Sandia National Laboratories

Operated for the United States Department of Energy
by National Technology and Engineering Solutions
of Sandia, LLC.

Albuquerque, New Mexico 87185
Livermore, California 94550

date: October 19th, 2024

to: Distribution

from: Benjamin Reedlunn, Principal Member of the Technical Staff, Org. 1558
Jibril B. Coulibaly, Post-Doctoral Appointee, Org. 1558
Devin O'Connor, Senior Member of the Technical Staff, Org. 1558
Frank Beckwith, Senior Member of the Technical Staff, Org. 1542

Reedlunn, Benjamin Digitally signed by Reedlunn, Benjamin
Date: 2024.12.11 17:49:35 -07'00'

subject: Status of Efforts to Model the Geomechanics of Empty Room Collapse and Reconsolidation at the Waste Isolation Pilot Plant, October 2024

This memorandum presents the status of ongoing efforts to simulate the collapse, rubble pile formation, and subsequent reconsolidation of empty, abandoned rooms at the Waste Isolation Pilot Plant. The memorandum focuses first on the development of bond-based failure and pinball contact for the Reproducing Kernel Particle Method (RKPM) capability within Sierra/SolidMechanics. The bond-based failure was improved by adding a secondary failure criterion to decay remnant bonds left over by the pre-existing primary criterion. The pinball contact was enhanced to include automatic insertion of pinballs on new fracture surfaces, two potential locations to place the pinballs, two ways to assemble pinball forces, and frictional forces. The salt constitutive model was also re-formulated and re-implemented to converge more reliably. These three capabilities, bond-based failure, pinball contact, and the salt model, were combined in preliminary empty room collapse simulations. Although attempts to predict a natural roof fall were elusive, artificially pre-damaging the salt above the room caused a block of salt to detach from the roof, the block to land on the floor, and the room to reconsolidate.

1 Introduction

The Waste Isolation Pilot Plant (WIPP) in southeast New Mexico is an operating geologic repository for defense-related transuranic nuclear waste. The repository is about 660 m beneath the surface in a bedded salt formation known as the Salado. The ability of the salt to isolate the waste from the biosphere is affected by the repository's porosity and permeability.

Sandia has models and estimates for the porosity and permeability of rooms filled with waste containers [Vignes et al., 2023] and crushed salt [Callahan, 1999], but Sandia does not have a model for the porosity or permeability of empty rooms. As discussed in Reedlunn et al. [2019, Section 1.1], abandoned empty rooms at the WIPP are prone to collapse, resulting in rubble piles on their floors. The surrounding salt will eventually compact these rubble piles, but predicting the evolution of rubble pile porosity and permeability is quite challenging.

Reedlunn et al. [2019] presented the first work towards modeling the geomechanics and geohydraulics of empty room collapse and reconsolidation. The bulk of the report focused on assessing two "meshfree" methods well-suited to simulating the relevant geomechanical processes. A team from Brown University assessed the Immersed Isogeometric Analysis (IGA) meshfree method, which reduces to a variant of the Material Point Method (MPM) when fluids are absent, while a team from University of California, San Diego (UCSD) assessed the Reproducing Kernel Particle Method (RKPM). Both methods accurately simulated the gradual closure of a two-dimensional room without any roof falls. More importantly, both methods reasonably simulated two-dimensional roof falls and the subsequent rubble pile compaction. Furthermore, results from both methods prompted the discovery of a previously overlooked mechanism of empty room closure (see Reedlunn et al. [2019, Chapter 6]). The final chapter of Reedlunn et al. [2019] also presented preliminary efforts to predict relationships between the porosity and permeability of synthetically deposited rubble piles.

This memorandum reviews the status of Sandia's continued efforts to model the geomechanics of empty room collapse and reconsolidation. Further efforts to study the geohydraulics of synthetically deposited rubble piles will be presented at a later date. Since Reedlunn et al. [2019] was released, the geomechanical modeling has focused on developing Sandia's implementation of RKPM, improving the robustness of the salt constitutive model, and using these new features in preliminary room collapse simulations. These three topics are covered in Sections 2 to 4, respectively.

2 RKPM Development

2.1 Background and Motivation

Reedlunn et al. [2019] showed that MPM and RKPM are both well-suited to simulating room collapse and reconsolidation, but they did not uncover any substantial advantages of one method over the other. We subsequently chose to focus on the RKPM herein due to UCSD's promising RKPM room collapse and reconsolidation simulations [Reedlunn et al., 2019, Chapter 4] and Sandia's previous investments into the RKPM [Koester and Beckwith, 2017, Koester et al., 2019]. It is conceivable that other meshless methods would also be well-suited to this problem. Fortunately, ideas from other meshless methods can often be adapted and applied to RKPM without excessive levels of effort.

Another related decision was whether to sub-contract with UCSD for further RKPM simulations of room collapse or to continue to develop an in-house capability at Sandia. The UCSD group has developed their code for many more years than Sandia has spent on its

RKPM code. We elected to continue developing Sandia's RKPM code, rather than repeatedly sub-contract with UCSD, for the following reasons:

1. Creating a new sub-contract for each new investigation is burdensome. An in-house capability can be used to answer small and large questions as they arise.

2. The UCSD group would have the final say about modifications to their code base, while we have the final say about which issues to focus on and how we wish to solve them with a Sandia code.

3. Sandia's RKPM performance scales better with the number of processors than UCSD's code [Reedlunn et al., 2019, Section 4.4.1.1 and 5.3]

4. Qualifying a research code developed by a university, per the requirements in Long [2024], would be a substantial, potentially problematic, task. Sandia's RKPM, however, was easily integrated into Sierra/SolidMechanics [Sierra/SolidMechanics Team, 2024], whose finite element method (FEM) implementation was recently qualified [Reedlunn, 2022b]. Extending the Sierra/SM qualification to cover its RKPM capabilities is easier than qualifying a wholly new piece of software. In addition, Sierra/SM has tighter software quality assurance procedures than university groups typically do.

5. Synchronizing Sandia's development of a new salt constitutive model with UCSD's RKPM code might be difficult because it can be challenging for Sandia to share source code. By contrast, Sandia's new salt constitutive model is being developed in Sierra/SM, so synchronization is trivial.

After deciding to use Sandia's RKPM code, the first activity was to integrate it into Sierra/SM. Sandia's RKPM code was initially referred to as the Conforming Reproducing Kernel (CRK) mini-app. Integrating the CRK mini-app into Sierra/SM was relatively straightforward because CRK had been built from the beginning using the Sierra Tool Kit.

Introducing the motivation for the next two RKPM development activities requires some background information on kernels. Kernels are analogous to the shape functions in the FEM, as they are used to construct the approximation functions. The schematics in Fig. 1 depict Lagrangian and semi-Lagrangian kernels in the reference ($t = 0$) and current ($t = t_n$) configurations. The Lagrangian kernel centered at material point I can distort in order to encompass the same neighboring material points in the reference and current configurations. The code constructs Lagrangian kernels and neighbor lists only once in the reference configuration. In contrast, the semi-Lagrangian kernel centered at material point I cannot distort, meaning that material points can move in and out of the kernel's support. The code, typically, constructs semi-Lagrangian kernels and neighbor lists on every time step. As a result, semi-Lagrangian kernels can handle more extreme deformations, yet are computationally more expensive. Integration methods of the semi-Lagrangian formulation also become more challenging due to the difficulty of generating conformal integration cells [Chen et al., 2017].

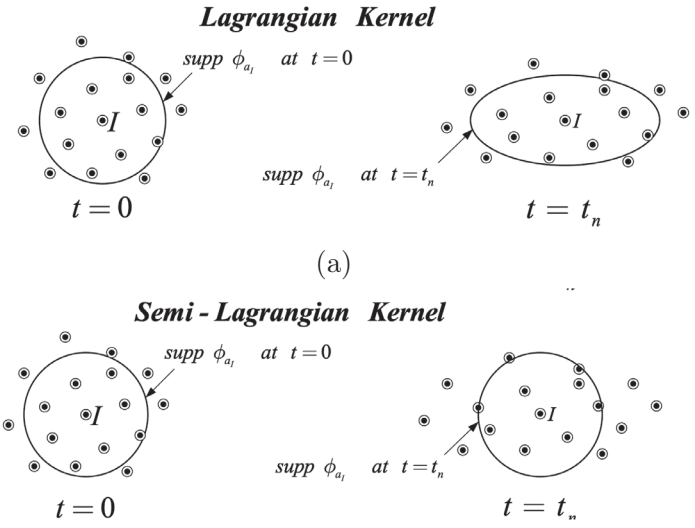


Figure 1: Schematics of Lagrangian and semi-Lagrangian kernels in the reference and current configurations [Chen et al., 2017, Chen and Hillman, 2020].

There can also be differences in the way Lagrangian and semi-Lagrangian kernels handle fracture and contact. A Lagrangian kernel requires a dedicated fracture method to cause it to start ignoring material points inside its support in the reference configuration. It also requires a dedicated contact enforcement algorithm to ensure material points, who are not neighbors in the reference configuration, begin interacting if they get close to one another in the current configuration. Semi-Lagrangian kernels, on the other hand, do not require dedicated fracture methods and contact enforcement. If one uses continuum damage mechanics or similar approaches, a material will become weak and easily deformable as it fails. Once material points with semi-Lagrangian kernels move far enough away from one another, they stop interacting, often called "numerical fracture." Similarly, semi-Lagrangian kernels allow material points, who are not neighbors in the reference configuration, to begin interacting once they move close enough to one another, referred to as "kernel contact" [Guan et al., 2011, Chi et al., 2015]. Additional modifications, however, are necessary to model friction with semi-Lagrangian kernels.

The RKPM in Sierra/SM currently only includes Lagrangian kernels. Sandia's nuclear weapons projects have been funding an implementation of semi-Lagrangian kernels, but this capability is not ready yet. Lagrangian kernels, fortunately, are believed to still be viable for several reasons:

1. The salt is not expected to undergo the extreme deformations (>100 % strain) that necessitate semi-Lagrangian kernels.

2. Large three-dimensional simulations of room collapse and reconsolidation are planned, where computational efficiency is a concern, and Lagrangian kernels are substantially

faster.

3. A promising Lagrangian fracture method, called bond-based fracture, was presented in Koester et al. [2019]. This approach allows any material to fracture once it meets a criterion, which seems preferable than requiring a material model to soften so much that material points separate on their own.

4. Although kernel contact may seem to be an advantage of semi-Lagrangian kernels, kernel contact is not expected to be a viable long-term solution. As discussed further in the introduction to Section 2.3, a dedicated contact scheme that includes explicit surfaces will likely be needed to accurately model friction, regardless of the kernel formulation.

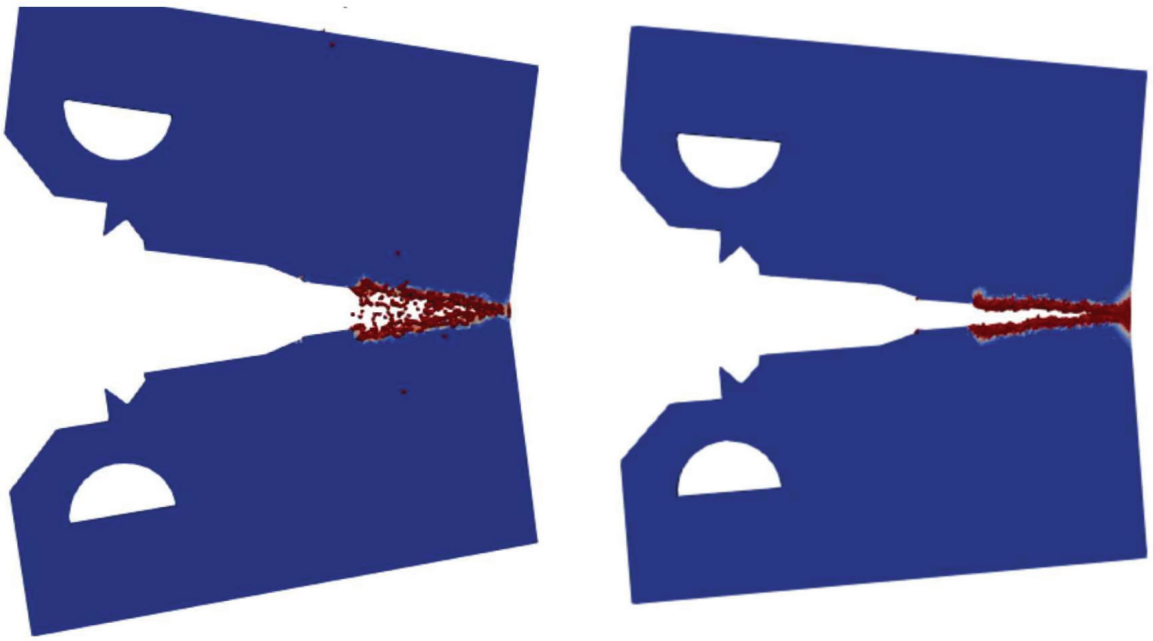
Thus, for the time being, we are pursuing Lagrangian kernels.

The remainder of this section presents Sandia's efforts to further develop two Lagrangian kernel capabilities: the bond-based fracture method (see Section 2.2) and pinball contact (see Section 2.3).

2.2 Bond-Based Failure Method

A bond-based fracture method was used in the empty room collapse simulations to allow materials to separate after fracturing without volume loss. This method adopts ideas from the peridynamics community and reformulates the internal force algorithm using pair-wise interactions between nodes. By expressing the internal forces as a collection of "bond" contributions, we may introduce fracture between neighboring nodes and degrade its contribution to the overall internal force vector according to a cohesive function of bond stretch. In a slight abuse of terminology, bond stretch as used here indicates the length of the bond, or $y_{ab} = \|\mathbf{x}_a - \mathbf{x}_b\|$, not an actual stretch in the traditional continuum mechanics sense. This introduces an intrinsic length scale to the failure formulation and therefore reduces mesh dependence often observed in other fracture methods, particularly unregularized continuum-damage and erosion-type methods. (Erosion here refers to deleting elements, or in the case of meshless methods, setting the particle's stress to zero, which is referred to as "particle erosion" [Koester et al., 2019].) As a result, the simulations show increased stability around fracture zones with less so-called "particle spray" (an instability involving significant particle disconnect due to excessive damage within a fracture zone). Fig. 2 illustrates the bond-based failure method with reduced particle spray compared to particle erosion. The bond-based failure method is formulated under a Lagrangian frame of reference, allowing for fracture to be represented free of tensile instabilities or numerical fracture.

Traditional failure methods such as particle erosion decay the total deformation gradient. This does not affect hyperelastic or viscoelastic materials but may lead to a load reversal for path-dependent materials. For example, if the failure criterion has been hit after a plasticity model has accrued some plastic strain under uniaxial tension, then degrading the deformation gradient back to the identity tensor would cause the stress to drop inside the yield surface



(a) Particle erosion with particle spray

(b) Bond-based failure method without particle spray

Figure 2: Comparison of compact-tension specimen using bond-based failure and particle erosion [Koester et al., 2019]

and become compressive, which is not physically observed during quasi-static failure. In contrast to traditional approaches, the bond-based failure method decays the *increment* of the deformation gradient instead of the total deformation gradient. This precludes the aforementioned behavior and would not cause an unphysical stress to develop as a result of strain unloading.

2.2.1 Bond-based Failure Preliminaries

A brief introduction to the bond-based failure method is described here. The key equations and a new cohesive degradation function (see Eq. (5)) are discussed. For a more complete description of bond-based failure and its formulation, the reader is referred to Koester et al. [2019, Section 5].

The total power of a domain is given as:

$$\dot{\mathcal{A}} = \sum_{a} \mathbf{P}_a : \dot{\mathbf{F}}_a V_a = \sum_{a,b|b<a} \mathbf{f}_{ab} \cdot \dot{\mathbf{y}}_{ab}, \tag{1}$$

where a represents a node of the simulation, $\mathbf{P}_a$ is the first Piola-Kirchhoff stress at node a, $\dot{\mathbf{F}}_a$ is the time rate of the deformation gradient, and V_a is the integration volume. The second term is equivalently re-expressed as a collection of bonds denoted by the pair of nodes

a and b, where $\mathbf{y}_{ab} = \mathbf{x}_b - \mathbf{x}_a$ is the current bond "stretch," and $\mathbf{f}_{ab}$ is the bond force between the two nodes as follows:

$$\mathbf{f}_{ab} = \mathbf{P}_a : \nabla N_b(\mathbf{X}_a) V_a - \mathbf{P}_b : \nabla N_a(\mathbf{X}_b) V_b, \tag{2}$$

where $N_a(\mathbf{X}_b)$ is the shape function of node a at material point $\mathbf{X}_b$.

The deformation gradient and its time derivative are approximated following,

$$\begin{aligned} \mathbf{F}_a &= \sum_b \nabla N_b(\mathbf{X}_a) \mathbf{y}_{ab}, \\ \dot{\mathbf{F}}_a &= \sum_b \nabla N_b(\mathbf{X}_a) \dot{\mathbf{y}}_{ab}, \end{aligned} \tag{3}$$

which are used in the constitutive law to estimate the first Piola-Kirchhoff stress, $\mathbf{P}_a$.

Assuming a bond has failed, the connecting force between two nodes a and b must be degraded for separation to occur. As mentioned above, the bond-based failure method degrades the deformation gradient increment. To this end, a *modified* deformation gradient rate is defined as follows,

$$\dot{\bar{\mathbf{F}}}_a = \sum_b \nabla N_b(\mathbf{X}_a) \dot{\bar{\mathbf{y}}}_{ab}, \tag{4}$$

where $\dot{\bar{\mathbf{y}}}_{ab} = c(y_{ab}) \dot{\mathbf{y}}_{ab}$ and $y_{ab} = \|\mathbf{y}_{ab}\|$. $c(y)$ is a cohesive degradation function acting over a failed bond defined as follows,

$$c(y) = \begin{cases} 1 & \text{if } y \leq y_c, \\ 1 - \dfrac{y_f}{y} \dfrac{y - y_c}{u_{cod}} & \text{if } y_c \leq y \leq y_c + u_{cod}, \\ 0 & \text{if } y_c + u_{cod} \leq y, \end{cases} \tag{5}$$

where y_c is the bond stretch at first failure, $y_f = y_c + u_{cod}$ is the final bond stretch corresponding to a fully broken bond, and u_{cod} is the critical opening displacement of the bond*. Note, this definition differs from the original degradation function in Koester et al. [2019] in that we have chosen a rational, cohesive decay function instead of linear and is more consistent with elastic-damage models [Lemaitre and Desmorat, 2006]. The criteria by which a bond may fail are described below in Section 2.2.2 and Section 2.2.3. The same cohesive degradation function, $c(y)$, is used for both cases. The modified deformation gradient rate, $\dot{\bar{\mathbf{F}}}_a$, is power conjugate to the modified first Piola-Kirchhoff stress, $\bar{\mathbf{P}}_a$, which similarly leads to a modified connecting force,

$$\bar{\mathbf{f}}_{ab} = \bar{\mathbf{P}}_a : \nabla N_b(\mathbf{X}_a) V_a - \bar{\mathbf{P}}_b : \nabla N_a(\mathbf{X}_b) V_b. \tag{6}$$

The total power of the system is then given by

$$\dot{\mathcal{A}} = \sum_{a,b|b<a} \bar{\mathbf{f}}_{ab} \cdot c(y_{ab}) \dot{\mathbf{y}}_{ab} = \sum_{a,b|b<a} c(y_{ab}) \bar{\mathbf{f}}_{ab} \cdot \dot{\mathbf{y}}_{ab}, \tag{7}$$

*The cohesive degradation function here is denoted $c(y)$ whereas Koester et al. [2019] uses $g(y)$. We changed the definition to avoid confusion with the gap function as described in Eq. (10) of Section 2.3.1.

which leads to an effective bond force as follows,

$$\mathbf{f}_{ab} = c(y_{ab})\bar{\mathbf{f}}_{ab}. \tag{8}$$

Finally, summing all of the bond forces about node a gives the final internal force in the bond-based failure method:

$$\mathbf{f}_a^{\text{int}} = \sum_b \mathbf{f}_{ab}. \tag{9}$$

Some remarks are noted:

1. The bond-based failure method models fracture by degrading the increment of deformation gradient and the bond-connectivity forces between nodes. No explicit failure surface is actually created in this process. A fracture surface is assumed to occur between those nodes which have fully disconnected from each other.

2. Though the increment of deformation gradient and the bond-connectivity forces between two nodes are degraded, *the Lagrangian shape functions and their displacement coefficients are not*. This effectively means the displacement of a neighboring node on the opposite side of a bond may still affect the other node despite full degradation of the bond. Further discussion may be found in Section 2.2.5.

2.2.2 Primary Bond Failure Criterion

The bond-based failure method formulated in Koester et al. [2019] used a single failure criterion whereby a bond was considered failed once both nodes of the bond reached some nodal failure criterion. We refer to this technique as the *primary bond failure criterion*. A common nodal failure criterion is a material state variable reaching a specific value: $\xi = \xi^{\text{fail}\dagger}$. Common state variables for use with the primary bond failure criterion are plastic strain or material damage if using a continuum-damage constitutive model.

While simple to implement, this primary bond criterion may lead to a delayed failure response and overestimate the overall reaction force as a result. A bond can fail once two nodes reach their failure criterion, but those same two nodes might be bonded to other nodes that have not reached their failure criterion. Much more deformation may be required to fail the remaining bonds and separate the material. In some extreme cases, a load path may be established from bonds that never reach full failure.

†Material state variables typically reside at element entities which contain material points. However, the stabilized conforming nodal integration technique (SCNI) smooths element field quantities, including stresses and strains, about each node [Chen et al., 2001]. This property allows the material state variables to be used as nodal criteria for the bond-based failure method.

2.2.3 Secondary Bond Failure Criterion

To combat these remnant bonds, the team introduced the idea of a *failure plane* which attempts to track the path of fracture. For every bond that has failed due to the primary bond criterion, a failure plane is created at the bond's center that is perpendicular to the bond. Fig. 3 illustrates the failure plane idea, which shows additional bonds which intersect, or "clip," the failure plane (note, only bonds which share a node of the failed bond are considered). These bonds are assumed to lie in the path of a potential crack, which would fail soon after.

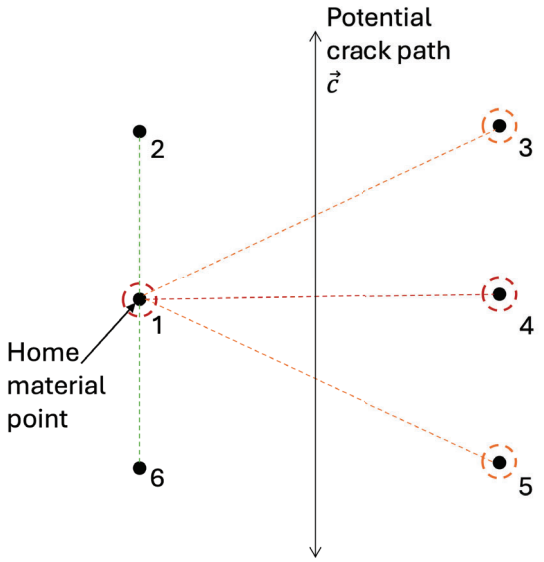


Figure 3: Illustration of a "failure plane," which marks only those bonds clipped by a plane as candidates for failure due to the secondary criterion. The red bond has failed due to the primary criterion, and the orange bonds are marked for potential failure due to the secondary failure criterion. The remaining green bonds are not clipped and thus not considered for the secondary criterion.

The failure plane marks these intersected bonds for consideration of a *secondary bond failure criterion*, which considers a bond failed if the following conditions are met:

1. The bond shares a node of a bond that has failed due to the primary bond failure criterion, or "primary-failed bond."
2. The bond is intersected, or "clipped," by a failure plane originating from the midpoint of the primary-failed bond. A failure plane may only clip bonds which share a node to the primary-failed bond from which the failure plane originates. As a result, the extents of a given failure plane is limited to the local vicinity of the primary-failed bond and its neighboring bonds.
3. The bond reaches a secondary opening displacement, u_{cs}, beyond the point when the bond was flagged as a secondary candidate due to the previous two conditions.

Here, a neighboring bond is one which shares a failed node with the bond in consideration. Note, failure planes are only created from bonds which fail due to the primary criterion. Once a bond has been marked for consideration of the secondary criterion, the current bond length is saved, denoted y_{cs}. The bond is not considered failed until it has further stretched by a secondary opening displacement, or "secondary crack" denoted u_{cs}. The value u_{cs} is usually set to be some percentage of the critical crack opening displacement, or u_{cod} (see Koester et al. [2019, Section 5.2]). Once this secondary threshold $y_c = y_{cs} + u_{cs}$ has been met, the bond degrades according the same cohesive degradation function given in Eq. (5). If the opposite node of the bond reaches its nodal failure criterion before this new criterion, then the bond fails according to the primary bond failure criterion above and y_c becomes the current bond stretch at that instant in time.

An alternative to the failure plane idea is to mark all bonds which share a node to a primary-failed bond for consideration of the secondary criterion. This idea assumes the crack will run through both nodes of the failed bond and introduces an even more relaxed criterion by which bonds can fail.

2.2.4 Bond-Based Failure Examples

The above ideas were implemented and demonstrated by modeling the Sandia Fracture Challenge scenario [Boyce et al., 2014] and plotting the force vs. displacement responses in Fig. 4. The upper and lower bounds as measured from the experiment are shown with the orange and blue lines, respectively. The simulation results using bond-based RKPM with the secondary criterion are given by the purple and red lines, where the purple line uses failure-plane clipping and the red line does not. When clipping is enabled, the response shows a slightly stronger response than the results where the secondary criterion is applied to all bonds which share a failed node. This behavior agrees with the expected result since fewer bonds will be identified for failure via the secondary criterion when using failure plane clipping, leading to a stiffer response. A comparison using finite elements is shown with the green line.

Despite the modifications made to the bond-based failure method, a small degree of residual forces persist. The residual forces lessen as the discretization is refined, but the problem does not completely go away. Fig. 5 shows an example problem where the upper yellow block and lower green block are pulled in opposite directions from each other leading to a tension failure. As bond damage accumulates and the bonds begin failing, the problem initially shows a drop in total force. (Note, "damage" here refers to the total degradation of a bond and is unrelated to material continuum damage described in Section 3.) However, the forces recover slightly before slowly decaying over time, yet they never reach a state of zero-force. We are still investigating the reason behind the residual forces where none are expected.

2.2.5 Limitations

Despite the effectiveness of bond-based failure, some limitations of the method still exist:

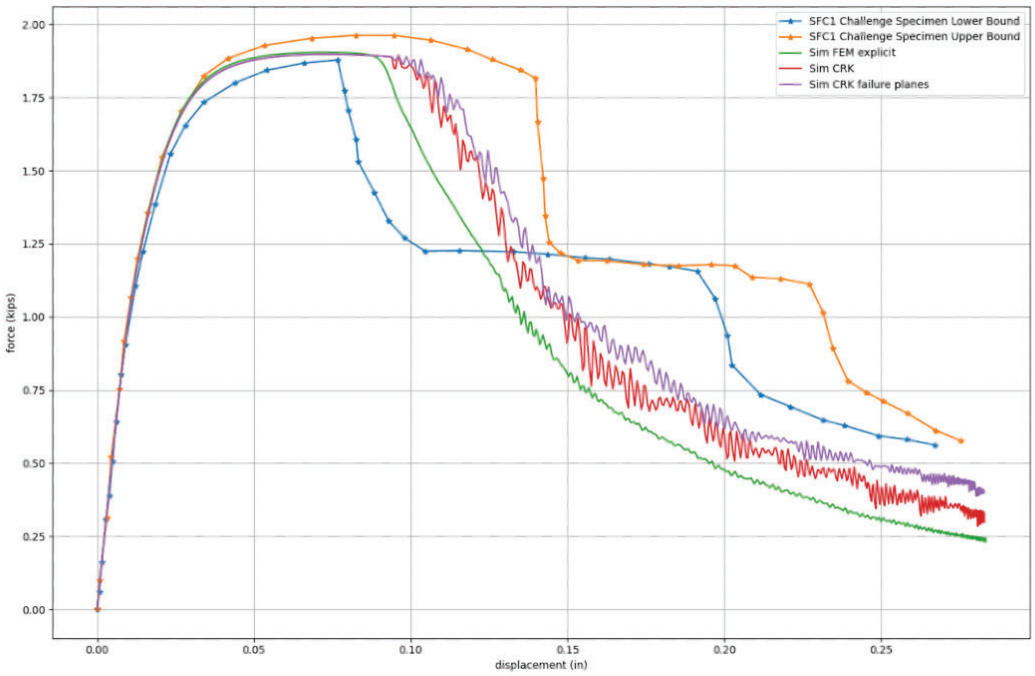
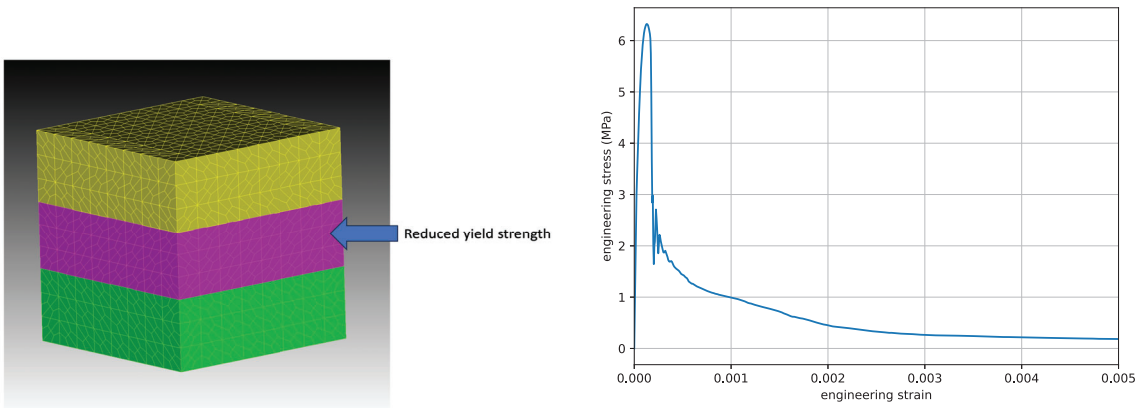


Figure 4: Force-displacement response of the Sandia Fracture Challenge with bond-based failure compared to experimental data.



(a) (b)

Figure 5: Simple "brick" geometry undergoing tension via displacement boundary conditions. The middle block is given a weaker material strength. (a) Illustration of the brick geometry undergoing tension. (b) Engineering stress - Engineering strain response of the brick geometry.

1. While the increment of deformation gradient and resulting bond forces decay according to Eq. (5), the Lagrangian shape functions do not. Because of this, displacement coefficients of a node's neighbors may still influence the position of the node even after the bonds have degraded. This causes unphysical movement of nodes that have completely disconnected from their original neighbors. Degrading the Lagrangian shape functions may negatively affect the completeness of the RK approximation and as such was limited to the increment of deformation gradient and the bond-connectivity force. An alternative may be to degrade the kernel weights and recompute the shape functions, but such an approach would change the approximation and require an updated or semi-Lagrangian formulation.

2. The bond-based failure method assumes a nodally integrated formulation, which is known to experience locking in nearly incompressible media. Sub-cell integration with the $\bar{\mathbf{F}}$-method [Moutsanidis et al., 2020, Koester et al., 2019] can alleviate this but it is not clear how to extend this to bond-based failure. Future investigations may consider modifications to the bond-based failure method and/or other failure methods that are compatible with integration schemes for near incompressibility.

2.3 Pinball Contact

Contact mechanics are necessary to accurately simulate the dynamic and quasi-static contact forces encountered during roof fall and room closure. Ideally RKPM would explicitly model contact surfaces and use well-accepted techniques such as augmented Lagrange contact [Laursen, 2003]. However, meshless methods struggle to represent surfaces due to the lack of connectivity between nodes. The presence of fracture further complicates these surfaces especially if extensive failure is predicted.

Due to these difficulties, the room collapse simulations make use of pinball contact in order to approximate contact between the various surfaces of the model. This method abandons a continuous surface representation and approximates contact by representing the surface by a collection of spheres, or "pinballs," and enforcing contact by a simple penalty parameter [Belytschko and Neal, 1991]. Although approximate, pinball contact is a simple method to approximate contact events during the room collapse simulation such as between a roof fragment and the floor. Additionally, pinballs may be inserted in damaged zones according to the bond-based failure above in order to represent contact between fracture surfaces.

The pinball contact approach has several limitations and should be considered an intermediate step towards a more accurate approach. Pinball contact discretely represents the surface with a collection of spheres, so the overall surface is rough and has many gaps between pinballs. As such, large normal contact pressures may cause small pinballs representing one surface to penetrate between small pinballs representing another surface. Treating the surface in this manner may also pose difficulties when applying pressure boundary conditions due to fluid trapped within the rubble pile. Applying a pressure to a pinball is unclear as there is no explicit surface nor is there a well defined surface normal. In the future, we will consider surface manifold techniques used with meshless methods [Gross et al., 2020, Wang et al., 2014] to more accurately simulate contact.

The pinball contact method was previously implemented in the CRK mini-app but was undocumented. The implementation only considered normal pinball forces. Section 2.3.1 describes the pinball formulation including the addition of frictional kernels. Section 2.3.2 explains how pinballs are generated with a description of pinball insertion at new fracture surfaces.

2.3.1 Pinball Contact Enforcement

Pinball contact weakly enforces a zero-gap constraint via a penalty parameter and applies the force in the radial direction of the two contacting pinballs. Using weak enforcement in this manner leads to *semi-rigid* pinball contact, where a small degree of interpenetration is expected due to the use of a penalty spring. The enforcement of pinball contact between each interaction a–b including frictional effects is described below.

2.3.1.1 Normal Contact Force The gap function, g, between two contacting pinballs must first be calculated and is defined as follows,

$$g = (R_a + R_b) - \|\mathbf{x}_a - \mathbf{x}_b\| - \delta, \tag{10}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the pinball centroids in current configuration, R_a and R_b are the pinball radii, and δ is the initial overlap. In convention with contact mechanics literature, positive values of g indicate the pinballs are penetrating whereas a negative value indicates an actual gap. The gap calculation above takes into account a pre-existing overlap, δ, to prevent spurious contact forces from generating. This is particularly useful when considering contact between newly instantiated pinballs along fracture surfaces, described in Section 2.3.2 below.

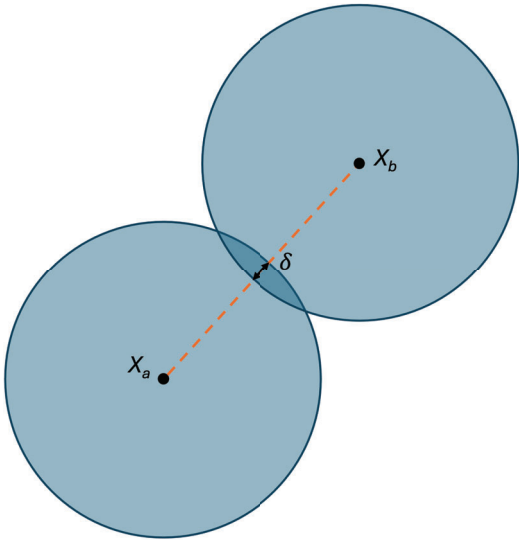
With the gap calculated above, the normal component of the local pinball force, $\mathbf{f}^c_{ab}$, is computed as

$$\mathbf{f}^c_N = k\langle g \rangle \mathbf{n}, \tag{11}$$

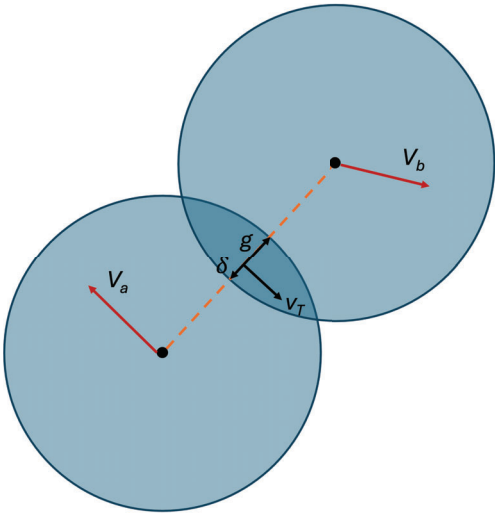
where the notation $\langle \cdot \rangle$ denotes Macaulay brackets and k is an averaged stiffness given by,

$$k = 2\beta \frac{B_a B_b R_a R_b}{B_a R_a + B_b R_b}, \tag{12}$$

where β is a dimensionless parameter scaling the penalty stiffness, and B_a and B_b are the bulk moduli associated with the two pinballs. The normal is simply the radial direction of the contacting pinballs, given by $\mathbf{n} = (\mathbf{x}_a - \mathbf{x}_b) / \|\mathbf{x}_a - \mathbf{x}_b\|$. Enhancements to the normal estimation such as taking an average surface normal between the two surfaces as described in Belytschko and Neal [1991, Section 7] were not considered. The estimation of the penalty stiffness, k, differs from previous works and treats the penalty stiffness as a spring in series instead of a spring in parallel. We feel this better represents the physics of contact should one body be much softer than the other.



(a) Pinballs in reference configuration with initial overlap, δ



(b) Pinballs in contact in current configuration

Figure 6: Illustration two pinballs in contact. $\mathbf{v}_a$ and $\mathbf{v}_b$ are the nodal velocities, g is the gap (penetration), $\mathbf{v}_T$ is the relative, tangential velocity, and δ is the initial overlap in reference configuration.

2.3.1.2 Frictional Contact Forces Friction between pinball contact is given by a regularized Coulomb friction law and is analogous to a elastic, perfectly-plastic constitutive model. The method described here solves a *penalty-regularized slip functional* as follows,

$$\Phi(\mathbf{f}_T^c, f_N^c) := \|\mathbf{f}_T^c\| - C_f f_N^c \leq 0, \tag{13a}$$

$$\dot{\mathbf{f}}_T^c = k\left[\mathbf{v}_T - \dot{\gamma}\frac{\mathbf{f}_T^c}{\|\mathbf{f}_T^c\|}\right] \tag{13b}$$

$$\dot{\gamma} \geq 0, \tag{13c}$$

$$\dot{\gamma}\Phi = 0, \tag{13d}$$

where k is the penalty stiffness as given before, C_f is the coefficient of friction, $\mathbf{v}_T$ is the relative tangent velocity, $\mathbf{f}_T^c$ and $\mathbf{f}_N^c$ are the tangent and normal contact forces, respectively, and $f_N^c = \|\mathbf{f}_N^c\|$. $\dot{\gamma}$ measures the *slip rate* of the Coulomb friction law and is analogous to the plastic strain rate in hypoelastic constitutive models. For a more comprehensive description of friction and its implementation in numerical methods, please refer to Laursen [2003].

The relative, tangential velocity vector, $\mathbf{v}_T$, is first defined

$$\begin{aligned} \mathbf{v}_{\text{rel}} &= \mathbf{v}_b - \mathbf{v}_a, \\ \mathbf{v}_T &= \mathbf{v}_{\text{rel}} - (\mathbf{n}\cdot\mathbf{v}_{\text{rel}})\,\mathbf{n}. \end{aligned} \tag{14}$$

Using a backward Euler integrator for the slip functional, a return map procedure is utilized to determine the friction at time step t_{n+1}. A trial friction is first computed using the value

of friction from the previous time step t_n as follows,

$$\mathbf{f}_T^{\text{trial}} = (\mathbf{f}_T^c)_n + k\Delta t \mathbf{v}_T, \tag{15}$$

where $\Delta t = t_{n+1} - t_n$. Finally, the trial friction is adjusted depending on whether the bodies are experiencing stick or slip, given by

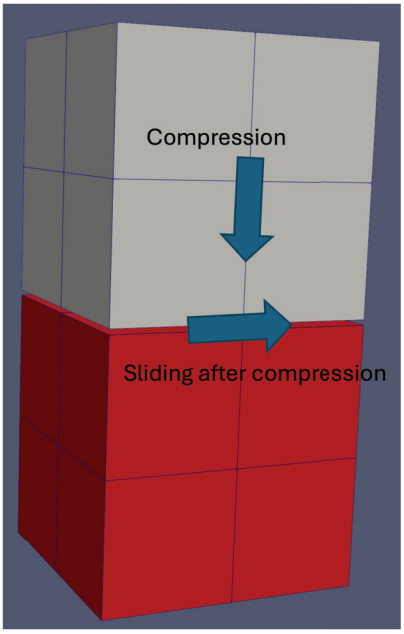
$$(\mathbf{f}_T^c)_{n+1} = \begin{cases} \mathbf{f}_T^{\text{trial}} & \text{if } \|\mathbf{f}_T^{\text{trial}}\| \leq C_f f_N^c, \\ C_f f_N^c \dfrac{\mathbf{f}_T^{\text{trial}}}{\|\mathbf{f}_T^{\text{trial}}\|} & \text{otherwise.} \end{cases} \tag{16}$$

A verification example of pinball contact with friction is given where two cubic bodies are initialized, pressed into each other to develop compressive forces, and then translated laterally to produce frictional forces.[‡] The Sierra/SM solution with RKPM employed sub-hex centered pinballs, and the displacement boundary conditions fully prescribe the problem such that the gap and relative, tangential displacement of the pinballs are known at all times. Using this information, the expected solution is obtained by explicitly integrating the above equations to compute the expected normal and frictional forces. A Python script was developed following this procedure, referred to as the "Python solution," and compared to the Sierra/SM solution as shown in Fig. 7. The Sierra/SM simulation correctly predicts a compressive force in the vertical y-direction subsequently followed by frictional forces in the transverse x-direction, which shows excellent agreement with the Python solution. Note, due to the spherical nature of the pinballs, both the compressive and frictional response decay to zero as the bodies translate because the radial distance between the two pinballs increases with sliding, thereby reducing the "gap" between the pinballs.

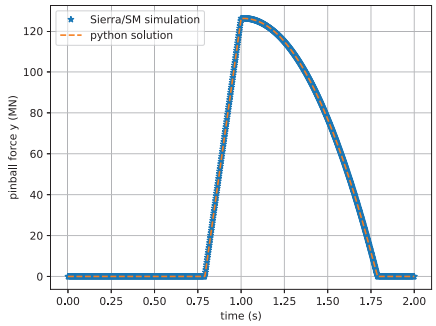
Potential future work could focus on verifying greater degrees of sliding. A test that applies a prescribed normal force to finely discretized blocks is anticipated to produce the emergence of a frictional "plateau." This plateau should have oscillations in the frictional force that hover near the anticipated mean friction value.

2.3.1.3 Assembly into the Global Force Vector The pinball contact force $\mathbf{f}_{ab}^c$ computes a local reaction between two pinballs in contact. That is, the normal and frictional forces described above are the contact forces of *the pinball interaction between nodes a and b*. These forces must be assembled into the global force vector in order for the contact forces to affect the response of the system. The pinball contact routine accomplishes this via a two-step procedure:

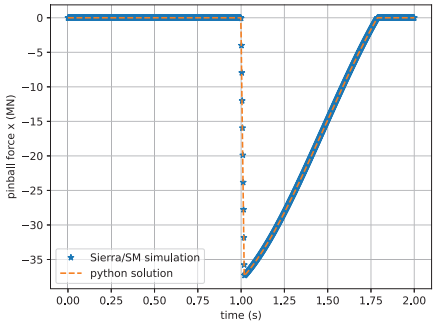
[‡]The domains in this example demonstrate the use of SCNI using a hex-primary mesh, which differs slightly to the traditional use of tetrahedral elements with RKPM. RKPM is agnostic to the meshing strategy so long as the final mesh is "refined" one level from the original mesh within Cubit. RKPM aggregates the resulting sub-hexes about each node and is completely agnostic to the original mesh they were generated from.



(a) Pinball sliding setup (compression followed by transverse displacement)



(b) Compressive pinball forces vs. time



(c) Frictional pinball forces vs. time

Figure 7: Manufactured pinball sliding verification example

1. Apply the contact force of the pinball interaction, $\mathbf{f}^c_{ab}$ to the two pinballs a and b, where the forces are equal and opposite;

$$\mathbf{f}^c_a = \sum_b \mathbf{f}^c_{ab}. \tag{17}$$

   (Note, the summation about pinball b from interaction a–b would sum $\mathbf{f}^c_{ba} = -\mathbf{f}^c_{ab}$ which is equal and opposite.)

2. Assemble the contact forces of each pinball a to the global force vector via a scattering routine or direct assembly.

In the above, $\mathbf{f}^c_a$ denotes the sum of all contact forces from each interaction involving pinball a. The pinball may be located either at nodes or at sub-hex centers, described in more detail below. The choice of assembly procedure into the global force vector in the second step above is described in the following.

"Scattered" assembly simply multiplies the local pinball force $\mathbf{f}^c_a$ by the shape functions N_I as follows:

$$\mathbf{F}^c_I = \sum_a N_I(\mathbf{X}_a)\mathbf{f}^c_a \quad \forall\, I, \tag{18}$$

where $\mathbf{F}_I^c$ is the global contact force located at node I and N_I is the shape function value of node I at the material point $\mathbf{X}_a$. This method is variationally consistent in that the forces are distributed to all nodes which affect the displacement of the two pinballs in contact. For sub-hex centered pinballs, the shape functions N_a are averaged about the sub-hex, where as for nodal pinballs the shape functions at the node are used directly. A discussion of sub-hex centered and nodally placed pinballs is discussed in Section 2.3.2 below.

Alternatively, direct assembly may be leveraged, where the contact force of a pinball is directly assigned into the contact force vector at node I:

$$\mathbf{F}_I^c = \mathbf{f}_a^c, \tag{19}$$

where DOF I and pinball a are coincident. Direct assembly is only supported where nodal pinballs are used since the pinball is coincident with the nodal DOF (see Section 2.3.2).

Both the scattered and direct assembly procedures have advantages and disadvantages. In the case of the scattered approach, the pinballs forces are evenly distributed consistently with the weak form of the contact problem. Since the shape functions for RKPM lack the Kronecker delta property, multiple nodes may contribute to the displacement of each pinball and, as such, a force contribution is assembled corresponding to these DOFs in a consistent manner. A situation, however, may arise where a fracture forms between two neighboring nodes and then the nodes are displaced a large distance from one another. Pinball contact forces applied to one node could also be applied to the other node, despite being located far apart. This is a direct consequence of the Lagrangian formulation applying forces across fracture boundaries since the shape functions are unaffected by the bond-based failure. The second case using direct assembly avoids this conundrum, but at the cost of consistency with the weak form.

2.3.2 Pinball Placement

The pinball contact method roughly approximates the contact surface due to the spherical nature of the approximation. To mitigate the surface roughness introduced by the pinballs, two methods are explored regarding the placement of pinballs.

Before describing the pinball placement in detail, a brief description of the RKPM discretization is required. Typically, a tet mesh is created and used for the discretization for RKPM, sometimes referred to as a "quasi-meshfree" method. However, the tet mesh is further decomposed into its barycentric dual for use with RKPM. This is achieved with the `thex` command within Cubit which decomposes each tet into four "sub-hexes"—one for each corner of the tet. These sub-hexes are then aggregated to form the nodal domains used in the SCNI integration method [Chen et al., 2001]. Pinballs may be placed at the sub-hex centers or at the original tet nodes of the mesh as described below.

2.3.2.1 Sub-Hex Centered Pinballs The first method locates pinballs at the centroid of each sub-hex which resides on the surface to represent the domain. This is not to be confused with integration sub-cells as described in Koester et al. [2019], but rather *all surface*

sub-hexes. Each pinball's radius is sized such that the volume of the pinball is equivalent to the volume of the containing hexahedron. Using this technique guarantees some degree of overlap between neighboring pinballs, causing excessive contact forces between those pinballs. To correct for this, the RKPM pinball contact only considers those pinballs such that the distance between two pinballs in the reference configuration satisfies the following condition:

$$\|\mathbf{X}_a - \mathbf{X}_b\|^2 \geq \alpha^2 \left(R_a + R_b\right)^2, \tag{20}$$

where α is an initial proximity factor with a default of $\sqrt{2}$. The condition above serves as an adjacency check and prevents overlapping pinballs from generating spurious pinball forces.

Placing pinballs at the centers of the surface sub-hexes improves the overall surface approximation because the spheres roughly approximate the hexes they reside in. Additionally, several pinballs are instantiated for each DOF in the simulation, which have a diameter of roughly half the characteristic size, h, of the discretization. As such, pinballs placed in this manner are less likely to interpenetrate through the gaps between pinballs of the opposing surface.

During large deformation events with significant fracture, the nodes on either side of a fracture can translate far from one another while the sub-hexes between these two nodes distort heavily. The pinballs centered within these sub-hexes, whose movement is dictated by the Lagrangian shape functions, will move roughly with the sub-hex centers. More accurately, their movement will be influenced by the displacement coefficients of the nodes *on the opposite fracture surface* as described in Section 2.2.5. As a result, further separation of the fracture surfaces will cause the sub-hex pinballs to drift away from the true fracture surface, which is physically located much closer to the nodes on either side of the failed bond. This "pinball drift" causes the implicit contact surface to disassociate from the true surface in an unphysical manner. An example of this issue is described in Section 4.2.2.

2.3.2.2 Nodal Pinballs An alternative to placing pinballs at sub-hex centroids is simply to place the pinballs directly at the nodes of the RKPM simulation. In contrast with the method above, pinballs are placed only at active nodes[§] as opposed to all nodes in the analysis. The radius of each nodal pinball is sized to $0.5 l_{\min}$ where $l_{\min}$ is the minimum nodal distance to the neighbors of the RKPM node. Sizing the pinballs in this manner guarantees no overlap between adjacent neighbors but may lead to relatively small pinballs with respect to the nodal domain if the integration cell has a high aspect ratio. Using an equivalent volume strategy similar to the sub-hex centered pinballs above would avoid this but would again require a proximity factor or an initial overlap estimation.

The use of nodal pinballs ensures contact will be directly enforced at each node without the possibility of pinball drift using the sub-hex centered technique described above. However,

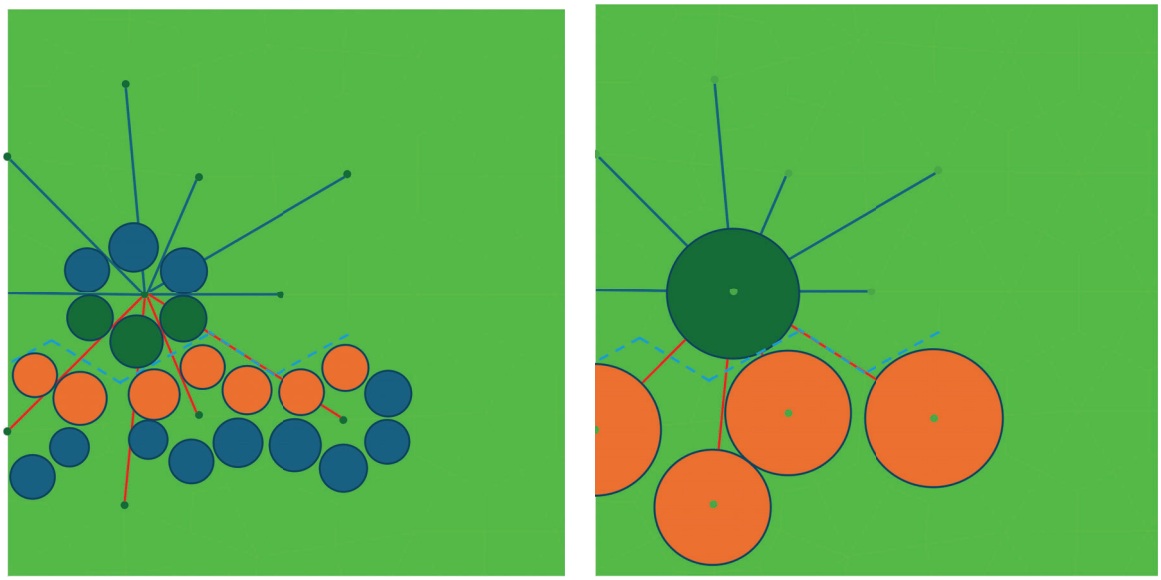
[§]Due to the use of an aggregated mesh, only those nodes of the original tet or hex-primary mesh are associated with DOFs of the simulation, or *active* nodes. The remaining points correspond to vertices of the nodal integration domains and are referred to as inactive nodes. Throughout this memo, the term "node" indicates an active node unless stated otherwise.

because RKPM nodes may reside on the surface itself, this strategy results in pinballs protruding from the surface of the domain by approximately $0.5l_{\min}$. As such, an initial overlap may occur between two separate bodies should the user fail to account for this additional protrusion. Finally, placing the pinballs at the nodes and sizing them to $0.5l_{\min}$ increases the possibility the pinballs slip past each other because the pinballs are generally larger than they would be using the sub-hex centered strategy. This increases the surface roughness and gaps between pinballs and leads to increased penetration as a result.

2.3.2.3 Pinballs at New Fracture Surfaces An enhancement to the original pinball contact in the CRK mini-app is the creation of pinballs at newly fractured surfaces as a result of bond-based fracture described in Section 2.2. This technique may be used with either the sub-hex centered pinballs or the nodal pinballs as described above. As damage accrues along a fracture zone, pinballs are inserted at the endpoints of a failed bond or in the sub-hexes of the nodes associated with the failed bond if using the nodal or sub-hex centered techniques, respectively. The new pinballs use the same sizing strategy as the techniques described above, noting that the sizes are determined using reference coordinates. This simple strategy allows fractured surfaces to re-contact in the case the overall deformation of the room leads to closure of the fracture surface.

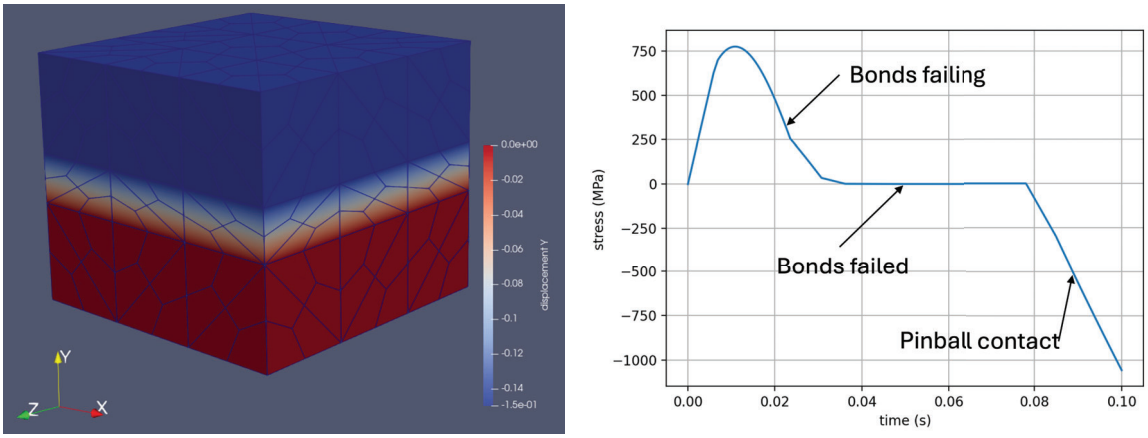
Although simple, inserting pinballs at fracture surfaces in this manner may present some negative side effects. Because the pinballs are sized with respect to the reference configuration, inserting new pinballs may lead to adjacent pinballs which overlap the moment they are spawned. As described in the enforcement, removing the initial overlap, δ, in the estimation of the gap function prevents non-physical forces from being generated. However, the contact constraint will prevent two nodes on the same side of the surface from deforming naturally should the two pinballs move towards each other. As a result, additional checks are necessary to determine whether the pinballs should be considered adjacent and subject to the proximity factor above or if the pinballs are on opposite sides of the fracture surface. This is an outstanding issue and several strategies are being pursued to mitigate against it, such as utilizing the failed bond as a "divider" to identify pinballs on opposing sides. Fig. 8 illustrates the pinball insertion procedure, which shows pinballs inserted about each node of a failed bond. Either the sub-hex centered or nodal pinball placement may be utilized at the user's discretion.

Fig. 9 demonstrates the pinball insertion capability. A block of material is initially pulled apart in tension until damage occurs. At this point, the simulation inserts pinballs at the sub-hex centroids and the two halves of the problem are pushed back toward each other. The scattered force assembly routine was employed to assemble the pinball forces into the global force vector. The new pinballs contact which leads to compressive forces in the domain simulating recontact of fracture surfaces. Note, this problem is completely prescribed such that all nodes in the lower half are displaced down and all nodes in the upper half are displaced up and then displaced back towards each other. As a result, all bonds are artificially forced to fail completely, which prevents residual forces from forming as shown in Fig. 5 above.



(a) Sub-hex centered (b) Nodal

Figure 8: Illustration of the new pinball placement after fracture. Pinballs on the upper surface of the fracture are shaded dark green whereas pinballs on the lower surface are shaded orange. The blue and red lines between small green dots represent intact and failed bonds, respectively. (a) Pinballs placed in each sub-hex surrounding the both nodes of a failed bond, indicated by a red line connecting two RKPM nodes. (b) Pinballs placed at each RKPM node.



(a) Pinball recontact problem setup. (b) Force-displacement response of recontact problem.

Figure 9: Pinball "recontact" problem in which a block is initially pulled apart until fracture initiates. After the fracture event, the two halves of the block are pushed back toward each other where the pinball contact forces induce compressive stresses.

3 Salt Material Model Development

This section describes the development of a new simple salt material model used in the room collapse simulations. (Another, more sophisticated, model is being developed to more accurately represent salt's behavior [Reedlunn, 2022a], but it is not ready to be used in room collapse simulations yet, so it is not discussed any further herein.) Section 3.1 motivates the development of the new model by describing several challenges with numerically integrating the old salt material model. Sections 3.2 and 3.3 cover the new model and the differences between the old and new models. Section 3.4 presents a parameter set for the new model and a selection of different behaviors with this parameter set. Sections 3.5 and 3.6 discuss the new model implementation and a verification test. Finally, Section 3.7 shows the results of convergence studies with the new model implementation.

3.1 Difficulties With the Old Material Model

Reedlunn et al. [2019, Section 2.2.3.1] defined the salt material model used in the simulations therein[¶]. This model was an isotropic, hypoelastic, viscoplastic, continuum damage model. It was designed to capture the essential physical processes, yet remain relatively simple for numerical method assessment. Accurately capturing salt's true behavior was not of paramount concern.

The first Sierra/SM implementation of this relatively simple model used an explicit integration scheme. The derivatives in the damage and viscoplastic strain evolution equations were replaced with forward Euler approximations, and the resulting system of non-linear algebraic equations were explicitly solved with sub-stepping. The sub-step size was based on Cormeau [1975, Eqn. 53], and the user could specify a maximum number of sub-steps. Cormeau [1975] derived his sub-step size for a fully-hardened viscoplastic material, without any damage, so we attempted ad-hoc modifications to account for damage. This explicit integration scheme was adequate for pure viscoplastic deformations, as well as viscoplastic deformations that included a moderate degree of gradually evolving damage. Attempts to simulate room collapse would predict bands of localized damage and deformation around the room, but the simulation would inevitably fail before a roof fall could occur. Typically, a material point reached the maximum number of sub-steps, killing the simulation, or the simulation would grind to a halt as it waited for a material point to go through an excessive number of sub-steps. If damage accumulated too quickly or became too large, the explicit integration scheme would fail. Presumably these failures corresponded to material points within the localized bands of high damage and deformation. We considered attempting to rigorously derive a sub-step size that took damage into account, but chose to pursue implicit integration instead. This choice was later confirmed upon discovering that Lemaitre and Desmorat [2006, Section 2.2.3] says that explicit schemes may be used in simulations

[¶]Note that Reedlunn et al. [2019, Eqn. 2.10] incorrectly defined the damage boundary as $\hat{\bar{\sigma}}^{\mathrm{db}} = B_2\,(\hat{p} - B_1)$, where B_1 and B_2 are material parameters, $\hat{p} = p/(1-\delta)$ is the Kachanov pressure, and δ is the damage. The damage boundary was actually implemented as $\hat{\bar{\sigma}}^{\mathrm{db}} = B_2\,(\hat{p} - B_1/(1-\delta))$, which allows one to write $\hat{\bar{\sigma}}^{\mathrm{db}}\,(1-\delta) = \bar{\sigma}^{\mathrm{db}} = B_2\,(p - B_1)$, as was done in the analytical solution in Reedlunn et al. [2019, Section 2.2.3.3].

that involve small amounts of damage, yet they advise their readers, "do not use explicit schemes for states of damage close to the occurrence of strain localization as instabilities are expected."

The second Sierra/SM implementation of the model used an implicit scheme. The derivatives were replaced with backward Euler approximations, and the resulting system of non-linear algebraic equations were implicitly solved using one of two methods. The scheme first attempted to obtain a second-order accurate solution with the Richardson extrapolated semi-implicit backward Euler method, as described in Hofstetter [2014, Chapter 1]. If the error associated with semi-implicit integration was too large, then the scheme would attempt to use the fully-implicit backward Euler method, in which Newton's method with a line search algorithm would attempt to solve the system of non-linear equations in a manner similar to that in Reedlunn [2018, Chapter 3]. This implicit scheme, unfortunately, was only marginally more robust than the original explicit scheme when used in room collapse simulations. Instead of failing due to too many sub-steps, the backward Euler scheme would fail to converge when the damage and strain heavily localized.

Studies of the implicit scheme's convergence behavior were performed. In these studies the material model was subjected to axisymmetric stress states, where the axial strain rate $\dot{\varepsilon}_{zz}$ and the axial strain increment $\Delta\varepsilon_{zz}$ were varied over wide ranges. When semi-implicit integration failed and fully-implicit integration was required, the robustness of Newton's method was found to be sensitive to the initial guess for the unknown variables. When damage was suppressed by high radial confining pressures, an initial guess that assumed the material was fully hardened converged in all cases studied, which is consistent with the studies in Schreyer [2002]. When damage was permitted to evolve at low radial confining pressures, the implicit scheme converged more often for higher quality initial guesses, but still struggled to converge in certain cases.

Examining the convergence failures revealed three different issues:

1. The first issue was connected to assigning a different principal damage to each principal stress (see Reedlunn et al. [2019, Eqn. 2.2 and 2.3]). The damage limit for a principal damage depended on whether the corresponding principal stress was compressive or tensile. The compressive damage limit δ^{c} was intended to be smaller than the tensile damage limit δ^{t}. A minor issue with this formulation was the implementation had to recognize when the eigen solver changed the ordering of the principal stresses. A major issue was the principal stress could switch sign, which could cause a principal damage to suddenly jump from the tensile damage limit to the compressive damage limit. The implicit scheme would usually fail to handle this jump.

2. The second issue was the damage limits were enforced, as the differential equations were being integrated, by simply setting the damage equal to the damage limit when the damage exceeded the damage limit. This approach was sufficient for the previous explicit sub-stepping scheme, since the damage would evolve small amounts within each sub-step, but led to problems with the implicit scheme, where a large strain increment

could cause the damage to significantly exceed the damage limit for a given Newton solver iteration. The implicit scheme would continue to iterate to attempt to return the damage to the damage limit, but the discontinuity in the damage evolution rate at the damage limit and/or damage exceeding 1 would often cause the implicit scheme to fail.

3. The third issue occurred at high strain rates ($\dot{\varepsilon}_{zz} > 10^{-4}$ 1/s) and low confining pressure ($\sigma_{rr} < 5$ MPa). These high strain rates were not common in room collapse simulations, but they did occur right as the strain was localizing within a damage band.

   Material point simulations under constant strain rate, axisymmetric compression, and zero radial confining pressure were performed at low and high strain rates to investigate the convergence failures. The Collapse parameter set in Reedlunn et al. [2019, Table 2-1] was utilized, except the compressive and tensile damage limits were set to $\delta^c = \delta^t = 0.9$. Responses from two of these simulations are shown in Fig. 10, where σ_{zz} is the axial stress, δ_3 is the principal damage corresponding to the axial stress (the most compressive principal stress), $\dot{\varepsilon}_{zz}$ is the axial strain rate, $\dot{\varepsilon}_{zz}^{el}$ is the elastic axial strain rate, $\dot{\varepsilon}_{zz}^{vp}$ is the viscoplastic axial strain rate, and ε_{zz} is the axial strain.

   At a medium strain rate of $\dot{\varepsilon}_{zz} = 10^{-8}$ 1/s in Fig. 10a, the stress exceeds the damage boundary and causes viscoplastic flow to occur. Damage gradually develops because the stress and viscoplastic flow rate are moderate (see Reedlunn et al. [2019, Eqn. 2.11]). Damage causes the stress to gradually drop until the damage limit is hit.

   At a high strain rate of $\dot{\varepsilon}_{zz} = 10^{0}$ 1/s in Fig. 10b, the stress exceeds the damage boundary, but viscoplastic flow becomes significant at a much higher stress (see Reedlunn et al. [2019, Eqn. 2.9]). The high stress and large viscoplastic flow rate cause damage to develop quickly. The high damage rate causes the stress to plummet over a very small total strain increment until the damage limit is hit.

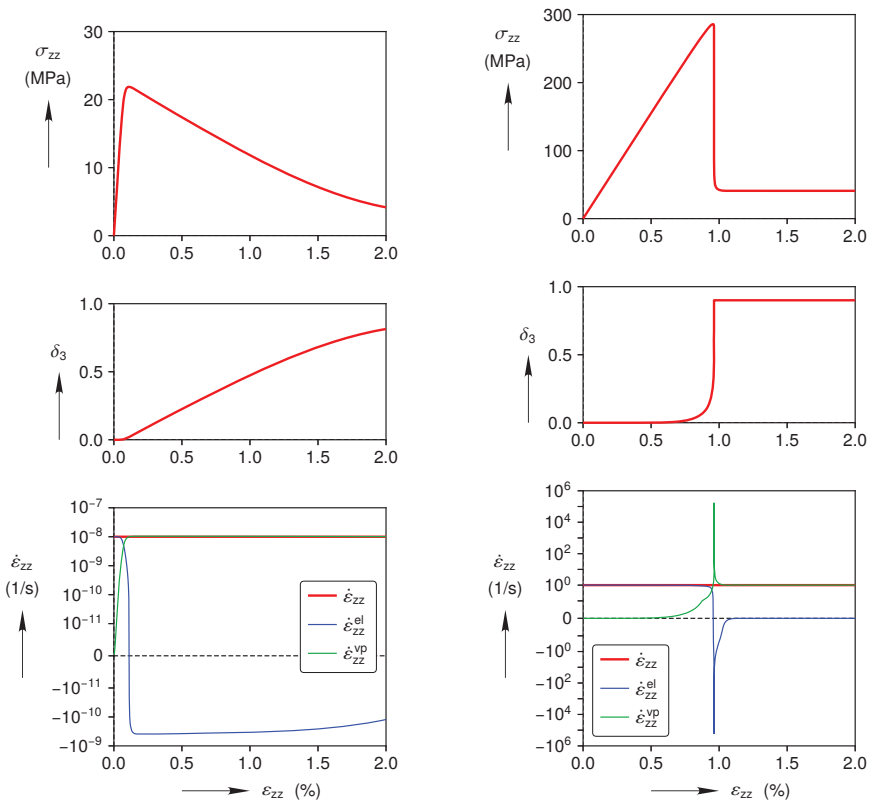
   The stress drop in Fig. 10b is, perhaps, an unexpected result. One might expect the stress drop to reduce the damage evolution rate, which it does, but the stress drop also causes a large negative elastic strain rate. The total strain rate is fixed, so the viscoplastic strain rate shoots upward to compensate for the negative elastic strain rate. The huge viscoplastic strain rate causes the damage to evolve rapidly, despite the smaller stress.

   Not surprisingly, the implicit integration scheme frequently struggled to handle these nearly instantaneous changes to the stress and damage at high strain rates.

Various modifications to the old model were explored to resolve these issues. The more successful modifications were collected together to form the basis for the new salt material model described in the following section.

3.2 New Model Formulation

The new material model remains a relatively simple isotropic, hypoelastic, viscoplastic, continuum damage model. This section presents the new model stripped down to its essential



(a) $\dot{\varepsilon}_{zz} = 10^{-8}$ 1/s (b) $\dot{\varepsilon}_{zz} = 10^{0}$ 1/s

Figure 10: Uniaxial compression response of the old model at a low strain rate and a high strain rate.

components. See Section A.1 for how the model was constructed within the framework of Rational Thermodynamics. See Section 3.3 for a discussion of the differences between the old and new model. The following presentation follows the geomechanics convention that compressive stresses are positive and tensile stresses are negative. Uppercase symbols denote material parameters, while lowercase symbols denote functions of other variables.

The formulation begins with a series of common definitions and decompositions of the stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$. Define the spherical (mean) stress scalar and spherical stress tensor as

$$\mathring{\sigma} = \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\sigma}), \tag{21}$$

$$\mathring{\boldsymbol{\sigma}} = \mathring{\sigma}\,\boldsymbol{I}, \tag{22}$$

respectively. Decompose the stress as

$$\boldsymbol{\sigma} = \mathring{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\sigma}}, \tag{23}$$

where $\tilde{\boldsymbol{\sigma}}$ is the deviatoric stress. Define the equivalent (deviatoric) stress $\tilde{\sigma}$ as

$$\tilde{\sigma} = \sqrt{\frac{3}{2}\,\tilde{\boldsymbol{\sigma}} : \tilde{\boldsymbol{\sigma}}}. \tag{24}$$

Define the spherical strain scalar and spherical strain tensor as

$$\mathring{\varepsilon} = \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\varepsilon}), \tag{25}$$

$$\mathring{\boldsymbol{\varepsilon}} = \mathring{\varepsilon}\,\boldsymbol{I}, \tag{26}$$

respectively. Decompose the strain as

$$\boldsymbol{\varepsilon} = \mathring{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}}, \tag{27}$$

where $\tilde{\boldsymbol{\varepsilon}}$ is the deviatoric strain tensor. Finally, decompose the strain into an elastic strain $\boldsymbol{\varepsilon}^{\mathrm{el}}$ and a viscoplastic strain $\boldsymbol{\varepsilon}^{\mathrm{vp}}$ as

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathrm{el}} + \boldsymbol{\varepsilon}^{\mathrm{vp}}, \tag{28}$$

where $\mathrm{tr}(\boldsymbol{\varepsilon}^{\mathrm{vp}}) = 0$. Dilation strain associated with damage (micro-cracking) is neglected for simplicity.

The model's elastic behavior follows generalized Hooke's law, modified to include degradation of the shear modulus. The elastic constitutive relationships are

$$\mathring{\boldsymbol{\sigma}} = 3\,E_1\,\mathring{\boldsymbol{\varepsilon}}, \tag{29}$$

$$\tilde{\boldsymbol{\sigma}} = 2\,(1-\delta)\,E_2\,(\tilde{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon}^{\mathrm{vp}}), \tag{30}$$

where $E_1 \geq 0$ and $E_2 \geq 0$ are the undamaged bulk and shear moduli, respectively, $0 \leq \delta < D_5$ is the damage, and $0 \leq D_5 < 1$ is the damage limit. The damaged shear modulus is $e_2 = (1-\delta)\,E_2$, while the bulk modulus is not degraded.

The viscoplastic portion of the model is designed to capture secondary and tertiary creep, but not primary creep. The viscoplastic strain evolves according to an associated flow rule

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = \dot{\tilde{\varepsilon}}^{\mathrm{vp}}\,\frac{\partial \tilde{\sigma}}{\partial \boldsymbol{\sigma}}, \tag{31}$$

where $\dot{\tilde{\varepsilon}}^{\mathrm{vp}}$ is the equivalent viscoplastic strain rate. This strain rate is decomposed as

$$\dot{\tilde{\varepsilon}}^{\mathrm{vp}} = \dot{\tilde{\varepsilon}}^{\mathrm{ps}} + \dot{\tilde{\varepsilon}}^{\mathrm{dg}}, \tag{32}$$

where $\dot{\tilde{\varepsilon}}^{\mathrm{ps}}$ is the equivalent pressure solution strain rate and $\dot{\tilde{\varepsilon}}^{\mathrm{dg}}$ is the equivalent dislocation glide strain rate. These viscoplastic strain rates are defined as

$$\dot{\tilde{\varepsilon}}^{\mathrm{ps}} = \frac{P_1}{(1-\delta)^{1/2}}\,\left(\frac{\tilde{\sigma}}{(1-\delta)^{1/2}}\right)^{P_2}, \tag{33}$$

$$\dot{\tilde{\varepsilon}}^{\mathrm{dg}} = \frac{G_1}{(1-\delta)^{1/2}}\,\left\{\sinh\left[\frac{\tilde{\sigma}}{G_2\,(1-\delta)^{1/2}}\right]\right\}^{G_3}, \tag{34}$$

where $P_j \geq 0$ and $G_j \geq 0$ are material constants. If δ and $\tilde{\sigma}$ remain constant, then these expressions give secondary creep behavior. If $\delta = 0$ and $P_2 = 1$, each relationship between stress and strain rate matches common secondary creep relations [Spiers et al., 1990, Garofalo, 1963] for isothermal conditions. If δ grows while $\tilde{\sigma}$ is held constant, these expressions give tertiary creep behavior.

The damage evolution law is defined as

$$\dot{\delta} = \frac{D_1}{1-\delta} \left\langle \tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \left[1 - \left(\frac{\delta}{D_5} \right)^{D_6} \right], \tag{35}$$

$$\tilde{\sigma}^{\,\mathrm{db}} = D_2 \left(\mathring{\sigma} - D_3 \right) \frac{1-\delta}{1 - D_4\,\delta}. \tag{36}$$

where $D_1 \geq 0$, D_2, D_3, $0 \leq D_4 \leq 1$, and $D_6 \geq 0$ are material constants; $\tilde{\sigma}^{\,\mathrm{db}}$ is the damage boundary; and the angled brackets denote $\langle x \rangle = (x + |x|)/2$. Damage only evolves if $\tilde{\sigma}$ is above $\tilde{\sigma}^{\,\mathrm{db}}$, $\dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} > 0$, and $\delta < D_5$. One can think of $\left\langle \tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}}$ as the damage work rate.

3.3 Discussion of Formulation Changes

The old and new models are not dramatically different, but significant differences exist. These differences are:

1. A single isotropic damage variable is employed, instead of the three principal damages in the old formulation, to address Issue 1 in Section 3.1. The new formulation does not include different damage limits for tension and compression principal stresses, but this feature may not be as important as previously thought. Damage can cause $\tilde{\sigma}$ to drop below $\tilde{\sigma}^{\mathrm{db}}$, shutting down damage evolution for $\delta < D_5$, as will be shown in Fig. 12b. Compressive spherical stress increases $\tilde{\sigma}^{\mathrm{db}}$ (presuming $D_2 > 0$), causing $\tilde{\sigma} - \tilde{\sigma}^{\mathrm{db}} = \dot{\delta} = 0$ at smaller δ and at larger $\tilde{\sigma}$.

2. The damage boundary includes an extra parameter D_4 that affects how the boundary evolves with damage. This feature gives the user some control over when the damage evolution halts at a given mean stress if $\delta < D_5$. For example, if a constant axial strain rate of $\dot{\varepsilon}_{\mathrm{zz}}$ is applied in an axisymmetric compression test, then one can combine $\tilde{\sigma} = \tilde{\sigma}^{\mathrm{db}}$, $\dot{\varepsilon}_{\mathrm{zz}} \approx \dot{\tilde{\varepsilon}}^{\mathrm{vp}}$, Eqs. (32) to (34) and solve the resulting implicit algebraic equation for δ. This will give the δ when $\dot{\delta} = 0$ for a given set of material parameters (including D_4), $\mathring{\sigma}$, and $\dot{\varepsilon}_{\mathrm{zz}}$.

3. The factor of $1 - (\delta/D_5)^{D_6}$ in Eq. (35) gradually slows down the damage evolution as δ nears D_5, instead of abruptly halting the damage evolution at D_5, to address Issue 2 in Section 3.1. Smaller values of $D_6 > 0$ cause the damage rate to slow down more gradually.

4. The elastic shear modulus degrades with damage, instead of being a constant, to address Issue 3 in Section 3.1. In a constant strain rate test, where the stress and

shear modulus both drop due to damage, the elastic strain decrease associated with the stress drop competes with the elastic strain increase associated with the shear modulus decrease. In all cases studied thus far, the elastic strain increase is larger, such that the elastic strain rate remains positive, the viscoplastic strain rate remains small, the damage rate remains small, and the stress drops gradually, as will be shown in Fig. 12c.

Degrading the shear modulus, without degrading the bulk modulus, is a compromise between accuracy and simplicity. Compressive stresses normal to a crack plane tend to close the crack, causing no further cracking, while tensile stresses normal to the crack plane tend to open the crack, causing cracks to grow. Shear is a combination of tension and compression in the principal frame, so the model degrades the shear modulus with damage. The compressive hydrostatic response should be relatively insensitive to cracks and tensile hydrostatic stresses are very rare compared to compressive hydrostatic stresses in geomechanical analyses, so the model does not degrade the bulk modulus. More sophisticated attempts to include the differences between compression and tension typically lead to complex models that decompose the principal stresses (or strains) into positive and negative terms (see Murakami [2012, Section 4.3.4, 9.2, and 9.3], for example). Such complications do not appear worthwhile at this stage.

5. The damage evolves with the dislocation glide strain, rather than the sum of the pressure solution and dislocation glide strain. Pressure solution is thought to be associated with healing of micro-cracks, rather than generating micro-cracks, so the damage should not evolve with pressure solution strain.

6. Each viscoplastic strain rate in Eqs. (33) and (34) contains $(1-\delta)^{1/2}$ in two locations, instead of containing $1-\delta$ in one location. This is due to following the procedure in Section A.1, rather than simply adopting an equivalence of strains between the fictitious undamaged configuration and the actual damaged configuration.

7. The dislocation glide strain rate uses the Garofalo [1963] expression, which includes a hyperbolic sine raised to a power, instead of a simple power law expression. The hyperbolic sine elegantly captures the breakdown of power-law steady-state creep at medium stresses to exponential steady-state creep at high stresses. Power-law breakdown (PLB) commonly occurs in metals and artificial salt (Sherby and Burke [1968, Fig. 32], Heard [1972, Fig. 11], and Raj and Pharr [1989, Fig. 2]), so it is reasonable to assume natural rock salt exhibits PLB as well.

3.4 Parameters Sets and Selected Model Behaviors

Two parameter sets are listed in Table 1: a Verification parameter set used in the verification test shown in Section 3.6 and a Collapse parameter set used in the room collapse and reconsolidation simulations shown in Section 4.

The Verification parameters were selected with three purposes in mind. First and foremost, the parameters had to enable an analytical solution to the verification scenario. Second, the

Table 1: New salt model parameter values.

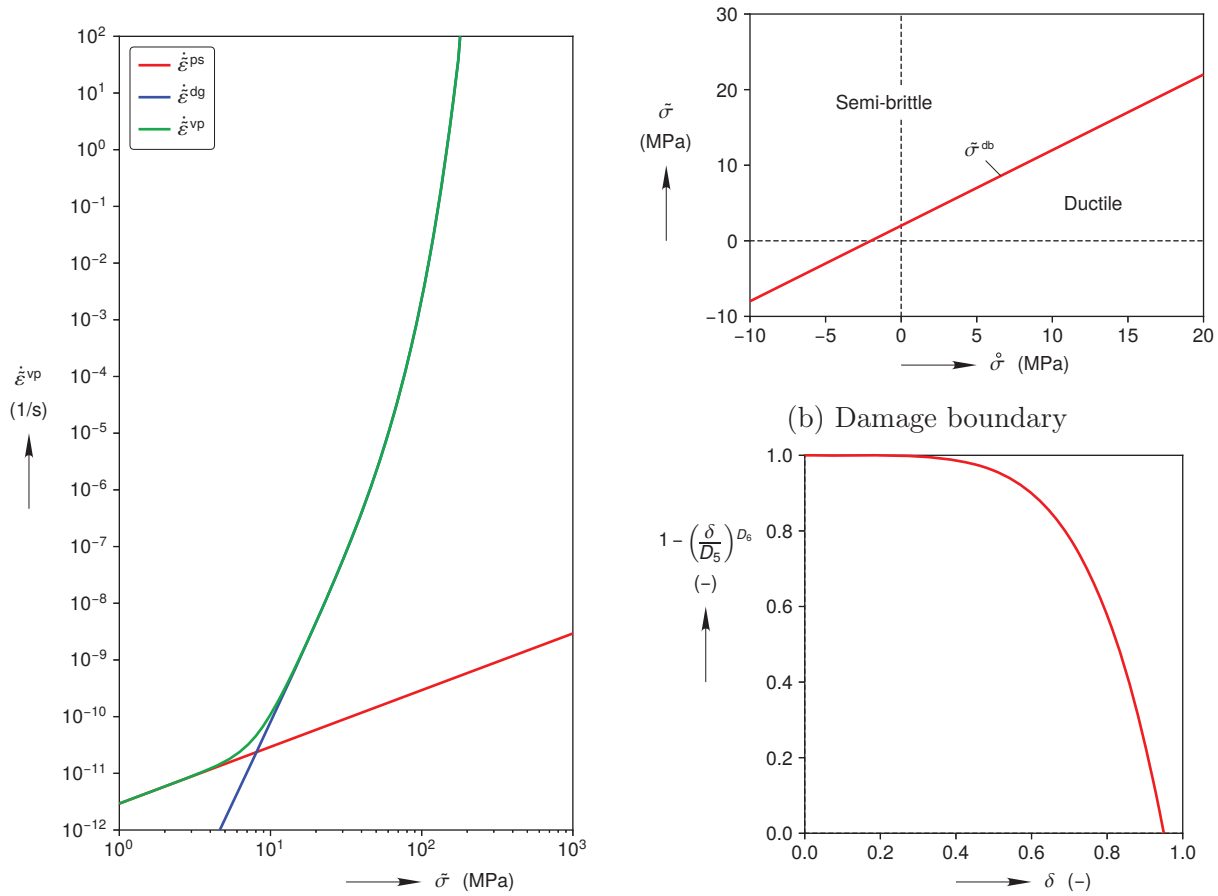
Parameter Type	Parameter	Units	Verification	Collapse
Elastic	E_1	GPa	20.7	20.7
	E_2	GPa	12.4	12.4
Viscoplastic	P_1	s^{-1}	0	2.930×10^{-18}
	P_2	–	1	1
	G_1	s^{-1}	7.8	2.959×10^{-7}
	G_2	MPa	10^3	43.50
	G_3	–	5	5.623
Damage	D_1	1/Pa	8×10^{-6}	8×10^{-6}
	D_2	–	1	1
	D_3	MPa	-2	-2
	D_4	–	1	1
	D_5	–	0.5	0.95
	D_6	–	100	5

parameters needed to result in simulation results that could easily be plotted and compared against the analytical solution both before and after the accumulation of substantial damage. Third, where convenient, values similar to those in the Collapse parameter set were chosen.

The Collapse parameter set was chosen in the following manner. The elastic and viscoplastic parameters were chosen based on the elastic and steady-state creep parameters in Reedlunn [2022a, Table 1]. Figure 11a depicts the steady-state creep behavior resulting from these viscoplastic parameters. The damage evolution rate D_1 was chosen to ensure the room roof fell within a few years of room excavation. The damage boundary parameters D_2 and D_3 were chosen to loosely capture the dilatancy boundary for WIPP salt [Düsterloh et al., 2015]. Fig. 11b shows the damage boundary corresponding to these parameter values. The damage parameter $D_4 = 1$ was selected because $D_4 = 0$ caused damage to spread through the entire pillar next to the room in a preliminary room collapse simulation. The damage limit D_5 and damage limit exponent D_6 were selected to allow the stress to smoothly approach a fairly small value prior to failure. Fig. 11c presents how the damage limit rate factor $1 - (\delta/D_5)^{D_6}$ varies with δ. Accurate numerical predictions of room collapse and rubble pile consolidation will eventually require higher quality parameter selection methods, but the Collapse parameters in Table 1 are thought to suffice for the assessments pursued herein.

Figure 12 shows the results of several material point simulations using the Collapse parameter set. Each simulation began by applying a spherical stress of $\mathring{\sigma} = \sigma_{\mathrm{rr}} = \sigma_{\mathrm{zz}}$ that induced an axial strain of $\varepsilon_{\mathrm{zz}}(t_0)$ at time t_0. Then, a constant axial strain rate $\dot{\varepsilon}_{\mathrm{zz}}$ was applied while holding σ_{rr} fixed, monitoring the equivalent stress $\tilde{\sigma} = |\sigma_{\mathrm{zz}} - \sigma_{\mathrm{rr}}|$, and monitoring the change in axial strain $\varepsilon_{\mathrm{zz}} - \varepsilon_{\mathrm{zz}}(t_0)$.

Figure 12a depicts mechanical responses for several strain rates when the radial confining pressure is high enough to suppress damage. The lack of hardening in the model causes $\tilde{\sigma}$



(a) Steady-state creep for $\delta = 0$

(b) Damage boundary

(c) Damage limit rate factor

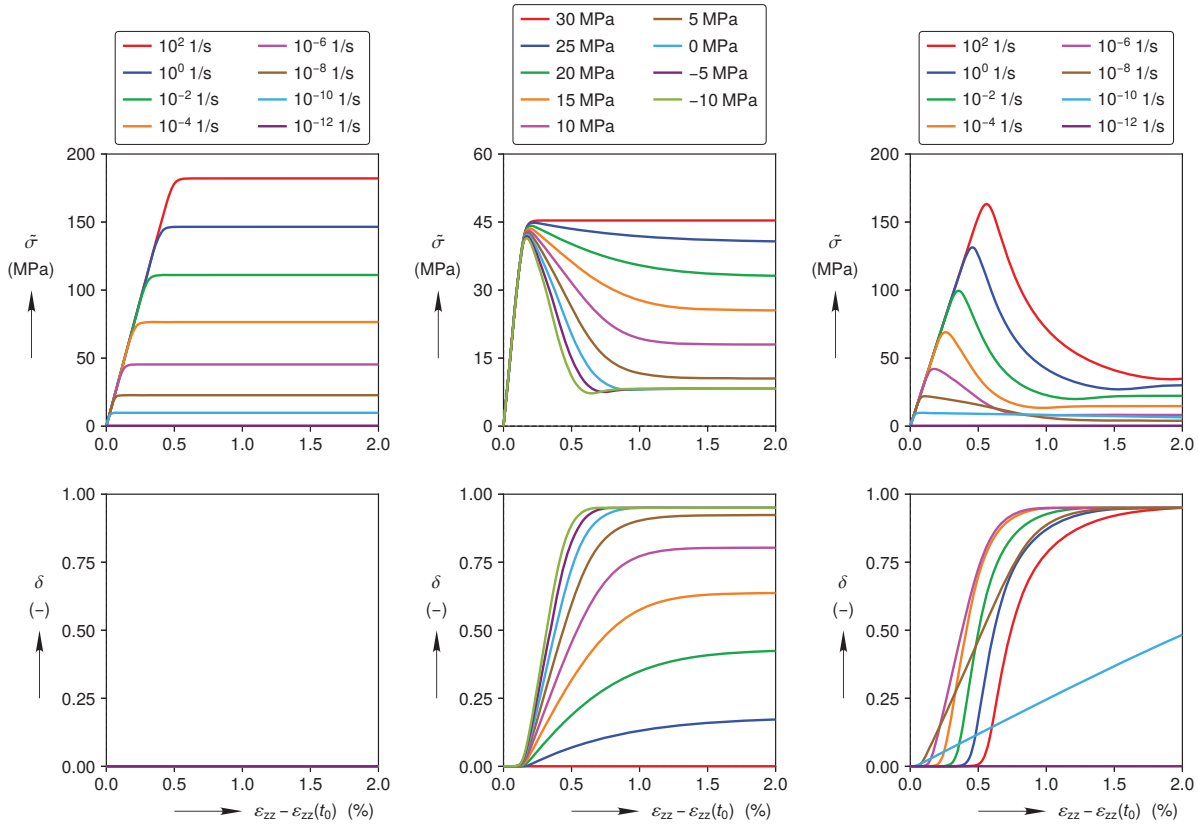
Figure 11: Steady-state creep, damage boundary, and damage limit rate factor for the Collapse parameter set in Table 1

to remain flat once fully developed plastic flow occurs.

Figure 12b presents the mechanical responses and damage evolutions for various σ_{rr}, all at $\dot{\varepsilon}_{zz} = 10^{-6}$ 1/s. Damage evolution is suppressed for $\sigma_{rr} = 30$ MPa. Reducing the radial confining stress to $\sigma_{rr} \leq 25$ MPa, however, reduces $\tilde{\sigma}^{db}$ and permits damage to accumulate. For 5 MPa $\leq \sigma_{rr} \leq 25$ MPa, $\tilde{\sigma}$ drops as damage develops, until $\tilde{\sigma}$ drops below $\tilde{\sigma}^{db}$. The residual stress, at which $\delta > 0$ and $\dot{\delta} = \dot{\tilde{\sigma}} - \dot{\tilde{\sigma}}^{db} = 0$, depends on σ_{rr}. For $\sigma_{rr} \leq 0$ MPa, $\tilde{\sigma}$ drops as damage develops, until $\delta = D_5$. The small dip in $\tilde{\sigma}$ at 0.5 % $< \varepsilon_{zz} - \varepsilon_{zz}(t_0) <$ 1 % for $\sigma_{rr} \leq 0$ MPa occurs because $\dot{\tilde{\varepsilon}}^{vp} < \dot{\tilde{\varepsilon}}^{el}$ for a period of time. The equivalent elastic strain rate is defined as $\dot{\tilde{\varepsilon}}^{el} = \sqrt{2/3\, \dot{\tilde{\boldsymbol{\varepsilon}}}^{el} : \dot{\tilde{\boldsymbol{\varepsilon}}}^{el}}$, where $\tilde{\boldsymbol{\varepsilon}}^{el} = \boldsymbol{\varepsilon}^{el} - \mathrm{tr}(\boldsymbol{\varepsilon}^{el})\boldsymbol{I}$.

Figure 12c shows the mechanical responses and damage evolutions for various $\dot{\varepsilon}_{zz}$, all at $\sigma_{rr} = 0$. Without a confining pressure, only the slowest strain rate of $\dot{\varepsilon}_{zz} = 10^{-12}$ 1/s avoids any damage. Higher strain rates lead to higher $\tilde{\sigma}$ above $\tilde{\sigma}^{db}$, which induces damage. The damage causes the stress to decay, but, unlike the old model, the stress decays gradually for

all strain rates considered.



(a) $\sigma_{rr} = 1$ GPa, vary $\dot{\varepsilon}_{zz}$ (b) Vary σ_{rr}, $\dot{\varepsilon}_{zz} = 10^{-6}$ 1/s (c) $\sigma_{rr} = 0$ MPa, vary $\dot{\varepsilon}_{zz}$

Figure 12: Stress versus strain curves at various strain rates and radial confining pressures for the Collapse parameter set in Table 1.

3.5 Implementation

The new model implementation uses a similar implicit solution scheme as the second implementation of the old model discussed in Section 3.1. This scheme first tries to use a similar Richardson extrapolated semi-implicit backward Euler method, and, if necessary, falls back on a similar fully-implicit backward Euler method. The scheme also uses the same solution convergence metric. The scheme, however, differs in two significant ways, which are reviewed after the solution convergence metric.

The solution convergence metric is computed as follows. The model's differential equations can be expressed in standard form as

$$\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z}) \tag{37}$$

where $\boldsymbol{z}$ is the vector of unknowns and $\boldsymbol{f}(\boldsymbol{z})$ is a non-linear function of $\boldsymbol{z}$. Discretizing Eq. (37) with backwards Euler approximations and rearranging gives the expression for the

residual vector:

$$\boldsymbol{r}_n = -(\boldsymbol{z}_n - \boldsymbol{z}_{n-1}) + \boldsymbol{f}(\boldsymbol{z}_n)\,\Delta t, \tag{38}$$

where $\boldsymbol{z}_n$ is the vector of unknowns at the current step n and $\Delta t = t_n - t_{n-1}$. Introduce the notation that the j^{th} component of $\boldsymbol{x}_n$ is $x_{j,n}$. Each component of a normalized residual vector $\boldsymbol{\omega}_n$ is computed as

$$\omega_{j,n} = \frac{r_{j,n}}{A_j^{\text{tol}} + R^{\text{tol}}\, z_{j,n-1}} \tag{39}$$

where A_j^{tol} is the absolute error tolerance for the j^{th} component and R^{tol} is the relative error tolerance. The simulations herein utilized $A_j^{\text{tol}} = E_2 \times 10^{-8}$ for stress-type quantities, $A_j^{\text{tol}} = 10^{-7}$ for quantities on the order of unity, and $R^{\text{tol}} = 10^{-4}$. This normalized residual vector is converted into a scalar as

$$\omega_n = \sqrt{\frac{\boldsymbol{\omega}_n \cdot \boldsymbol{\omega}_n}{J}} \tag{40}$$

where J is the number of components in the vector $\boldsymbol{z}_n$. If $\omega_n \leq 1$ then a solution $\boldsymbol{z}_n$ is accepted, otherwise the implicit solution scheme keeps attempting to reduce ω_n, provided number of fully implicit iterations k is less than the maximum $k_{\max} = 100$. With the solution convergence metric ω defined, the differences between the old implicit solution scheme and the new implicit solution scheme can be presented.

The first difference is the old implicit scheme for the old model solved a system of seven equations for $\boldsymbol{z}_n = \{\sigma_{1,n},\ \sigma_{2,n},\ \sigma_{3,n},\ \sigma_{4,n},\ \sigma_{5,n},\ \sigma_{6,n},\ \delta_n\}$, where $\sigma_{j,n}$ is the j^{th} component of the stress $\boldsymbol{\sigma}_n$ and δ_n is the damage before applying the damage limits δ^{c} and δ^{t}, while the new implicit scheme for the new model solves a system of two equations for $\boldsymbol{z}_n = \{\tilde{\sigma}_n,\ \delta_n\}$. This reduction was made possible by extending the radial return algorithm to include damage, as detailed in Section A.3. Solving two equations instead of seven has three advantages:

1. Solving two equations is more computationally efficient.
2. Solving for $\tilde{\sigma}_n$ instead of $\boldsymbol{\sigma}_n$ is more robust. Preliminary convergence studies with the seven equation implementation showed that the solver could get stuck in a local minimum of $\omega_n > 1$ where the components $\sigma_{j,n}$ did not fully satisfy the stress component equations, yet they corresponded to the $\tilde{\sigma}_n$ that satisfied the damage evolution equation. Switching to the two equation implementation eliminated this issue.
3. Two equations and two unknowns are easier to troubleshoot. One can easily visualize each iteration of the solver as it tries to drive $\omega_n < 1$ when one is dealing with only two unknowns, as will be shown in Section 3.7.

The second difference between the old and new implicit solution schemes concerns the initial guess for the fully-implicit backward Euler method. Preliminary convergence studies found the robustness of the implementation was sensitive to this initial guess. Experimentation led to the procedure to determine the initial guess detailed in Section A.4.

3.6 A Verification Test

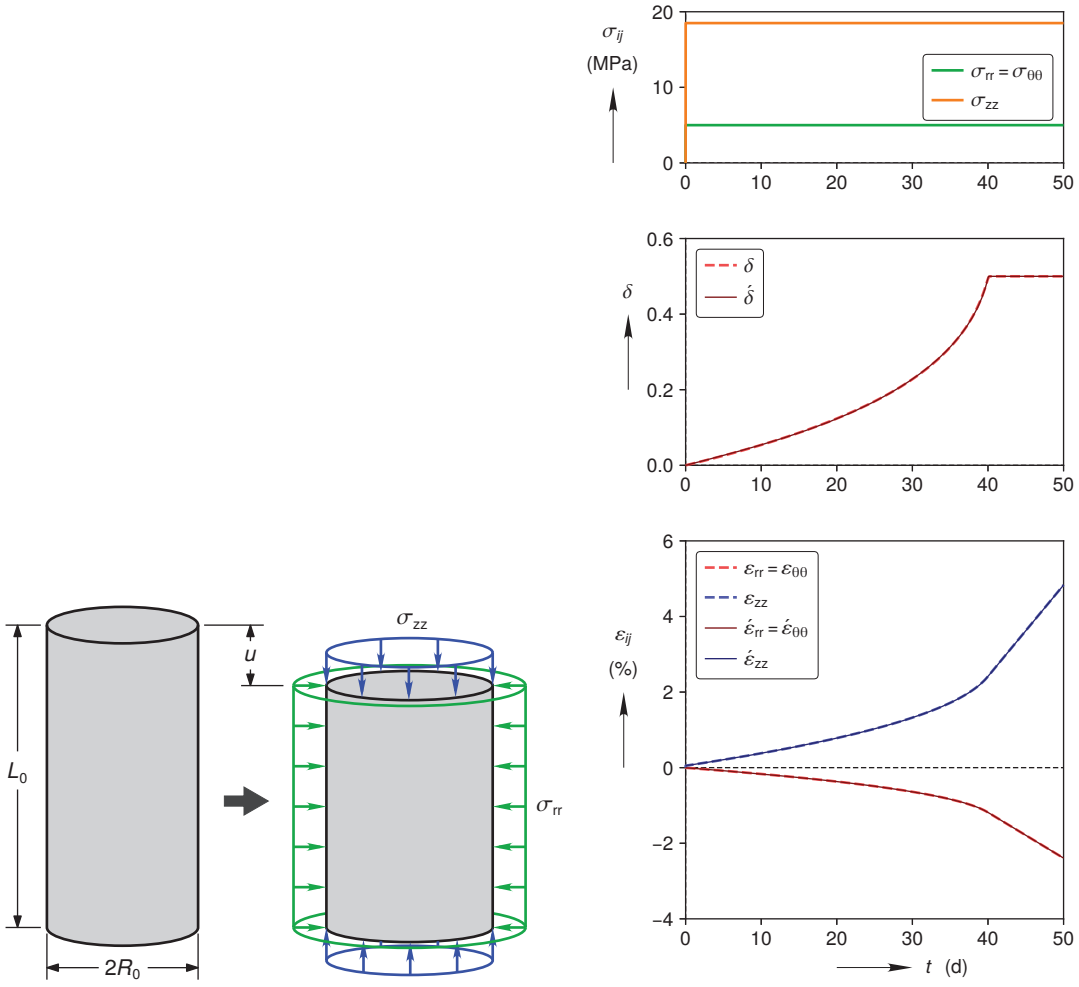
The prevalence and simplicity of axisymmetric compression creep make it well suited as a material model verification test. In axisymmetric compression creep, a cylindrical specimen is subjected to a constant Cauchy stress of

$$\boldsymbol{\sigma} = \sigma_{\mathrm{rr}}\, \boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} + \sigma_{\mathrm{rr}}\, \boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \sigma_{\mathrm{zz}}\, \boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}}, \tag{41}$$

as shown in Fig. 13a, while measuring the corresponding logarithmic strain

$$\boldsymbol{\varepsilon} = \varepsilon_{\mathrm{rr}}\, \boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} + \varepsilon_{\mathrm{rr}}\, \boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \varepsilon_{\mathrm{zz}}\, \boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}}, \tag{42}$$

where $(\boldsymbol{e}_{\mathrm{r}},\ \boldsymbol{e}_{\theta},\ \boldsymbol{e}_{\mathrm{z}})$, $(\sigma_{\mathrm{rr}},\ \sigma_{\theta\theta},\ \sigma_{\mathrm{zz}})$, and $(\varepsilon_{\mathrm{rr}},\ \varepsilon_{\theta\theta},\ \varepsilon_{\mathrm{zz}})$ are unit vectors, stress components, and strain components respectively aligned with the radial, hoop, and axial directions. The axisymmetry causes $\sigma_{\theta\theta} = \sigma_{\mathrm{rr}}$ and $\varepsilon_{\theta\theta} = \varepsilon_{\mathrm{rr}}$.



(a) Schematic (b) Numeric vs. Analytical

Figure 13: Axisymmetric compression creep verification test.

The model in Section 3.2 affords an analytical solution under axisymmetric compression creep if the Verification parameter set in Table 1 is utilized. The stress state can be decomposed into spherical and deviatoric parts as

$$\mathring{\boldsymbol{\sigma}} = \frac{2\,\sigma_{\mathrm{rr}} + \sigma_{\mathrm{zz}}}{3}\,\boldsymbol{I} \tag{43}$$

$$\tilde{\boldsymbol{\sigma}} = -\frac{\sigma_{\mathrm{zz}} - \sigma_{\mathrm{rr}}}{3}\,\boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} - \frac{\sigma_{\mathrm{zz}} - \sigma_{\mathrm{rr}}}{3}\,\boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \frac{2(\sigma_{\mathrm{zz}} - \sigma_{\mathrm{rr}})}{3}\boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}} \tag{44}$$

which results in $\tilde{\sigma} = |\sigma_{\mathrm{zz}} - \sigma_{\mathrm{rr}}|$ and $\tilde{\sigma}^{\mathrm{db}} = D_2\,[(2\,\sigma_{\mathrm{rr}} + \sigma_{\mathrm{zz}})/3 - D_3]$ because $D_4 = 1$. When $\delta < D_5$ and D_6 is sufficiently large, then $1 - (\delta/D_5)^{D_6} \approx 1$ and Eq. (35) becomes

$$\dot{\delta} = \frac{D_1}{1-\delta}\,\left\langle \tilde{\sigma} - \tilde{\sigma}^{\mathrm{db}} \right\rangle\,\dot{\tilde{\varepsilon}}^{\mathrm{dg}}. \tag{45}$$

Once $\delta = D_5$ then $\dot{\delta} = 0$. When $\tilde{\sigma}$ is held constant, then Eq. (45) can be analytically integrated to

$$\delta = 1 - \sqrt{1 - 2\,D_1\,(\tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}})\,\tilde{\varepsilon}^{\,\mathrm{dg}}}, \tag{46}$$

where $\delta = 0$ is assumed at $\tilde{\varepsilon}^{\,\mathrm{dg}} = 0$. If $P_1 = 0$, G_2 is sufficiently large that $\sinh(x) \approx x$, and $G_3 = 5$, then one can analytically integrate Eq. (34) to obtain

$$\tilde{\varepsilon}^{\,\mathrm{dg}} = \frac{1 - \left[1 - 5\,G_1\,D_1\,\left(\frac{\tilde{\sigma}}{G_2}\right)^5\,t\,\left(\tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}}\right)\right]^{2/5}}{2\,D_1\,\left(\tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}}\right)}, \tag{47}$$

where $\tilde{\varepsilon}^{\,\mathrm{dg}} = 0$ is assumed at $t = 0$. Finally, Eqs. (29) to (31) can be utilized to obtain the total strain as

$$\boldsymbol{\varepsilon} = \frac{\mathring{\boldsymbol{\sigma}}}{3\,E_1} + \frac{\tilde{\boldsymbol{\sigma}}}{2\,E_2\,(1-\delta)} + \tilde{\varepsilon}^{\,\mathrm{dg}}\,\frac{\partial \tilde{\sigma}}{\partial \boldsymbol{\sigma}}, \tag{48}$$

where $\partial\tilde{\sigma}/\partial\boldsymbol{\sigma} = -1/2\,\boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} - 1/2\,\boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}}$.

Plots of this analytical solution and the numerically computed solution are shown in Fig. 13b. An acute above a variable $(\acute{\cdot})$ in Fig. 13b denotes the analytical solution for that variable, while a variable without an acute above it $(\cdot)$ denotes the numerical solution for the variable or the applied conditions. The top plot shows that the applied stress components were abruptly increased from zero to $\sigma_{\mathrm{zz}} = 18.5$ MPa and $\sigma_{\mathrm{rr}} = 5$ MPa at $t = 0$, causing $\tilde{\sigma} = 13.5$ MPa and $\tilde{\sigma}^{\mathrm{db}} = 11.5$ MPa. This stress state is held for the next 50 days, leading to damage evolution and tertiary creep until $t = 40.1$ d. At $t = 40.1$ d, the damage and the creep rate stop evolving because the damage has hit the damage limit ($\delta = D_5$). The numerical and analytical solutions for the damage and total strain agree quite well.

3.7 Convergence Studies

Once the new model was implemented and (partially) verified, convergence studies similar to those briefly mentioned in Section 3.1 were performed. Each convergence probe began by

specifying the initial spherical stress $\mathring{\sigma}_{n-1}$, the initial equivalent stress $\tilde{\sigma}_{n-1}$, and the initial damage δ_{n-1}, and the axial strain rate $\dot{\varepsilon}_{zz,n}$ and the axial strain increment $\Delta\varepsilon_{zz} = \dot{\varepsilon}_{zz,n}\,\Delta t$. The initial stress state $\boldsymbol{\sigma}_{n-1}$ was constructed as

$$\boldsymbol{\sigma}_{n-1} = \left(\mathring{\sigma}_{n-1} - \frac{\tilde{\sigma}_{n-1}}{3}\right) \boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} - \left(\mathring{\sigma}_{n-1} - \frac{\tilde{\sigma}_{n-1}}{3}\right) \boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \left(\mathring{\sigma}_{n-1} + \frac{2\,\tilde{\sigma}_{n-1}}{3}\right) \boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}}. \quad (49)$$

while the strain rate was prescribed as

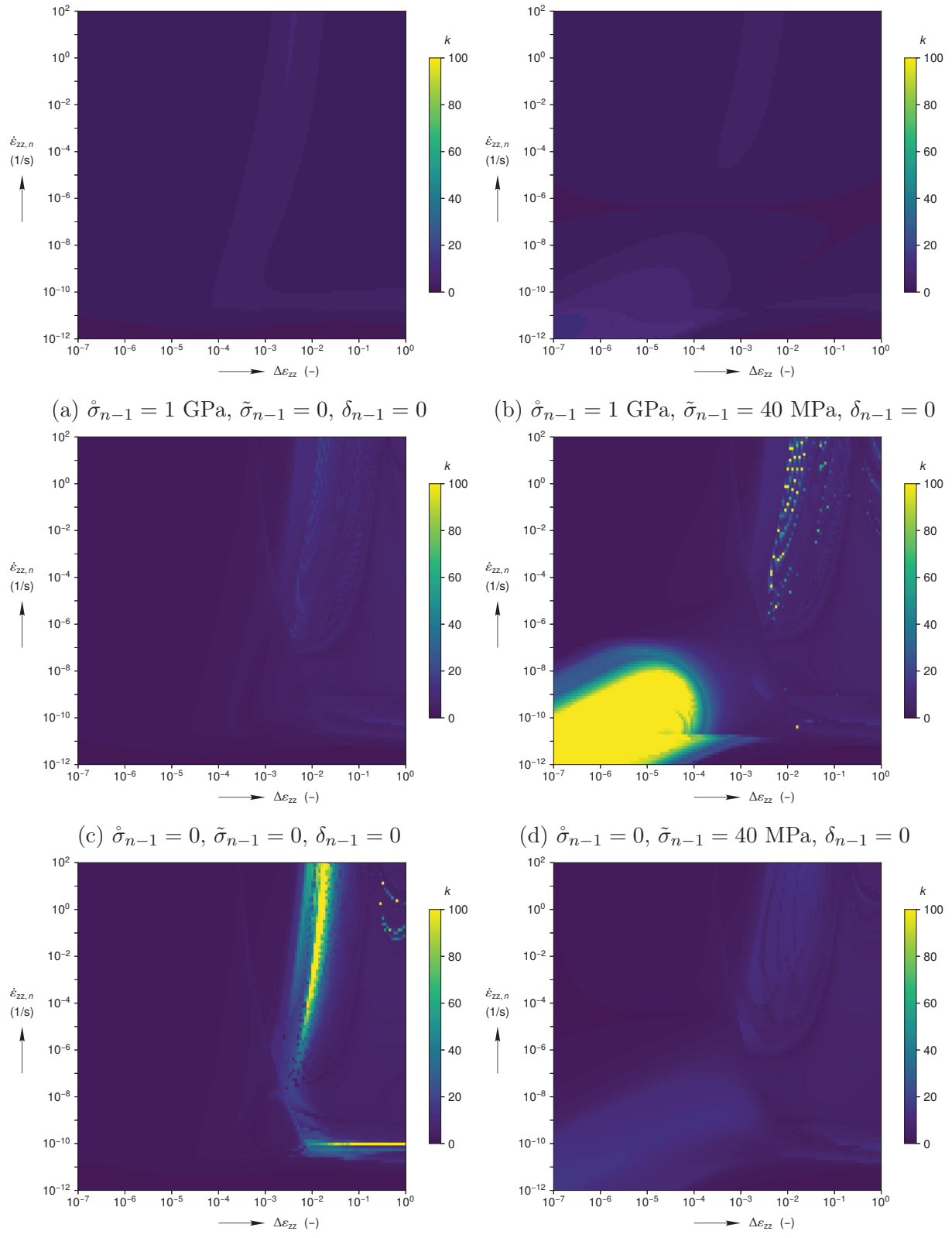
$$\dot{\boldsymbol{\varepsilon}}_n = -\frac{\dot{\varepsilon}_{zz,n}}{2}\, \boldsymbol{e}_{\mathrm{r}} \otimes \boldsymbol{e}_{\mathrm{r}} - \frac{\dot{\varepsilon}_{zz,n}}{2}\, \boldsymbol{e}_{\theta} \otimes \boldsymbol{e}_{\theta} + \dot{\varepsilon}_{zz,n}\, \boldsymbol{e}_{\mathrm{z}} \otimes \boldsymbol{e}_{\mathrm{z}}. \quad (50)$$

Note that $\mathrm{tr}(\dot{\boldsymbol{\varepsilon}}_n) = 0$, such that $\mathring{\sigma}_{n-1} = \mathring{\sigma}_n$. Damage free behavior was studied first by using $\mathring{\sigma}_{n-1} = 1$ GPa, $\delta_{n-1} = 0$, and $\tilde{\sigma}_{n-1} = \{0,\ 20\ \mathrm{MPa},\ 40\ \mathrm{MPa}\}$, while continuously varying $10^{-12}\ 1/\mathrm{s} \leq \dot{\varepsilon}_{zz,n} \leq 10^{2}\ 1/\mathrm{s}$ and $10^{-7} \leq \Delta\varepsilon_{zz} \leq 10^{0}$. Damaged behavior was studied next by using $\mathring{\sigma}_{n-1} = \{0,\ 20\ \mathrm{MPa},\ 40\ \mathrm{MPa}\}$, $\delta_{n-1} = \{0,\ 0.4\ , 0.8\}$, and $\tilde{\sigma}_{n-1} = \{0,\ 20\ \mathrm{MPa},\ 40\ \mathrm{MPa}\}$, while continuously varying $10^{-12}\ 1/\mathrm{s} \leq \dot{\varepsilon}_{zz,n} \leq 10^{2}\ 1/\mathrm{s}$ and $10^{-7} \leq \Delta\varepsilon_{zz} \leq 10^{0}$. Each study purposely bypassed the material model's semi-implicit scheme to probe the number of iterations k for the fully-implicit scheme to converge to an acceptable solution ($\omega_n \leq 1$). All cases utilized the Collapse parameter set in Table 1.

Figure 14 shows a selection of the results from the convergence studies. The initial conditions for each case are shown in the caption of each sub-figure. When $\mathring{\sigma}_{n-1} = 1$ GPa, the model reliably converges for $\tilde{\sigma}_{n-1} = 0$ and 40 MPa, as shown in Figs. 14a and 14b. When $\mathring{\sigma}_{n-1} = 0$ GPa and $\delta_{n-1} = 0$, the model reliably converges if $\tilde{\sigma}_{n-1} = 0$ (see Fig. 14c), yet fails to converge for most cases in the lower left of Fig. 14d, a few scattered cases in the upper right of Fig. 14d, and a single isolated case in the lower right of Fig. 14d if $\tilde{\sigma}_{n-1} = 40$ MPa. When $\mathring{\sigma}_{n-1} = 0$ GPa and $\delta_{n-1} = 0.4$, the model fails to converge for some $\Delta\varepsilon_{zz} \approx 10^{-2}$ cases, some $\dot{\varepsilon}_{zz} = 10^{-10}\ 1/\mathrm{s}$ cases, and a few scattered cases if $\tilde{\sigma}_{n-1} = 0$ MPa (see Fig. 14e), but converges in all cases if $\tilde{\sigma}_{n-1} = 40$ MPa (see Fig. 14f). The failures to converge in Figs. 14d and 14e are not ideal, but these cases probably do not occur very often in the room collapse simulations. A material point does not typically reach $\mathring{\sigma}_{n-1} = 0$ and $\tilde{\sigma}_{n-1} = 40$ MPa without $\delta > 0$ and a material point does not typically reach $\delta = 0.4$ with $\tilde{\sigma}_{n-1} = 0$ MPa. Nevertheless, it is still worthwhile to study the causes of these convergence failures.

Figure 15 investigates a selection of cases from Fig. 14 more closely by plotting the individual iterates as the fully-implicit scheme searches for a satisfactory solution. Each convergence path begins with a solid light red dot, signifying the initial guess, followed by either hollow or solid red dots, respectively signifying an abandoned full Newton iterate or an accepted line search iterate. If the solution converged ($\omega_n \leq 1$), then the converged solution is marked with a dark solid dot. The merit function landscape for each case is plotted behind each convergence path. (The lower limit for the merit function color scale is 10^4 instead of 1 so that one can easily detect gradients.)

The initial conditions, strain rate, and strain increment for Fig. 15a correspond to a typical case near the center of Fig. 14c. The defining feature of the merit landscape in Fig. 15a is a long narrow curving canyon. In six iterations, the fully-implicit scheme started with a fairly poor initial guess and quickly converged to the solution at the bottom of the canyon.



(a) $\mathring{\sigma}_{n-1} = 1$ GPa, $\tilde{\sigma}_{n-1} = 0$, $\delta_{n-1} = 0$

(b) $\mathring{\sigma}_{n-1} = 1$ GPa, $\tilde{\sigma}_{n-1} = 40$ MPa, $\delta_{n-1} = 0$

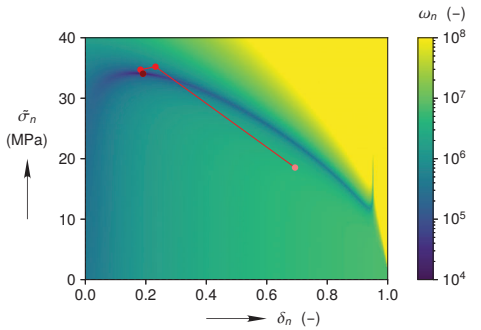
(c) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 0$, $\delta_{n-1} = 0$

(d) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 40$ MPa, $\delta_{n-1} = 0$

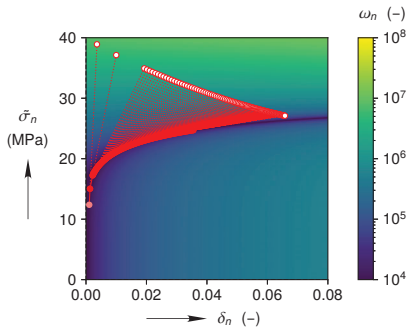
(e) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 0$, $\delta_{n-1} = 0.4$

(f) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 40$ MPa, $\delta_{n-1} = 0.4$

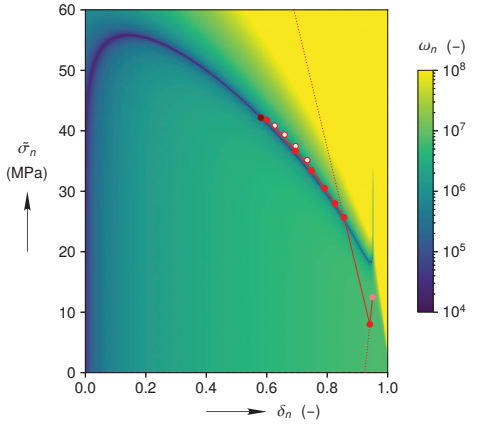
Figure 14: Number of iterations for the fully-implicit method to converge for a range of strain rates and strain increments, starting from a selection of initial conditions.



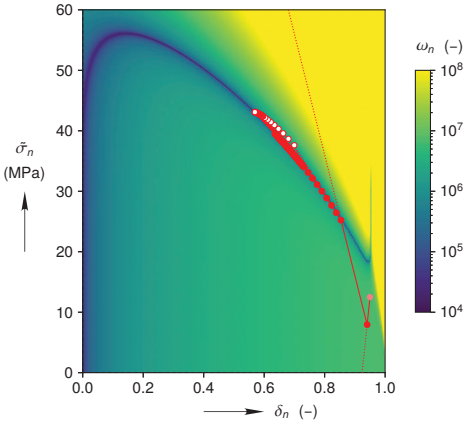
(a) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 0$, $\delta_{n-1} = 0$, $\dot{\varepsilon}_{zz,n} = 10^{-6}$ 1/s, $\Delta\varepsilon_{zz} = 2 \times 10^{-3}$



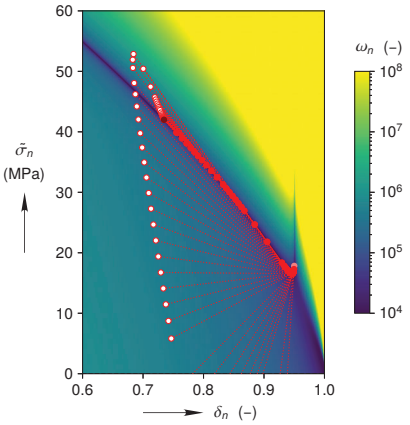
(b) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 40$ MPa, $\delta_{n-1} = 0$, $\dot{\varepsilon}_{zz,n} = 10^{-9}$ 1/s, $\Delta\varepsilon_{zz} = \times 10^{-5}$



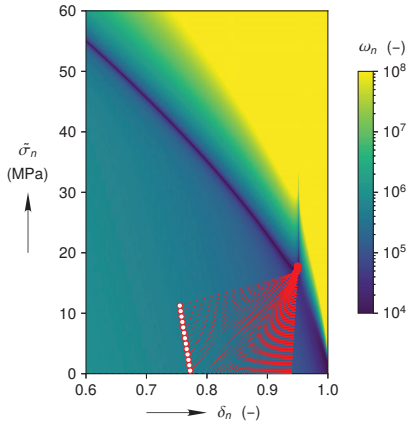
(c) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 40$, $\delta_{n-1} = 0$, $\dot{\varepsilon}_{zz,n} = 1.78 \times 10^{-4}$ 1/s, $\Delta\varepsilon_{zz} = 4.7 \times 10^{-3}$



(d) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 40$ MPa, $\delta_{n-1} = 0$, $\dot{\varepsilon}_{zz,n} = 1.78 \times 10^{-4}$ 1/s, $\Delta\varepsilon_{zz} = 4.467 \times 10^{-3}$



(e) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 0$, $\delta_{n-1} = 0.4$, $\dot{\varepsilon}_{zz,n} = 10^{-2}$ 1/s, $\Delta\varepsilon_{zz} = 10^{-2}$



(f) $\mathring{\sigma}_{n-1} = 0$, $\tilde{\sigma}_{n-1} = 0$ MPa, $\delta_{n-1} = 0.4$, $\dot{\varepsilon}_{zz,n} = 10^{-2}$ 1/s, $\Delta\varepsilon_{zz} = 1.1 \times 10^{-2}$

Figure 15: Fully-implicit iterates and merit function landscapes for certain initial conditions, applied strain rates, and applied strain increments. The solid light red dot on each path indicates the initial guess. Hollow red dots indicate a Newton iterate that was abandoned. Solid red dots indicate an accepted Newton or line search iterate. A solid dark red dot indicates a converged solution. (b), (d), and (f) do not contain a solid dark red dot.

The initial conditions, strain rate, and strain increment for Fig. 15b correspond to a failure to converge in the lower left corner of Fig. 14d. The initial guess starts in the canyon, but each full Newton step goes outside the curving canyon, causing line search cutbacks until an acceptable iteration is found back in the canyon. In this way, the fully-implicit scheme grinds its way along the canyon, but it makes too little progress in 100 iterations. All the other failed cases in the lower left corner of Fig. 14d fail in a similar manner: the fully-implicit scheme cannot efficiently navigate along the bottom of a sharply curving canyon.

Figures 15c and 15d investigate two neighboring cases in the spotty region in the upper right of Fig. 14d. The initial guess in Fig. 15c begins outside of the canyon, but the second iterate, after a line search, arrives at the upper rim of the canyon. The fully-implicit scheme efficiently progresses along the gently curving canyon rim until the convergence path dives down to the converged solution at the canyon bottom. The initial guess in Fig. 15d also begins outside of the canyon, but the second iterate, after a line search, lands inside the canyon. The fully-implicit scheme appears to almost reach the converged solution in $k \leq 100$, but fails. The other failed cases in the upper right corner of Fig. 14d are presumed to fail in a similar manner: despite an initial guess outside the canyon and a gently curving canyon, the convergence path lands prematurely inside the canyon and progresses too slowly to reach the converged solution before $k = 100$.

The last two cases in Figs. 15e and 15f correspond to the left of and inside the nearly vertical yellow band at $\Delta\varepsilon_{zz} \approx 10^{-2}$ in Fig. 14e. In both cases, the initial guess is located just above a sharp turn in the canyon. The fully-implicit scheme manages to converge in 53 iterations in Fig. 15e, yet fails to get past the sharp turn in 100 iterations in Fig. 15f. These convergence differences could be due to slight differences in the curvature of the turn, canyon narrowness, or the location of the initial guess within the canyon. Increasing $k_{\max}$ to 500 for the $\Delta\varepsilon_{zz} = 1.1 \times 10^{-2}$ case permitted the implicit solution scheme to slowly progress along the bottom of the canyon to the solution in 160 iterations, suggesting the solver was not stuck in a local minimum after the first 100 iterations.

Increasing $k_{\max}$ from 100 to 500 reduced the total percentage of failed cases from 1.05 % to 0.03 %, and resolved all of the failed cases in Fig. 14. (Fig. 14 only shows 22 % of the total number of cases considered.) This improvement, coupled with the study in Fig. 15, suggests that nearly all the failures were related to the fully-implicit scheme struggling to navigate the curving narrow canyon. The 0.03 % of remaining failures probably ran into overflow errors or local minima, but further study would be needed to say for certain.

Fortunately, several methods have been developed in the optimization literature that are well-suited to navigating narrow canyons, such as conjugate gradient and quasi-Newton methods [Press et al., 2007, Chapter 10]. In addition, Shterenlikht and Alexander [2012] had success integrating a plasticity–damage model with a trust region method and their merit function exhibited narrow valleys, sometimes with a very steep localized drop to the global minimum. If, in the future, improving the initial guesses and increasing $k_{\max}$ to 500 is insufficient for the room collapse simulations, methods well suited to reliably navigate curving canyons will be considered.

4 Room Collapse and Reconsolidation Simulations

This section outlines the development and status of structural simulations of the room collapse and reconsolidation. First, attempts to replicate the results of our external collaborators from Reedlunn et al. [2019, Section 4] are presented. Then, motivations for distancing from Reedlunn et al. [2019] and proposing new modeling approaches are detailed. Finally, the current status of our modeling capabilities is shown.

4.1 Replication Attempts

The model setup presented in Reedlunn et al. [2019, Section 2.2] was used; only half of the room was modeled with a vertical symmetry boundary condition. Contrary to the 2D RKPM code of UCSD, Sierra/SM only allows 3D domains so that the 2D elevation plan of the room from Reedlunn et al. [2019, Section 2.2] had to be extruded over a small thickness. The old salt material model (see Reedlunn et al. [2019, Section 2.2.3]) was utilized with the parameter sets listed in Reedlunn et al. [2019, Table 2-1], with the modification to the damage limits mentioned below. Damage was suppressed in the gradual room closure simulations and it was allowed to evolve in the room collapse simulations.

Gradual room closure simulations using the Sierra/SM RKPM capability performed well and provided results similar to those of Reedlunn et al. [2019, Chapter 5] obtained using the CRK mini-app. Contact between the roof, floor and walls were handled well by both Dash contact and sub-hex-centered pinball contact on existing outer surfaces. For Dash contact, some overlapping of the hexahedral faces was noticed near the sharp corners of the room, likely due to contact search and detection issues; this behavior did not compromise the gradual room closure simulation results in general.

Room collapse simulations, with the introduction of bond-based failure, were challenging due to several numerical difficulties:

1. Particle spray was prevalent, in spite of the efforts to reduce it with the bond-based failure method (see Section 2.2). The sprayed particles, i.e., RKPM nodes, carried with them the sub-hex integration domains, each of which had a pinball located at its center. This led to highly distorted sub-hexes and contact between distant parts of the domain that caused unphysical behavior and likely contributed to the numerical difficulties of the material models presented in the next 2 items. As a consequence, further replication simulations did not include contact.

2. The first simulations performed using the old explicitly integrated material model presented in Section 3 would consistently freeze at some point during execution. As detailed in Section 3.1, extremely small time sub-steps were used to ensure numerical stability when large values of damage were reached, which prevented simulations from completing within an acceptable time period, regardless of computational resources. A tensile damage limit no larger than $\delta^{\mathrm{t}} = 0.6$ was required to complete simulations, much smaller than the value of $\delta^{\mathrm{t}} = 0.999999$ used in Reedlunn et al. [2019]. However, using

a low tensile limit was a necessary but not a sufficient condition to avoid freezing of the simulations. Exploratory simulations with slight variations of the parameters from Reedlunn et al. [2019, Table 2-1] have led several simulations to freeze with $\delta^{\mathrm{t}} = 0.6$, probably due to sudden large strain rates caused by bond failure and particle spray. This highlights the numerical fragility of the old explicitly integrated material model in loading conditions essential to the room collapse modeling.

3. The use of the implicitly integrated version of the material model resolved the issue of simulations freezing. However, the material model failed to converge when the compressive and tensile damage limits were different $\delta^{\mathrm{c}} \neq \delta^{\mathrm{t}}$, or when the damage limits were larger than $\delta^{\mathrm{c}} = \delta^{\mathrm{t}} = 0.7$. The formulation and solver challenges detailed in Section 3.1 explain these convergence problems.

In spite of these difficulties, successful simulation of damage initiation and progress through the domain, leading to failure, separation, and free fall of a block of the roof was achieved with the implicitly integrated old material model using low and equal damage limits of $\delta^{\mathrm{c}} = \delta^{\mathrm{t}} = 0.6$. Subsequent reconsolidation could not be simulated due to the lack of a robust contact model.

4.2 Modeling Limitations and Updated Approach

The attempts to replicate the RKPM simulations of Reedlunn et al. [2019, Section 4] summarized in Section 4.1 highlighted several difficulties and limitations that can be attributed to a few key factors.

4.2.1 Material Model

Every replication simulation that failed to complete did so due to either the explicitly integrated material model freezing, or the implicitly integrated material model not converging. Therefore, the numerical robustness of the material model was recognized as critical to the modeling capability. Investigation of the material model convergence failures revealed several issues, as discussed in Section 3.1.

Besides these numerical aspects, the initiation and accumulation of damage were slower in our simulations when compared to identical configurations from Reedlunn et al. [2019, Section 4]. This was the case at early stages of the deformation, when little damage was present and differences between the Lagrangian and semi-Lagrangian approaches were expected to be negligible. While this discrepancy remains partially unexplained, it suggested that our replication attempts would not be entirely successful even after resolving the numerical difficulties. This observation supported pivoting to a different material model, that would mitigate and alleviate the limitations encountered. The improved viscoplastic continuum damage model whose formulation and implementation are described in Section 3.2 to 3.7 was used in all further simulations presented in Section 4.3.

4.2.2 Contact Method

Differences in behavior observed for simulations performed with and without pinball contact, even at early stages before the faces of the room actually come into contact, seem to indicate some undesirable interactions between the forces generated by the sub-hex centered pinballs and the stress calculations by the material model. These interactions may have contributed to the material model convergence failures.

The sub-hexes only serve as RKPM integration zones, and do not represent physical material locations. As detailed in Section 2.3.2, locating pinballs at the centroid of each sub-hex proves problematic when the material is damaged. Sub-hexes may get distorted and move large distance as a result of large deformations and particle spray. The associated pinballs tend to fill in and clog the room, preventing the roof from falling. Simulations performed while deactivating pinball contact for highly distorted sub-hexes alleviated the issue: the roof fell and landed on the floor. However, such approach leads to significant material interpenetration during reconsolidation, in addition to lacking robustness, and was not pursued.

The option to place the pinballs directly at the nodes (see Section 2.3.2) was developed to mitigate pinball drift. This option limited the interactions between the pinball forces and material model stress calculations, leading to more robust simulations. Locating the pinballs at the nodes also resolved the problem of the room getting clogged with pinballs. While we do not yet fully understand why this happens, several hypotheses may be advanced. First, simulations seem to indicate that particle spray is less prominent with nodal pinballs than with sub-hex pinballs, leading to less pinball drift. Then, the number of nodes being much smaller than the number of sub-hexes, fewer nodal pinballs are present. Even though the volume of a nodal pinball is larger than that of a sub-hex-centered pinball, this reduced number of interactions, and possibly the influence it has on the force assembly (different set of nodes in (18)) is believed to be a possible reason why nodal pinballs do not clog the room when particle spray occurs. In addition, pinballs are initially placed on the outer surfaces of the discretization, and a large number of pinballs are located on the front and back surfaces of the thin 3D domain where no contact interactions are expected. In a truly 2D domain, pinballs would only be located on the perimeter of the room. These additional and unwarranted contact interactions are believed to contribute to the non-physical response and clogging of the room.

Another issue is nodes that are not initially subject to the symmetry boundary condition may move outside of the domain. To prevent this loss of material, an inert and fixed block of material is located at the symmetry plane so that any node reaching the symmetry plane at the center of the room, and their associated pinball bounce against the block and remain inside the domain.

4.3 Current Status

New simulations employing the new material model and nodal pinballs were performed. The improved robustness of the model allowed simulations of the room collapse and reconsolidation problem far beyond what was achievable during the replication attempts in Section 4.1.

Table 2: Bond-based failure and pinball contact parameters employed for the current simulations of the room collapse and reconsolidation simulations.

Type	Parameter	Units	Values
Bond-based failure	ξ^{fail}	–	0.949
	u_{cod}	m	1.0×10^{-6}
	u_{cs}	m	1.0×10^{-6}
Pinball contact	β	–	0.01
	C_f	–	0.0
	α	–	1.1

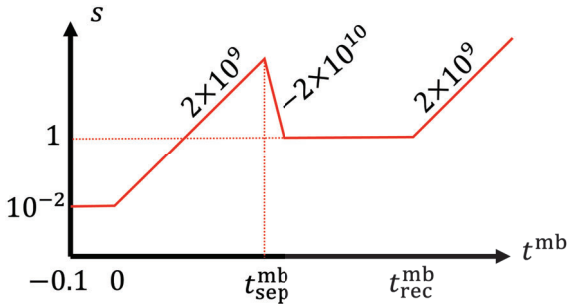
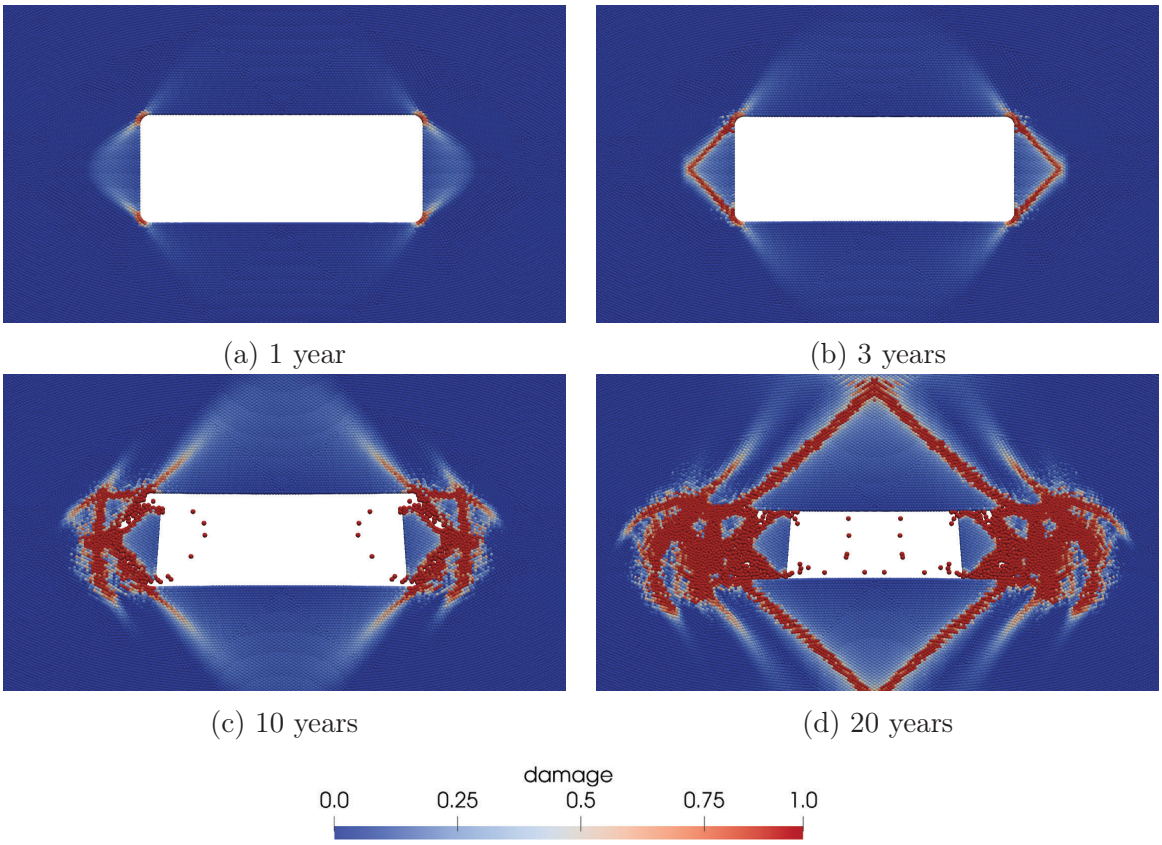


Figure 16: Time profile of viscoplastic scaling. For the room with and without a clay seam: $t^{\text{mb}}_{\text{sep}} = 0.8$s and $t^{\text{mb}}_{\text{rec}} = 2.1$s, for the manufactured problem with pre-damaged triangular band above the roof: $t^{\text{mb}}_{\text{sep}} = 0.3$s and $t^{\text{mb}}_{\text{rec}} = 1.18$s.

The domain was discretized finely (0.1 m sub-hex size) inside an elliptical region (horizontal radius = 3× room width, vertical radius = 2× room height) centered around the room, and coarsely away from the room (2 m largest sub-hex size). The thickness of the domain was chosen equal to the fine discretization size of 0.1 m. This discretization ensures the fine region encompasses all damaged material. Prior simulations using a finely discretized region of smaller dimensions had damage rapidly propagate and spread horizontally through the entire coarse region, likely due to the insufficient spatial resolution. Additionally, since nodal pinballs are sized based on the minimum distance to their neighbors, high aspect ratio discretizations like the one employed in the coarse region led to undersized pinballs and contact on new fracture surfaces not being properly enforced, leading to the top part of the domain above the damaged region to simply fall through the bottom part below. The Collapse parameter set from Table 1 was used. The bond-based failure, and pinball parameters are given in Table 2. The nodal failure criterion $\xi > \xi^{\text{fail}}$ is based on the material damage variable, i.e. $\xi \equiv \delta$, with ξ^{fail} the nodal failure limit. Values of u_{cod} and u_{cs} are chosen to be very small to foster separation. All simulations used the scattered pinball force assembly method; attempts to use direct force assembly generally resulted in less robust simulations, i.e., the material model failed to converge earlier in the simulation. Viscoplastic scaling $s = dt/dt^{\text{mb}}$ was employed to simulate each long material time dt using explicit time integration which requires very short momentum balance time step dt^{mb}. Detailed description of this technique is given in Reedlunn et al. [2019, Section 4.4.2.1]. The time-profile of the

viscoplastic scaling was prescribed a priori; attempts at an adaptive automated viscoplastic scaling were partially successful but hindered by the difficulty to robustly identify when material was undergoing free fall. As shown in Fig. 16, the scaling is initially very low to model elastic excavation. At $t^{\mathrm{mb}} = 0$, the scaling is ramped up to model creep and damage propagation until the roof separates at time $t^{\mathrm{mb}}_{\mathrm{sep}}$, then ramped down to a value of 1 to model free fall in real time, before being ramped up again to model subsequent reconsolidation at time $t^{\mathrm{mb}}_{\mathrm{rec}}$.



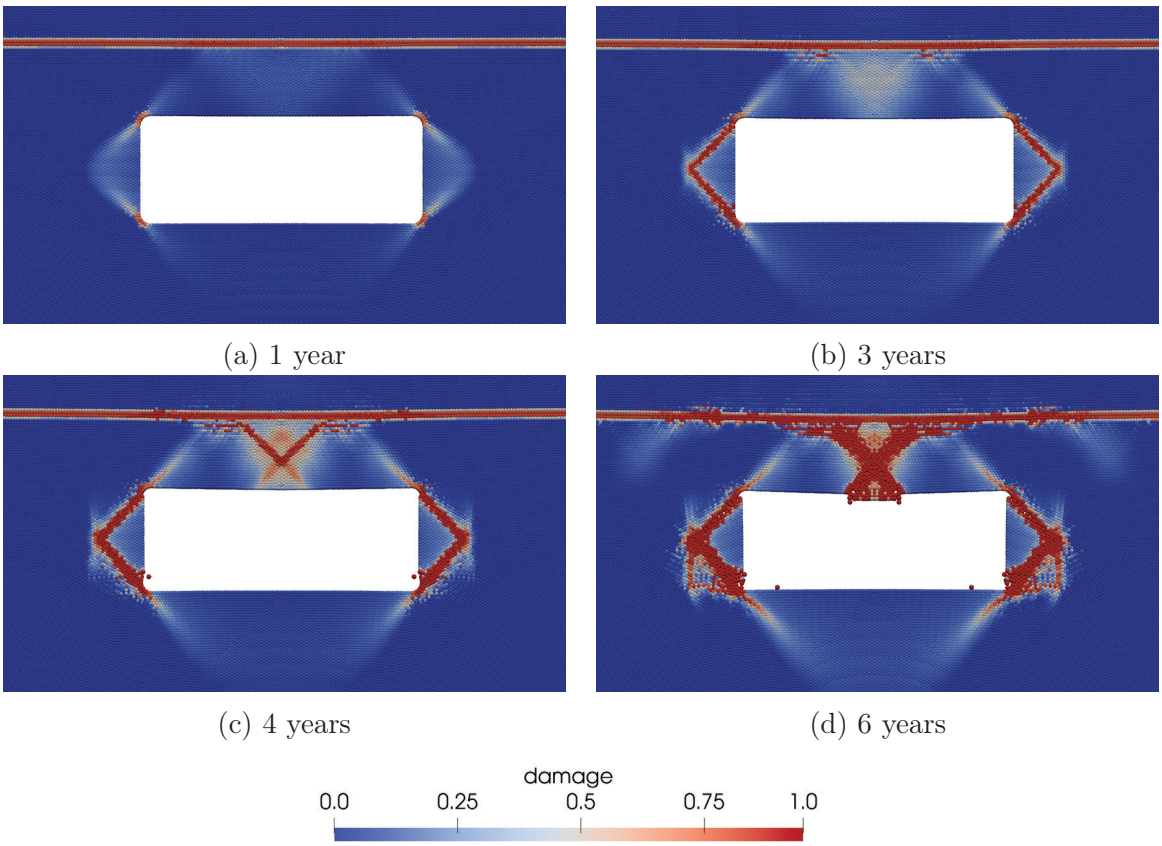
(a) 1 year (b) 3 years

(c) 10 years (d) 20 years

Figure 17: Simulation of the room collapse and reconsolidation with frictionless contact and without a clay seam. The spheres in the visualization represent the nodes.

Simulation of the room without a clay seam shows damage progressing in the pillar and above the room, cutting out triangular blocks from the wall and roof (Figure 17). Under frictionless contact, separation is obtained but the chronology is different from that of Reedlunn et al. [2019, Section 4]: the walls detach earlier than the roof and are pushed into the room, providing support between the floor and roof and preventing free fall of the triangular block of the roof that was cut out by the damage (Figure 17). The fact that the walls appear to be propelled inside the room seems to indicate that the contact forces generated by the pinballs might be larger and more abrupt than anticipated and may require further investigation. A more realistic frictional contact was attempted for several values of the friction coefficient $C_f = 0.1$, 0.2, and 0.5. The damage fields were fairly similar but the roof and the walls did

not separate from the rest of the domain. The lack of separation was believed to be due to interlocking of the rugged contact surfaces provided by the pinballs, and frictional contact was abandoned for now.



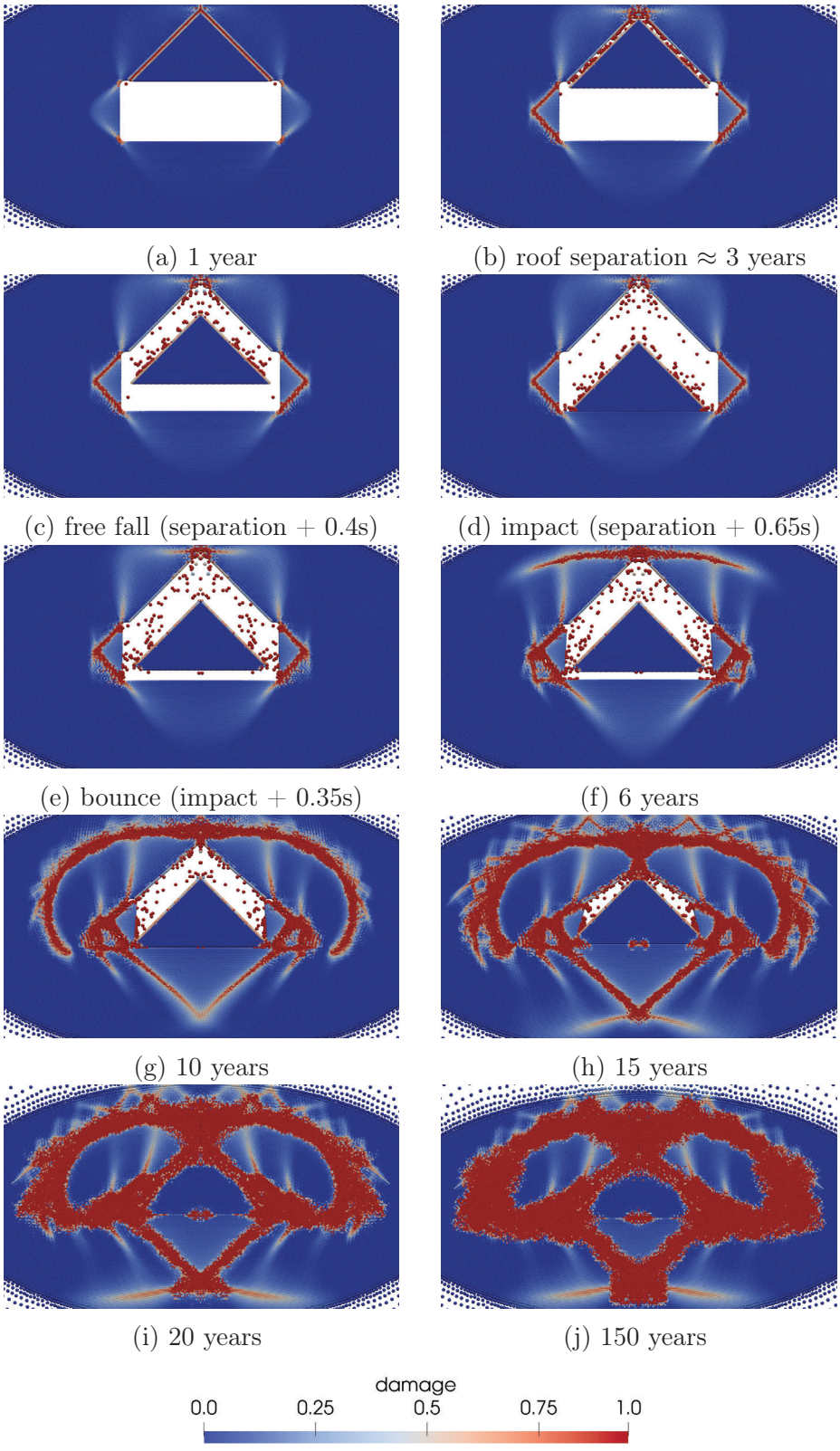
(a) 1 year (b) 3 years

(c) 4 years (d) 6 years

Figure 18: Simulation of the room collapse and reconsolidation with clay seam G modeled as a damaged band of material. The spheres in the visualization represent the nodes.

In an attempt to obtain a roof fall before the wall has progressed too far into the room, clay seam G was modeled to enhance damage above the room and separation of the roof. Using two distinct material blocks in contact is not possible with pinballs located at the nodes because of the surface protrusion described in Section 2.3.2. Since pinballs on opposite contact surfaces are co-planar with the two parallel contact surfaces, the overlap between the blocks is inadequately detected as a separation between the pinballs by the algorithm. The clay seam was modeled as an initially damaged thin band of material instead.

Damage progresses from the top corner of the room towards the clay seam in Fig. 18, but is not sufficient to cut out a block of material from the roof before the simulation failed after only six years of simulated time. Damage propagates rapidly inside and around the clay seam, likely due to interactions between the pinballs on new fracture surfaces located inside the clay seam, and the salt material, as conjectured previously. Further troubleshooting and development would be needed if it becomes necessary to resolve this issue.



(a) 1 year

(b) roof separation ≈ 3 years

(c) free fall (separation + 0.4s)

(d) impact (separation + 0.65s)

(e) bounce (impact + 0.35s)

(f) 6 years

(g) 10 years

(h) 15 years

(i) 20 years

(j) 150 years

Figure 19: Simulation of the room collapse and reconsolidation with frictionless contact, without a clay seam, and with a manufactured pre-damaged triangular band above the room. The spheres in the visualization represent the nodes.

Challenged with the inability to obtain a roof fall occurring naturally from a realistic initial configuration, a manufactured problem with a pre-damaged triangular band above the room was developed; this simulation did not employ a clay seam. The purpose was to provide confidence that, once the desired chronology of the wall and roof separation can be obtained, the ensuing phenomena would be modeled properly. Further details of the simulation setup can be found in the Cubit journal file and Sierra/SM input file in Appendix B.

Images from the simulation with a pre-damaged roof are shown in Fig. 19. These images demonstrate the ability of our modeling framework to capture the separation and free fall of the roof, the impact of the roof with the floor, and the subsequent reconsolidation of the room with contact on new fracture surfaces. The roof may bounce on the floor multiple times due to the small amount of energy dissipation. To avoid long and expensive simulations, the viscoplastic scaling is increased soon after the first bounce which explains why the roof fall block is still up in the air at a material time of 6 years (Figure 19f). Once a block of material detaches from the surface of the room, the material above that new fracture surface continues to damage and begins to fracture until the rubble pile inside the room provides support and these cascading events stop. Once the room is reconsolidated, damage continues to slowly progress, and the damage bands tend to broaden (compare thinner pattern at 20 years in Figure 19i to thicker pattern at 150 years in Figure 19j) before appearing to reach a stable equilibrium. While this smearing of the damage is not fully explained, it is conjectured to be related to fluctuations in the pressure field. Pressure checkerboarding is a known issue with the proposed integration method, and since damage is pressure dependent, that could lead to excessive and cascading damage propagation between adjacent sub-hexes. Pressure smoothing techniques do not currently work with bond-based failure, alternative strategies will be explored in future work.

The damage patterns in Figure 19 presents interesting differences from those of Reedlunn et al. [2019, Figure 4-17]. First, the roof fall block breaks into smaller pieces upon hitting the floor in Reedlunn et al. [2019, Figure 4-17], yet it remains largely intact in the present simulation, because their simulations employed a brittleness parameter. When they set their brittleness parameter to zero, similar to the current material model, their roof fall block did not break into smaller pieces upon impacting the floor Reedlunn et al. [2019, Figure 4-24]. Second, in their simulations, the damage bands propagate horizontally from the top of where the roof first detached, and the damage bands propagate vertically from the top of where the walls first detach intersect to cut new triangles above the fallen roof. In the present simulation, these bands do not intersect and the horizontal one propagates out and down, forming spirals which eventually join the walls and cut large lobes around the room. These differences in the damage patterns might be due to the finer discretization employed herein.

5 Summary and Future Work

The ceilings and walls of abandoned empty rooms at the WIPP tend to collapse over time, causing rubble piles of salt to form on floors of these rooms. A previous collaboration with two universities [Reedlunn et al., 2019] showed that meshless methods, such as MPM and

RKPM, are well-suited to modeling the collapse, rubble pile formation, and reconsolidation of these empty rooms. This memorandum details Sandia's recent efforts to develop a capability to run similar meshless simulations within Sierra/SM.

The team quickly identified several areas for improvement. Sandia's RKPM research code needed to be polished and integrated into Sierra/SM. More significantly, the fledgling fracture and contact capabilities required enhancements. The bond-based fracture method appeared promising, but the (primary) bond failure criterion would often leave many bonds intact. The existing pinball contact method was straightforward, but only included normal contact on pre-existing surfaces. The simple salt constitutive model at the core of any empty room simulation was also in need of improvement. Once damage became severe, the salt model's explicit integration scheme would call for an excessive number of sub-steps, causing the simulation to effectively freeze. These issues prevented the team from assessing whether the empty room model could simulate a roof fall. Accordingly, the team developed RKPM and the salt material model while continually exercising these capabilities on empty room simulations.

The RKPM development began by integrating the research code into Sierra/SM, and then pursued improvements to the fracture and contact methods. A secondary bond failure criterion was developed to degrade the remnant bonds that intersect the failure plane between two nodes that previously met the primary bond failure criterion. Although the secondary criterion appeared to work as intended, small residual forces were detected in a simple tension test and further study is needed to determine the cause. The pinball contact implementation was extended to automatically place pinballs at new fracture surfaces and to model frictional forces between pinballs, both of which are important for the reconsolidation phase of the empty room simulations. Tests of these new features show they are behaving as expected. The user was also given the ability to (1) specify whether a pinball force is scattered to nearby nodes or directly applied to one node, and (2) select whether pinballs are placed at the centroid of the sub-hexahedral integration domains or at the nodes.

The salt constitutive model development began by switching to an implicitly integrated implementation. This implicit implementation was only marginally better than the explicit implementation, failing to converge too often for practical use in the empty room simulations. These convergence failures prompted a re-formulation of the model within the framework of Rational Thermodynamics to eliminate several model behaviors, such as the sharp stress drop at high strain rates, that caused numerical issues. This new model was also re-implemented to solve for two unknowns, instead of seven, and with improved initial guesses for the nonlinear solve. After verifying the implementation was working properly, the convergence of the new model was systematically probed. Investigating a selection of cases that failed to converge in 100 iterations found the line search augmented Newton solver was struggling to efficiently navigate along the bottom of a curving canyon in the merit function space. Consistent with this finding, the convergence failure rate was 1.05 % when the maximum number of iterations was 100, but the failure rate to 0.03 % when the maximum was increased to 500.

The initial empty room simulations were challenging due to issues with the bond-based fracture, pinball contact, and material model failures, but these issues progressively lessened as the capabilities were refined. After turning off sub-hex pinball contact and reducing the salt damage limits, damage bands developed from the upper corners of the room to cut out a triangular block of salt that fell from the roof. Contact of this block with the floor and subsequent reconsolidation could not be simulated until the pinball contact issues were resolved. Fortunately, the combination of placing pinballs at nodes, scattered pinball force assembly, and the material model improvements allowed simulations with pinball contact and new pinballs at fracture surfaces to run. Simulations with and without a clay seam, using frictionless or frictional contact all failed to produce a roof fall for various reasons. An artificial scenario was employed to encourage a triangular roof fall in which the salt above the room was artificially pre-damaged and frictionless contact was used. This manufactured scenario behaved as expected: the triangular block fell and landed on the floor, after which the surrounding rock salt closed in around the block.

Reedlunn et al. [2019, Chapter 8] listed a number of potential areas of future work. This list has been updated and included below:

1. Geomechanical Numerical Method

   (a) As discussed in Section 2.3, pinball contact's implicit surface representation is only an intermediate step toward a more sophisticated approach. The team eventually hopes to develop a method that creates explicit surfaces upon fracture. Such surfaces could be used with Sierra/SM's existing Dash contact capability, which should more accurately model normal and frictional contact. Pressure boundary conditions could also be applied to these surface to capture the effects of fluid trapped within the rubble pile.

   (b) As discussed in Section 2.2.5, the bond-based failure does not degrade the Lagrangian shape functions, which can lead to unphysical movement of a node after disconnecting from its original neighbors. An updated-Lagrangian or semi-Lagrangian formulation would be needed to resolve this issue.

   (c) As mentioned in Section 2.2.5, nodally integrated formulations exhibit "pressure checkerboarding" in nearly incompressible materials. Pressure checkerboarding can artificially spread out damage bands and prevent/delay fracture. Bond-based failure does not currently work with techniques to reduce pressure checkerboarding, so it may need to be modified or replaced with a different failure method.

   (d) Various numerical choices affect the width and behavior of highly damaged regions. These regions can behave similar to a fluid since damaged material has little shear resistance. The behavior of highly damaged regions may need to be tuned to reasonably match experimental and in-situ observations.

   (e) Material softening associated with damage and failure, without any regularization, is known to cause pathological discretization dependence. Some form of regularization will need to be applied in the future. The Reproducing Kernel

Strain Regularization (RKSR) employed in Reedlunn et al. [2019, Section 4.3] appears to be a promising approach.

(f) The empty room simulations utilized viscoplastic rate scaling to enable explicit integration of the equations of motion. The scaling history was input at the start of the simulation, making it difficult to know how much to increase the scaling during a quasi-static period, or know when to reduce the scaling for a dynamic event. Unless an automatically adaptive scaling can be implemented, it may be worthwhile to investigate implicitly integrating the equations of motion.

2. Salt Constitutive Model

(a) The model in Section 3.2 assumes the salt is always fully hardened for simplicity. Undisturbed salt, however, starts with minimal hardening and becomes fully hardened after substantial deformations, which is important when calibrating damage evolution parameters against experiments and when simulating empty room collapse and reconsolidation. Reedlunn [2022a] proposed a model with isotropic and kinematic hardening that captured salt's damage-free behavior over strain rates ranging from 10^{-12} 1/s to 10^{-4} 1/s. A damage evolution equation and degradation of salt's strength with damage need to be added to this more sophisticated constitutive model.

(b) Fractures in salt heal under sufficient confining pressure. Adding healing could make the rubble pile stiffen and behave less like a fluid.

(c) Salt macroscopically dilates as micro-cracks form and contracts as micro-cracks heal. These volumetric strains could play a role in the rubble pile mechanical response.

(d) Salt mechanical behavior at strain rates above 10^{-3} 1/s has not been thoroughly investigated, yet dynamic strain rate behavior is probably important during rock fall impact.

(e) Fracturing releases brine from isolated pores. Fluid assisted viscoplasticity (pressure solution redeposition) could accelerate the rate of rubble pile consolidation.

3. Geomechanical Model

(a) We hope to explicitly represent the macroscopic flow channels, but it remains to be seen whether a three-dimensional rubble pile simulation is computationally tractable.

(b) Reedlunn et al. [2019, Chapter 4] showed that detachment at clay seams causes roof falls to happen significantly earlier. Experimental characterization of clay seam tensile and shear behavior could help create a more accurate clay seam model.

(c) Potential back pressure due to fluids trapped within the room will alter the room's final porosity and permeability, as well as the time required to reach this final porosity and permeability.

(d) The geomechanical model results will need to be compared against laboratory and in-situ observations. *Predicting* when a roof fall will happen, the shape of the roof fall block, and the characteristics of the rubble pile may prove too difficult. It may be necessary to tune various model inputs, such as the distribution of defect sites in the salt above the room, to match observed roof falls and rubble piles. The model does, however, need to predict the empty room reconsolidation process, because we probably will not have in-situ observations to compare against. We, therefore, plan to compare rubble compaction predictions against laboratory experiments on various rubble sizes and at various strain rates.

4. Hydraulic Model

   (a) A robust method to transfer the geomechanical simulation results from RKPM simulations into a fluid flow code needs to be developed.

   (b) Computational fluid dynamics (CFD) was used to simulate flow through a single fracture in Reedlunn et al. [2019, Chapter 7], but it remains to be seen whether we have the computational power to simulate flow through an entire rubble pile with sufficient mesh resolution. Lower fidelity techniques, such as pore network modeling [Mitchell et al., 2024], may be a viable solution if they agree well with CFD simulations.

   (c) Interconnected micro-cracks permit fluid flow through salt. The permeability of micro-cracks could affect the effective permeability of the rubble pile once the consolidation process is nearly complete.

Despite the large number of potential improvements, the modeling results thus far are promising. Improving bond-based failure, pinball contact, and the salt constitutive model allowed the team to simulate artificial roof fall and subsequent reconsolidation with Sierra/SM. Future efforts will improve the models and begin to compare predictions against laboratory and in-situ observations.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This research is funded by WIPP programs administered by the Office of Environmental Management (EM) of the U.S. Department of Energy.

Bibliography

T. Belytschko and M. O. Neal. Contact-impact by the pinball algorithm with penalty and Lagrangian methods. *International Journal for Numerical Methods in Engineering*, 31(3): 547–572, 1991.

B. L. Boyce, S. L. B. Kramer, H. E. Fang, T. E. Cordova, M. K. Neilsen, K. Dion, A. K. Kaczmarowski, E. Karasz, L. Xue, A. J. Gross, A. Ghahremaninezhad, K. Ravi-Chandar, S. P. Lin, S. W. Chi, J. S. Chen, E. Yreux, M. Rüter, D. Qian, Z. Zhou, S. Bhamare, D. T. O'Connor, S. Tang, K. I. Elkhodary, J. Zhao, J. D. Hochhalter, A. R. Cerrone, A. R. Ingraffea, P. A. Wawrzynek, B. J. Carter, J. M. Emery, M. G. Veilleux, P. Yang, Y. Gan, X. Zhang, Z. Chen, E. Madenci, B. Kilic, T. Zhang, E. Fang, P. Liu, J. Lua, K. Nahshon, M. Miraglia, J. Cruce, R. DeFrese, E. T. Moyer, S. Brinckmann, L. Quinkert, K. Pack, M. Luo, and T. Wierzbicki. The sandia fracture challenge: blind round robin predictions of ductile tearing. *International Journal of Fracture*, 186(1):5–68, 2014. doi: 10.1007/s10704-013-9904-6. URL https://doi.org/10.1007/s10704-013-9904-6.

G. D. Callahan. Crushed salt constitutive model. Technical Report SAND98-2680, Sandia National Laboratories, Albuquerque, NM, USA, February 1999.

J.-L. Chaboche. Thermodynamic formulation of constitutive equations and application to the viscoplasticity and viscoelasticity of metals and polymers. *International Journal of Solids and Structures*, 34(18):2239–2254, June 1997.

J.-S. Chen and M. Hillman. Introduction to meshfree methods: Fundamentals and application. Center for Extreme Events Research (CEER) Intense Short Course, July 2020.

J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001.

J.-S. Chen, M. Hillman, and S.-W. Chi. Meshfree methods: Progress made after 20 years. *Journal of Engineering Mechanics*, 143(4):040170001, 2017.

S.-W. Chi, C.-H. Lee, J.-S. Chen, and P.-C. Guan. A level set enhanced natural kernel contact algorithm for impact and penetration modeling. *International Journal for Numerical Methods in Engineering*, 102(3-4):839–866, 2015. doi: https://doi.org/10.1002/nme.4728. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4728.

I. Cormeau. Numerical stability in quasi-static elasto/visco-plasticity. *International Journal for Numerical Methods in Engineering*, 9(1):109–127, 1975.

U. Düsterloh, K. Herchen, K.-H. Lux, K. Salzer, R.-M. Günther, W. Minkley, A. Hampel, J. G. Argüello Jr, and F. D. Hansen. Joint Project III on the comparison of constitutive models for the thermomechanical behavior of rock salt. III. Extensive laboratory test program with argillaceous salt from WIPP and comparison of test results. In L. Roberts, K. D. Mellegard, and F. D. Hansen, editors, *Proc. 8th Conference on the Mechanical Behavior of Salt*, pages 13–21, 2015.

F. Garofalo. An empirical relation defining the stress dependence of minimum creep rate in metals. *Trans. AIME*, 227:351–356, 1963.

B. Gross, N. Trask, P. Kuberry, and P. Atzberger. Meshfree methods on manifolds for hydrodynamic flows on curved surfaces: A generalized moving least-squares (GMLS) approach. *Journal of Computational Physics*, 409:109340, 2020. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2020.109340. URL https://www.sciencedirect.com/science/article/pii/S0021999120301145.

P. Guan, S. Chi, J. Chen, T. Slawson, and M. Roth. Semi-lagrangian reproducing kernel particle method for fragment-impact problems. *International Journal of Impact Engineering*, 38(12):1033–1047, 2011. ISSN 0734-743X. doi: https://doi.org/10.1016/j.ijimpeng.2011.08.001. URL https://www.sciencedirect.com/science/article/pii/S0734743X11001217.

N. Hansen and H. Schreyer. A thermodynamically consistent framework for theories of elastoplasticity coupled with damage. *International Journal of Solids and Structures*, 31 (3):359–389, 1994.

H. C. Heard. Steady-state flow in polycrystalline halite at pressure of 2 kilobars. *Flow and fracture of rocks*, 16:191–209, 1972.

G. Hofstetter. *Computational Engineering*. Springer, 2014.

J. Koester and F. Beckwith. Sierra/rkpm: Current state and applications. Memorandum, December 2017.

J. J. Koester, M. R. Tupek, and S. A. Mitchell. An agile design-to-simulation workflow using a new conforming moving least squares method. Technical Report SAND2019-11851, Sandia National Laboratories, September 2019.

T. A. Laursen. *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer Berlin, Heidelberg, 2003.

J. Lemaitre and R. Desmorat. *Engineering damage mechanics: ductile, creep, fatigue and brittle failures*. Springer Science & Business Media, 2006.

J. Long. *Nuclear Waste Management Procedure NP 19-1 Software Requirements, Revision 20*. Sandia National Laboratories, 2024.

C. Mitchell, E. Matteo, T. Dewers, R. Lander, L. Bonnell, and J. Guilkey. Analysis of simulated rubble piles: Insights for room collapse at the Waste Isolation Pilot Plant (WIPP). Presented at the Fourteenth US/German Workshop on Salt Repository Research, Design and Operation, June 2024.

G. Moutsanidis, J. J. Koester, M. R. Tupek, J.-S. Chen, and Y. Bazilevs. Treatment of near-incompressibility in meshfree and immersed-particle methods. *Computational Particle Mechanics*, 7(2):309–327, 2020. doi: 10.1007/s40571-019-00238-z. URL https://doi.org/10.1007/s40571-019-00238-z.

S. Murakami. *Continuum damage mechanics: a continuum mechanics approach to the analysis of damage and fracture*, volume 185. Springer Science & Business Media, 2012.

N. S. Ottosen and M. Ristinmaa. *The mechanics of constitutive modeling*. Elsevier, 2005.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes*. Cambridge university press Cambridge, third edition, 2007.

S. V. Raj and G. Pharr. Creep substructure formation in sodium chloride single crystals in the power law and exponential creep regimes. *Materials Science and Engineering: A*, 122 (2):233–242, 1989.

B. Reedlunn. Enhancements to the Munson-Dawson model for rock salt. Technical Report SAND2018-12601, Sandia National Laboratories, Albuquerque, NM, USA, November 2018.

B. Reedlunn. A new rock salt constitutive model with back stress and drag stress hardening. In J. de Bresser, M. Drury, P. A. Fokker, M. Gazzani, S. Hangx, A. Niemeijer, and C. Spiers, editors, *The Mechanical Behavior of Salt X*, July 2022a. doi: 10.1201/9781003295808.

B. Reedlunn. *WIPP PA Software QA Plan for Sierra/SolidMechanics Version 5.4.1*. Sandia National Laboratories, 2022b.

B. Reedlunn, G. Moutsanidis, J. Baek, T.-H. Huang, J. Koester, E. Matteo, X. He, K. Taneja, H. Wei, Y. Bazilevs, J.-S. Chen, C. Mitchell, R. Lander, and T. Dewers. Initial simulations of empty room collapse and reconsolidation at the Waste Isolation Pilot Plant. Technical Report SAND2019-15351, Sandia National Laboratories, December 2019.

H. Schreyer. On time integration of viscoplastic constitutive models suitable for creep. *International journal for numerical methods in engineering*, 53(3):637–652, 2002.

O. D. Sherby and P. M. Burke. Mechanical behavior of crystalline solids at elevated temperature. *Progress in Materials Science*, 13:323–390, Jan. 1968. ISSN 0079-6425. doi: 10.1016/0079-6425(68)90024-8. URL http://dx.doi.org/10.1016/0079-6425(68)90024-8.

A. Shterenlikht and N. Alexander. Levenberg–marquardt vs powell's dogleg method for gurson–tvergaard–needleman plasticity model. *Computer Methods in Applied Mechanics and Engineering*, 237–240:1–9, Sept. 2012. ISSN 0045-7825. doi: 10.1016/j.cma.2012.04.018. URL http://dx.doi.org/10.1016/j.cma.2012.04.018.

Sierra/SolidMechanics Team. *Sierra/SolidMechanics 5.20 User's Guide*. Sandia National Laboratories, Albuquerque, NM, USA; Livermore, CA, USA, 5.20.1 edition, June 2024.

J. C. Simo and T. J. Hughes. Computational inelasticity. *New York*, 1998.

C. Spiers, P. Schutjens, R. Brzesowsky, C. Peach, J. Liezenberg, and H. Zwart. Experimental determination of constitutive parameters governing creep of rocksalt by pressure solution. *Geological Society, London, Special Publications*, 54(1):215–227, 1990.

C. Vignes, J. Bean, and B. Reedlunn. Improved modeling of Waste Isolation Pilot Plant disposal room porosity. Technical Report SAND2023-04826, Sandia National Laboratories, June 2023.

H.-P. Wang, C.-T. Wu, and J.-S. Chen. A reproducing kernel smooth contact formulation for metal forming simulations. *Computational Mechanics*, 54(1):151–169, 2014. doi: 10.1007/s00466-014-1015-3. URL https://doi.org/10.1007/s00466-014-1015-3.

A Additional Details of the New Salt Model

A.1 Model Construction

This section will show how the model can be derived within the framework of Rational Thermodynamics [Ottosen and Ristinmaa, 2005, Chapter 21]. This framework, first and foremost, ensures a constitutive model does not violate the second law of thermodynamics. Rare violations of the second law with only certain combinations of parameters, internal state variables, and external state variables can cause problems for years before the root cause is tracked down and fixed. It is, thus, preferable to avoid such intermittent violations during the model design phase using the framework of Rational Thermodynamics. The process of ensuring the model satisfies the second law also forces the model to conform to a certain organizational structure. Complicated models, especially those with damage, can quickly become overwhelming for the analyst and, sometimes, even the developer. Defining the model's free energy, dissipation, and a thermodynamic driving force conjugate to each internal state variable can make a model easier to understand and reduces the tendency to add features in an ad-hoc fashion.

To derive the model within this framework, a few preliminaries are necessary. Continuum damage mechanics employs the concept of a actual damaged configuration, in which the actual stress $\boldsymbol{\sigma}$ exists, and a fictitious undamaged configuration, in which the effective stress $\hat{\boldsymbol{\sigma}}$ exists [Murakami, 2012, Section 2.3]. In the actual damaged configuration, the spherical (mean) stress tensor $\mathring{\boldsymbol{\sigma}}$, spherical stress scalar $\mathring{\sigma}$, deviatoric stress tensor $\tilde{\boldsymbol{\sigma}}$, and equivalent (deviatoric) stress $\tilde{\sigma}$ are defined as

$$\mathring{\boldsymbol{\sigma}} = \mathring{\sigma}\,\boldsymbol{I} \tag{51}$$

$$\mathring{\sigma} = \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\sigma}). \tag{52}$$

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \mathring{\boldsymbol{\sigma}} \tag{53}$$

$$\tilde{\sigma} = \sqrt{\frac{3}{2}\,\tilde{\boldsymbol{\sigma}} : \tilde{\boldsymbol{\sigma}}}. \tag{54}$$

In the fictitious undamaged configuration, the effective stresses are defined as

$$\hat{\mathring{\boldsymbol{\sigma}}} = \mathring{\boldsymbol{\sigma}} \tag{55}$$

$$\hat{\mathring{\sigma}} = \mathring{\sigma} \tag{56}$$

$$\hat{\tilde{\boldsymbol{\sigma}}} = \frac{\tilde{\boldsymbol{\sigma}}}{(1-\delta)^{1/2}} \tag{57}$$

$$\hat{\tilde{\sigma}} = \sqrt{\frac{3}{2}\,\hat{\tilde{\boldsymbol{\sigma}}} : \hat{\tilde{\boldsymbol{\sigma}}}} = \frac{\tilde{\sigma}}{(1-\delta)^{1/2}}. \tag{58}$$

Thus, the damage δ is assumed to affect only the deviatoric part of the stress.

Define the Gibbs free energy, per unit mass, $g = g(\boldsymbol{\sigma}, \delta)$ as

$$M\,g = -\frac{\mathring{\sigma}^2}{2\,E_1} - \frac{\tilde{\sigma}^2}{6\,e_2}, \tag{59}$$

where M is the mass density, E_1 is the elastic bulk modulus, and $e_2 = e_2(\delta)$ is the degraded shear modulus. Prior to any damage, $e_2(0) = E_2$.

To determine the form of e_2, an equivalence of Gibbs free energy between the actual damaged configuration and the fictitious undamaged configuration ($g(\boldsymbol{\sigma}, \delta) = g(\hat{\boldsymbol{\sigma}}, 0)$) is adopted. This equivalence gives

$$e_2 = E_2\,(1 - \delta). \tag{60}$$

Thus, the $(1-\delta)^{1/2}$ in the denominator of Eq. (57) and the energy equivalence above causes the shear modulus to linearly decay with increasing δ.

Although Eq. (60) is a straightforward result, the relationship between the elastic stiffness tensor and damage is not so simple if the damage is a second or fourth-order tensor. As demonstrated in Murakami [2012, Section 2.4], energy equivalences ensure the degraded elastic stiffness tensor remains symmetric, while strain equivalences result in un-symmetric stiffness tensors that are typically symmeterized in an ad-hoc manner. Energy equivalences, therefore, are preferred by the authors in case higher order damage tensors are employed in the future. A consequence of energy equivalences, though, is the strain in the actual damaged configuration is also mapped into an effective strain in the fictitious undamaged configuration, as will be shown shortly.

With the Gibbs free energy fully defined, Coleman's relations between conjugate quantitites can be computed. First, decompose the strain $\boldsymbol{\varepsilon}$ into elastic $\boldsymbol{\varepsilon}^{\mathrm{el}}$ and viscoplastic parts $\boldsymbol{\varepsilon}^{\mathrm{vp}}$:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathrm{el}} + \boldsymbol{\varepsilon}^{\mathrm{vp}}. \tag{61}$$

Following the usual thermodynamic arguments (see Ottosen and Ristinmaa [2005, Section 21.2] and Murakami [2012, Section 3.2.3], for example), the elastic constitutive relation is

$$-M\,\frac{\partial g}{\partial \boldsymbol{\sigma}} = \boldsymbol{\varepsilon}^{\mathrm{el}} = \frac{\mathring{\boldsymbol{\sigma}}}{3\,E_1} + \frac{\tilde{\boldsymbol{\sigma}}}{2\,E_2\,(1-\delta)}, \tag{62}$$

and the damage driving force is

$$-M\,\frac{\partial g}{\partial \delta} = d = \frac{\tilde{\sigma}^2}{6\,E_2\,(1-\delta)^2}. \tag{63}$$

Equation (63) indicates that the decrease in elastic energy, due to decreasing the shear modulus, with increasing damage gives rise to the damage driving force. As shown in Murakami [2012, Section 4.3.1], d is proportional to the deviatoric strain energy release rate from fracture mechanics when $\tilde{\sigma}$ is held constant.

The internal dissipation is used to motivate the relationship between the viscoplastic strain in the fictitious undamaged configuration and in the actual damaged configuration, as proposed in Hansen and Schreyer [1994, Section 2.2.2]. Again following the usual thermodynamic arguments (see Ottosen and Ristinmaa [2005, Section 21.2]), the internal dissipation inequality, expressed in the actual damaged configuration, is

$$\mathscr{d}^{\mathrm{int}} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} + d\,\dot{\delta} \geq 0. \tag{64}$$

The effective viscoplastic strain rate in the fictitious undamaged configuration is defined as

$$\dot{\hat{\varepsilon}}^{\mathrm{vp}} = (1-\delta)^{1/2}\,\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} \tag{65}$$

in order to cleanly re-express the internal dissipation inequality in the fictitious undamaged configuration as

$$\mathscr{d}^{\mathrm{int}} = \hat{\boldsymbol{\sigma}} : \dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}} + d\,\dot{\delta} \geq 0. \tag{66}$$

Note that Eq. (65) differs from the more commonly employed relationships $\hat{\tilde{\boldsymbol{\varepsilon}}} = (1-\delta)^{1/2}\,\tilde{\boldsymbol{\varepsilon}}$ and $\hat{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = (1-\delta)^{1/2}\,\boldsymbol{\varepsilon}^{\mathrm{vp}}$. The more common relationships ensure that conjugate stresses and strains in the actual damaged configuration remain conjugate in the fictitious undamaged configuration. The relationship in Eq. (65), in contrast, allows one to cleanly express evolution equations for strain-type internal state variables entirely within the fictitious undamaged configuration, as will be shown shortly.

Many ways exist to ensure the internal state variable evolution equations satisfy the internal dissipation inequality. Following Chaboche [1997, Section 3.3] and Hansen and Schreyer [1994, Section 2.2.4], we adopt a flexible approach that permits multiple dissipation potentials, each with its own weighting function. The viscoplastic and damage dissipation potentials are defined as

$$\mathscr{D}^{\mathrm{vp}} = \hat{\tilde{\sigma}} \tag{67}$$

$$\mathscr{D}^{\mathrm{dm}} = d, \tag{68}$$

from which the evolution equations are simply

$$\dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}} = w^{\mathrm{vp}}\,\frac{\partial \mathscr{D}^{\mathrm{vp}}}{\partial \hat{\boldsymbol{\sigma}}} = w^{\mathrm{vp}}\,\frac{\partial \hat{\tilde{\sigma}}}{\partial \hat{\boldsymbol{\sigma}}} \tag{69}$$

$$\dot{\delta} = w^{\mathrm{dm}}\,\frac{\partial \mathscr{D}^{\mathrm{dm}}}{\partial d} = w^{\mathrm{dm}} \tag{70}$$

where w^{vp} and w^{dm} are the associated weighting functions. The equivalent stress in the fictitious undamaged configuration is a homogeneous function of degree one with respect to $\hat{\boldsymbol{\sigma}}$, such that

$$\hat{\boldsymbol{\sigma}} : \frac{\partial \hat{\tilde{\sigma}}}{\partial \hat{\boldsymbol{\sigma}}} = \hat{\tilde{\sigma}} \tag{71}$$

Insertion of Eqs. (69) to (71) into Eq. (66) gives

$$\mathscr{d}^{\mathrm{int}} = \hat{\tilde{\sigma}}\,w^{\mathrm{vp}} + d\,w^{\mathrm{dm}} \geq 0. \tag{72}$$

Inspection shows that $\hat{\tilde{\sigma}} \geq 0$ and $d \geq 0$, so one can trivially enforce the internal dissipation inequality by ensuring $w^{\mathrm{vp}} \geq 0$ and $w^{\mathrm{dm}} \geq 0$.

The weighting function w^{vp} can be given meaning by adopting an equivalence between the multi-axial viscoplastic work rate and the uniaxial viscoplastic work rate:

$$\hat{\boldsymbol{\sigma}} : \dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}} = \hat{\tilde{\sigma}}\,\dot{\hat{\tilde{\varepsilon}}}^{\mathrm{vp}}, \tag{73}$$

where $\dot{\hat{\varepsilon}}^{\mathrm{vp}}$ is the equivalent viscoplastic strain rate in the fictitious undamaged configuration. Combine Eqs. (69), (71) and (73) to obtain

$$w^{\mathrm{vp}} = \dot{\hat{\varepsilon}}^{\mathrm{vp}} \tag{74}$$

$$\dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}} = \dot{\hat{\varepsilon}}^{\mathrm{vp}} \frac{\partial \hat{\tilde{\sigma}}}{\partial \hat{\boldsymbol{\sigma}}}. \tag{75}$$

Additionally, the relationship between $\dot{\hat{\varepsilon}}^{\mathrm{vp}}$ and $\dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}}$ can be derived by inserting $\partial\,\hat{\tilde{\sigma}}\,/\,\partial\,\hat{\boldsymbol{\sigma}} = 3/2\,\hat{\tilde{\boldsymbol{\sigma}}}/\hat{\tilde{\sigma}}$ into Eq. (75), double contracting with $2/3\,\hat{\boldsymbol{\varepsilon}}^{\mathrm{vp}}$, and employing Eq. (58) to obtain

$$\dot{\hat{\varepsilon}}^{\mathrm{vp}} = \sqrt{\frac{2}{3}\,\dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}} : \dot{\hat{\boldsymbol{\varepsilon}}}^{\mathrm{vp}}}. \tag{76}$$

Now that Eqs. (74) and (76) are established, one can elect to define $\dot{\hat{\varepsilon}}^{\mathrm{vp}}$ as

$$\dot{\hat{\varepsilon}}^{\mathrm{vp}} = \dot{\hat{\varepsilon}}^{\mathrm{ps}} + \dot{\hat{\varepsilon}}^{\mathrm{dg}} \tag{77}$$

$$\dot{\hat{\varepsilon}}^{\mathrm{ps}} = P_1\,\hat{\tilde{\sigma}}^{P_2} \tag{78}$$

$$\dot{\hat{\varepsilon}}^{\mathrm{dg}} = G_1 \left[\sinh\left(\frac{\hat{\tilde{\sigma}}}{G_2}\right)\right]^{G_3} \tag{79}$$

The advantage of adopting Eq. (65) is now evident: the effective pressure solution strain rate $\dot{\hat{\varepsilon}}^{\mathrm{ps}}$ and effective dislocation glide strain rate $\dot{\hat{\varepsilon}}^{\mathrm{dg}}$ have the same form as one would use for intact, undamaged, material. Note that the parameters P_1, G_1, G_2 are all required to be greater than zero to ensure $w^{\mathrm{vp}} = \dot{\hat{\varepsilon}}^{\mathrm{vp}} \geq 0$.

These viscoplastic strain rates can be easily transformed into the actual damaged configuration. Insert Eq. (65) into Eq. (76) and let the equivalent viscoplastic strain rate in the actual damaged configuration be defined as

$$\dot{\tilde{\varepsilon}}^{\mathrm{vp}} = \sqrt{\frac{2}{3}\,\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} : \dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}}} \tag{80}$$

to obtain

$$\dot{\hat{\varepsilon}}^{\mathrm{vp}} = (1-\delta)^{1/2}\,\dot{\tilde{\varepsilon}}^{\mathrm{vp}}. \tag{81}$$

Transform the flow rule by inserting Eqs. (57), (58), (65) and (81) into Eq. (75), resulting in

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{vp}} = \dot{\tilde{\varepsilon}}^{\mathrm{vp}} \frac{\partial \tilde{\sigma}}{\partial \boldsymbol{\sigma}}. \tag{82}$$

Let the equivalent pressure solution and dislocation glide strain rates transform as

$$\dot{\hat{\varepsilon}}^{\mathrm{ps}} = (1-\delta)^{1/2}\,\dot{\tilde{\varepsilon}}^{\mathrm{ps}} \tag{83}$$

$$\dot{\hat{\varepsilon}}^{\mathrm{dg}} = (1-\delta)^{1/2}\,\dot{\tilde{\varepsilon}}^{\mathrm{dg}} \tag{84}$$

and then employ Eq. (58) in Eqs. (77) to (79) to receive

$$\dot{\tilde{\varepsilon}}^{\,\mathrm{vp}} = \dot{\tilde{\varepsilon}}^{\,\mathrm{ps}} + \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \tag{85}$$

$$\dot{\tilde{\varepsilon}}^{\,\mathrm{ps}} = \frac{P_1}{(1-\delta)^{1/2}} \left[\frac{\tilde{\sigma}}{(1-\delta)^{1/2}}\right]^{P_2} \tag{86}$$

$$\dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} = \frac{G_1}{(1-\delta)^{1/2}} \left\{\sinh\left[\frac{\tilde{\sigma}}{G_2\,(1-\delta)^{1/2}}\right]\right\}^{G_3}. \tag{87}$$

To complete the model formulation, elect to define the damage weighting function as

$$w^{\mathrm{dm}} = D_1 \left\langle (6\,E_2\,d)^{1/2} - \frac{\tilde{\sigma}^{\,\mathrm{db}}}{1-\delta} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \left[1 - \left(\frac{\delta}{D_5}\right)^{D_6}\right] \tag{88}$$

$$\tilde{\sigma}^{\,\mathrm{db}} = D_2\,(\mathring{\sigma} - D_3)\frac{1-\delta}{1-D_4\,\delta}. \tag{89}$$

The parameters D_1 and D_5 must be greater than zero and $\delta \leq D_5$ to ensure $w^{\mathrm{dm}} \geq 0$. Combine Eqs. (63), (70) and (88) to obtain

$$\dot{\delta} = \frac{D_1}{1-\delta} \left\langle \tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \left[1 - \left(\frac{\delta}{D_5}\right)^{D_6}\right]. \tag{90}$$

A.2 Alternative Damage Evolution Equation

In hindsight, it may have been slightly more elegant and slightly more numerically robust to define the damage weighting function as

$$w^{\mathrm{dm}} = D_1 \left\langle \hat{\tilde{\sigma}} - \hat{\tilde{\sigma}}^{\,\mathrm{db}} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \left[1 - \left(\frac{\delta}{D_5}\right)^{D_6}\right] \tag{91}$$

$$\hat{\tilde{\sigma}}^{\,\mathrm{db}} = \frac{D_2\,(\mathring{\sigma} - D_3)}{(1-\delta)^{1/2}} \frac{1-\delta}{1-D_4\,\delta} \tag{92}$$

such that one could combine Eqs. (58), (70), (79), (91) and (92) to obtain

$$\dot{\delta} = D_1 \left\langle \tilde{\sigma} - \tilde{\sigma}^{\,\mathrm{db}} \right\rangle \dot{\tilde{\varepsilon}}^{\,\mathrm{dg}} \left[1 - \left(\frac{\delta}{D_5}\right)^{D_6}\right] \tag{93}$$

$$\tilde{\sigma}^{\,\mathrm{db}} = D_2\,(\mathring{\sigma} - D_3)\,\frac{1-\delta}{1-D_4\,\delta}. \tag{94}$$

This alternative formulation more cleanly maps back and forth between the undamaged and damaged configurations, though not perfectly due to the $1-(\delta/D_5)^{D_6}$ factor. Eq. (93) may also be slightly easier to integrate since it lacks the $(1-\delta)^{-1}$ factor present in Eq. (90), which causes large damage rates when δ approaches 1. This damage evolution equation, however, was not used in any simulations presented herein. It is simply noted for potential future use.

A.3 Extension of Radial Return Mapping to Include Damage

The Radial Return Mapping Algorithm is a well known method to integrate the differential equations for associative flow von Mises plasticity and viscoplasticity. As shown in Simo and Hughes [1998, Section 3.3.1 and 3.7.1], it discretizes the differential equations into a system of non-linear algebraic equations, and reduces this system from six equations down to one equation that can be solved in an iterative manner. This section extends the method to incorporate isotropic damage that only affects the deviatoric behavior, as presented in Section 3.2.

The differential equations in Section 3.2 are first reduced to a system of seven non-linear algebraic equations. To begin, Eq. (30) is converted to rate form and both Eqs. (30) and (35) are discretized with backwards Euler approximations (continuous variables are replaced with their value at the current step n and derivatives are replaced with first order finite difference approximations) to obtain

$$\tilde{\boldsymbol{\sigma}}_n = \tilde{\boldsymbol{\sigma}}_{n-1} + 2\,E_2\,(1-\delta_n)\,(\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{\mathrm{vp}}) - 2\,E_2\,\Delta\delta(\tilde{\boldsymbol{\varepsilon}}_n - \boldsymbol{\varepsilon}_n^{\mathrm{vp}}) \tag{95}$$

$$\delta_n = \delta_{n-1} + \frac{D_1}{1-\delta_n}\left\langle \tilde{\sigma}_n - D_2\,(\mathring{\sigma}_n - D_3)\frac{1-\delta_n}{1-D_4\,\delta_n}\right\rangle \Delta\tilde{\varepsilon}^{\,\mathrm{dg}}\left[1-\left(\frac{\delta_n}{D_5}\right)^{D_6}\right]. \tag{96}$$

where $\Delta x = x_n - x_{n-1}$. Next, Eqs. (30) and (31) are discretized and substituted into Eq. (95) to receive

$$\tilde{\boldsymbol{\sigma}}_n = \tilde{\boldsymbol{\sigma}}_{n-1} + 2\,E_2\,(1-\delta_n)\,(\Delta\boldsymbol{\varepsilon} - \frac{3\,\Delta\tilde{\varepsilon}^{\,\mathrm{vp}}}{2\,\tilde{\sigma}_n}\,\tilde{\boldsymbol{\sigma}}_n) - \frac{\Delta\delta_n}{1-\delta_n}\,\tilde{\boldsymbol{\sigma}}_n. \tag{97}$$

Note that substituting Eq. (30) eliminates the need to compute $\tilde{\boldsymbol{\varepsilon}}_n$ and $\tilde{\boldsymbol{\varepsilon}}_n^{\mathrm{vp}}$, which requires one to store $\tilde{\boldsymbol{\varepsilon}}_{n-1}$ and $\tilde{\boldsymbol{\varepsilon}}_{n-1}^{\mathrm{vp}}$. The seven unknowns to solve for are now the six components of $\tilde{\boldsymbol{\sigma}}_n$ and δ_n.

The core Radial Return Mapping concepts are extended to include damage by defining the trial stress $\check{\boldsymbol{\sigma}}_n$ for step n as

$$\check{\boldsymbol{\sigma}}_n = \tilde{\boldsymbol{\sigma}}_{n-1} + 2\,E_2\,(1-\delta_n)\,\Delta\boldsymbol{\varepsilon} \tag{98}$$

such that Eq. (97) becomes

$$\tilde{\boldsymbol{\sigma}}_n = \check{\boldsymbol{\sigma}}_n - 3\,E_2\,(1-\delta_n)\,\frac{\Delta\tilde{\varepsilon}^{\,\mathrm{vp}}}{\tilde{\sigma}_n}\,\tilde{\boldsymbol{\sigma}}_n - \frac{\Delta\delta_n}{1-\delta_n}\,\tilde{\boldsymbol{\sigma}}_n. \tag{99}$$

Rearranging Eq. (99) as

$$\check{\boldsymbol{\sigma}}_n = \left[1 - 3\,E_2\,(1-\delta_n)\,\frac{\Delta\tilde{\varepsilon}^{\,\mathrm{vp}}}{\tilde{\sigma}_n} - \frac{\Delta\delta_n}{1-\delta_n}\right]\,\tilde{\boldsymbol{\sigma}}_n, \tag{100}$$

shows that $\check{\boldsymbol{\sigma}}_n$ is aligned with $\tilde{\boldsymbol{\sigma}}_n$, such that

$$\frac{\tilde{\boldsymbol{\sigma}}_n}{\tilde{\sigma}_n} = \frac{\check{\boldsymbol{\sigma}}_n}{\check{\sigma}_n}, \tag{101}$$

where $\tilde{\sigma}_n$ is the discretized form of Eq. (24) and the trial equivalent stress is

$$\check{\sigma}_n = \sqrt{\frac{3}{2}\,\check{\boldsymbol{\sigma}}_n : \check{\boldsymbol{\sigma}}_n}. \tag{102}$$

The alignment expressed in Eq. (101) allows one to double contract both sides of Eq. (99) with $3/2\,\tilde{\boldsymbol{\sigma}}_n/\tilde{\sigma}_n$ to give

$$\tilde{\sigma}_n = \check{\sigma}_n - 3\,E_2\,(1-\delta_n)\,\Delta\tilde{\varepsilon}^{\,\mathrm{vp}} - \frac{\Delta\delta_n}{1-\delta_n}\,\tilde{\sigma}_n. \tag{103}$$

Thus, the seven equations for $\tilde{\boldsymbol{\sigma}}_n$ and δ_n in Eqs. (95) and (96) are reduced to two non-linear equations for $\tilde{\sigma}_n$ and δ_n in Eq. (103) and (96).

One typically uses the following process to integrate the differential equations:

1. The old deviatoric stress $\tilde{\boldsymbol{\sigma}}_{n-1}$, the total strain rate $\dot{\boldsymbol{\varepsilon}}_n$, and time increment Δt are supplied to the constitutive model implementation by the host code.
2. Initial guesses for $\tilde{\sigma}_n$ and δ_n are made. (See Section A.4 for further discussion.)
3. The trial deviatoric stress $\check{\boldsymbol{\sigma}}_n$ and trial equivalent stress $\check{\sigma}_n$ are computed using Eqs. (98) and (102).
4. An iterative solver, such as Newton's method, is used to find $\tilde{\sigma}_n$ and δ_n by solving Eq. (103) and (96). Note that, unlike the classical Radial Return Mapping Algorithm, Eqs. (98) and (102) depend on δ_n, so $\check{\sigma}_n$ must be updated each time the iterative solver updates δ_n.
5. Once satisfactory values of $\tilde{\sigma}_n$ and δ_n have been found, Eqs. (98) and (101) are used to compute $\tilde{\boldsymbol{\sigma}}_n$
6. The internal state variable δ_n and any other variables of interest are stored for the next step, and $\tilde{\boldsymbol{\sigma}}_n$ is returned to the host code.

A.4 Initial Guess

The salt model implementation used the strain rate $\dot{\boldsymbol{\varepsilon}}$ and time increment Δt input into the model to construct the initial guess for the equivalent stress $\tilde{\sigma}_{\mathrm{ig}}$ and damage δ_{ig} in the following manner:

1. The equivalent viscoplastic strain rate $\dot{\tilde{\varepsilon}}^{\,\mathrm{vp}}$ is assumed to be equal to the deviatoric strain rate input into the material model: $\dot{\tilde{\varepsilon}}^{\,\mathrm{vp}} = \dot{\tilde{\varepsilon}} = \sqrt{2/3\,\dot{\tilde{\boldsymbol{\varepsilon}}} : \dot{\tilde{\boldsymbol{\varepsilon}}}}$.
2. A test state is computed, in which the damage does not evolve, according to the following steps:

(a) The test damage is simply the damage from the previous converged step ($\delta_{\text{test}} = \delta_{n-1}$).

(b) The test stress $\tilde{\sigma}_{\text{test}}$, corresponding to saturated (fully-developed) viscoplastic flow, is estimated. (The following approach avoids solving Eqs. (32) to (34) for $\tilde{\sigma} = \tilde{\sigma}_{\text{test}}$, which does not have an analytical solution, by ignoring interactions between the pressure solution and dislocation glide terms.)

i. Equation (33) is solved for $\tilde{\sigma} = \tilde{\sigma}^{\text{ps}}_{\text{test}}$, using $\delta = \delta_{\text{test}}$ and $\dot{\tilde{\varepsilon}}^{\text{ps}} = \dot{\tilde{\varepsilon}}^{\text{vp}}$

ii. Equation (34) is solved for $\tilde{\sigma} = \tilde{\sigma}^{\text{dg}}_{\text{test}}$, using $\delta = \delta_{\text{test}}$ and $\dot{\tilde{\varepsilon}}^{\text{dg}} = \dot{\tilde{\varepsilon}}^{\text{vp}}$

iii. The smaller of the two stresses is adopted for the test stress: $\tilde{\sigma}_{\text{test}} = \min(\tilde{\sigma}^{\text{ps}}_{\text{test}},\ \tilde{\sigma}^{\text{dg}}_{\text{test}})$.

(c) The test damage boundary $\tilde{\sigma}^{\text{db}}_{\text{test}}$ is computed from Eq. (36), using $\sigma^{\text{sp}} = 3\,E_1\,\varepsilon^{\text{sp}}$, Eq. (25), and $\delta = \delta_{\text{test}}$.

3. If $\tilde{\sigma}_{\text{test}} < \tilde{\sigma}^{\text{db}}_{\text{test}}$, then the test values are adopted as the initial guesses: $\tilde{\sigma}_{\text{ig}} = \tilde{\sigma}_{\text{test}}$ and $\delta_{\text{ig}} = \delta_{\text{test}}$. If $\tilde{\sigma}_{\text{test}} > \tilde{\sigma}^{\text{db}}_{\text{test}}$ then initial guesses, in which the damage evolves, are computed via the following additional steps:

(a) The initial damage rate $\dot{\delta}_{\text{i}}$ is roughly estimated by utilizing $\tilde{\sigma} = \tilde{\sigma}_{\text{test}}$ and $\delta = \delta_{n-1}$ in Eqs. (34) to (36).

(b) The damage initial guess is computed as $\delta_{\text{ig}} = \min\left(\delta_{n-1} + \dot{\delta}_{\text{i}}\,\Delta t,\ D_5\right)$.

(c) The damaged stress is estimated in the following manner:

i. Equation (33) is solved for $\tilde{\sigma} = \tilde{\sigma}^{\text{ps}}_{\text{dmg}}$, using $\delta = \delta_{\text{ig}}$ and $\dot{\tilde{\varepsilon}}^{\text{ps}} = \dot{\tilde{\varepsilon}}^{\text{vp}}$.

ii. Equation (34) is solved for $\tilde{\sigma} = \tilde{\sigma}^{\text{dg}}_{\text{dmg}}$, using $\delta = \delta_{\text{ig}}$ and $\dot{\tilde{\varepsilon}}^{\text{dg}} = \dot{\tilde{\varepsilon}}^{\text{vp}}$.

iii. The smaller of the two stresses is adopted: $\tilde{\sigma}_{\text{dmg}} = \min(\tilde{\sigma}^{\text{ps}}_{\text{dmg}},\ \tilde{\sigma}^{\text{dg}}_{\text{dmg}})$.

(d) The initial guess for the equivalent stress is scaled as $\tilde{\sigma}_{\text{ig}} = X\,\tilde{\sigma}_{\text{dmg}}$. Preliminary testing found that $X = 0.8$ helped the solver avoid getting stuck in local minima better than $X = 1.0$.

One potential enhancement to the initial guess procedure discussed above would be to compute the $\delta = \delta^{\text{bdb}}$ at which $\tilde{\sigma}$ drops below $\tilde{\sigma}^{\text{db}}$ and then replace the damage initial guess in Step #3b with $\delta_{\text{ig}} = \min\left(\delta_{n-1} + \dot{\delta}_{\text{i}}\,\Delta t,\ D_5,\ \delta^{\text{bdb}}\right)$. Change #2 in Section 3.3 discusses an approach to compute δ^{bdb}, but it requires one to solve a non-linear implicit algebraic equation. Preliminary convergence studies found that the damage initial guess in Step #3b was sufficient for $\mathring{\sigma}_{n-1} < 50$ MPa with the Collapse parameter set, so this enhancement was left for potential future work.

B Mesh and Simulation Input files

The Cubit journal file and Sierra/SM input file used to perform the room collapse and reconsolidation simulation with the manufactured triangular damage band are given herein.

B.0.1 Cubit Journal File

```
# Journal file for manufactured band of damaged material in the diagonal
   over the roof
# Add a block on the symmetry plane to mirror pinballs

reset

# Domain definition for lower horizon [m]
# Domain bottom elevation = {domain_bottom = -54.19}
# Domain top elevation = {domain_top = 52.87}
# Domain width = {domain_width = 20.27}
# Room width = {room_width = 5.03}
# Height of the room = {room_height = 3.96}
# Room bottom elevation = {room_bottom = -6.39}
# Room top elevation = {room_top = room_bottom + room_height}
# Room mid-height elevation = {room_mid_height = room_bottom + 0.5*
   room_height}
# Fillet in room corners = {r_fil = 18.0 * 25.4 / 1000} # 18 in

# Coarse mesh size = {coarse_size = 2.0}
# Fine mesh size = {fine_size = 0.1}
# Domain thickness = {half_thickness = 0.5*fine_size}

create vertex {domain_width}, {half_thickness}, {domain_bottom} # V1
create vertex 0.0, {half_thickness}, {domain_bottom} # V2
create vertex 0.0, {half_thickness}, {room_bottom} # V3

create vertex {room_width - r_fil}, {half_thickness}, {room_bottom} # V4
create vertex {room_width}, {half_thickness}, {room_bottom + r_fil} # V5

create vertex {room_width}, {half_thickness}, {room_top - r_fil} # V6
create vertex {room_width - r_fil}, {half_thickness}, {room_top} # V7

create vertex 0.0, {half_thickness}, {room_top} # V8
create vertex 0.0, {half_thickness}, {domain_top} # V9
create vertex {domain_width}, {half_thickness}, {domain_top} # V10

create curve polyline vertex 1 2 3 4
create curve arc vertex 4 5 radius {r_fil} normal 0 -1 0
create curve polyline vertex 5 6
create curve arc vertex 6 7 radius {r_fil} normal 0 -1 0
create curve polyline vertex 7 8 9 10 1

create surface curve all
sweep surface 1 vector 0 -1 0 distance {2*half_thickness}

# Cut an elliptical domain around the room
create surface ellipse major radius {room_width*3} minor radius {
   room_height*2} yplane
move Surface 13  location 0 {2*half_thickness} {room_mid_height}
   include_merged
webcut volume 1  sweep surface 13  perpendicular inward through_all
```

```
delete surface 13

# Diagonal band cutting the roof with 45 degree angle
# Thickness of the band = {band_thickness = 2 * fine_size}
webcut volume 3 with plane zplane offset {1.2 + band_thickness} rotate 45
   about y center 0 0 0
webcut volume 4 with plane zplane offset 1.2 rotate 45 about y center 0 0
   0

undo group begin
imprint volume all
merge volume all
undo group end

# Split vertical edge of the room to make center vertex for wall center
split curve 74 distance {0.5*(room_height-2*r_fil)}

# Add block to mirror pinballs
brick x {fine_size} y {fine_size} z {domain_top - domain_bottom}
move vol 7 x {-1.5*fine_size} z {0.5 * (domain_bottom + domain_top)}

volume all scheme tetmesh proximity layers off geometry approximation
   angle 15
volume all tetmesh growth_factor 1
set trimesher surface gradation 1.3
set trimesher volume gradation 1.3
trimesher geometry sizing on

volume all except 1 size {fine_size}
volume 1 size {coarse_size}

mesh volume all

# Blocks, nodesets and sidesets

block 1 add volume all except 4 7 name "B_halite"
block 2 add volume 4 name "B_band"
block 3 add volume 7 name "B_mirror"

# {tol = 1e-6}
nodeset 1 add surface with y_coord >= {half_thickness - tol} name "
   NS_front_surf"
nodeset 2 add surface with y_coord <= {tol - half_thickness} name "
   NS_back_surf"
nodeset 3 add surface with x_coord <= {0 + tol} name "NS_left_surf"
nodeset 4 add surface with x_coord >= {domain_width - tol} name "
   NS_right_surf"
nodeset 5 add surface with z_coord >= {domain_top - tol} name "NS_top_surf
   "
nodeset 6 add surface with z_coord <= {domain_bottom + tol} name "
   NS_bot_surf"
nodeset 7 add vertex 66 name "NS_right_wall_ctr"
nodeset 8 add vertex 23 name "NS_floor_ctr"
```

```
nodeset 9 add vertex 28 name "NS_roof_ctr"

sideset 1 add surface with x_coord <= {0 + tol} name "SS_left_surf"
sideset 2 add surface with x_coord >= {domain_width - tol} name "
   SS_right_surf"
sideset 3 add surface with z_coord >= {domain_top - tol} name "SS_top_surf
   "
sideset 4 add surface with z_coord <= {domain_bottom + tol} name "
   SS_bot_surf"
sideset 5 add surface 4 5 7 26 30 37 44 50 name "SS_room_surf"
sideset 6 add surface 4 name "SS_floor"
sideset 7 add surface 5 name "SS_floor_corner"
sideset 8 add surface 37 26 name "SS_wall"
sideset 9 add surface 7 name "SS_roof_corner"
sideset 10 add surface 30 44 50 name "SS_roof"


export mesh "wipp_refined_rounded_corners_thin_band_damage_mirror.tet_{
   fine_size}.g" overwrite

reset
import mesh geometry "wipp_refined_rounded_corners_thin_band_damage_mirror
   .tet_{fine_size}.g" feature_angle 135 merge

# Select active nodes on the fron and back for display
# Needs to be after the reset / import otherwise non-active nodes are
   added by thexing
nodeset 10 add node with y_coord >= {half_thickness - tol} name "
   NS_active_front"
nodeset 11 add node with y_coord <= {tol - half_thickness} name "
   NS_active_back"


thex volume all
export mesh "wipp_refined_rounded_corners_thin_band_damage_mirror.nodal_{
   fine_size}.g" overwrite
```

B.0.2 Sierra/SM Input File

```
begin sierra WIPP Room Closuse

  title Simulation of WIPP Room Closure

  #---------- Variables ---------

  #Define the time periods
  ${t_0_0 = -1.0e-1} #s
  ${t_0_1 = 0.0} #s
  ${t_1_0 = t_0_1} #s
  ${t_1_1 = 1000.0} #s (This is an upper bound for the momentum balance
     time.  The simulation should terminate upon reaching the final
     material time.)
```

```
#Define the scale factors
${s_0_0 = 2.0e-2}
${s_0_1 = 2.0e-2}
${s_1_0 = 2.0e-2}
${s_ramp = 2e9}

#Define the material time periods
${t_m_0_0 = t_0_0 * s_0_0}
${t_m_0_1 = 0.0}
${t_m_1_0 = t_m_0_1} #s
${t_m_1_1 = 86400.0 * 365.25 * 150.0} #s

# Define the material damage limit
$ Damage limit Dlim = {Dlim = 0.95}
$ Primary criterion for bond failure Dfail = {Dfail = 0.999*Dlim}

# Define time periods for variable vp scaling
${t_sep = 0.3} # Estimated time of roof separation
${t_fall = 1.18} # Estimated time of roof contact with floors (
   conservative)

#---------- Directions --------

define direction z with vector 0.0 0.0 1.0
define direction negative_z with vector 0.0  0.0 -1.0
define point origin with coordinates 0.0 0.0 0.0

#---------- Functions ---------

begin function fluid_density_ramp
  #Specify the density to give the appropriate lithostatic pressure at
     room at start of simulation and then ramp it down to zero
  type is analytic
  evaluate expression is "2300.0 / 2.0 * (1.0 - cos(pi * time / {t_0_1 -
      t_0_0}))"
end

#---------- Viscoplastic Rate Scaling Functions ---------

# Sierra/SM calculates user variables at the beginning and the end of
   any time step
# where data is written to the exodus database.  Normally this is fine,
   but if
# variable x is an input and an output then x would get updated twice if
    data were
# written to the exodus database instead of once in a time step.  To
   make sure x is
# only calculated at the beginning (or end) of the time step we compare
   the global
# variable time against a user variable called stored_time.  The global
   variable
# time is always the time at the end of the time step.  (This is a bit
```

```
   weird since
# all other variables can change from the start to the end of the time
   step, but one
# can always get the time at the beginning of a step by calculating t-dt
   .)  As long
# as we update stored_time AFTER x, then the beginning of a time step
   corresponds to
# time != stored_time, and the end of the time step corresponds to
# time == stored_time.  If data is not written to the exodus database,
   then I think
# user variables are calculated only at the beginning of a time step.

begin function stored_time_function
  type is analytic
  expression variable: t_stored = global time
  evaluate expression = "t_stored"
end

begin function vp_rate_scale_factor_function
  #The start time column specifies when the corresponding temperature
     expression
  #starts to be used.
  #(We have to add a small amount to start times to make sure the last
     time step
  #of the previous time period does not use the temperature expression
     for the next
  #time period.)
  type is piecewise analytic
  begin expressions
    {t_0_0}             "{s_0_0}"
    {t_0_1 + 1e-10}     "{s_1_0} + {s_ramp} * time"
    {t_sep}             "max(1, {s_1_0} + {s_ramp} * {t_sep} - {10*
       s_ramp}*(time - {t_sep}))"
    {t_fall + 0.001}    "1 + {s_ramp} * (time - {t_fall})"
  end
end

begin function vp_rate_scale_factor_broadcast_function
  type is analytic
  expression variable: vp_rate_scale_factor = global
     vp_rate_scale_factor
  evaluate expression = "vp_rate_scale_factor"
end

begin function mat_time_function
  type is analytic
  expression variable: t = global time
  expression variable: dt = global timestep
  expression variable: t_stored = global stored_time
  expression variable: t_m = global mat_time
  expression variable: s = global vp_rate_scale_factor
  evaluate expression = "                                    \#
  (t != t_stored) ? (                                        \#
```

```
  t_m + s * dt                                                \#
  ):(                                                             \#
  t_m                                                         \#
  )                                                               \#
  "
end

  begin function size_scale_function
      type is analytic
      expression variable: m = nodal mass
      evaluate expression = "pow(m,{1.0/3.0})"
  end

#---------- Materials ----------

begin material halite

  density = 2300.0

  begin parameters for model power_law_creep_damage_v4

    # Calibration 3: manually changing the damage accumulation rate D1
       until we get a desirable response
    bulk modulus  = 20.7e9 # Pa  compute bulk modulus from E=31e+09 Pa
       and nu = 0.25
    shear modulus = 12.4e9  # Pa
    P1 = 2.930e-18 # 1/sec
    P2 = 1.0
    G1 = 2.959e-7 # 1/sec
    G2 = 43.50e6 # Pa
    G3 = 5.623
    D1 = 8e-6 # Default was 3e-6
    D2 = -1.0
    D3 = 2.0e6 # default is 2e6
    D4 = 1.0
    D5 = {Dlim} # Damage limit
    D6 = 5.0

  end parameters for model power_law_creep_damage_v4

end

begin crk section solid_1

  formulation = nodal_from_thex
  shape function type = spherical
  support size type = relative_local
  support size = 1.1 # Default 1.1
  number of integration subcells = 1
  #thex active nodes = centroids
  #thex active nodes = nodes

  failure method = bond
```

```
  failure state name = damage # Scalar damage variable
  failure criterion = {Dfail}

  bond opening displacement = 1e-6
  bond secondary criterion = 1e-6

  pinball contact = on
  pinball contact type = nodal
  new surface pinball contact = on
  #pinball force scatter type = direct

  pinball penalty = 1e-2
  pinball proximity factor = 1.1
  pinball friction = 0.0
  pinball damping ratio = 0.5

end

#---------- Finite Element Model -----------

begin finite element model room
  Database name = wipp_refined_rounded_corners_thin_band_damage_mirror.
     nodal_0.1.g
  Database type = exodusII

  begin parameters for block b_halite b_band b_mirror
    material halite
    model = power_law_creep_damage_v4
    section = solid_1
  end

end

begin adagio procedure The_Procedure.1

  begin time control

    begin time stepping block p0
      start time = {t_0_0}
      begin parameters for presto region SmRegion.1
        time step scale factor = 0.8
      end
    end

    begin time stepping block p1
      start time = {t_1_0}
      begin parameters for presto region SmRegion.1
        time step scale factor = 0.8
      end
    end

    termination time = {t_1_1}
```

```
end

#---------- Time Step Control ----------


begin presto region SmRegion.1

  use finite element model room

  begin solution termination
    terminate global mat_time > {t_m_1_1}
  end


  #--------- Boundary Conditions ---------

  begin gravity
    include all blocks
    gravitational constant = 9.79  # SAND88-2948 used 9.79 m/s^2 and
       so did the Germans in JPIII.  9.79 m/s^2 is close to the local
       value of gravity at WIPP (9.79483 m/s^2).
    direction = negative_z
  end

  begin initial temperature
    include all blocks
    magnitude = 300.
  end

  begin prescribed temperature
    include all blocks
    function = sierra_constant_function_one
    scale factor = 300.0
  end

  begin pressure
    surface = SS_top_surf
    function = sierra_constant_function_one
    scale factor = 13.57E+06
  end

  begin fixed displacement
    node set = NS_bot_surf
    components = z
  end

  begin fixed displacement
    node set = NS_left_surf
    components = x
  end

  begin fixed displacement
    node set = NS_right_surf
```

```
  components = x
end

begin fixed displacement
  include all blocks
  components = y
end

begin fluid pressure
  surface = SS_room_surf
  active periods = p0
  #Specify which direction corresponds to the depth of fluid.
  fluid surface normal = z
  #The depth function is defined relative to the reference point.
  #See the figure in the fluid pressure section of the Sierra/SM
     manual for more info.
  reference point = origin
  #Set the fluid height, relative to the reference point, to give
     the appropriate lithostatic pressure at room
  #p1 = -15.97e6 Pa
  #p2 = -13.57e6 Pa
  #z1 = -54.19 m
  #z2 = 52.87 m
  #slope = (p2-p1)/(z2-z1)
  #p = slope*(z-z1)+p1
  #z0 = depth = elevation of fluid surface
  #depth = -p1 / slope + z1
  depth = 658.2050833333333 #m = height
  #Specify the function to ramp the fluid density down by time = 0
  density function = fluid_density_ramp
  gravitational constant = 9.79
end

begin fixed displacement
  block = B_mirror
  components = x y z
end

#--------- Initial Conditions ---------

begin initial condition
  block = b_halite b_band
  #It is important to initialize unrotated_stress and not stress,
     because
  #stress is just an output that gets calculated as needed, while
  #unrotated_stress gets stored and used by the constitutive models.
  #As of November 2015, initializing stress does nothing.
  initialize variable name = unrotated_stress
  variable type = element
  subroutine real parameter: top = 52.87
  subroutine real parameter: bot = -54.19
  subroutine real parameter: p1 = -13.57e6
  subroutine real parameter: po = -15.97e6
```

```
  subroutine real parameter: kvert_xx = 1.0
  subroutine real parameter: kvert_yy = 1.0
  subroutine real parameter: kvert_zz = 1.0
  subroutine real parameter: kvert_xy = 0.0
  subroutine real parameter: kvert_yz = 0.0
  subroutine real parameter: kvert_zx = 0.0
  subroutine string parameter: dir = Z
  element block subroutine = geo_is
end

# Model band as intially damaged band
begin initial condition
  block = b_band
  initialize variable name = damage
  variable type = element
  magnitude = {0.95*Dlim}
end initial condition

#--------- Viscoplastic Rate Scaling ---------

# See note above stored_time_function.
begin user variable stored_time
  type = global real length = 1
  global operator = min
  initial value = {t_0_0}
end

begin user variable mat_time
  type = global real length = 1
  global operator = min
  initial value = {t_m_0_0}
end

begin user variable vp_rate_scale_factor
  type = global real length = 1
  global operator = min
  initial value = {s_0_0}
end

begin user output
  block = b_halite b_band
  compute global vp_rate_scale_factor as function
     vp_rate_scale_factor_function
  # Although vp_scale_factor is really a global variable, it needs
     to be at the integration points
  # so the material model can read it.
  compute element vp_rate_scale_factor as function
     vp_rate_scale_factor_broadcast_function
  compute at every step
end

begin user output
  compute global mat_time as function mat_time_function
```

```
  # See note above stored_time_function.  Note that stored_time must
      be calculated
  # AFTER other functions that use it.
  compute global stored_time as function stored_time_function
  compute at every step
end

#--------- User Output Variable Definitions ----------

begin user output
  node set = NS_roof_ctr
  compute global roof_disp as average of nodal physical_displacement
end

begin user output
  node set = NS_floor_ctr
  compute global floor_disp as average of nodal
     physical_displacement
end

begin user output
  node set = NS_right_wall_ctr
  compute global wall_disp as average of nodal physical_displacement
end

begin user output
  surface = SS_room_surf
  surface subroutine = aupst_cavity_volume
  subroutine string parameter: outputVarName = room_volume
  subroutine real parameter: cavitysymmetrypoint_x = 0.0
  subroutine real parameter: cavitysymmetrypoint_y = -0.10479167
  subroutine real parameter: cavitysymmetrypoint_z = -4.41000000
  subroutine real parameter: volumeScale = 1.0
  compute at every step
end

begin user output
  copy element variable damage to nodal variable damage
end

begin user output
  compute nodal size_scale as function size_scale_function
end

#--------- Results Output ---------

begin results output output_1
  database name = %B.e
  database type = exodusII
  #Output data every 1 years of material time
  ${yr = -1}
  {Loop(151)}
  ${yr += 1}
```

```
      additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr *
         86400.0 * 365.25))) / s_ramp}
      {EndLoop}
      global time as mom_bal_time
      global mat_time as time
      nodal variables = physical_displacement as displacement
      nodal variables = node_failure_status
      nodal variables = force_internal
      nodal variables = force_external
      nodal variables = acceleration
      nodal variables = pinball_force
      nodal variables = pinball_status
      nodal variables = pinball_radius
      nodal variables = pinball_force_scatter
      element variables = stress
      element variables = unrotated_stress
      element variables = hydrostatic_stress
      element variables = eq_ps_strain
      element variables = eq_dg_strain
      element variables = cell_id
      element variables = sub_cell_id
      element variables = damage
      nodal variables = damage
    end results output output_1

    #--------- History Output ---------

    begin history output output_3
      database name = %B.h
      database type = exodusII
      At time {t_0_0}, increment = 0.001
      global time as mom_bal_time
      global wall_clock_time as run_time
      global mat_time as time
      global vp_rate_scale_factor
      global roof_disp
      global floor_disp
      global wall_disp
      global room_volume
      global external_energy
      global internal_energy
      global kinetic_energy
      global cpu_time
      global cpu_time_per_step
      global wall_clock_time_per_step
    end

  end

 end

end
```

External Distribution:

U.S. Department of Energy Carlsbad Field Office
Attn: Anderson Ward and George Basabilvazo
4021 National Parks Highway
Carlsbad, NM 88220

ZG-Geo Ltd.
Attn: Gordan Gjerapic
9426 Morning Glory Way
Highlands Ranch, CO 80130

Aperi Computational Mechanics Consulting
Attn: Jacob Koester
34 Riverside Dr.
Peralta, NM 87042

Internal Distribution:

MS 1318	K. Peterson	Org. 1442
MS 0840	W. Witkowski	Org. 1550
MS 0840	J. Bean	Org. 1554
MS 0840	C. Mitts	Org. 1554
N/A	J. Coulibaly	Org. 1558
MS 0840	W. Huber	Org. 1558
MS 0840	S. Nelson	Org. 1558
MS 0840	C. Vignes	Org. 1558
MS 1395	M. Rascon	Org. 8138
MS 0747	M. Mills	Org. 8842
MS 0779	K. Kuhlman	Org. 8842
MS 0779	E. Matteo	Org. 8842
MS 0779	C. Mitchell	Org. 8842
MS 1395	G. Roselle	Org. 8880
MS 1395	S. King	Org. 8881
MS 1395	S. Wagner	Org. 8881
MS 0735	C. Choens	Org. 8914
MS 0735	C. Herrick	Org. 8914