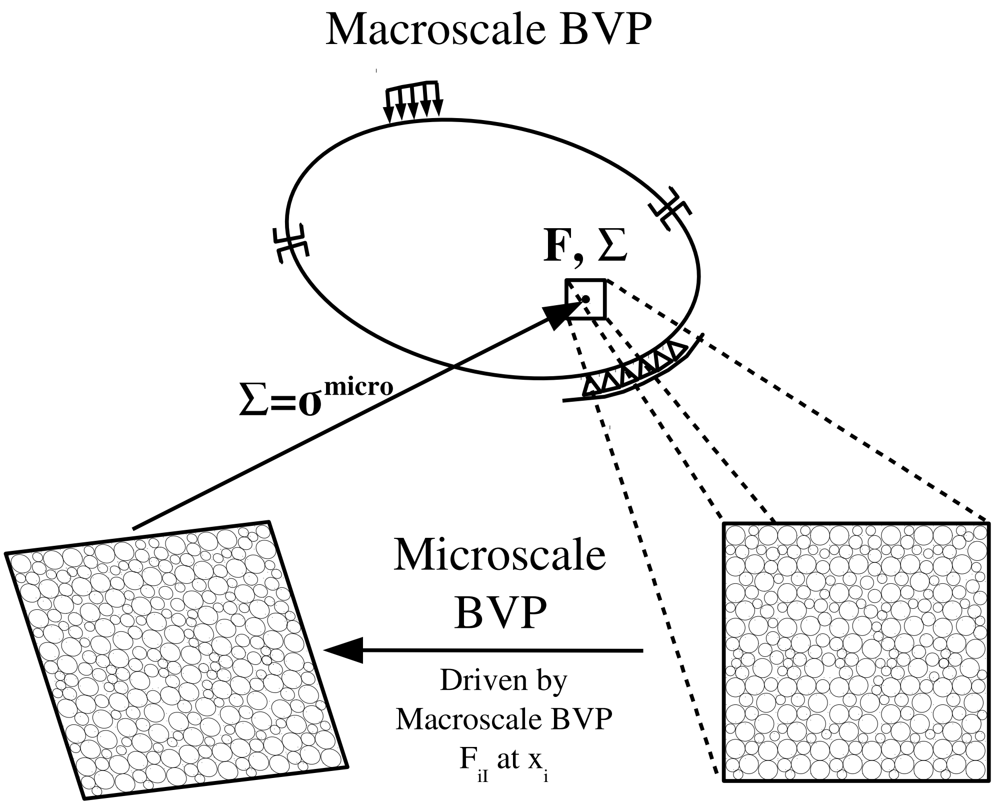
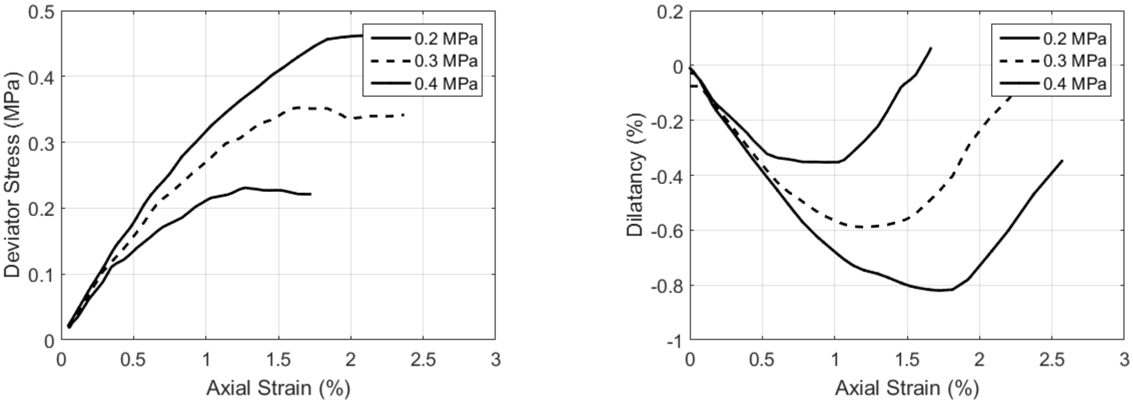


Figure 2.5: General Idea of FEM-DEM Multiscale Computational Homogenization Models

Kaneko et al. (2003) applied these concepts of FEM-DEM global-local analysis to a model that used two dimensional, circular DEM particles with linear elastic contact and Coulomb friction models to study both cohesive and cohesionless materials. They conducted biaxial compression tests on a cohesionless granular material and were able to qualitatively capture the dilatant behavior and small strain deviatoric stress response that is seen in geotechnical testing on dry sands (figure 2.6). The simulations were able to capture the macroscale stress concentrations that lead to strain localization, but were unable to capture the actual shear strength of the soil as a slip surface could not form. The researchers also conducted bending simulations on a cohesive granular material. As seen on figure 2.7, they were able to show the development of microcracks, or debonding of the grains, resulting in permanent deformation of the beam.



(a) Deviatoric Stress

(b) Dilatacny

Figure 2.6: Biaxial Compression Results from Kaneko et al. (2003)

Meier et al. (2008) and Meier et al. (2009) pointed to an issue with Kaneko et al. (2003) in that for simulating quasi-static tests, their FEM-DEM model required implementing an explicit dynamic relaxation scheme at the macroscale to account for the inherent dynamics of the DEM code. To solve this problem, Meier et al. outlined a procedure to estimate an approximate tangent operator from the contact forces between particles for use in a Newton-Raphson iteration scheme. It should be noted that to approximate quasi-static conditions, background damping still needs to be included in the DEM model. They then implemented the procedure into a two dimensional, circular particle FEM-DEM model with a linear elastic contact model and frictionless particles. In general, the authors were able to achieve quadratic convergence of the Newton-Raphson algorithm, but noted that occasionally due to non-smooth restructuring of the interparticle contacts with the deformation of the RVE, the convergence rate of the global solution suffered. They suggested implementing an adaptive timestepping scheme to mitigate these potential convergence issues. Meier et al. (2009) conducted slope stability and biaxial compression simulations using the FEM-DEM code developed in Meier et al. (2008). Similarly to Kaneko et al. (2003), they were able to qualitatively capture the expected behavior of both the slope stability and biaxial compression simulations, but were unable to predict the shear strength of the modeled soil.

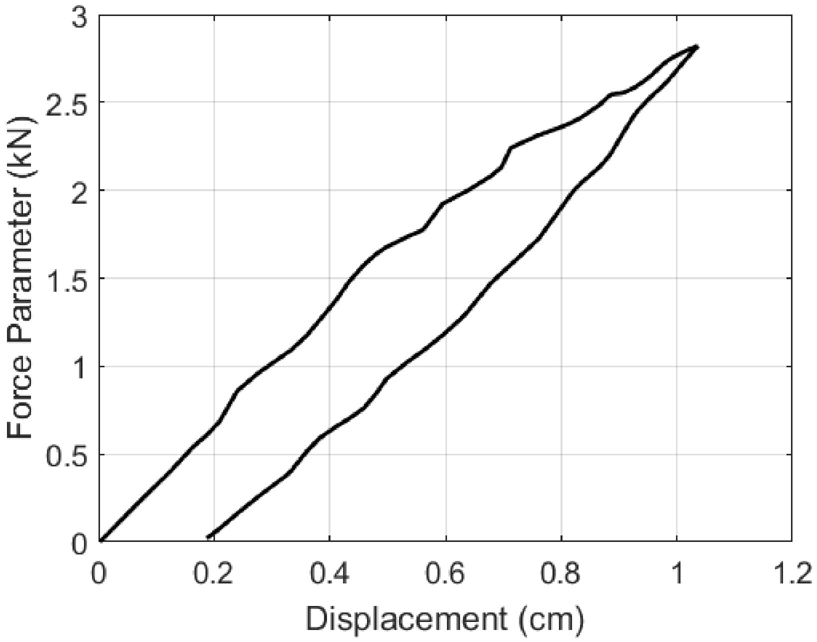


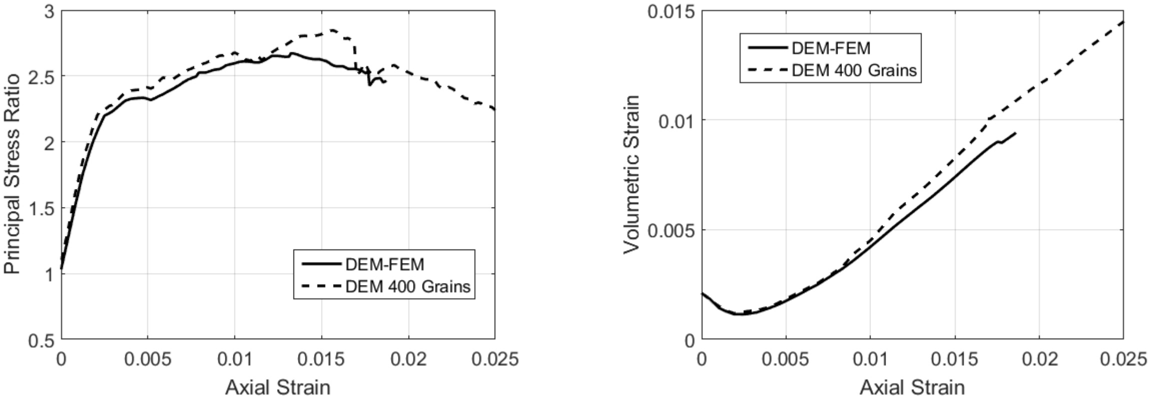
Figure 2.7: Beam in Bending Results from Kaneko et al. (2003)

Meier et al. also studied the effect of applied boundary conditions on the RVE. In keeping with the assumed periodicity of the DEM microstructure, they analyzed two variants of periodic boundary conditions: the Taylor-Voigt and restricted Taylor boundary conditions. For the Taylor-Voigt boundary conditions, the macroscale deformation gradient is applied to each particle in DEM assembly through

$$\boldsymbol{x} = \boldsymbol{F} \cdot \boldsymbol{X} \tag{2.2}$$

where $\boldsymbol{x}$ is the current position of the particle, $\boldsymbol{X}$ is the initial position of the particle, and $\boldsymbol{F}$ is the macroscale deformation gradient. For the restricted Taylor boundary conditions, the macroscale deformation gradient is only applied to the particles along the boudnary of the DEM assembly. Meier et al. concluded that the Taylor-Voigt boundary conditions produced an overly stiff response in the granular assembly and the restricted Taylor boundary conditions better captured the dilatant behavior of the soil.

Nitka et al. (2011) used a similar FEM-DEM model as Meier et al. (2008) - i.e. quasi-static, two dimensional, circular particles, linear elastic contact, and a tangent calculation for using in a Newton-Raphson algorithm - to study oedometer and biaxial compression simulations compared to full DEM analyses of the same problems. Unlike Meier et al. (2008), Nitka et al. chose to use a perturbation method to estimate the tangent operator and included Coulomb friction between particles. Results from the biaxial compression tests can be found on figure 2.8. The FEM-DEM simulations matched the full DEM simulations well, but suffered from convergence issues with increasing axial strain.



(a) Principal Stress Ratio

(b) Volumetric Strain

Figure 2.8: Biaxial Compression Results from Nitka et al. (2011)

As part of a larger work developing a number of open source multiscale models for granular materials, Stránský and Jirásek (2012) developed a FEM-DEM model using the codes OOFEM (FEM) and YADE (DEM). The primary goal of the work was to implement a series of preexisting models into the two open source C++ codes including a surface coupling model, two volume coupling models (master/slave and Arlequin methods), FEM-DEM hierarchical multiscale model, and a contact analysis model. The FEM-DEM model improved on other works by being a full three dimensional implementation of both the FEM and DEM models. Contact between the spherical DEM particles was governed using a frictionless linear elastic contact model, and the boundary conditions were imposed in a manor similar to the Taylor-Voigt method described in Meier et al. (2009) (Stránský and Jirásek, 2011). Like Nitka et al. (2011), Stránský and Jirásek (2012) used (and fully derived) the approximate homogenized elastic tangent for use in a Newton-Raphson algorithm. Although the focus of the paper wasn't specifically the FEM-DEM model, the authors did present reasonable pictorial, qualitative results from a uniaxial strain simulation.

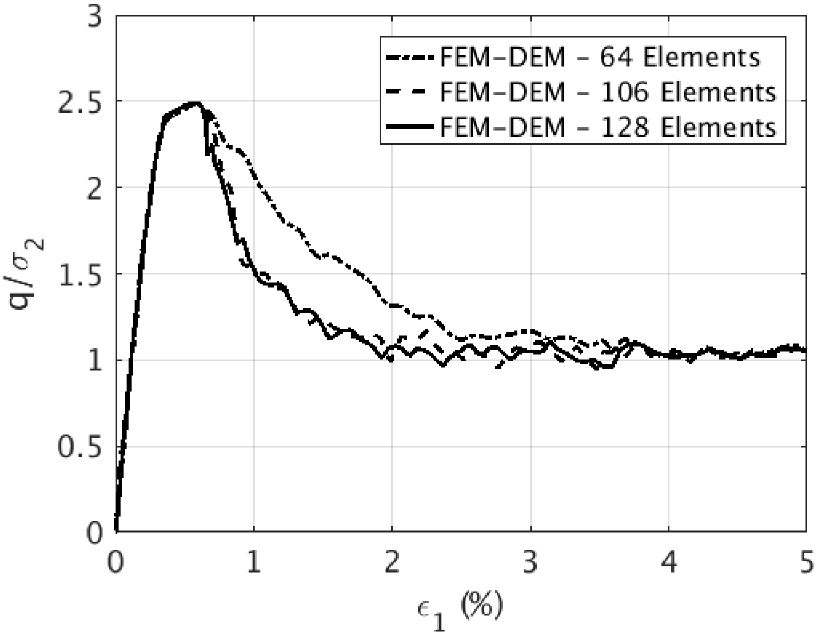


Figure 2.9: Biaxial Compression Results from Nguyen et al. (2014)

Like Nitka et al. (2011) and Meier et al. (2008), Nguyen et al. (2014) developed a quasi-static, two dimensional FEM-DEM model with circular particles, a linear elastic contact model with cohesion. The preexisting code "Lagamine" was used for the FEM model, which can handle finite strains (although the authors did not probe the finite strain regime), and a perturbation method was used to estimate the consistent tangent using in a Newton-Raphson iteration. Nguyen et al. demonstrated the behavior of their cohesive DEM model through two "pure DEM" biaxial simulations (one biaxial compression and the other biaxial extension) that successfully generated a reasonable Mohr-Coulomb failure envelope. The FEM-DEM model was then used to simulate three biaxial compression tests with different FEM meshes (64 and 128 element structured meshes and a 106 element non-structured mesh). Results from the simulations can be found on figure 2.9. The axial strain (in the shearing direction) is plotted along the abscissa (ϵ_1), and the deviator stress ($\sigma_1 - \sigma_2$) normalized by the confining stress (σ_2) along the ordinate. The FEM-DEM model converged to a single solution upon refinement of the mesh.

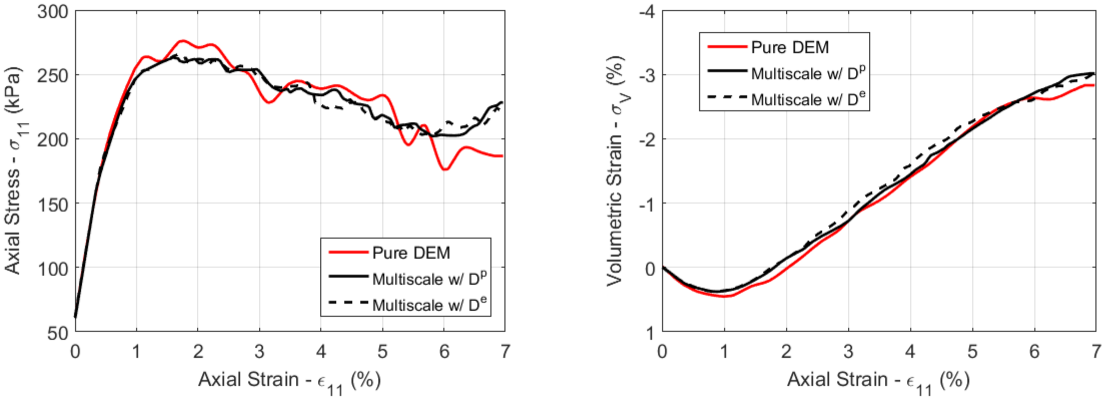
Guo and Zhao (2014) also developed a quasi-static, two dimensional, circular particles, linear elastic contact with Coulomb friction FEM-DEM model including a tangent calculation for using in an implicit Newton-Raphson algorithm. Rather than only implementing one of the two tangent operator approximations, the authors implemented both. The tangent operator using the perturbation method ($\boldsymbol{D}^p$), also implemented by Nitka et al. (2011), is

$$\boldsymbol{D}^p = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}}\bigg|_{\epsilon=\epsilon_0} \approx \frac{\boldsymbol{\sigma}(\boldsymbol{\epsilon}_0 + \xi \boldsymbol{\Delta}^{kl}) - \boldsymbol{\sigma}(\boldsymbol{\epsilon}_0)}{\xi} \tag{2.3}$$

where a small perturbation of magnitude ξ to the current strain ($\boldsymbol{\epsilon}_0$) is summed over each of the kl directions. For each kl, $\boldsymbol{\Delta}^{kl}$ is a second order tensor of zeros except for in the kl direction, which has a value of one. For the second tangent operator approximation, the operator is estimated from the homogenized bulk elastic modulus by summation over the interparticle contacts using the equation

$$\boldsymbol{D}^{\boldsymbol{e}} = \frac{1}{V}\sum_{c}\left(k_n \boldsymbol{n}^c \otimes \boldsymbol{d}^c \otimes \boldsymbol{n}^c \otimes \boldsymbol{d}^c + k_t \boldsymbol{t}^c \otimes \boldsymbol{d}^c \otimes \boldsymbol{t}^c \otimes \boldsymbol{d}^c\right) \tag{2.4}$$

where V is the volume of the DEM RVE, c is the total number of particle contacts, k_n and k_t are the contact normal and tangential stiffness from the selected contact model, $\boldsymbol{n}^c$ and $\boldsymbol{t}^c$ are the unit normal and tangential directions at the contact surface, and $\boldsymbol{d}^c$ is a vector between the two particles in contact's centroid. Unlike Meier et al. (2009), Guo and Zhao used "prescribed displacement" boundary conditions to apply strain to the DEM assemblies. Rather than directly prescribing the displacement of all or a subset of the DEM particles, the authors prescribed the displacement of the four corners of the DEM assembly thereby causing the bounding walls of the assembly to move. The moving walls then transferred force to contacting particles. Finally, Guo and Zhao parallelized their code by sending each Gauss point calculation of a DEM RVE to a separate processor and recombining the results on a master node.



(a) Stress

(b) Volumetric Strain

Figure 2.10: Single Element Biaxial Compression Results from Guo and Zhao (2014)

Guo and Zhao compared results from one element biaxial compression tests using both tangent operator approximations and a full DEM simulations (figure 2.10). Unlike Meier et al. (2009) whose model achieved quadratic convergence, the simulations with either tangent operator achieved sub-linear convergence, which they attributed to the added frictional contact between particles, but the authors were able to conclude that $\boldsymbol{D}^e$ converged faster and was more stable given biaxial compression conditions. Overall, they were able to capture the stress and dilatant response of the soil well, particularly as compared to the full DEM simulations. Similarly, results from multi-element biaxial compression tests also were able to capture the post-peak response of the soil, but as in FEM models, finer meshes generally result in convergent, but larger softening responses than coarser meshes (see figure 2.11). In addition to the biaxial compression tests, Guo and Zhao also ran a cyclic shear simulation using the FEM-DEM model and were able to capture the hysteretic nature of the stress-strain response.

Citing the mesh dependence of previous FEM-DEM biaxial compression results, Liu et al. (2016) developed an FEM-DEM model to alleviate the mesh dependence of the post peak response. After formally deriving the governing equations using mathematical asymptotic homogenization theory, the authors developed a dynamic, corotational FEM formulation with a staggered nonlocal

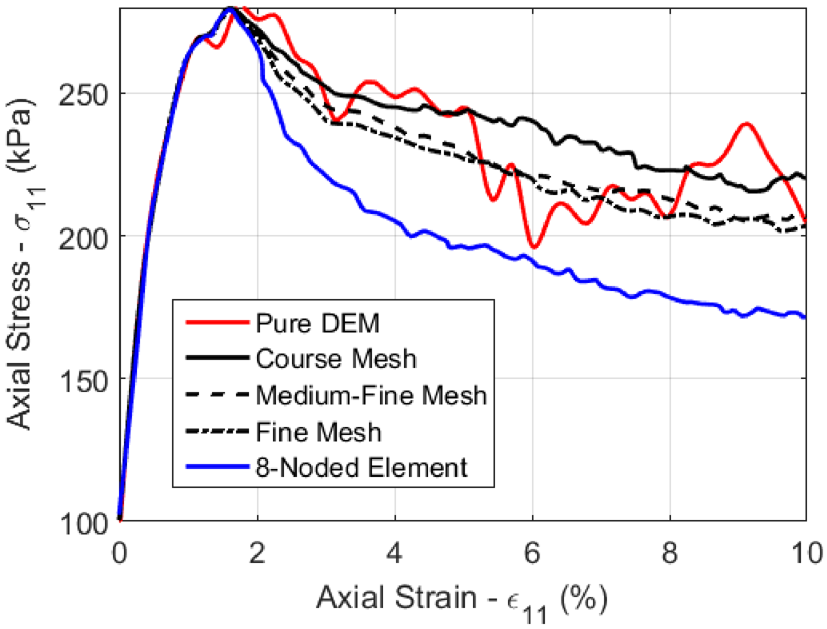


Figure 2.11: Multi-Element Biaxial Compression Results from Guo and Zhao (2014)

operator to formally introduce the length scale minimizing the thickness of the shear bands that develop during the simulations. For the DEM model, Liu et al. developed a three dimensional, spherical particle model with a Hertz-Mindlin contact model. Although the FEM model included inertial terms, the DEM model was always run using a dynamic relaxation technique most often used to mimic quasi-static problems. Displacements were applied the the DEM RVEs using a hybrid prescribed displacement and restricted Taylor boundary conditions where a layer of particles were embedded into the walls the RVEs. The movement of these particles was then directly prescribed by the applied corotational strain. The model was validated through simple shear, biaxial compression, and one dimensional wave propagation simulations. The results from all simulations compared well with and were significantly faster than DEM simulations. Also, the biaxial compression simulation results were less mesh dependent than previous works.

Building on the FEM-DEM multiscale computational homogenization idea of using grain-scale information to inform macroscale models, Andrade and Tu (2009) developed a model that informed a continuum constitutive model, rather than fully replacing it. As with the other FEM-

DEM models, the finite element method was used to discretize the macroscale boundary value problem. At each integration point, the macroscale strain was calculated and passed to a DEM RVE as boundary conditions. The DEM RVE was then allowed to deform, but rather than calculating the homogenized stress, the authors used the DEM to approximate the friction angle and dilatancy parameter, which was then passed directly into an elastoplastic continuum constitutive model. Essentially, the method replaced the phenomenological internal state variable evolution equations rather than the entire constitutive model. This concept was extended by Chen et al. (2011) using assumed enhanced strain (AES) methods implemented in the macroscale FEM to capture macroscale shear banding. Also, Andrade et al. (2011) used the same concept in a unit cell model. A full DEM simulation of the problem domain was first conducted and used to estimate friction angle and dilatancy parameter curves as a function of the deformation. These curves were then used in place of traditional hardening/softening evolution equations in an elastoplastic FEM model of the same problem. All three works compared their results to experimental data primarily for triaxial experiments with reasonably good agreement.

Chapter 3

Arena Geomaterial Constitutive Model

Arena is an isotropic, pressure sensitive, and semi-empirical geomaterial constitutive model developed for the ONR funded Soil Blast MURI Project by Brannon et al. (2014) at the University of Utah in Salt Lake City. Three distinct versions of the model were developed called Arena 2.0, 3.0, and 4.0. Arena 2.0 allows for isotropic hardening, only allows for associative plasticity, does not contain a limit surface to model ultimate failure, and is not Lode angle (3rd invariant) dependent (i.e. does not allow for differences in triaxial compression and extension). It can make empirical adjustments to the modeled material's bulk modulus to account for fluid filled porous effects, uses an overstress model to include viscoplasticity, and now includes a void disaggregation algorithm to account for the added volume when grains disaggregate under explosive loading. Arena 3.0 adds nonassociative plasticity and the ability to model more complicated shear behaviors other than linear Drucker-Prager elastoplasticity, and Arena 4.0 further adds Lode angle dependence. All three models were implemented in the MPM code Uintah, which is also developed at the University of Utah, using a novel, geometrically based return algorithm. Arena 3.0 is the most developed of the three versions hence it will be the focus of further discussion.

The following chapter outlines the implementation of the Arena geomaterial model into an ABAQUS User Material Subroutine (UMAT) beginning with an overview of the thermodynamics of elastoplasticity (section 3.1). Section 3.2 describes the equations of the Arena model, and section 3.3 its implementation in an ABAQUS UMAT. In section 3.4, the ABAQUS implementation of the constitutive model was calibrated against a series of geotechnical experiments on both Colorado

Mason Sand and Boulder Clay.

3.1 Summary of Small Strain Elastoplasticity

The following is a basic summary of the equations of small strain elastoplasticity as they relate to Arena. The derivation and notation of the following model mimics the work of Foster et al. (2005), but the model follows from the works Fossum and Brannon (2004), Brannon et al. (2009), and Brannon et al. (2014) as well as Foster et al. (2005). The Clausius-Duhem inequality in localized form for isothermal conditions and only considering isotropic hardening reads

$$\sigma_{ij}\dot{\epsilon}^{p}_{ij} - \kappa\dot{\zeta} \geq 0 \tag{3.1}$$

where σ_{ij} is the Cauchy stress, $\dot{\epsilon}^{p}_{ij}$ is the infinitesimal plastic strain rate, κ is the stress-like internal state variable (ISV) governing isotropic hardening, and $\dot{\zeta}$ is the thermodynamically conjugate strain-like ISV. Assuming linear elasticity and a linear dependence of κ on ζ, a Helmholtz free energy function ($\rho\Psi$) can be written,

$$\rho\Psi(\epsilon^{e}_{ij}, \zeta) = \frac{1}{2}\epsilon^{e}_{ij}c^{e}_{ijkl}\epsilon^{e}_{kl} + \frac{1}{2}c^{\kappa}\zeta^{2} \tag{3.2}$$

such that

$$\sigma_{ij} = \rho\frac{\partial\Psi}{\partial\epsilon^{e}_{ij}}; \quad \kappa = \rho\frac{\partial\Psi}{\partial\zeta} \tag{3.3}$$

where ϵ^{e}_{ij} is the infinitesimal elastic strain, c^{e}_{ijkl} is the 4th order elastic compliance tensor, and c^{κ} is a constant hardening modulus. The constitutive equations in rate form can then be written as

$$\dot{\sigma}_{ij} = c^{e}_{ijkl}\dot{\epsilon}^{e}_{kl} = c^{e}_{ijkl}(\dot{\epsilon}_{kl} - \dot{\epsilon}^{p}_{kl}); \quad \dot{\kappa} = c^{\kappa}\dot{\zeta} \tag{3.4}$$

after assuming the additive decomposition of the strain tensor (i.e. $\epsilon_{ij} = \epsilon^{e}_{ij} + \epsilon^{p}_{ij}$).

In plasticity theory, a yield function dependent on the stress state and ISVs ($f(\sigma_{ij}, \kappa)$) is developed such that when $f < 0$ the material behaves elastically. If the stress experienced by the material causes $f = 0$, the stress state is on the yield surface and plastic dissipation occurs with any further applied load. Values of $f > 0$ aren't allowed in order that during plastic flow the stress state remains on the yield surface. In associative plasticity, the plastic strain rate is assumed to be governed by a flow rule dependent on the yield function

$$\dot{\epsilon}^{p}_{ij} = \dot{\gamma}\frac{\partial f}{\partial \sigma_{ij}} \tag{3.5}$$

that determines the direction of plastic flow ($\frac{\partial f}{\partial \sigma_{ij}}$) and the amount of dissipation through the plastic multiplier or consistency parameter ($\dot{\gamma}$). By substituting this equation into the constitutive equation for $\dot{\sigma}_{ij}$ (equation 3.4), the evolution of the Cauchy stress rate can be written as follows.

$$\dot{\sigma}_{ij} = c^{e}_{ijkl}\left(\dot{\epsilon}_{kl} - \dot{\gamma}\frac{\partial f}{\partial \sigma_{ij}}\right) \tag{3.6}$$

Through isotropic hardening, the yield surface is allowed to expand under plastic loading allowing for the material's elastic region to grow. The hardening is assumed to be proportional to the plastic multiplier such that

$$\dot{\zeta} = \dot{\gamma}h(\sigma_{ij}, \kappa) \tag{3.7}$$

where $h(\sigma_{ij}, \kappa)$ is a hardening function that governs the behavior of κ given the stress state. When substituted into the constitutive equation for $\dot{\kappa}$ (equation 3.4), the following evolution equation for the isotropic hardening is derived.

$$\dot{\kappa} = \dot{\gamma}c^{\kappa}h(\sigma_{ij}, \kappa) = \dot{\gamma}h^{q}(\sigma_{ij}, \kappa) \tag{3.8}$$

Because $f \leq 0$ and the material state is required to remain on the yield surface during plastic flow, the time derivative of the yield surface must remain zero. This requirement is known as the consistency condition. By taking the time derivative of the yield function using the chain rule and setting it equal to zero, the following equation results.

$$\dot{f} = 0 = \frac{\partial f}{\partial \sigma_{ij}}\dot{\sigma}_{ij} + \frac{\partial f}{\partial \kappa}\dot{\kappa} \tag{3.9}$$

After substituting the evolution equations for the Cauchy stress rate (equation 3.6) and isotropic hardening ISV (equation 3.8) into the consistency condition, the following equation for the plastic multiplier results.

$$\dot{\gamma} = \frac{\frac{\partial f}{\partial \sigma_{ij}} c^{e}_{ijkl}\dot{\epsilon}_{kl}}{\frac{\partial f}{\partial \sigma_{ij}} c^{e}_{ijkl}\frac{\partial f}{\partial \sigma_{kl}} - \frac{\partial f}{\partial \kappa}h^{q}} = \frac{1}{\chi}\frac{\partial f}{\partial \sigma_{ij}}c^{e}_{ijkl}\dot{\epsilon}_{kl}; \quad \chi = \frac{\partial f}{\partial \sigma_{ij}}c^{e}_{ijkl}\frac{\partial f}{\partial \sigma_{kl}} - \frac{\partial f}{\partial \kappa}h^{q} \tag{3.10}$$

Finally, the evolution equation for $\dot{\sigma}_{ij}$ can then be written

$$\dot{\sigma}_{ij} = \left(c^{e}_{ijkl} - \frac{1}{\chi}c^{e}_{ijab}\frac{\partial f}{\partial \sigma_{ab}}\frac{\partial f}{\partial \sigma_{cd}}c^{e}_{cdkl} \right)\dot{\epsilon}_{kl} = c^{ep}_{ijkl}\dot{\epsilon}_{kl} \tag{3.11}$$

where c^{ep}_{ijkl} is called the continuum elastoplastic tangent.

3.2 Arena Geomaterial Constitutive Model

Arena's yield surface is a two function equation composed of a function that controls the shear behavior (F_f) and an elliptical cap that controls the compressive behavior (F_c) (Brannon et al., 2014).

$$f = \sqrt{J_2} - \begin{cases} F_f(I_1) & : I_1 \geq \kappa \\ F_f(I_1)F_c(I_1, X) & : I_1 < \kappa \end{cases} \tag{3.12}$$

I_1 is the first invariant of the Cauchy stress, J_2 is the second invariant of the deviatoric Cauchy stress tensor, and κ and X are interdependent variables that control the location of the cap, which will be described later. For the ABAQUS implementations, the yield function will be described in its square form (equation 3.13) to simplify some of the yield function derivatives.

$$f^{\text{sq}} = J_2 - \begin{cases} F_f^2(I_1) & : I_1 \geq \kappa \\ F_f^2(I_1)F_c^2(I_1, X) & : I_1 < \kappa \end{cases} \tag{3.13}$$

As mentioned by Foster et al. (2005), both f and f^{sq} will predict an elastic response or plastic yielding for the same stress state effectively making them equivalent for our purposes.

Arena 3.0 allows for nonlinear shear limit functions of the form

$$F_f = A_1 - A_3 \mathrm{e}^{A_2 I_1} - A_4 I_1 \tag{3.14}$$

where A_1, A_2, A_3, and A_4 are user supplied material parameters. For simplicity, this implementation will use a linear Drucker-Prager shear limit function that is used in Arena 2.0. F_f will then be the simple, linear function (figure 3.1)

$$F_f = \beta(I_1^{\max} - I_1) \tag{3.15}$$

where $I_1^{\max}$ is the tensile limit of the first stress invariant, and β is the tangent of geomaterial's friction angle (ϕ). Therefore, $A_1 = \beta I_1^{\max}$, $A_3 = 0$, and $A_4 = \beta$.

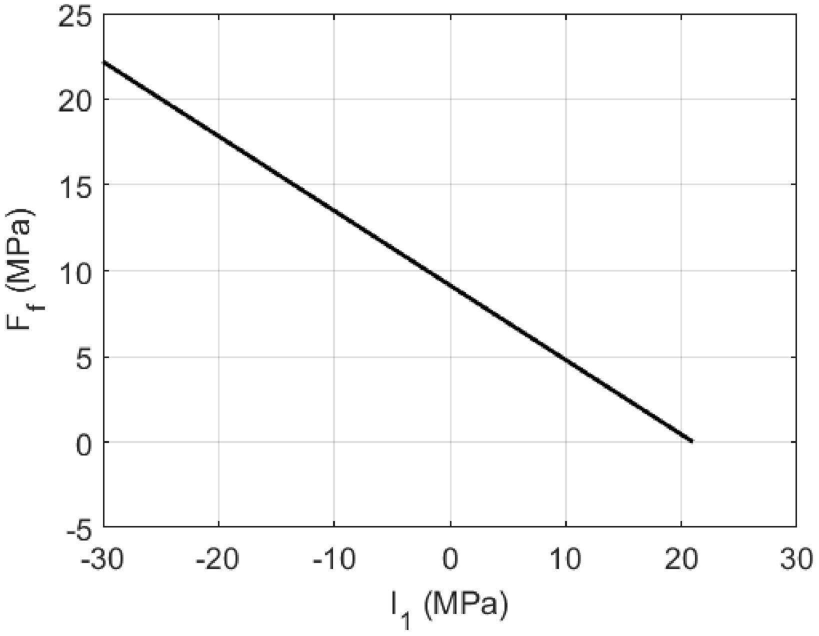


Figure 3.1: Example Shear Limiting Function (F_f)

It should be noted that for both Arena implementations (the ABAQUS and Uintah versions), the user inputs β and $I_1^{\max}$ rather than A_1, A_2, A_3, and A_4 directly as the "A" parameters do not have obvious physical interpretations and the alternative user inputs better lend themselves to statistical perturbations as described by Strack et al. (2015). In order to access other shear envelopes, users can specify two other parameters in addition to β and $I_1^{\max}$. These are c and γ, which represent any internal cohesion in the material and the high stress slope of the shear surface. Depending on which parameters are input, Arena infers cooresponding values for A_1, A_2, A_3, or A_4. If β and $I_1^{\max}$ are specified and c and γ are set to zero, then the linear Drucker-Prager surface outlined above is generated. If only c is specified and the other variables are zero, then the von Mises surface is generated, where $A_1 = c$ and A_2, A_3, and A_4 are equal to zero. Other, more complicated surfaces can be generated through specifying additional combinations of the input parameters. A full list can be found in the "computeLimitParamters" function in the Arena 3.0 source code. For completeness, β, $I_1^{\max}$, c, and γ are called FSLOPE, PEAKI1, STREN, and YSLOPE in Uintah input files.

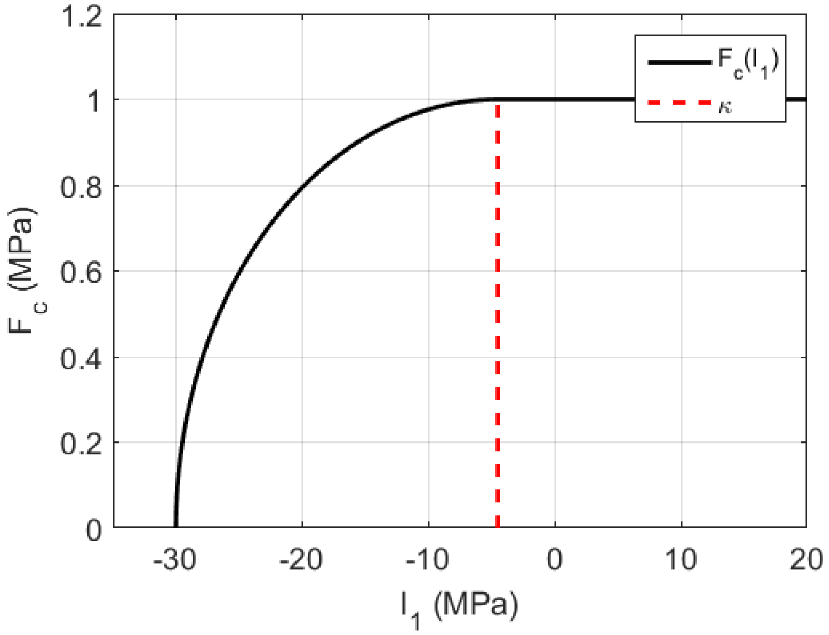


Figure 3.2: Example Elliptical Cap Function (F_c)

Under increasing confining pressure - i.e. decreasing I_1, the micro-cracks or interparticle contacts in the material are compressed thus increasing the shear stress required to yield the material, which effectively increases the strength of the material. Geomaterials are also generally porous, so to allow for plastic yielding due to collapse of these pores, F_f is multiplied by an elliptical cap function. F_c, plotted on figure 3.2, causes the yield surface to return to the hydrostatic axis such that the material can yield under purely isotropic compression. In Arena 3.0, $F_c = \sqrt{F_c^{\mathrm{sq}}}$ is defined as

$$F_c^{\mathrm{sq}} = \begin{cases} 1 & : I_1 \geq \kappa \\ 1 - \left(\frac{\kappa - I_1}{\kappa - X}\right)^2 & : I_1 < \kappa \end{cases} \tag{3.16}$$

where X is the intersection of the cap with the hydrostatic axis and κ, the isotropic hardening ISV from the previous section, controls the curvature of the cap and is defined as the point along the hydrostat where the yield surface begins to diverge from F_f.

Therefore, the full yield surface, in both square root and square form, can be written as follows (Brannon et al., 2014).

$$f = \sqrt{J_2} - \begin{cases} \beta(I_1^{\max} - I_1) & : I_1 \geq \kappa \\ \beta\left(I_1^{\max} - I_1\right)\sqrt{1 - \left(\frac{\kappa - I_1}{\kappa - X}\right)^2} & : I_1 < \kappa \end{cases} \tag{3.17}$$

$$f^{\mathrm{sq}} = J_2 - \begin{cases} \beta^2(I_1^{\max} - I_1)^2 & : I_1 \geq \kappa \\ \beta^2\left(I_1^{\max} - I_1\right)^2\left(1 - \left(\frac{\kappa - I_1}{\kappa - X}\right)^2\right) & : I_1 < \kappa \end{cases} \tag{3.18}$$

Initially, X is set to the elastic compressive limit (p_0). Any higher stress (more negative in compression) than p_0 along the hydrostatic axis causes pore collapse and an isotropic hardening of the yield surface through a decreasing κ (increase in absolute value). As κ evolves, X does as well; the two variables do not evolve independently. X is calculated from the updated value of κ by a function dependent on a user supplied cap shape, or cap ellipticity, parameter. The shape parameter (C_R) controls the geometry of the yield surface, and it is assumed that during yielding, the yield surface geometry remains constant. C_R's definition varies depending on the version of Arena. In Arena 2.0, C_R is the ratio between the width and height of the principal axes of the elliptical cap. In

Arena 3.0, C_R is defined as the ratio between the distance between the tensile limit and κ and the tensile limit and X.

$$C_R = \frac{I_1^{\max} - \kappa}{I_1^{\max} - X} \tag{3.19}$$

Therefore, the relation between κ and X is

$$X = \frac{\kappa - I_1^{\max}(C_R - 1)}{C_R} \tag{3.20}$$

and the initial value of κ (κ_0) is found by substituting the initial value of X (p_0) into the above equation and solving for κ_0 (Brannon et al., 2009).

$$\kappa_0 = I_1^{\max} - C_R(I_1^{\max} - p_0) \tag{3.21}$$

Figure 3.3 shows a plot of the initial yield surface given the following parameters for Berea Sandstone taken from an example simulation done by the University of Utah: $I_1^{\max}$= 21 MPa, β= 0.4354, p_0= -30 MPa, and C_R= 0.5. In order for the profile to not be distorted, the yield surface needs to be plotted in Lode coordinates, such that $z = I_1/\sqrt{3}$ and $r = \sqrt{2J_2}$ (Brannon et al., 2014).

Evolution of κ is controlled by an empirical "crush curve" fit to hydrostatic compression data where, ideally, the strains reached in the experiment allowed for complete collapse of the material's pores. The curve is of the form

$$\epsilon_v^p = p_3 \left(\mathrm{e}^{-p_1(p_0 - X)} - 1 \right) \tag{3.22}$$

where ϵ_v^p is the volumetric plastic strain, p_0 is three times the mean pressure at the elastic limit, p_1 is directly proportional to the initial slope of the porosity vs. pressure curve, and p_3 is the maximum plastic volumetric strain. Arena never explicitly refers to porosity, but uses the plastic volumetric strain as an indirect measure of the material's change in porosity. In other words, $p_3 + \epsilon_v^p$ is considered directly proportional to the porosity of the material (Brannon et al., 2009). For the tensile response, Arena uses a hyperbolic function to describe the position of the cap with

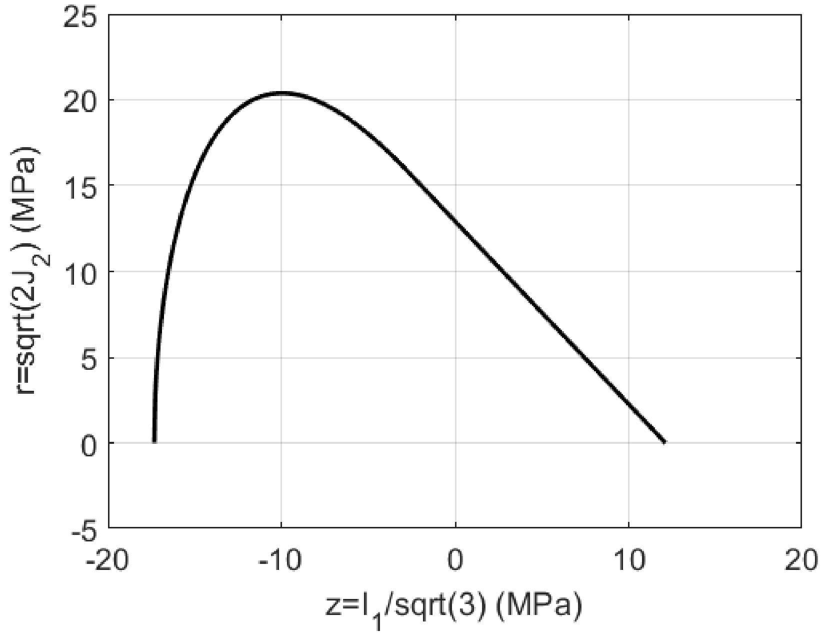


Figure 3.3: Example Initial Yield Surface for Berea Sandstone (Meridional Profile)

volumetric plastic strain, which is combined with the compressive crush curve to create a piecewise function (Brannon et al., 2014).

$$X = p_0 \left(\epsilon_v^p + 1\right)^{\frac{1}{p_0 p_1 p_3}} \tag{3.23}$$

A plot of the entire crush curve can be found on figure 3.4.

From the crush curve, the evolution equation for κ (equation 3.8) is found by first using the flow rule (equation 3.5) to find an expression for the volumetric plastic strain rate ($\dot{\epsilon}_v^p$).

$$\dot{\epsilon}_v^p = \text{tr}(\dot{\boldsymbol{\epsilon}}^p) = 3\dot{\gamma}\frac{\partial f}{\partial I_1} \tag{3.24}$$

The time derivative of $\dot{\epsilon}_v^p$ as a function of $\dot{\kappa}$ is

$$\dot{\epsilon}_v^p = \frac{\partial \epsilon_v^p}{\partial X}\frac{\partial X}{\partial \kappa}\dot{\kappa} \tag{3.25}$$

using the chain rule, and as a result, the evolution equation for κ is as follows (Foster et al., 2005).

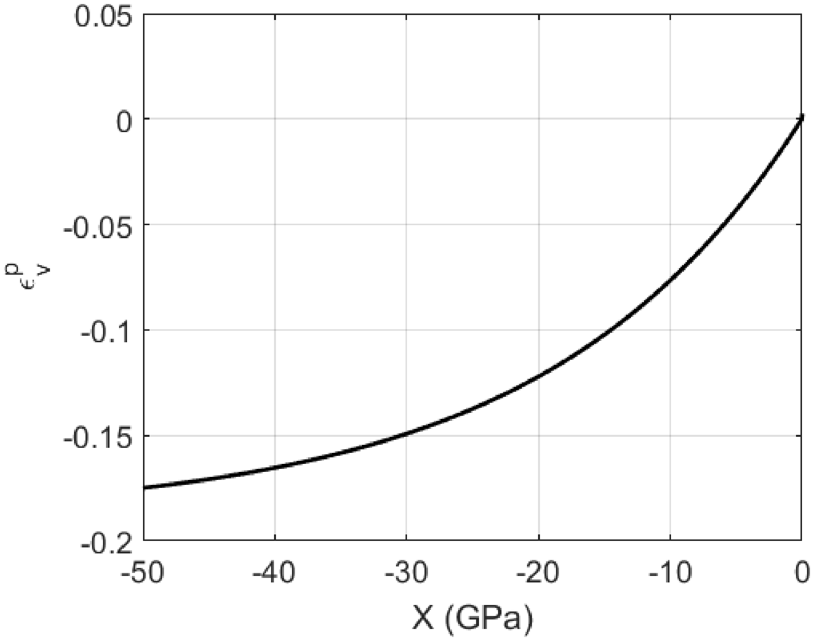


Figure 3.4: Crush Curve for Berea Sandstone

$$\dot{\kappa} = 3\dot{\gamma}\frac{\frac{\partial f}{\partial I_1}}{\frac{\partial \epsilon_v^p}{\partial X}\frac{\partial X}{\partial \kappa}}; \quad h^q(\sigma_{ij}, \kappa) = 3\frac{\frac{\partial f}{\partial I_1}}{\frac{\partial \epsilon_v^p}{\partial X}\frac{\partial X}{\partial \kappa}} \tag{3.26}$$

3.3 ABAQUS User Material Subroutine Implementation

Although all versions of Arena are implemented using a new geometrically based return algorithm, a more traditional implicit elastoplasticity implementation is presented here. The following sections summarize the required derivatives needed, the numerical time integration of the constitutive equations, and an overview of an implementation of Arena in an ABAQUS User Material Subroutine (UMAT).

3.3.1 Yield Function Derivatives

The implementation requires a number of derivatives starting with the gradient of the yield function with respect to the Cauchy stress.

$$\frac{\partial f}{\partial \sigma_{ij}} = \frac{\partial f}{\partial I_1}\frac{\partial I_1}{\partial \sigma_{ij}} + \frac{\partial f}{\partial J_2}\frac{\partial J_2}{\partial \sigma_{ij}} \tag{3.27}$$

The derivatives of the invariants with respect to the stress are well known to be $\frac{\partial I_1}{\partial \sigma_{ij}} = \delta_{ij}$ and $\frac{\partial J_2}{\partial \sigma_{ij}} = s_{ij}$, where s_{ij} is the deviatoric part of the Cauchy stress. Using the yield function for both the square root (equation 3.12) and square (equation 3.13) forms, the derivatives of f and f^{sq} with respect to I_1 and J_2 are as follows.

$$\frac{\partial f}{\partial I_1} = -\begin{cases} \frac{\partial F_f}{\partial I_1} & : I_1 \geq \kappa \\ \frac{\partial F_f}{\partial I_1} F_c + F_f \frac{\partial F_c}{\partial I_1} & : I_1 < \kappa \end{cases} \tag{3.28}$$

$$\frac{\partial f}{\partial J_2} = \frac{1}{2\sqrt{J_2}} \tag{3.29}$$

$$\frac{\partial f^{\mathrm{sq}}}{\partial I_1} = -\begin{cases} 2F_f \frac{\partial F_f}{\partial I_1} & : I_1 \geq \kappa \\ 2F_f \frac{\partial F_f}{\partial I_1} F_c^{\mathrm{sq}} + F_f^2 \frac{\partial (F_c^{\mathrm{sq}})}{\partial I_1} & : I_1 < \kappa \end{cases} \tag{3.30}$$

$$\frac{\partial f^{\mathrm{sq}}}{\partial J_2} = 1 \tag{3.31}$$

The derivative of the shear function (equation 3.15) with respect to I_1 is

$$\frac{\partial F_f}{\partial I_1} = -\beta \tag{3.32}$$

and the derivative of the cap function (equation 3.16) and square of the cap function with respect to I_1 are the following.

$$\frac{\partial F_c}{\partial I_1} = \begin{cases} 0 & : I_1 \geq \kappa \\ \frac{\frac{\kappa - I_1}{(\kappa - X)^2}}{\sqrt{1 - \left(\frac{\kappa - I_1}{\kappa - X}\right)^2}} & : I_1 < \kappa \end{cases} \tag{3.33}$$

$$\frac{\partial (F_c^{\mathrm{sq}})}{\partial I_1} = \begin{cases} 0 & : I_1 \geq \kappa \\ \frac{2(\kappa - I_1)}{(\kappa - X)^2} & : I_1 < \kappa \end{cases} \tag{3.34}$$

The calculation of the plastic multiplier also requires the derivative of the yield functions with respect to κ

$$\frac{\partial f}{\partial \kappa} = -\begin{cases} 0 & : I_1 \geq \kappa \\ F_f \frac{\partial F_c}{\partial \kappa} & : I_1 < \kappa \end{cases} \tag{3.35}$$

$$\frac{\partial f^{\mathrm{sq}}}{\partial \kappa} = -\begin{cases} 0 & : I_1 \geq \kappa \\ F_f^2 \frac{\partial (F_c^{\mathrm{sq}})}{\partial \kappa} & : I_1 < \kappa \end{cases} \tag{3.36}$$

where the derivative of the cap functions with respect to κ are the following equations.

$$\frac{\partial F_c}{\partial \kappa} = \begin{cases} 0 & : I_1 \geq \kappa \\ \frac{1}{(2)\sqrt{1-\left(\frac{\kappa-I_1}{\kappa-X}\right)^2}} \left((-2)\left(\frac{\kappa-I_1}{\kappa-X}\right)\right)\left(\frac{(1)(\kappa-X)-(\kappa-I_1)(1-\frac{\partial X}{\partial \kappa})}{(\kappa-X)^2}\right) & : I_1 < \kappa \end{cases} \tag{3.37}$$

$$\frac{\partial (F_c^{\mathrm{sq}})}{\partial \kappa} = \begin{cases} 0 & : I_1 \geq \kappa \\ -2\left(\frac{\kappa-I_1}{\kappa-X}\right)\left(\frac{(1)(\kappa-X)-(\kappa-I_1)(1-\frac{\partial X}{\partial \kappa})}{(\kappa-X)^2}\right) & : I_1 < \kappa \end{cases} \tag{3.38}$$

Using the derivative of the cap function with respect to I_1 (equation 3.33) and derivative of X with respect to κ (equation 3.42), $\frac{\partial F_c}{\partial \kappa}$ and $\frac{\partial (F_c^{\mathrm{sq}})}{\partial \kappa}$ simplify to the following.

$$\frac{\partial F_c}{\partial \kappa} = -\frac{\partial F_c}{\partial I_1}\left(\frac{(\kappa - X) + C_R \beta (\kappa - I_1)}{\kappa - X}\right) \tag{3.39}$$

$$\frac{\partial (F_c^{\mathrm{sq}})}{\partial \kappa} = -\frac{\partial (F_c^{\mathrm{sq}})}{\partial I_1}\left(\frac{(\kappa - X) + C_R \beta (\kappa - I_1)}{\kappa - X}\right) \tag{3.40}$$

Finally, two additional derivatives are required to calculate $h^q(\sigma_{ij}, \kappa)$ for the evolution of κ (equation 3.26). Using the compressive component of the total crush curve, the derivative of the volumetric plastic strain with respect to X is

$$\frac{\partial \epsilon_v^p}{\partial X} = p_1 p_3 \mathrm{e}^{-p_1(p_0 - X)} \tag{3.41}$$

and the derivative of the relationship between X and κ (equation 3.20) with respect to κ is the following.

$$\frac{\partial X}{\partial \kappa} = \frac{1}{C_R} \tag{3.42}$$

3.3.2 Numerical Time Integration

ABAQUS is a general purpose finite element analysis software that allows users to implement their own constitutive models through a number of pre-made FORTRAN templates. The basic user material (UMAT) is for use with implicit time integration schemes. The finite element program provides the user the solution to the balance of linear momentum by giving the current strain increment ($\Delta\epsilon_{ij} = (\epsilon_{ij})_{n+1} - (\epsilon_{ij})_n$ or `DSTRAN` in the notation used by ABAQUS) and the Cauchy stress at the previous time increment ($(\sigma_{ij})_n$ or `STRESS`). The user is then tasked with updating the stress, consistent tangent ($\frac{\Delta(\sigma_{ij})_{n+1}}{\Delta(\epsilon_{ij})_{n+1}}$ or `DDSDDE`), and any internal state variables (κ_{n+1} or `STATEV`) using the constitutive model.

Using a backward (implicit) Euler approximation, the evolution equation for the Cauchy stress (equation 3.6) can be written as

$$\Delta\sigma_{ij} = c^e_{ijkl}\left(\Delta\epsilon_{kl} - \Delta\gamma\left(\frac{\partial f}{\partial\sigma_{kl}}\right)_{n+1}\right) \tag{3.43}$$

and the evolution equation for the isotropic hardening ISV κ (equation 3.26) is the following.

$$\Delta\kappa = 3\Delta\gamma\left(\frac{\frac{\partial f}{\partial I_1}}{\frac{\partial\epsilon^p_v}{\partial X}\frac{\partial X}{\partial\kappa}}\right)_{n+1} \tag{3.44}$$

It's convenient to write the stress evolution equation in the following form

$$(\sigma_{ij})_{n+1} = (\sigma^{\text{tr}}_{ij})_{n+1} - \Delta\gamma c^e_{ijkl}\left(\frac{\partial f}{\partial\sigma_{kl}}\right)_{n+1} \tag{3.45}$$

where $(\sigma^{\text{tr}}_{ij})_{n+1}$ is the trial predictor stress defined as the following.

$$(\sigma^{\text{tr}}_{ij})_{n+1} = (\sigma_{ij})_n + c^e_{ijkl}\Delta\epsilon_{kl} \tag{3.46}$$

The trial stress is the stress state at the current time step if the material undergoes only elastic deformations. In other words, if the yield function evaluated at the trial stress is less than zero, the material only experiences elastic deformations and the trial stress is the Cauchy stress at the current time step. If the yield function evaluated at the trial stress is greater than zero, the material undergoes plastic deformations and the yield surface needs to harden via equation 3.44 and the stress needs to be updated via equation 3.45. Both of these processes are proportional to the plastic multiplier, whose discrete equation is as follows.

$$\Delta\gamma = \frac{\frac{\partial f}{\partial(\sigma_{ij})_{n+1}} c^{e}_{ijkl} \Delta\epsilon_{kl}}{\frac{\partial f}{\partial(\sigma_{ij})_{n+1}} c^{e}_{ijkl} \frac{\partial f}{\partial(\sigma_{kl})_{n+1}} - (\frac{\partial f}{\partial\kappa} h^{q})_{n+1}} \tag{3.47}$$

Also, in order to track the plastic volumetric strain, equation 3.24 also needs to be integrated (Foster et al., 2005).

$$(\epsilon^{p}_{v})_{n+1} = (\epsilon^{p}_{v})_{n} + 3\Delta\gamma \left(\frac{\partial f}{\partial I_1}\right)_{n+1} \tag{3.48}$$

This entire process is called the return mapping algorithm and is summarized below.

Return Mapping Algorithm:

```
(1) Calculate trial stress ($(\sigma^{\mathrm{tr}}_{ij})_{n+1}$): equation 3.46
(2) Evaluate the yield function $f((\sigma^{\mathrm{tr}}_{ij})_{n+1}, \kappa_n)$: equation 3.17
(3) Check for yielding: is $f > 0$?
    (a) If no, material behaves elastically:
        (i) Set $(\sigma_{ij})_{n+1} = (\sigma^{\mathrm{tr}}_{ij})_{n+1}$
        (ii) Update $\kappa_{n+1} = \kappa_n$
        (iii) Update $X_{n+1} = X_n$
        (iv) Exit
    (b) If yes, plastic flow:
        (i) Calculate the plastic multiplier ($\Delta\gamma$): equation 3.47
        (ii) Update the Cauchy stress ($(\sigma_{ij})_{n+1}$): equation 3.45
```

(iii) Calculate $\Delta\kappa$: equation 3.44

(iv) Update $\kappa_{n+1} = \kappa_n + \Delta\kappa$

(v) Calculate X_{n+1} using κ_{n+1}: equation 3.20

(vi) Exit

For this implementation, the continuum elastoplastic tangent is being used as an approximation of the full material consistent tangent the discrete version of which is the following.

$$c^{ep}_{ijkl} = c^{e}_{ijkl} - \frac{1}{\chi}\left(c^{e}_{ijab}\frac{\partial f}{\partial(\sigma_{ab})_{n+1}}\right)\left(\frac{\partial f}{\partial(\sigma_{cd})_{n+1}}c^{e}_{cdkl}\right) \tag{3.49}$$

3.3.3 Model Verification and Testing

A series of models were developed to test the ABAQUS UMAT implementation of Arena, which can be found in appendix A. These models were taken from verification tests developed by the University of Utah (Brannon et al. (2014)) to test their implementation of Arena in the material point method code Uintah. In these tests, a single material point is placed inside a unit cube background grid (5x5x5) in what is analogous to the finite element method's single element test. Therefore, for each of these verification tests, a single 1x1x1 hexahedral linear finite element was created in ABAQUS, and then the various boundary and loading conditions necessary for the test were applied to it. All of the tests were done under pseudo-static conditions with the loads ramped linearly over the entire step.

The first verification test performed was a uniaxial strain test. The material properties, as given by the University of Utah (where material the parameters were taken from wasn't specified), were set to $K = 10$ kPa (bulk modulus), $\mu = 3.75$ kPa (shear modulus), $\beta = 0.0577$, $I_1^{\max} = 612.4$ Pa, $C_R = 0.5$, $p_0 = -1837.1$ Pa, $p_1 = 6.667\text{x}10^{-4}$ Pa^{-1}, and $p_3 = 0.5$ m/m. To create the uniaxial conditions, the two faces of the element in both the x and y directions as well as the bottom of the element, the side in the $-z$ direction, were all fixed in space, and a displacement boundary condition was assigned to the top, the side in the $+z$ direction, of the element. The displacement of the model was defined in Uintah through the used of a "F-table", which specifies the nine components of the deformation gradient, an angle of rotation (θ), and the axis of rotation ($\boldsymbol{a}$) over the time of the simulation. The F-table applied to this simulation can be found in table 3.1. As can be seen, no deformation or rotation is specified except in the $-z$, or F_{33}, direction. Therefore, the displacement boundary condition applied to the top of the ABAQUS model was 0.09 in the $-z$ direction (Brannon et al., 2014).

Table 3.1: F-table for the Arena Uniaxial Strain Verification Test

Time	F_{11}	F_{12}	F_{13}	F_{21}	F_{22}	F_{23}	F_{31}	F_{32}	F_{33}	θ	a_1	a_2	a_3
0.0	1	0	0	0	1	0	0	0	1	0	1	0	0
1.0	1	0	0	0	1	0	0	0	0.91	0	1	0	0

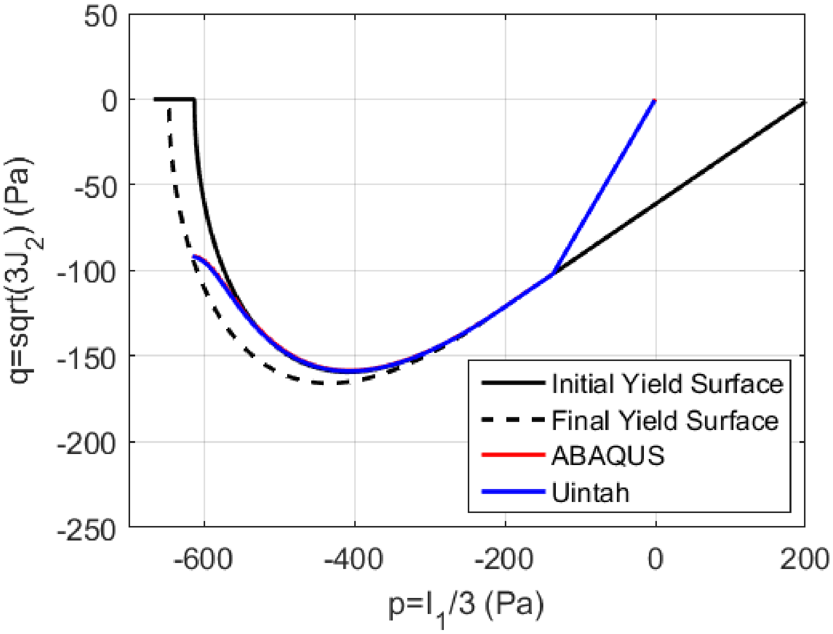


Figure 3.5: Arena UMAT Verification Results: Uniaxial Strain

A comparison between the ABAQUS UMAT and Uintah implementations of Arena for the uniaxial strain test can be found on figure 3.5. As is apparent from the figure, the ABAQUS model predicts the same stress path as the Uintah model, and assuming that the Uintah version is “truth”, this result helps verify that the ABAQUS model is working properly. Both models load elastically until the yield surface is reached. Because the models are not allowed to shear harden or soften, the stress path then follows the yield surface until it reaches the compactive side at which point the material begins to isotropically harden under the increased pressure (in compression).

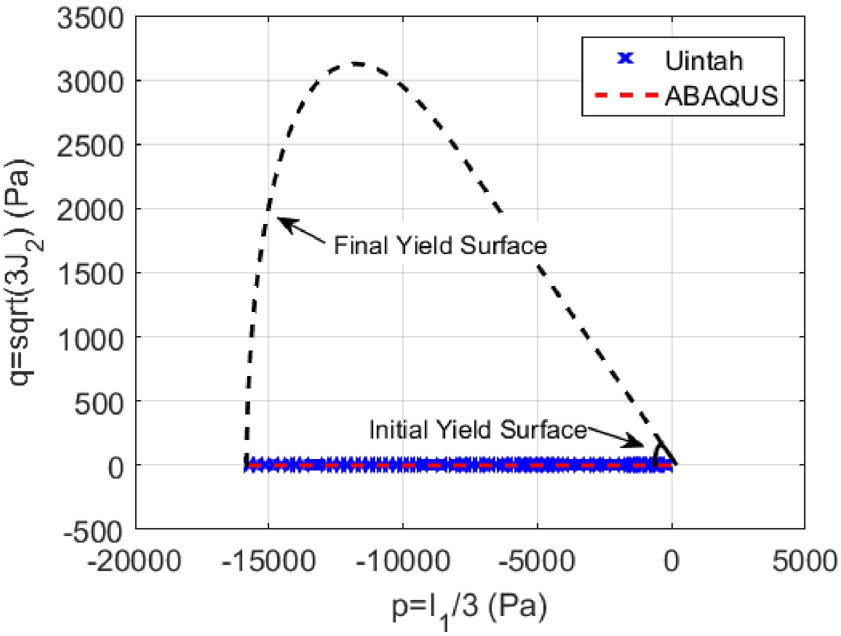


Figure 3.6: Arena UMAT Verification Results: Isotropic Compression

Second, an isotropic compression test was run primarily to test the capabilities of the cap hardening feature of Arena through the crush curve. The same material properties were used as in the uniaxial strain test, except the shear modulus was increased to μ = 15 kPa. To simulate isotropic compression, the $-x$, $-y$, and $-z$ faces of the single cubic element were restricted such that no displacement was allowed normal to the plane of the face, and displacement boundaries of -0.5 m were applied to the $+x$, $+y$, and $+z$ faces corresponding to the F-table (table 3.2) used in the Uintah simulation.

A stress path comparison between Uintah and ABAQUS can be found on figure 3.6. The material undergoes massive plastic deformation and hardens significantly, yet both the ABAQUS

Table 3.2: F-table for the Arena Isotropic Compression Verification Test

Time	F_{11}	F_{12}	F_{13}	F_{21}	F_{22}	F_{23}	F_{31}	F_{32}	F_{33}	θ	a_1	a_2	a_3
0.0	1	0	0	0	1	0	0	0	1	0	1	0	0
1.0	0.5	0	0	0	0.5	0	0	0	0.5	0	1	0	0

UMAT and Uintah simulations predict the same stress path. A closer examination of the isotropic hardening of the material can be found on figure 3.7. As stated earlier, $\epsilon_v^p + p_3$ is approximately equal to the porosity of the material. In this simulation, the material deforms elastically until $I_1 = p_0 = -1837.1$ Pa when the yield surface is reached in compression. All further negative hydrostatic deformation causes the material to deform plastically and harden following the crush curve nearly to the point of full void closure. As with the stress path plot, both the ABAQUS UMAT and Uintah simulations predict the same hardening through void closure with I_1.

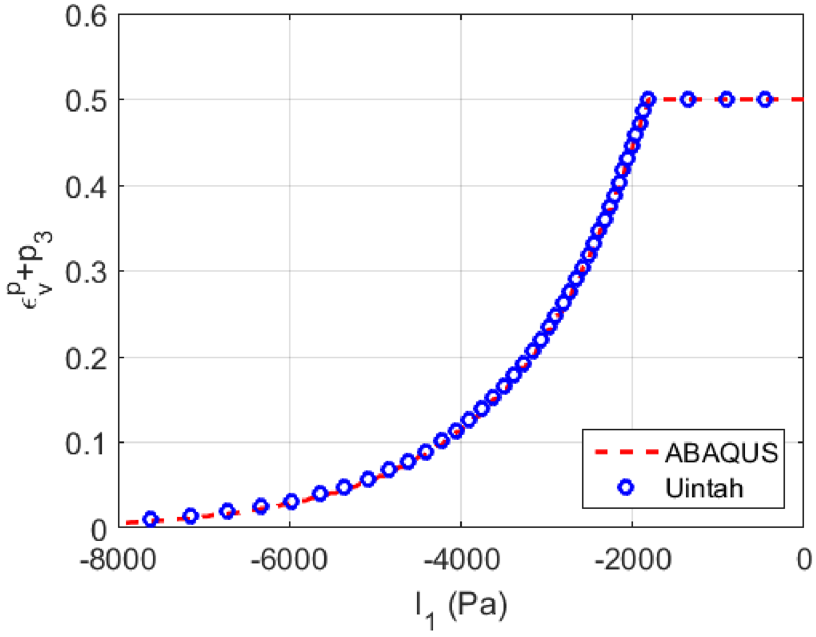


Figure 3.7: Arena UMAT Verification Results: Crush Curve

3.3.4 ABAQUS Explicit User Material Subroutine Implementation

A second Arena implementation was also developed for use in ABAQUS Explicit in an ABAQUS Explicit User Material Subroutine (VUMAT) (see appendix B). As some of the applications of the ABAQUS Arena formulation were modeling highly dynamic experiments, it was determined that these experiments would be better modeled using ABAQUS Explicit. Although explicit formulations suffer from conditional stability requiring small timesteps, implicit formulations can suffer from convergence issues for problems with high strain rates. Regardless, the implementation of Arena in a VUMAT is nearly the same as a UMAT. Once again, ABAQUS provides the user with the current strain increment ($\Delta\epsilon_{ij}$ or `STRAININC` in the notation used by ABAQUS Explicit) and the Cauchy stress at the previous time increment ($(\sigma_{ij})_n$ or `STRESSOLD`). The user again updates the stress (`STRESSNEW`) and internal state variables (`STATENEW`), but the consistent tangent is no longer needed.

3.4 Parameter Calibration of the Arena Geomaterial Model

The primary use of the ABAQUS implementations of Arena were to assist in calibrating the material parameters to experimental data that was generated for the ONR Soil Blast MURI Project. Two “natural” soils were chosen for the project: a low plasticity clay (CL) called Boulder Clay and a poorly graded sand (SP) called Colorado Mason Sand (CL and SP according to the United Soil Classification System (ASTM-D2487, 1985)). Both of these soils as well as various soil mixtures underwent a wide range of geotechnical engineering tests to classify their behaviors under drained and undrained conditions, in saturated and unsaturated states, and subjected to a wide range of loading rates from quasi-static to impulse loading from explosives. The implicit ABAQUS implementation was used to calibrate the lower strain rate data as the code has stability and speed advantages over explicit methods, and the explicit implementation was used to calibrate the high strain rate loading conditions.

3.4.1 Boulder Clay

Boulder Clay is a low plasticity clay that was collected from a construction site on the University of Colorado Boulder campus. Although the clay was air-dried, crushed, and processed post collection, it more closely approximates a "natural" soil than is usually used in most geotechnical engineering experimental research. A series of standard geotechnical engineering tests were performed on the clay - hydrometer, specific gravity, Atterberg limits, oedometer, and Proctor compaction tests among others - the results from which are found summarized in table 3.3 (Mun and McCartney, 2015).

Table 3.3: Geotechnical Properties of Boulder Clay (Mun and McCartney, 2015)

Material Property	**Value**
D_{10}	<1.7x10^{-4}
D_{30}	<0.001 mm
D_{50}	0.001 mm
% Fines	100%
Liquid Limit (LL)	40
Plastic Limit (PL)	17.5
Plasticity Index (PI)	22
Activity (A)	0.75
Max Dry Unit Weight ($\gamma_d^{\max}$)	17.3 kN/m^3
Optimum Water Content (w_{opt})	17.5%
Compression Index (C_c)	0.233
Recompression Index (C_r)	0.041
Preconsolidation Stress (σ_c')	110 kPa
Drained Friction Angle (ϕ')	33°

As suggested by the Arena User's Manual (Brannon et al., 2014), the first few parameters that should be calibrated are the elastic bulk modulus and crush curve ideally from hydrostatic compression tests that a) include a number of elastic unloading curves and b) compress the material until complete void closure has occurred. Therefore, the first data set analyzed was the high pressure isotropic compression data for drained, saturated Boulder Clay conducted by Drs. Woongju Mun and John McCartney at the University of Colorado Boulder (Mun and McCartney, 2015). A picture of Dr. Woongju Mun and the high pressure isotropic cell he designed with a soil sample being prepared for testing can be found on figure 3.8. Figure 3.11 shows the applied isotropic pressure versus the total volumetric strain of the tested specimen of Boulder Clay. As is shown, the specimen was loaded to a pressure of approximately 35 MPa before being unloaded and then reloaded to the maximum pressure of the test apparatus (160 MPa). The sample was then once again unloaded.

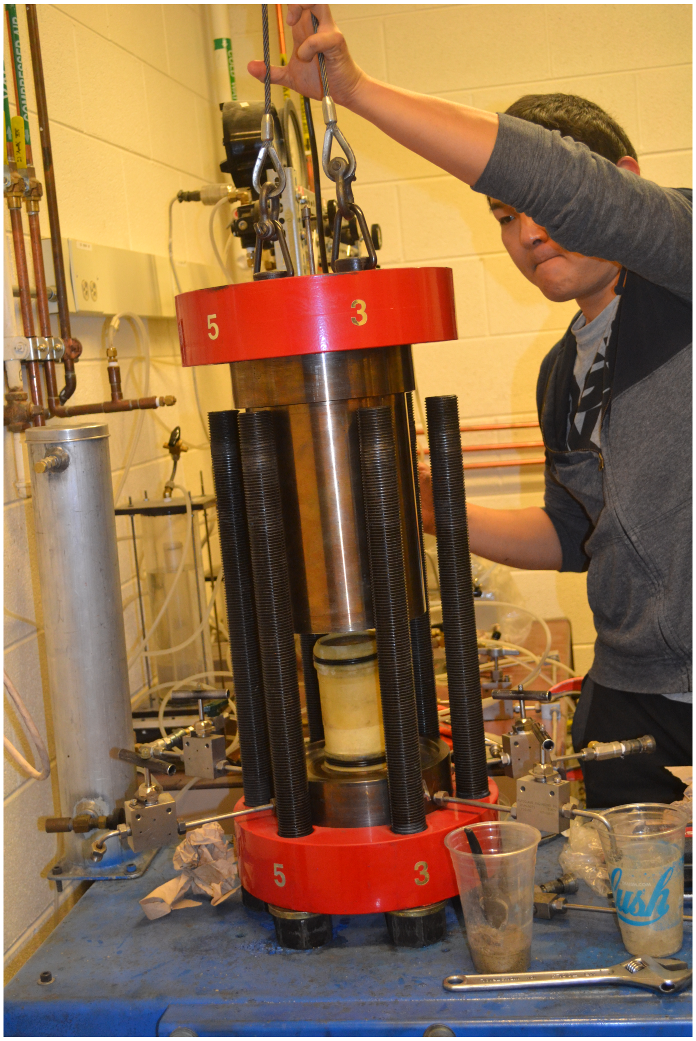


Figure 3.8: High Pressure Isotropic Compression Test Apparatus and Soil Sample

To model isotropic compression of the soil sample, a quarter symmetry FEM model was created in ABAQUS using the Arena UMAT as the soil's constitutive model. Screenshots of the model's geometry and boundary conditions and finite element mesh can be found on figures 3.9

and 3.10. A pressure boundary condition matching the loading conditions (see figure 3.11) was applied to the top ($+z$ face) and cylindrical side of the specimen's geometry, and the displacement of bottom ($-z$ face) and two flat sides ($-x$ and $-y$ faces) were restricted in the $-z$, $-x$, and $-y$ directions, respectively, to achieve quarter symmetry. The selected mesh was comprised of trilinear hexahedral elements with reduced integration to help decrease the overall time of the simulation.

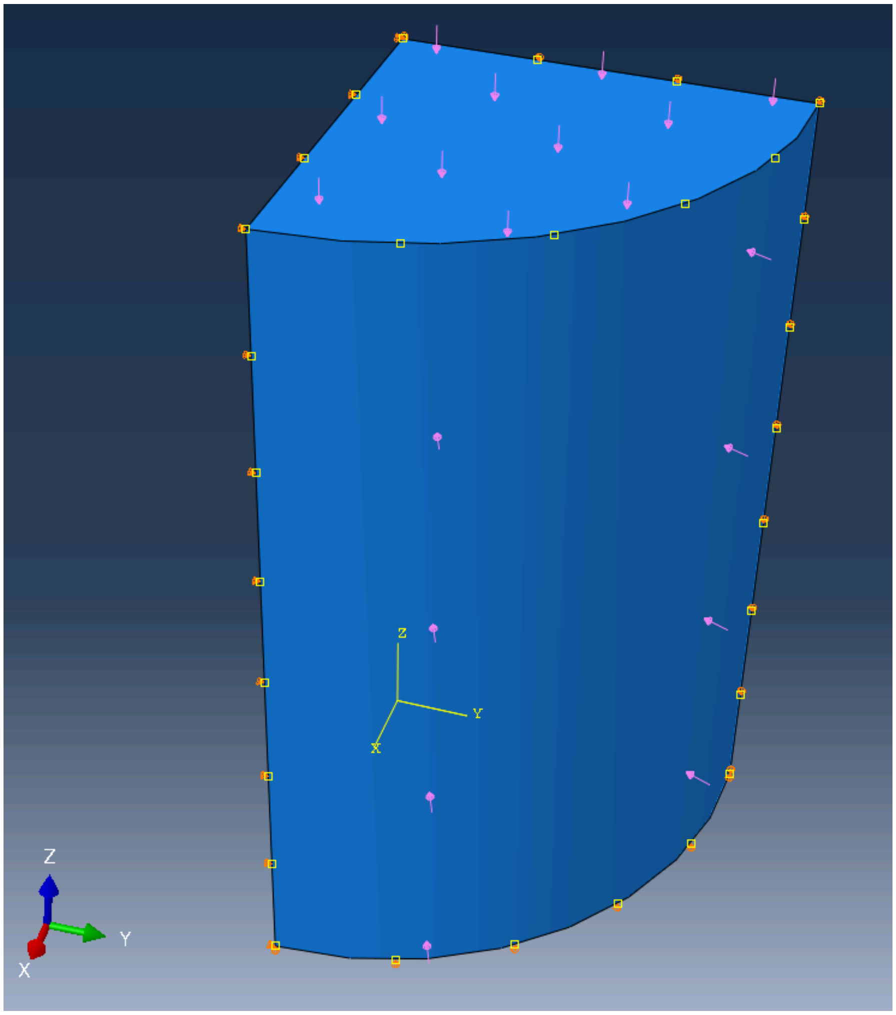


Figure 3.9: High Pressure Isotropic Test Simulation Geometry and Boundary Conditions

The blue lines on figure 3.11 show an initial fitting of the high pressure isotropic compression data using the following values for the four variables (K, p_0, p_1, and p_3).

- The elastic bulk modulus (K) of the material, as seen by the linear unloading of the model,

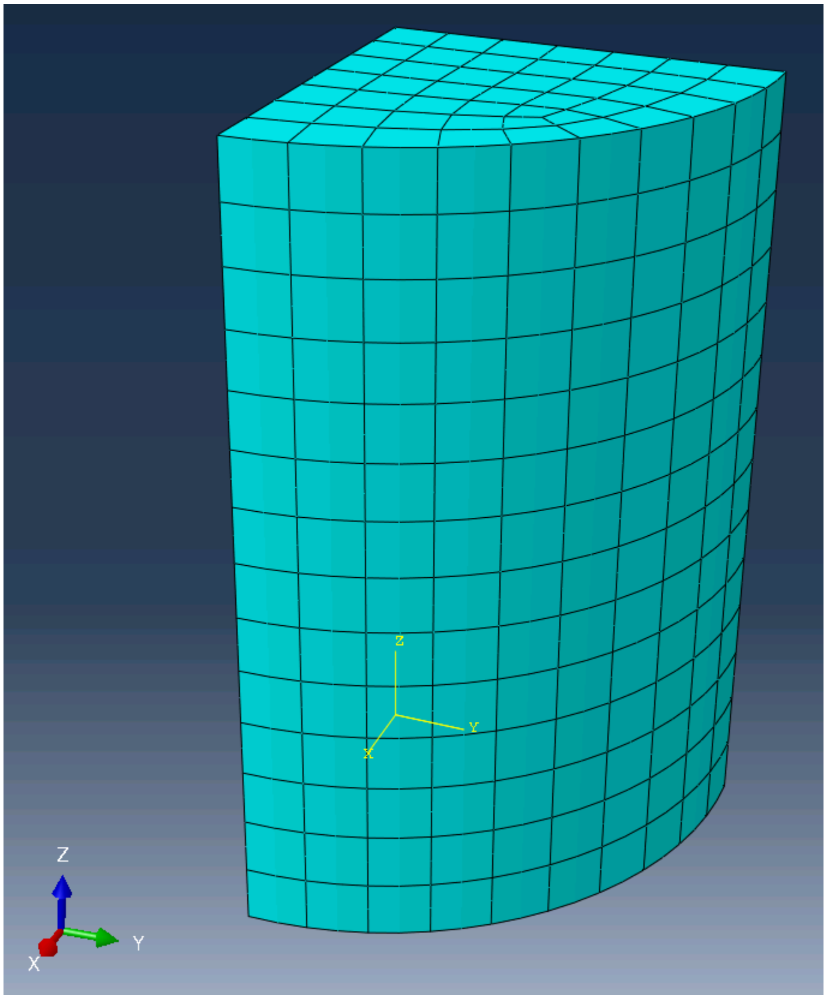


Figure 3.10: High Pressure Isotropic Test Simulation Mesh

was set to 2 GPa.

- p_0 is defined as the initial position of the cap along the hydrostatic axis. Given a preconsolidation pressure (σ'_c) of 110 kPa, p_0 was set to three times this value at -330 kPa or -0.33 MPa (negative in compression).

- p_1 governs the initial slope of the curve and was the parameter that was varied to achieve the closest fit. It was found to be approximately 0.01 MPa^{-1}

- p_3 is the maximum achievable plastic volumetric strain, which according to the data, is approximately 0.26 m/m. In order to achieve a better fit, p_3 was set to 0.25 m/m.

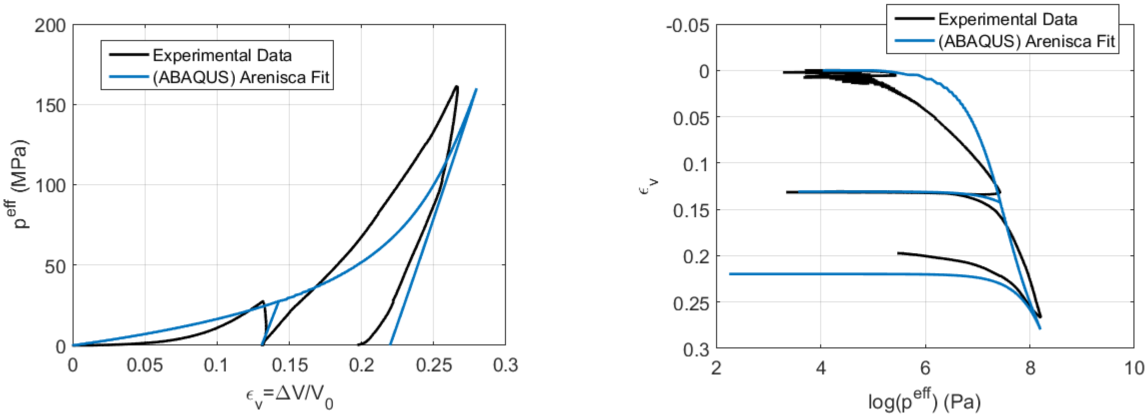


Figure 3.11: High Pressure Isotropic Compression of Drained, Saturated Boulder Clay Calibration Results

On figure 3.11, there is some evidence to suggest that the Boulder Clay exhibits some “creep”. At this time, however, it is difficult to discern if the apparent creep in these results is the result of consolidation (dissipation of excess pore water pressure causing solid skeleton volumetric compaction) or actual deformation of the solid skeleton’s molecular structure under constant pressure. Further preliminary testing has been completed by Dr. Mun to quantify the creep rate of the Boulder Clay, but the results are still being analyzed. Also, in the results for the test presented here, there is additional evidence suggesting the loading rates were too quick to allow the dissipation of all excess pore water pressure. Regardless, these confounding issues and the linear elasticity of the Arena implementation explain the differences between the experimental and simulation unloading curves.

3.4.2 Dry Colorado Mason Sand

As the hierarchical multiscale modeling effort (see chapters 4, 5, and 6) primarily focused on using the discrete element method (DEM) to replace continuum constitutive models, the bulk of the parameter calibration effort was focused on modeling dry Colorado Mason Sand. Colorado Mason Sand is a poorly graded, angular sand purchased from a quarry, Colorado Materials, located in Longmont, CO. Table 3.4 contains a list of standard geotechnical properties of the sand from Mun (2015). The ultimate goal of the Soil Blast MURI Project is to be able model saturated or partially saturated sand-clay mixtures, but for testing of the mutltiscale algorithms, DEM was used to model the dry sand case. To calibrate the Arena soil model for the dry Colorado Mason Sand, four geotechnical tests were modeled in ABAQUS: the high pressure isotropic compression test, the split Hopkinson pressure bar, and two triaxial tests. One of the triaxial tests was normal geotechnical lab scale, and the second was a small sample that was tested inside a synchrotron microscope.

Table 3.4: Geotechnical Properties of Colorado Mason Sand (Mun, 2015)

Material Property	**Value**
D_{10}	0.15 mm
D_{30}	0.28 mm
D_{50}	0.50 mm
Coefficient of Uniformity (C_u)	3.33
Coefficient of Gradation (C_c)	1.05
Specific Gravity (G_s)	2.62
Minimum Void Ratio (e_{min})	0.50
Maximum Density (ρ_{max})	1.74 kg/m^3
Maximum Void Ratio (e_{max})	0.78
Minimum Density (ρ_{min})	1.47 kg/m^3

Similar to the drained, saturated Boulder Clay, high pressure isotropic compression test data conducted by Drs. Woongju Mun and John McCartney on the Colorado Mason Sand were also used to calibrate the Arena soil model (Mun, 2015). Using the same loading pattern, simulation geometry, boundary conditions, and mesh as with the clay, the results of the test and calibration of the crush curve and elastic bulk modulus can be seen on figure 3.12. It was found that $K = 0.5$ GPa, $p_0 = -900$ kPa (preconsolidation pressure), $p_1 = 0.005$ MPa^{-1}, and $p_3 = 0.42$ m/m for the dry Colorado Mason Sand.

Using the calibrated crush curve and elastic bulk modulus, Arena's shear surface was then probed through the results of a split Hopkinson pressure bar test and the two triaxial compression tests. The split Hopkinson pressure bar (SHPB) test (Luo et al., 2011) is a high strain rate test where a cylindrical sample of soil is placed between two steel bars, and then one of the bars is rapidly accelerated into the sample (see figure 3.13 for a schematic). Stress and strain in the soil

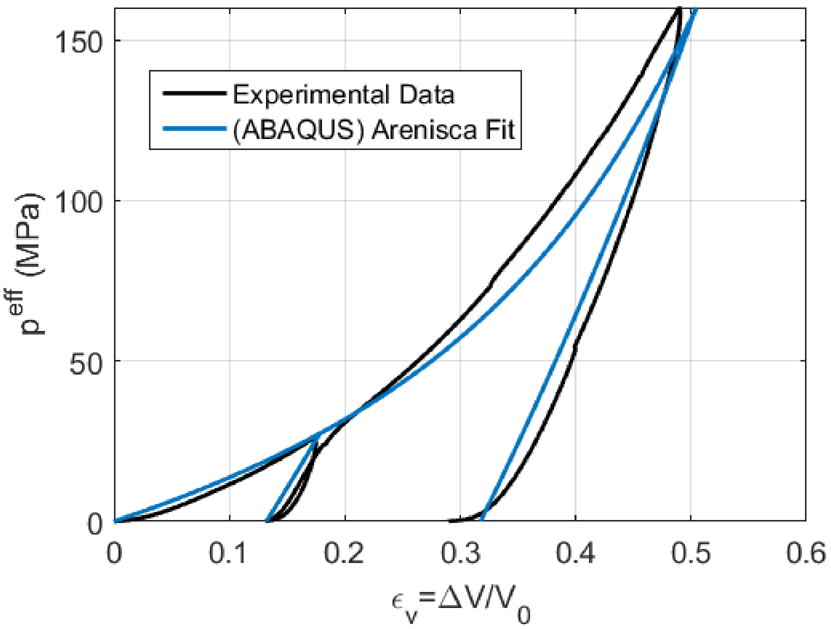


Figure 3.12: High Pressure Isotropic Compression Tests on Dry Colorado Mason Sand Calibration Results

specimen is extrapolated via strain gauges on the steel bars that measure the various generated pressure waves.

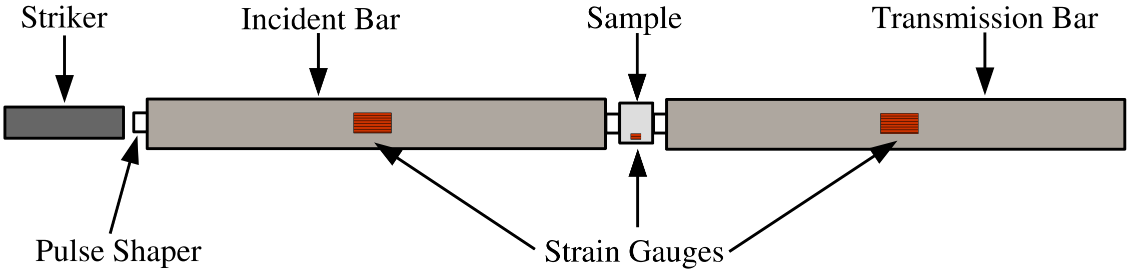


Figure 3.13: Split Hopkinson Pressure Bar Experimental Schematic Adapted from Luo et al. (2011)

Similar to the high pressure isotropic test simulations, a quarter symmetry model of the split Hopkinson pressure bar test was developed. The geometry and boundary conditions of the simulation can be found on figure 3.14 and finite element mesh on figure 3.15. In order to mimic the uniaxial strain nature of the experiment, the $-z$ face was fixed in all directions and the $-x$ and $-y$ faces were fixed in the x and y directions, only allowing for displacement of the model in the z

direction. To restrict the movement of the cylindrical face, a discrete rigid shell was generated to model the steel tube confining the soil specimen in the experiment. It was assumed that although the steel in the experiment could slightly deform, this deformation was negligible due to the relative stiffness of the steel. Contact between the rigid shell and soil specimen was modeled as frictionless in the tangential direction and a “hard” contact model (no penetration) in the normal direction. Trilinear hexahedral elements with reduced integration were again used for the finite element mesh.

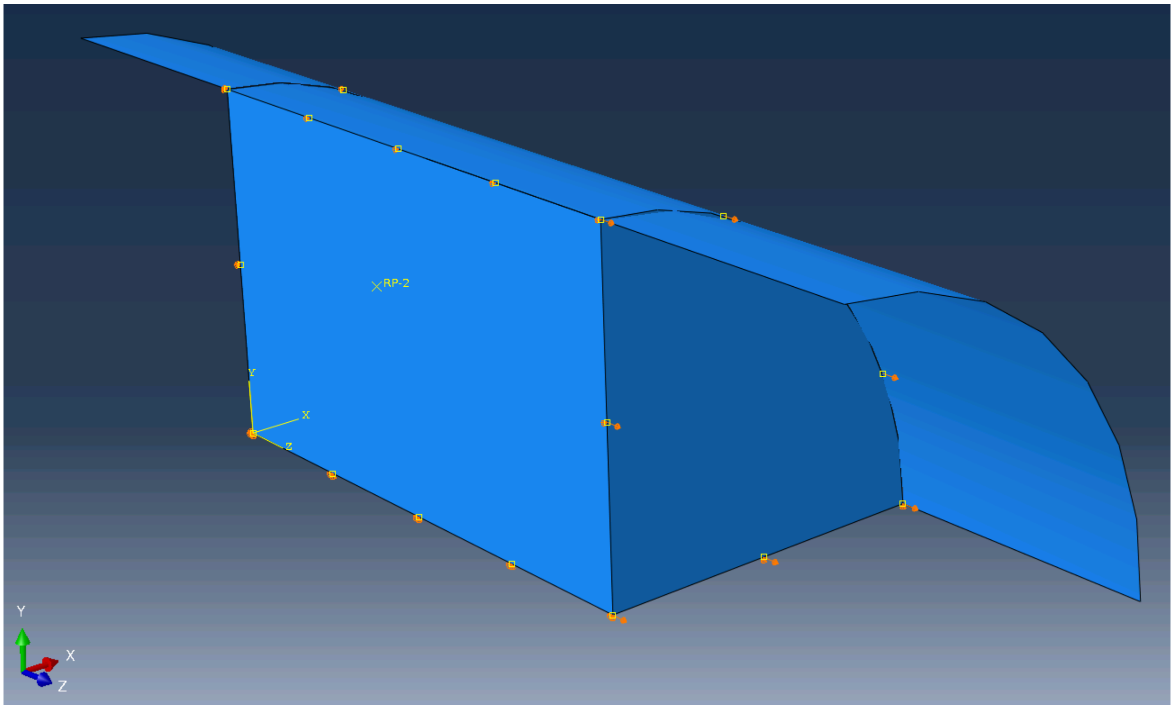


Figure 3.14: Split Hopkinson Pressure Bar Simulation Geometry and Boundary Conditions

One major assumption of SHPB testing is that the strain rate is constant in the soil sample. Figure 3.16 compares the strain rate from the experiment with that of the strain rate applied to an ABAQUS simulation of the SHPB experiment using the Arena UMAT. The strain rate of the experimental data is constant from approximately 0.05 to 0.75 ms. For simplicity, the strain rate was made to be constant over 0 to 0.1 ms of the ABAQUS simulation, so it was expected that the simulation would over predict the final strain in the sand. A more thorough discussion of the SHPB experiment can be found in section 4.3.

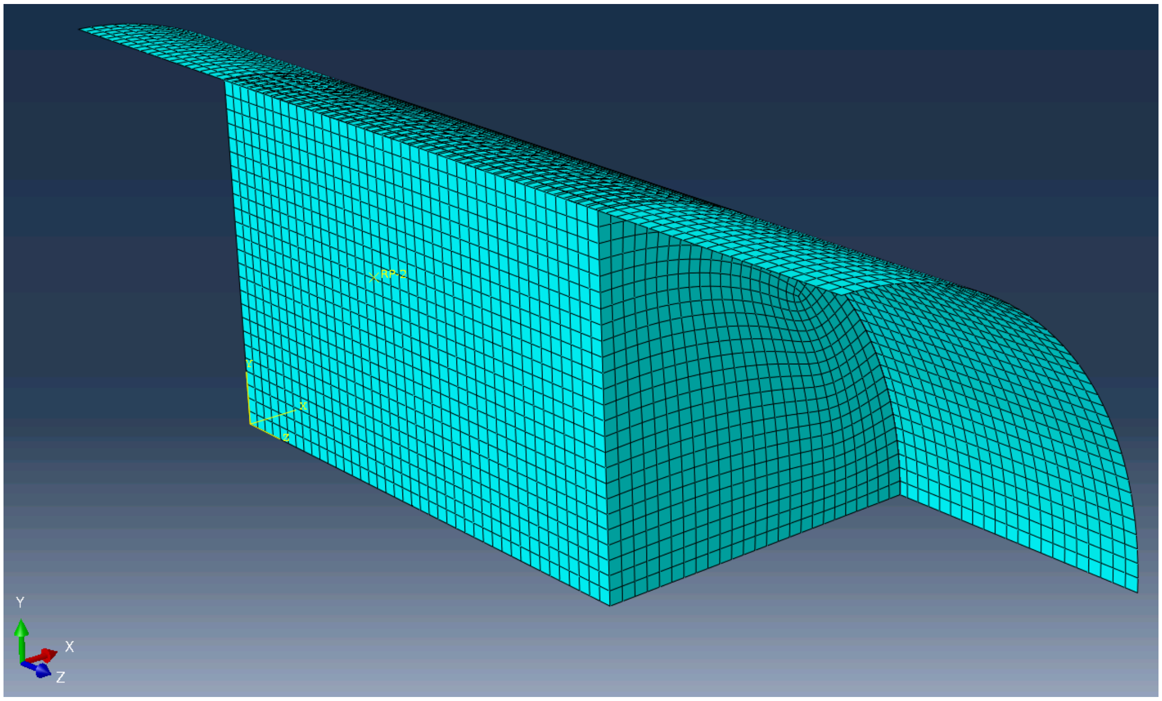


Figure 3.15: Split Hopkinson Pressure Bar Simulation Mesh

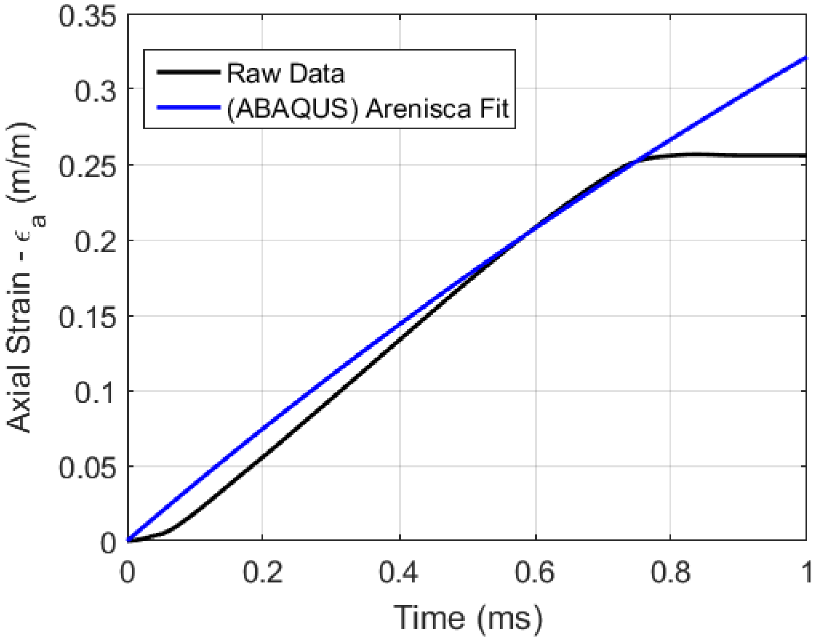


Figure 3.16: Split Hopkinson Pressure Bar Tests on Dry Colorado Mason Sand Axial Strain Comparison

Arena's linear Drucker-Prager shear surface is calibrated through the tensile limit ($I_1^{\max}$), the angle of the surface with respect to the hydrostat (β), and the cap ellipticity parameter (C_R). For the cohesionless, dry Colorado Mason Sand, it was assumed that $I_1^{\max}$ was zero; although, it was set to 10 Pa in order to avoid numerical issues from dividing by zero. The friction angle of the sand was found to be $\phi = 34.5°$ from the suite of standard geotechnical tests performed on the material (Mun, 2015). Therefore, the angle was set to $\beta = \tan(\phi) = 0.6873$. The cap ellipticity parameter, which essentially controls how much of the yield surface is dominated by the elliptical cap, was found to be approximately 0.5.

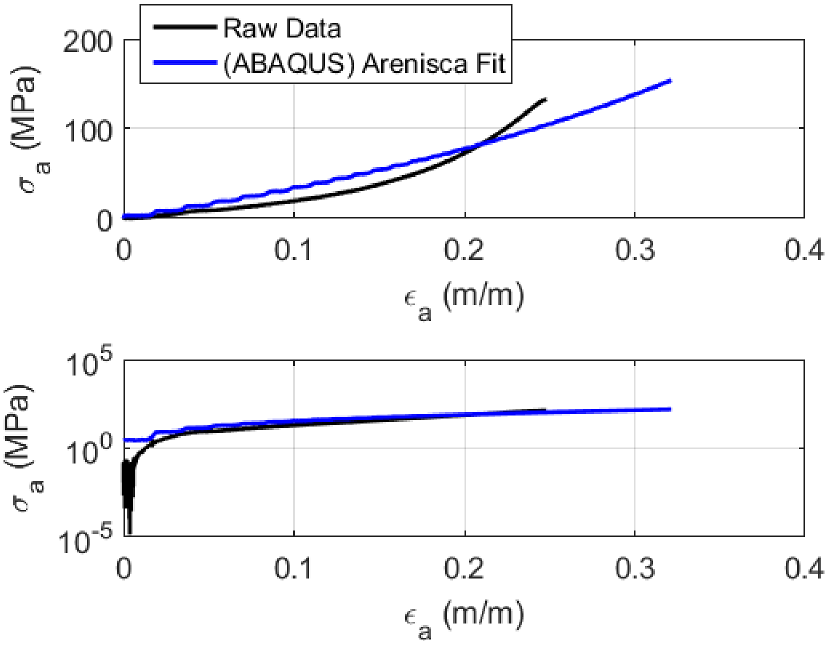


Figure 3.17: Split Hopkinson Pressure Bar Tests on Dry Colorado Mason Sand Calibration Results

Due to the high strain rate nature of the SHPB experiment, the ABAQUS UMAT version of Arena suffered from convergence issues. To counteract these issues, a second, explicit version of the user material subroutine (VUMAT) with the Arena soil model was developed, and the SHPB experiment was re-modeled using ABAQUS Explicit. The explicit version suffered from mesh stability issues (see figure 3.18 to see example of elements collapsing), but after experimenting with the characteristic element size (mesh refinement) and time integration discretization parameters, a

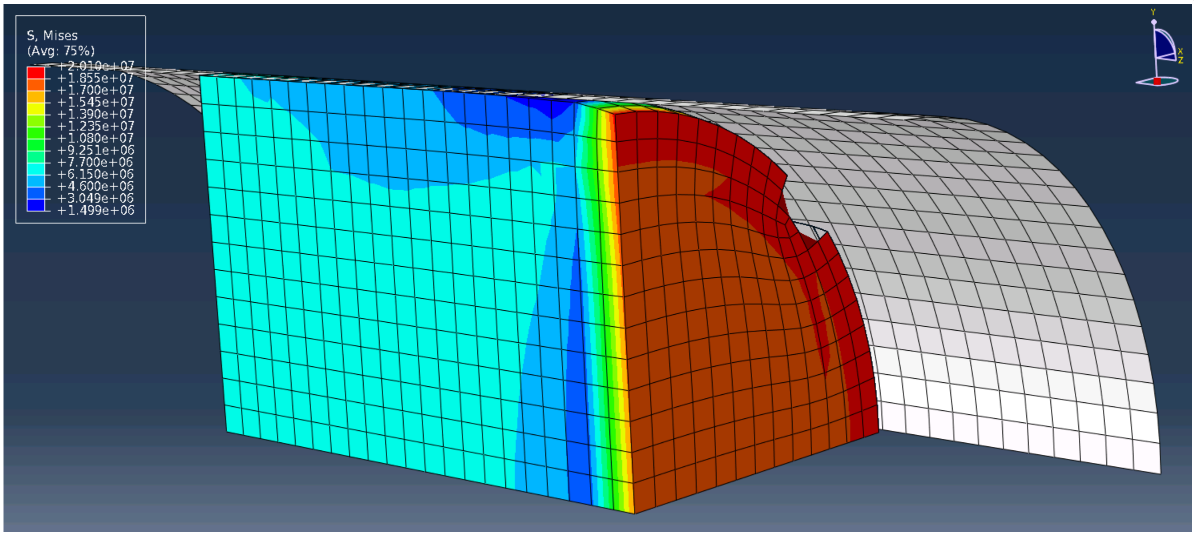


Figure 3.18: Split Hopkinson Pressure Bar Simulation Mesh Stability Issues

SHPB simulation with the Arena VUMAT was eventually completed, the results from which can be found on figure 3.17. A contour plot of the final axial stress (S33) can be found on figure 3.19. As expected, the ABAQUS simulation over estimated the axial strain, but overall, the crush curve and elastic bulk modulus fits predicted the SHPB data reasonably well.

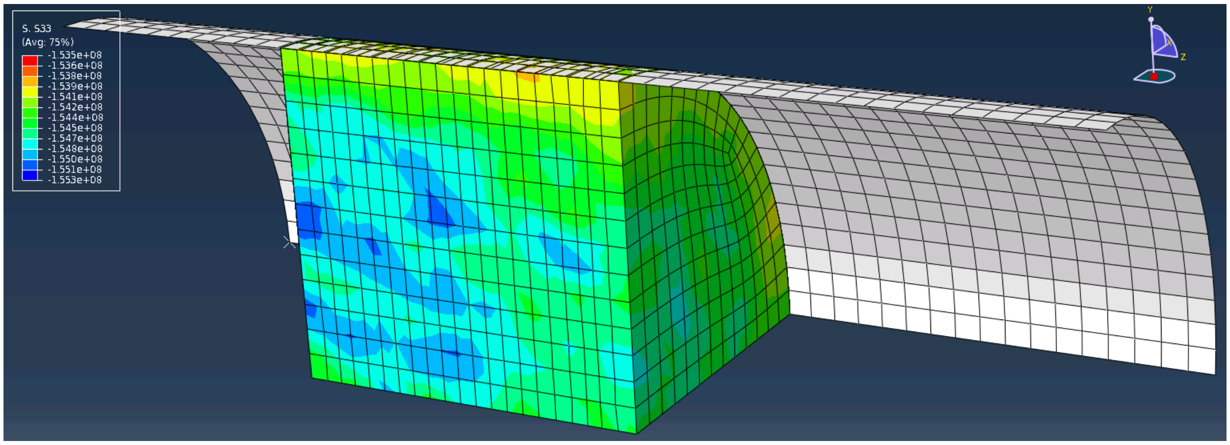


Figure 3.19: Split Hopkinson Pressure Bar Simulation Axial Stress Contours

As mentioned before, one of the goals of pulse shaping in SHPB experiments is to generate a constant strain rate throughout the specimen. The pulse is also applied such that the specimen experiences said constant strain rate almost immediately. As a result, the ABAQUS VUMAT simulation of the SHPB experiment was rerun, but with an initial velocity applied to generate the initial constant strain rate. A contour plot of the final axial stress (S33) can be found on figure 3.20 as well as axial strain time histories and stress versus strain comparisons on figures 3.21 and 3.22, respectively. Although the results of the SHPB simulations with and without the initialized velocity were the same, there were minor mesh stability improvements by initializing the velocity.

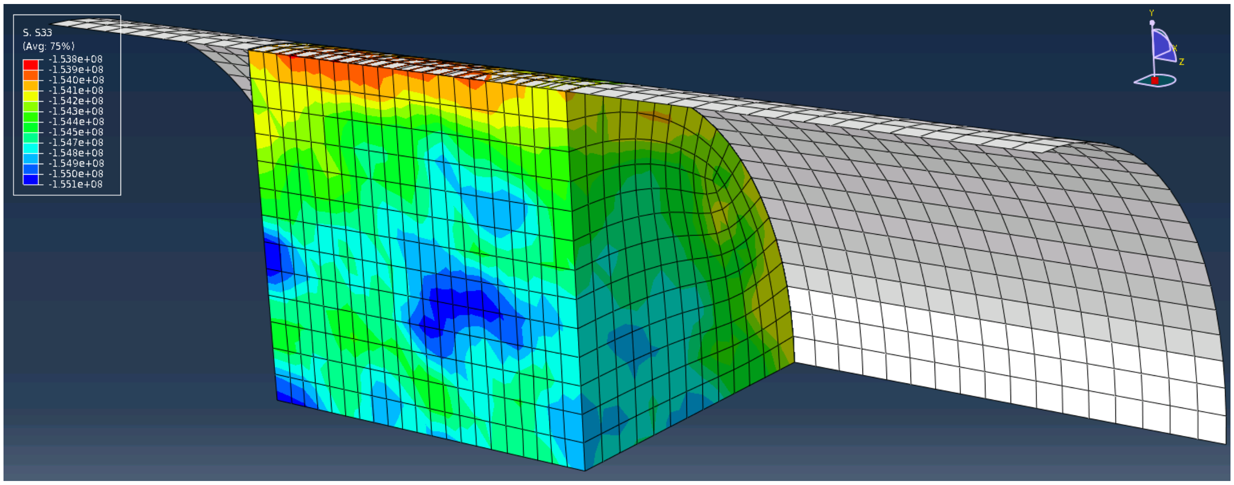


Figure 3.20: Split Hopkinson Pressure Bar Simulation Axial Stress Contours with Initialized Velocity

With the model now calibrated for a large stress and high strain rate problem, the ABAQUS UMAT version of Arena was used to model two triaxial compression tests. The first, conducted by Jenna Svoboda at the University of Colorado Boulder (Svoboda, 2013), was a standard geotechnical triaxial compression test (see figure 3.23). The second, hereby referenced as the “small” triaxial compression test, was a modified version that was designed by Drs. Andrew Druckrey and Khalid Alshibli of the University of Tennessee, Knoxville, to run inside a synchrotron electron microscope located at the Advanced Photon Source, Argonne National Laboratory (see figure 3.24). This smaller test was compressed in loading stages and scanned by the microscope in between these stages to create synchrotron micro-computed tomography (SMT) images of the sample while shearing.

As with the previous models, quarter symmetry models of the triaxial tests were created in ABAQUS. Both models were the same except with different dimensions, and a screenshot of the

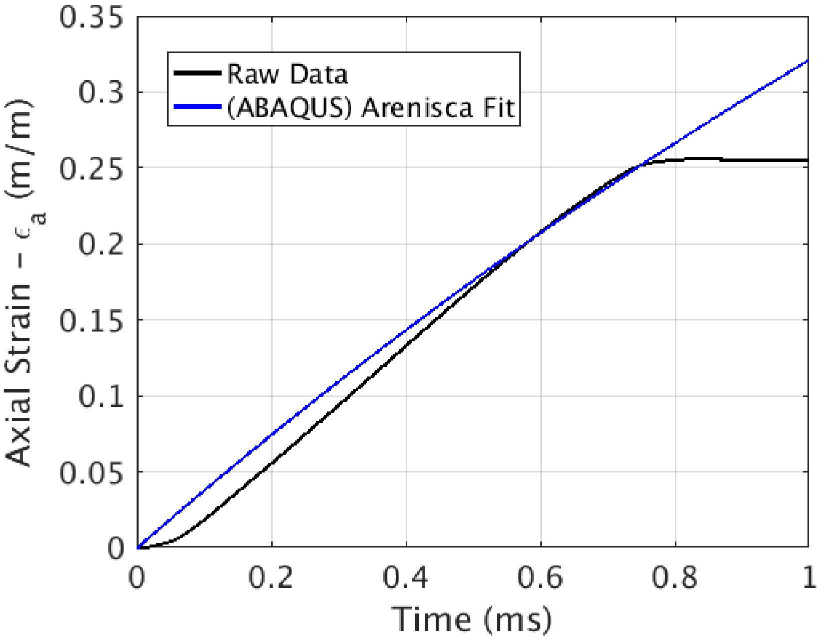


Figure 3.21: Split Hopkinson Pressure Bar Tests on Dry Colorado Mason Sand Axial Strain Comparison with Initialized Velocity

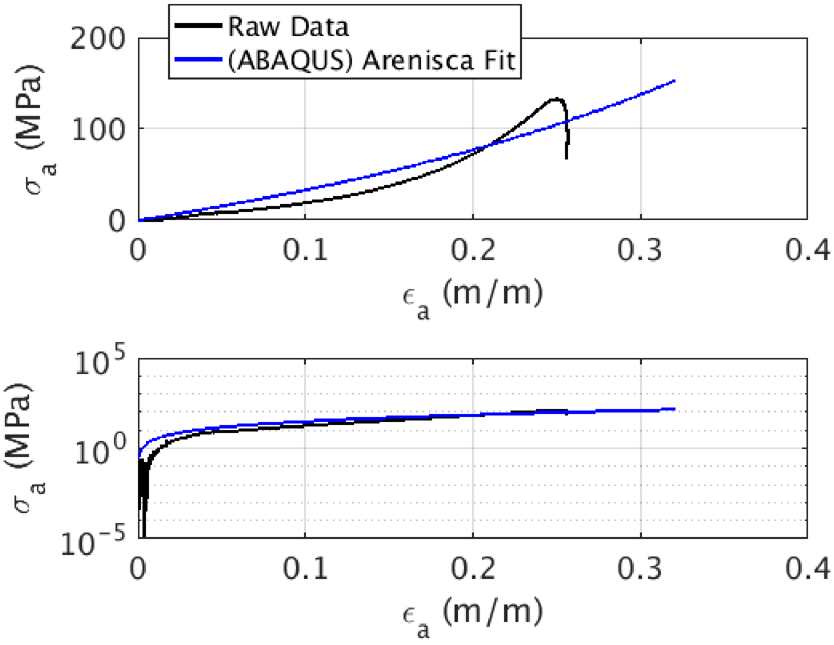


Figure 3.22: Split Hopkinson Pressure Bar Tests on Dry Colorado Mason Sand Calibration Results with Initialized Velocity

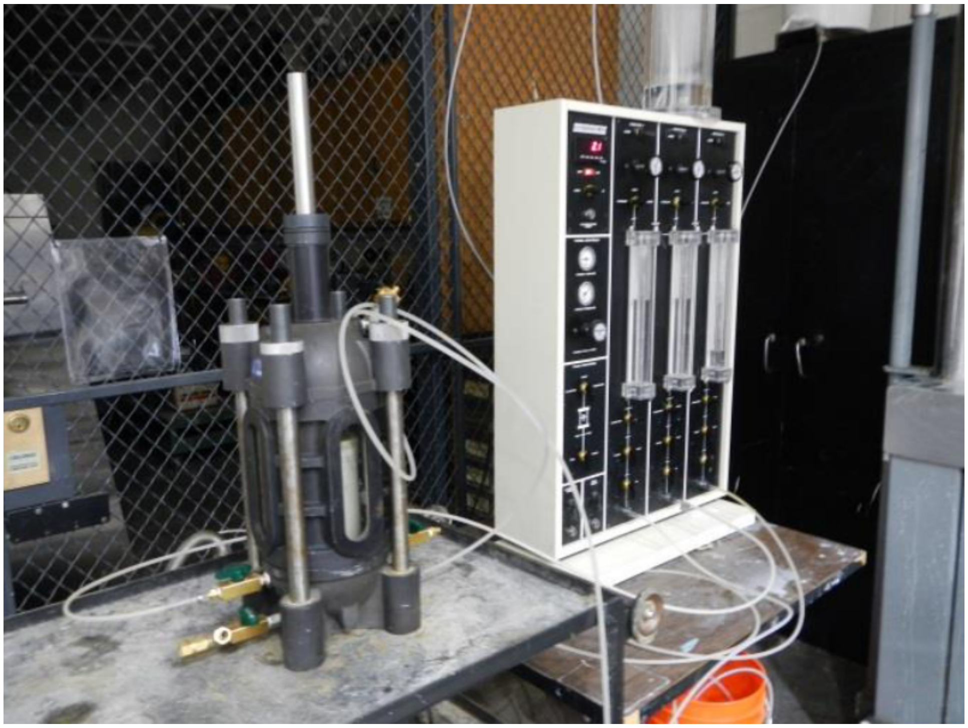


Figure 3.23: Triaxial Compression Test Apparatus from Svoboda (2013)

standard sized model and mesh can be found on figures 3.25 and 3.26. The displacements of the $-x$, $-y$, and $-z$ faces were restricted as they were for the high pressure isotropic compression simulations to create the quarter symmetry. The loading, however, was applied over two steps of the simulation to mimic the triaxial tests: isotropic compression and shearing. In the isotropic compression step, pressure boundary conditions were applied to the $+z$ and cylindrical faces to simulate the confining pressure applied to the soil specimen through pressurizing the water in the triaxial apparatus' cell. The cell pressure for the standard triaxial test was $\sigma_{\text{cell}} = 206$ kPa and $\sigma_{\text{cell}} = 40$ kPa for the small triaxial test. After the isotropic compression step and holding the confining stress constant, the simulations were sheared through applying a displacement boundary condition to the top ($+z$ face) of the samples during the second step. For both triaxial tests, the displacement was applied until roughly 15% axial strain was reached. Like the other models, linear hexahedral finite elements with reduced integration were used to model the triaxial tests.

Results from the calibration of the standard triaxial compression test can be found on figures

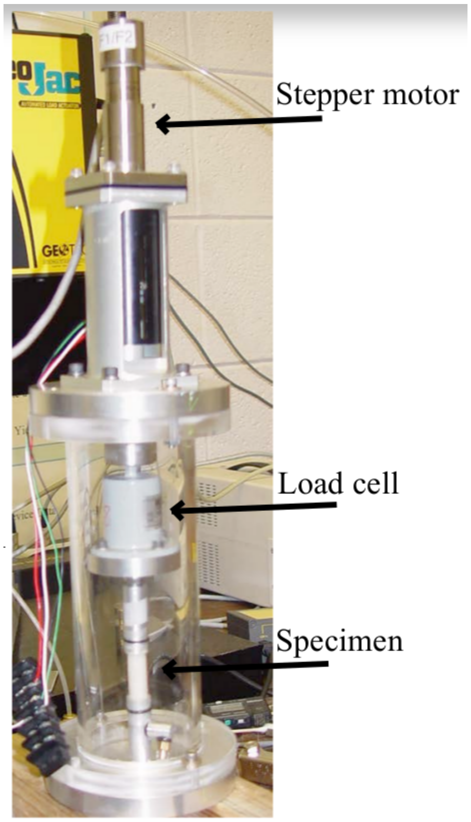


Figure 3.24: Small Triaxial Compression Test Apparatus from Druckrey and Alshibli (2014)

3.27 and 3.28. First, a calibration attempt using the parameters derived from the previous two tests grossly over-predicted the stress response, so the elastic bulk modulus was lowered to $K = 35MPa$. This illuminates the need to add non-linear, pressure-dependent elasticity as the elastic response of the material is pressure dependent. Also, Arena is currently unable to capture the post-peak softening behavior of the sand, which accounts for the deviation of the experimental results from the calibrated model fit after approximately 2.5% axial strain. This behavior, which is common for dense sand specimens, is also unable to be captured by the full Arena model implemented in Uintah. Shear softening models do exist to capture this behavior, but they often have spatial discretization convergence issues as the softening is often localized. Although stress paths through the shear surface will be perfectly plastic in the current implementation, the stress path contacts the compressive cap hardening the material through pore collapse.

Similarly to the standard triaxial test results, the small triaxial compression test results and

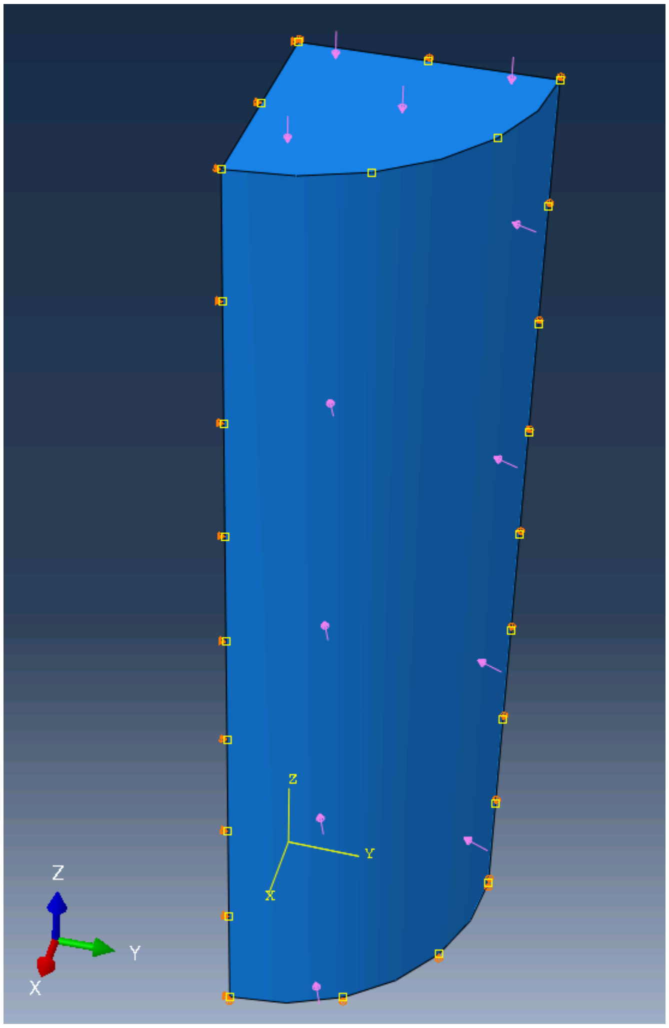


Figure 3.25: Triaxial Compression Test Simulation Geometry and Boundary Conditions

calibration of the Arena ABAQUS UMAT soil model can be found on figures 3.29 and 3.30. As the small triaxial test involved even smaller applied stresses than the standard triaxial tests, the elastic bulk modulus needed to be further reduced to $K = 3.5$ MPa, or 10% of the bulk modulus used in the standard tests. These results further emphasize the need to include non-linear, pressure-sensitive, elasticity and the inability of the continuum constitutive model to capture the post-peak shear behavior of the dry Colorado Mason Sand, which could be modeled using a critical-state sand plasticity model such as Dafalias and Manzari (2004), along with proper element technology to handle shear banding in finite element meshes.

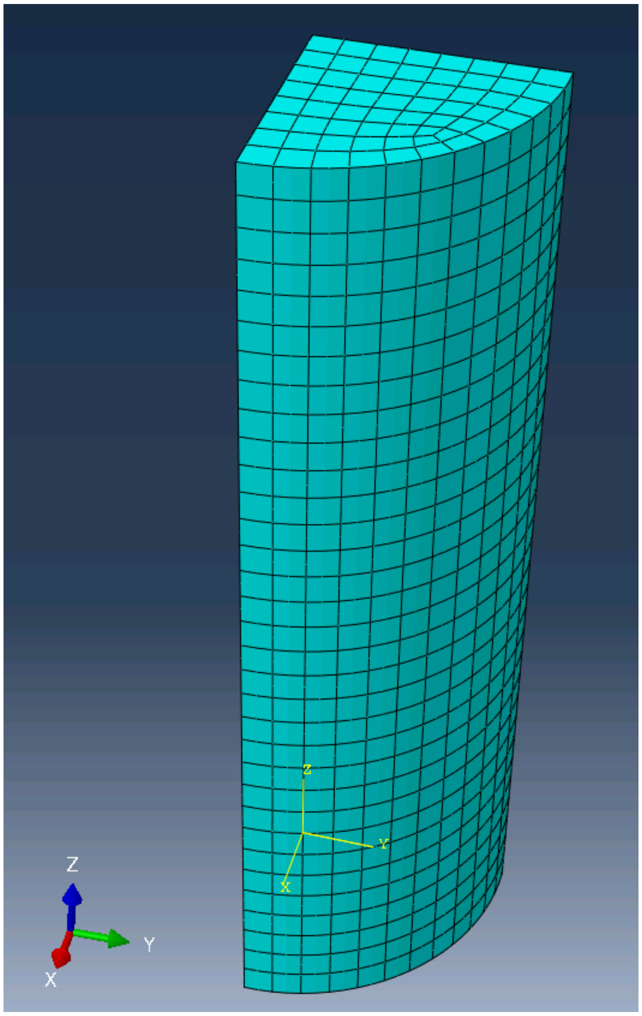


Figure 3.26: Triaxial Compression Test Simulation Mesh

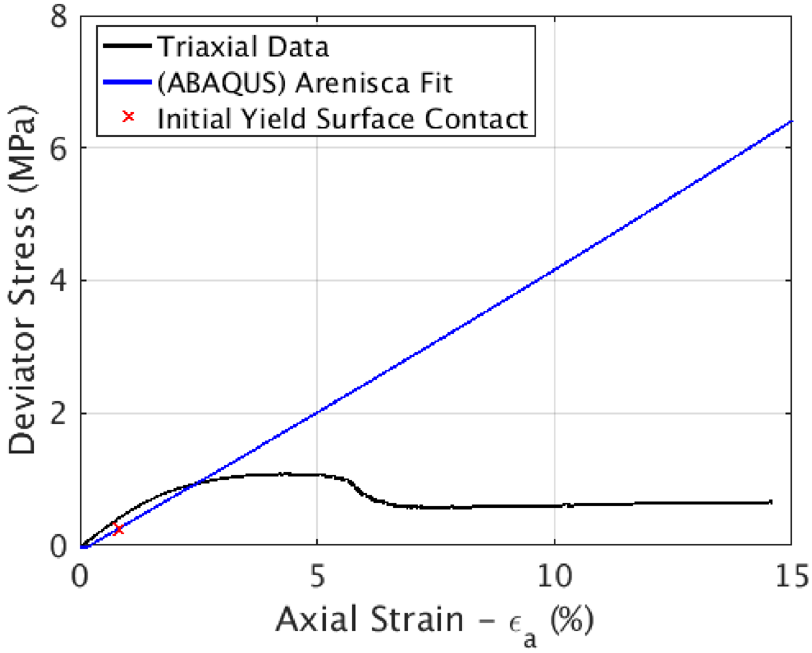


Figure 3.27: Triaxial Compression Tests on Dry Colorado Mason Sand Axial Strain Calibration Results

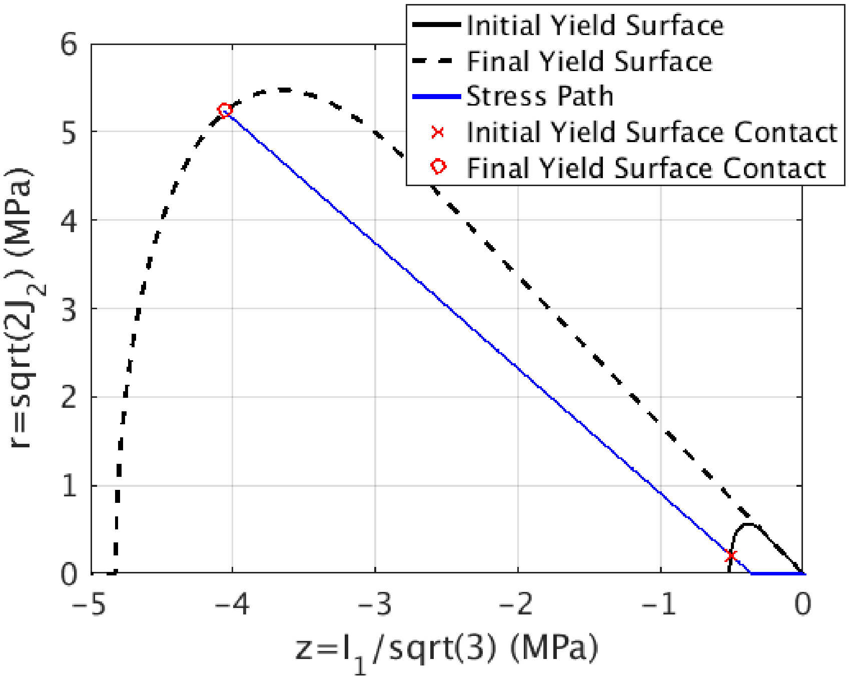


Figure 3.28: Triaxial Compression Tests on Dry Colorado Mason Sand Yield Surface

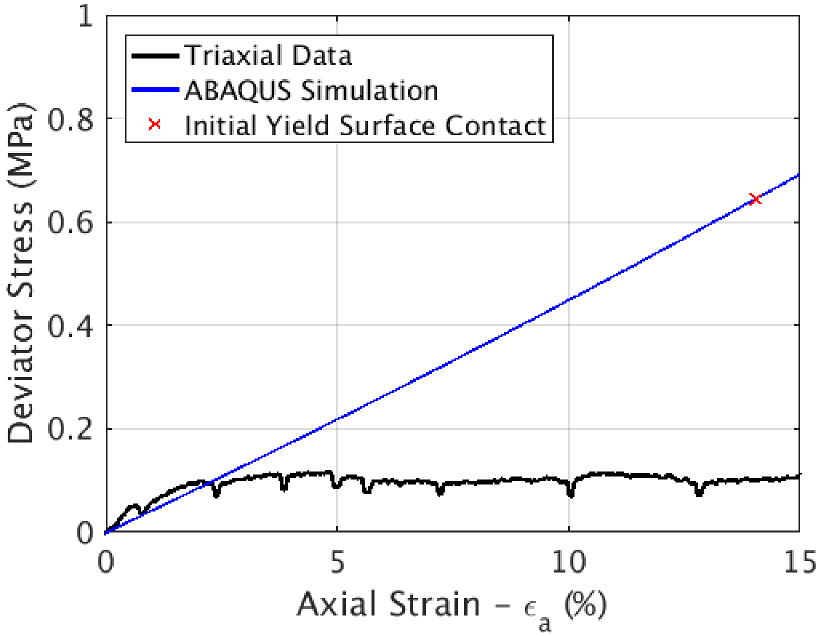


Figure 3.29: Small Triaxial Compression Tests on Dry Colorado Mason Sand Axial Strain Calibration Results

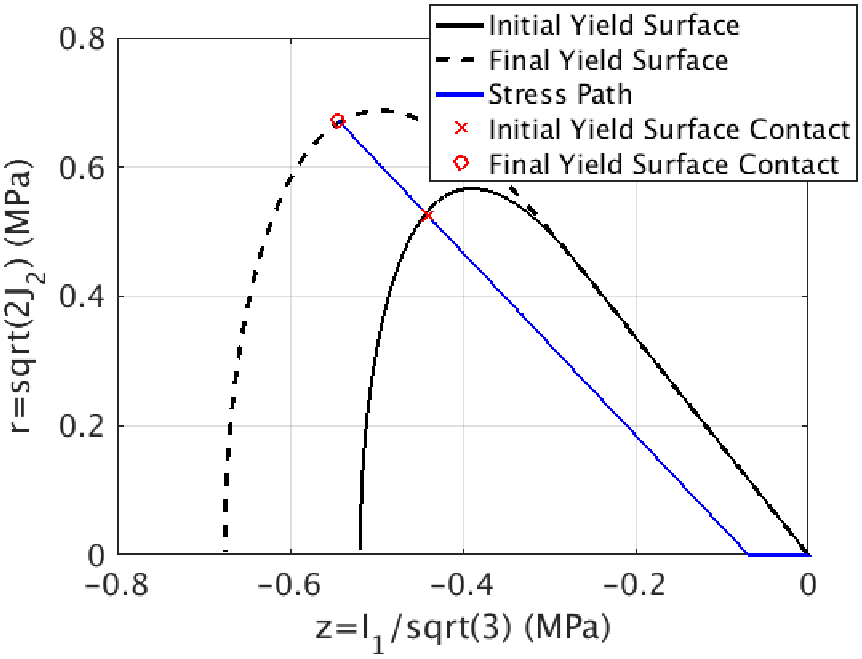


Figure 3.30: Small Triaxial Compression Tests on Dry Colorado Mason Sand Yield Surface

Chapter 4

1D FEM-DEM Global-Local Analysis

In an effort to strike a balance between the fidelity of direct numerical simulations such as discrete element modeling (DEM) and the relative speed of macroscale finite element analysis, a hierarchical multiscale modeling technique was developed and used to simulate an experiment conducted for the Soil Blast MURI Project Modeling project. Full direct numerical simulations of even small laboratory experiments on soil require access to supercomputers to be tractable, and finite element models are only as accurate as the kinematics and constitutive models one can reasonably implement. The hierarchical multiscale modeling algorithm uses the framework of the finite element method (FEM), but replaces the constitutive model with representative volume elements (RVEs) of soil that deform with the FEM mesh. Essentially, the algorithm replaces the phenomenological stress-strain response of soil with a potentially more accurate direct numerical simulation.

Ultimately, the goal of the project was to upscale the sand-grain-scale modeling effort with combined DEM representing sand particles, smoothed particle hydrodynamics (SPH) modeling free liquid water, peridynamics (PD) representing saturated and partially saturated clay matrix, and computational fluid dynamics (CFD) to model the background air and liquid water vapor to a full material point method (MPM) simulation in Uintah. However, in order to study the hierarchical multiscale modeling algorithm more closely, the model has been simplified replacing MPM for FEM and the full grain-scale DEM-SPH-PD-CFD model with only DEM to remove unnecessary complexity at this time.

Regardless, the general concept is the same. At every timestep, a macroscale continuum code (FEM) calculates the deformation gradient ($\boldsymbol{F}$) given the boundary conditions applied to the mesh. The deformation gradient is then passed to DEM RVEs located at each integration point in the FEM mesh as boundary conditions applied to each RVE. The RVEs are allowed to deform given the applied displacements, and then a homogenization technique is used to calculate the macroscale Cauchy stress ($\boldsymbol{\Sigma}$) in each RVE. Thus, what is traditionally done by a constitutive model has been replaced by the DEM RVEs. The updated Cauchy stress is passed back to the FEM code, and the simulation then moves to the next timestep. Such technique can be similarly applied to other variables that a constitutive model would traditionally handle such as liquid pore pressure or temperature over the RVE.

To date, the hierarchical multiscale modeling has been used to simulate one laboratory experiment that was conducted for the Soil Blast MURI Project. A one dimensional, uniaxial strain, finite strain model with inertial terms of the Split Hopkinson Pressure Bar (SHPB) experiment on dry Colorado Mason Sand conducted by Luo et al. (2011) was developed to test the efficacy of the model. The SHPB is a uniaxial strain, high strain rate experiment where a sample of soil is placed between two steel bars and one of the bars is rapidly accelerated into the sample (see figure 3.13 for schematic). The high strain rate coupled with the inherent nonlinearities of soil's constitutive behavior pose a number of challenges for traditional finite element modeling. To address some of these issues, the hierarchical multiscale modeling algorithm was applied to a simple one dimensional, finite uniaxial strain representation of the SHPB experiment on dry Colorado Mason Sand. The dry sand tests were chosen to eliminate the complications of pore water and clay both in the FEM kinematics and DEM particle representations. The full finite strain kinematics were implemented as the SHPB experiments routinely cause strains on the order of 30%, so accounting for the nonlinear geometric effects was deemed important. Also, the model was intentionally kept simple through use of a two element FEM mesh, linear interpolation functions, and only DEM particles in order to more closely focus on the hierarchical multiscale modeling rather than issues relating to the finite or discrete element methods themselves.

4.1 Finite Element Model Development

Starting with the localized form of the balance of linear momentum with inertial terms

$$\frac{\partial P_{iI}}{\partial X_I} + \rho_0 g_i - \rho_0 a_i = 0 \tag{4.1}$$

where P_{iI} is the first Piola-Kirchhoff stress, X_I is a position vector in the reference configuration ($\mathcal{B}_0$), ρ_0 is the initial density of the material, g_i is the acceleration vector due to gravity, and a_i is the spatial acceleration vector (all in index notation, assuming Cartesian coordinates), the method of weighted residuals was applied to obtain the weak form of the balance equation.

$$G(u_i, w_i) = \int_{\mathcal{B}_0} w_i \left(\frac{\partial P_{iI}}{\partial X_I} + \rho_0 g_i - \rho_0 a_i \right) \mathrm{d}V = 0 \tag{4.2}$$

$\mathrm{d}V$ is the differential volume of the reference configuration, u_i is the spatial displacement vector, w_i an arbitrary weighting vector, and the entire weak form is defined as the function $G(u_i, w_i)$. Recognizing that $w_i \frac{\partial P_{iI}}{\partial X_I} = \frac{\partial (w_i P_{iI})}{\partial X_I} - \frac{\partial w_i}{\partial X_I} P_{iI}$ by the chain rule and $\int_{\mathcal{B}_0} \frac{\partial (w_i P_{iI})}{\partial X_I} \mathrm{d}V = \int_{\Gamma_0^t} (w_i P_{iI} N_I)\, \mathrm{d}A = \int_{\Gamma_0^t} (w_i t_i^\sigma)\, \mathrm{d}A$ through the divergence theorem (where $t_i^\sigma = P_{iI} N_I$ is a traction boundary condition applied in the N_I direction along the boundary surface (Γ_0^t) of the specimen in the reference configuration), the final variational form of the balance of linear momentum is as follows.

$$G(u_i, w_i) = \underbrace{\int_{\mathcal{B}_0} (\rho_0 w_i a_i)\, \mathrm{d}V}_{G_1^{\text{int}}} + \underbrace{\int_{\mathcal{B}_0} \left(\frac{\partial w_i}{\partial X_I} P_{iI} \right) \mathrm{d}V}_{G_2^{\text{int}}} - \underbrace{\int_{\mathcal{B}_0} (\rho_0 w_i g_i)\, \mathrm{d}V}_{G_1^{\text{ext}}} - \underbrace{\int_{\Gamma_0^t} (w_i t_i^\sigma)\, \mathrm{d}A}_{G_2^{\text{ext}}} = 0 \tag{4.3}$$

G_1^{int} is the force due to the acceleration of the specimen, G_2^{int} is the force due to the internal strain energy, G_1^{ext} is the force due to the acceleration due to gravity, and G_2^{ext} is the force due to an applied traction. For this analysis, it was assumed that due to the small size of the soil specimen the acceleration due to gravity was negligible and no external tractions were applied; therefore, G_1^{ext} and G_2^{ext} were set equal to zero.

The DEM model outputs the stress in the current configuration (Cauchy stress). The first

Piola-Kirchhoff stress can be mapped to the Cauchy stress (σ_{ij}) through the Piola transform $P_{iI} = J\sigma_{ij}F^{-1}_{Ij}$, where J is the Jacobian of the deformation gradient F_{iI}, such that G_2^{int} is then equal to

$$G_2^{\text{int}} = \int_{\mathcal{B}_0} \left(J\frac{\partial w_i}{\partial X_I}\sigma_{ij}F^{-1}_{Ij} \right) \mathrm{d}V \tag{4.4}$$

and the resulting variational form of the balance of linear momentum is as follows.

$$G(u_i, w_i) = G_1^{\text{int}} + G_2^{\text{int}} = \int_{\mathcal{B}_0} (\rho_0 w_i a_i)\,\mathrm{d}V + \int_{\mathcal{B}_0} \left(J\frac{\partial w_i}{\partial X_I}\sigma_{ij}F^{-1}_{Ij} \right) \mathrm{d}V \tag{4.5}$$

The full three dimensional equation (equation 4.5) was then simplified for the one dimensional uniaxial strain case. The weighting vector (w_i), displacement vector ($u_i = x_i - X_i$), and spatial acceleration vector a_i simplify to

$$w_i = \begin{bmatrix} w \\ 0 \\ 0 \end{bmatrix}, \quad u_i = \begin{bmatrix} u \\ 0 \\ 0 \end{bmatrix}, \quad a_i = \begin{bmatrix} \ddot{u} \\ 0 \\ 0 \end{bmatrix} \tag{4.6}$$

and $\mathrm{d}V = A\mathrm{d}X$, where A is the constant cross-sectional area of the sand specimen. The deformation gradient ($F_{iI} = \frac{\partial x_i}{\partial X_I} = \frac{\partial(u_i + X_i)}{\partial X_I} = \delta_{iI} + \frac{\partial u_i}{\partial X_I}$) simplifies to $F_{11} = 1 + \frac{\partial u}{\partial X}$ and the Jacobian determinant of the deformation gradient (J) simplifies to $J = 1 + \frac{\partial u}{\partial X}$. As a result, the Cauchy stress (σ_{11}) is equal to the first Piola-Kirchhoff stress (P_{11}). Therefore, the one dimensional, uniaxial strain simplification of the variational form of the balance of linear momentum is

$$G(u, w) = G_1^{\text{int}} + G_2^{\text{int}} = \int_0^L (\rho_0 w \ddot{u})\, A\mathrm{d}X + \int_0^L \left(\frac{\partial w}{\partial X}\sigma_{11} \right) A\mathrm{d}X \tag{4.7}$$

where L is the initial length of the specimen.

In finite element analysis, the balance equations are discretized into their Galerkin (discretized) form and solved in each element of a mesh that approximates the problem domain. The Galerkin form of the balance of linear momentum is

$$G^h(u,w) = G_1^{\mathrm{int},h} + G_2^{\mathrm{int},h} = \int_0^L \left(\rho_0 w^h \ddot{u}^h\right) A\mathrm{d}X + \int_0^L \left(\frac{\partial w^h}{\partial X}\sigma_{11}^h\right) A\mathrm{d}X \tag{4.8}$$

This equation is solved over each of two finite elements in the simplified SHPB experiment simulation. For this simulation, one of the boundary nodes was fixed in space, and a displacement boundary condition was applied to the other boundary node resulting in one degree of freedom at the center node. A schematic of the resulting finite element mesh containing two elements and one degree of freedom (d_1) can be found on figure 4.1.

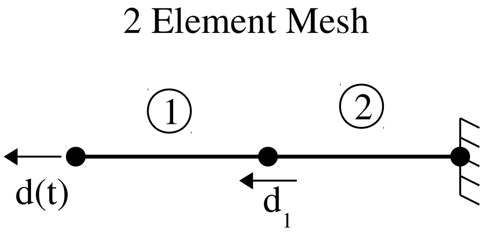


Figure 4.1: Two Element SHPB Mesh

Due to the inclusion of inertial terms, a Newmark time integration scheme was used to discretize the balance equation in time. In general, the balance equation is solved at the current timestep $(n+1)$ such that $G_{n+1}^h = 0$. To do this, the acceleration at the current time ($\ddot{u}_{n+1}^h$) is predicted from the velocity and displacement at the previous timestep ($\dot{u}_n^h$ and u_n^h, respectively) via the following equations

$$u_{n+1}^h = \underbrace{u_n^h + \Delta t \dot{u}_n^h + \frac{1}{2}\Delta t^2 (1-2\beta)\ddot{u}_n^h}_{\tilde{u}^h} + \Delta t^2 \beta \ddot{u}_{n+1}^h \tag{4.9}$$

$$\dot{u}_{n+1}^h = \underbrace{\dot{u}_n^h + \Delta t (1-\gamma)\ddot{u}_n^h}_{\tilde{v}^h} + \Delta t \gamma \ddot{u}_{n+1}^h \tag{4.10}$$

where Δt is the timestep, $\tilde{u}^h$ and $\tilde{v}^h$ are defined as the displacement and velocity "predictors", and β and γ are Newmark integration parameters. If $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$, the Newmark method simplifies to the implicit, second order accurate trapeziodal rule, and if $\beta = 0$ and $\gamma = \frac{1}{2}$, the Newmark method simplifies to explicit, second order accurate central difference in time.

4.1.1 Implicit Formulation

As it was assumed that the DEM model substituted for a traditional constitutive model would be highly nonlinear, the balance equation was linearized to solve in a Newton-Raphson algorithm using the acceleration method, such that

$$\mathcal{L}G_{n+1}^{k+1,h} = G_{n+1}^{k,h} + \delta G_{n+1}^{k+1,h} = 0 \tag{4.11}$$

where $\mathcal{L}G_{n+1}^{k+1,h}$ is the linearized version of the balance of linear momentum at the current timestep and Newton-Raphson iteration $(k+1)$, $G_{n+1}^{k,h}$ is the value of the balance of linear momentum at the previous Newton-Raphson iteration (k), and $\delta G_{n+1}^{k+1,h}$ is the difference (increment) between the two solutions. The increment of displacement and velocity (δu^h and δv^h, respectively) as a function of increment of acceleration (δa^h) are as follows.

$$\delta u^h = \Delta t^2 \beta \delta a^h \tag{4.12}$$

$$\delta v^h = \Delta t \gamma \delta a^h \tag{4.13}$$

Further expansion of the linearization leads to

$$G_{n+1}^{k,h} = (G_1^{\mathrm{int}})_{n+1}^{k,h} + (G_2^{\mathrm{int}})_{n+1}^{k,h} \tag{4.14}$$

where

$$(G_1^{\mathrm{int}})_{n+1}^{k,h} = \int_0^L \left(\rho_0 w^h \ddot{u}_{n+1}^{k,h}\right) A\mathrm{d}X \tag{4.15}$$

$$(G_2^{\mathrm{int}})_{n+1}^{k,h} = \int_0^L \left(\frac{\partial w^h}{\partial X}(\sigma_{11})_{n+1}^{k,h}\right) A\mathrm{d}X \tag{4.16}$$

Similarly, the increment between the solutions to the balance of linear momentum expands to

$$\delta G_{n+1}^{k+1,h} = \delta G_1^{\text{int},h} + \delta G_2^{\text{int},h} \tag{4.17}$$

where the $\delta G_1^{\text{int},h}$ and $\delta G_2^{\text{int},h}$ equal the following.

$$\delta G_1^{\text{int},h} = \int_0^L \left(\rho_0 w^h \delta a^h\right) A\mathrm{d}X \tag{4.18}$$

$$\delta G_2^{\text{int},h} = \int_0^L \left(\frac{\partial w^h}{\partial X}\delta\sigma_{11}^h\right) A\mathrm{d}X \tag{4.19}$$

For simplicity, the finite elements were interpolated in space through linear interpolation functions (N_a). A schematic of the element coordinate systems (global (X) and natural (ξ)) can be seen on figure 4.2. d_1^e and d_2^e are the displacements at the two nodes on each element e, and X_1^e and X_2^e are the global positions of the two nodes. To aid with the spatial integration, the position of the nodes are mapped to the natural coordinate system where node 1 is at the position $\xi = -1$ and node 2 is at $\xi = 1$. H^e is the distance between the two nodes (also known as the representative element size). As a result, the nodal displacement (u^{h^e}), weighting function (w^{h^e}), and acceleration (a^{h^e}) for each element were interpolated in the natural coordinate system using the following equations

$$u^{h^e}(\xi, t) = \sum_{a=1}^{2} N_a(\xi) d_a^e(t) = \boldsymbol{N}^e \cdot \boldsymbol{d}^e = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} d_1^e \\ d_2^e \end{bmatrix} \tag{4.20}$$

$$w^{h^e}(\xi) = \boldsymbol{N}^e \cdot \boldsymbol{c}^e = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} c_1^e \\ c_2^e \end{bmatrix} \tag{4.21}$$

$$a^{h^e}(\xi, t) = \boldsymbol{N}^e \cdot \boldsymbol{a}^e = \begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} a_1^e \\ a_2^e \end{bmatrix} \tag{4.22}$$

where c_1^e and c_2^e are the interpolated weighting functions at the nodes and the linear interpolation functions (N_a) in the natural coordinate system are equal to $N_1 = \frac{1}{2}(1 - \xi)$ and $N_2 = \frac{1}{2}(1 + \xi)$. Finally, the derivatives of the nodal displacement ($\frac{\partial u^{h^e}}{\partial X}$) and weighting ($\frac{\partial w^{h^e}}{\partial X}$) function with respect to the reference coordinates of the nodes (X) are

$$\frac{\partial u^{h^e}}{\partial X} = \boldsymbol{B}^e \cdot \boldsymbol{d}^e = \begin{bmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial X} \end{bmatrix} \begin{bmatrix} d_1^e \\ \\ d_2^e \end{bmatrix} \tag{4.23}$$

$$\frac{\partial w^{h^e}}{\partial X} = \boldsymbol{B}^e \cdot \boldsymbol{c}^e = \begin{bmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial X} \end{bmatrix} \begin{bmatrix} c_1^e \\ \\ c_2^e \end{bmatrix} \tag{4.24}$$

where $\boldsymbol{B}^e$ is the strain-displacement matrix and for one dimensional uniaxial strain is equal to $\boldsymbol{B}^e = \frac{1}{H^e}[-1 \quad 1]$.

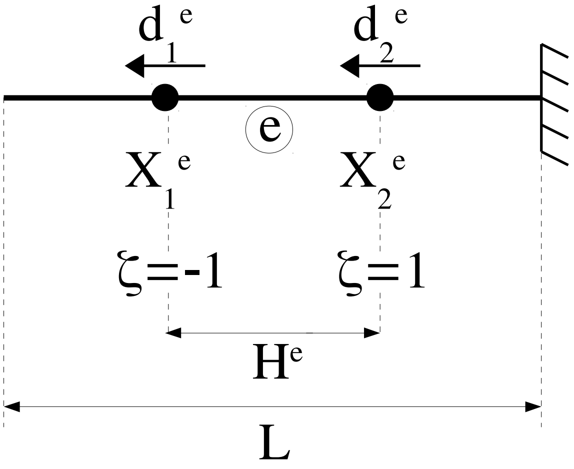


Figure 4.2: Finite Element Geometry in Global and Natural Coordinate Systems

Therefore, the Galerkin form of the balance equation was calculated in each element and then assembled to arrive at the final finite element equations.

$$(G_1^{\mathrm{int}})_{n+1}^{k,h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\rho_0 w^{h^e} \ddot{u}_{n+1}^{k,h^e}\right) A^e \mathrm{d}X = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} \rho_0 (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{N}^e A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{m}^e} \cdot \boldsymbol{a}^e \tag{4.25}$$

$$(G_2^{\mathrm{int}})_{n+1}^{k,h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\frac{\partial w^{h^e}}{\partial X} (\sigma_{11})_{n+1}^{k,h^e}\right) A^e \mathrm{d}X = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} (\sigma_{11})_{n+1}^{k,h^e} A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{f}_2^{\mathrm{int},e}} \tag{4.26}$$

$$\delta G_1^{\mathrm{int},h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\rho_0 w^{h^e} \delta a^{h^e}\right) A^e \mathrm{d}X = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} \rho_0 (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{N}^e A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{m}^e} \cdot \delta \boldsymbol{a}^e \tag{4.27}$$

$$\delta G_2^{\mathrm{int},h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\frac{\partial w^{h^e}}{\partial X} \delta \sigma_{11}^{h^e}\right) A^e \mathrm{d}X = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \left(\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} \delta \sigma_{11}^{h^e} A^e \frac{H^e}{2} \mathrm{d}\xi\right) \tag{4.28}$$

where n_{el} is the total number of finite elements, $\boldsymbol{m}^e$ is the given element's mass matrix, $\boldsymbol{f}_2^{\mathrm{int},e}$ is the elemental internal forcing vector due to the internal strain energy, j^e is the Jacobian of the coordinate transformation from the global coordinate system ($\mathrm{d}X$) to the natural coordinate system ($\mathrm{d}\xi$), and $\mathbf{A}_{e=1}^{n_{el}}$ is the assembly operator, which signifies piecing together the element matrices that have been calculated within each of the elements. Given the one dimensional uniaxial strain simplification, $j^e = \frac{H^e}{2}$ such that $\mathrm{d}X = j^e \mathrm{d}\xi = \frac{H^e}{2}\mathrm{d}\xi$. Finally, the material consistent tangent ($\boldsymbol{D}$) in finite strain is defined as $\frac{\partial P_{iI}}{\partial F_{aA}}$, which simplifies to $D_{11} = \frac{\partial \sigma_{11}}{\partial F_{11}}$ for one dimensional uniaxial strain. Recognizing that $F_{11} = 1 + \frac{\partial u}{\partial X}$, the discretized increment of the Cauchy stress then equals $\delta\sigma_{11}^{h^e} = D_{11}\frac{\partial(\delta u)}{\partial X}$. When reformulated using the acceleration method (equation 4.12), the $\delta G_2^{\mathrm{int},h}$ term in the balance of linear momentum equation (equation 4.28) becomes

$$\delta G_2^{\mathrm{int},h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} D_{11}^{h^e} \boldsymbol{B}^e A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{k}^e} \cdot \delta \boldsymbol{a}^e (\Delta t^2 \beta) \tag{4.29}$$

where $\boldsymbol{k}^e$ is the given element's consistent tangent. Like the Cauchy stress, D_{11} is approximated for each element using the discrete element method code.

Therefore, the discretized and linearized balance of linear momentum simplifies to the following at the current Newton-Raphson iteration ($k+1$) and timestep ($n+1$), which have been left off for simplicity.

$$\mathcal{L}G_{n+1}^{k+1,h} = \mathcal{L}G^h = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \boldsymbol{c}^{\mathrm{T}} \cdot \left\{ \boldsymbol{m}^e \cdot \boldsymbol{a} + \boldsymbol{f}_2^{\mathrm{int},e} + \boldsymbol{m}^e \cdot \delta\boldsymbol{a}^e + (\Delta t^2 \beta)\boldsymbol{k}^e \cdot \delta\boldsymbol{a}^e \right\} = 0 \tag{4.30}$$

After applying the essential boundary conditions and assuming an arbitrary $\boldsymbol{c}^{\mathrm{T}}$ for all degrees of freedom (global displacement ($\boldsymbol{d}$), velocity ($\boldsymbol{v}$), and acceleration ($\boldsymbol{a}$)), the final assembled version of the balance of linear momentum becomes

$$\boldsymbol{M} \cdot \boldsymbol{a} + \boldsymbol{F}_2^{\mathrm{int}} + \big(\boldsymbol{M} + (\Delta t^2 \beta)\boldsymbol{K}\big) \cdot \delta\boldsymbol{a} = \boldsymbol{0} \tag{4.31}$$

where $\boldsymbol{M}$ and $\boldsymbol{K}$ are the global mass and stiffness matrices, respectively. Then, equation 4.31 is solved for $\delta\boldsymbol{a}$

$$\delta\boldsymbol{a} = \big(\boldsymbol{M} + (\Delta t^2 \beta)\boldsymbol{K}\big)^{-1} \left(-\boldsymbol{M} \cdot \boldsymbol{a} - \boldsymbol{F}_2^{\mathrm{int}}\right) \tag{4.32}$$

and the acceleration for the current Newton-Raphson iteration is updated.

$$\boldsymbol{a}_{n+1}^{k+1} = \boldsymbol{a}_{n+1}^{k} + \delta\boldsymbol{a}_{n+1} \tag{4.33}$$

Using the updated acceleration, the Newton-Raphson residual ($\|\boldsymbol{R}^{k+1}\|$) is checked for convergence such that

$$\frac{\|\boldsymbol{R}^{k+1}\|}{\|\boldsymbol{R}^{0}\|} < \mathrm{tol} \tag{4.34}$$

where $\boldsymbol{R}^{k+1} = \boldsymbol{M} \cdot \boldsymbol{a}^{k+1} + \boldsymbol{F}_2^{\mathrm{int}}(\boldsymbol{d}^{k+1})$ is the residual at the current Newtwon-Raphson iteration $(k+1)$, $\boldsymbol{R}^0$ is the residual in the first Newton-Raphson iteration $(k=0)$, and tol is a user defined tolerance.

4.1.2 Explicit Formulation

Despite the assumed nonlinearity of the DEM model and the numerical stability advantages of the implicit Newton-Raphson algorithm, a second explicit version of the finite element formulation was also developed in order to compare and contrast the two methods. Starting with the Galerkin form of the balance of linear momentum for one dimensional, uniaxial finite strain without including external tractions and gravity from the implicit formulation,

$$G^h(u,w) = G_1^{\text{int},h} + G_2^{\text{int},h} = \int_0^L \left(\rho_0 w^h \ddot{u}^h\right) A\text{d}X + \int_0^L \left(\frac{\partial w^h}{\partial X}\sigma_{11}^h\right) A\text{d}X \tag{4.35}$$

the balance equation was again integrated in time using the Newmark time integration scheme.

$$u_{n+1}^h = \underbrace{u_n^h + \Delta t\dot{u}_n^h + \frac{1}{2}\Delta t^2(1-2\beta)\ddot{u}_n^h}_{\tilde{u}^h} + \Delta t^2\beta\ddot{u}_{n+1}^h \tag{4.36}$$

$$\dot{u}_{n+1}^h = \underbrace{\dot{v}_n^h + \Delta t(1-\gamma)\ddot{u}_n^h}_{\tilde{v}^h} + \Delta t\gamma\ddot{u}_{n+1}^h \tag{4.37}$$

Rather than setting $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ to use the implicit trapezoidal rule, β and γ parameters were set to 0 and $\frac{1}{2}$, respectively, to use the explicit central difference in time algorithm, which further simplifies equations 4.36 and 4.37 to the following.

$$u_{n+1}^h = \underbrace{u_n^h + \Delta t\dot{u}_n^h + \frac{1}{2}\Delta t^2\ddot{u}_n^h}_{\tilde{u}^h} \tag{4.38}$$

$$\dot{u}_{n+1}^h = \underbrace{\dot{v}_n^h + \frac{1}{2}\Delta t\ddot{u}_n^h}_{\tilde{v}^h} + \frac{1}{2}\Delta t\ddot{u}_{n+1}^h \tag{4.39}$$

In this case, the current displacement and displacement predictor are equal and only dependent on the kinematic variables of the previous timestep ($u_{n+1}^h = \tilde{u}^h$).

To solve for the current acceleration ($\ddot{u}_{n+1}^h$), the balance of linear momentum was again discretized into finite elements as was done in the implicit formulation. First, it was substituted into the Newmark time integration scheme,

$$(G_1^{\text{int}})_{n+1}^h = \int_0^L \left(\rho_0 w^h \ddot{u}_{n+1}^h\right) A\text{d}X \tag{4.40}$$

$$(G_2^{\text{int}})_{n+1}^h = \int_0^L \left(\frac{\partial w^h}{\partial X}(\sigma_{11})_{n+1}^h\right) A\text{d}X \tag{4.41}$$

and then using equations 4.20 through 4.24, converted to the finite element matrix form in the natural coordinate system,

$$(G_1^{\mathrm{int}})^h_{n+1} = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\rho_0 w^{h^e} \ddot{u}^{h^e}_{n+1}\right) A^e \mathrm{d}X = \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} \rho_0 (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{N}^e A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{m}^e} \cdot \boldsymbol{a}^e \qquad (4.42)$$

$$(G_2^{\mathrm{int}})^h_{n+1} = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \int_{X_1^e}^{X_2^e} \left(\frac{\partial w^{h^e}}{\partial X} (\sigma_{11})^{h^e}_{n+1}\right) A^e \mathrm{d}X = \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \underbrace{\left(\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} (\sigma_{11})^{k,h^e}_{n+1} A^e \frac{H^e}{2} \mathrm{d}\xi\right)}_{\boldsymbol{f}_2^{\mathrm{int},e}} \qquad (4.43)$$

which simplifies to the following.

$$G^h_{n+1} = \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \left\{\boldsymbol{m}^e \cdot \boldsymbol{a}^e + \boldsymbol{f}_2^{\mathrm{int},e}\right\} = 0 \qquad (4.44)$$

Finally, after assembly, the resulting system of equations

$$\boldsymbol{M} \cdot \boldsymbol{a} + \boldsymbol{F}_2^{\mathrm{int}} = \boldsymbol{0} \quad \Longrightarrow \quad \boldsymbol{a} = \boldsymbol{M}^{-1} \cdot (-\boldsymbol{F}_2^{\mathrm{int}}) \qquad (4.45)$$

can be used to solve for the acceleration at the current timestep, which is subsequently used to update the velocity at the current timestep via equation 4.39. For more efficient, large-scale computation, $\boldsymbol{M}$ may be diagonalized for trivial calculation of $\boldsymbol{a}$.

4.2 Discrete Element Model Development

In traditional finite element analysis, the variational and discretized form of the balance of linear momentum (equations 4.25 through 4.28) is calculated for each element using Gauss quadrature. For one dimensional, linear elements, the integrals are calculated at two integration points within the elements. Traditionally, the stress terms are computed through the definition of a material constitutive model; however, hierarchical multiscale models (FEM-DEM models, in this case) replace the stress-strain relationship with a discrete element assembly that directly models the particle-to-particle interactions of the soil. The result is that at each finite element integration point, the constitutive subroutine of the finite element program has been replaced with a call to a discrete element method code called ellip3D that calculates the stress response of the soil.

Ellip3D is a discrete element method (DEM) code that was developed at the University of Colorado Boulder by Yan (2008). It represents individual grains with ellipsoids, which is an improvement over many commercial DEM codes that use spheres. Accurately representing the geometry of soil particles remains a challenge for DEM researchers, so adding geometric complexity helps the DEM particles more accurately represent the modeled soil. According to work by Zhang et al. (2017), it was determined that there is a difference in porosity and fabric when depositing spheres, ellipsoids, and poly-ellipsoids under gravity into a square cross-section container. More accurately modeling these soil properties increases the predictive capabilities of ellip3D. For the Soil Blast MURI Project, ellip3D was further extended to represent particles as poly-ellipsoids, to include a rudimentary particle fracture model to represent grain crushing and shearing, and has been massively parallelized for use on supercomputers to significantly decrease the overall computation time. For the purposes of the hierarchical multiscale models, only the added fracture model, implemented in the poly-ellipsoidal code (polyEllip3D), was considered at this time.

DEM uses a strong form representation of the balance of linear and angular momentum solved in each particle, which can be combined to form the standard equation of motion for rigid body dynamics.

$$\boldsymbol{M}_i\boldsymbol{a}_i + \boldsymbol{C}_i\boldsymbol{v}_i + \boldsymbol{P}_i = \boldsymbol{F}_i \tag{4.46}$$

$\boldsymbol{M}_i$, $\boldsymbol{C}_i$, $\boldsymbol{P}_i$, and $\boldsymbol{F}_i$ are the lumped mass matrix, damping matrix, contact force vector, and external forcing vector for the ith particle in the granular assembly. The mass matrix takes the generalized form

$$\boldsymbol{M}_i = \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & I_i & 0 & 0 \\ 0 & 0 & 0 & 0 & I_i & 0 \\ 0 & 0 & 0 & 0 & 0 & I_i \end{bmatrix} \tag{4.47}$$

where m_i and I_i are the mass and moment of inertia of the ith particle, and the damping matrix is

$$\boldsymbol{C}_i = \alpha_1 \boldsymbol{M}_i \tag{4.48}$$

where α_1 is the coefficient of mass proportional damping. Mass proportional damping is referred to as "background" damping and is generally only included in the formulation for simulating quasi-static problems in the context of a dynamic relaxation algorithm. The SHPB experiment is a dynamic simulation, so α_1 was set to zero for these simulations. The generalized acceleration and velocity vectors of the ith particle ($\boldsymbol{a}_i$ and $\boldsymbol{v}_i$, respectively) are,

$$\boldsymbol{v}_i^{\mathrm{T}} = \begin{bmatrix} (v_x)_i & (v_y)_i & (v_z)_i & (\omega_x)_i & (\omega_y)_i & (\omega_z)_i \end{bmatrix} \tag{4.49}$$

$$\boldsymbol{a}_i^{\mathrm{T}} = \begin{bmatrix} (\dot{v}_x)_i & (\dot{v}_y)_i & (\dot{v}_z)_i & (\dot{\omega}_x)_i & (\dot{\omega}_y)_i & (\dot{\omega}_z)_i \end{bmatrix} \tag{4.50}$$

where v is the velocity and ω is the angular velocity. The contact force/moment vector is comprised of the forces exerted on the ith particle by every particle j in contact with it.

$$\boldsymbol{P}_i^{\mathrm{T}} = \begin{bmatrix} \sum_{j=1}^{n_c} F_x^{j,i} & \sum_{j=1}^{n_c} F_y^{j,i} & \sum_{j=1}^{n_c} F_z^{j,i} & \sum_{j=1}^{n_c} M_x^{j,i} & \sum_{j=1}^{n_c} M_y^{j,i} & \sum_{j=1}^{n_c} M_z^{j,i} \end{bmatrix} \tag{4.51}$$

where $F^{j,i}$ is the contact force exerted on the ith particle by the nearby jth, $M^{j,i}$ is the contact moment exerted on the ith particle by the jth, and n_c is the total number of contacts associated with the ith particle. External forces on the particles include the force due to gravity and other body forces. The balance equation (equation 4.46) is integrated in time using the explicit central difference in time method. As the central difference in time is a conditionally stable integration technique, ellip3D requires use of a timestep (Δt) smaller than the theoretical critical timestep

$$\Delta t < 2\sqrt{\frac{m}{k}} \tag{4.52}$$

where m and k are the smallest particle mass and largest interparticle contact stiffness, respectively (Yan et al., 2010).

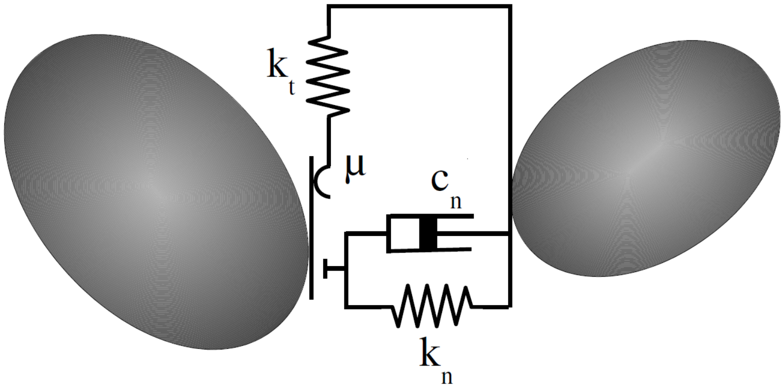


Figure 4.3: DEM Normal and Tangential Contact Model Schematic Adapted from Yan et al. (2010)

As background damping is ignored in dynamic simulations, contact damping is needed to model the dissipative interactions between particles along with frictional sliding. Figure 4.3 depicts the normal and tangential contact models used in ellip3D, where k_n and k_t are the normal and tangential contact stiffness, c_r is the contact normal damping coefficient, and μ is the shear friction coefficient. The viscous contact damping is modeled as a dashpot where the viscous normal force on a given particle is described by $\boldsymbol{F}_d = c_r \boldsymbol{v}_r$ where $\boldsymbol{v}_r$ is the normal relative velocity vector between two particles. The normal damping coefficient is assumed to be a fraction (ξ) of the critical damping (C_{cr}) which is derived as $c_r = \xi C_{cr} = \xi 2\sqrt{\frac{m_1 m_2 k_n}{m_1+m_2}}$ (Taylor and Preece, 1992) (Onate and Rojek, 2004). Selection of an appropriate damping ratio ξ is the subject of modeling experience and model calibration; the values typically fall within 1-5%. The normal and tangential contact stiffnesses are modeled using the non-linear elastic Hertz-Mindlin contact model often used in DEM codes (Hertz, 1882) (Mindlin, 1949). Hertz's model is used to model the normal contact behavior and Mindlin's model the tangential (Yan et al., 2010). Material properties used in the DEM simulation calibrated for Colorado Mason Sand can be found in table 4.1.

Table 4.1: DEM Material Properties for Colorado Mason Sand

Ellip3D Parameters	Value
Interparticle Damping (c_n)	0.7
Coef. of Friction (μ)	0.8
Young's Modulus (E)	107.8 GPa
Poisson's Ratio (ν)	0.18

A number of updates to ellip3D were performed for the Soil Blast MURI Project, one of which was specifically necessary for use in the hierarchical multiscale models. Zhang and Regueiro (2015) added computational homogenization techniques to accurately calculate the Cauchy stress and a number of finite strain measures from a DEM assembly. The Cauchy stress is calculated via an equation adapted from Christoffersen et al. (1981).

$$\Sigma_{ij} = \frac{1}{V} \sum_{c} l_i^c F_j^c \tag{4.53}$$

Σ_{ij} is the homogenized Cauchy (effective) stress averaged over the particle contacts (c), V is the volume of the DEM assembly, l_i^c is the branch vector between two particles in contact, and F_j^c is the contact force vector between two particles. A pictorial representation of the two vectors can be found on figure 4.4. The strain measures are all derived from the homogenized deformation gradient of the particle assembly ($F_{iI}^{\text{homogenized}}$), which is calculated via Delaunay tessellation. As seen on figure 4.5, the granular assembly is represented as a continuous tetrahedral mesh. The displacements of this mesh associated with particle centroid motions are used to calculate the deformation gradient.

Lastly, following the work of Guo and Zhao (2014), the material consistent tangent was approximated from ellip3D through estimating the homogenized elastic bulk modulus of the DEM particle assembly. Using what Meier et al. (2009) called the Taylor assumption, an analytical solution was derived by assuming a uniform strain field.

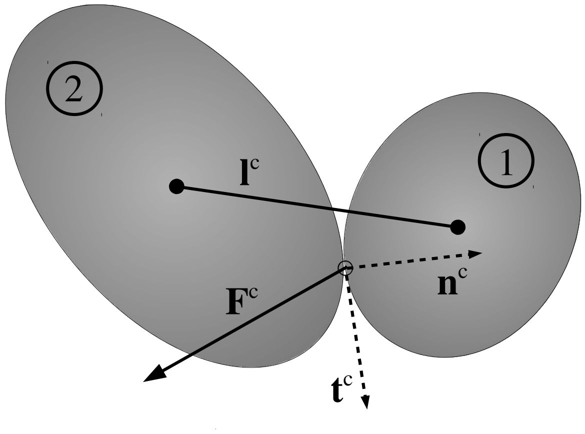


Figure 4.4: DEM Particle-to-particle Contact Vector Definitions for Stress Homogenization

$$\boldsymbol{D} = \frac{1}{V}\sum_{c}\left(k_n \boldsymbol{n}^c \otimes \boldsymbol{l}^c \otimes \boldsymbol{n}^c \otimes \boldsymbol{l}^c + k_t \boldsymbol{t}^c \otimes \boldsymbol{l}^c \otimes \boldsymbol{t}^c \otimes \boldsymbol{l}^c\right) \tag{4.54}$$

where $\boldsymbol{n}^c$ and $\boldsymbol{t}^c$ are unit normals in the normal and tangential directions of the contact vector $\boldsymbol{F}^c$ and $\boldsymbol{l}^c$ is the branch vector between the centroids of the particles in contact (see figure 4.4). Although there are other methods for estimating the material consistent tangent (namely the perturbation method (Nitka et al., 2011)), Guo and Zhao (2014) found estimating the homogenized elastic bulk modulus to converge faster in a Newton-Raphson algorithm.

4.3 Hierarchical Multiscale Modeling Technique

With both the macroscale and sand-grain-scale models developed, the two were coupled via a hierarchical multiscale modeling algorithm. As mentioned, the general goal was to replace the constitutive model at the integration points in the finite element code with a representative volume element of a soil modeled using the discrete element method. The DEM model then acts as the link between deformation and stress in each of the finite elements. To test the algorithm, a split Hopkinson pressure bar (SHPB) test on dry Colorado Mason Sand was modeled using the hierarchical multiscale model.

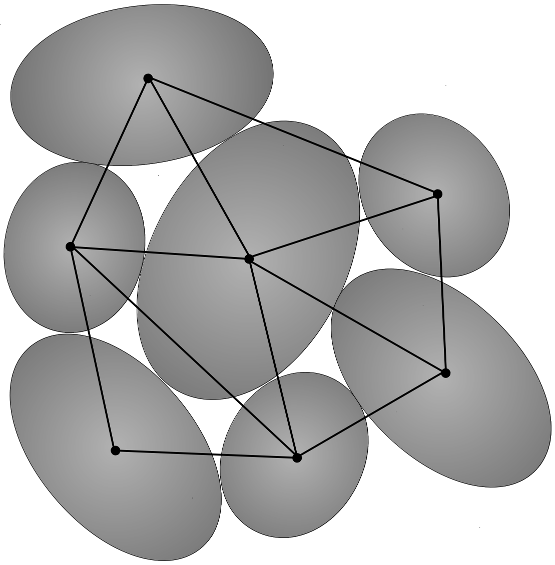


Figure 4.5: Two Dimensional Version of Mesh Used for Delaunay Tesselation

4.3.1 Applied Global Displacement

The FEM mesh used to simulate the SHPB experiment can be seen on figure 4.1. In SHPB tests, one of the goals is to "shape" the pressure pulse experienced by the specimen such that the strain rate is constant (Luo et al., 2011). Therefore, the displacement applied to the FEM mesh was designed in order that the strain rate would be constant given the finite, uniaxial deformation of the problem domain. Figure 4.6 shows the general geometry of the specimen where L_0 is the initial specimen length and $L(t)$ is the specimen length given the deformation $d(t)$, such that $L(t) = L_0 + d(t)$ (d is negative in compression). The logarithmic strain ($\epsilon(t)$) experienced by the specimen is derived as follows.

$$\int_0^{\epsilon} \mathrm{d}\epsilon = \int_{L_0}^{L} \frac{\mathrm{d}L}{L} \quad \Longrightarrow \quad \epsilon(t) = \ln(L)\Big|_{L_0}^{L} = \ln(L) - \ln(L_0) \quad \therefore \quad \epsilon(t) = \ln\left(1 + \frac{d(t)}{L_0}\right) \tag{4.55}$$

Assuming a constant strain rate ($\dot{\epsilon}_0$) such that $\epsilon(t) = \dot{\epsilon}_0 t$ and substituting into 4.55, the applied displacement equals

$$d(t) = L_0 \left(\mathrm{e}^{\dot{\epsilon}_0 t} - 1\right) \tag{4.56}$$

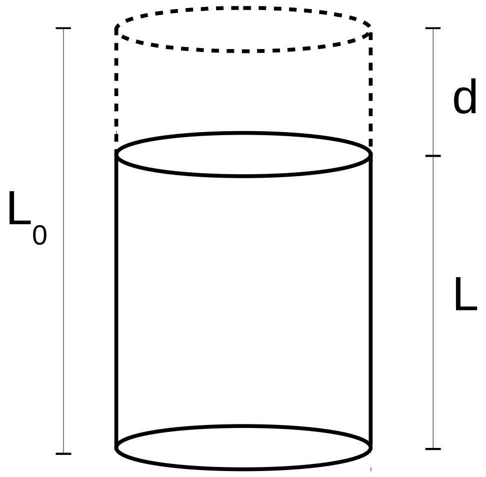


Figure 4.6: Geometry of the Global Displacement Function

where $d(t)$ is negative in compression. Figure 4.7 shows a comparison between the global applied Hencky (ϵ) strain and the strain measured during the SHPB experiment. The SHPB results are interpreted in a small strain context, which accounts for the difference between the applied Hencky strain (blue line) and the data (black line). For reference, the red line is the applied engineering strain given the same constant strain rate calculated from the small strain displacement function (equation 4.57).

$$d(t) = L_0 \dot{\epsilon}_0 t \tag{4.57}$$

The measured axial strain in the dry sand specimen increased linearly over approximately 0.05 to 0.75 ms meaning the constant strain rate assumption was valid over this time domain. For numerical simplicity and to push the model's ability to handle larger strains, the displacement function was applied over the full 1.0 ms.

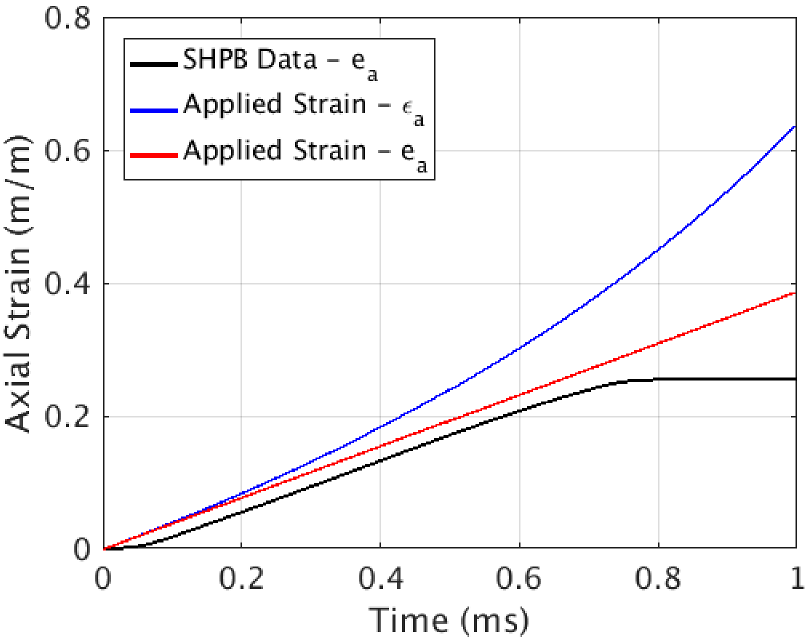


Figure 4.7: Comparison Between Applied Displacement and Strain in SHPB Experiment

4.3.2 Representative Volume Element of Dry Colorado Mason Sand

As the FEM mesh deforms given the applied global displacement, the Hencky strain (ϵ) at each element's integration points is calculated from the deformation gradient. The Hencky strain is then used to calculate the displacement that needs to be applied to the top of the DEM RVE in order to match the strain over the given global timestep (Δt) in a procedure that is similar to the work of Guo and Zhao (2014). Figure 4.8 shows the FEM mesh and a rendering of one of the DEM RVEs. The RVE contains 644 ellipsoidal particles and has approximately the same grain size distribution as the dry Colorado Mason Sand that was used in the SHPB experiment as it was generated using synchrotron micro-computed tomography (SMT) images of the sand (Zhang et al., 2017). Although no formal analysis of the required RVE size was done, it was assumed from past modeling experience that a roughly 600 particle assembly would accurately represent the material (Zhang and Regueiro, 2015). While it is possible that the assembly is too small to accurately represent the material, it is more likely that the RVE is too large. Considering the heuristic developed by Ehlers et al. (2003), the RVE's dimensions are all greater than 5x than the average particle diameter, which is considered to be "enough" particles for an RVE. This is visually corroborated on figure 4.9 where it's clear that the dimensions of the RVE can easily fit five average sized particles. Also, it is possible to introduce statistical-variability in DEM RVE by introducing probability density functions on particle size, orientation, and material properties. These differing RVEs could be used at different integration points in an FEM-DEM model or used in a unit cell sense to populate a statistically variable data set for use in models like Andrade et al. (2011).

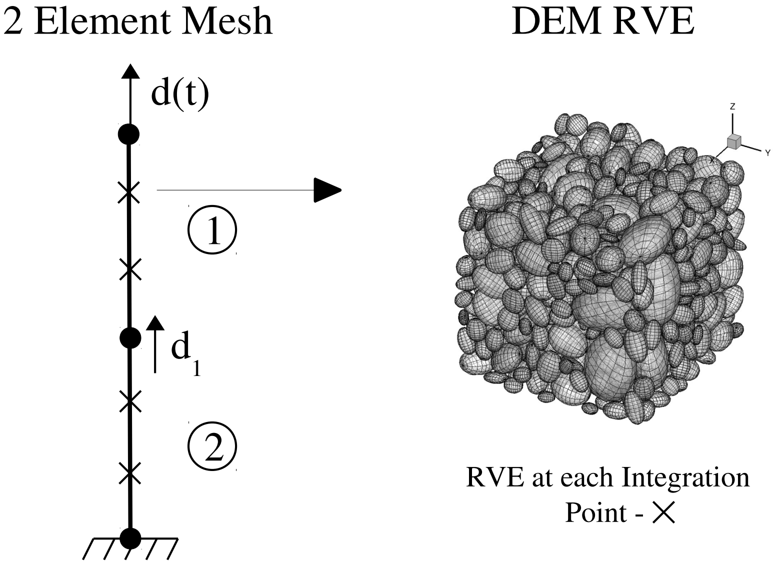


Figure 4.8: FEM to DEM RVE

Figure 4.9 contains a closer look at the DEM RVE and the geometry of the applied displacement to the RVE. The height of the RVE at a given time is $H^{\text{RVE}}(t) = H_0^{\text{RVE}} - d^{\text{RVE}}(t)$ where H_0^{RVE} is the initial height of the DEM assembly and $d^{\text{RVE}}(t)$ is the applied displacement to the RVE. The Hencky strain in the RVE is, by definition, equal to $\epsilon = \ln\left(\frac{H^{\text{RVE}}}{H_0^{\text{RVE}}}\right)$. After substituting the $H^{\text{RVE}}(t)$ equation into the definition of the Hencky strain, the displacement applied to the top of the RVE is derived as

$$d^{\mathrm{RVE}}(t) = H_0^{\mathrm{RVE}}\left(1 - \mathrm{e}^{\epsilon(t)}\right) \tag{4.58}$$

where the Hencky strain ($\epsilon(t)$) is provided to the DEM through the FEM calculation and $d^{\mathrm{RVE}}(t)$ is applied to the DEM RVE.

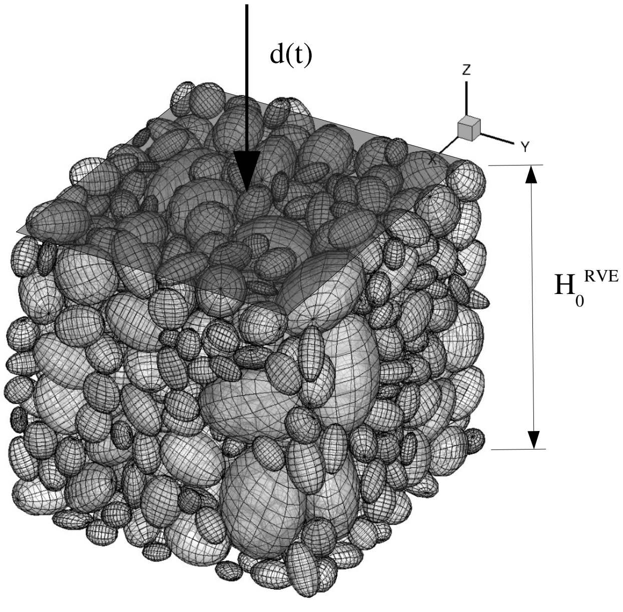


Figure 4.9: Applying Displacement to the DEM RVE

Ellip3D was originally designed to take in the initial positions of all the particle centroids and the location of any rigid boundaries, which are described in two text files titled `input_particle_file` and `input_boundary_file`, respectively, and output into another file the updated particle positions given the deformation of the assembly. In the case of running a uniaxial strain problem, the displacements of the particles are bounded by four rigid walls (the sides in the $\pm x$ and $\pm y$ directions on figure 4.9), and displacements are prescribed for the top and bottom ($\pm z$ directions). For the hierarchical multiscale modeling application, ellip3D was modified to output the updated particle positions into a new input file. Over each Δt prescribed by the FEM, the algorithm makes a call to ellip3D to update the stress, which means that the output particle positions from the previous timestep are needed to determine how the RVEs deform over the current timestep. Similarly, the particle positions at the end of the current timestep are then used as starting points for the following timestep.

4.3.3 FEM-DEM Modeling Formulation Specifics

With the global and RVE displacements defined, ellip3D and the finite element implementation were integrated into one code. There are many similarities between the implicit and explicit FEM-DEM specifics, and the few differences will be noted. However, as the implicit formulation is more complicated, this section is from the perspective of the implicit formulation.

Due to the Newton-Raphson iteration scheme, ellip3D is called many times for each FEM integration point over each global timestep as the FEM solution converges. To prevent the output of the previous iteration from being used as an input to the current iteration, the ellip3D outputs from the previous timestep are saved at the beginning of each Newton-Raphson iteration. These saved files are then used as a fresh input to each successive iteration. Once the solution converges, the ellip3D outputs are then the correct outputs over the Δt, and the FEM code moves to the next global timestep. In the explicit formulation, there is no iteration and therefore no need to save the input files at the beginning of the timestep.

In addition to the final output file, ellip3D also originally allowed for a number of “snapshot” files to be output that contained the particle location information at prescribed time intervals. These intermediate particle locations could then be used to construct videos of the assembly deforming. Because ellip3D is called many times over the hierarchical upscaling simulation, it was also modified to take an input parameter that specifies for what Δt it was being called. Then, for each Δt, the total number of snapshots was reduced to one, such that the hierarchical upscaling code outputs the particle locations only once at the end of each ellip3D call.

As shown on figure 4.7, the SHPB hierarchical upscaling simulation was run for a total of 0.001 seconds. Through experimentation, it was found that $\Delta t = 1 \times 10^{-7}$ seconds provided reasonable

Newton-Raphson convergence given an absolute and relative tolerance of 1×10^{-4}. Therefore, the time domain was discretized into 10,000 total timesteps. For the DEM, the stable timestep for the explicit time integration was found to be 2×10^{-10} seconds given the small grain-size of the Colorado Mason Sand. Then, for each global timestep, the DEM needed to run 500 timesteps in order to keep the time at the two scales the same $(500 \times (2 \times 10^{-10}) = 1 \times 10^{-7})$. The applied displacement to the RVE was assumed to be applied linearly over the total number of DEM timesteps as each of these displacements only applied a small strain to the RVE given the relatively small Δt. For the explicit formulation, Δt was matched with the stable DEM timestep of 2×10^{-10}.

Ellip3D was also originally designed to run a basic openMP parallelization scheme where the problem domain can be further discretized and run on individual threads of a computer node. For personal computers with multi-core processors, ellip3D can run on all of the computer's cores via this parallelization scheme through specifying the number of threads on which to run the code. For supercomputers, ellip3D can monopolize all the resources of a single compute node through this feature. After considering all the updates necessary to run ellip3D inside the hierarchical upscaling algorithm, each call to the problem takes the following form

```
./ellip3d topDisp botDisp numCalls [numThreads]
```

where `ellip3d` is a call to the ellip3D executible located in the current directory (./ on unix systems), `topDisp` is the displacement applied to the top of the RVE, `botDisp` is the displacement applied to the bottom of the RVE, `numCalls` is the number of the global timestep, and `numThreads` is the number of openMP threads on which to run the code. The brackets specify that `numThreads` is an optional parameter, and ellip3D runs serially if left blank.

More importantly, to dramatically increase the overall computational efficiency of the code, the code was parallelized with help from the Department of Defense High Performance Computing's Productivity Enhancement, Technology Transfer, and Training (PETTT) program. With the help of expert computer programmers, a C++ implementation of the hierarchical multiscale model was developed. Because each integration point is a separate calculation, FEM models naturally lend themselves to parallelization. In the new parallization scheme, the DEM model at each integration point is sent to a different compute node and then the results from each calculation are assembled on a master node that is running the FEM model (see figure 4.10). This utilizes both the openMP parallelization already built into ellip3D with a new MPI parallelization scheme to utilize more compute nodes. With all four integration points running in parallel, the overall runtime was decreased to nearly 25% of the original serial computation.

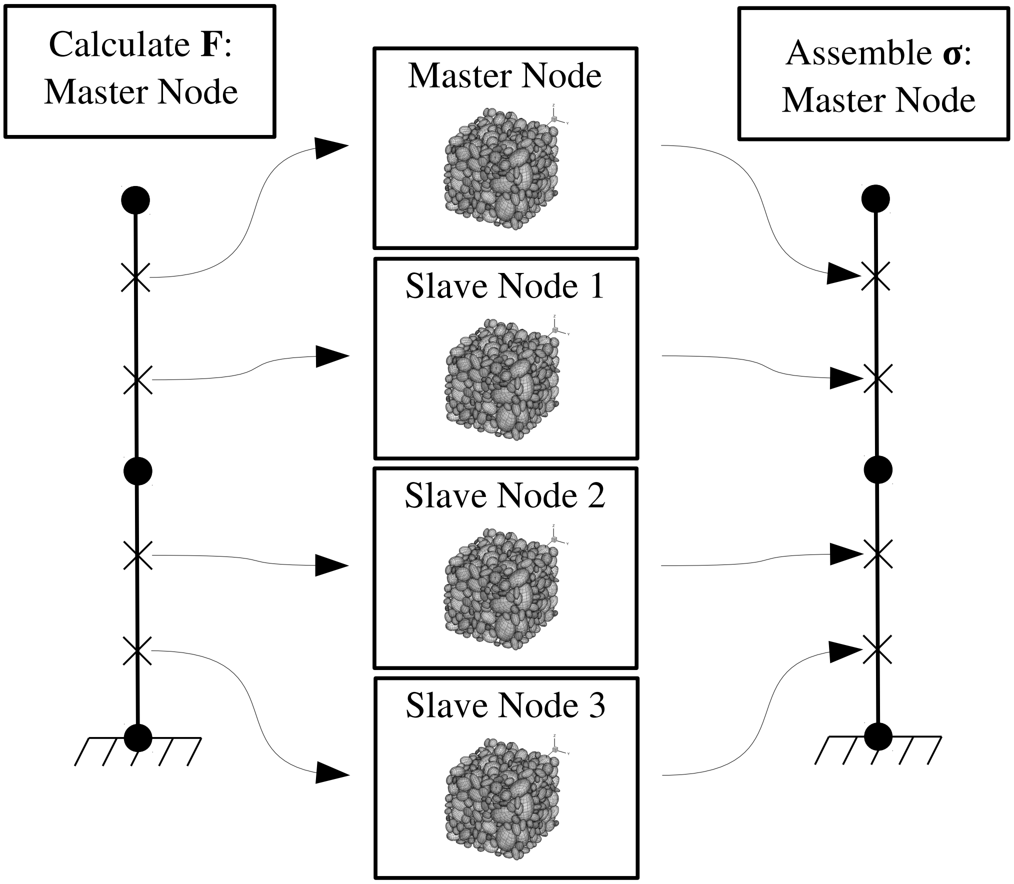


Figure 4.10: FEM-DEM Parallelization Scheme

As mentioned, each FEM integration point has an associated DEM particle assembly. All of the RVEs begin with the same particle assembly, but each is allowed to evolve separately given the global deformation. Each integration point could be assigned a different DEM particle assembly to model different sand microstructures, but only one initial assembly was used in these simulations for simplicity. Because ellip3D is designed to input files with specific names, the FEM code creates a unique output folder for each RVE designated by the specific element and integration point within the element that an RVE represents. As the FEM code iterates through the element loop and nested integration point loop, it changes the current directory to the directory containing the specific RVE's information and calls ellip3D from this particular location. This also allows for individual videos to be created out of the snapshots for each RVE. The FEM code then retreats back to the base directory to output the global variables. The global displacement, velocity, acceleration, stress, strain, and internal state variable information for each timestep, node, and integration point (where applicable) are stored in text files at the end of each timestep. They are stored at the end of each timestep rather than at the end of the overall simulation so that the results of the simulation can be checked before it has completed.

Occasionally, one of the DEM RVEs would experience a particularly large deformation from a major particle rearrangement such as a large particle finally slipping past another. During these events, the Newton-Raphson iteration often failed to converge causing the simulation to fail. As a result, a simple timestepping feature was added to the macroscale model. If the Newton-Raphson iteration didn't converge in five iterations, the global timestep was set to $\Delta t = 1 \times 10^{-8}$ seconds or $\frac{1}{10}$ of the original, and the simulation was restarted replacing the original global timestep with ten of the updated timesteps. If $\Delta t = 1 \times 10^{-8}$ seconds failed to converge, the timestepping algorithm would again divide the global timestep by ten and restart the simulation. None of the individual ellip3D outputs from these smaller timesteps were saved; only the final output representing the deformation over $\Delta t = 1 \times 10^{-7}$ seconds, the original global timestep. As noted before, saved versions of the inputs to ellip3D for each of the timesteps were needed to avoid over deforming the particle assembly over each successive Newton-Raphson iteration.

4.4 SHPB Simulation Results

A series of hierarchical multiscale modeling simulations were completed modeling results from a split Hopkinson pressure bar experiment conducted on dry Colorado Mason Sand by Dr. Hongbing Lu at the University of Texas at Dallas. Figures 4.11 and 4.12 show a stress time history and stress-strain response measured in the specimen during the experiment plotted in geomechanics convention (positive in compression). The sand specimen not only experiences large stresses peaking at approximately 130 MPa, but a high strain rate around 0.35 (m/m)/ms and large deformation with a final axial strain at approximately 0.27 m/m. Although the exact strain time history was not matched in the hierarchical multiscale modeling simulations (see figure 4.7 and discussion in section 4.3.1), these results were the primary data to which the simulations were compared.

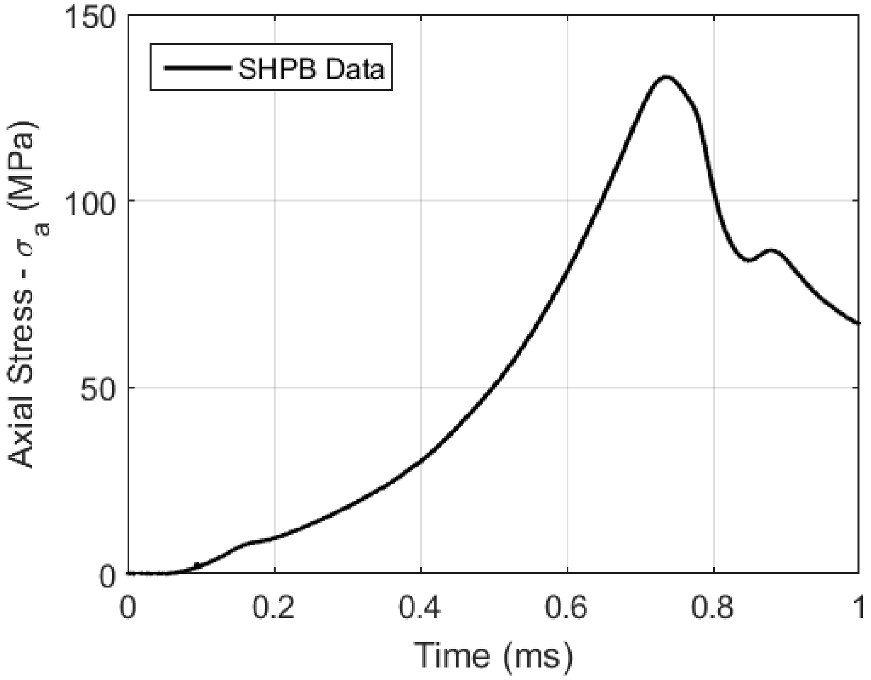


Figure 4.11: Stress-Time History of the SHPB Experiment on Dry Colorado Mason Sand

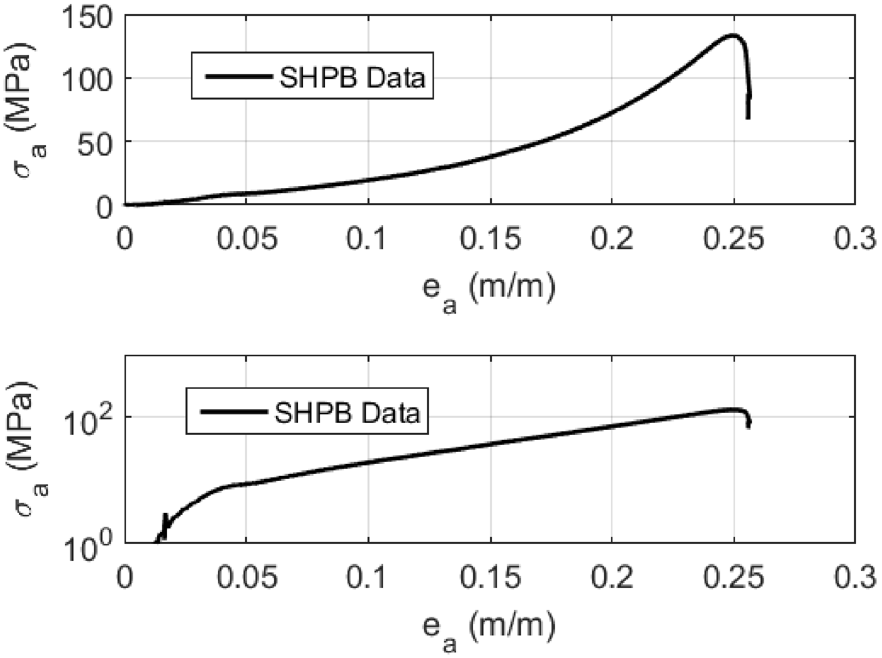


Figure 4.12: Stress-Strain Response of the SHPB Experiment on Dry Colorado Mason Sand

4.4.1 Implicit FEM-DEM Modeling Results

The first hierarchical multiscale modeling simulation of the SHPB test was run on the Army Research Laboratory, Department of Defense supercomputer Excalibur. Excalibur has 3,098 total compute nodes with each node containing 32 processors. For simplicity, it was assumed that the full displacement in a given global timestep was applied to the top of the DEM RVE - i.e. `topDisp` was set to the full RVE displacement given the experienced Hencky strain and `botDisp` was set to zero. A full list of the material parameters used in the DEM RVE simulations can be found in table 4.1. In particular, the shear friction coefficient (μ) and contact damping ratio (ξ) used in the DEM contact model were set to 0.7 and 0.8, respectively. Although these values intuitively appear large, they were calibrated to mimic the coefficient of restitution for the Colorado Mason Sand. The sand is approximated as ellipsoids, so these large values account for additional angularity and interlocking that would exist between sand grains.

Results from the hierarchical multiscale modeling simulation can be found on figures 4.13, 4.14, and 4.17, which compare stress-time histories, strain-time histories, and stress-strain relationships between the simulation and experimental results. Because of differences in the definitions of the applied strain and the strain measured during the SHPB experiment, it is unsurprising that the strain time histories from the first integration point of both elements do not match the experimental results, but the applied and simulation output Hencky strains do match after some initial waves. The model includes a full dynamic implementation of the balance of linear momentum, so it is also unsurprising that without any macroscale damping waves develop in the FEM model. There are no dissipative terms at the macroscale, so it's likely that the interparticle friction and contact damping in the DEM model caused the waves to dampen.

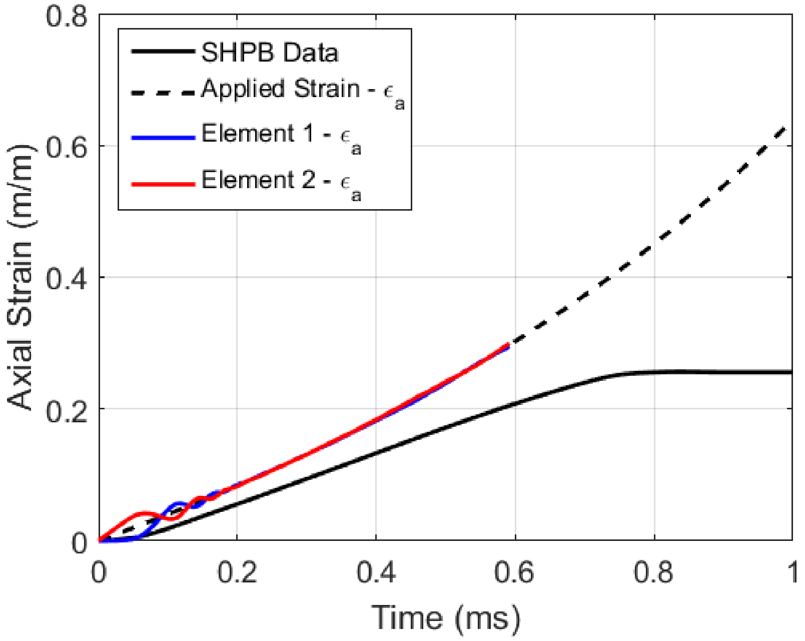


Figure 4.13: Strain Time History Comparison Between the SHPB Experiment and Implicit FEM-DEM Hierarchical Upscaling Simulation

The Cauchy stress-time histories from the simulation for the first integration point of both elements as compared to data from the SHPB experiment plotted in two different scales can be seen on figure 4.14. The dynamic response of the FEM model is more pronounced than in the strain results, and it's easier to track the dissipation as the waves more clearly shrink in amplitude with time. When compared to the Cauchy stress results of the experiment, the stress time histories begin to diverge around approximately 0.3 ms. It is believed that the cause of the divergence is grain crushing.

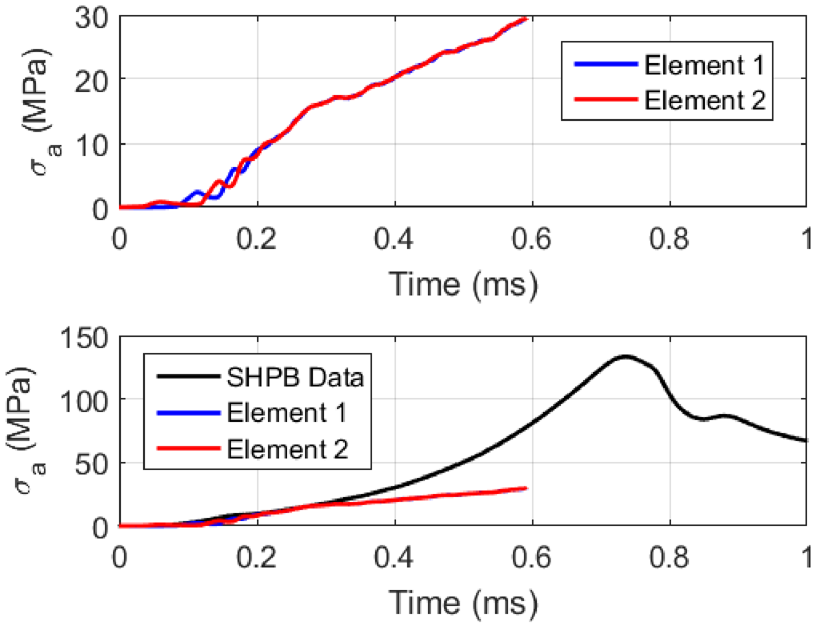
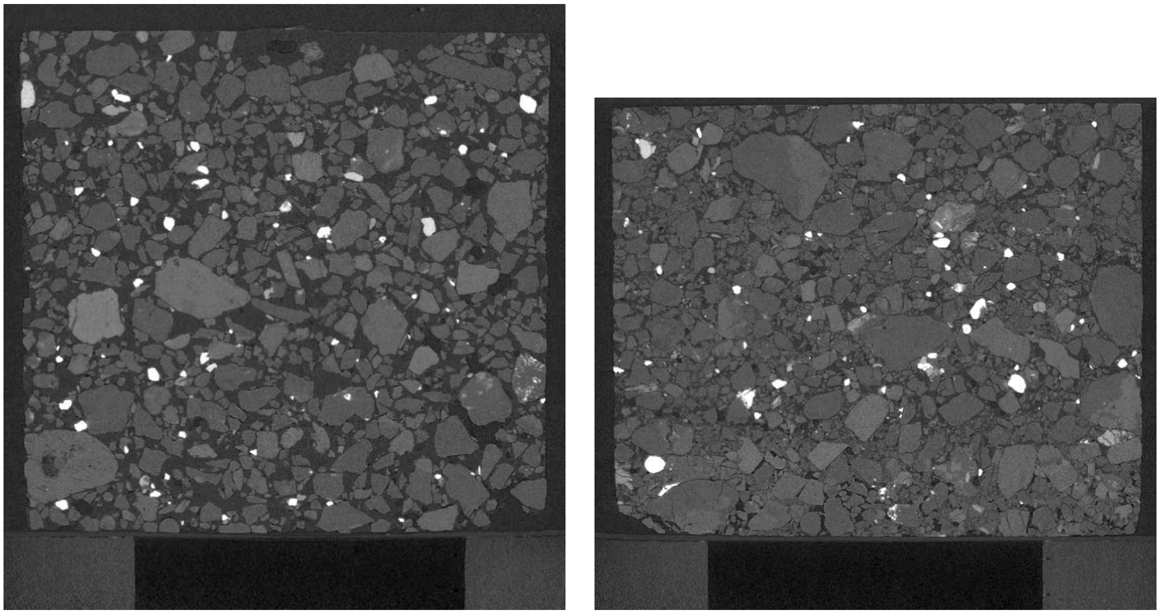


Figure 4.14: Stress Time History Comparison Between the SHPB Experiment and Implicit FEM-DEM Hierarchical Upscaling Simulation

Figure 4.15 show two synchrotron micro-computed tomography (SMT) images of SHPB results on Colorado Mason Sand performed by Drs. Khalid Alshibli and Andrew Druckrey of the University of Tennessee at the Advanced Photon Source, Argonne National Laboratory for the Soil Blast MURI project. The impacted specimen (figure 4.15(b)) is an image from the exact specimen from which the "SHPB Data" was taken, and the intact image (figure 4.15(a)) is of a different specimen at the same initial density. It was logistically impractical to obtain images of the same specimen before and after a SHPB test. First, it is clear that during SHPB experiments on Colorado Mason Sand the volume of voids significantly decreases, but a careful inspection of the impacted image also reveals that a number of particles have fractured or been pulverized into smaller grains. Figure 4.16 shows the grain size distribution before and after a SHPB experiment on silica sand.

Although a different material, the shift of the grain size distribution to the finer direction is further evidence to expect grain crushing in SHPB tests. Because of this evidence, it was believed that given the current DEM model's inability to account for particle fracture, the true behavior of the Colorado Mason Sand could not be captured. As a result, further simulations were performed with DEM RVE's containing a particle fracture model that was also developed for the Soil Blast MURI Project.



(a) Intact Specimen

(b) Impacted Specimen

Figure 4.15: CT Images of Colorado Mason Sand Specimens Before and After SHPB Experiments

Figure 4.17 compares the stress-strain response of the SHPB data to the Cauchy stress versus engineering strain and Cauchy stress versus Hencky strain response of the hierarchical upscaling simulation. The differences between the experimental and modeling results are due to the applied strain differences and divergence between the stress results likely caused by grain crushing, as seen more clearly on figures 4.13 and 4.14.

As can be seen on both figures 4.13 and 4.14, the implicit hierarchical multiscale model did not run to completion; the time histories terminate at approximately 0.6 ms. This is not due to convergence issues in the Newton-Raphson iteration, but because that was as far as the simulation

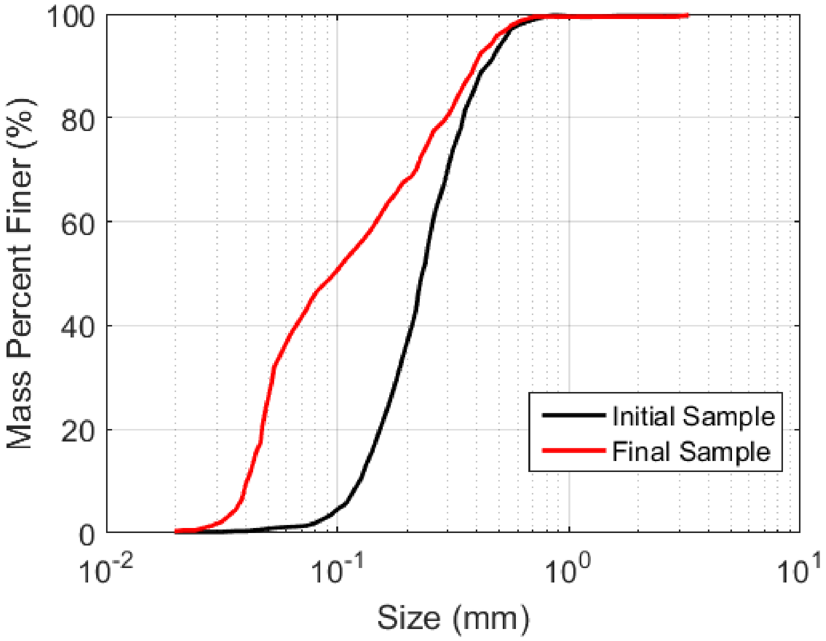


Figure 4.16: Grain Size Distribution Curves Before and After a SHPB Experiment on Silica Sand Adapted from Luo et al. (2011)

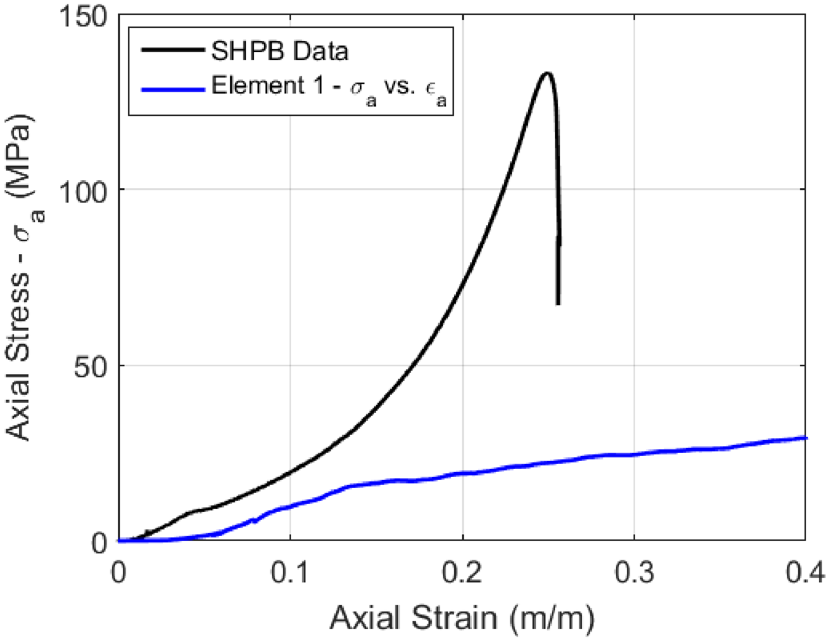


Figure 4.17: Stress-Strain Response Comparison Between the SHPB Experiment and Implicit FEM-DEM Hierarchical Upscaling Simulation

reached over the two months it ran. One of the primary reasons for using a hierarchical multiscale model is increased computational efficiency over direct numerical simulations, and although it is likely that the simulation is running faster than many exclusively DEM softwares would given the same problem, for all practical purposes, the simulations currently run too long. Through simply watching the outputs update as the simulation progresses, it is immediately clear that the FEM code's computational time is negligible compared to the DEM. As a result, the time integration scheme was changed from the implicit trapezoidal rule to the explicit central difference method. The overall length of the global timestep was reduced significantly greatly increasing the overall number of timesteps, but the explicit scheme does not require a Newton-Raphson iteration. By removing the iteration over each timestep, the overall number of DEM timesteps that need to be run was reduced.

4.4.2 Explicit FEM-DEM Modeling Results

With the explicit FEM-DEM model completed, the same SHPB experiment that was modeled using the implicit FEM-DEM formulation was again simulated. The axial strain and stress time history as well as the stress-strain curve can be found on figures 4.18, 4.19, and 4.20, respectively. Unlike the implicit model results, which were terminated at approximately 60% completion, the explicit model was able to run to full completion in less than two full weeks. The result of the change from an iterative solution method was a greater than 75% reduction in the total time required to run the simulations from the implicit version of the code. This reduction is even greater if you consider that the implicit simulation wasn't allowed to fully finish, although it's difficult to estimate exactly how much more time the implicit version would have required.

As can be seen on figure 4.18, there is a slight deviation from the element strain results from the applied strain curve around 0.6 ms. The "standard" Excalibur job queue (Excalibur uses a portable batch system (PBS) to manage its computing resources whereby researchers submit "jobs" to a queue that is managed by a PBS job scheduler) has a walltime (maximum number of hours allowed to individual researchers per job) of one week. The FEM-DEM model therefore

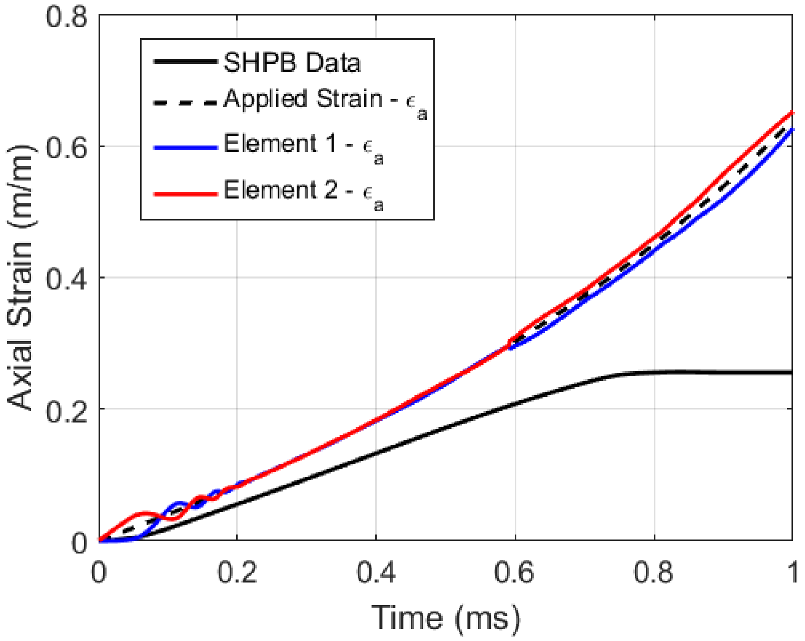


Figure 4.18: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation

needed to be restarted requiring re-initializing the kinematics and correctly managing the DEM outputs to feed the restarted FEM model the correct inputs. The deviations on figure 4.18 are the results of a minor accuracy issue re-initializing the kinematics. The displacement results output to the log file used to track the simulation's progress were to fewer significant figures than the displacement expected by the simulation. This resulted in a slight "jump" between the kinematics in the simulation at the walltime and the re-initialized kinematics upon the restart. The overall results likely did not suffer due to the relatively small difference between the displacement values, although the issue is noticeable.

As expected, however, an appreciable increase in the stiffness of the DEM assemblies was not found past the 60% completion mark. The gradual increase in stress continued roughly linearly through the end of the applied strain, unlike the increase in stiffness shown in the SHPB experimental data. Figure 4.21 shows the DEM assembly from the first integration point of the first element before and after the FEM-DEM SHPB simulation. First, the applied axial strain at t = 1.0 ms

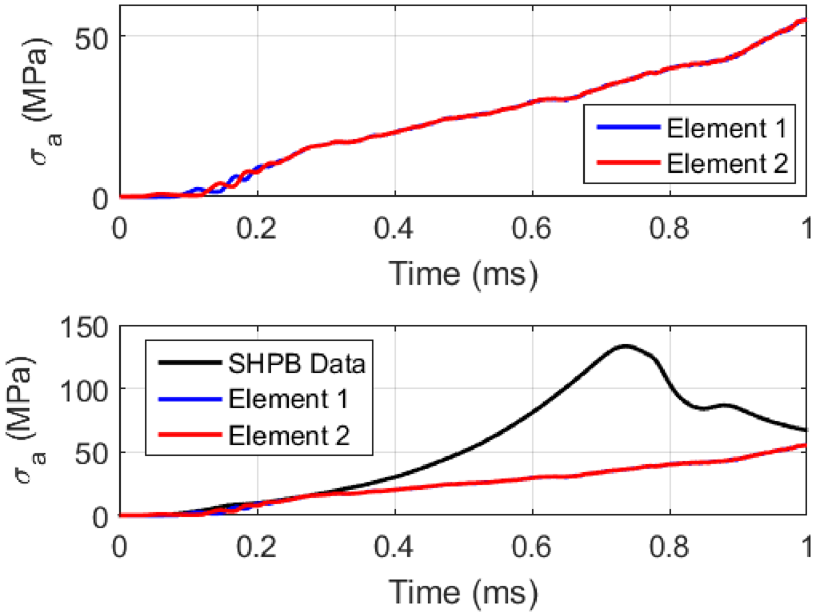


Figure 4.19: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation

was over 60%, which is visually corroborated by figure 4.21(b). Second, the DEM assemblies were deformed by physically moving the bounding walls the assemblies (see figure 4.9). Contact forces in ellip3D are calculated from the Hertz-Mindlin contact model, which relies on the slight overlap of particle-to-particle and particle-to-boundary contacts to calculate the interparticle contact force and particle-to-boundary contact force. If the particle-to-boundary overlap becomes too large (if the boundary passes over the centroid of the particle), the particle no longer is considered in contact with the wall causing the particle to "float" in space. Once outside the wall boundary, the particles remain suspended in space for the remainder of the simulation and other particles are allowed to fill the space they once occupied in the particle assembly. A few examples of these particles can be seen in figure 4.21(b). Although this undoubtedly decreased the homogenized stress calculated in the DEM assembly, the loss of a handful of particle-to-particle contacts relative to the total number of contacts in the assembly likely resulted in only a slight decrease in the stress results.

Figure 4.21(b) also shows a few unphysically overlapped particle-to-particle contacts; al-

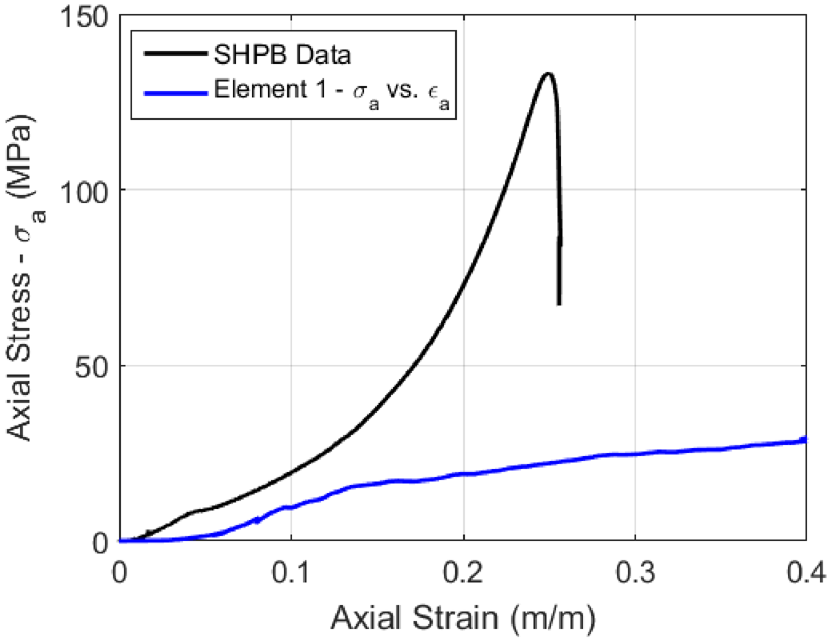
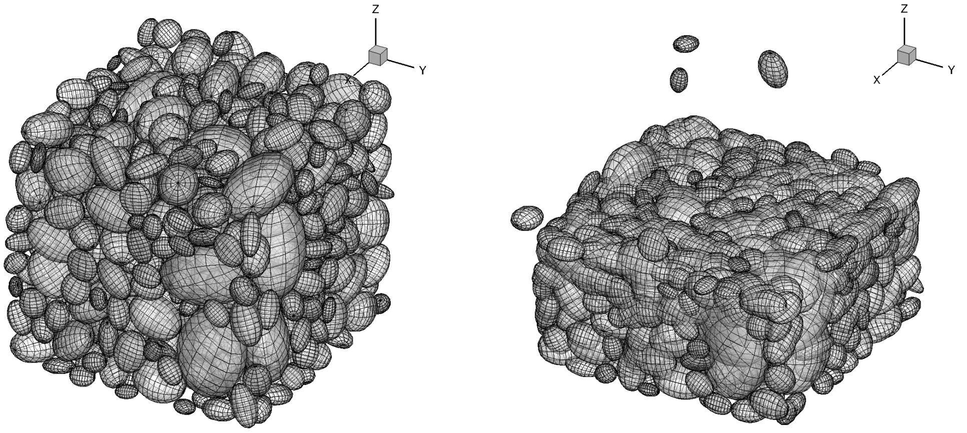


Figure 4.20: Stress-Strain Response Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation

though, it should be noted that the slightly translucent visualization of the particles makes the particles seem more overlapped than they actually are. In the Hertz-Mindlin contact model, normal contact force (P) is proportional to the overlap distance (ρ) by $P \propto \rho^{3/2}$. The stress is calculated by summing the distance between particle centroids multiplied by the contact force over the total number of particle contacts. Therefore, overlapped particles experience greater contact force while simultaneously decreasing the homogenized stress by being physically closer to one another. The two opposing variables make it unclear the effect the unphysical particle overlaps have on the homogenized stress calculation. Regardless, allowing the particles to fracture, which subsequently allows the DEM assemblies to densify via a decrease in the void ratio, will likely allow the increase in stiffness needed to more accurately model the experimental data.

Finally, it's important to remember that all of the presented results were generated from a hierarchical multiscale model with a two element macroscale finite element mesh. Prior to parallelization, the simple mesh was used for primarily for computational tractability, but post par-



(a) $t = 0$ ms

(b) $t = 1.0$ ms

Figure 4.21: DEM Assembly Before and After the Explicit FEM-DEM Simulation

allelization and provided enough compute nodes are available, it's possible to do a formal mesh refinement study. However, it was determined that this work would focus on enhancing the DEM RVE with a particle fracture model and adding additional complexity to the simple models rather than conduct a mesh refinement study.

4.4.3 Explicit FEM-DEM Modeling with Fracture Results

The particle fracture model used in this work was also developed as a part of the Soil Blast MURI Project by Drs. Boning Zhang and Richard Regueiro at the University of Colorado Boulder. Dr. Zhang also worked on the model with researchers at Lawrence Livermore National Laboratory, Livermore, CA (Zhang et al., 2015). The model was incorporated into a newer version of ellip3D that replaced the ellipsoidal particles with more adaptable poly-ellipsoids called polyEllip3D (also developed for the Soil Blast MURI Project). The eight octants of poly-ellipsoids are each its own ellipsoid that are combined to create a smooth, but non-symmetric shape (Peters et al., 2009) that is better at capturing the shapes of sand grains than ellipsoids and certainly spheres.

The particle fracture model is comprised of two steps. First, particles are flagged as potential breakage candidates using two possible criteria. The first is the Hoek-Brown fracture criterion

(Hoek and Martin, 2014), which essentially defines a yield function controlled by two parameters σ_{f_c} and m_i

$$f = \sigma_1 - \left(\sigma_3 + \sigma_{f_c}\sqrt{m_i\frac{\sigma_3}{\sigma_{f_c}} + 1}\right) \tag{4.59}$$

where σ_{f_c} is the compressive strength and m_i is a material constant. Over each particle domain, an average continuum stress tensor is calculated, and the major (σ_1) and minor (σ_3) principal stresses (positive in compression) are input into the yield function. The particle is flagged for potential breakage if $f \geq 0$. The criterion is used to account for the differences between uniaxial and biaxial compression as particles are more likely to fracture in unconfined uniaxial compression than biaxial compression. However, the Hoek-Brown fracture criterion doesn't allow for particles to fracture under equi-biaxial compression. There should exist an applied isotropic compressive stress that induces particle crushing, so a second criterion, the local contact tensile stress fracture criterion, was added based on work by Timoshenko (1970) to account for this fracture mechanism. From Hertzian contact theory, the inter-particle contact force is calculated. The overlapping of particles causes tensile stress (σ_t) to develop locally at the contact site, which is calculated from the inter-particle contact force. If a particle experiences too much overlap with another particle, the calculated local tensile stress will be greater than the tensile strength of the material (σ_{f_t}), and the particle is flagged for potential breakage. Particles flagged by either fracture criterion are then split into eight new poly-ellipsoidal particles, one for each octant of the original poly-ellipsoid, that are bonded together.

In the second step of the particle fracture model, the once single particle breaks into up to its eight separate pieces if the bonds' fracture strength (σ_f) is exceeded. A Weibull function is used to introduce statistical variation in the particle's fracture strength depending on particle's size as larger particles have been found to fracture more easily. It should be noted after breakage, it is possible for the newly formed particles to have small radius of curvatures in parts of the particle domain. It is unclear at this time, and out of the scope of this research, if these small radius of curvature violate the assumptions of the Hertz-Mindlin contact model.

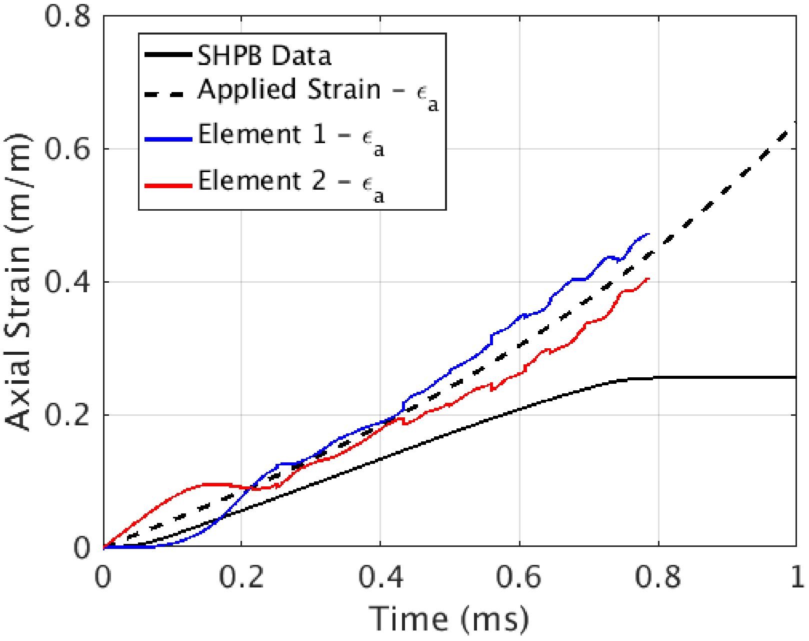


Figure 4.22: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation Including Particle Fracture

Figures 4.22 and 4.23 contain the strain and stress time histories from the same FEM-DEM SHPB simulation as the explicit model but including the particle fracture model. The strain results, relative to the implicit and explicit simulations, are less smooth, which is the result of the same accuracy issue described in the explicit simulation results. Unlike the explicit simulation that only needed to be restarted once, the addition of the particle fracture model significantly increased the computational time required to complete the simulation. The un-smooth nature of the strain results is the compounding of the same accuracy issue over the four separate restarts of the simulation. Despite the issue, the strain results still follow the applied strain curve relatively well.

The stress results for the explicit simulation including the particle fracture (figure 4.19) model were significantly lower (less axial compressive stress) and more stochastic than the explicit simulation (figure 4.23). These results are likely caused by a poorly calibrated fracture model. As can be seen on figure 4.24, at roughly 79% through the simulation (t = 0.785 ms), the entire top

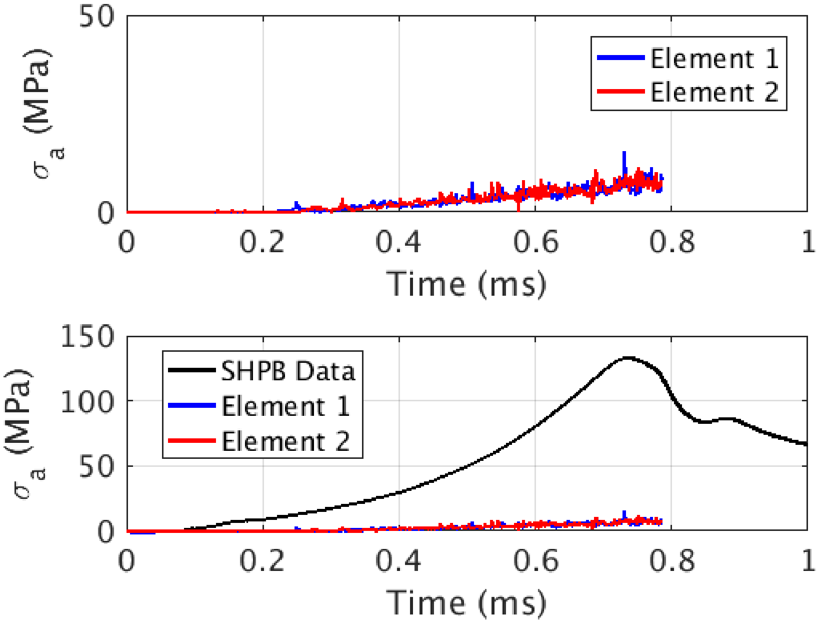
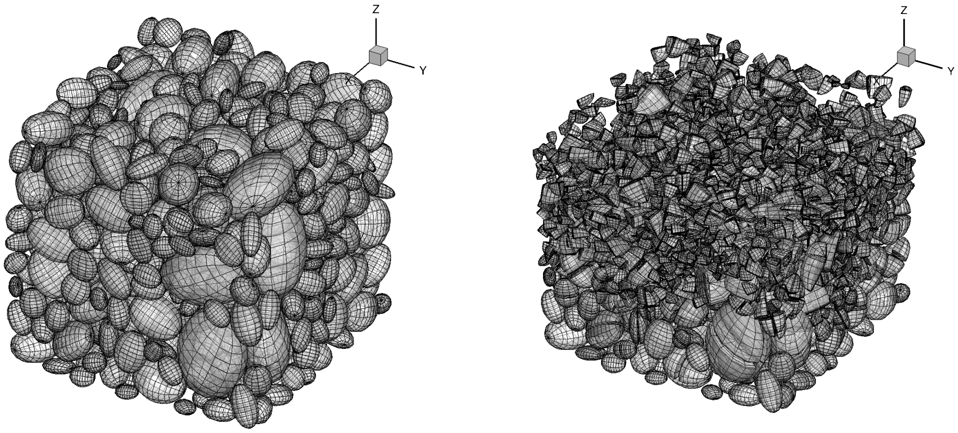


Figure 4.23: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation Including Particle Fracture



(a) $t = 0$ ms

(b) $t = 0.785$ ms

Figure 4.24: DEM Assembly Before and After the Explicit FEM-DEM Simulation Including Particle Fracture

layer of particles and most of the particles in the top half of the assembly had already been fractured. Rather than displacing the top layer of particles downward increasing the axial compressive stress through densification, the particles fractured instead. Any decrease in void ratio was local to the fracturing particle and therefore didn't translate into an increase in the homogenized axial compressive stress. As can be seen on figure 4.15, there is CT evidence that sand particles fracture during the SHPB experiment, but not to the extent that is seen on figure 4.24 at that time in the loading history. Also, due to the decrease in the fractured particles' average radii, it became easier for the boundary wall to pass over the centroid of the fracture particles. As discussed before, if the boundary passes over the centroid of a particle, the particle is no longer considered in contact with the wall and is left to float in space; therefore, it was easier for fractured particles to "leave" the assembly. Comparing figures 4.21(b) and 4.24(b), it is clear that under the same loading, many more particles "left" the DEM assembly with the inclusion of the particle fracture model. Unlike the explicit simulation, the loss of these particles likely does contribute to the lower axial stress results as a significant number of interparticle contacts were removed from the DEM RVE.

The values for σ_{f_c}, m_i, σ_{f_t}, and σ_f were all calibrated using force versus displacement data from unconfined uniaxial compression tests on single sand grains conducted by Dr. Alshibli's group at the University of Tennessee at Knoxville. However, the calibration was done using data from Elgin sand not Colorado Mason Sand. Considering that the fracture model allows for too many particles to fracture, a parameter sweep was run with larger (i.e. stronger) values for the four parameters. Also included in the parameter sweep were the contact damping (c_r) and coefficient of friction (μ) between particles. Lowering c_r and μ lowers the overall energy dissipation in the DEM assemblies allowing the particles to densify more quickly. Table 4.2 contains the parameters and their values for each of the eight tests completed for the parameter sweep the results of which can be found on figure 4.25. "Test 1" are the original parameter values.

A single DEM RVE was run for each of the eight tests of the parameter sweep with the fracture model and particle-to-particle contact parameters set as found in table 4.2. The top of the assembly was displaced 0.001 m linearly over the full 1 ms to achieve approximately 30% axial strain at the end of the simulation. The displacement applied to the DEM RVEs is calculated from the strain in the FEM for the hierarchical upscaling algorithm, so the displacement was applied more quickly in the parameter sweep tests. Also, as is apparent in the results on figure 4.25, none

Table 4.2: Particle Fracture Model: Parameter Sweep

Test	σ_{f_c} (Pa)	m_i	σ_{f_t} (Pa)	σ_f (Pa)	c_r	μ
1	200.54×10^7	32.4	2.72×10^7	4.13×10^7	0.7	0.8
2	200.54×10^7	32.4	2.72×10^7	4.13×10^7	0.1	0.1
3	200.54×10^{10}	32.4	2.72×10^7	4.13×10^7	0.7	0.8
4	200.54×10^7	32.4	2.72×10^{10}	4.13×10^7	0.7	0.8
5	200.54×10^{10}	32.4	2.72×10^{10}	4.13×10^7	0.7	0.8
6	200.54×10^7	32.4	2.72×10^7	4.13×10^{10}	0.7	0.8
7	200.54×10^{10}	32.4	2.72×10^{10}	4.13×10^{10}	0.7	0.8
8	200.54×10^{10}	32.4	2.72×10^{10}	4.13×10^{10}	0.1	0.1

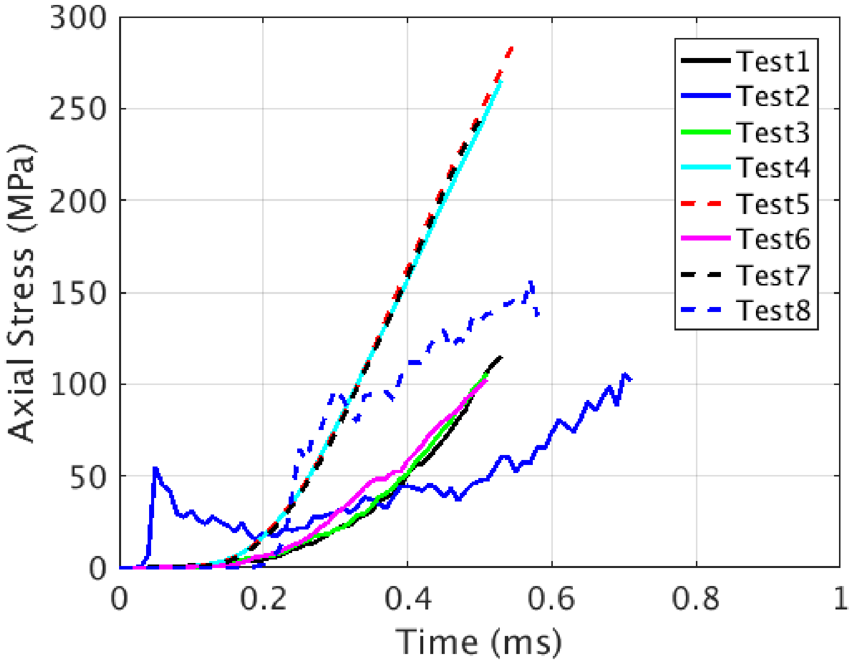


Figure 4.25: Fracture Parameter Sweep to Calibrate the Particle Fracture Model

of the simulations completed during Excalibur's allotted 7 day walltime.

Regardless, enough information was gathered to make a few generalizations regarding the importance of the parameters on the behavior of the DEM with the particle fracture model. First, lowering the interparticle contact parameters, as was done for tests 2 and 8, causes the axial stress results to appear more stochastic. Particles are able to slip past and bounce off each other more easily causing an increase in the number of contacts created and removed, which results in a more

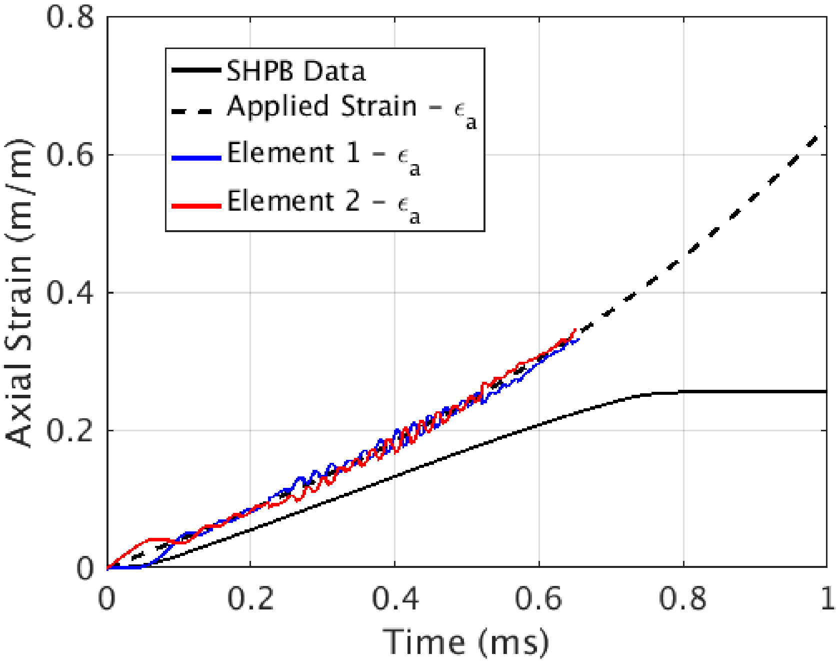


Figure 4.26: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation Including Particle Fracture with Calibrated Parameters

variable stress response. Second, significantly increasing σ_{f_c} and σ_f doesn't significantly affect the stress results. The stress history curves for tests 1, 3, and 6 as well as tests 4, 5, and 7 are approximately the same. The difference between the two sets of histories was an increase in σ_{f_t} suggesting that given the confined uniaxial strain loading condition, the local tensile stress flag is the most important parameter in determining the behavior of the DEM RVE. Given the findings of the parameter sweep, it was determined that test 8, which combined the lower contact parameters and increase in the local tensile stress flag, best matched the desired data (see figure 4.27). It is clear that many more parameter sweep tests should be run (and run to completion), but the primary goal of the sweep was to quickly find values that more closely matched the data than the original parameters.

Using the parameters from test 8, an updated explicit FEM-DEM simulation including the particle fracture model was run. Strain and stress time histories for the updated simulation can be found on figures 4.26 and 4.27. As expected, the reduction of the dissipative components of the

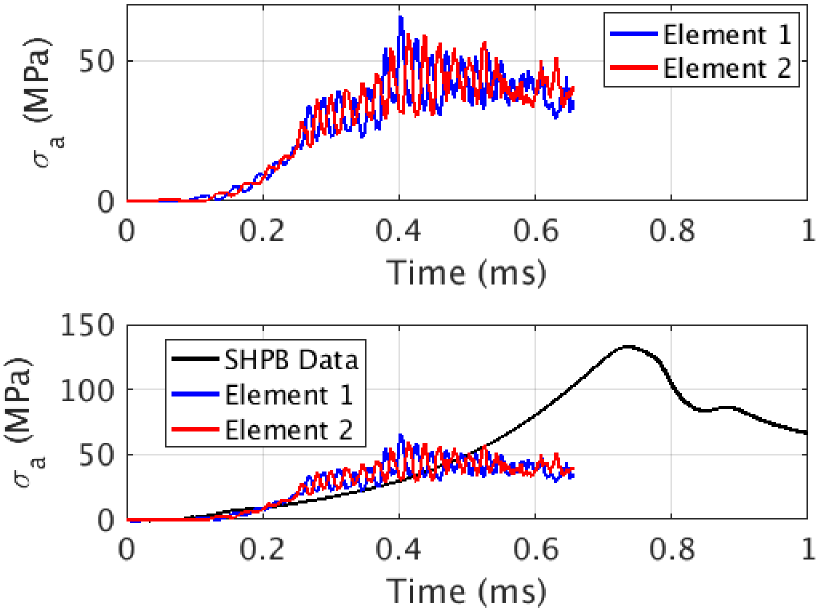
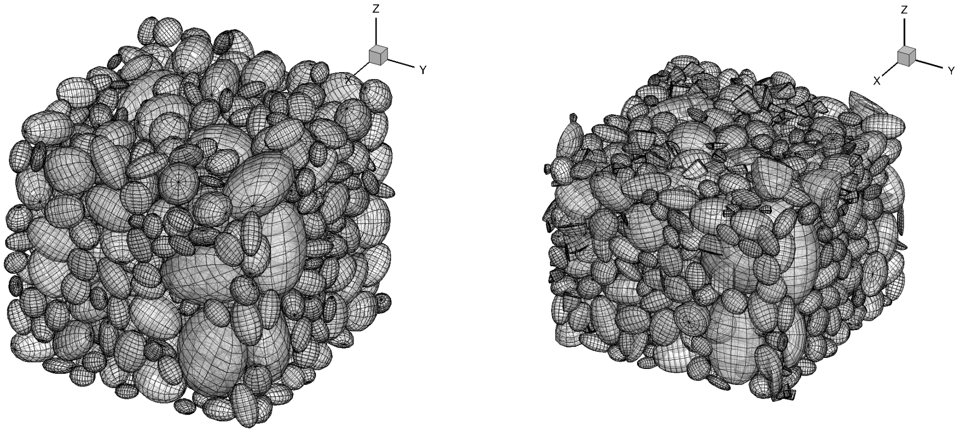


Figure 4.27: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation Including Particle Fracture with Calibrated Parameters

Hertz-Mindlin particle contact model (c_r and μ) resulted in more waves in both the strain and stress responses. The increases in the amplitude of the waves around $t = 0.25$ and $t = 0.36$ milliseconds were caused by the same accuracy issue upon restarting the simulation. Overall, the stress results more closely match the data from the SHPB experiment. The deformed DEM assembly (figure 4.28(b)) shows a significant decrease in the number of fractured particles over the original fracture parameters (figure 4.24(b)), but there are a handful of particles that have fractured as of $t = 0.655$ seconds. Like with the FEM-DEM simulation results using the original parameters for the fracture model however, there is still an issue with the smaller, fractured particles "leaving" the RVEs resulting in fewer particle-to-particle contacts and a reduction in the axial compressive stress. As can be seen on figure 4.28(b), no particles or fractured particles had escaped the particle assembly as of $t = 0.360$ ms and the axial compressive stress was increasing. Yet by $t = 0.655$ ms (figure 4.29), roughly the entire top layer of particles were no longer in the assembly and the axial compressive stress was decreasing.



(a) $t = 0$ ms

(b) $t = 0.360$ ms

Figure 4.28: DEM Assembly Before and During the Explicit FEM-DEM Simulation Including Particle Fracture with Calibrated Parameters

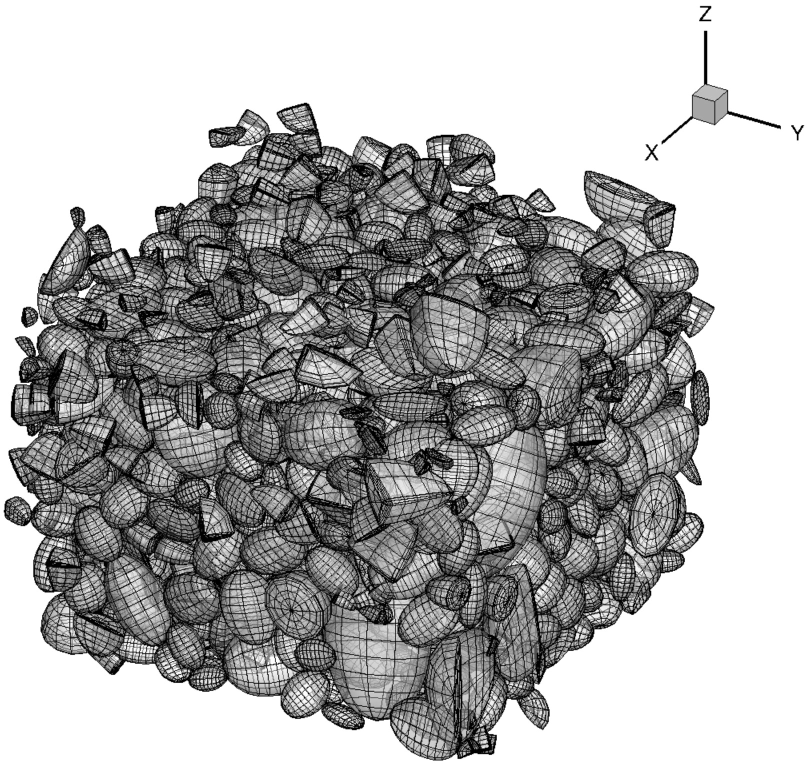


Figure 4.29: DEM Assembly After the Explicit FEM-DEM Simulation Including Particle Fracture with Calibrated Parameters ($t = 0.655$ ms)

4.4.4 Explicit FEM-DEM Modeling with Other DEM Boundary Conditions

In addition to the wall boundary conditions described in 4.3.3, two additional boundary condition formulations were tested using the FEM-DEM SHPB simulation. As described in Meier et al. (2009) and reiterated in chapter 2, both the Taylor-Voigt and Restricted Taylor boundary conditions were tested. An updated version of ellip3D was written where the input displacement ($d(t)$ in equation 4.56) was used to derive the displacement of each individual particle ($d_p(t)$) rather than the top wall's position. Using the geometry on figure 4.31 with an example particle highlighted in red, the displacement of each particle is

$$d_p(t) = \left(\frac{H_0^p}{H_0^{\mathrm{RVE}}}\right) d^{\mathrm{RVE}}(t) = H_0^p(\mathrm{e}^{\epsilon(t)} - 1) \tag{4.60}$$

where H_0^p is the initial height of particle p's centroid. For the Taylor-Voigt boundary conditions, the position of every particle in the DEM assembly was determined using equation 4.60, and for the Restricted Taylor boundary conditions, the position of only the top layer of particles was determined using the equation. The rest of the particles were allowed to displace as they would normally (through interparticle contact forces). Also, a second, fully periodic, DEM RVE was generated (figure 4.31) where multiple of the same assemblies could be seamlessly placed on any side of the RVE to generate a true locally periodic assembly.

Initial axial stress and strain time history results for both boundary conditions can be found on figures 4.32, 4.33, 4.34, and 4.35. First, neither simulation was allowed to finish as the addition of the periodic assembly and two new boundary conditions greatly increased the overall time required to run the simulation. Both sets of results were obtained after the simulations were killed after Excalibur's 7 day walltime was exceeded. By directly controlling the displacement of even a subset of the particles, the number of contacts and the force between those contacts greatly increased from the version with the wall boundary condition. Second, as predicted by Meier et al. (2009), the Taylor-Voigt boundary conditions result in overly stiff responses in the DEM RVEs, which in turn over-predict the axial stress. Interestingly, the Restricted Taylor boundary conditions

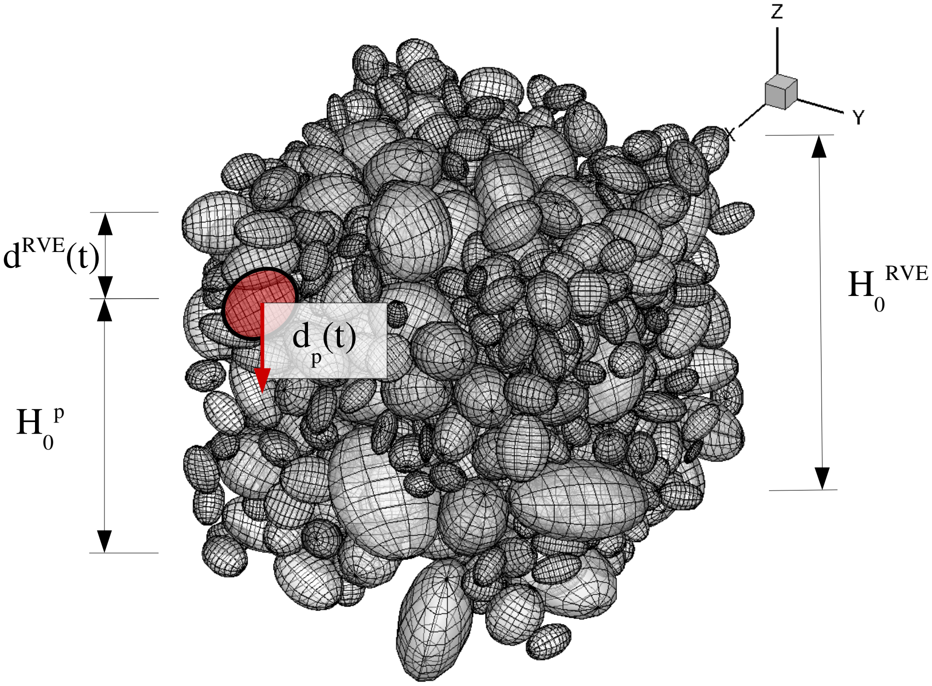


Figure 4.30: Applying Displacement to the Periodic DEM RVE

also over-predict the axial stress response. These results highlight a common issue with other hierarchical upscaling and DEM research, in general. It is rare that any simulation results are compared to experimental data, and where it might be true that Restricted Taylor boundary conditions better qualitatively predict results as asserted by Meier et al., the assumptions might not be better quantitatively. In the case of the highly dynamic, finite strain SHPB simulation, the better boundary condition option appears to be wall boundary conditions (as used by Guo and Zhao (2014)). Further investigation of these boundary conditions is suggested.

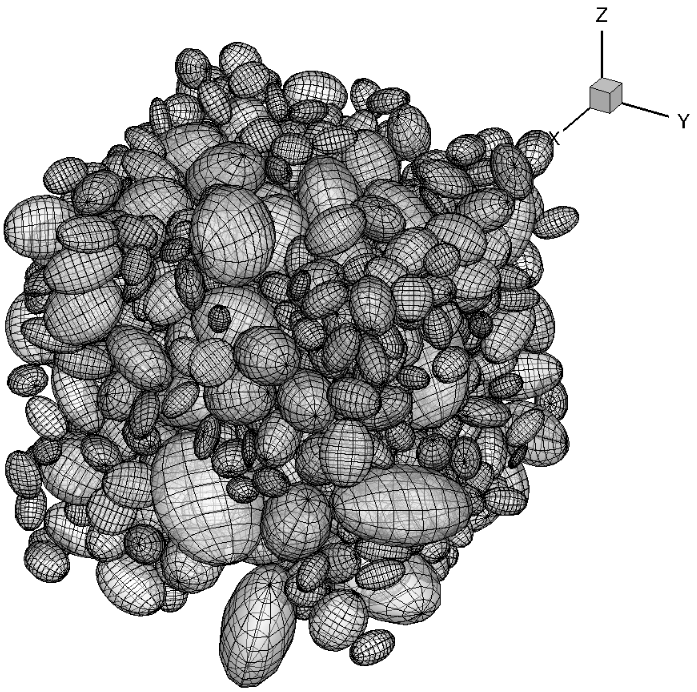


Figure 4.31: Periodic DEM Assembly

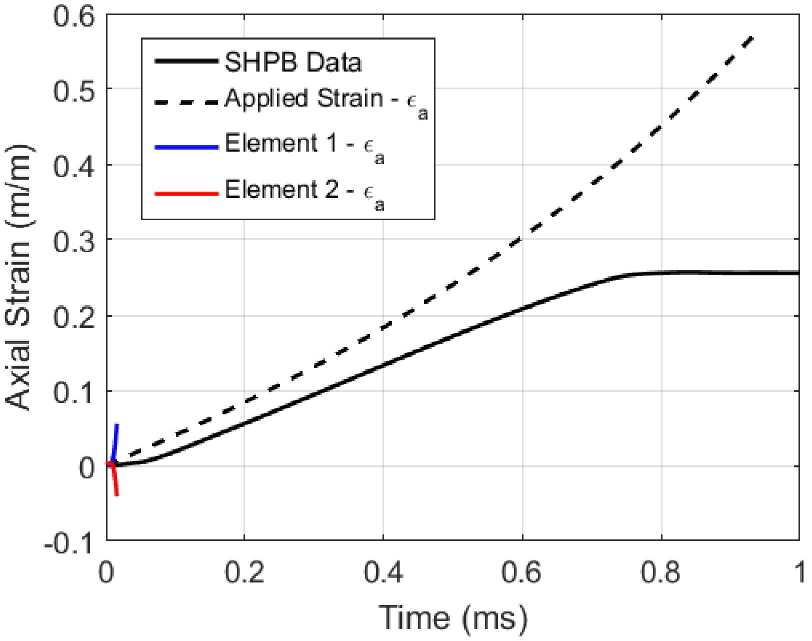


Figure 4.32: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Taylor-Voigt Boundary Conditions

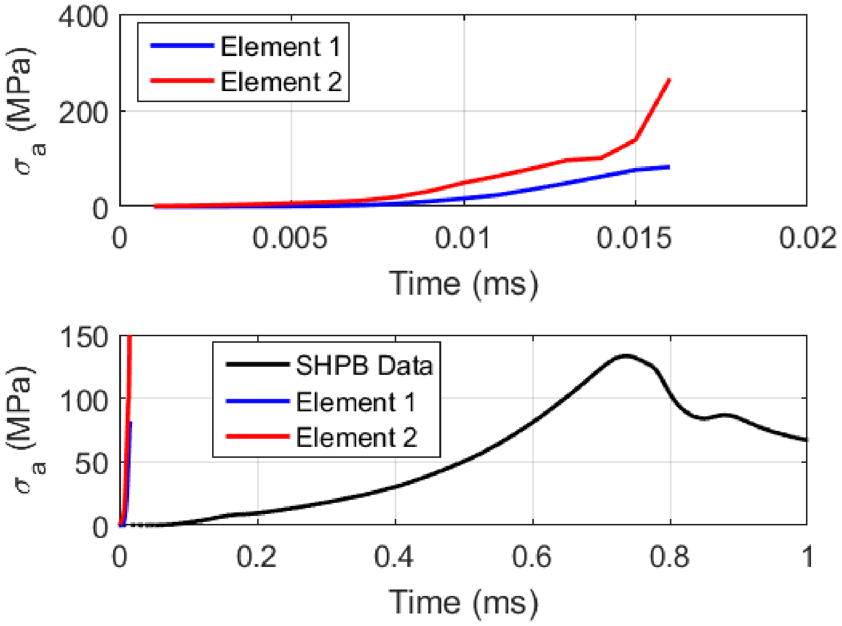


Figure 4.33: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Taylor-Voigt Boundary Conditions

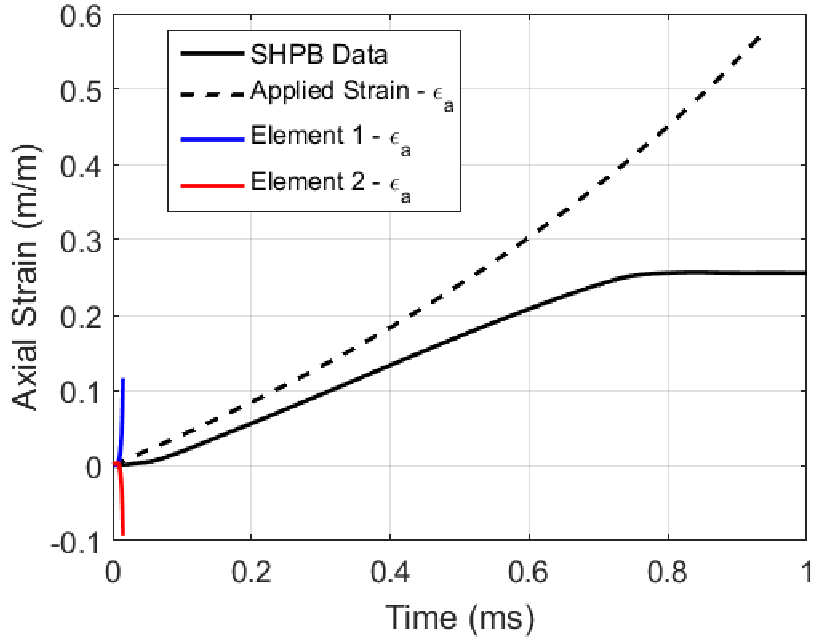


Figure 4.34: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Restricted Taylor Boundary Conditions

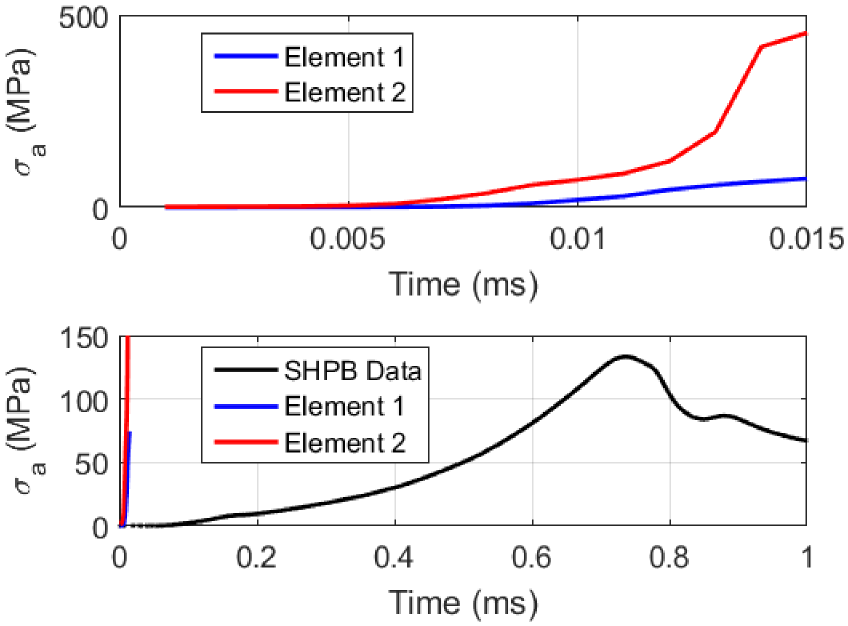


Figure 4.35: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Restricted Taylor Boundary Conditions

4.4.5 Other FEM-DEM SHPB Models

Two additional FEM-DEM SHPB simulations were run using the wall boundary conditions and original DEM RVE: one with a correctly initialized velocity and another with a more accurate applied displacement. The correctly initialized velocity mimics the ABAQUS Arena SHPB simulation in section 3.4.2. For the more accurate applied displacement simulation, the same displacement function (equation 4.56) was applied only over 0.05 to 0.75 ms to more closely mimic the SHPB Data (see figure 4.38).

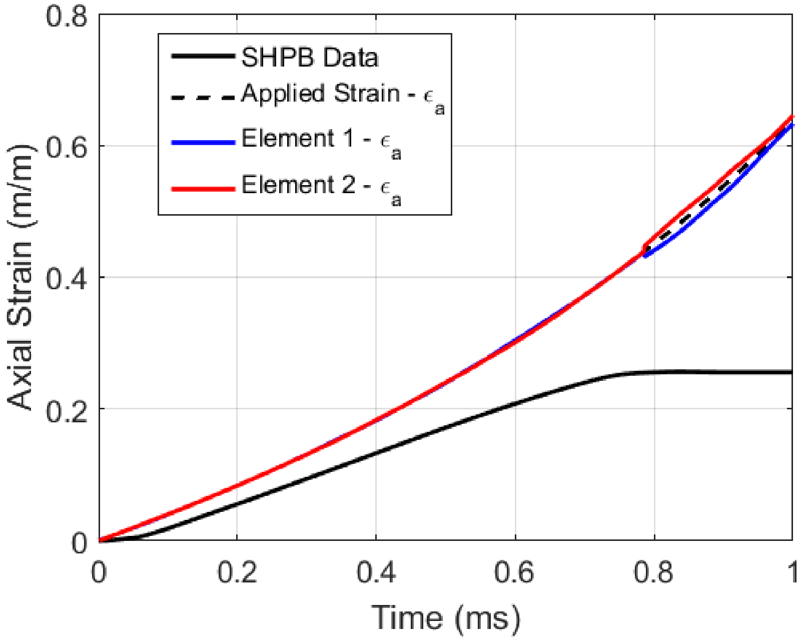


Figure 4.36: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Correctly Initialized Velocity

Stress and strain time histories from the initialized velocity simulation can be found on figures 4.36 and 4.37, and stress and strain time histories from the more accurate displacement on figures 4.38 and 4.39. Results from the initialized velocity simulation are nearly identical to the original hierarchical upscaling results, as they were for the two ABAQUS Arena SHPB simulations. With a more accurate displacement function, the stress results were lower than the original simulation likely due to the 0.05 ms lag between simulations. More interestingly, when the applied axial strain

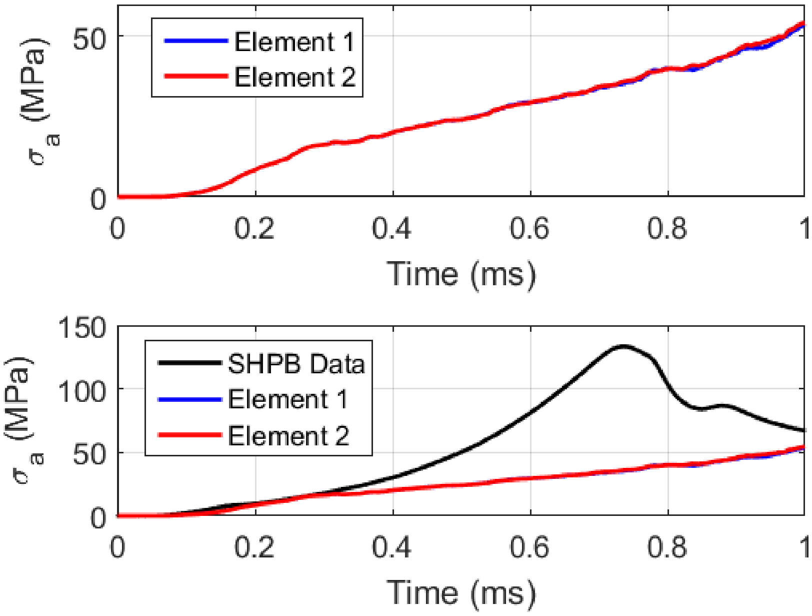


Figure 4.37: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Correctly Initialized Velocity

remains constant after 0.75 ms, the same “post experiment” rebound response seen in the data also occurs. This is likely the result of the redistribution of particles and microstress after the end of the applied displacement.

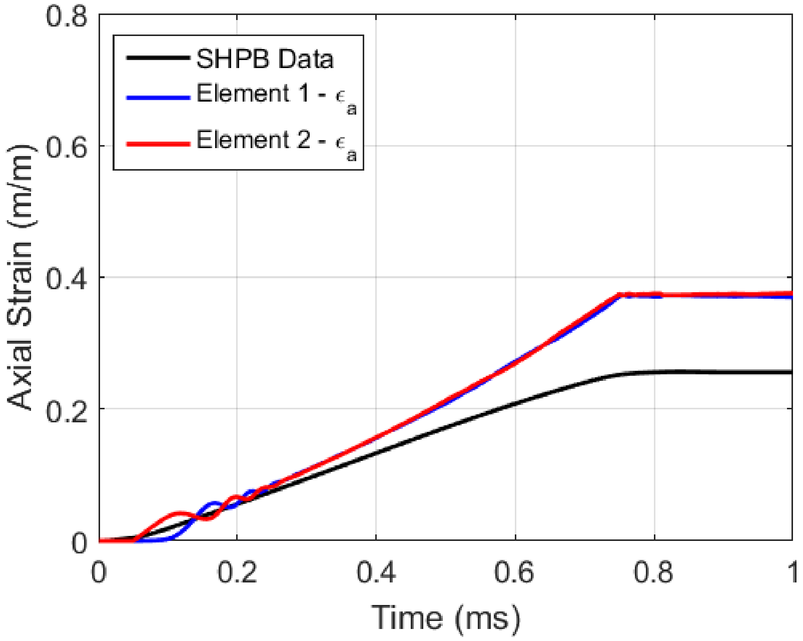


Figure 4.38: Strain Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Accurate Applied Displacement

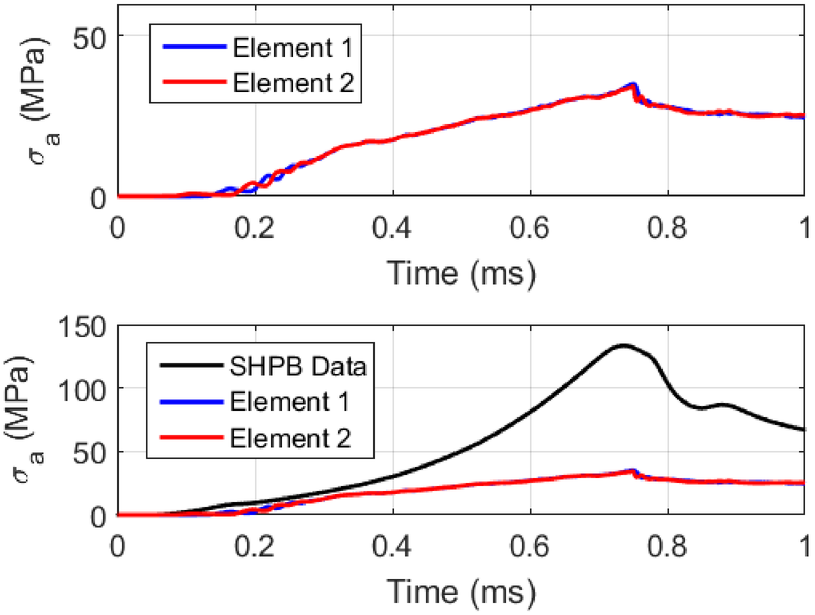


Figure 4.39: Stress Time History Comparison Between the SHPB Experiment and Explicit FEM-DEM Hierarchical Upscaling Simulation with Accurate Applied Displacement

Chapter 5

1D MPM-DEM Global-Local Analysis

In chapter 4, the development of a 1D FEM-DEM global-local analysis for modeling a split-Hopkinson pressure bar test on dry Colorado Mason was described. Considering that one of the ultimate goals of the Soil Blast MURI Project was to find methods to link a complicated coupled DEM-SPH-PD-CFD sand-grain-scale model with a macroscale MPM model (see figure 1.1), the FEM-DEM hierarchical multiscale model was extended to a novel MPM-DEM global-local analysis.

The general concept of the two models is the same. The constitutive model used to calculate the stress in the modeled material given an applied strain is replaced with a representative assembly of DEM particles. The strain is converted to boundary conditions applied to the DEM representative volume element, and after the assembly is allowed to deform given the applied boundary conditions, a homogenization technique is used to approximate the stress in the RVE. However, in the FEM-DEM model, the DEM RVEs are assumed to be located at the integration points of the FEM mesh. The Lagrangian FEM mesh deforms with the model, but in the material point method, the momentum equation is solved using an Eulerian mesh, and the material state (i.e. density, velocity, stress, etc.) is stored in “material points” that are allowed to move through the mesh given the deformation of the material. In each iteration, the material’s state is mapped to nodes in the Eulerian mesh, the momentum equation is solved, and the updated state is mapped back to the material points. As a result, rather than positioning the DEM RVEs at the integration points, the RVEs in the MPM-DEM hierarchical multiscale model are assumed to be located at the material points.

This chapter presents the basic derivation and one dimensional, uniaxial strain implementation of the material point method and the subsequent creation of a MPM-DEM global-local analysis model. Similar to FEM-DEM model described in chapter 4, the MPM-DEM model was also used to model a split Hopkinson pressure bar experiment conducted on dry Colorado Mason Sand.

5.1 Material Point Method Model Development

In the material point method, the Lagrangian description of an arbitrary body (Ω) is described by a set of material points (p), which store information about the body's position ($\boldsymbol{x}_p$), velocity ($\boldsymbol{v}_p$), density (ρ_p), mass (M_p), and Cauchy stress ($\boldsymbol{\sigma}_p$). A body can be described by any number of material points from 1 to n_p, or the total number of material points. These points are then overlaid on an Eulerian mesh, or "background grid" in MPM parlance, designed to spatially cover the initial position of Ω and all possible locations to which the body is expected to move or deform. During each timestep, the state information stored in the material points is mapped to the grid via shape functions. The balance of linear momentum is then solved using the grid, and updated state information is mapped back to the material points from the deformed grid. The grid is then reset to its original position, and the algorithm moves to the next timestep. As a result, the material points are able to incrementally move through the background grid with the deformation of the body. This allows for massive deformations without remeshing the background grid, as would likely be required in a FEM version of the same problem. The following derivation is a compilation of the original work on the material point method by Sulsky et al. (1994) and Sulsky et al. (1995) with numerical implementation interjections and notational changes from a primer on MPM by Nguyen (2014).

The variational (weak) form of the balance of linear momentum without external tractions or including gravitational terms (during the split Hopkinson pressure bar test the effect due to gravity is assumed to be negligible and the simulation will be strain controlled) is written

$$\int_{\mathcal{B}} \rho \boldsymbol{w} \cdot \boldsymbol{a} \mathrm{d}\Omega + \int_{\mathcal{B}} \rho \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}} : \boldsymbol{\sigma}^{\mathrm{s}} \mathrm{d}\Omega = 0 \tag{5.1}$$

where ρ is the mass density of the body (assumed constant over the body), $\boldsymbol{a}$ is the acceleration of the body, $\boldsymbol{w}$ is an arbitrary weighting function, and $\boldsymbol{\sigma}^{\mathrm{s}}$ is the specific Cauchy stress ($\boldsymbol{\sigma} = \rho\boldsymbol{\sigma}^{\mathrm{s}}$) all defined in the current configuration ($\mathcal{B}$) of the body (Ω). The mass (and volume), weighting function, acceleration, and stress are all functions of the position of the material points ($\boldsymbol{x}_p$). Furthermore, the mass is assumed to be concentrated at the material points such that

$$\rho(\boldsymbol{x}, t) = \sum_{p=1}^{n_p} M_p \delta(\boldsymbol{x} - \boldsymbol{x}_p) \tag{5.2}$$

where $\boldsymbol{x}$ is the position in body Ω and δ is the Dirac delta function with units of one over volume. Upon substituting equation 5.2 into equation 5.1 and reorganizing the operation of the Dirac delta function ($\int_{\mathcal{B}} f(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}_p)\mathrm{d}\Omega = f(\boldsymbol{x}_p)$), the weak form of the balance of linear momentum is then a summation over all the material points (n_p) (Sulsky et al., 1994).

$$\sum_{p=1}^{n_p} M_p \boldsymbol{w}(\boldsymbol{x}_p) \cdot \boldsymbol{a}(\boldsymbol{x}_p) + \sum_{p=1}^{n_p} M_p \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_p} : \boldsymbol{\sigma}^{\mathrm{s}}(\boldsymbol{x}_p) = 0 \tag{5.3}$$

The Eulerian mesh, or background grid, can be thought of like a typical finite element mesh complete with shape functions (N_a) that map the kinematic variables to the mesh's nodes (a). For example, the position of a location in the body ($\boldsymbol{x}$) is calculated by mapping the material point locations to the background grid by the following equation

$$\boldsymbol{x}(\boldsymbol{x}_p) = \sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{x}_a = \begin{bmatrix} N_I & N_{II} & \cdots & N_{n_n} \end{bmatrix} \cdot \begin{bmatrix} (x_1)_I & (x_2)_I & (x_3)_I \\ (x_1)_{II} & (x_2)_{II} & (x_3)_{II} \\ \vdots & \vdots & \vdots \\ (x_1)_{n_n} & (x_2)_{n_n} & (x_3)_{n_n} \end{bmatrix} = \boldsymbol{N} \cdot \boldsymbol{x}_m \tag{5.4}$$

where $\boldsymbol{x}_a$ is the position of node a, n_n is the total number of nodes in the entire background grid, $\boldsymbol{N}$ is the shape functions in vector form, $\boldsymbol{x}_m$ is the "matrix" of nodal position vectors, and assuming $\boldsymbol{x}$

is three dimensional such that x_1, x_2, and x_3 are the components of $\boldsymbol{x}$ in each of the three directions. Background grid nodes are numbered using Roman numerals $(a = I, II, \cdots, n_n)$. The background grid spatially covers the initial positions of the body, but also all of the positions to which the body is reasonably expected to move or deform. The discretization results in three equations for each background grid node.

The shape functions (N_a) are a function of the material point's position $(\boldsymbol{x}_p)$, and if the material point is too far away from the position of the node $(\boldsymbol{x}_a)$, its value will be zero. For example, one-dimensional, linear shape functions for elements that are all the same length (l_x)

$$N_a = \begin{cases} 1 - \frac{|x_p - x_a|}{l_x} & \text{if} \quad |x_p - x_a| < l_x \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

will be equal to zero unless the material point's position is within one element length of the position of node a. In other words, N_a will be non-zero if the material point is between nodes $a-1$ and $a+1$ in this one dimensional example, and multiple material points can be mapped to the same node. The displacement $(\boldsymbol{u})$, velocity $(\boldsymbol{v})$, and acceleration $(\boldsymbol{a})$ of the body are similarly discretized (Sulsky et al., 1994).

$$\boldsymbol{u}(\boldsymbol{x}_p) = \sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{u}_a \tag{5.6}$$

$$\boldsymbol{v}(\boldsymbol{x}_p) = \sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{v}_a \tag{5.7}$$

$$\boldsymbol{a}(\boldsymbol{x}_p) = \sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{a}_a \tag{5.8}$$

Similarly and following the Bubnov-Galerkin method, the weighting function and spatial derivative of the weighting function are discretized as follows.

$$\boldsymbol{w}(\boldsymbol{x}_p) = \sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{w}_a \tag{5.9}$$

$$\left.\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_p} = \sum_{a=1}^{n_n} \left.\frac{\partial N_a}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_p} \boldsymbol{w}_a \tag{5.10}$$

It should be noted that the simple shape functions described in equation 5.5 can cause errors when material points cross into new elements in the background grid. Material points crossing into new elements often cause a different number of particles to be in adjacent elements, which due to the discontinuous nature of the shape function gradient, causes force imbalances to develop. Modern shape functions, such as those found in more general MPM variants like the Generalized Interpolation Material Point (GIMP) method (Bardenhagen and Kober, 2004) or Convected Particle Domain Interpolation (CPDI) method (Sadeghirad et al., 2011), use the average of N_a over the particle domain, not the value of N_a at the particle center, which helps alleviate this problem.

The discretized kinematic variables are then substituted into the material point dependent balance of linear momentum (equation 5.3).

$$\sum_{p=1}^{n_p} M_p \left[\sum_{a=1}^{n_n} N_a(\boldsymbol{x}_p)\boldsymbol{w}_a\right] \cdot \left[\sum_{b=1}^{n_n} N_b(\boldsymbol{x}_p)\boldsymbol{a}_b\right] + \sum_{p=1}^{n_p} M_p \left[\sum_{a=1}^{n_n} \left.\frac{\partial N_a}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_p} \boldsymbol{w}_a\right] : \boldsymbol{\sigma}^{\mathrm{s}}(\boldsymbol{x}_p) = 0 \tag{5.11}$$

Once again assuming three dimensions, the balance of linear momentum can be reorganized and rewritten in matrix form

$$\sum_{p=1}^{n_p} M_p \left[\boldsymbol{w}_m^{\mathrm{T}} \cdot \left(\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}_p) \cdot \boldsymbol{N}(\boldsymbol{x}_p) \cdot \boldsymbol{a}_m + \left(\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_p}\right)^{\mathrm{T}} : \boldsymbol{\sigma}^{\mathrm{s}}(\boldsymbol{x}_p)\right)\right] = \boldsymbol{0} \tag{5.12}$$

where $\boldsymbol{w}_m$ and $\boldsymbol{a}_m$ are $n_n \times 3$ "matrix" nodal weighting function and acceleration that are similar to $\boldsymbol{x}_m$ in equation 5.4 (total number of background grid nodes by three spatial dimensions), $\boldsymbol{N}$ is a $1 \times n_n$ shape function vector (total number of background grid nodes), and $\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_p}$ is the $3 \times n_n$ shape function gradient (three spatial dimensions by total number of background grid nodes). As there are three equations for each background grid node (a), the matrix form is often represented using a variant of indicial notation where the summation over the total number of nodes is assumed. Equation 5.12 can then be written

$$\boldsymbol{w}_a^{\mathrm{T}} \cdot \left[\underbrace{\left(\sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_p) N_b(\boldsymbol{x}_p) \right)}_{M_{ab}} \boldsymbol{a}_b + \underbrace{\left(\sum_{p=1}^{n_p} M_p \frac{\partial N_a}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_p} \cdot \boldsymbol{\sigma}^{\mathrm{s}}(\boldsymbol{x}_p) \right)}_{\boldsymbol{F}_a^{\mathrm{int}}} \right] = \mathbf{0} \tag{5.13}$$

where M_{ab} is the consistent mass matrix and $\boldsymbol{F}_a^{\mathrm{int}}$ is the internal force vector. When summed over the total number of background grid nodes, the consistent mass matrix is a $n_n \times n_n$ matrix, the matrix of nodal acceleration vectors is a $n_n \times 3$ matrix, and the matrix of internal force vectors is a $n_n \times 3$ matrix. So, after assuming the arbitrary weighting function (except at the nodes where prescribed displacement or velocity boundary conditions are enforced), equation 5.13 further reduces to the following.

$$M_{ab}\boldsymbol{a}_b + \boldsymbol{F}_a^{\mathrm{int}} = \mathbf{0} \tag{5.14}$$

Equation 5.14 is the balance of linear momentum discretized over the background grid nodes using kinematic information from the material points as all three variables (M_{ab}, $\boldsymbol{a}_b$, and $\boldsymbol{F}_a^{\mathrm{int}}$) are functions of the material point positions ($\boldsymbol{x}_p$). It can be further simplified through lumping the mass matrix by taking sum of each row

$$M_{ab} \implies M_a = \sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_p) \tag{5.15}$$

such that it becomes fully invertible.

$$M_a\boldsymbol{a}_a + \boldsymbol{F}_a^{\mathrm{int}} = \mathbf{0} \tag{5.16}$$

The internal force vector ($\boldsymbol{F}_a^{\mathrm{int}}$), after considering the definition of specific stress ($\boldsymbol{\sigma}^{\mathrm{s}} = \sigma/\rho$), is

$$\boldsymbol{F}_a^{\mathrm{int}} = \sum_{p=1}^{n_p} M_p \frac{\partial N_a}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_p} \cdot \boldsymbol{\sigma}^{\mathrm{s}}(\boldsymbol{x}_p) = \sum_{p=1}^{n_p} \frac{M_p}{\rho} \frac{\partial N_a}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_p} \cdot \boldsymbol{\sigma}(\boldsymbol{x}_p) = \sum_{p=1}^{n_p} V_p \frac{\partial N_a}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_p} \cdot \boldsymbol{\sigma}(\boldsymbol{x}_p) \tag{5.17}$$

where V_p is the volume of a given material point (Sulsky et al., 1994) (Nguyen, 2014).

The material point method often uses an explicit time integration scheme using the velocity, or momentum, formulation. Using Euler forward, the acceleration of the body is integrated in time by

$$\boldsymbol{v}_{a(n+1)} = \boldsymbol{v}_{a(n)} + \boldsymbol{a}_{a(n)} \Delta t \tag{5.18}$$

where $\boldsymbol{v}_{a(n+1)}$ is the nodal velocity at the current time (t_{n+1}), and $\boldsymbol{v}_{a(n)}$ is the nodal velocity at the previous time (t_n), $\boldsymbol{a}_{a(n)}$ is the nodal acceleration at the previous time, and $\Delta t = t_{n+1} - t_n$ is the timestep. When substituted into equation 5.16, the balance equation becomes

$$M_{a(n)} \left(\boldsymbol{v}_{a(n+1)} - \boldsymbol{v}_{a(n)} \right) = \boldsymbol{F}_{a(n)} \Delta t \tag{5.19}$$

where $\boldsymbol{F}_{a(n)}$ is the total force vector at the previous time (i.e. $\boldsymbol{F}_a = -\boldsymbol{F}_a^{\text{int}}$), $(M\boldsymbol{v})_{a(n+1)}$ is the nodal momentum at the current time, and $(M\boldsymbol{v})_{a(n)}$ is the nodal momentum at the previous time. The masses in the nodal momentum at the current and previous time are both $M_{a(n)}$; only the nodal velocity changes over the timestep (Nguyen, 2014).

The time integration scheme progresses by starting with the position and velocity of all the material points at the previous time. Initial conditions will need to be applied to each material point at $t = 0$. Recall that M_a and $\boldsymbol{F}_a$ are functions of the positions of the material points (equations 5.15 and 5.17) such that calculating the nodal mass, momentum, and forcing vector at the previous time ($M_{a(n)}$, $(M\boldsymbol{v})_{a(n)}$, and $\boldsymbol{F}_{a(n)}$, respectively) requires first mapping the material point's kinematic variables to the nodes via

$$M_{a(n)} = \sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_{p(n)}) \tag{5.20}$$

$$(M\boldsymbol{v})_{a(n)} = \sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_{p(n)})\boldsymbol{v}_{p(n)} \tag{5.21}$$

$$\boldsymbol{F}_{a(n)} = -\boldsymbol{F}^{\text{int}}_{a(n)} = -\sum_{p=1}^{n_p} V_{p(n)} \frac{\partial N_a}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_{p(n)}} \cdot \boldsymbol{\sigma}(\boldsymbol{x}_{p(n)}) \tag{5.22}$$

where $\boldsymbol{v}_p$ is the velocity of a given material point (p). Note that the material point mass (M_p) is considered constant in order to satisfy the conservation of mass; however, the density and volume of a point are allowed to evolve with the deformation of the body (Nguyen, 2014).

From the nodal momentum and forcing vector at the previous time (t_n), the nodal momentum at the current time (t_{n+1}) is then calculated from equation 5.19 (i.e. $(M\boldsymbol{v})_{a(n+1)} = (M\boldsymbol{v})_{a(n)} + \boldsymbol{F}_{a(n)}\Delta t$). The material point positions and velocities are then updated to the current time by mapping the nodal acceleration and updated momentum back to the material points using the same shape functions that mapped the material points to the nodes.

$$\boldsymbol{v}_{p(n+1)} = \boldsymbol{v}_{p(n)} + \Delta t \left[\sum_{a=1}^{n_n} N_a(\boldsymbol{x}_{p(n)}) \frac{\boldsymbol{F}_{a(n)}}{M_{a(n)}}\right] \tag{5.23}$$

$$\boldsymbol{x}_{p(n+1)} = \boldsymbol{x}_{p(n)} + \Delta t \left[\sum_{a=1}^{n_n} N_a(\boldsymbol{x}_{p(n)}) \frac{(M\boldsymbol{v})_{a(n+1)}}{M_{a(n)}}\right] \tag{5.24}$$

In equation 5.23, $\frac{\boldsymbol{F}_{a(n)}}{M_{a(n)}}$ is the nodal acceleration vector calculated from the balance of linear momentum at the previous time (equation 5.16), and $\frac{(M\boldsymbol{v})_{a(n+1)}}{M_{a(n)}}$ in equation 5.24 is the updated nodal velocity vector calculated from the definition of momentum. The summations over the total number of background grid nodes (a) have been added for clarity. Note that the mass at background grid node a at the previous and current time ($M_{a(n)}$ and $M_{a(n+1)}$, respectively) are equal to satisfy the balance of mass in the Eulerian mesh. However, with the update to the material point positions (equation 5.24) and reset of the background grid at the end of the timestep, a given material point might move to another element in the grid meaning that the nodal mass at the beginning of the following timestep might not be equal to the current mass (Sulsky et al., 1994) (Nguyen, 2014).

With the updated material point positions and velocities, the stress in the body is calculated. Updating the stress in the current timestep is called the "update stress last," or USL, implementation of the material point method. In order to calculate the updated Cauchy stress at the material points, the particle velocity gradient at the current time ($\boldsymbol{L}_{p(n+1)}$) is calculated by mapping the updated nodal velocities to the material points using the gradient of the shape functions via

$$\boldsymbol{L}_{p(n+1)} = \frac{\partial \boldsymbol{v}_{p(n+1)}}{\partial \boldsymbol{x}} = \sum_{a=1}^{n_n} \frac{\partial N_a(\boldsymbol{x}_{p(n)})}{\partial \boldsymbol{x}} \boldsymbol{v}_{a(n+1)} \tag{5.25}$$

where $\boldsymbol{v}_{a(n+1)}$ is the updated nodal velocity, which is equal to

$$\boldsymbol{v}_{a(n+1)} = \frac{(M\boldsymbol{v})_{a(n+1)}}{M_{a(n)}} \tag{5.26}$$

as described in equation 5.24 (Nguyen, 2014).

However, calculation of the nodal velocity at the current time can be problematic if the nodal mass, calculated using equation 5.20, is very small as the nodal velocity will tend to infinity as the nodal mass goes to zero. This condition, generally, occurs in background grid elements that contain only one material point that is located near to one of the grid's nodes. In one dimension, for example, if a linear element contains only one material point that is located close to node 1, the mass at the other node, node 2, will be calculated to be near zero. Although there are a number of proposed solutions in the material point method literature, one of the solutions, as proposed by Sulsky et al. (1994), is to map the current particle velocities to the current nodal velocities as was done for the previous particle and nodal velocities (equation 5.21).

$$(M\boldsymbol{v})_{a(n+1)} = \sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_{p(n)}) \boldsymbol{v}_{p(n+1)} \tag{5.27}$$

The equation for the updated nodal velocities then becomes the following.

$$\boldsymbol{v}_{a(n+1)} = \frac{(M\boldsymbol{v})_{a(n+1)}}{M_{a(n)}} = \frac{\sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_{p(n)}) \boldsymbol{v}_{p(n+1)}}{\sum_{p=1}^{n_p} M_p N_a(\boldsymbol{x}_{p(n)})} \tag{5.28}$$

The inclusion of the shape functions in both the numerator and denominator cancel the effect of mapping nearly all the mass of a material point to a single node.

With the calculation of the nodal velocity at the current time, the particle velocity gradient can then be calculated using equation 5.25. From the particle velocity gradient, the particle deformation gradient is calculated by first recognizing that the time derivative of the deformation gradient ($\dot{\boldsymbol{F}}$) is equal to the product of the velocity gradient and deformation gradient (i.e. $\dot{\boldsymbol{F}} = \boldsymbol{L}\boldsymbol{F}$). Again using the explicit Euler forward to integrate in time, the particle deformation gradient at the current time ($\boldsymbol{F}_{p(n+1)}$) is equal to

$$\frac{\boldsymbol{F}_{p(n+1)} - \boldsymbol{F}_{p(n)}}{\Delta t} = \boldsymbol{L}_{p(n)}\boldsymbol{F}_{p(n)} \implies \boldsymbol{F}_{p(n+1)} = \left(\boldsymbol{I} + \boldsymbol{L}_{p(n)}\Delta t\right)\boldsymbol{F}_{p(n)} \tag{5.29}$$

where $\boldsymbol{I}$ is the identity matrix. The particle deformation gradient, or change in particle deformation gradient ($\Delta\boldsymbol{F}_p = \boldsymbol{F}_{p(n+1)} - \boldsymbol{F}_{p(n)}$), is then sent to the material constitutive model in order to calculate the change in Cauchy stress in every material point ($\Delta\boldsymbol{\sigma}_p$). The particle Cauchy stress at the current time is then updated via the following.

$$\boldsymbol{\sigma}_{p(n+1)} = \boldsymbol{\sigma}_{p(n)} + \Delta\boldsymbol{\sigma}_p \tag{5.30}$$

Finally, the volume of each material point at the current time ($V_{p(n+1)}$) is updated by recognizing the volumetric deformation of the body is equal to the determinant of each particle's deformation gradient

$$V_{p(n+1)} = \det(\boldsymbol{F}_{p(n+1)})V_{p(0)} \tag{5.31}$$

where $V_{p(0)}$ is the initial volume of the given material point (Sulsky et al., 1994) (Nguyen, 2014).

5.2 Split Hopkinson Pressure Bar Simulation Development

Similar to the FEM-DEM global-local analysis outlined in chapter 4, a one-dimensional, uniaxial strain implementation of the material point method was developed to model a split Hopkinson pressure bar experiment conducted on dry Colorado Mason Sand using a novel MPM-DEM hierarchical multiscale model. Whereas chapter 4 focused on studying the effects of periodicity and particle fracture in the DEM assemblies, the impact of DEM representative volume element (RVE) size, and implicit versus explicit FEM, the focus of the MPM-DEM version of the model was the effect of the material point method on the global-local analysis. As a result, the DEM portion of the model will not again be discussed, but one is referred to chapter 4 for further information.

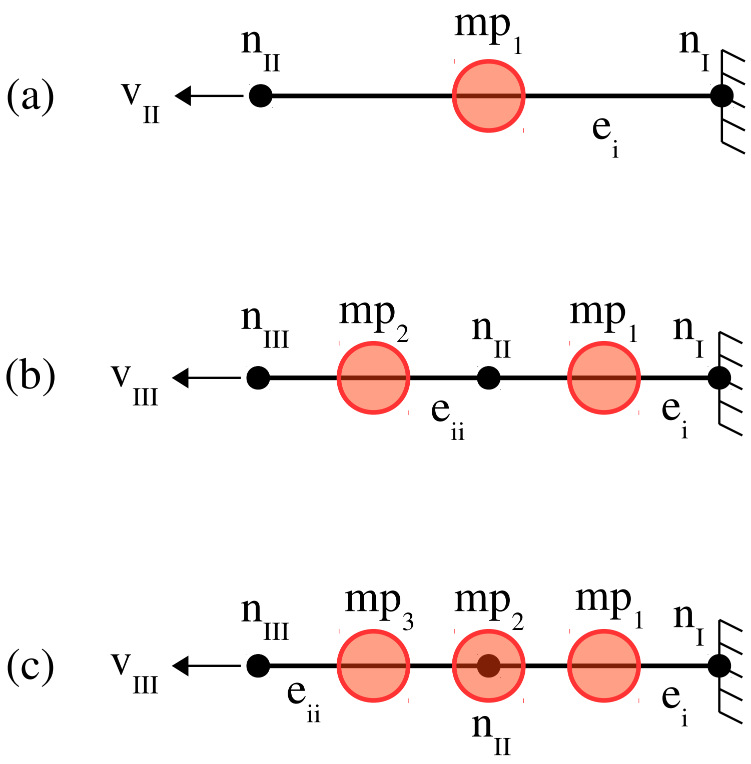


Figure 5.1: MPM-DEM Model Geometries

Three versions of the MPM-DEM model were developed varying the number of background grid element and material points: a one element one material point model, a two element two material point model, and a two element three material point model. Visualizations of these configurations can be found on figure 5.1. These variants were selected for their simplicity, but also to highlight the ability of the material points to move through different elements of the grid. In the figure, n and e represent the node and element numbers, and mp represents the material point numbers. Uppercase and lowercase Roman numerals are used to denote nodes and elements, respectively, and Arabic numerals are used to denote material points. Therefore, n_{II} is node two, e_i is element one, and mp_1 is material point one. Both are numbered from right to left. Similarly, nodal variables are also denoted using uppercase Roman numerals and Arabic numerals are used to denote material point variables. For example, x_I is the position of background grid node one and x_1 is the position of material point one.

The applied boundary conditions were the same for each version of the model and were designed to mimic the applied displacement used in the FEM-DEM global-local analysis model (refer to figure 4.6). The MPM implementation uses the momentum formulation for the balance equation (equation 5.19), so rather than applying a displacement, the velocity and acceleration of the right most nodes were set to zero and the velocity of the left most nodes (n_{II} or n_{III} if one or two background grid elements are included, respectively) were prescribed. As the SHPB experiment is designed such that the strain rate of the soil specimen is constant, the applied velocity in a small strain context (v_{II} or v_{III}) was set to

$$v_{II} = L_0 \dot{\epsilon}_0 \tag{5.32}$$

where L_0 is the initial length of the soil specimen and $\dot{\epsilon}_0$ is the constant strain rate. To fully mimic the finite displacement from chapter 4, v_{II} should have been set to $\dot{d}(t) = (L_0 \dot{\epsilon}_0)\mathrm{e}^{\dot{\epsilon}_0 t}$, which would have resulted in a larger applied velocity for all time greater than zero. The difference between the two, however, is small.

The background grid does not change throughout the simulation; all of the material's state information is stored in the material points. As a result, the initial conditions are applied to the material points. The initial positions of the material points can be seen on figure 5.1. For each material point that is initially inside an element, it is located at the center of that element, and in the case of the two element three material point model, the middle material point is initially located directly over the middle node. The initial stress of every material point was set to zero, and the initial velocity of every material point (v_1, v_2, and v_3) was set to $(v_{II})_0$. The initial mass and volume of the material points were assumed from the initial density of the DEM RVE, which like in the FEM-DEM models, were initially the same assembly for all material points, but were allowed to deform independently throughout the simulations. For the three implementations, typical one-dimensional, linear FEM shape functions (equation 5.5) were used for the shape functions.

In order to better understand the overall MPM-DEM algorithm, the following is an overview of the one element, one material point version of the model.

Algorithm for the One Element, One Material Point MPM-DEM Model:

(1) Initializations (t_0):

(a) Initial position of the material point located at the center of soil specimen: $x_1 = \frac{1}{2}L_0$

(b) Initial mass of material point (A is the cross-sectional area of the specimen): $M_1 = \rho A L_0$

(c) Initial volume of the material point: $V_1 = AL_0$

(d) Initial velocity of the material point: $v_1 = L_0\dot{\epsilon}_0$ (or $(L_0\dot{\epsilon}_0)\mathrm{e}^{\dot{\epsilon}_0 t}$)

(e) Initial stress of the material point: $\sigma_1 = 0$

(f) Initial momentum of the material point: $(Mv)_1 = M_1 v_1$

(g) Initial deformation gradient of the material point: $F_1 = 1$

(2) Map material points to background grid nodes at the previous time (t_n):

(a) Calculate the shape functions and the gradient of the shape functions.

- $N_I = \frac{|x_1 - x_I|}{L_0}$
- $N_{II} = \frac{|x_1 - x_{II}|}{L_0}$

- $\frac{\partial N_I}{\partial x} = \frac{-1}{L_0}$
- $\frac{\partial N_{II}}{\partial x} = \frac{1}{L_0}$

(b) Note: If the model has more than one material point and more than one background grid element, use the positions of the material points to locate which material points are in which element.

(c) Map the material point mass to the nodes.

- $M_I = N_I M_1$
- $M_{II} = N_{II} M_1$

(d) Map the material point momentum to the nodes. As there are only two nodes, both have boundary conditions.

- $(Mv)_I = N_I (Mv)_1 = 0$
- $(Mv)_{II} = N_{II} (Mv)_1 = M_{II} \dot{\epsilon}_0 L_0$

(e) Calculate the nodal force vector. To prevent the material point from moving outside of the background grid, the nodal acceleration (and therefore nodal force) is set to zero.

- $f_I = -V_1 \sigma_1 \frac{\partial N_I}{\partial x} = 0$
- $f_{II} = -V_1 \sigma_1 \frac{\partial N_{II}}{\partial x}$

(3) Use the balance of linear momentum to update the nodal momentum at the current time (t_{n+1}):

- $(Mv)_{I(n+1)} = (Mv)_{I(n)} + f_{I(n)} \Delta t$
- $(Mv)_{II(n+1)} = (Mv)_{II(n)} + f_{II(n)} \Delta t$

(4) Map background grid nodes to the material point at the current time (t_{n+1}):

(a) Update the material point position: $x_{1(n+1)} = x_{1(n)} + \Delta t \left(N_I \frac{(Mv)_{I(n+1)}}{M_I} + N_{II} \frac{(Mv)_{II(n+1)}}{M_{II}} \right)$

(b) Update the material point velocity: $v_{1(n+1)} = v_{1(n)} + \Delta t \left(N_I \frac{f_{I(n)}}{M_I} + N_{II} \frac{f_{II(n)}}{M_{II}} \right)$

(c) Update the material point momentum: $(Mv)_{1(n+1)} = M_1 v_{1(n+1)}$

(5) Update nodal velocity using method proposed by Sulsky et al. (1994). As there are only two nodes, both have boundary conditions.

- $v_I = \frac{N_I M_1 v_1}{M_I} = 0$
- $v_{II} = \frac{N_{II} M_1 v_1}{M_{II}} = \dot{\epsilon}_0 L_0$

(6) Calculate the material point velocity gradient: $L_1 = \frac{\partial N_I}{\partial x} v_I + \frac{\partial N_{II}}{\partial x} v_{II}$

(7) Calculate the material point deformation gradient: $F_{1(n+1)} = (1 + L_{1(n)} \Delta t) F_{1(n)}$

(8) Using the deformation gradient, apply boundary conditions to the DEM assembly located at the material point (see section 4.3.2). Allow the DEM assembly to deform and calculate the material point stress (σ_1) at the current timestep.

(9) Move to the next timestep: $t_{n+1} = t_n + \Delta t$

5.3 Split Hopkinson Pressure Bar MPM-DEM Modeling Results

To test the MPM-DEM global-local analysis model, the SHPB simulation described in chapter 4 was recreated using the three geometries shown in figure 5.1. Model (a) was one material point positioned in the middle of one background grid element (1mp-1el). Strain and stress time histories for the 1mp-1el model can be found on figure 5.2 and 5.3. As can be seen, the results from the 1mp-1el model compare favorably to the results of the FEM-DEM SHPB model. The maximum axial stress is lower in the 1mp-1el simulation; however, this was expected due to the lower axial strain applied by the velocity boundary condition (equation 5.32).

With the confidence that the MPM-DEM implementation was able to model the SHPB experiment with similar accuracy as the FEM-DEM model, the second MPM-DEM model (b) was developed to test the scalability of the implementation. Model (b) had two material points

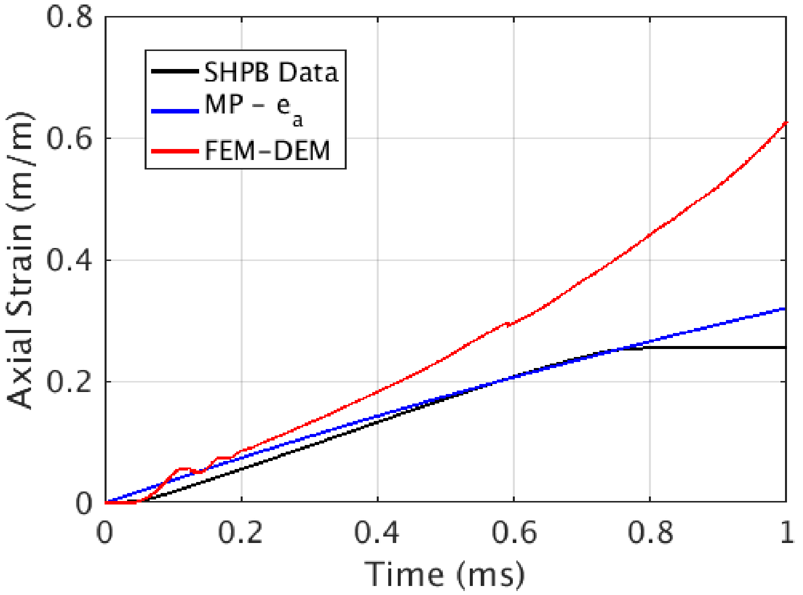


Figure 5.2: 1 Material Point, 1 Background Grid Element: SHPB Strain Time History

positioned at the center of two background grid elements (2mp-2el). Axial strain and stress results from the 2mp-2el model can be found on figures 5.4 and 5.5. The introduction of a second material point reintroduced axial strain and stress waves as the applied velocity was transferred from material point 2 to 1. Like in the FEM-DEM models, these waves quickly dissipated due to the friction and interparticle damping in the DEM. As with the 1mp-1el simulation, the results compared favorably to the FEM-DEM model after adjusting for the differences in applied strain.

Finally, a three material point model with two background grid elements was developed to test the model's ability to handle material point's behavior near a background grid node. Two of the material points were positioned in the center of each of the two background grid elements, and the third directly in the center of the simulation geometry at the position of the middle background grid node (3mp-2el). Axial strain and stress results from the 3mp-2el model can be found on figures 5.6 and 5.7. As can be seen, the results are different between the 3mp-2el simulation and the 1mp-1el and 2mp-2el simulations. The order of magnitude and general trends of the axial strain and stress results are the same as previous simulations, but the values are different. Considering the strain results, the middle material point (mp_2) displaces into the first background grid element, which

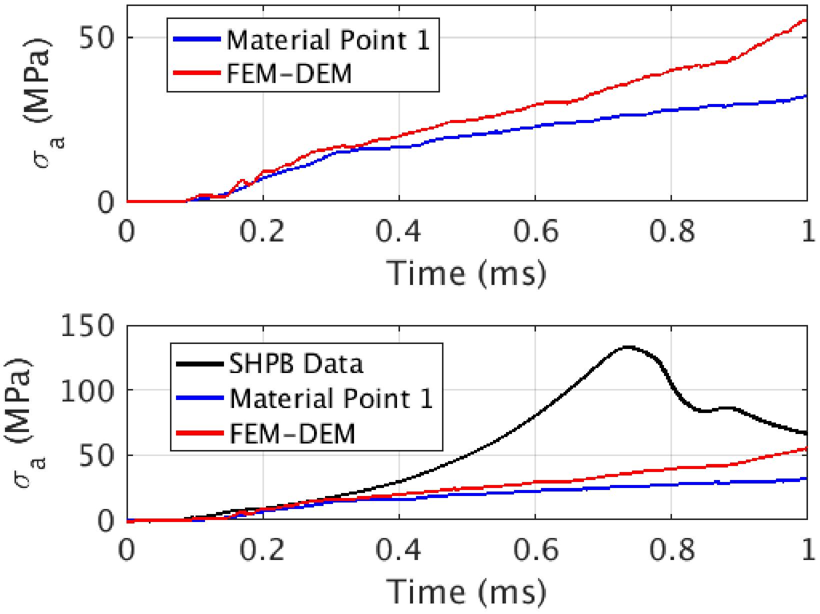


Figure 5.3: 1 Material Point, 1 Background Grid Element: SHPB Stress Time History

then negatively influences the following results. These types of unphysical results stemming from material points near background grid nodes and crossing into new background grid elements are known issues with linear interpolation functions in MPM. Clearly, this MPM-DEM implementation needs to be updated with more modern interpolation functions, such as those found in more general MPM variants like the Generalized Interpolation Material Point (GIMP) method (Bardenhagen and Kober, 2004) or Convected Particle Domain Interpolation (CPDI) method (Sadeghirad et al., 2011), that will likely remove this spurious behavior.

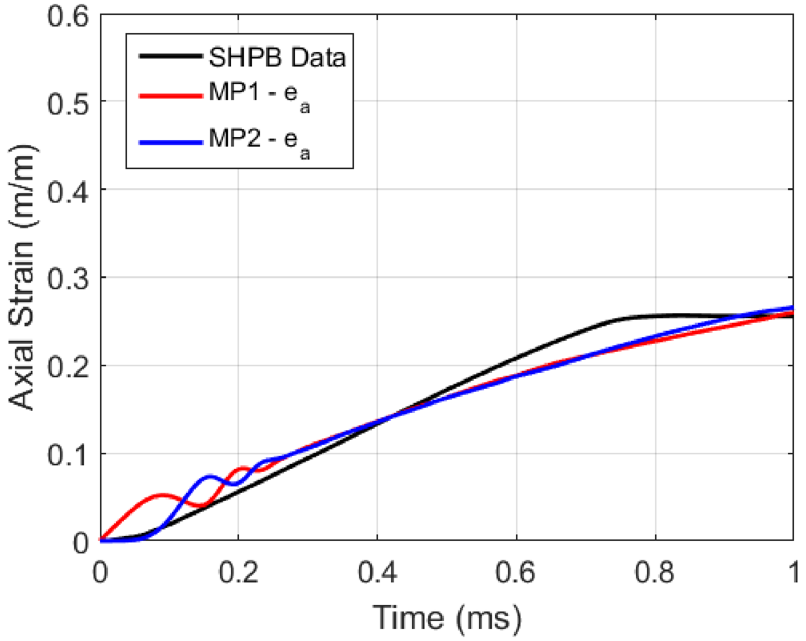


Figure 5.4: 2 Material Point, 2 Background Grid Element: SHPB Strain Time History

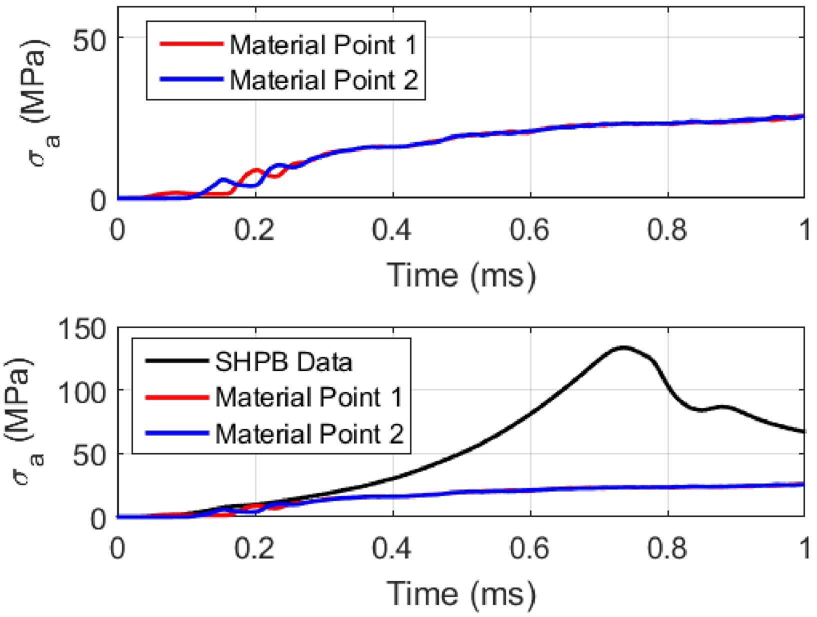


Figure 5.5: 2 Material Point, 2 Background Grid Element: SHPB Stress Time History

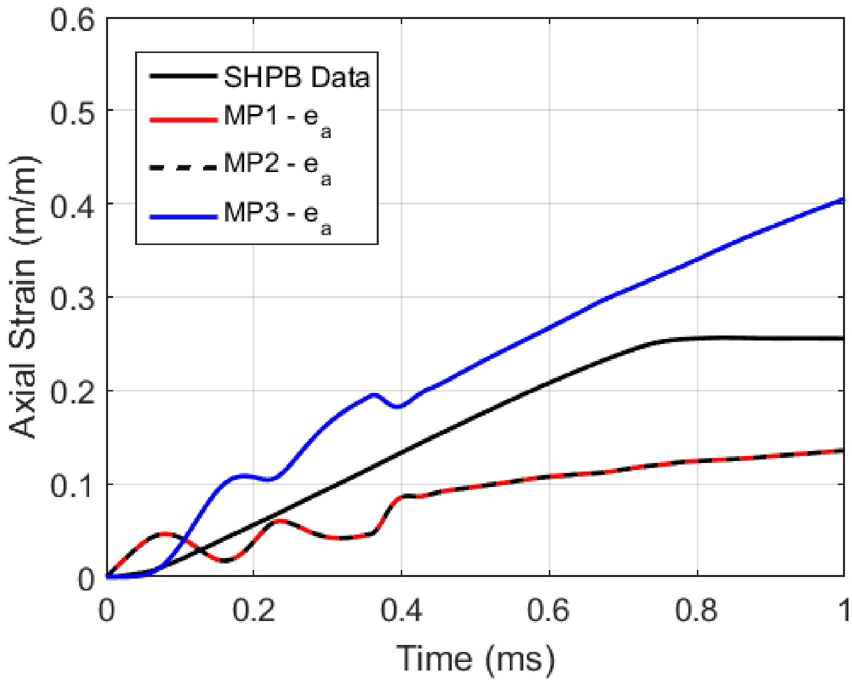


Figure 5.6: 3 Material Point, 3 Background Grid Element: SHPB Strain Time History

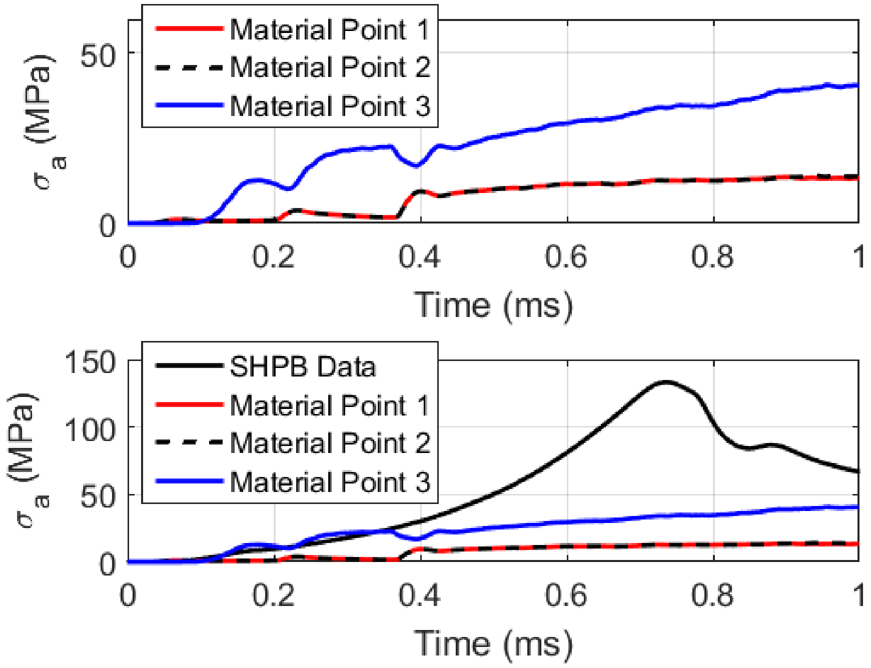


Figure 5.7: 3 Material Point, 3 Background Grid Element: SHPB Stress Time History

Chapter 6

2D Axisymmetric FEM-DEM Global-Local Analysis

In chapter 4, a 1D (uniaxial strain) FEM-DEM global-local analysis model was developed to simulate a split Hopkinson pressure bar experiment (SHPB) conducted on dry Colorado Mason Sand. Many other experiments were also conducted on dry Colorado Mason Sand (as well as Boulder Clay and mixtures of the two soils) for the Soil Blast MURI Project including drained, traditional and small-scale, triaxial testing. These experiments were used to help calibrate the Arena material model (see 3.4.2). As with the 1D FEM-DEM model and the SHPB experiments, the 1D FEM-DEM model implementation was extended to a 2D axisymmetric version to model the triaxial tests. Sections 6.1 and 6.2 will discuss the development of the model, sections 6.3 through 6.5 will present modeling results, and limitations of the model will be discussed in section 6.6.

6.1 Finite Element Model Development

Unlike SHPB experiments, triaxial testing is generally considered quasi-static and small strain (but could extend to finite strain in the post-peak response). If one also ignores body forces, assuming the effect of the acceleration due to gravity is negligible in the small, drained sand specimens, the localized strong form of the balance of linear momentum is expressed as

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0 \tag{6.1}$$

where σ_{ij} is the Cauchy stress tensor in a body ($\mathcal{B}$) and x_j is a position vector to a location in the body. Following the method of weighted residuals (see 4.1 for more details), the weak form of the balance of linear momentum is

$$\int_{\mathcal{B}} \frac{\partial w_i}{\partial x_j} \sigma_{ij} \mathrm{d}v - \int_{\Gamma^t} w_i t_i \mathrm{d}a = 0 \tag{6.2}$$

where w_i is an arbitrary weighting function and Γ^t is the boundary of the body over which the external traction vector (t_i) is applied.

In the axisymmetric finite element method, a cylindrical coordinate system is used to account for the influence of the radial dimension on the FEM solution. Positions in the body are described by their radial (r), axial (z), and circumferential (θ) locations. Considering torsionless axisymmetry (i.e. the behavior of the material is independent of the circumferential dimension), the displacement ($\boldsymbol{u}$) of a point in $\mathcal{B}$ is described using the following vector

$$\boldsymbol{u} = \begin{bmatrix} u_r(r,z) \\ u_z(r,z) \end{bmatrix} \tag{6.3}$$

where u_r is the displacement in the radial direction and u_z is the displacement in the axial direction. The weighting function ($\boldsymbol{w}$) then becomes

$$\boldsymbol{w} = \begin{bmatrix} w_r(r,z) \\ w_z(r,z) \end{bmatrix} \tag{6.4}$$

and the Cauchy stress ($\boldsymbol{\sigma}$) and strain ($\boldsymbol{\epsilon}$) are described as the following in Voigt notation.

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{rr} \\ \sigma_{zz} \\ \sigma_{rz} \\ \sigma_{\theta\theta} \end{bmatrix} \quad ; \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{rr} \\ \epsilon_{zz} \\ 2\epsilon_{rz} \\ \epsilon_{\theta\theta} \end{bmatrix} \tag{6.5}$$

The stress and strain in the rr direction is the normal radial stress and strain, zz directions is the normal axial stress and strain, rz direction is the shear stress and strain, and $\theta\theta$ direction is called the hoop stress and strain. Although the body isn't allowed to displace in the circumferential dimension ($u_\theta = 0$), the stress and strain in the circumferential dimension are still calculated. The strain-displacement relationships follow as

$$\epsilon_{rr} = \frac{\partial u_r}{\partial r} \tag{6.6}$$

$$\epsilon_{zz} = \frac{\partial u_z}{\partial z} \tag{6.7}$$

$$\epsilon_{rz} = \frac{1}{2}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) \tag{6.8}$$

$$\epsilon_{\theta\theta} = \frac{u_r}{r} \tag{6.9}$$

or, in matrix form, as the following.

$$\begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \\ \frac{1}{r} & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_z \end{bmatrix} = \begin{bmatrix} \epsilon_{rr} \\ \epsilon_{zz} \\ 2\epsilon_{rz} \\ \epsilon_{\theta\theta} \end{bmatrix} \tag{6.10}$$

Also due to the change of coordinates, the differential area and volume in the balance of linear momentum (6.2) reduce to (assuming the integration in the θ direction has been applied, $\int_0^{2\pi}(\cdot)\mathrm{d}\theta = 2\pi$ if the integrand is independent of θ)

$$\mathrm{d}v = 2\pi r \mathrm{d}r \mathrm{d}z \tag{6.11}$$

$$\mathrm{d}a = 2\pi r\sqrt{\mathrm{d}r^2 + \mathrm{d}z^2} \tag{6.12}$$

resulting in the following expression, in cylindrical coordinates, for the weak form of the balance of linear momentum.

$$2\pi\left[\int_{\mathcal{B}} \frac{\partial w_i}{\partial x_j}\sigma_{ij} r^h \mathrm{d}r\mathrm{d}z - \int_{\Gamma^t} w_i t_i r^h \sqrt{\mathrm{d}r^2 + \mathrm{d}z^2}\right] = 0 \tag{6.13}$$

The integration of the balance equation is done over a series of finite elements. The Galerkin, or discretized, form of the balance equation is written

$$2\pi\left[\int_{\mathcal{B}^h}\frac{\partial w_i^h}{\partial x_j^h}\sigma_{ij}^h r\mathrm{d}r\mathrm{d}z-\int_{\Gamma^t}w_i^h t_i r\sqrt{\mathrm{d}r^2+\mathrm{d}z^2}\right]=0 \tag{6.14}$$

where the superscript h denotes variables discretized into finite elements. The geometry of the finite elements are described in the two dimensional natural coordinate system where the global position vector $\boldsymbol{r}=\begin{bmatrix} r & z\end{bmatrix}$ is mapped to the natural position vector $\boldsymbol{\xi}=\begin{bmatrix}\xi & \eta\end{bmatrix}$. Given bilinear quadrilateral interpolation functions (N) for each of the four element nodes, the element displacement ($\boldsymbol{u}^{h^e}$) and weighting function ($\boldsymbol{w}^{h^e}$) are discretized as

$$\boldsymbol{u}^{h^e}(\boldsymbol{\xi},t)=\sum_{a=1}^{4}N_a(\boldsymbol{\xi})\boldsymbol{d}_a^e \tag{6.15}$$

$$\boldsymbol{w}^{h^e}(\boldsymbol{\xi},t)=\sum_{a=1}^{4}N_a(\boldsymbol{\xi})\boldsymbol{c}_a^e \tag{6.16}$$

where N_a is the nodal interpolation functions, $\boldsymbol{d}_a^e$ is the element nodal displacement vector, $\boldsymbol{c}_a^e$ is the element nodal weighting function vector, e is the finite element number, and a is the element node number. In matrix form, the displacement and weighting function are

$$\boldsymbol{u}^{h^e}(\boldsymbol{\xi},t)=\begin{bmatrix}\boldsymbol{N}_1^e & \boldsymbol{N}_2^e & \boldsymbol{N}_3^e & \boldsymbol{N}_4^e\end{bmatrix}\begin{bmatrix}\boldsymbol{d}_1^e\\ \boldsymbol{d}_2^e\\ \boldsymbol{d}_3^e\\ \boldsymbol{d}_4^e\end{bmatrix} \tag{6.17}$$

$$\boldsymbol{w}^{h^e}(\boldsymbol{\xi},t)=\begin{bmatrix}\boldsymbol{N}_1^e & \boldsymbol{N}_2^e & \boldsymbol{N}_3^e & \boldsymbol{N}_4^e\end{bmatrix}\begin{bmatrix}\boldsymbol{c}_1^e\\ \boldsymbol{c}_2^e\\ \boldsymbol{c}_3^e\\ \boldsymbol{c}_4^e\end{bmatrix} \tag{6.18}$$

where, per node a, the interpolation matrix ($\boldsymbol{N}_a^e$), nodal displacement, and nodal weighting function are the following matrices.

$$\boldsymbol{N}_a^e = \begin{bmatrix} N_a & 0 \\ 0 & N_a \end{bmatrix} \tag{6.19}$$

$$\boldsymbol{d}_a^e = \begin{bmatrix} d_{r(a)}^e \\ d_{z(a)}^e \end{bmatrix} \tag{6.20}$$

$$\boldsymbol{c}_a^e = \begin{bmatrix} c_{r(a)}^e \\ c_{z(a)}^e \end{bmatrix} \tag{6.21}$$

As a result, the calculation of the displacement and weighting function is between the 2×8 matrix of interpolation functions and the 8×1 matrix of the displacement or weighting functions. Similarly, the strain ($\boldsymbol{\epsilon}$) is calculated by $\boldsymbol{\epsilon}^{h^e} = \boldsymbol{B}^e \cdot \boldsymbol{d}^e$ where $\boldsymbol{B}^e$ is the strain-displacement matrix.

$$\boldsymbol{B}^e = \frac{\partial \boldsymbol{N}^e}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{B}_1^e & \boldsymbol{B}_2^e & \boldsymbol{B}_3^e & \boldsymbol{B}_4^e \end{bmatrix} \quad ; \quad \boldsymbol{B}_a^e = \begin{bmatrix} \frac{\partial N_a}{\partial r} & 0 \\ 0 & \frac{\partial N_a}{\partial z} \\ \frac{N_a}{r} & 0 \\ \frac{\partial N_a}{\partial z} & \frac{\partial N_a}{\partial r} \end{bmatrix} \tag{6.22}$$

Therefore, the balance of linear momentum in finite element matrix form in the natural coordinate system is the following equation

$$\underset{e=1}{\overset{n_{el}}{\mathbf{A}}} \Bigg[2\pi (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \Bigg(\underbrace{\int_{-1}^{1}\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} \cdot \boldsymbol{\sigma}\; r(\boldsymbol{\xi}) j^e \mathrm{d}\xi \mathrm{d}\eta}_{\boldsymbol{f}^{\mathrm{int}}} - \underbrace{\int_{-1}^{1} (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{t}\; r(\xi = 1) \sqrt{\left(\frac{\partial r}{\partial \xi}\right)^2 + \left(\frac{\partial z}{\partial \xi}\right)^2} \mathrm{d}\xi}_{\boldsymbol{f}^{\mathrm{ext}}_{\eta=1}}$$

$$- \underbrace{\int_{-1}^{1} (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{t}\; r(\eta = 1) \sqrt{\left(\frac{\partial r}{\partial \eta}\right)^2 + \left(\frac{\partial z}{\partial \eta}\right)^2} \mathrm{d}\eta}_{\boldsymbol{f}^{\mathrm{ext}}_{\xi=1}} \Bigg) \Bigg] = 0 \tag{6.23}$$

where n_{el} is the total number of finite elements, j^e is the Jacobian of the coordinate transformation (i.e. $j^e = \det(\boldsymbol{J}^e)$ where $\boldsymbol{J}^e = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\xi}}$), and the radius ($r$) is a function of the integration point's position in the natural coordinate system. After applying essential boundary conditions, assuming an arbitrary $\boldsymbol{c}^e$ for all degrees of freedom, and assembling the matrix equation over all finite elements (signified by the assembly operator $\mathop{\mathbf{A}}_{e=1}^{n_{el}}$), the final assembled matrix form of the balance of linear momentum is

$$\boldsymbol{r}(\boldsymbol{d}) = \boldsymbol{f}^{\text{int}}(\boldsymbol{d}) - \boldsymbol{f}^{\text{ext}} = \boldsymbol{0} \tag{6.24}$$

where $\boldsymbol{f}^{\text{int}}$ is the internal force vector, $\boldsymbol{f}^{\text{ext}} = \boldsymbol{f}^{\text{ext}}_{\xi=1} + \boldsymbol{f}^{\text{ext}}_{\eta=1}$ is the sum of the external force vectors, and $\boldsymbol{r}(\boldsymbol{d})$ is the residual force vector. External tractions can theoretically be applied to any element boundary, but looking forward to simulating triaxial tests, the external tractions will be applied to both the $\xi = 1$ and $\eta = 1$ boundaries of some finite elements.

Considering the expected non-linearity of the DEM assembly's stress-strain response as incorporated into the FEM-DEM hierarchical multiscale model, the balance of linear momentum was solved using the Newton-Raphson algorithm. In general, the displacement solution at the current iteration of the algorithm ($\boldsymbol{d}^{k+1}$) is assumed to be equal to the previous iteration solution ($\boldsymbol{d}^{k}$) added with the increment of displacement ($\delta\boldsymbol{d}^{k}$) (i.e. $\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \delta\boldsymbol{d}^{k}$). The residual of iteration (k) is calculated by taking the difference between the internal and external forcing vectors ($\boldsymbol{r}(\boldsymbol{d}^{k}) = \boldsymbol{f}^{\text{int}}(\boldsymbol{d}^{k}) - \boldsymbol{f}^{\text{ext}}$) after recognizing that $\boldsymbol{f}^{\text{int}}$ is a function of the displacement in that the Cauchy stress ($\boldsymbol{\sigma}$) is a function of the displacement through the strain. Following the Newton-Raphson algorithm, $\delta\boldsymbol{d}^{k}$ is calculated by the following.

$$\delta\boldsymbol{d}^{k} = \left(\frac{\partial \boldsymbol{r}(\boldsymbol{d}^{k})}{\partial \boldsymbol{d}}\right)^{-1} \left[-(\boldsymbol{r}(\boldsymbol{d}^{k}))\right] \tag{6.25}$$

The internal force vector is the only component of $\boldsymbol{r}(\boldsymbol{d}^{k})$ that is a function of the displacement; therefore, the derivative of the residual with respect to the displacement is

$$\frac{\partial \boldsymbol{r}^{e}}{\partial \boldsymbol{d}^{e}} = \frac{\partial \boldsymbol{f}^{\text{int},e}}{\partial \boldsymbol{d}^{e}} = \left(\underbrace{\int_{-1}^{1}\int_{-1}^{1} (\boldsymbol{B}^{e})^{\text{T}} \cdot \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} \cdot \boldsymbol{B}^{e}\ r(\boldsymbol{\xi}) j^{e} \text{d}\xi \text{d}\eta}_{\boldsymbol{k}^{\text{int},e}(\boldsymbol{d}^{e})}\right) \tag{6.26}$$

where $\boldsymbol{k}^{\text{int},e}$ is the element stiffness matrix, which when assembled, is equal to the global stiffness matrix ($\boldsymbol{K}$). As a result, $\frac{\partial \boldsymbol{r}(\boldsymbol{d}^{k})}{\partial \boldsymbol{d}} = \boldsymbol{K}$. The material consistent tangent ($\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}}$) is approximated in the hierarchical upscaling algorithm by the DEM assembly using equation 2.4. The algorithm iterates until the residual is satisfactorily minimized to approximately zero, or a specified tolerance approaching zero (tol)

$$\frac{|\boldsymbol{r}(\boldsymbol{d}^k)|}{|\boldsymbol{r}(\boldsymbol{d}^0)|} < \mathrm{tol} \tag{6.27}$$

where $\boldsymbol{r}(\boldsymbol{d}^0)$ is the residual in the first iteration of the algorithm, at which time $\boldsymbol{d}^{k+1}$ is taken as the correct displacement at the current timestep and the algorithm moves to the next timestep.

6.1.1 Modified and Quasi-Newton Methods

During initial testing of the two dimensional axisymmetric FEM-DEM hierarchical multiscale model, it was discovered that the Newton-Raphson algorithm had difficulty reaching convergence. As a result, two additional implicit Newton-family algorithms were developed to solve the balance of linear momentum: the modified Newton method and the quasi-Newton method via the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Fletcher, 1987). The modified Newton method is essentially the same as the Newton-Raphson algorithm except that the tangent is not updated in every iteration. Instead, it is calculated during the first iteration ($k = 0$) and then used to update $\delta\boldsymbol{d}^k$ during each successive iteration (equation 6.28).

$$\delta\boldsymbol{d}^k = \left(\frac{\partial\boldsymbol{r}(\boldsymbol{d}^0)}{\partial\boldsymbol{d}}\right)^{-1}\left[-(\boldsymbol{r}(\boldsymbol{d}^k))\right] \tag{6.28}$$

The BFGS algorithm is a quasi-Newton, or “secant”-like, method where the tangent is approximated by directly calculating the change in residual over the change in displacement ($\frac{\partial\boldsymbol{r}(\boldsymbol{d}^k)}{\partial\boldsymbol{d}} \approx \frac{\Delta\boldsymbol{r}}{\Delta\boldsymbol{d}} = \frac{(\boldsymbol{r}(\boldsymbol{d}^{k+1})-\boldsymbol{r}(\boldsymbol{d}^k))}{\boldsymbol{d}^{k+1}-\boldsymbol{d}^k}$). This type of approximation is an advantage if a closed form consistent tangent cannot be derived. In the case of the hierarchical multiscale modeling with DEM, the consistent tangent is already approximated by the homogeneous elastic bulk modulus of the DEM assembly, and issues with the tangent manifest as difficulty of the Newton-Raphson or Modified Newton algorithms converging. Using the BFGS algorithm replaces the approximate consistent tangent with a different approximation derived from the residual. Information regarding the BFGS algorithm's implementation as well as example code can be found in Kelley (1995).

6.1.2 Explicit Formulation with Dynamic Relaxation

Similar to the one dimensional, uniaxial strain FEM-DEM model, a second, explicit formulation of the two dimensional axisymmetric FEM-DEM model was also developed. However, in order to maintain the quasi-static nature of the modeled triaxial tests, a dynamic relaxation technique was used to overdamp the system and find the static solution (Underwood, 1983). This was also done to significantly increase the critical timestep required for numerical stability of the solution.

Starting with the localized strong form of the balance of linear momentum including inertial terms

$$\frac{\partial \sigma_{ij}}{\partial x_j} - \rho \ddot{u}_i = 0 \tag{6.29}$$

where ρ is the mass density of the material and $\ddot{u}_i$ is the acceleration of the body, the weak form of the balance of linear momentum is derived using the method of weighted residuals.

$$\int_{\mathcal{B}} \rho w_i \ddot{u}_i \mathrm{d}v + \int_{\mathcal{B}} \frac{\partial w_i}{\partial x_j} \sigma_{ij} \mathrm{d}v - \int_{\Gamma^t} w_i t_i \mathrm{d}a = 0 \tag{6.30}$$

After converting to the cylindrical coordinate system (equations 6.3, 6.4, 6.5, 6.10, and 6.11), the balance of linear momentum becomes the following.

$$2\pi \left[\int_{\mathcal{B}} \rho w_i \ddot{u}_i r \mathrm{d}r \mathrm{d}z + \int_{\mathcal{B}} \frac{\partial w_i}{\partial x_j} \sigma_{ij} r \mathrm{d}r \mathrm{d}z - \int_{\Gamma^t} w_i t_i r \sqrt{\mathrm{d}r^2 + \mathrm{d}z^2} \right] = 0 \tag{6.31}$$

Using the same discretizations as the implicit versions (equations 6.15 through 6.22) including an equation for the element acceleration vector ($\ddot{\boldsymbol{u}}^{h^e}$)

$$\ddot{\boldsymbol{u}}^{h^e}(\boldsymbol{\xi}, t) = \sum_{a=1}^{4} N_a(\boldsymbol{\xi}) \boldsymbol{a}_a^e \tag{6.32}$$

where $\boldsymbol{a}^e_a$ is the element nodal acceleration vector, the balance of linear momentum in finite element matrix form in the natural coordinate system is the following equation

$$\begin{multline}\mathop{\mathsf{A}}_{e=1}^{n_{el}} \Big[2\pi (\boldsymbol{c}^e)^{\mathrm{T}} \cdot \Big(\underbrace{\int_{-1}^{1}\int_{-1}^{1} \rho (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{N}^e r(\boldsymbol{\xi}) j^e \mathrm{d}\xi \mathrm{d}\eta}_{\boldsymbol{m}^e} \cdot \boldsymbol{a}^e + \underbrace{\int_{-1}^{1}\int_{-1}^{1} (\boldsymbol{B}^e)^{\mathrm{T}} \cdot \boldsymbol{\sigma}\; r(\boldsymbol{\xi}) j^e \mathrm{d}\xi \mathrm{d}\eta}_{\boldsymbol{f}^{\mathrm{int}}} \\ \underbrace{- \int_{-1}^{1} (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{t}\; r(\xi = 1) \sqrt{\left(\frac{\partial r}{\partial \xi}\right)^2 + \left(\frac{\partial z}{\partial \xi}\right)^2} \mathrm{d}\xi}_{\boldsymbol{f}^{\mathrm{ext}}_{\eta=1}} \\ \underbrace{- \int_{-1}^{1} (\boldsymbol{N}^e)^{\mathrm{T}} \cdot \boldsymbol{t}\; r(\eta = 1) \sqrt{\left(\frac{\partial r}{\partial \eta}\right)^2 + \left(\frac{\partial z}{\partial \eta}\right)^2} \mathrm{d}\eta}_{\boldsymbol{f}^{\mathrm{ext}}_{\xi=1}} \Big) \Big] = 0 \tag{6.33}\end{multline}$$

where $\boldsymbol{m}^e$ is the element mass matrix. After assembly, the balance of linear momentum becomes

$$\boldsymbol{M} \cdot \boldsymbol{a} + \boldsymbol{f}^{\mathrm{int}}(\boldsymbol{d}) - \boldsymbol{f}^{\mathrm{ext}} = \boldsymbol{0} \tag{6.34}$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{a}$ is the acceleration vector, and, like the implicit formulation, $\boldsymbol{f}^{\mathrm{ext}} = \boldsymbol{f}^{\mathrm{ext}}_{\xi=1} + \boldsymbol{f}^{\mathrm{ext}}_{\eta=1}$.

The explicit formulation is discretized in time using the Newmark method for use with the dynamic relaxation algorithm. First, mass proportional damping is added to the assembled balance of linear momentum

$$\boldsymbol{M} \cdot \boldsymbol{a} + \boldsymbol{C} \cdot \boldsymbol{v} + \boldsymbol{f}^{\mathrm{int}}(\boldsymbol{d}) - \boldsymbol{f}^{\mathrm{ext}} = \boldsymbol{0} \tag{6.35}$$

where $\boldsymbol{v}$ is the velocity vector and $\boldsymbol{C}$ is the damping matrix, which is proportional to the mass by a factor c (i.e. $\boldsymbol{C} = c\boldsymbol{M}$). The kinematic variables are discretized in time by

$$\boldsymbol{d}_{n+1} = \underbrace{\boldsymbol{d}_n + \Delta t \boldsymbol{v}_n + \frac{1}{2}\Delta t^2(1-2\beta)\boldsymbol{a}_n}_{\tilde{\boldsymbol{d}}_{n+1}} + \Delta t^2 \beta \boldsymbol{a}_{n+1} \tag{6.36}$$

$$\boldsymbol{v}_{n+1} = \underbrace{\boldsymbol{v}_n + \Delta t(1-\gamma)\boldsymbol{a}_n}_{\tilde{\boldsymbol{v}}_{n+1}} + \Delta t \gamma \boldsymbol{a}_{n+1} \tag{6.37}$$

where $n+1$ signifies the current timestep and n signifies the previous timestep. The Newmark parameters β and γ were set to 0 and $\frac{1}{2}$, respectively, to use the explicit central difference in time method. The discretized kinematics are substituted into the balance of linear momentum

$$\boldsymbol{M} \cdot \boldsymbol{a}_{n+1} + \boldsymbol{C} \cdot (\tilde{\boldsymbol{v}}_{n+1} + \gamma \Delta t \boldsymbol{a}_{n+1}) = \boldsymbol{f}^{\text{ext}} - \boldsymbol{f}^{\text{int}}(\boldsymbol{d}_{n+1}) \tag{6.38}$$

and the equation is solved in each timestep for the acceleration.

$$\boldsymbol{a}_{n+1} = (\boldsymbol{M} + \gamma \Delta t \boldsymbol{C})^{-1} \cdot (\boldsymbol{f}^{\text{ext}} - \boldsymbol{f}^{\text{int}}(\boldsymbol{d}_{n+1}) - \boldsymbol{C} \cdot \tilde{\boldsymbol{v}}_{n+1}) \tag{6.39}$$

Note that given the Newmark parameters for the central difference in time method, the displacement and displacement predictor are equal ($\boldsymbol{d}_{n+1} = \tilde{\boldsymbol{d}}_{n+1}$).

6.2 Hierarchical Multiscale Model Development

With the finite element model and multiple solution schemes developed, the DEM code ellip3D (described in detail in section 4.2) was again integrated with the FEM implementations to create the two dimensional axisymmetric FEM-DEM hierarchical multiscale model. Just as with the one dimensional model outlined in chapter 4, the strain calculated by solving the global (FEM) boundary value problem is passed to DEM RVEs located at each of the integration points of the finite elements. Each DEM RVE is allowed to deform given the applied strain, and the homogenized stress and an approximate consistent tangent (if required) at the end of the deformation are calculated and passed back to the FEM. The deformations of the DEM assemblies, in essence, replaces the material's constitutive model.

The 2D axisymmetric FEM-DEM model was tested against data from two triaxial experiments on dry, drained Colorado Mason Sand completed as part of the Soil Blast MURI Project. Both tests were consolidated drained (CD) experiments signifying that pore air was allowed to drain from the sand specimen during both the isotropic compression and shear phases of the triaxial test. The first was a standard triaxial experiment conducted by Jenna Svoboda and Dr. John McCartney of the University of Colorado Boulder (Dr. McCartney is currently an Associate Professor at the University of California San Diego), and the second was a small-scale triaxial experiment performed and imaged in a synchrotron electron microscope at the Argonne National Laboratory by Drs. Andrew Druckrey and Khalid Alshibli of the University of Tennessee Knoxville. For both cases, the FEM-DEM models were the same with the exception of differing initial geometry, applied displacement, and applied tractions to represent the specific experimental design.

To model the triaxial experiments and to keep the FEM-DEM model as simple as possible during testing, a one, bilinear quadrilateral element mesh was generated. To mimic the triaxial tests, the simulations were split into two separate simulations. First, an isotropic compression simulation (preconditioner) was developed to model the "consolidated" stage of the triaxial tests. To do this, the cell pressure (σ_c) prescribed by the experiments was applied to both the top and outer edge of the element. Given the applied cell pressure, the top and outer edge were allowed to displace while the bottom was fixed in the axial (z) direction and the inner edge was fixed in the radial (r) direction. A schematic of the FEM mesh, applied tractions, and degrees of freedom can be found on figure 6.1.

Second, a shearing simulation was developed to mimic the preconditioner and to model the "drained" stage of the triaxial experiments. The simulation was run from 0-2 pseudo-seconds; the isotropic compression simulation was run again over 0-1 pseudo-seconds and the shearing simulation over 1-2 pseudo-seconds. As seen on figure 6.2, the natural boundary condition along the top of the specimen from the preconditioner (figure 6.1) was replaced with an applied displacement for the shearing simulation. The isotropic compression simulation was then mimicked over the first pseudo-second by applying the displacement solution (d_3 and d_4) from the preconditioner as the

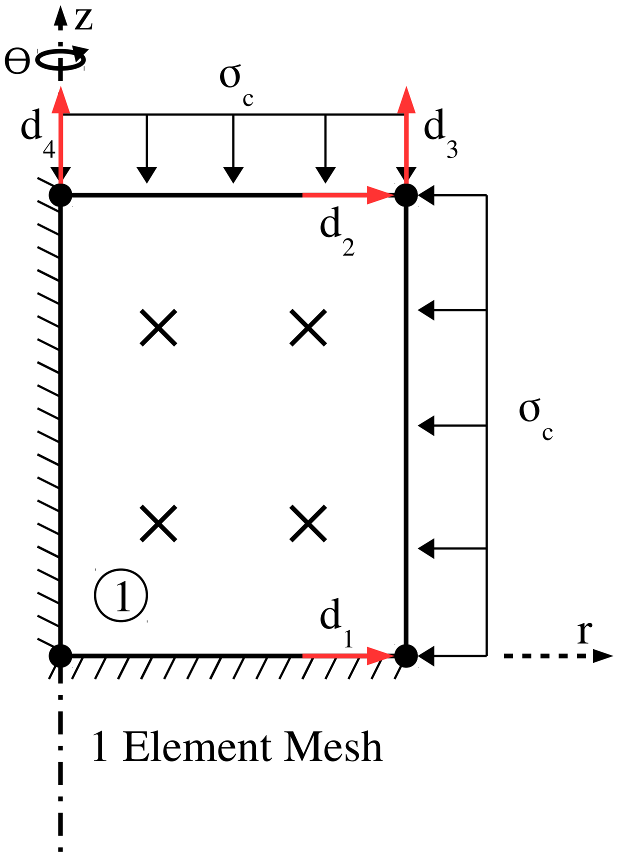


Figure 6.1: One Element Triaxial Test Mesh: Isotropic Compression Step

applied displacement over the first pseudo-second in the shearing simulation. With the isotropic step completed, the specimen was sheared by specifying the maximum displacement from the triaxial experiments and linearly ramping the displacement of the top of the specimen over the second pseudo-second through $d(t)$. The cell pressure (σ_c) was applied to the outer edge of the specimen during both stages of the shearing simulation, and the displacements of the bottom and inner edges of the specimen were once again restricted as in the isotropic compression simulation.

To model the Colorado Mason Sand, DEM RVEs were positioned at each of the four integration points of the single element mesh (figure 6.3). Strain ($\boldsymbol{\epsilon}$) calculated in the FEM was passed to each RVE as boundary conditions, and after the DEM was allowed to deform, the homogenized Cauchy stress ($\boldsymbol{\sigma}$) and an approximate tangent ($\boldsymbol{D}$) were then passed back to the FEM. To apply

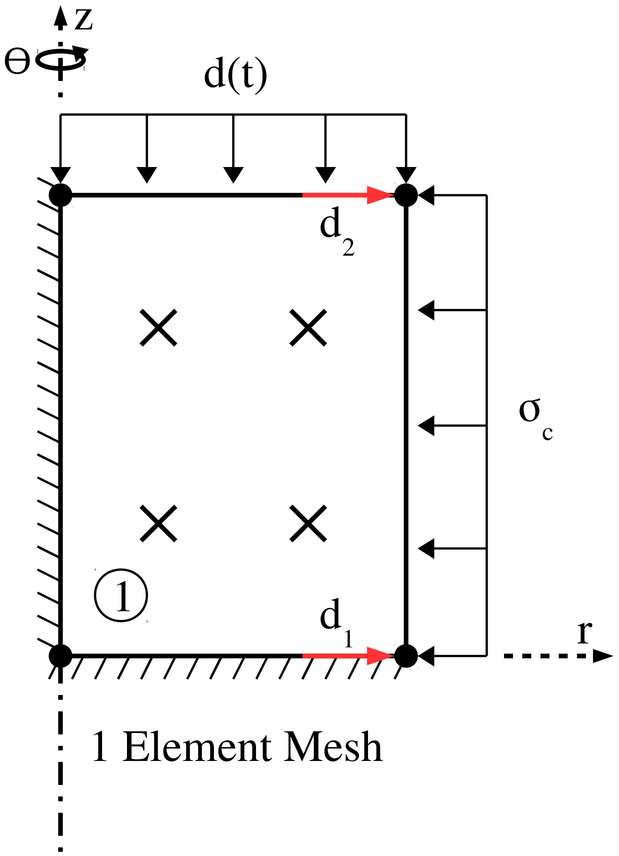


Figure 6.2: One Element Triaxial Test Mesh: Shearing Step

boundary conditions to the 3D DEM RVE, the 2D axisymmetric strain from the FEM was converted to wall displacements in the three normal directions. The displacement of each wall was derived from the strain by the following set of equations

$$\epsilon_{rr} = \frac{\partial u_r}{\partial r} \approx \frac{d_y(t)}{L_0^{\text{RVE}}} \quad \Longrightarrow \quad d_y(t) = L_0^{\text{RVE}} \epsilon_{rr} \tag{6.40}$$

$$\epsilon_{zz} = \frac{\partial u_z}{\partial z} \approx \frac{d_z(t)}{H_0^{\text{RVE}}} \quad \Longrightarrow \quad d_z(t) = H_0^{\text{RVE}} \epsilon_{zz} \tag{6.41}$$

$$\epsilon_{\theta\theta} = \frac{u_r}{r} \approx \frac{d_x(t)}{W_0^{\text{RVE}}} \quad \Longrightarrow \quad d_x(t) = W_0^{\text{RVE}} \epsilon_{\theta\theta} \tag{6.42}$$

as can be seen on figure 6.4. The axisymmetric shear strain (ϵ_{rz}) should also be applied to the DEM RVEs through displacing the $+z$ wall in the y-direction, but current versions of ellip3D do

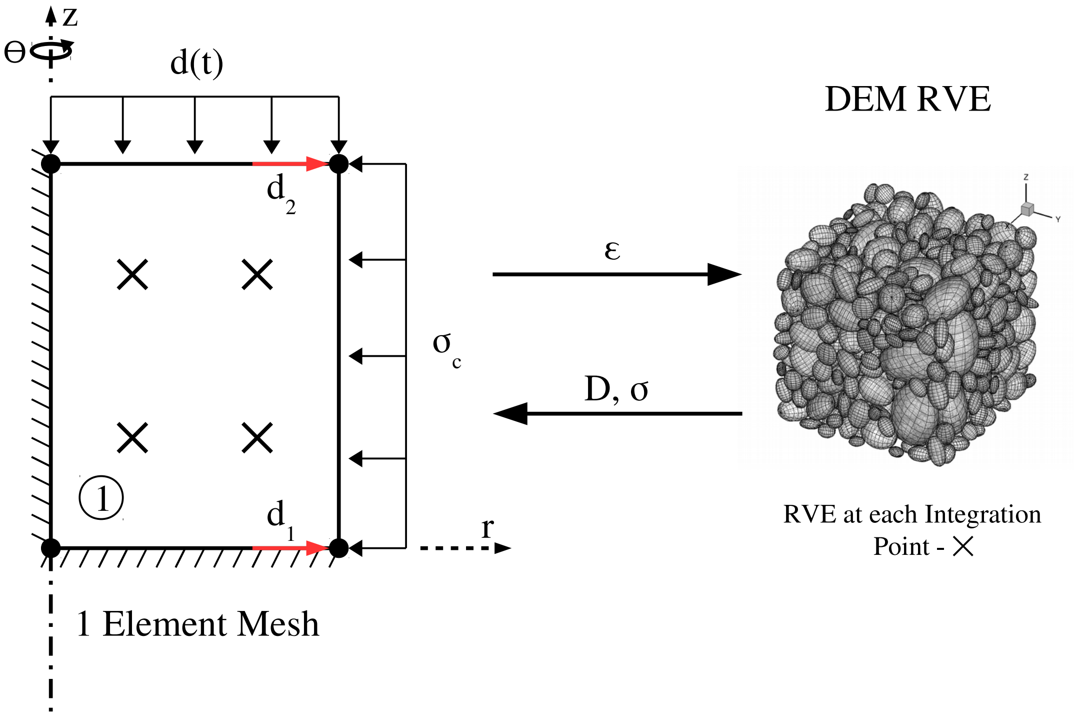


Figure 6.3: 2D Axisymmetric FEM to DEM RVE

not allow for this type of wall deformation.

Just as the 2D axisymmetric strain was converted to three dimensions as boundary conditions on the DEM RVE, the 3D homogenized Cauchy stress and consistent tangent were reduced to equivalent 2D asymmetric versions. The full (non-symmetric) 3D homogenized Cauchy stress vector calculated using equation 4.53 reduces to the 2D axisymmetric Cauchy stress vector (Voigt notation) by

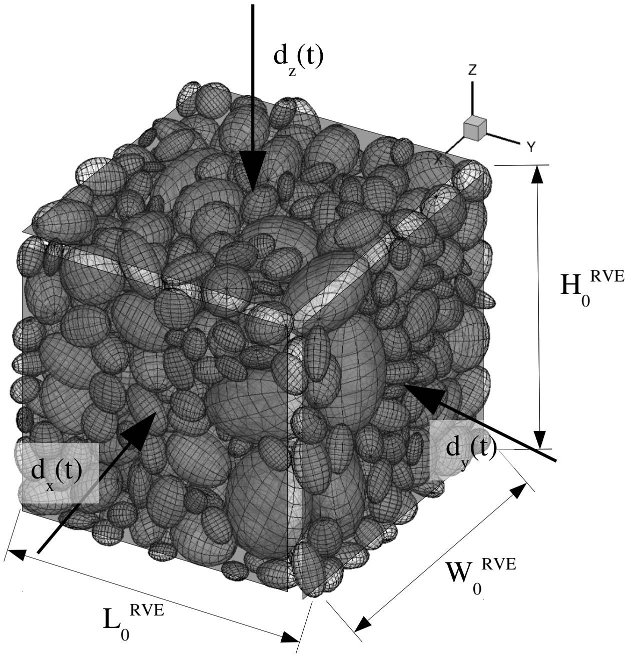


Figure 6.4: Applying Displacement to the DEM RVE

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{zy} \\ \sigma_{zz} \end{bmatrix} \Longrightarrow \begin{bmatrix} \sigma_{rr}^{\text{axi}} \\ \sigma_{zz}^{\text{axi}} \\ \sigma_{rz}^{\text{axi}} \\ \sigma_{\theta\theta}^{\text{axi}} \end{bmatrix} \tag{6.43}$$

where the following components are equivalent.

$$\begin{aligned} \sigma_{yy} &= \sigma_{rr}^{\text{axi}} \qquad & \sigma_{zz} &= \sigma_{zz}^{\text{axi}} \\ \sigma_{yz} &= \sigma_{rz}^{\text{axi}} \qquad & \sigma_{xx} &= \sigma_{\theta\theta}^{\text{axi}} \end{aligned} \tag{6.44}$$

Similarly, the full (non-symmetric) 3D approximate consistent tangent calculated by equation 4.54 is reduced to the 2D axisymmetric approximate consistent tangent by

$$\begin{bmatrix} D_{xxxx} & D_{xxxy} & D_{xxxz} & D_{xxyx} & D_{xxyy} & D_{xxyz} & D_{xxzx} & D_{xxzy} & D_{xxzz} \\ D_{xyxx} & D_{xyxy} & D_{xyxz} & D_{xyyx} & D_{xyyy} & D_{xyyz} & D_{xyzx} & D_{xyzy} & D_{xyzz} \\ D_{xzxx} & D_{xzxy} & D_{xzxz} & D_{xzyx} & D_{xzyy} & D_{xzyz} & D_{xzzx} & D_{xzzy} & D_{xzzz} \\ D_{yxxx} & D_{yxxy} & D_{yxxz} & D_{yxyx} & D_{yxyy} & D_{yxyz} & D_{yxzx} & D_{yxzy} & D_{yxzz} \\ D_{yyxx} & D_{yyxy} & D_{yyxz} & D_{yyyx} & D_{yyyy} & D_{yyyz} & D_{yyzx} & D_{yyzy} & D_{yyzz} \\ D_{yzxx} & D_{yzxy} & D_{yzxz} & D_{yzyx} & D_{yzyy} & D_{yzyz} & D_{yzzx} & D_{yzzy} & D_{yzzz} \\ D_{zxxx} & D_{zxxy} & D_{zxxz} & D_{zxyx} & D_{zxyy} & D_{zxyz} & D_{zxzx} & D_{zxzy} & D_{zxzz} \\ D_{zyxx} & D_{zyxy} & D_{zyxz} & D_{zyyx} & D_{zyyy} & D_{zyyz} & D_{zyzx} & D_{zyzy} & D_{zyzz} \\ D_{zzxx} & D_{zzxy} & D_{zzxz} & D_{zzyx} & D_{zzyy} & D_{zzyz} & D_{zzzx} & D_{zzzy} & D_{zzzz} \end{bmatrix} \Longrightarrow \begin{bmatrix} D_{rrrr}^{\text{axi}} & D_{rrzz}^{\text{axi}} & D_{rrrz}^{\text{axi}} & D_{rr\theta\theta}^{\text{axi}} \\ D_{zzrr}^{\text{axi}} & D_{zzzz}^{\text{axi}} & D_{zzrz}^{\text{axi}} & D_{zz\theta\theta}^{\text{axi}} \\ D_{rzrr}^{\text{axi}} & D_{rzzz}^{\text{axi}} & D_{rzrz}^{\text{axi}} & D_{rz\theta\theta}^{\text{axi}} \\ D_{\theta\theta rr}^{\text{axi}} & D_{\theta\theta zz}^{\text{axi}} & D_{\theta\theta rz}^{\text{axi}} & D_{\theta\theta\theta\theta}^{\text{axi}} \end{bmatrix} \tag{6.45}$$

where the following components are equivalent after assuming isotropy (Hughes, 1987). DEM can be used to consider a soil's fabric and other inherent or induced anisotropy, so it is likely that assuming isotropy is not necessarily a good assumption. For these initial simulations and during code testing, isotropy was assumed for simplicity, but can be relaxed in the future.

$$
\begin{aligned}
&D_{yyyy} = D^{\text{axi}}_{rrrr} && D_{yyzz} = D^{\text{axi}}_{rrzz} = D^{\text{axi}}_{zzrr} && D_{yyyz} = D^{\text{axi}}_{rrrz} = D^{\text{axi}}_{rzrr} \\
&D_{yyxx} = D^{\text{axi}}_{rr\theta\theta} = D^{\text{axi}}_{\theta\theta rr} && D_{zzzz} = D^{\text{axi}}_{zzzz} && D_{zzyz} = D^{\text{axi}}_{zzrz} = D^{\text{axi}}_{rzzz} \\
&D_{zzxx} = D^{\text{axi}}_{zz\theta\theta} = D^{\text{axi}}_{\theta\theta zz} && D_{yzyz} = D^{\text{axi}}_{rzrz} && D_{yzxx} = D^{\text{axi}}_{rz\theta\theta} = D^{\text{axi}}_{\theta\theta rz} \\
& && D_{xxxx} = D^{\text{axi}}_{\theta\theta\theta\theta} &&
\end{aligned}
\tag{6.46}
$$

6.3 Implicit Formulation Triaxial Compression Results

Considering the potential numerical stability advantages, the first simulations of the triaxial compression tests were attempts to model the standard scale triaxial compression experimental results using the implicit FEM formulation with the full Newton-Raphson algorithm. However, convergence issues plagued the attempts. As briefly described in section 4.3.3 with regards to the 1D FEM-DEM model, it was possible that over a given timestep the particles could dramatically rearrange if, for example, one larger particle slid past another. The Newton-Raphson algorithm was designed, in general, to solve non-linear differential equations in discrete form, but can have difficulty if the consistent tangent doesn't push the residual to zero. In the case of the 2D FEM-DEM model, the approximation of the consistent tangent coupled with the occasionally highly non-linear deformations render the Newton-Raphson algorithm unfit for this type of problem. The simulations rarely converged in less than 100 iterations sometimes requiring well over 1000 and lower than usual tolerances to reach convergence. Considering that the DEM component of FEM-DEM models almost exclusively determines the computational speed of the simulations, convergence rates of these magnitudes quickly make the Newton-Raphson algorithm intractable for these simulations.

Two other implicit formulations were also attempted: the modified Newton-Raphson method and a quasi-Newton method called the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. As the modified Newton-Raphson algorithm is essentially the Newton-Raphon algorithm but with the consistent tangent held constant, it was used to test if the approximate consistent tangent was even capable of driving the solution to convergence. Although it was more stable (i.e. more likely to converge) than the full Newton-Raphson simulations, it required an average of 500 iterations also making it intractable. The BFGS algorithm was also implemented in an attempt to remove the solution's dependence on the approximate consistent tangent as quasi-Newton methods rely on the value of the displacement and residual only. However, the BFGS algorithm did not work well with the DEM in this application. If the algorithm overestimated the displacement, as happens often during the iterative process, the applied displacements to the DEM assemblies were also overestimated. If these displacements were too large, the walls of the assembly overlapped causing unphysical deformations and homogenized stress values. The unphysical stress values caused unphysical residuals, which had difficulty converging.

6.4 Explicit Formulation Standard Triaxial Compression Results

After the difficulties with the three implicit formulations, the standard triaxial compression experimental results were modeled using the explicit FEM-DEM formulation. Like with the 1D FEM-DEM simulations (see chapter 4), the explicit version had the advantage of removing the iterations from the implicit versions only requiring one run of ellip3D per integration point per global timestep, which significantly lowered the overall runtime of the code. However, the conditional stability of the formulation required extensive tuning of the timestep and dynamic relaxation parameters before physically meaningful results were simulated. Difficulties tuning the FEM dynamic relaxation parameters were compounded by the need to determine the minimum number of DEM timesteps required to accurately simulate the soil using dynamic relaxation in the DEM to model the quasi-static nature of the experiments. Too few timesteps and the DEM simulations didn't complete (i.e. the particle movements didn't reach steady-state under the dynamic relaxation) resulting in unphysical results, but too many timesteps unnecessarily increased the overall length of the simulation. After tuning the FEM-DEM simulations, the standard triaxial compression experimental results were simulated using the two step approach outlined in section 6.2. The following two sections present those results.

6.4.1 Isotropic Compression Step

FEM displacement and strain results for the 2D FEM-DEM model over the isotropic compression step can be found on figures 6.5 and 6.6. The degree of freedom numbering scheme can be found on figure 6.1. The displacement of both side nodes inward (d_1 and d_2) and both top nodes downward (d_3 and d_4) were approximately the same meaning that the entire top and bottom of the specimen displaced the same distance, as expected given the applied tractions. Also, as can be seen on figure 6.7, the homogenized Cauchy stress in integration point 3 (top right side) was higher in the radial (rr) than the axial (zz) direction. This indicates that in the initial state of the DEM RVEs, the top wall (z on figure 6.4) needed to displace further than the sides (x and y) to generate the required isotropic stress. The DEM assemblies were generated through a gravity deposition process with the particles dropped into the assembly from the z direction, so it's likely that the particles were initially packed more tightly in the x and y directions.

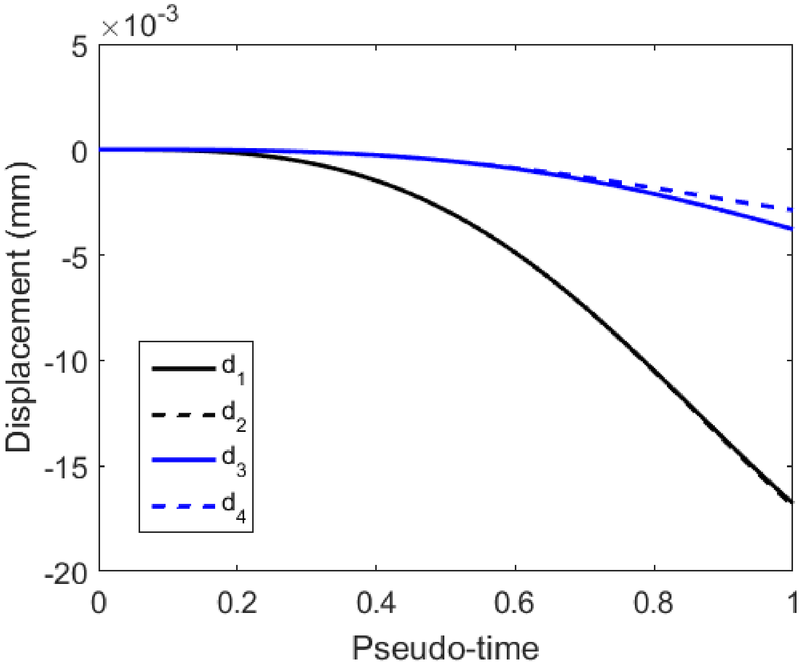


Figure 6.5: Isotropic Compression Step: FEM Displacement Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

At the end of the one pseudo-second simulation, the Cauchy stress in both the axial and

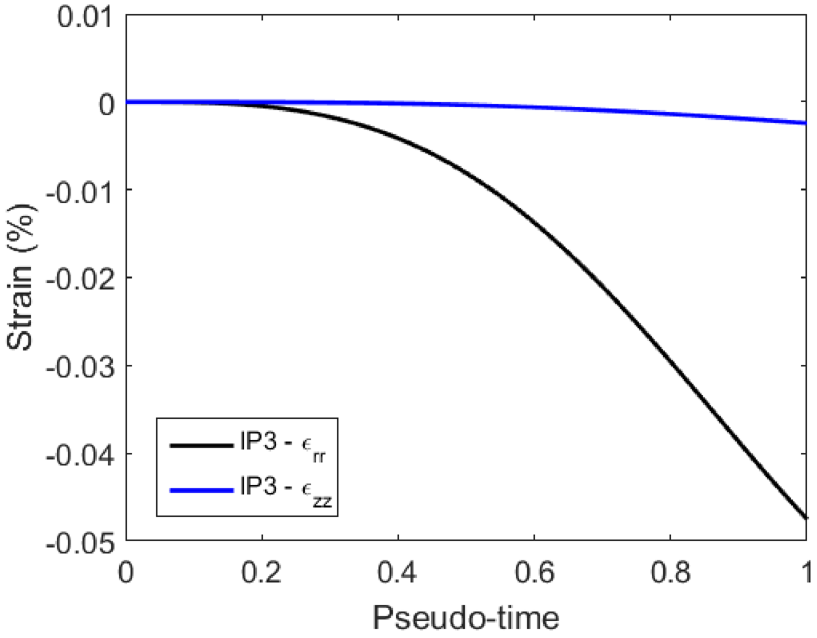


Figure 6.6: Isotropic Compression Step: FEM Strain Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

radial directions were approximately the value of the applied cell pressure (σ_{cell}). The stress in the radial direction was a little higher, likely due to the aforementioned gravity deposition. Although the results were expected to be highly non-linear, it is difficult to determine if the shape of the stress and strain curves is the result of non-linearity of the stress-strain response in the DEM RVEs or an artifact of over-damping the FEM dynamic relaxation algorithm. It was expected that small waves would be visible in the results and, as can be seen on figures 6.5, 6.6, and 6.7, the results are non-linear but smooth. With an over-damped dynamic relaxation algorithm the unphysically large inertia of the problem domain can smooth the results by causing large amplitude and wavelength waves that are uncaptured by the length of the simulation. This issue is mitigated by tuning parameters that are large enough for stability through significantly raising the critical timestep, but small enough to avoid such waves. Considering the approximate accuracy of the stress results relative to the applied cell pressure, it is believed that the presented simulations are, at least slightly, over-damped. As previously stated however, the goal of the isotropic compression step was to precondition the shearing step with a displacement solution that roughly generated an isotropic stress state in the RVEs prior to shearing. The stresses generated during shearing were two orders of magnitude larger and no difference was noted when using less accurate isotropic compression step results than the ones presented. Because of this, the results were considered accurate enough.

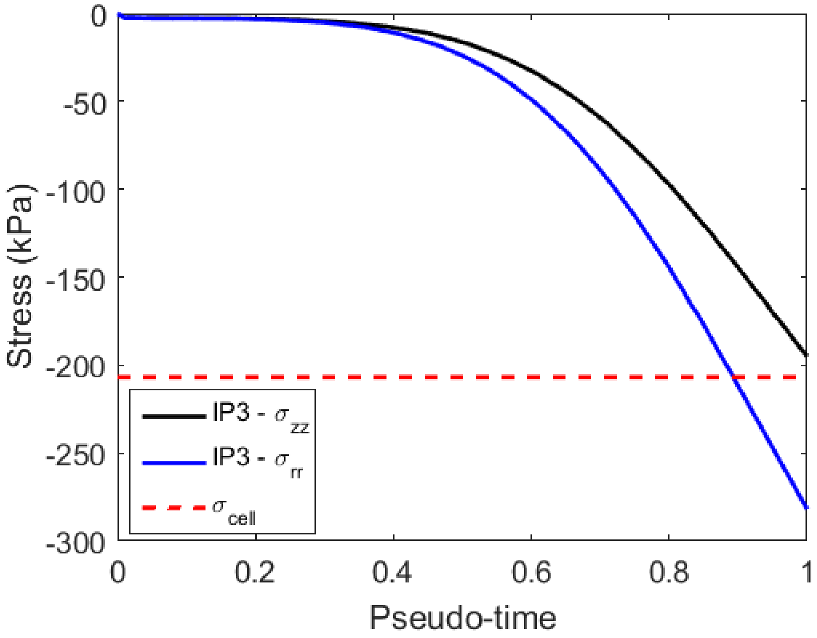


Figure 6.7: Isotropic Compression Step: FEM Stress Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

6.4.2 Shearing Step

After the completion of the isotropic compression step, the displacement results (figure 6.5) were then used to drive the first pseudo-second of deformation during the shearing simulation. As shown by the results of the isotropic compression simulation, this seeding is necessary to insure that the DEM assemblies are nearly isotropic before shearing. Then, displacement was applied to the top of the FEM geometry to simulate the shearing of the soil specimen. The displacement of the two side nodes outward (d_1 and d_2) can be found on figure 6.8. The displacement of the nodes over the first pseudo-second are identical to the displacements on figure 6.5 although the scale required to

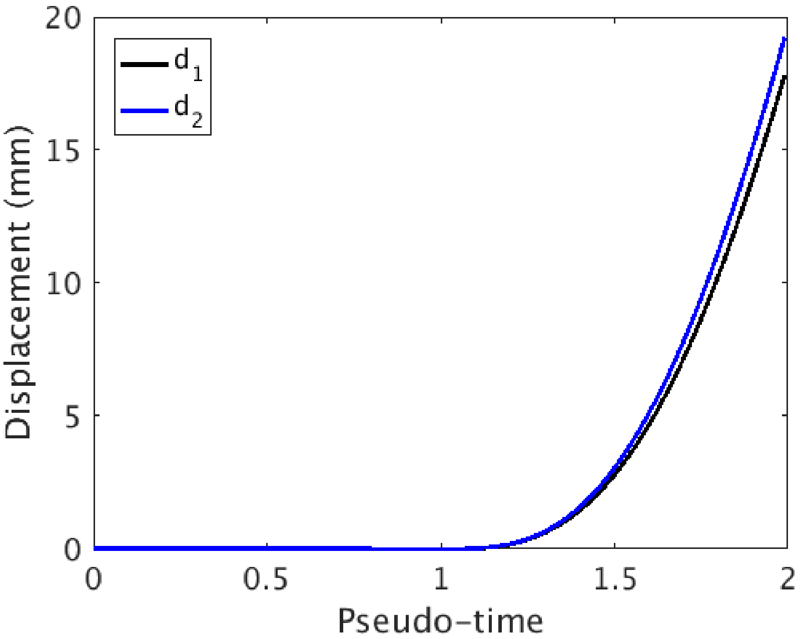


Figure 6.8: Shearing Step: FEM Displacement Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

display the shearing results is much larger making the isotropic compression displacements appear to be zero. As expected, the side of the specimen moves outward during the shearing displacement applied to the top of the specimen. Also, the strain results (figure 6.9) are reasonable in that the radial (ϵ_{rr}) and circumferential ($\epsilon_{\theta\theta}$) are identical and outward, while the axial strain (ϵ_{zz}) reflects the applied strain. As with the isotropic compression step, it's difficult to confirm that the non-linearity is the result of the DEM assembly, but the same exhaustive tuning was done to the shearing simulation to minimize the possibility of the dynamic relaxation affecting the results.

Normal and shear stress results for the shearing step can be found on figures 6.10 and 6.11. The radial (σ_{rr}) and circumferential ($\sigma_{\theta\theta}$) Cauchy stress are similar and less than the axial (σ_{zz}) Cauchy stress, as expected given the applied displacement. As can be seen, the results show a significant softening at approximately 10% axial strain (roughly 1.6 pseudo-seconds). The shear stress (σ_{rz}), although an order of magnitude smaller, also shows a decrease at about 1.6 pseudo-seconds. The decrease in shear strength in soils can be from a loss of frictional resistance between

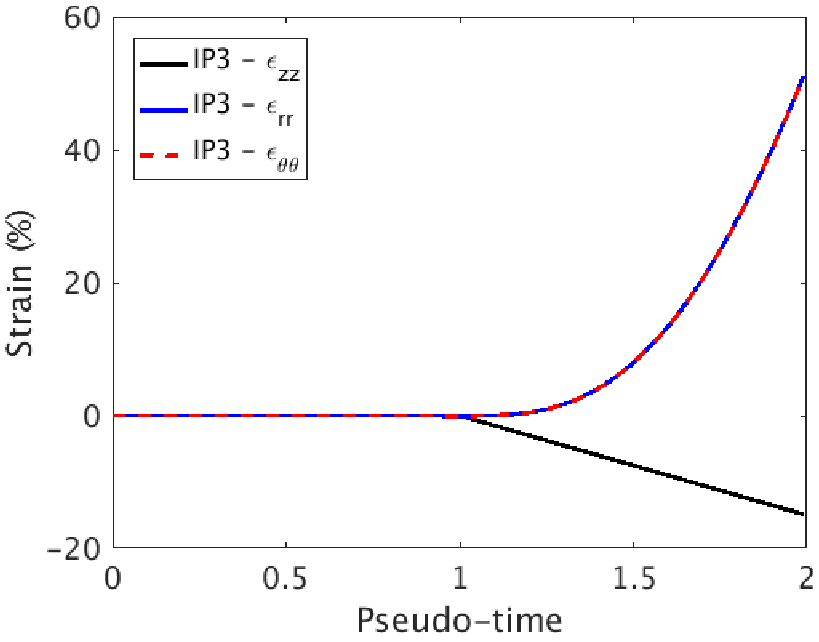


Figure 6.9: Shearing Step: FEM Strain Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

soil grains or soil grains rolling past one another. The decrease of shear stress in the DEM RVE means that the tangential force between DEM particles decreased in the simulation, which could result from both of the same mechanisms. Figure 6.12 shows the original DEM assembly and the assembly deformed at the end of the simulation. Although no shear band has formed, some of the particles have rolled past others as the radial and circumferential walls moved outward.

Finally, the deviatoric stress versus axial strain plot for both the simulation and experimental results can be found on figure 6.13. Although the DEM RVEs are clearly stiffer than Colorado Mason Sand, the results are nonetheless encouraging particularly considering this work is, to the author's knowledge, the first example of an FEM-DEM hierarchical multiscale model being directly compared to experimental data. Qualitatively, as with previous research, the results are reasonable, but it is clear that work needs to be done to match the experimental results (see section 6.6). Particularly, the effect refining the mesh has on the simulation results should be investigated. Other researchers, such as Nguyen et al. (2014), have seen mesh dependence in FEM-DEM models.

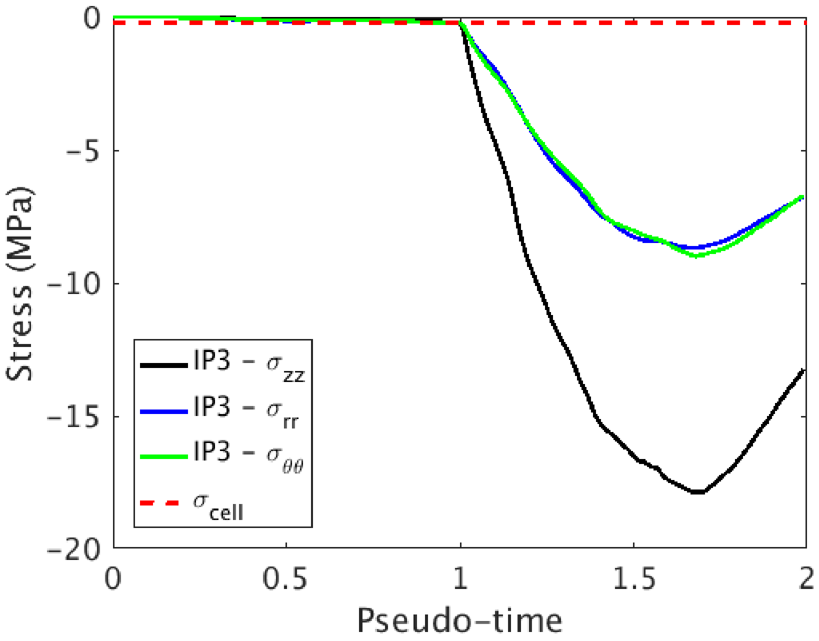


Figure 6.10: Shearing Step: FEM Stress Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test

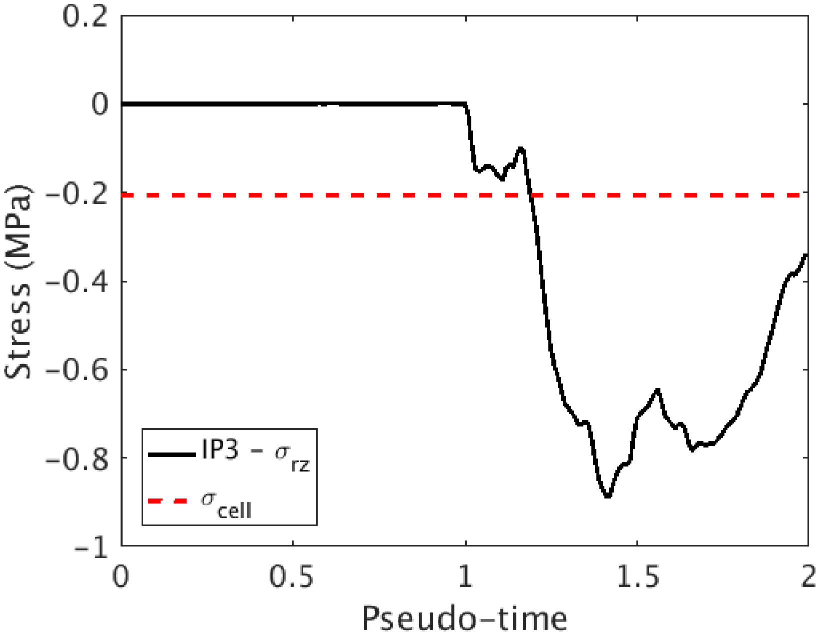
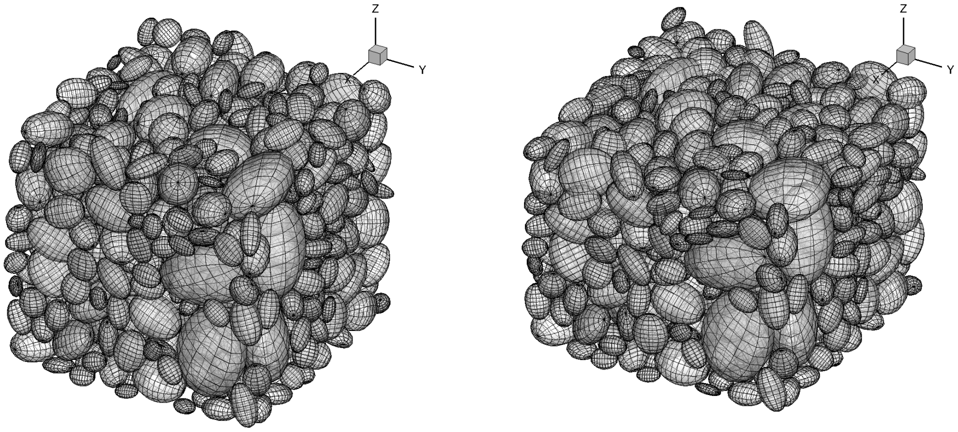


Figure 6.11: Shearing Step: FEM Shear Stress Time History for the Explicit Formulation Simulation of the Standard Triaxial Compression Test



(a) $t = 0$ ms

(b) $t = 2.0$

Figure 6.12: DEM Assembly: Before and After the Simulation

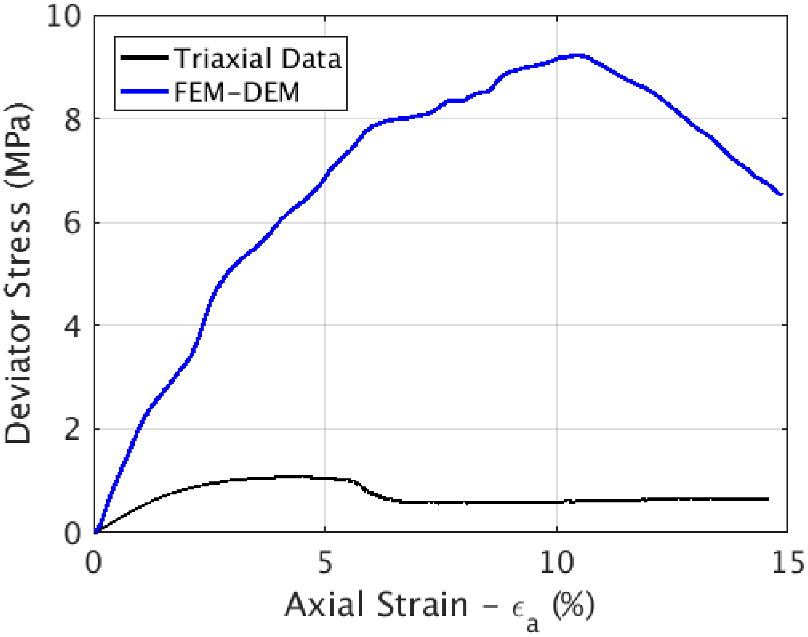


Figure 6.13: Comparison Between Hierarchical Upscaling Simulation and Standard Triaxial Compression Experimental Results

6.5 Small Scale Triaxial Results

Using the same model as the standard scale triaxial simulations, the specimen geometry and applied displacement were changed to model the small-scale triaxial experiments. It should be noted that the diameter to height ratio stayed approximately constant from the standard scale to small-scale experiments and simulations. As expected, the general trends found in the standard scale triaxial results were found in the small-scale triaxial results. Displacement, strain, normal stress, and shear stress time histories can be found on 6.14, 6.15, 6.16, and 6.17, respectively. A comparison of the DEM RVE located at integration point one in the single FEM element before and after the triaxial simulation can be found on figure 6.18, and the stress-strain relationship compared to the experimental data on can be found on figure 6.19.

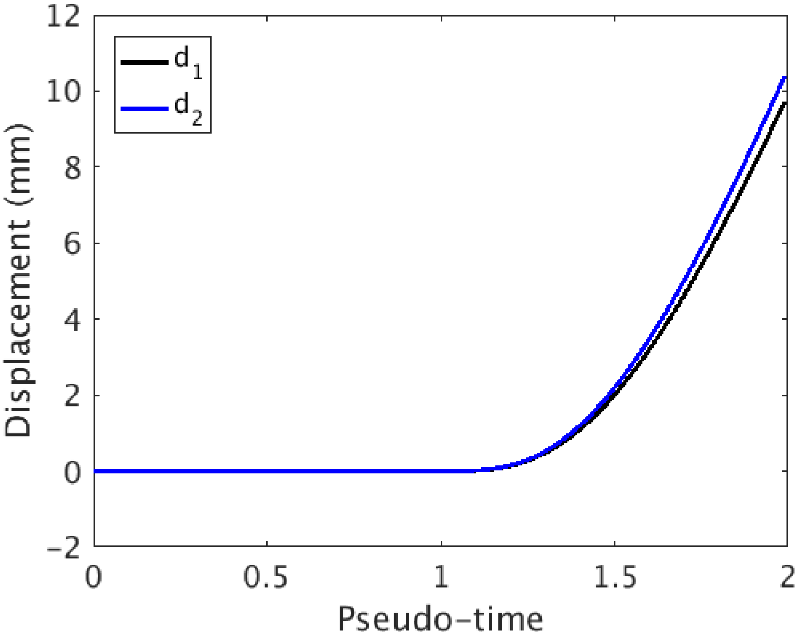


Figure 6.14: Shearing Step: FEM Displacement Time History for the Explicit Formulation Simulation of the Small-scale Triaxial Compression Test

As discussed in section 3.4.2, the primary difference between the standard and small-scale triaxial test experimental results is that there is clear evidence that the stress-strain response of Colorado Mason Sand is pressure-sensitive. The lower the applied cell pressure (confining stress), the lower the deviatoric stress given the applied axial strain. Although the small-scale FEM-DEM simulation deviatoric stress results are lower than the standard scale results, they're only lower by approximately 2MPa at the maximum. The small-scale experimental results are lower than the standard scale results by nearly a factor of ten. This does show, however, that the FEM-DEM model is inherently, at least slightly, pressure-sensitive. Modeling the same problems with ABAQUS using the Arena VUMAT required manually reducing the bulk modulus of the material. No changes were made to the DEM RVEs between the standard and small-scale simulations and the applied axial strain was the same. The only difference between the two models was the applied cell pressure. As with the standard scale results, more work is needed.

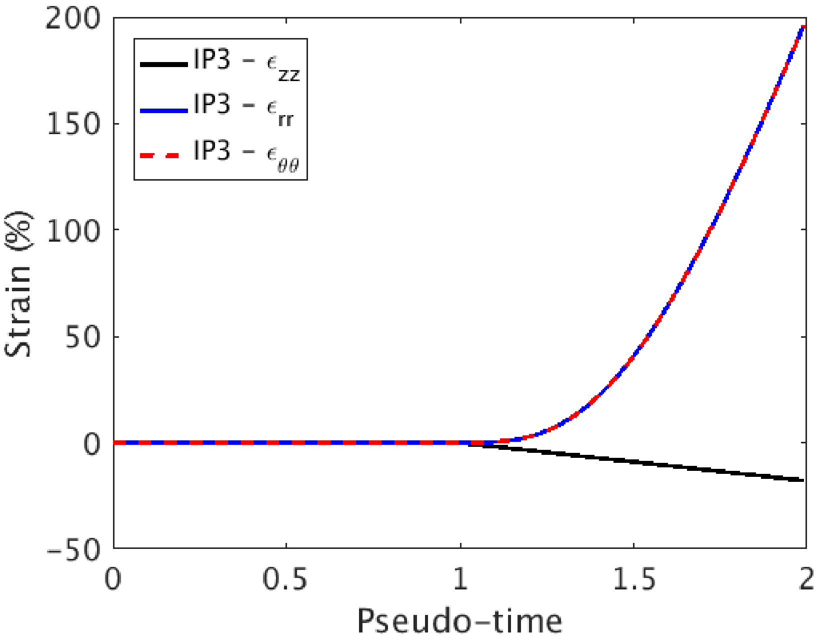


Figure 6.15: Shearing Step: FEM Strain Time History for the Explicit Formulation Simulation of the Small-scale Triaxial Compression Test

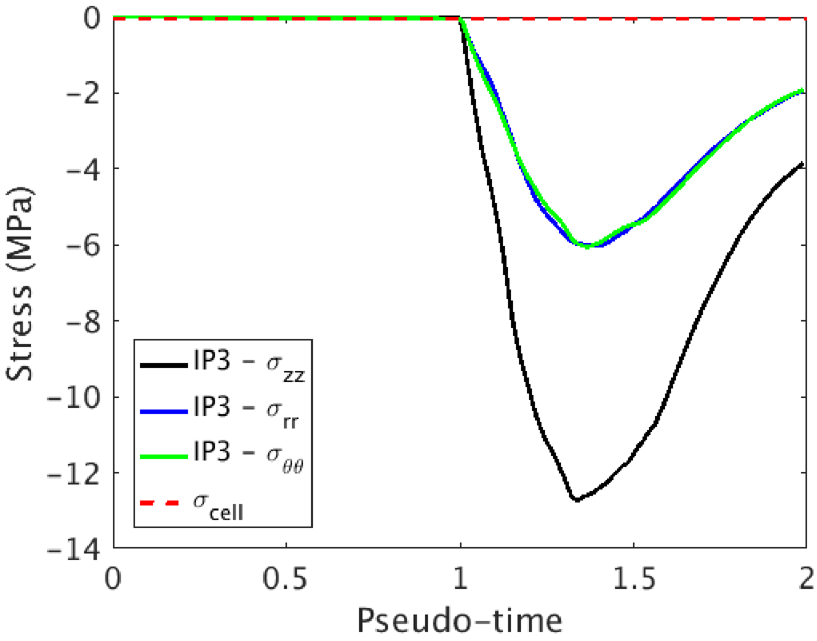


Figure 6.16: Shearing Step: FEM Stress Time History for the Explicit Formulation Simulation of the Small-scale Triaxial Compression Test

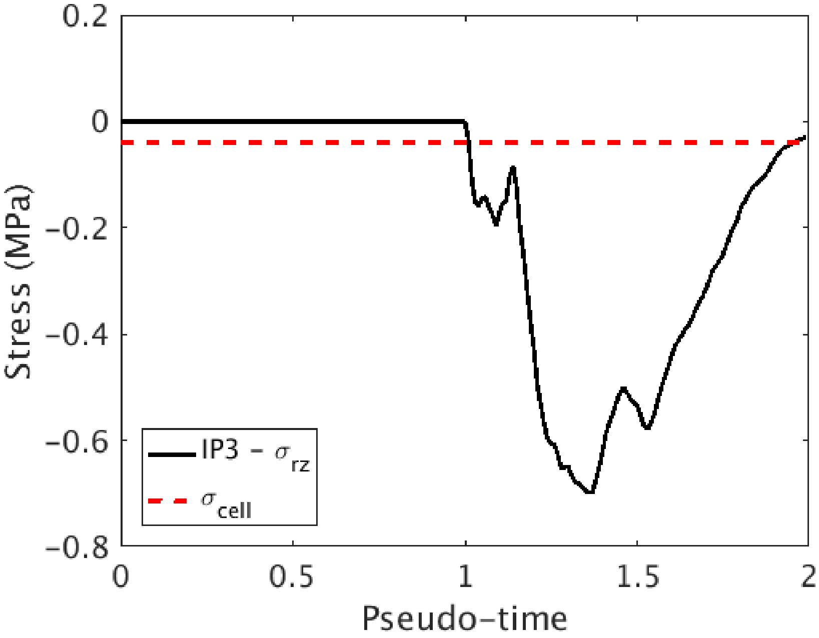
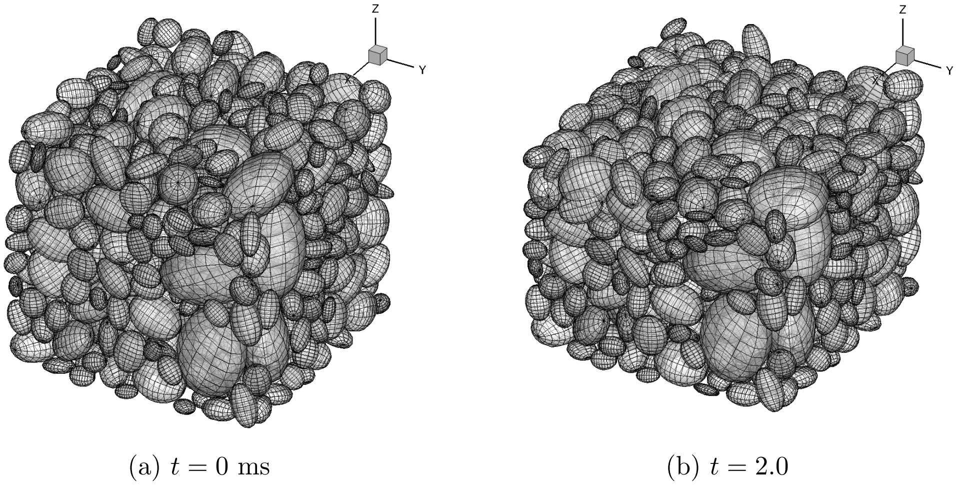


Figure 6.17: Shearing Step: FEM Shear Stress Time History for the Explicit Formulation Simulation of the Small-scale Triaxial Compression Test



(a) $t = 0$ ms (b) $t = 2.0$

Figure 6.18: DEM Assembly: Before and After the Simulation

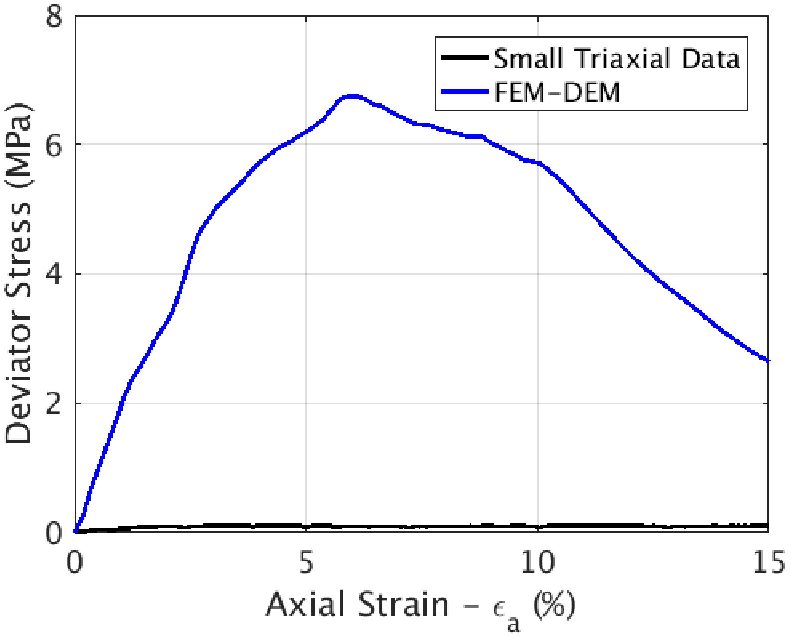


Figure 6.19: Comparison Between Hierarchical Upscaling Simulation and Small-scale Triaxial Compression Experimental Results

6.6 Model Limitations

As currently implemented, the model has three main limitations. As mentioned in section 6.2, the shear strain from the FEM (ϵ_{rz}) should be applied to the wall boundaries of the DEM RVEs. With the $+z$ and $-z$ walls of the RVE able to displace laterally relatively to one another, a further reduction in the deviatoric stress response is expected, which will more closely match the experimental data for both scale triaxial experiments. However, considering that the stress induced by the macroscale deformation was primarily in the normal directions, it is expected that adding this capability will not effect the results too significantly. Second, the FEM implementation is currently unable to capture the macroscale shear band observed experimentally. Even with further mesh refinement, the development of large deviatoric stress at the onset of strain localization at the macroscale will cause mesh instability without further enhancements to the FEM implementation. Unfortunately, this phenomenon can not be currently studied further as the computational expense of the current serial implementation of the model is prohibitively high. In order to further refine the mesh, a parallel version of the code will need to be developed. Finally, using an axisymmetric formulation for the macroscale FEM prevents the FEM from also limiting the computational speed of the overall FEM-DEM model as it might in a full three dimensional implementation. However, one needs to be careful interpreting the results. For example, if the FEM model was updated to accommodate macroscale shear banding and one formed diagonally across the mesh, the code is actually modeling a “cone” shape as the band is forming at all degrees in the θ direction.

Chapter 7

Conclusions and Future Work

Traditional finite and discrete element methods have been used extensively and successfully for the modeling of granular materials, but both methods have faults. Continuum constitutive models implemented in an FEM framework struggle to capture strain localizations. Statistical models to capture aleatory uncertainty (Strack et al., 2015) and finite element models with built-in strain localization enhancements can be implemented to model the post-peak response of materials, but can be difficult to calibrate and have convergence and mesh issues for high strain rate problems. Direct numerical simulation methods such as DEM more accurately represent the material, but suffer from intractable computational times for even small scale problems. Although more calibration and mesh refinement needs to be done, this thesis shows the ability to model highly nonlinear and strain rate dependent experimental data from “natural” soils using hierarchical multiscale modeling.

Without a traditional phenomenological continuum constitutive model, the SHPB simulation was able to capture the small strain stress results of the experiment, and it is expected that with additional calibration and improvements to the particle fracture model, it will be able to capture the large deformations as well. No other attempts have been made by researchers to fully include the kinematics of finite strain dynamics, to use more accurate three dimensional RVEs, and, most importantly, to compare FEM-DEM models to actual experimental data. Also, in order to begin developing the novel link to the MPM code Uintah, a one dimensional MPM version of the SHPB FEM-DEM simulation was also be developed.

The method was extended to a two dimensional, axisymmetric version in order to model triaxial compression tests. Arena, as currently implemented in ABAQUS without a shear softening model, has difficulty capturing the behavior of the triaxial compression data, and given the success other researchers have had capturing the post-peak response using FEM-DEM models under biaxial compression, it was hypothesized that an axisymmetric version of this FEM-DEM model would also produce more "predictive" results. Similarly to the one dimensional versions of the models, mesh refinement and model improvements need to be done to quantitatively replicate the experimental data, but the initial results are qualitatively accurate.

There are a number of studies and small improvements to the presented hierarchical multiscale models that would make the models more predictive. First, the ability of the particles to "escape" the DEM RVEs if the wall boundary conditions pass over the particle centroid needs to be resolved. The reduction in the number of interparticle contacts when this occurs unphysically decreases the stress results. Second, the particle fracture model needs to be further calibrated with data from crushing experiments on Colorado Mason Sand. Third, a more modern MPM formulation should be implemented to alleviate the issues caused by material points moving from one background element to another. Finally, the ability to apply the macroscale shear strain to the wall boundary conditions in the two dimensional, axisymmetric FEM-DEM model needs to be added.

In the longer term, there are a number of larger improvements and additions to the hierarchical multiscale models. Using the same general MPI parallelization scheme as the one dimensional FEM-DEM model implementation (i.e. send each integration point to its own computational node to run the DEM simulations in parallel), the 2D axisymmetric FEM-DEM model needs to be parallelized and updated to run on Department of Defense supercomputers. The implementation currently runs serially, and in order to refine the mesh, the code will need to run in parallel to computationally tractable. Also, as the DEM component of the hierarchical upscaling models essentially replaces a continuum constitutive model, subroutines for ABAQUS and Uintah should be written to natively incorporate the FEM-DEM and MPM-DEM models into their frameworks. This will first require creating a general DEM RVE that can deform given the full, three dimensional strain tensor, and then writing a UMAT for ABAQUS or constitutive model for Uintah that applies the strain tensor to the DEM RVEs located at each integration or material point. The advantage would be that one could use preexisting FEM and MPM implementations (including parallelization) in three dimensions rather than independently developing the macroscale codes. Finally, other possible additions to the current models include but are not limited to a full finite strain poromechanical FEM implementation with a coupled DEM-SPH grain-scale model to study partially and fully saturated Colorado Mason Sand and using a coupled DEM-PD grain-scale model to study Colorado Mason Sand and Boulder Clay mixes.

The FEM-DEM framework could also be extended to modeling soil disaggregation. As mentioned in chapter 3, Arena has a soil disaggregation model implemented to attempt to model the geotechnical centrifuge blast experiments conducted for the Soil Blast MURI Project. However, the soil disaggregation model is empirical, and it's difficult to calibrate as the model parameters do not have obvious physical meaning. Using Taylor-Voigt boundary conditions and the newly implemented FEM-DEM model in Uintah, it's possible to model soil disaggregation under explosive loading with either fully replacing Arena in the region around the explosive with DEM RVEs or using a similar methodology as Andrade and Tu (2009) to directly inform Arena's soil disaggregation model. Also, these simulations could be compared directly to fully coupled DEM-CFD simulations of the same problem, or the DEM-CFD simulations could be used in a unit cell sense to inform the soil disaggregation model similar to work by Andrade et al. (2011).

Bibliography

José E. Andrade and Xuxin Tu. Multiscale Framework for Behavior Prediction in Granular Media. Mechanics of Materials, 41(6):652–669, 2009.

Jose E. Andrade, C. F. Avila, S. A. Hall, N. Lenoir, and G. Viggiani. Multiscale Modeling and Characterization of Granular Matter: From Grain Kinematics to Continuum Mechanics. Journal of the Mechanics and Physics of Solids, 59(2):237–250, 2011.

ASTM-D2487. Standard Practice for Classification of Soils for Engineering Purposes (Unified Soil Classification System): Active Standard ASTM D2487. In Book of Standards Volume: 04.08, pages 395–408. American Socitey for Testing and Materials, 1985.

Katalin Bagi. Stress and Strain in Granular Assemblies. Mechanics of Materials, 22(3):165–177, 1996.

G. Bao, J. W. Hutchinson, and R. M. McMeeking. Particle reinforcement of Ductile Matrices Against Plastic Flow and Creep. Acta Metallurgica et Materialia, 39(8):1871–1882, 1991.

S. G. Bardenhagen and E. M. Kober. The Generalized Interpolation Material Point Method. Computer Modeling in Engineering and Sciences, 5(6):477–495, 2004.

Alain Bensoussan, Jacques-Louis Lions, and George Papanicolaou. Asymptotic Analysis for Periodic Structures. North-Holland, Amsterdam, 1978.

Ronaldo I. Borja and Jon R. Wren. Micromechanics of Granular Media Part I: Generation of Overall Constitutive Equation for Assemblies of Circular Disks. Computer Methods in Applied Mechanics and Engineering, 127(1-4):13–36, 1995.

Rebecca M. Brannon, Arlo F. Fossum, and O. Erik Strack. KAYENTA: Theory and User's Guide. Technical Report March, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2009.

Rebecca M. Brannon, Michael Homel, Emad Ghodrati, David Ausin, and Jim E. Guilkey. Arena: Theory and User's Guide. Technical report, Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, 2014.

J. R. Brockenbrough, S. Suresh, and H. A. Wienecke. Deformations of Metal-Matrix Composites with Continuous Fibers: Geometrical Effects of Fiber Distribution and Shape. Acta Metallurgica et Materialia, 39(5):735–752, 1991.

Qiushi Chen, José E. Andrade, and Esteban Samaniego. AES for multiscale localization modeling in granular media. Computer Methods in Applied Mechanics and Engineering, 200(33-36):2473–2482, 2011.

T. Christman, A. Needleman, and S. Suresh. An Experimental and Numerical Study of Deformation in Metal-Ceramic Composites. Acta Metallurgica et Materialia, 37(11):3029–3050, 1989.

J. Christoffersen, M. M. Mehrabadi, and Sia Nemat-Nasser. A Micromechanical Description of Granular Material Behavior. Journal of Applied Mechanics, 48(June 1981):339, 1981.

Crystal Rae Clendennen and Pedro Romero. Evaluating the Representative Volume Element of Asphalt Concrete Mixture Beams for Testing in the Bending Beam Rheometer. In Niki Kringos, Bjorn Birgisson, David Frost, and Linbing Wang, editors, Multi-Scale Modeling and Characterization of Infrastructure Materials: Proceedings of the International RILEM Symposium, Stockholm, Sweden, 2013.

P. A. Cundall and O. D. L. Strack. A Discrete Numerical Model for Granular Assemblies. Geotechnique, 29(1):47–65, 1979.

G. A. D'Addetta, E. Ramm, S. Diebels, and W. Ehlers. A Particle Center Based Homogenization Strategy for Granular Assemblies. Engineering Computations, 21(2/3/4):360–383, 2004.

Yannis F Dafalias and Majid T Manzari. Simple Plasticity Model Accounting for Fabric Change Effects. Journal of Engineering Mechanics, 130(6):622–634, 2004.

F. DeVries, H. Dumontet, G. Duvaut, and F. Lene. Homogenization and Damage of Composite Structures. International Journal for Numerical Methods in Engineering, 27:285–298, 1989.

Andrew Druckrey and Khalid Alshibli. 3D Behavior of Sand Particles Using X-Ray Synchrotron Micro-Tomography. In Geo-Congress 2014 Technical Papers, pages 2814–2821, Atlanta, GA, 2014.

W.J. Drugan and J.R. Willis. A Micromechanics-based Nonlocal Constitutive Equation and Estimates of Representative Volume Element Size for Elastic Composites. Journal of the Mechanics and Physics of Solids, 44(4):497–524, 1996.

W. Ehlers, E. Ramm, S. Diebels, and G. A. D'Addetta. From Particle Ensembles to Cosserat Continua: Homogenization of Contact Forces Towards Stresses and Couple Stresses. International Journal of Solids and Structures, 40(24):6681–6702, 2003.

Frédéric Feyel and Jean-Louis Chaboche. FE2 Multiscale Approach for Modelling the Elastoviscoplastic Behaviour of Long Fibre SiC/Ti Composite Materials. Computer Methods in Applied Mechanics and Engineering, 183(3-4):309–330, 2000.

Jacob Fish, Qing Yu, and Kamlun Shek. Computational Damage Mechanics for Composite Materials Based on Mathematical Homogenization. International Journal for Numerical Methods in Engineering, 45:1657–1679, 1999.

R. Fletcher. Practical Methods of Optimization. John Wiley & Sons, New York, NY, second edi edition, 1987.

Arlo F. Fossum and Rebecca M. Brannon. The Sandia Geomodel: Theory and User's Guide. Technical report, Sandia National Laboratories, 2004.

C. D. Foster, Richard Regueiro, Arlo F. Fossum, and Ronaldo I. Borja. Implicit Numerical Integration of a Three-invariant, Isotropic/Kinematic Hardening Cap Plasticity Model for Geomaterials. Computer Methods in Applied Mechanics and Engineering, 194(50-52):5109–5138, 2005.

M. G. D. Geers, V. G. Kouznetsova, and W. A. M. Brekelmans. Multi-scale Computational Homogenization: Trends and Challenges. Journal of Computational and Applied Mathematics, 234 (7):2175–2182, aug 2010.

Somnath Ghosh, Kyunghoon Lee, and Suresh Moorthy. Multiple Scale Analysis of Heterogeneous Elastic Structures using Homogenization Theory and Voronoi Cell Finite Element Method. International Journal of Solids and Structures, 32(1):27–62, 1995.

Somnath Ghosh, Kyunghoon Lee, and Suresh Moorthy. Two Scale Analysis of Heterogeneous Elastic-plastic Materials with Asymptotic Homogenization and Voronoi Cell Finite Element Model. Computer Methods in Applied Mechanics and Engineering, 132(1-2):63–116, 1996.

Somnath Ghosh, Kyunghoon Lee, and Prasanna Raghavan. A Multi-Level Computational Model for Multi-scale Damage Analysis in Composite and Porous Materials. International Journal of Solids and Structures, 38(14):2335–2385, 2001.

Jose Miranda Guedes and Noboru Kikuchi. Preprocessing and Postprocessing for Materials Based on the Homogenization Method with Adaptive Finite Element Methods. Computer Methods in Applied Mechanics and Engineering, 83:143–198, 1990.

Ning Guo and Jidong Zhao. A Coupled FEM/DEM Approach for Hierarchical Multiscale Modelling of Granular Media. International Journal for Numerical Methods in Engineering, pages 1885–1891, 2014.

Heinrich Hertz. Ueber die Beruhrung Fester Elastischer Korper (On the Fixed Elastic Body Contact). Journal fur die Reine unt Angewandte Mathematik, 92:156–171, 1882.

R. Hill. Elastic Properties of Reinforced Solids: Some Theoretical Principles. Journal of the Mechanics and Physics of Solids, 11(357-372), 1963.

E. Hoek and C. D. Martin. Fracture Initiation and Propagation in Intact Rock - A Review. Journal of Rock Mechanics and Geotechnical Engineering, 6:287–300, 2014.

S. J. Hollister and Noboru Kikuchi. A Comparison of Homogenization and Standard Mechanics Analyses for Periodic Porous Composites. Computational Mechanics, 10:73–95, 1992.

Thomas J. R. Hughes. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Prentice-Hall, Englewood Cliffs, NJ, 1987.

F. Huq, L. Graham-Brady, and Rebecca M. Brannon. Efficient Binning of Random Samples Applied to a Micromechanics Damage Model for Brittle Materials. In Engineering Mechanics Institute Conference 2015, Stanford University, Palo Alto, CA, 2015.

K. Kaneko, Kenjiro Terada, Takashi Kyoya, and Y. Kishino. Global-Local Analysis of Granular Media in Quasi-Static Equilibrium. International Journal of Solids and Structures, 40(15):4043–4069, 2003.

Yevgeniy Kaufman. The Influence of Representative Volume Element Size, Soil Fabric, and Interparticle Elasto-Plasticity in Three-dimensional Ellipsoidal Discrete Element Modeling of Granular Assemblies. M.s. thesis, University of Colorado Boulder, 2012.

C. T. Kelley. Iterative Methods for Linear and Nonlinear Equations. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995.

V. G. Kouznetsova. Computational Homogenization for the Multi-scale Analysis of Multi-phase Materials. Ph.d. thesis, Eindhoven University of Technology, 2002.

Xikui Li, Qipeng Liu, and Junbo Zhang. A Micro-Macro Homogenization Approach for Discrete Particle Assembly - Cosserat Continuum Modeling of Granular Materials. International Journal of Solids and Structures, 47(2):291–303, 2010a.

Xikui Li, Xue Zhang, and Junbo Zhang. A Generalized Hill's Lemma and Micromechanically Based Macroscopic Constitutive Model for Heterogeneous Granular Materials. Computer Methods in Applied Mechanics and Engineering, 199(49-52):3137–3152, 2010b.

Yang Liu, WaiChing Sun, Zifeng Yuan, and Jacob Fish. A Nonlocal Multiscale Discrete-Continuum Model for Predicting Mechanical Behavior of Granular Materials. International Journal for Numerical Methods in Engineering, 106:129–160, 2016.

Huiyang Luo, Hongbing Lu, William L. Cooper, and R. Komanduri. Effect of Mass Density on the Compressive Behavior of Dry Sand Under Confinement at High Strain Rates. Experimental Mechanics, 51(9):1499–1510, feb 2011.

H. A. Meier, P. Steinmann, and E. Kuhl. Towards multiscale computation of confined granular media-Contact forces, stresses and tangent operators. Technische Mechanik, 16(1):77–88, 2008.

H. A. Meier, P. Steinmann, and E. Kuhl. On the Multiscale Computation of Confined Granular Media. In J. Eberhardsteiner, editor, ECCOMAS Multidisciplinary Jubilee Symposium, Computational Methods in Applied Sciences, pages 121–133. Springer Science + Business Media, 2009.

Christian Miehe and Joachim Dettmar. A Framework for Micro-Macro Transitions in Periodic Particle Aggregates of Granular Materials. Computer Methods in Applied Mechanics and Engineering, 193(3-5):225–256, 2004.

Christian Miehe, Jan Schotte, and Jörg Schröder. Computational MicroMacro Transitions and Overall Moduli in the Analysis of Polycrystals at Large Strains. Computational Materials Science, 16(1-4):372–382, 1999a.

Christian Miehe, Jörg Schröder, and Jan Schotte. Computational Homogenization Analysis in Finite Plasticity Simulation of Texture Development in Polycrystalline Materials. Computer Methods in Applied Mechanics and Engineering, 171(3-4):387–418, 1999b. ISSN 00457825.

Raymond D. Mindlin. Compliance of Elastic Bodies in Contact. Transactions of the ASME, Journal of Applied Mechanics, 16(3):259–268, 1949.

Woongju Mun. Compression Mechanisms of Soils under High Stresses. Ph.d. dissertation, University of Colorado Boulder, 2015.

Woongju Mun and John McCartney. Compression Mechanisms of Unsaturated Clay under High Stresses. Canadian Geotechnical Journal, Accepted:1–53, 2015.

T. Nakamura and S. Suresh. Effects of Thermal Residual Stresses and Fiber Packing on Deformation of Metal-Matrix Composites. Acta Metallurgica et Materialia, 41(6):1665–1681, 1993.

Sia Nemat-Nasser and Muneo Hori. Micromechanics: Overall Properties of Heterogeneous Materials. Elsevier, Amsterdam, 1993.

Trung Kien Nguyen, Gael Combe, Denis Caillerie, and Jacques Desrues. FEM x DEM Modelling of Cohesive Granular Materials : Numerical Homogenisation and Multi-scale Simulation. Acta Geophysica, 62(5):1109–1126, 2014.

Vinh Phu Nguyen. Material Point Method: Basics and Applications. Technical report, Monash University, Australia, 2014.

Michał Nitka, Gaël Combe, Cristian Dascalu, and Jacques Desrues. Two-Scale Modeling of Granular Materials: A DEM-FEM Approach. Granular Matter, 13(3):277–281, 2011.

E. Onate and J. Rojek. Combination of Discrete Element and Finite Element Methods for Dynamic Analysis of Geomechanics Problems. Computer Methods in Applied Mechanics and Engineering, 193(27-29):3087–3128, 2004.

John F. Peters, Mark A. Hopkins, Raju Kala, and Ronald E. Whal. A Polyellipsoid Particle for Nonspherical Discrete Element Method. International Journal for Computer-Aided Engineering and Software, 26(6):645–657, 2009.

A. Sadeghirad, Rebecca M. Brannon, and J. Burghardt. A Convected Particle Domain Interpolation Technique to Extend Applicability of the Material Point Method for Problems Involving Massive Deformations. International Journal for Numerical Methods in Engineering, 2011.

E. Sanchez-Palencia. Non-homogeneous Media and Vibration Theory. Springer-Verlag, Berlin, 1980.

R. J. M. Smit, W. A. M. Brekelmans, and H. E. H. Meijer. Prediction of the Mechanical Behavior of Nonlinear Heterogeneous Systems by Multi-Level Finite Element Modeling. Computer Methods in Applied Mechanics and Engineering, 155(97):181–192, 1998.

O. Erik Strack, R. B. Leavy, and Rebecca M. Brannon. Aleatory Uncertainty and Scale Effects in Computational Damage Models for Failure and Fragmentation. International Journal for Numerical Methods in Engineering, 102:468–495, 2015.

J Stránský and M Jirásek. Open Source Fem-Dem Coupling. In 18th International Conference Engineering Mechanics, pages 1237–1251, Svratka, Czech Republic, 2012.

Jan Stránský and Milan Jirásek. Calibration of Particle-Based Models Using Cells with Periodic Boundary Conditions. In E. Onate and D.R.J Owen, editors, International Conference on Particle-based Methods - Fundamentals and Applications, 2011.

Deborah Sulsky, Zhen Chen, and Howard L. Schreyer. A Particle Method for History-Dependent Materials. Computer Methods in Applied Mechanics and Engineering, 118:179–196, 1994.

Deborah Sulsky, Shi Jian Zhou, and Howard L. Schreyer. Application of a Particle-in-Cell Method to Solid Mechanics. Computer Physics Communications, 87:236–252, 1995.

Pierre M. Suquet. Local and Global Aspects in the Mathematical Theory of Plasticity. Plasticity Today: Modelling, Methods and Applications, pages 279–310, 1985.

Jenna Svoboda. Impact of Strain Rate on the Shear Strength and Pore Water Pressure Generation of Clays and Sands. M.s. thesis, University of Colorado Boulder, 2013.

Lee M. Taylor and Dale S. Preece. Simulation of Blasting Induced Rock Motion Using Spherical Element Models. Engineering Computations, 9(2):243–252, 1992.

Kenjiro Terada and Noboru Kikuchi. A Class of General Algorithms for Multi-scale Analyses of Heterogeneous Media. Computer Methods in Applied Mechanics and Engineering, 190(40-41): 5427–5464, jul 2001.

S. P. Timoshenko. Theory of Elasicity. McGraw-Hill, 1970.

A. Toledano and H. Murakami. A High-Order Mixture for Periodic Particulate Composites. International Journal of Solids and Structures, 23(7):989–1102, 1987.

V. Tvergaard. Analysis of Tensile Properties for a Whisker-Reinforced Metal-Matrix Composite. Acta Metallurgica et Materialia, 38(2):185–194, 1990.

P. Underwood. Dynamic Relaxation. In T. Belytschko and T. J. R. Hughes, editors, Computational Methods for Transient Analysis, pages 245–265. Elsevier, Amsterdam, 1983.

Olaf van der Sluis. Homogenisation of Structured Elastoviscoplastic Solids. Ph.d. dissertation, TU Eindhoven, Eindhoven, The Netherlands, 2001.

Olaf van der Sluis, P. J. G. Schreurs, and H. E. H. Meijer. Effective Properties of a Viscoplastic Constitutive Model Obtained by Homogenization. Mechanics of Materials, 31(11):743–759, 1999.

Jon R. Wren and Ronaldo I. Borja. Micromechanics of Granular Media Part II: Overall Tangential Moduli and Localization Model for Periodic Assemblies of Circular Disks. Computer Methods in Applied Mechanics and Engineering, 141(3-4):221–246, 1997.

Beichuan Yan. 3D Discrete Element Modeling of Granular Materials and its Coupling with Finite Element Method. Ph.d. dissertation, University of Colorado Boulder, 2008.

Beichuan Yan, Richard Regueiro, and Stein Sture. Three-Dimensional Ellipsoidal Discrete Element Modeling of Granular Materials and its Coupling with Finite Element Facets. International Journal for Computer-Aided Engineering and Software, 27(4):519–550, 2010.

Boning Zhang and Richard Regueiro. On Large Deformation Granular Strain Measures for Generating Stress-strain Relations Based Upon Three-Dimensional Discrete Element Simulations. International Journal of Solids and Structures, 66:151–170, 2015.

Boning Zhang, E. B. Herbold, and Michael Homel. DEM Particle Fracture Model. Technical report, Lawrence Livermore National Laboratory, Livermore, CA, 2015.

Boning Zhang, Richard Regueiro, Andrew Druckrey, and Khalid Alshibli. Construction of Poly-ellipsoidal Grain Shapes from SMT Imaging on Sand, and the Development of a New DEM Contact Detection Algorithm. Engineering Computations, In Review, 2017.

Appendix A

Arenisca ABAQUS UMAT

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,STRAN,DSTRAN,
     2 TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,MATERL,NDI,NSHR,NTENS,
     3 NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,CELENT,
     4 DFGRD0,DFGRD1,NOEL,NPT,KSLAY,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 MATERL
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),
     4 DFGRD0(3,3),DFGRD1(3,3)
C
      DIMENSION S_DEV(6),DFDSIG(6),DUNIT(6),DCEDFDSIG(6),
     1 DFDSIGCE(6),DELSTRESS(6),DPRODUCT(6,6)
      PARAMETER (ONE=1.0D0,TWO=2.0D0,THREE=3.0D0,SIX=6.0D0)
      DATA SMALL/1.D-12/
```

```
C
C ----------------------------------------------------------------------
C       UMAT FOR ISOTROPIC SMALL STRAIN ELASTO-PLASTICITY
C       ARENISCA PLASTICITY MODEL
C       --LINEAR ISOTROPIC ELASTICITY, ASSOCIATIVE PLASTICITY
C       --NO THIRD INVARIANT DEPENDENCE, NO LIMIT SURFACE
C       GENERAL 3D FORMULATION
C ----------------------------------------------------------------------
C       MATERIAL PROPERTIES:
C       PROPS(1) - K - BULK MODULUS
C       PROPS(2) - G OR MU - SHEAR MODULUS
C       PROPS(3) - BETA - TAN(FRICTION ANGLE)
C       PROPS(4) - I1MAX - TENSILE LIMIT
C       PROPS(5) - CR - ELIPTICITY PARAMETER
C       PROPS(6) - P0 - CRUSH CURVE - PRESSURE AT ELASTIC LIMIT
C       PROPS(7) - P1 - CRUSH CURVE - SLOPE OF PORO VS. PRESS CURVE
C       PROPS(8) - P3 - CRUSH CURVE - MAX PLAST VOL STRAIN
C ----------------------------------------------------------------------
C       INTERNAL STATE VARIABLES:
C       STATEV(1) - KAPPA
C       STATEV(2) - X
C       STATEV(3) - GAMMA (just for output purposes)
C       STATEV(4) - DELGAMMA (just for output purposes)
C       STATEV(5) - EVP (plastic volumetric strain)
C       STATEV(6) - EV (total volumetric strain)
C       STATEV(7) - E11
C       STATEV(8) - E22
```

```
C     STATEV(9) - E33
C     STATEV(10) - E12
C     STATEV(11) - E13
C     STATEV(12) - E23
C ----------------------------------------------------------------------
C     STRESS CONVENTION:
C     STRESS(1) - SIGMA_11
C     STRESS(2) - SIGMA_22
C     STRESS(3) - SIGMA_33
C     STRESS(4) - SIGMA_12
C     STRESS(5) - SIGMA_13
C     STRESS(6) - SIGMA_23
C ----------------------------------------------------------------------
C
C     ELASTIC PARAMETERS
C
      EK=PROPS(1)
      EMU=PROPS(2)
      ELAMB=EK-(TWO*EMU/THREE)
C
C     HARDENING PARAMETERS
C
      DBETA=PROPS(3)
      DI1MAX=PROPS(4)
      DCR=PROPS(5)
      DP0=PROPS(6)
      DP1=PROPS(7)
      DP3=PROPS(8)
C
C     RECOVER ISVS
C
      DKAPPA=STATEV(1)
      DX=(DKAPPA+DI1MAX*(DCR-ONE))/DCR
C
C     ELASTIC TANGENT MODULUS (IN MATRIX FORM)
C     INITIALIZE
      DO 20 K1=1,NTENS
        DO 10 K2=1,NTENS
          DDSDDE(K2,K1)=0.0D0
 10     CONTINUE
 20   CONTINUE
C     ASSIGN PARAMETERS
      DO 40 K1=1,NDI
        DO 30 K2=1,NDI
          DDSDDE(K2,K1)=ELAMB
 30     CONTINUE
        DDSDDE(K1,K1)=TWO*EMU+ELAMB
 40   CONTINUE
      DO 50 K1=NDI+1,NTENS
        DDSDDE(K1,K1)=EMU
 50   CONTINUE
C
C     CALCULATE TRIAL STRESS USING INCREMENTAL TOTAL STRAIN
C
```

```
      DO 70 K1=1,NTENS
        DO 60 K2=1,NTENS
          STRESS(K2)=STRESS(K2)+DDSDDE(K2,K1)*DSTRAN(K1)
 60     CONTINUE
 70   CONTINUE
C
C     CALCULATE MEAN AND DEVIATORIC TRIAL STRESSES (CREATE UNIT TENSOR)
C
      DO 80 K1=1,NTENS
        S_DEV(K1)=STRESS(K1)
        IF (K1.LE.NDI) THEN
          DUNIT(K1)=ONE
        ELSE
          DUNIT(K1)=0.0D0
        ENDIF
 80   CONTINUE
      STRESS_MEAN=(STRESS(1)+STRESS(2)+STRESS(3))/THREE
      S_DEV(1)=S_DEV(1)-STRESS_MEAN
      S_DEV(2)=S_DEV(2)-STRESS_MEAN
      S_DEV(3)=S_DEV(3)-STRESS_MEAN
C
C     CALCULATE TRIAL STRESS INVARIANTS
C
      DI1=STRESS_MEAN*THREE
      D2J2=S_DEV(1)*S_DEV(1)+S_DEV(2)*S_DEV(2)+
     1      S_DEV(3)*S_DEV(3)+TWO*(S_DEV(4)*S_DEV(4)+
     1      S_DEV(5)*S_DEV(5)+S_DEV(6)*S_DEV(6))
```

```
      DJ2=D2J2/TWO
C
C     CALCULATE TRIAL YIELD FUNCTION
C
      IF (DI1.GE.DKAPPA) THEN
         DFC=ONE
      ELSE
         DFC=(DKAPPA-DI1)/(DKAPPA-DX)
         DFC=DFC*DFC
         DFC=ONE-DFC
      ENDIF
      DFF=DBETA*(DI1MAX-DI1)
      IF (DI1.GE.DKAPPA) THEN
         FTRIAL=DJ2-DFF*DFF
      ELSE
         FTRIAL=DJ2-DFF*DFF*DFC
      ENDIF
C
C     CHECK FOR YIELDING
C
      IF (FTRIAL.GE.0.0) THEN
C
C     DFDSIG DERIVATIVES
      IF (DI1.GE.DKAPPA) THEN
         DFCDI1_FC=0.0D0
      ELSE
         DFCDI1_FC=(DKAPPA-DI1)/((DKAPPA-DX)*(DKAPPA-DX))
      ENDIF
      DFFDI1=-DBETA
      IF (DI1.GE.DKAPPA) THEN
          DFDI1=-TWO*DFF*DFFDI1
      ELSE
          DFDI1=-TWO*(DFF*DFFDI1*DFC+DFF*DFF*DFCDI1_FC)
      ENDIF
      DFDJ2=ONE
C     DFDKAPPA DERIVATIVES
      IF (DI1.GE.DKAPPA) THEN
          DFCDKAPPA=0.0D0
          DFDKAPPA=0.0D0
      ELSE
          DFCDKAPPA=-TWO*DFCDI1_FC
          DFCDKAPPA_FC=DFCDKAPPA*
     1                 (ONE+DCR*DBETA*(DKAPPA-DI1)/(DKAPPA-DX))
          DFDKAPPA=-DFF*DFF*DFCDKAPPA_FC
      ENDIF
C     HARDENING FUNCTION DERIVATIVES
      DEPSVPDX=DP1*DP3*EXP(-DP1*(DP0-DX))
      DXDKAPPA=ONE/DCR
      DHQ=THREE*DFDI1/(DEPSVPDX*DXDKAPPA)
C     MATRIX MULTIPLICATION
      DO 90 K1=1,NTENS
        DFDSIG(K1)=DFDJ2*S_DEV(K1)+DFDI1*DUNIT(K1)
 90   CONTINUE
      DNUMER=0.0D0
      DDENOM=0.0D0
C     ININTIALIZE
      DO 95 K1=1,NTENS
        DCEDFDSIG(K1)=0.0D0
        DELSTRESS(K1)=0.0D0
 95   CONTINUE
      DO 110 K1=1,NTENS
        DO 100 K2=1,NTENS
          DCEDFDSIG(K2)=DCEDFDSIG(K2)+DDSDDE(K2,K1)*DFDSIG(K1)
C         NOTE: CE:DFDSIG AND DFDSIG:CE ARE EQUAL
          DFDSIGCE(K2)=DCEDFDSIG(K2)
          DELSTRESS(K2)=DELSTRESS(K2)+DDSDDE(K2,K1)*DSTRAN(K1)
 100    CONTINUE
 110  CONTINUE
      DO 120 K1=1,NTENS
        DNUMER=DNUMER+DFDSIG(K1)*DELSTRESS(K1)
        DDENOM=DDENOM+DFDSIG(K1)*DCEDFDSIG(K1)
 120  CONTINUE
      DDENOM=DDENOM-DFDKAPPA*DHQ
      DELGAMMA=DNUMER/DDENOM
      IF (DELGAMMA.LT.0.0D0) THEN
           DELGAMMA=0.0D0
      ENDIF
C
C     UPDATE TOTAL STRESS
C
      DO 130 K1=1,NTENS
```

```
            STRESS(K1)=STRESS(K1)-DELGAMMA*DCEDFDSIG(K1)
 130    CONTINUE
C
C       UPDATE KAPPA
C
        DKAPPA=DKAPPA+DELGAMMA*DHQ
C
C       UPDATE X
C
        DX=(DKAPPA+DI1MAX*(DCR-1))/DCR
C
C       UPDATE CONSISTENT ELASTO-PLASTIC TANGENT
C
        DO 150 K1=1,NTENS
            DO 140 K2=1,NTENS
                DPRODUCT(K1,K2)=DFDSIGCE(K1)*DCEDFDSIG(K2)
 140        CONTINUE
 150    CONTINUE
        DO 170 K1=1,NTENS
            DO 160 K2=1,NTENS
                DDSDDE(K1,K2)=DDSDDE(K1,K2)-(ONE/DDENOM)*DPRODUCT(K1,K2)
 160        CONTINUE
 170    CONTINUE
C
C       END PLASTIC FLOW
        ELSE
C
```

```
C       ELASTIC
        DELGAMMA=0.0D0
C
        ENDIF
C
C       TOTAL VOLUMETRIC STRAIN CALC
        DELEV=DSTRAN(1)+DSTRAN(2)+DSTRAN(3)
C
C       STORE VARIABLES IN STATE VARIABLE ARRAY
C
        STATEV(1)=DKAPPA
        STATEV(2)=DX
        STATEV(3)=STATEV(3)+DELGAMMA
        STATEV(4)=DELGAMMA
C       UPDATE EVP
        STATEV(5)=STATEV(5)+THREE*DELGAMMA*DFDI1
        STATEV(6)=STATEV(6)+DELEV
        STATEV(7)=STATEV(7)+DSTRAN(1)
        STATEV(8)=STATEV(8)+DSTRAN(2)
        STATEV(9)=STATEV(9)+DSTRAN(3)
        STATEV(10)=STATEV(10)+DSTRAN(4)
        STATEV(11)=STATEV(11)+DSTRAN(5)
        STATEV(12)=STATEV(12)+DSTRAN(6)
C
        RETURN
        END
C
```

Appendix B

Arenisca ABAQUS VUMAT

```
      SUBROUTINE VUMAT(
C Read only:
     1 NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL,
     2 STEPTIME, TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH,
     3 PROPS, DENSITY, STRAININC, RELSPININC,
     4 TEMPOLD, STRETCHOLD, DEFGRADOLD, FIELDOLD,
     3 STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD,
     6 TEMPNEW, STRETCHNEW, DEFGRADNEW, FIELDNEW,
C Write only:
     7 STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
C.
      INCLUDE 'VABA_PARAM.INC'
C
C...Arrays dimensioned by (*) are not used
      DIMENSION PROPS(NPROPS), DENSITY(NBLOCK),
     1 COORDMP(NBLOCK,*),
     2 CHARLENGTH(NBLOCK), STRAININC(NBLOCK,NDIR+NSHR),
     3 RELSPININC(NBLOCK,NSHR), TEMPOLD(NBLOCK),
     4 STRETCHOLD(NBLOCK,NDIR+NSHR),
     4 DEFGRADOLD(NBLOCK,NDIR+NSHR+NSHR),
     5 FIELDOLD(NBLOCK,NFIELDV), STRESSOLD(NBLOCK,NDIR+NSHR),
     6 STATEOLD(NBLOCK,NSTATEV), ENERINTERNOLD(NBLOCK),
     7 ENERINELASOLD(NBLOCK), TEMPNEW(NBLOCK),
     8 STRETCHNEW(NBLOCK,NDIR+NSHR),
     8 DEFGRADNEW(NBLOCK,NDIR+NSHR+NSHR),
     9 FIELDNEW(NBLOCK,NFIELDV),
     1 STRESSNEW(NBLOCK,NDIR+NSHR), STATENEW(NBLOCK,NSTATEV),
     1 ENERINTERNNEW(NBLOCK), ENERINELASNEW(NBLOCK)
C
C and for locally defined arrays
      DIMENSION S_DEV(6),DFDSIG(6),DUNIT(6),DCEDFDSIG(6),
     1 DFDSIGCE(6),DELSTRESS(6),DPRODUCT(6,6),DDSDDE(6,6),
     2 STRESS(6),DSTRAN(6),STRESS_N0(6),STRESS_N1(6),
     3 STRESSINC(6)
C.
      CHARACTER*80 CMNAME
C.
      PARAMETER (ONE=1.0D0,TWO=2.0D0,THREE=3.0D0,SIX=6.0D0,
     1 SMALL=1.D-12,DHALF=0.5D0,ZERO=0.0D0)
C
C ----------------------------------------------------------------
C      VUMAT FOR ISOTROPIC SMALL STRAIN ELASTO-PLASTICITY
C      ARENISCA PLASTICITY MODEL
C      --LINEAR ISOTROPIC ELASTICITY, ASSOCIATIVE PLASTICITY
C      --NO THIRD INVARIANT DEPENDENCE, NO LIMIT SURFACE
C      GENERAL 3D FORMULATION
```

```
C ----------------------------------------------------------------------
C       STRESS CONVENTION:
C       STRESSNEW(1) - SIGMA_11
C       STRESSNEW(2) - SIGMA_22
C       STRESSNEW(3) - SIGMA_33
C       STRESSNEW(4) - SIGMA_12
C       STRESSNEW(5) - SIGMA_23
C       STRESSNEW(6) - SIGMA_13
C ----------------------------------------------------------------------
C       DEFGRAD CONVENTION:
C       DEFGRADNEW(1) - F_11
C       DEFGRADNEW(2) - F_22
C       DEFGRADNEW(3) - F_33
C       DEFGRADNEW(4) - F_12
C       DEFGRADNEW(5) - F_23
C       DEFGRADNEW(6) - F_31
C       DEFGRADNEW(7) - F_21
C       DEFGRADNEW(8) - F_32
C       DEFGRADNEW(9) - F_13
C ----------------------------------------------------------------------
C       MATERIAL PROPERTIES:
C       PROPS(1) - K - BULK MODULUS
C       PROPS(2) - G OR MU - SHEAR MODULUS
C       PROPS(3) - BETA - TAN(FRICTION ANGLE)
C       PROPS(4) - I1MAX - TENSILE LIMIT
C       PROPS(5) - CR - ELIPTICITY PARAMETER
C       PROPS(6) - P0 - CRUSH CURVE - PRESSURE AT ELASTIC LIMIT
```

```
C     PROPS(7) - P1 - CRUSH CURVE - SLOPE OF PORO VS. PRESS CURVE
C     PROPS(8) - P3 - CRUSH CURVE - MAX PLAST VOL STRAIN
C     PROPS(9) - TAU - VISCOUS REGULARIZATION PARAMETER
C ----------------------------------------------------------------
C     INTERNAL STATE VARIABLES:
C     STATEV(1) - KAPPA
C     STATEV(2) - X
C     STATEV(3) - GAMMA (just for output purposes)
C     STATEV(4) - DELGAMMA (just for output purposes)
C     STATEV(5) - EVP (plastic volumetric strain)
C     STATEV(6) - EV (total volumetric strain)
C     STATEV(7) - E11
C     STATEV(8) - E22
C     STATEV(9) - E33
C     STATEV(10) - E12
C     STATEV(11) - E13
C     STATEV(12) - E23
C ----------------------------------------------------------------
C
C
      NTENS=NDIR+NSHR
C.
      DO 900 I=1,NBLOCK
C
C     ELASTIC PARAMETERS
C.
      EK=PROPS(1)
      EMU=PROPS(2)
      ELAMB=EK-(TWO*EMU/THREE)
C
C     HARDENING PARAMETERS
C
      DBETA=PROPS(3)
      DI1MAX=PROPS(4)
      DCR=PROPS(5)
      DP0=PROPS(6)
      DP1=PROPS(7)
      DP3=PROPS(8)
C
C     RECOVER ISVS
C.
      DKAPPA=STATEOLD(I,1)
C     SAVE DKAPPA OLD FOR VISCOUS REGULARIZATION
      DKAPPA0=DKAPPA
      DX=(DKAPPA+DI1MAX*(DCR-ONE))/DCR
C
C     ELASTIC TANGENT MODULUS (IN MATRIX FORM)
C     INITIALIZE
      DO 20 K1=1,NTENS
        DO 10 K2=1,NTENS
          DDSDDE(K2,K1)=0.0D0
 10     CONTINUE
 20   CONTINUE
C     ASSIGN PARAMETERS
```

```
      DO 40 K1=1,NDIR
        DO 30 K2=1,NDIR
          DDSDDE(K2,K1)=ELAMB
 30     CONTINUE
        DDSDDE(K1,K1)=TWO*EMU+ELAMB
 40   CONTINUE
C.    ”The␣shear␣strain␣components␣in␣user␣subroutine␣VUMAT␣are␣stored␣as␣tensor
C␣␣␣␣␣components␣and␣not␣as␣engineering␣components;␣this␣is␣different␣from␣user
C␣␣␣␣␣subroutine␣UMAT␣in␣Abaqus/Standard,␣which␣uses␣engineering␣components.”
      DO 50 K1=NSHR+1,NTENS
        DDSDDE(K1,K1)=TWO*EMU
 50   CONTINUE
C
C     EXTRACT STRESSOLD AND STRAININC
      DO 51 K1=1,NTENS
C       STRESS = STRESS AT T_N0
        STRESS(K1)=STRESSOLD(I,K1)
C       SAVE STRESSOLD FOR VISCOUS REGULARIZATION
        STRESS_N0(K1)=STRESSOLD(I,K1)
        DSTRAN(K1)=STRAININC(I,K1)
 51   CONTINUE
C
C     CALCULATE TRIAL STRESS USING INCREMENTAL TOTAL STRAIN
C
C.    INITIALIZE STRESSINC
      DO 52 K1=1,NTENS
        STRESSINC(K1)=ZERO
 52   CONTINUE
      DO 70 K1=1,NTENS
        DO 60 K2=1,NTENS
          STRESSINC(K2)=STRESSINC(K2)+DDSDDE(K2,K1)*DSTRAN(K1)
 60     CONTINUE
 70   CONTINUE
      DO 71 K1=1,NTENS
C        STRESS NOW EQUALS STRESS_TR
         STRESS(K1)=STRESS(K1)+STRESSINC(K1)
 71   CONTINUE
C
C     CALCULATE MEAN AND DEVIATORIC TRIAL STRESSES (CREATE UNIT TENSOR)
C.
      DO 80 K1=1,NTENS
         S_DEV(K1)=STRESS(K1)
         IF (K1.LE.NDIR) THEN
           DUNIT(K1)=ONE
         ELSE
           DUNIT(K1)=0.0D0
         ENDIF
 80   CONTINUE
      STRESS_MEAN=(STRESS(1)+STRESS(2)+STRESS(3))/THREE
      S_DEV(1)=S_DEV(1)-STRESS_MEAN
      S_DEV(2)=S_DEV(2)-STRESS_MEAN
      S_DEV(3)=S_DEV(3)-STRESS_MEAN
C
C     CALCULATE TRIAL STRESS INVARIANTS
```

```
C.
      DI1=STRESS_MEAN*THREE
      D2J2=S_DEV(1)*S_DEV(1)+S_DEV(2)*S_DEV(2)+
     1       S_DEV(3)*S_DEV(3)+TWO*(S_DEV(4)*S_DEV(4)+
     1       S_DEV(5)*S_DEV(5)+S_DEV(6)*S_DEV(6))
      DJ2=D2J2/TWO
C
C     CALCULATE TRIAL YIELD FUNCTION
C.
      IF (DI1.GE.DKAPPA) THEN
             DFC=ONE
      ELSE
             DFC=(DKAPPA-DI1)/(DKAPPA-DX)
             DFC=DFC*DFC
             DFC=ONE-DFC
      ENDIF
      DFF=DBETA*(DI1MAX-DI1)
      IF (DI1.GE.DKAPPA) THEN
             FTRIAL=DJ2-DFF*DFF
      ELSE
             FTRIAL=DJ2-DFF*DFF*DFC
      ENDIF
C
C     CHECK FOR YIELDING
C.
      IF (FTRIAL.GE.0.0) THEN
C
```

```
C     PLASTIC FLOW
C
C     DFDSIG DERIVATIVES
      IF (DI1.GE.DKAPPA) THEN
          DFCDI1_FC=0.0D0
      ELSE
          DFCDI1_FC=(DKAPPA-DI1)/((DKAPPA-DX)*(DKAPPA-DX))
      ENDIF
      DFFDI1=-DBETA
      IF (DI1.GE.DKAPPA) THEN
          DFDI1=-TWO*DFF*DFFDI1
      ELSE
          DFDI1=-TWO*(DFF*DFFDI1*DFC+DFF*DFF*DFCDI1_FC)
      ENDIF
      DFDJ2=ONE
C     DFDKAPPA DERIVATIVES
      IF (DI1.GE.DKAPPA) THEN
          DFCDKAPPA=0.0D0
          DFDKAPPA=0.0D0
      ELSE
          DFCDKAPPA=-TWO*DFCDI1_FC
          DFCDKAPPA_FC=DFCDKAPPA*
     1                    (ONE+DCR*DBETA*(DKAPPA-DI1)/(DKAPPA-DX))
          DFDKAPPA=-DFF*DFF*DFCDKAPPA_FC
      ENDIF
C     HARDENING FUNCTION DERIVATIVES
      DEPSVPDX=DP1*DP3*EXP(-DP1*(DP0-DX))
      DXDKAPPA=ONE/DCR
      DHQ=THREE*DFDI1/(DEPSVPDX*DXDKAPPA)
C     MATRIX MULTIPLICATION
      DO 90 K1=1,NTENS
         DFDSIG(K1)=DFDJ2*S_DEV(K1)+DFDI1*DUNIT(K1)
 90   CONTINUE
      DNUMER=0.0D0
      DDENOM=0.0D0
C     ININTIALIZE
      DO 95 K1=1,NTENS
         DCEDFDSIG(K1)=0.0D0
         DELSTRESS(K1)=0.0D0
 95   CONTINUE
      DO 110 K1=1,NTENS
         DO 100 K2=1,NTENS
            DCEDFDSIG(K2)=DCEDFDSIG(K2)+DDSDDE(K2,K1)*DFDSIG(K1)
C           NOTE: CE:DFDSIG AND DFDSIG:CE ARE EQUAL
            DFDSIGCE(K2)=DCEDFDSIG(K2)
            DELSTRESS(K2)=DELSTRESS(K2)+DDSDDE(K2,K1)*DSTRAN(K1)
 100     CONTINUE
 110  CONTINUE
      DO 120 K1=1,NTENS
         DNUMER=DNUMER+DFDSIG(K1)*DELSTRESS(K1)
         DDENOM=DDENOM+DFDSIG(K1)*DCEDFDSIG(K1)
 120  CONTINUE
      DDENOM=DDENOM-DFDKAPPA*DHQ
      DELGAMMA=DNUMER/DDENOM
      IF (DELGAMMA.LT.0.0D0) THEN
         DELGAMMA=0.0D0
      ENDIF
C
C     UPDATE TOTAL STRESS
C.
      DO 130 K1=1,NTENS
C        STRESS IS NOW EQUAL TO STRESSNEW
         STRESS(K1)=STRESS(K1)-DELGAMMA*DCEDFDSIG(K1)
C        SAVE STRESSNEW FOR VISCOUS REGULARIZATION
         STRESS_N1(K1)=STRESS(K1)
  130 CONTINUE
C
C     UPDATE KAPPA
C.
      DKAPPA=DKAPPA+DELGAMMA*DHQ
C
C     END PLASTIC FLOW
      ELSE
C
C     ELASTIC
      DO 140 K1=1,NTENS
C.          NEED TO SAVE STRESSNEW FOR VISCOUS REGULARIZATION
            STRESS_N1(K1)=STRESS(K1)
  140 CONTINUE
      DELGAMMA=0.0D0
C
```

```
C
      ENDIF
C
C     POST STRESS UPDATE PROCESSING
C
C...  INCORPORATE VISCOPLASTICITY
      DTAU=PROPS(9)
C
C
      IF(DTAU.LT.SMALL) THEN
        DFACTOR1=ZERO
        DFACTOR2=DTAU/DT
      ELSE
        DFACTOR1=EXP(-DT/DTAU)
        DFACTOR2=DTAU/DT
      ENDIF
C
C
      DO 240 K1=1,NTENS
         STRESSNEW(I,K1)=DFACTOR1*STRESS_N0(K1)
     1        +(ONE-DFACTOR1)*STRESS_N1(K1)
     2        +DFACTOR2*(ONE-DFACTOR1)*STRESSINC(K1)
  240 CONTINUE
      DKAPPA=DFACTOR1*DKAPPA0+(ONE-DFACTOR1)*DKAPPA
C
C     UPDATE X
      DX=(DKAPPA+DI1MAX*(DCR-1))/DCR
```

```
C
C        TOTAL VOLUMETRIC STRAIN CALC
         DELEV=DSTRAN(1)+DSTRAN(2)+DSTRAN(3)
C
C...STORE VARIABLES IN STATE VARIABLE ARRAY
C.
         STATENEW(I,1)=DKAPPA
         STATENEW(I,2)=DX
         STATENEW(I,3)=STATEOLD(I,3)+DELGAMMA
         STATENEW(I,4)=DELGAMMA
         STATENEW(I,5)=STATEOLD(I,5)+THREE*DELGAMMA*DFDI1
         STATENEW(I,6)=STATEOLD(I,6)+DELEV
         STATENEW(I,7)=STATEOLD(I,7)+DSTRAN(1)
         STATENEW(I,8)=STATEOLD(I,8)+DSTRAN(2)
         STATENEW(I,9)=STATEOLD(I,9)+DSTRAN(3)
         STATENEW(I,10)=STATEOLD(I,10)+DSTRAN(4)
         STATENEW(I,11)=STATEOLD(I,11)+DSTRAN(5)
         STATENEW(I,12)=STATEOLD(I,12)+DSTRAN(6)
C
C
  900    CONTINUE
C
C
         RETURN
         END
```