SANDIA REPORT
SAND2025-12225
Printed September 2025

Sandia National Laboratories

An Assessment of the Error Due to Computing Waste Isolation Pilot Plant Porosity Using the Porosity Response Surface Approach

Benjamin Reedlunn, Seth King, Brad Day, Craig Hamel, James Bean

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

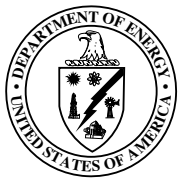
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods





ABSTRACT

The Waste Isolation Pilot Plant Performance Assessment (WIPP PA) must predict the likelihood that radionuclides will escape into the biosphere via mechanisms that depend on geohydraulic flow. Ideally, one would predict the geohydraulic flow using coupled geohydraulic and geomechanical simulations, but such coupled simulations are not computationally tractable. Instead, Sandia has historically used a look-up table of porosities for a given fluid pressure and time, called the porosity response surface, but this approach can introduce porosity errors because it largely ignores the porosity's dependence on the past fluid pressure history. This report discusses efforts to quantify these porosity errors for both the legacy and new porosity response surfaces. Six hundred different fluid pressure histories were fed through the legacy/new geomechanical model and the legacy/new porosity response surface to generate six hundred porosity error histories. The error associated with the legacy porosity surface was substantial, while the error associated with the new porosity surface was typically small, except when fluid pressures exceeded the lithostatic pressure at the repository. In response to the errors at high pressures, a preliminary study of the WIPP PA's sensitivity to these porosity errors was conducted. The study found that reducing the porosity errors at high pressures negligibly affected predictions of radionuclide releases. Finally, an initial machine-learned model for porosity was developed. This ML model significantly reduced the porosity error at high pressures, but sizable errors remained, so more development is necessary before coupling an ML model to the geohydraulic model.

This page intentionally left blank.

ACKNOWLEDGMENT

The authors would like to acknowledge two individuals for their contributions. Chet Vignes helpfully adapted some of the scripts and Sierra/SM input files developed in Vignes et al. (2023) to the needs of this study. Clifford Hansen provided many useful comments when he formally reviewed a draft version of this report.

This page intentionally left blank.

CONTENTS

Acknowledgement **5**

Symbol Glossary **12**

1. Introduction **13**

2. Porosity Error Assessment **19**
2.1. Procedure 19
2.2. Results 22
2.2.1. Recreated Legacy Porosity Response Surface 22
2.2.2. New Porosity Response Surface 24

3. Preliminary Performance Assessment Sensitivity Study **33**
3.1. Conditionally Limited Porosity Surface 33
3.2. Comparison of Radionuclide Releases 35

4. Preliminary Machine Learned Model for Porosity **41**

5. Summary **45**

Bibliography **47**

Appendices **49**

A. Explicit Time Integration With Viscoplastic Scaling **49**

B. Recreated Legacy Porosity Response Surface **53**

C. Sierra/SM Input Files **57**
C.1. Recreated Legacy Geomechanical Model 57
C.2. New Geomechanical Model 70

This page intentionally left blank.

LIST OF FIGURES

Figure 1-1. A schematic of fully coupled geohydraulic-geomechanical simulations. 13
Figure 1-2. Schematic of the process used to recreate the legacy porosity response surface. 14
Figure 1-3. Schematic of the uncoupled porosity surface approach. 16
Figure 1-4. Schematic of the process used to generate the new porosity response surface. . . 17

Figure 2-1. A schematic of the procedure used to assess the error with the uncoupled porosity surface approach. (Note that the porosity surface, pressure histories, geomechanical model schematic, and porosity histories correspond to the new porosity surface, but the same procedure was used to asses the error for the recreated legacy porosity surface.) 20
Figure 2-2. Fluid pressure histories overlaid on the recreated legacy porosity response surface (top row of plots), as well as porosities predicted by the recreated legacy geomechanical model and the recreated legacy porosity surface (bottom row of plots) for representative cases with low, medium, and high pressures. 23
Figure 2-3. The porosity error associated with the recreated legacy porosity surface for all 600 cases combined. 25
Figure 2-4. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the new geomechanical model and the new legacy porosity surface (bottom row of plots) for representative cases with low, medium, and high pressures. 26
Figure 2-5. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the new geomechanical model and the new porosity surface (bottom row of plots) for two representative, yet less common, cases. 28
Figure 2-6. The porosity error associated with the new porosity surface, across all 600 cases, is compared to the porosity error when excluding cases where fluid pressure exceeds lithostatic pressure. Note that the error ranges in the lower plot of (b) are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}), pink for range($e^{\mathrm{ps}}_{\mathrm{p_f}<\mathrm{p_{ls}}}$), and lavender where range(e^{ps}) and range($e^{\mathrm{ps}}_{\mathrm{p_f}<\mathrm{p_{ls}}}$) overlap. 29
Figure 2-7. Fluid pressure histories overlaid on the new porosity response surface (top plot), porosities predicted by the new geomechanical model and the new porosity surface (middle plot), and the porosity error associated with the new porosity surface (bottom plot) of all 600 cases during the first 0.6×10^3 yr. 30

Figure 3-1. The porosity error associated with the new porosity surface and the approximate porosity error associated with the conditionally limited new porosity surface for all 600 cases combined. Note that the error ranges are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}) (or range($\breve{e}^{\mathrm{ps}}$)), pink for range(e^{clps}) (or range($\breve{e}^{\mathrm{clps}}$)), and lavender where range(e^{ps}) and range(e^{clps}) (or range($\breve{e}^{\mathrm{ps}}$) and range($\breve{e}^{\mathrm{clps}}$)) overlap. ... 34
Figure 3-2. Replicate 1 brine pressure, brine saturation, and porosity individual histories and mean in the representative waste panel for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathrm{clps}} \leq \phi_{\max}$ if $p_{\mathrm{f}} > p_{\mathrm{ls}}$) subject to Scenario S1-BF ... 36
Figure 3-3. Replicate 1 brine pressure, brine saturation, and porosity individual histories and mean in the representative waste panel for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathrm{clps}} \leq \phi_{\max}$ if $p_{\mathrm{f}} > p_{\mathrm{ls}}$) subject to Scenario S2-BF ... 37
Figure 3-4. Replicate 1 brine pressure, brine saturation, and porosity means in the representative waste panel, north rest-of-repository, and west rest-of-repository for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathrm{clps}} \leq \phi_{\max}$ if $p_{\mathrm{f}} > p_{\mathrm{ls}}$) subject to Scenarios S1-BF, S2-BF, S4-BF, and S6-BF ... 39
Figure 3-5. Replicate 1 direct brine, spallings, and total mean releases for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathrm{clps}} \leq \phi_{\max}$ if $p_{\mathrm{f}} > p_{\mathrm{ls}}$) . 40

Figure 4-1. Simple schematics of a NN and an RNN. ... 41
Figure 4-2. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the preliminary ML model, the new geomechanical model, and the new legacy porosity surface (bottom row of plots) for select cases. ... 42
Figure 4-3. The porosity error associated with the new porosity surface and the machine learned model for porosity for all 600 cases combined. Note that the error ranges in the lower plot of (b) are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}), light orange for range(e^{ml}), and bluish orange where range(e^{ps}) and range(e^{ml}) overlap. ... 43

Figure A-1. Ratio of kinetic energy over internal energy for the recreated legacy geomechanical model simulations in each of the six PA scenarios. ... 50
Figure A-2. Ratio of kinetic energy over internal energy for the new geomechanical model simulations in each of the six geohydraulic scenarios. ... 51

Figure B-1. Comparison of porosity and fluid pressure histories from the original legacy model and two recreations of the legacy model for a lower horizon disposal room. 54
Figure B-1. (Continued) Comparison of porosity and fluid pressure histories from the original legacy model and two recreations of the legacy model for a lower horizon disposal room. ... 55

SYMBOL GLOSSARY

Symbol	Description
t	Time
t_n	Time at step n
t^{mb}	Momentum balance time
V_{r}	Original volume of room
θ	Temperature
n_{f}	Number of moles of gas (fluid) generated
$n_{\mathrm{f}}^{\mathrm{pot}}$	Maximum potential number of moles of gas (fluid) generated
v_{f}	Volume available to store fluid within the room
R	Universal gas constant
f	Fraction of maximum expected gas generation history
f^{rel}	Relative frequency of a given quantity
$\boldsymbol{s}_n$	Vector of internal state variables at step n
p_{f}	Fluid (gas and/or brine) pressure
$p_{\mathrm{f},n}$	Fluid pressure at step n
p_{ls}	Lithostatic pressure at the repository
p_{rc}	Back pressure supplied by the room contents
$p_{\mathrm{f}}^{\mathrm{fc}}$	Fluid pressure computed by a hypothetical fully coupled geomechanical–geohydraulic model
$p_{\mathrm{f}}^{\mathrm{cl}}$	Fluid pressure hypothetically computed by the conditionally limited porosity surface
ϕ	Porosity (generic)
ϕ_n	Porosity at step n
ϕ^{ps}	Porosity computed by either the recreated legacy or new porosity surface
ϕ_{max}	Maximum porosity permitted by the conditionally limited porosity surface
ϕ^{clps}	Porosity computed using the conditionally limited porosity surface
ϕ^{gm}	Porosity computed by either the recreated legacy or new geomechanical model
ϕ^{clgm}	Porosity hypothetically computed by feeding $p_{\mathrm{f}}^{\mathrm{cl}}$ histories into the new geomechanical model
ϕ^{ml}	Porosity computed by the machine-learned model for porosity
ϕ^{fc}	Porosity computed by a hypothetical fully coupled geomechanical–geohydraulic model
e	Porosity error (generic)
e^{ps}, $\breve{e}^{\mathrm{ps}}$	Porosity error and signed error due to using either the recreated legacy or new porosity surface
$e^{\mathrm{ps}}_{p_{\mathrm{f}}<p_{\mathrm{ls}}}$	Porosity error due to using the new porosity surface when cases where fluid pressure exceeds lithostatic pressure are excluded
e^{clps}, $\breve{e}^{\mathrm{clps}}$	Porosity error and signed error due to using the conditionally limited porosity surface
e^{ml}	Porosity error due to using the machine-learned model for porosity

1. INTRODUCTION

Waste disposal rooms at the Waste Isolation Pilot Plant (WIPP) gradually close, principally due to shear stresses in the surrounding rock salt. The room closes relatively quickly at first, but the closure slows down as the back pressure supplied by the room contents asymptotically approaches lithostatic pressure p_{ls} at the WIPP, causing the shear stresses to become negligible. The room contents may include waste containers, brine, and gases. The amount of fluid (brine and/or gas) depends on the flow of fluids in/out of the room, as well as the rate of various geochemical reactions occurring within the room. If the back pressure from the room contents exceeds p_{ls}, which can occur if fluids cannot escape the room fast enough, then the room will expand after it has partially or fully closed. Room closure/expansion affects the porosity (void volume) available to store and transport brine and gases within disposal rooms.

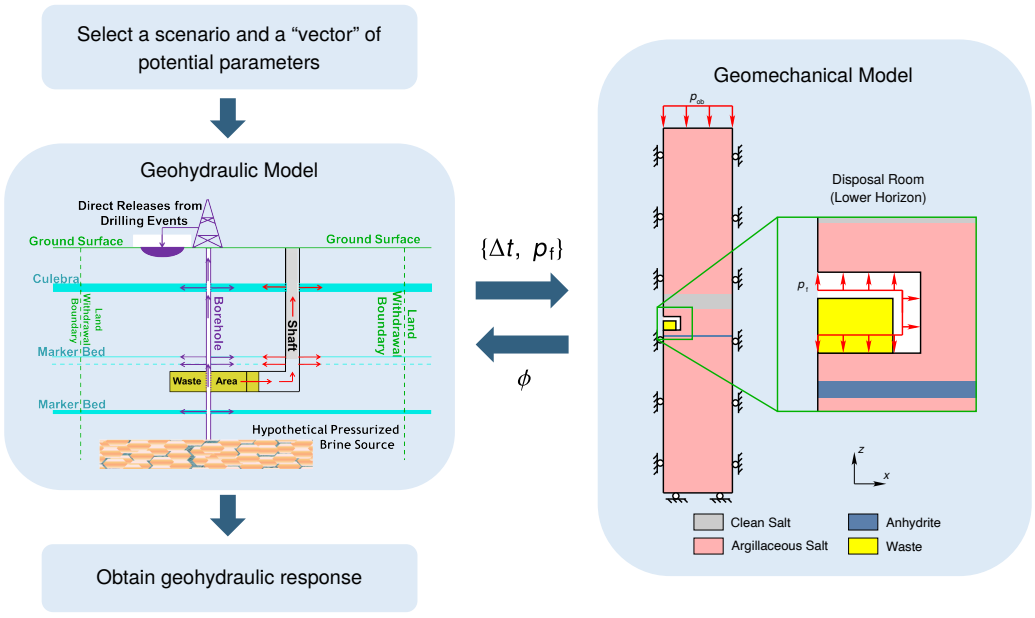


Figure 1-1. A schematic of fully coupled geohydraulic-geomechanical simulations.

The porosity of a disposal room at the Waste Isolation Pilot Plant (WIPP) is an input into the WIPP Performance Assessment (PA). Disposal room porosity principally affects the potential for radionuclide releases due to spallings and direct brine releases. "Spallings are solid materials blown into the borehole by gas flow through the waste at the time of a drilling intrusion." (Hansen et al. 2004) "Direct brine releases (DBR) are volumes of contaminated brine that may flow from the waste into a borehole during and immediately after a drilling intrusion." (Hansen et al. 2004) Greater disposal room porosity can decrease spalling releases if greater porosity is associated with

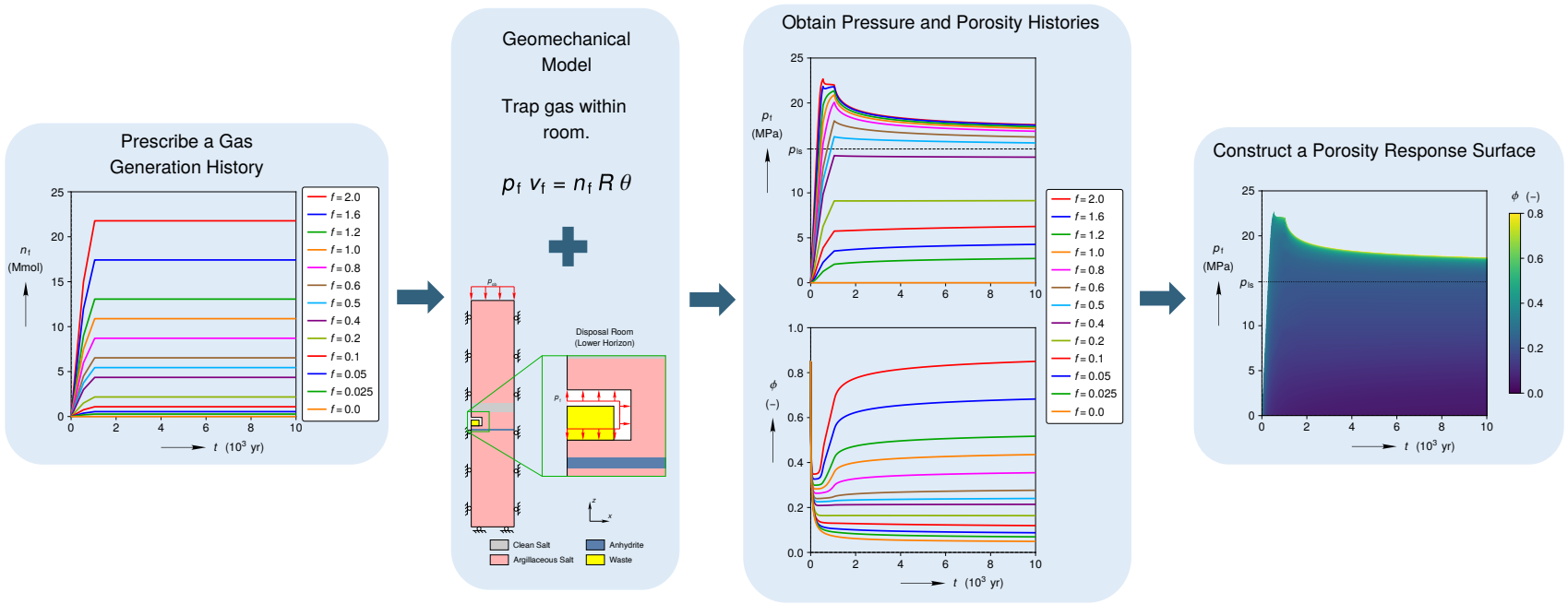


Figure 1-2. Schematic of the process used to recreate the legacy porosity response surface.

decreased fluid pressure within the repository. Direct brine releases have a more complex relationship with porosity: greater porosity increases the volume that can store contaminated fluids but also lowers the fluid pressure that drives releases.

Ideally, one would perform fully coupled geohydraulic–geomechanical simulations to predict the room porosity and flow of radionuclides through the repository. One would supply a scenario and a "vector" of potential parameters to the geohydraulic model. The geohydraulic model would pass a time increment Δt and a fluid pressure p_f to the geomechanical model at each time step, or at each iteration, and the geomechanical model would pass back a disposal room porosity ϕ (see Fig. 1-1). Throughout this report, the (Lagrangian) porosity is defined as $\phi = v_f/V_r$, where v_f is the volume available to store fluids and V_r is the original volume of the room. This coupling would allow the geohydraulic model to compute the geohydraulic response. The WIPP PA, however, must perform thousands of geohydraulic simulations to encompass the uncertainty in the large number of model parameters, and thousands of fully coupled simulations are simply not tractable with the current combination of software and hardware qualified for use by the WIPP PA. Instead, the PA currently uses an uncoupled approach to link the geomechanical simulations to the geohydraulic simulations.

The (uncoupled) geomechanical simulations include the mechanical behavior of the rock mass, waste, and gas within the room. (Gas is the only fluid within the room in the geomechanical model for simplicity.) The rock mass in these simulations consists of rock salt (halite), and may also include anhydrite, clay seams, and polyhalite. The rock mass and waste behavior are both path dependent, with the rock salt likely having the most path dependence. The response of a path dependent material depends on the current state of deformation *and* the past sequence of deformations applied to the material. Including path dependent materials in the geomechanical model causes the room porosity to depend on the current gas pressure and past sequence of gas pressures within the room. The gas is modeled using the ideal gas law: $p_f\, v_f = n_f\, R\, \theta$, where n_f is the number of moles of gas, R is the universal gas constant, and θ is the temperature. The geomechanical simulations, however, do not permit the gas to escape the room and the gas generation history is prescribed a priori as $n_f = f\, n_f^{pot}$, where f is a scale factor and $n_f^{pot} = n_f^{pot}(t)$ is the potential number of moles of gas generated by a given t. As shown in Fig. 1-2, each geomechanical simulation uses a different scale factor f to compute a corresponding pair of ϕ and p_f histories. Several simulations are run, each with a different f value, to create a set of ϕ and p_f histories. This set of histories are used to construct what is called a porosity response surface, in which ϕ is interpolated as a function of t and p_f. (See DOE (2014b) for the details of this interpolation process.)

The (uncoupled) geohydraulic simulations, on the other hand, include the two-phase (brine and gas) flow within the repository, as well as chemistry-based predictions of the gas generation rates for different scenarios. Accordingly, these simulations are well suited to compute p_f for a given ϕ. As shown in Fig. 1-3, each geohydraulic simulation passes t and p_f to the porosity response surface to obtain ϕ. Thus, the geohydraulic simulation uses the porosity response surface as a "look-up table" to determine the porosity at a given fluid pressure and time.

The accuracy of the uncoupled approach depends on the fluid pressurization history, the path dependence of the mechanical behavior, and the timescale of interest. Freeze et al. (1995) and Freeze (1996) found the uncoupled approach was accurate, provided the geohydraulic simulations

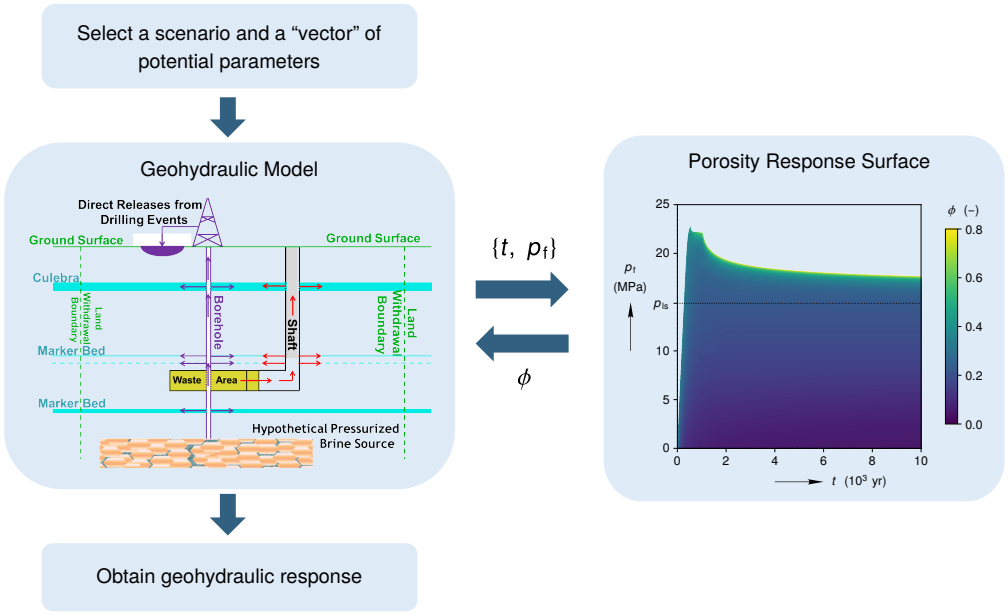


Figure 1-3. Schematic of the uncoupled porosity surface approach.

involved fluid pressure histories similar to those used in the geomechanical simulations or pressures remained low enough that room expansion did not occur. The geohydraulic simulations, however, are not constrained to follow the same pressure histories considered in the geomechanical simulations and room expansion is predicted to occur in some cases. Consider a case, for example, where the pressure is small and then suddenly increases to a much larger value at 1,000 yr. A fully coupled approach would predict a gradual increase in porosity after 1,000 yr because of the rock salt's viscoplastic path dependence. The uncoupled approach, on the other hand, would predict a nearly instantaneous porosity increase at 1,000 yr because it obtains the porosity from an interpolation based on qualitatively different pressure histories. The fully coupled and uncoupled approaches may eventually predict approximately the same porosity because salt's viscoplasticity has "fading memory", in which deformations from the distant past have less influence on the current response than recent deformations, but it may require many years for the two porosity predictions to agree.

The errors incurred by ignoring these transient path dependent effects have been historically regarded as acceptably small largely because Hansen et al. (2004) showed that disposal room porosity had a mild impact on repository performance. Hansen et al. (2004) compared radionuclide releases from the PA for the 2004 Compliance Recertification Application against releases from a PA that incorporated uncertainty in the porosity surface by statistically sampling from four porosity surfaces. The four porosity surfaces represented bounding cases for waste container strength and initial porosity. Hansen et al. (2004, Section 4.3) found that this porosity surface uncertainty caused a negligible increase in total radionuclide releases.

More recently, however, Vignes et al. (2023) constructed a new porosity response surface using a new geomechanical model to replace the legacy geomechanical model and legacy porosity surface

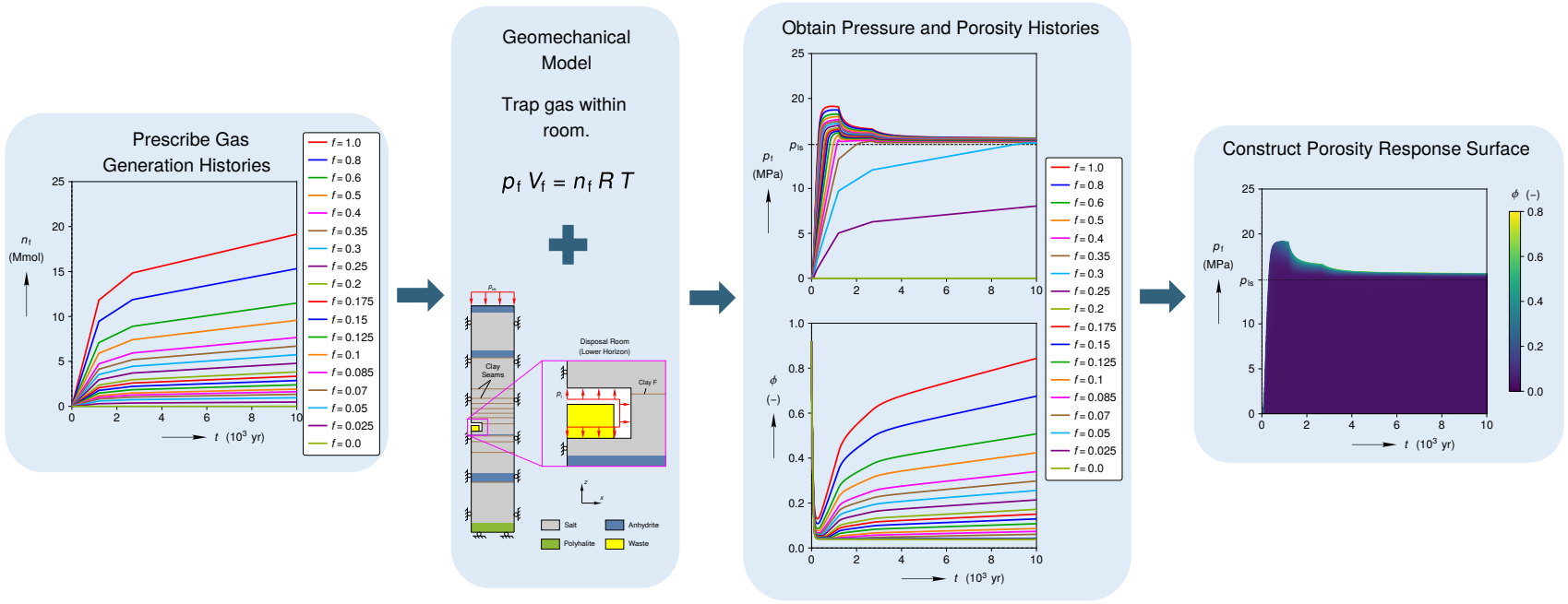


Figure 1-4. Schematic of the process used to generate the new porosity response surface.

created by Stone (1997a). Vignes et al. (2023) contains a full account of the differences between the legacy and new geomechanical models, but the most significant change was probably the addition of salt's creep behavior at low stresses (see Vignes et al. (2023, Section 2.1.5)). Comparing the left sides of Figs. 1-2 and 1-4, one can see that the gas generation curves and stratigraphies also differ. These model differences caused significant changes to the pressure and porosity histories that, in turn, led to changes to the porosity surface. Comparing the right sides of Figs. 1-2 and 1-4, the new surface reaches lower porosities at shorter times, and the porosity is insensitive to fluid pressure, provided $p_f < p_{ls}$ and $t \gtrsim 600$ yr. (Lithostatic pressure at the WIPP is taken as $p_{ls} = 14.88$ MPa herein.) For $p_f > p_{ls}$, the lithostatic pressure is supported entirely by the fluid pressure, the surrounding rock applies no compressive loads to the waste containers, and the room expands.

After Vignes et al. (2023) was published, Day (2025) compared releases associated with the legacy and new porosity surfaces using the PA model for the Replacement Panels Planned Change Request (RPPCR) (DOE 2024) as the baseline. The new porosity surface caused mean total releases to increase by 1.6× and 2.4× at the 0.1 and 0.001 compliance points, respectively. This greater than expected sensitivity to disposal room porosity could be because the new Vignes et al. (2023) porosity surface differs from the four surfaces considered by Hansen et al. (2004). For example, the Hansen et al. (2004) porosity surfaces increase the porosity compared to the legacy porosity surface, while the new porosity surface decreases porosity for $p_f < p_{ls}$ and $t > 72$ yr. The sensitivity could also be due to the interaction of the new porosity surface with numerous other changes to the PA model between 2004 and 2025. Regardless of the causes, the realization that room porosity can significantly affect releases has renewed interest in quantifying the errors associated with the uncoupled porosity surface approach, and in assessing potential methods to enhance the coupling between the geomechanical and geohydraulic models.

The remainder of this report is organized as follows. Chapter 2 presents an assessment of the errors associated with the uncoupled porosity surface approach. Chapter 3 reports on a preliminary study to quantify the impact of the porosity surface error on radionuclide releases. Chapter 4 describes a preliminary attempt to develop a machine-learned model for disposal room porosity. Chapter 5 summarizes the results.

The porosity surface error study and the development of a machine-learned model are in accordance with AP-209 (Reedlunn and King 2025), while the sensitivity study in Chapter 3 is a step beyond the plans in AP-209. Stepping beyond AP-209 is acceptable here because the sensitivity study is cursory in nature. The files associated with the results in Chapters 2 to 4 are archived at `/data/cvs/CVSLIB/WIPP_EXTERNAL/ap209_porositysurface` on Sandia Carlsbad's Linux server.

2. POROSITY ERROR ASSESSMENT

2.1. Procedure

This section describes the procedure used to characterize the porosity error associated with the legacy and new porosity response surfaces. The legacy surface was used for the original Compliance Certification Application (CCA), Compliance Recertification Application (CRA) 2004, CRA-2009, CRA-2014, and CRA-2019 (DOE 1996; DOE 2004; DOE 2009; DOE 2014a; DOE 2019), so it is good to be aware of the error associated with having used the legacy surface in those analyses, even though it will likely not be used in future CRAs. The need to characterize the error with the new porosity response surface is clear: it was used in the RPPCR (DOE 2024) and it will be used in CRA-2026. Computing the error for both surfaces also gives a sense of how much the error depends on the porosity surface being used. An understanding of this sensitivity will help future analysts decide if they should characterize the error associated with any future revisions to the porosity surface.

The most straightforward procedure to quantify the porosity error would be to simply compare the porosity ϕ^{ps} computed by a given porosity surface against the porosity ϕ^{fc} computed by the fully coupled approach for a series of representative scenarios and vectors. Developing the means to robustly couple the geohydraulic model to the geomechanical model, however, would have been a labor intensive process and might have been practically useful only for this particular error assessment.

Instead, a different procedure was used to quantify the errors associated with each porosity surface. As shown in Fig. 2-1, the steps in this procedure were:

1. A set of 600 representative, non-trivial, p_{f} histories and a corresponding set of 600 ϕ^{ps} histories was generated using the geohydraulic model from the RPPCR with a given porosity surface. This set of 600 histories came from considering Replicate 1, which involved 100 vectors for each of the six geohydrualic scenarios. These scenarios were named S1-BF, S2-BF, S3-BF, S4-BF, S5-BF, and S6-BF.

2. Each p_{f} history was applied as a time-varying pressure boundary condition on the surfaces of the room in the geomechanical model corresponding to the specified porosity surface. Each geomechanical simulation computed a porosity history ϕ^{gm}.

3. The porosity error history for each pressure history was computed as

$$e^{\mathrm{ps}} = \frac{|\phi^{\mathrm{ps}} - \phi^{\mathrm{gm}}|}{\phi^{\mathrm{gm}}}. \tag{2.1}$$

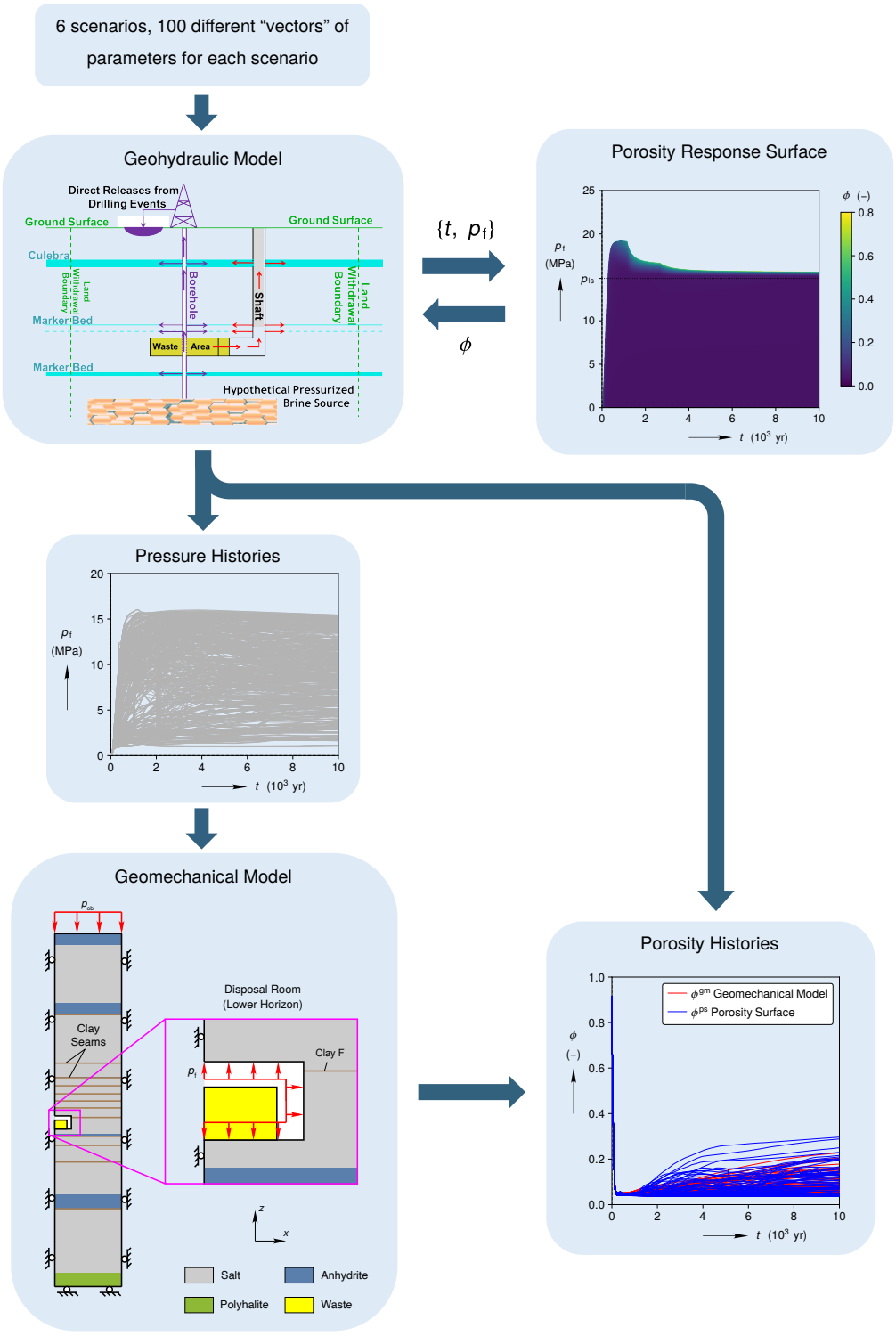


Figure 2-1. A schematic of the procedure used to assess the error with the uncoupled porosity surface approach. (Note that the porosity surface, pressure histories, geomechanical model schematic, and porosity histories correspond to the new porosity surface, but the same procedure was used to asses the error for the recreated legacy porosity surface.)

The numerator contains absolute value signs because this study usually was not concerned with the sign of the error, and because absolute value signs allow one to compute the average error magnitude by simply averaging a group of e^{ps} values.

The above procedure computed the porosity error corresponding to a set of representative p_{f} histories. Each p_{f} history was not strictly equivalent to the fluid pressure history p_{f}^{fc} that would have been computed by the fully coupled approach, but a strict equivalence was not necessary. By computing each p_{f} history using the geohydraulic model from the RPPCR, the set of p_{f} histories should be representative of the set of p_{f}^{fc} histories. The porosity errors computed by the above procedure should, therefore, be representative of the porosity errors one would compute by comparing ϕ^{ps} to ϕ^{fc}.

Note that the procedure above treated ϕ^{gm} as the "target" porosity that ϕ^{ps} aspired to be similar to. Any geomechanical model, of course, is not perfect and contains its own approximations (Vignes et al. 2023, Ch. 2), but those approximations were not the focus of this error study. This study was focused on the error due to using a porosity response surface instead of the fully coupled approach.

A few numerical details of geomechanical simulations used to compute ϕ^{gm} bear mentioning:

- All simulations were run using the finite element method capability in Sierra/SolidMechanics (2022) version 5.4.1, as qualified in Reedlunn (2023).
- Stone (1997a) created the legacy geomechanical model using SANTOS (Stone 1997b), which is no longer supported at Sandia, so the legacy geomechanical model was recreated herein as closely as possible using Sierra/SolidMechanics (2022). The recreated legacy geomechanical simulations presented herein used the same numerical setup as the recreated legacy simulations in Vignes et al. (2023, Sec. 2.4 and App. B), except for two deviations:
  1. The simulations herein used explicit time integration with viscoplastic scaling instead of implicit time integration. See Appendix A for details of the viscoplastic scaling used in the recreated legacy geomechanical model simulations.
  2. The waste block was discretized herein with selective deviatoric elements because uniform gradient elements would sometimes exhibit undesirable hourglass deformations.
- To be consistent, the legacy porosity response surface was recreated using the slightly different numerical setup mentioned above. See Appendix B for a comparison of the three legacy porosity surfaces: the original in Stone (1997a), the recreation in Vignes et al. (2023), and the recreation herein.
- The new geomechanical model herein used the same numerical setup as the new model detailed in Vignes et al. (2023, Sec. 2.4 and App. B). See Appendix A for details of the viscoplastic scaling used in the new geomechanical model simulations.
- See Vignes et al. (2023, Appendix C.1) for the mesh generation files, and see Appendix C for the Sierra/SM input files.

2.2. Results

This section contains two sub-sections. Section 2.2.1 shows the porosity error associated with the recreated legacy porosity surface. Section 2.2.2 depicts the porosity error associated with new porosity surface.

The plots in each sub-section either consider one case at a time or combine all the cases together. Presenting one case at a time is a straightforward way to examine how closely the porosity response surface approach captures, or fails to capture, the path dependence in the geomechanical model. Studying 600 individual cases, though, is an arduous task, so the following steps were taken to combine and analyze them:

1. Each p_f history and each e^ps history for a given case was linearly interpolated with respect to time and sampled at 1 yr intervals to eliminate any sampling bias.
2. The 1 yr sampled histories from all 600 cases were pooled together, leading to 6×10^6 p_f values and 6×10^6 e^ps values.
3. The error distribution was computed by sorting the 6×10^6 e^ps values into 1.29 width (logarithmic scale) error bins and calculating the relative frequency f^rel within each bin.
4. The pressure distribution was computed by sorting the 6×10^6 p_f values into 200 kPa width pressure bins and calculating the relative frequency f^rel within each bin.
5. The pressure dependence of the error was computed by sorting the 6×10^6 $\{p_\mathrm{f},\ e^\mathrm{ps}\}$ pairs into 200 kPa width pressure bins, and finding the mean(e^ps) and range(e^ps) within each bin.

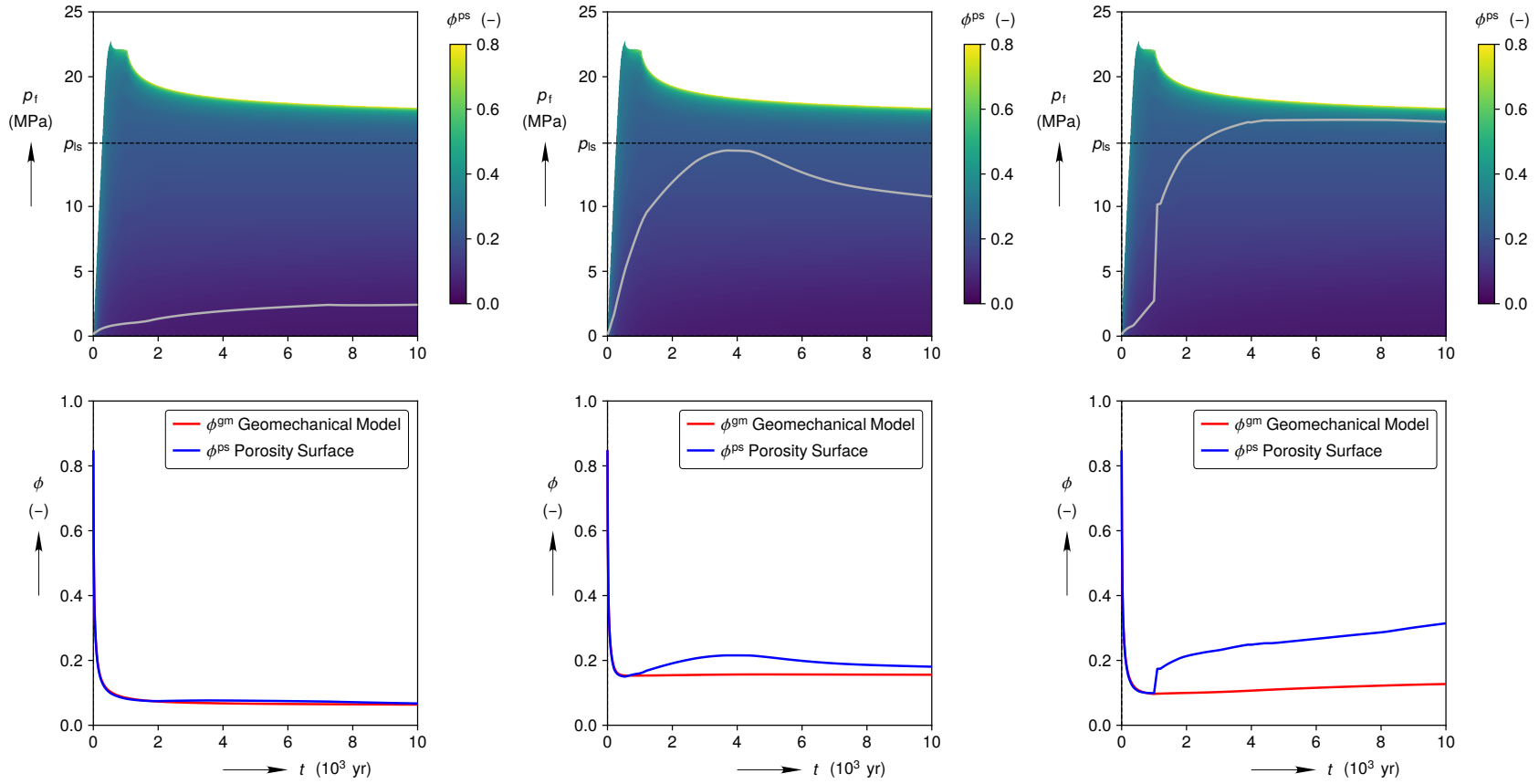
The relative frequency of quantity x within bin i was calculated as $f_i^\mathrm{rel} = n_i / \left(\sum_{j=1}^{J} n_j \right)$, where n_i is the number of x values within bin i and J is the total number of bins. The error distributions, pressure distributions, pressure dependence of e^ps are all plotted below.

To be clear, mean(e^ps) and range(e^ps) are plotted against p_f, rather than plotting e^ps against p_f, because $e^\mathrm{ps}(t)$ is not a simple mathematical function of $p_\mathrm{f}(t)$. One can only compute $\phi^\mathrm{gm}(t)$ and, hence, $e^\mathrm{ps}(t)$ by considering $p_\mathrm{f}(t)$, *plus* the full p_f history prior to t. One can, nevertheless, find useful correlations between $e^\mathrm{ps}(t)$ and $p_\mathrm{f}(t)$, as shown below.

2.2.1. Recreated Legacy Porosity Response Surface

Representative cases of low, medium, and high pressure are shown in Fig. 2-2. Each subfigure caption includes the scenario name and vector number for reference. The top plot in each subfigure shows the fluid pressure history in light gray, overlaid on top of the recreated legacy porosity response surface. The bottom plot in each subfigure shows the porosity predicted by the geomechanical model in red and the recreated legacy porosity response surface in blue. The difference between the red and blue is the raw (not relative) error.

As discussed in Chapter 1, the porosity error associated with using the recreated legacy porosity surface is significant when fluid pressure histories deviate from the fluid pressure histories used to construct the recreated legacy porosity response surface. In Fig. 2-2a, the fluid pressure history is



(a) Low pressure (Scenario S1-BF, Vector 26)

(b) Medium pressure (Scenario S5-BF, Vector 28)

(c) High pressure (Scenario S3-BF, Vector 65)

Figure 2-2. Fluid pressure histories overlaid on the recreated legacy porosity response surface (top row of plots), as well as porosities predicted by the recreated legacy geomechanical model and the recreated legacy porosity surface (bottom row of plots) for representative cases with low, medium, and high pressures.

similar to the $f = 0$ and 0.025 pressure histories in Fig. 1-2, except that it rises more gradually, which causes ϕ^{ps} to deviate from ϕ^{gm} by a small amount. The pressure history in Fig. 2-2b is reasonably similar to the $f = 0.1$, 0.2, and 0.4 pressure histories in Fig. 1-2 for $t \lesssim 10^3$ yr. The pressure, however, gradually grows to $p_f = 14.3$ MPa for $10^3 \lesssim t \lesssim 4 \times 10^3$ yr and gradually decreases to $p_f = 10.8$ MPa by $t = 10^4$ yr, while the $f = 0.1$, 0.2, and 0.4 pressure histories remain flat. This pressure deviation causes ϕ^{ps} to substantially deviate from ϕ^{gm} by $t = 4 \times 10^3$ yr and then deviate to a lesser degree by $t = 10^4$ yr. This pressure history in Fig. 2-2c is close to the $f = 0.025$ and 0.05 pressure histories in Fig. 1-2 for $t < 10^3$ yr. At $t = 10^3$ yr, however, a drilling intrusion causes an abrupt pressure increase to $p_f = 10.2$ MPa, which dramatically differs from the nearly flat $f = 0.025$ and 0.05 pressure histories. The subsequent gradual increase to fluid pressures above p_{ls} only increases the discrepancy. These deviations cause ϕ^{ps} to substantially differ from ϕ^{gm} at $t = 10^3$ yr and gradually differ by a greater degree as time goes on. In each of these three cases, as well as the 597 cases not shown, ϕ^{gm} responds to any pressure deviation more slowly than ϕ^{ps} does, as expected from the discussion in Chapter 1.

Fig. 2-3a depicts the distribution of error associated with the recreated legacy porosity response surface for all 600 cases combined. The error distribution is skewed towards higher errors and the most common error is $e^{ps} \approx 60$ %. The mean error over all data points is 38.6 %.

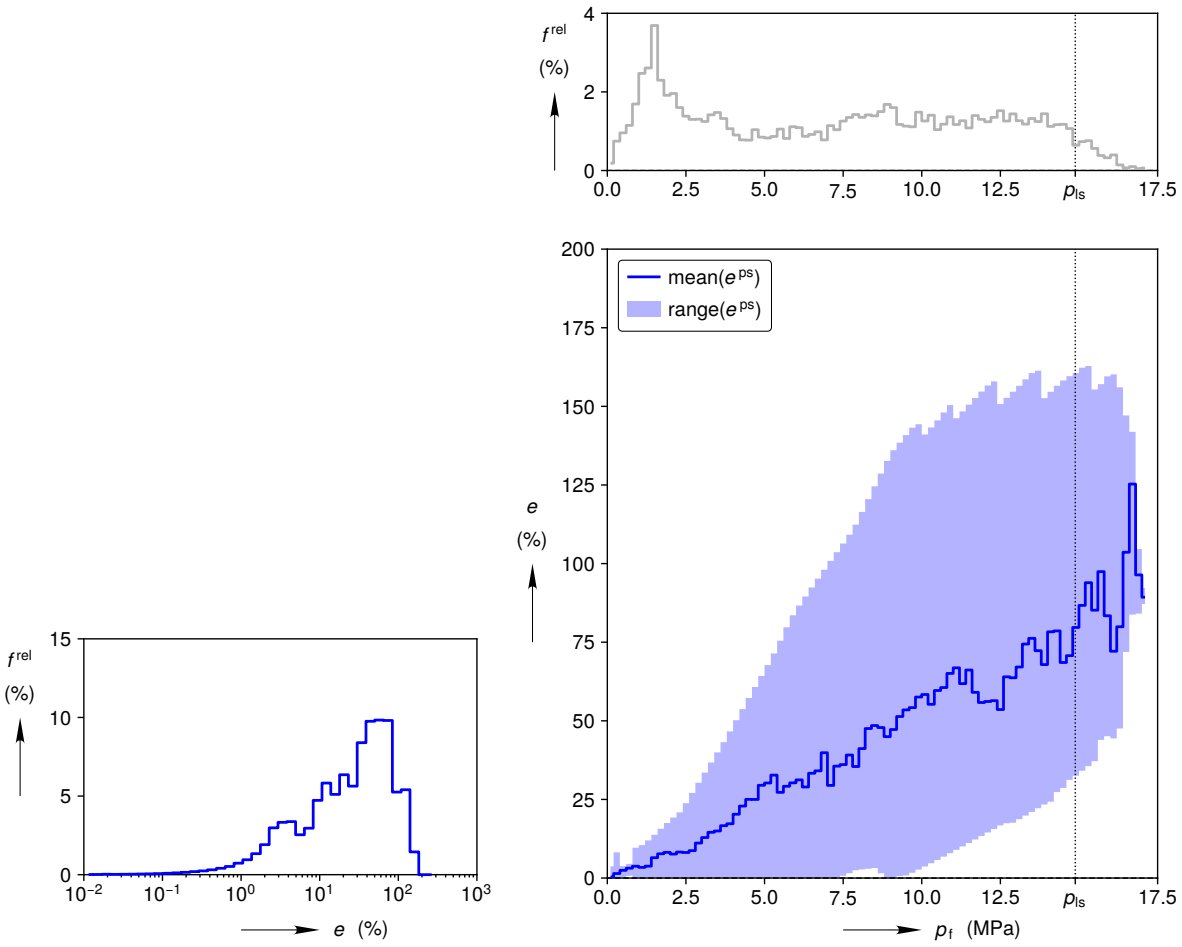
The top plot in Fig. 2-3b shows the fluid pressure distribution for all 600 cases combined. Fluid pressures are fairly uniformly distributed, except 1 MPa $\leq p_f \leq$ 2.2 MPa is more common and $p_f \geq p_{ls}$ is less common.

The bottom plot in Fig. 2-3b, more importantly, shows the pressure dependence of the legacy porosity surface error for all 600 cases combined. Clearly, mean(e^{ps}) and range(e^{ps}) increase with increasing p_f, which confirms the anecdotal evidence seen in the low, medium, and high pressure cases in Fig. 2-2. Although the wide range(e^{ps}) means that several factors influence e^{ps} besides p_f, three causes for the mean(e^{ps}) vs. p_f trend seem plausible:

1. If a p_f history deviates from the interpolated $f = x$ curve it had been roughly following, the pressure usually grows, rather than shrinks, such that subsequent errors correspond to higher pressures.
2. The difference $p_{ls} - p_{rc}$, where p_{rc} is the back pressure supplied by the room contents, is the driving force for room closure/expansion. If p_f starts at 1 MPa and increases by 10 %, $p_{ls} - p_{rc}$ only changes by 0.1 MPa. If p_f starts at 10 MPa and increases by 10 %, $p_{ls} - p_{rc}$ changes by 1 MPa.
3. The slope $\mathrm{d}\,\phi^{ps} / \mathrm{d}\,p_f$ is greater for $p_f > p_{ls}$ than for $p_f < p_{ls}$.

2.2.2. New Porosity Response Surface

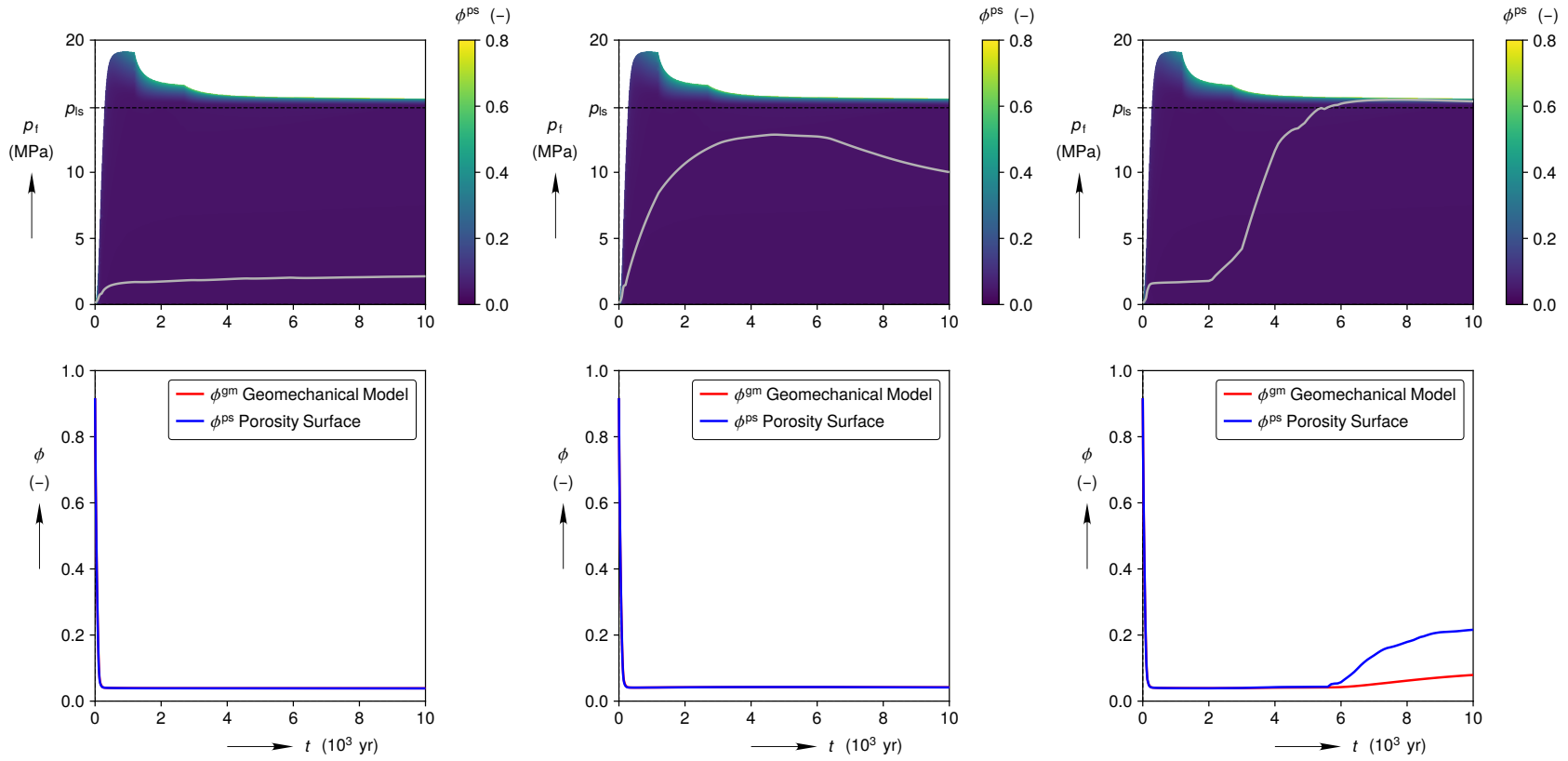
This section also begins with three representative cases in Fig. 2-4 to show how the ϕ^{ps} and ϕ^{gm} histories agree or differ, depending on the fluid pressure. The fluid pressure history in the top plot of Fig. 2-4a remains low, increases monotonically, and can be viewed as a scaled version of the pressure histories in Fig. 1-4. It is, therefore, not surprising that the ϕ^{ps} history overlaps very well with the ϕ^{gm} history in the bottom plot of Fig. 2-4a. The fluid pressure in the top plot of Fig. 2-4b,



(a) Error Distribution

(b) Pressure Distribution and Pressure Dependence of Error

Figure 2-3. The porosity error associated with the recreated legacy porosity surface for all 600 cases combined.



(a) Low pressure (Scenario S1-BF, Vector 49)

(b) Medium pressure (Scenario S5-BF, Vector 46)

(c) High pressure (Scenario S6-BF, Vector 5)

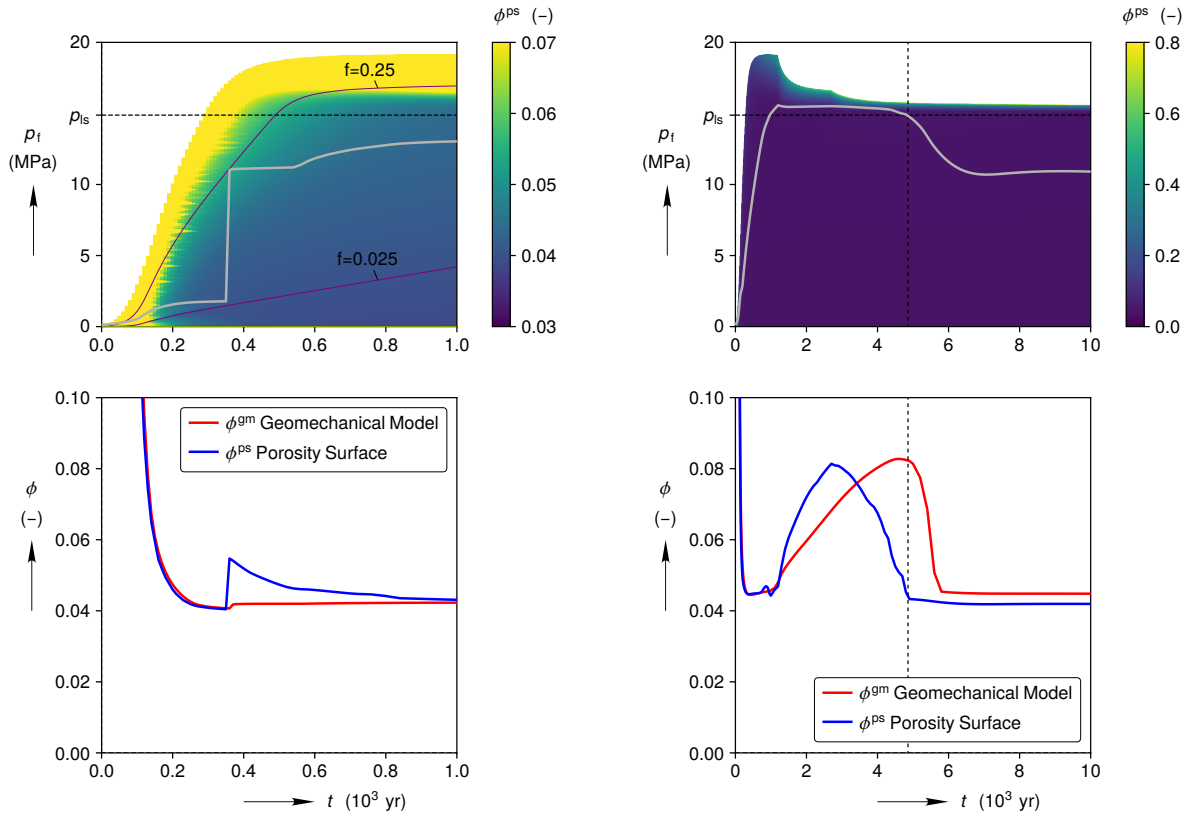
Figure 2-4. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the new geomechanical model and the new legacy porosity surface (bottom row of plots) for representative cases with low, medium, and high pressures.

by contrast, grows to medium values, evolves non-monotonically, and cannot be qualitatively captured using a simple scaling of the pressure histories in Fig. 1-4. The ϕ^{ps} history, nevertheless, overlaps very well with ϕ^{gm} history in the bottom plot of Fig. 2-4b. In the top plot of Fig. 2-4c, the fluid pressure increases, plateaus until $t = 2 \times 10^3$ yr, gradually increases, exceeds p_{ls} at $t = 5.6 \times 10^3$ yr, and remains slightly above p_{ls} for the rest of the simulation. Again, a simple scaling of the various pressure histories in Fig. 1-4 would not qualitatively capture the p_f history in Fig. 2-4c, yet the ϕ^{ps} and ϕ^{gm} histories agree quite well for $t < 5.6 \times 10^3$ yr. Once $p_f > p_{ls}$, though, the porosity surface approach predicts ϕ^{ps} grows far faster than ϕ^{gm}.

Although large differences between ϕ^{ps} and ϕ^{gm} most often occur when $p_f > p_{ls}$ with the new porosity surface, significant differences can still occur when $p_f < p_{ls}$. Porosity differences can be caused by an abrupt pressure increase, early in the 10×10^3 yr regulatory period, as shown in Fig. 2-5a. At $t = 0.35 \times 10^3$ yr, a drilling intrusion causes the pressure to jump from $p_f = 1.78$ MPa, near the $f = 0.025$ pressure history, to $p_f = 11.0$ MPa, near the $f = 0.25$ pressure history. This pressure jump causes the porosity predicted by the porosity surface to jump from $\phi^{ps} = 0.040$ to 0.055, while the porosity predicted by the geomechanical model only jumps from $\phi^{gm} = 0.040$ to 0.042. The $f = 0.25$ simulation reached $p_f = 11.0$ MPa along a significantly different path than the pressure history with the jump at $t = 0.35 \times 10^3$ yr, so it is expected that ϕ^{ps} and ϕ^{gm} would differ. The difference, however, is transient: the two porosity predictions are nearly identical by $t = 10^3$ yr. Every vector in Scenario S2-BF includes a drilling intrusion with a similar early, abrupt, pressure increase of varying magnitude, which creates many other pressures beneath p_{ls} with $e^{ps} > 5$ %. (Scenario S6-BF also includes abrupt pressure increases, but they occur at $t = 2 \times 10^3$ yr, which is in the region where the porosity surface is insensitive for $p_f < p_{ls}$.) Differences between ϕ^{ps} and ϕ^{gm} can also occur when $p_f > p_{ls}$ for a period of time and then dips back below p_{ls}, similar to the top plot in Fig. 2-5b. At $t = 4.86 \times 10^3$ yr, p_f drops below p_{ls}, causing the porosity predicted by the porosity surface to drop to $\phi^{ps} = 0.043$, but the geomechanical model requires an additional 0.94×10^3 yr to drop to $\phi^{gm} = 0.045$. A small porosity difference persists even at $t = 10 \times 10^3$ yr, possibly due to small pockets of high porosity related to friction between the waste and the room surfaces. Analogous sequences of events cause 48 cases to have $e^{ps} > 5$ % while $p_f < p_{ls}$ at some point during the case's history.

The plots in Fig. 2-6 compare the error and pressure for all 600 cases combined, under two conditions. The data labeled as e^{ps} and f^{rel} corresponds to the error and pressure under nominal conditions. The data labeled as $e^{ps}_{p_f<p_{ls}}$ and $f^{rel}_{p_f<p_{ls}}$ corresponds to the error and pressure for cases where the fluid pressure never exceeds lithostatic pressure. Fluid pressures can only exceed lithostatic pressure if fluid cannot quickly escape the rooms and drifts. The ability of fluid to escape depends on the permeability of the excavation damage zone (a.k.a. disturbed rock zone) around the room, the permeability of anhydrite strata above and below the room, and the permeability due to fluid pressure driven percolation through the relatively undamaged salt around the room (Popp et al. 2014). The model for the permeability of the excavation damage zone and the anhydrite has uncertainties, as all models do, and the geohydraulic model currently does not include fluid pressure driven percolation, so it can be helpful to consider what might occur if p_f never exceeds p_{ls}.

Fig. 2-6a shows the distributions of error associated with the new porosity response surface are either bimodal or unimodal. The e^{ps} distribution has one peak at $e^{ps} = 1.2$ % and another smaller



(a) Early abrupt pressure increase (Scenario S2-BF, Vector 7)

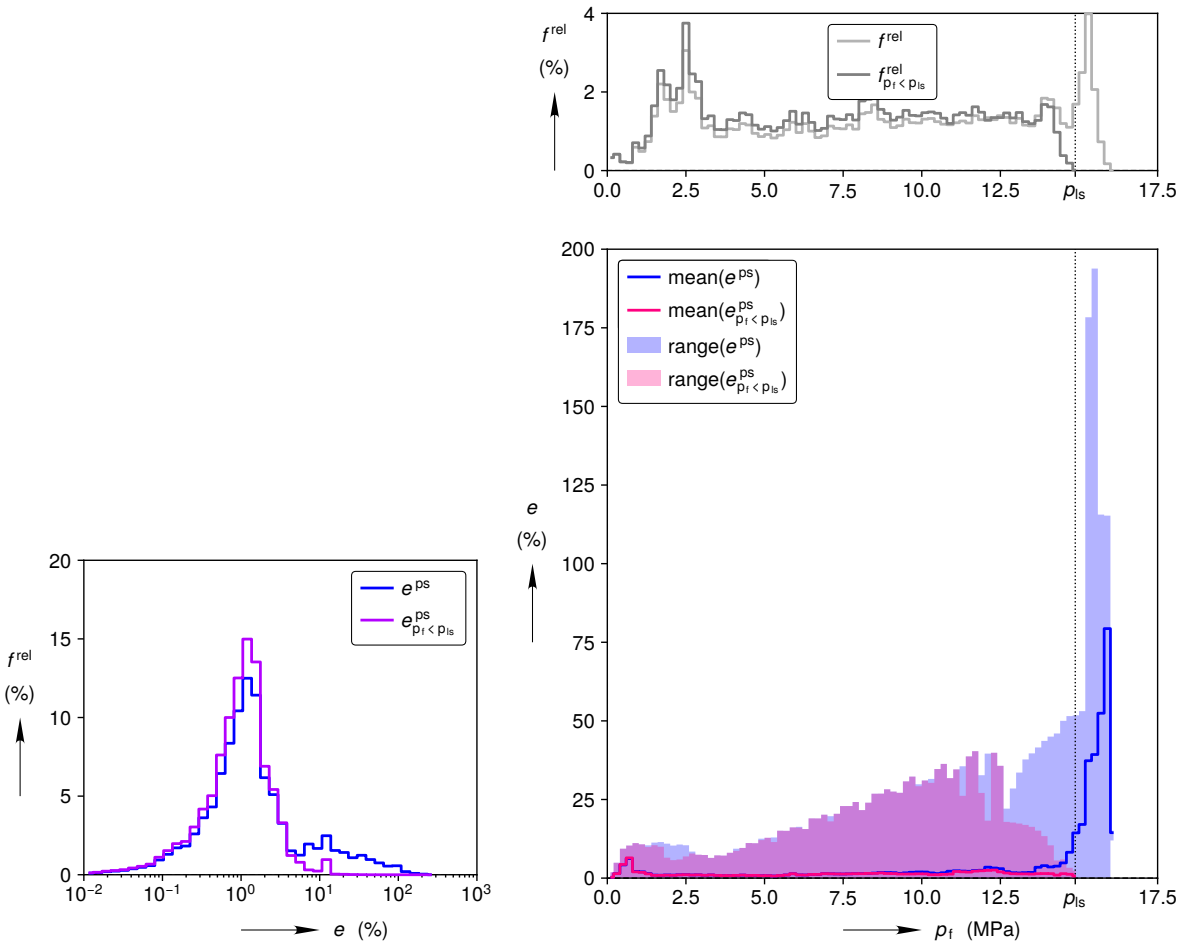
(b) Above, then below, p_{ls} (Scenario S5-BF, Vector 28)

Figure 2-5. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the new geomechanical model and the new porosity surface (bottom row of plots) for two representative, yet less common, cases.

peak at $e^{ps} = 12.1$ %. Averaging e^{ps} over all 600 cases gives 5.1 %, which is far less than the 38.6 % overall average error associated with the recreated legacy porosity response surface. The $e^{ps}_{p_f<p_{ls}}$ distribution has only one significant peak at $e^{ps} = 1.2$ %, showing that the second smaller peak in e^{ps} occurs for $p_f > p_{ls}$. The overall average $e^{ps}_{p_f<p_{ls}}$ is 1.3 %.

The top plot in Fig. 2-6b shows that the fluid pressure distributions are fairly uniform. The f^{rel} distribution, nevertheless, has small peaks at 1.4 MPa $< p_f \leq 3$ MPa and $p_{ls} < p_f \leq 15.6$ MPa. Note that the small peak at $p_{ls} < p_f \leq 15.6$ MPa did not exist with the recreated legacy porosity surface (see the top plot in Fig. 2-3b). As one might expect, the lower porosities predicted by the new porosity surface cause higher fluid pressures. The $f^{rel}_{p_f<p_{ls}}$ distribution is as expected: $f^{rel}_{p_f<p_{ls}} = 0$ for $p_f > p_{ls}$ and slightly larger $f^{rel}_{p_f<p_{ls}}$ values than f^{rel} values in most of the other pressure bins.

The bottom plot in Fig. 2-6b shows that e^{ps} depends non-linearly on pressure. The mean error within a fluid pressure bin, mean(e^{ps}), is less than 8.2 % for $p_f < p_{ls}$, but mean(e^{ps}) reaches as high as 79 % for $p_f > p_{ls}$. This further confirms the anecdotal observations made for the low,



(a) Error Distribution

(b) Pressure Distribution and Pressure Dependence of Error

Figure 2-6. The porosity error associated with the new porosity surface, across all 600 cases, is compared to the porosity error when excluding cases where fluid pressure exceeds lithostatic pressure. Note that the error ranges in the lower plot of (b) are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}), pink for range($e^{ps}_{p_f<p_{ls}}$), and lavender where range(e^{ps}) and range($e^{ps}_{p_f<p_{ls}}$) overlap.

medium, and high pressure cases in Fig. 2-4: the porosity surface approach usually predicts room porosity accurately for $p_f < p_{ls}$, but its predictions involve substantial errors when $p_f > p_{ls}$. Although mean(e^{ps}) is fairly low for $p_f < p_{ls}$, range(e^{ps}) is well above 5 % for nearly all pressures beneath p_{ls}, reaching as high as 51.5 % for p_f = 14.7 MPa. The upper bound of range(e^{ps}) for $p_f < p_{ls}$ is due to cases similar to those shown in Fig. 2-5.

The bottom plot in Fig. 2-6b also compares the pressure dependence of e^{ps} and $e^{ps}_{p_f<p_{ls}}$. This comparison shows that most of the larger e^{ps} values corresponding to 12.5 MPa < p_f < p_{ls} (and a few of the larger e^{ps} values corresponding to lower p_f) are due to fluid pressures exceeding p_{ls} and then dropping back beneath p_{ls}, similar to Fig. 2-5b.

It was not obvious at the outset of this study that mean(e^{ps}) would be quite so low for $p_f < p_{ls}$.

One might point to the new porosity surface's lack of sensitivity to fluid pressure for $p_f < p_{ls}$ and $t \gtrsim 600$ yr to argue mean(e^{ps}) will be small, but that only means that the ϕ^{ps} histories will all be similar for these conditions. It does not indicate whether ϕ^{ps} and ϕ^{gm} will agree or not.

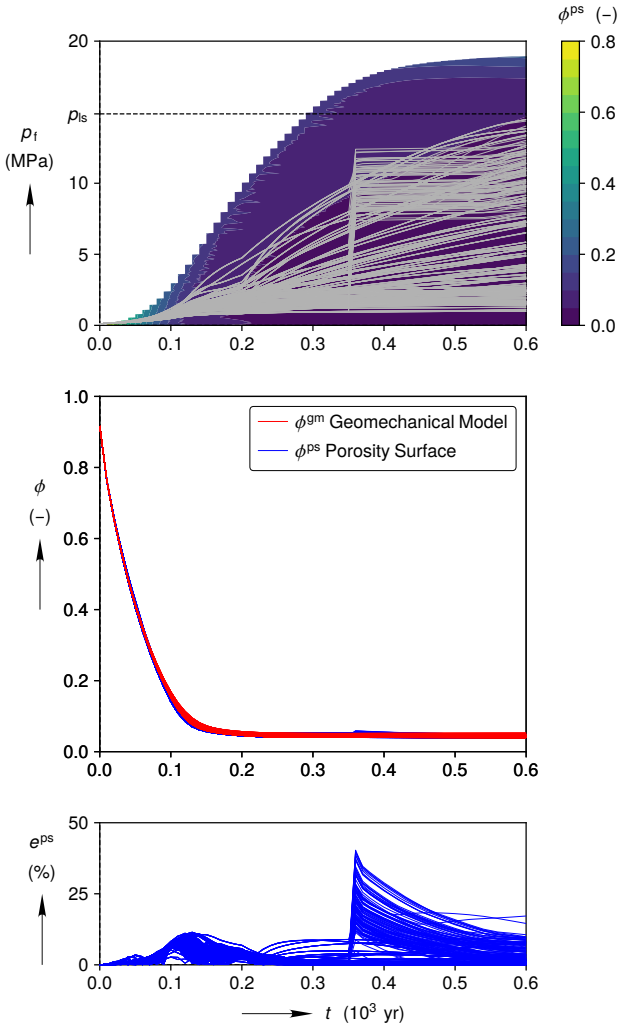


Figure 2-7. Fluid pressure histories overlaid on the new porosity response surface (top plot), porosities predicted by the new geomechanical model and the new porosity surface (middle plot), and the porosity error associated with the new porosity surface (bottom plot) of all 600 cases during the first 0.6×10^3 yr.

The reason why mean(e^{ps}) is low for $p_f < p_{ls}$ is *both* ϕ^{ps} and ϕ^{gm} are only mildly sensitive to the p_f histories considered, provided $p_f < p_{ls}$. Fig. 2-7 depicts the pressure histories, the porosity histories, and the error histories for all 600 cases during the first 0.6×10^3 yr. The top plot includes the porosity surface, plotted as a contour map instead of an intensity map, in the background. The contours for $0 < t \leq 0.12 \times 10^3$ yr largely run south to north, meaning ϕ^{ps} depends more on t than on p_f. By $t = 0.12 \times 10^3$ yr, the room is mostly finished closing and the variations in ϕ^{gm} and ϕ^{ps} are not large. The p_f histories do cross several of the pressure histories used to generate the porosity surface during $0 < t \leq 0.12 \times 10^3$ yr, which is why $e^{ps} \approx 7$ % at

$t = 0.12 \times 10^3$ yr (and why mean(e^{ps}) = 6.2 % for the bin centered at $p_f = 0.7$ MPa in Fig. 2-6b), but such errors may be small enough to ignore. Larger errors only develop when abrupt large pressure increases, similar to Fig. 2-5a, occur at $t = 0.35 \times 10^3$ yr. These larger errors, though, still are not big enough or frequent enough to substantially affect mean(e^{ps}) for $p_f < p_{ls}$.

The preceding paragraph begs the question, "Why is the new geomechanical model (and the resulting new porosity surface) only mildly sensitive to the $p_f < p_{ls}$ histories considered?" The reason is the initial room closure and waste compaction largely outrun the fluid pressurization, so that p_f is rarely large enough to unload the waste and affect ϕ. To explain in greater detail, during $0 < t \lesssim 0.09 \times 10^3$ yr, the fluid pressure has the potential to significantly slow down room closure because the waste does not supply much back pressure (Jensen et al. 2023, Section 3.4), but the fluid pressures in Fig. 2-7 are too small (0.1 MPa $< p_f <$ 1 MPa) to have a strong effect. During 0.09×10^3 yr $\lesssim t \lesssim 0.12 \times 10^3$ yr, the waste stiffens and begins to supply significant back pressure to resist the room closure (Jensen et al. 2023, Section 3.4), while the fluid pressures grow a small amount more. During 0.12×10^3 yr $< t < 0.6 \times 10^3$ yr, fluid pressures continue to grow and some nearly reach p_{ls} by $t = 0.6 \times 10^3$ yr, but most do not exceed the waste back pressure, so they have no effect on ϕ. Those fluid pressures that do exceed the waste back pressure are still less than p_{ls}, so they only serve to slow the closure of a nearly completely closed room.

The preceding analysis suggests two conditions that might encourage future researchers to repeat this error study. If changes to the geomechanical model cause room closure to progress significantly more slowly and/or if changes to the geohydraulic model cause high fluid pressures to occur significantly earlier, then porosity errors associated with the porosity surface approach could increase.

This page intentionally left blank.

3. PRELIMINARY PERFORMANCE ASSESSMENT SENSITIVITY STUDY

This chapter presents a preliminary study of the sensitivity of radionuclide releases to the porosity error associated with the new porosity surface. Ideally, this study would compare radionuclide releases computed using the new porosity surface against those computed using the fully coupled approach. As previously mentioned, however, the fully coupled approach is not computationally tractable. Consequently, this preliminary study compares releases computed using the new porosity surface against those computed using a conditionally limited porosity surface.

3.1. Conditionally Limited Porosity Surface

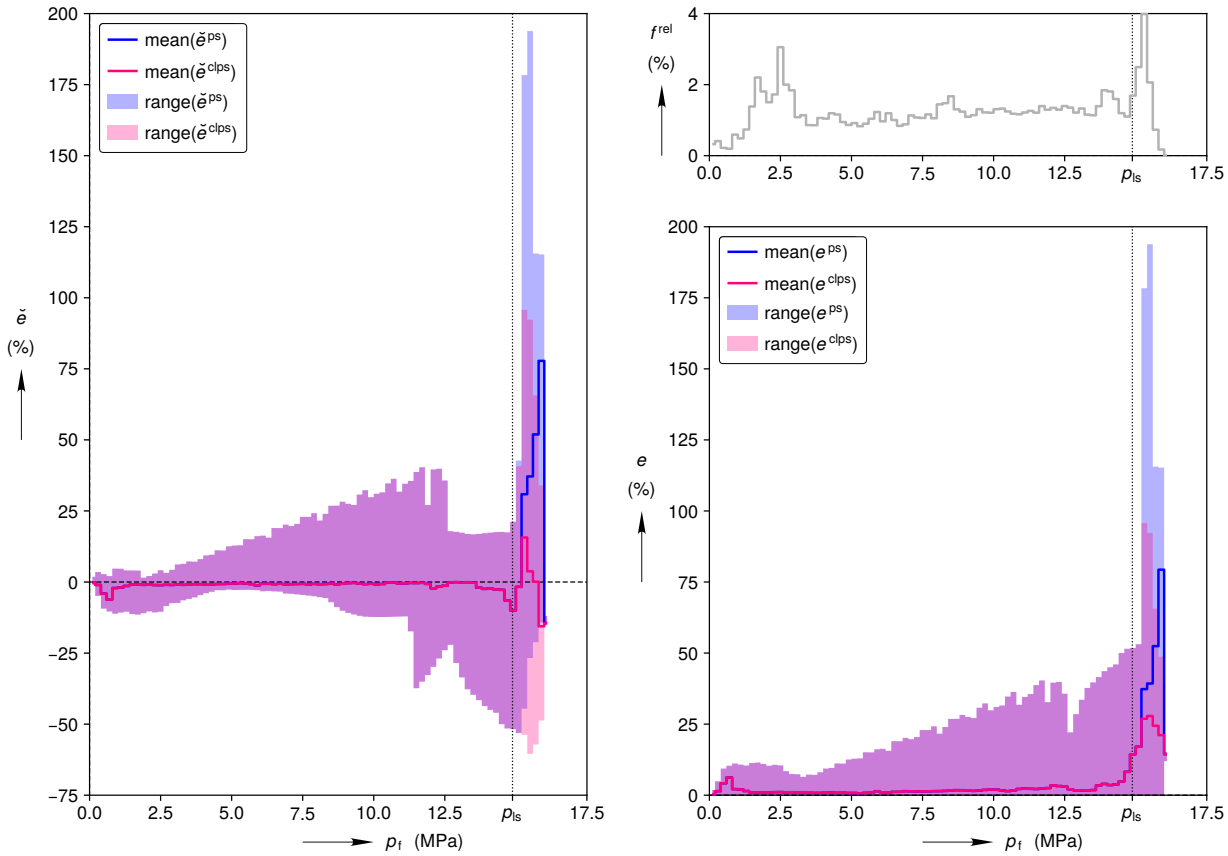
The conditionally limited porosity surface approach is a crude, yet simple, way to modify the new porosity surface so that it predicts porosities that are closer to those expected from the fully coupled approach. It was implemented as a modification to the geohydraulic model's porosity surface lookup routine. It computes ϕ^{ps} using the new porosity response surface and then enforces $\phi^{\mathrm{ps}} \leq \phi_{\max}$ if $p_{\mathrm{f}} > p_{\mathrm{ls}}$, where $\phi_{\max} = 9$ %. After the conditional limit is enforced, ϕ^{ps} is renamed to ϕ^{clps}. Although this conditional treatment does not properly consider the path dependence of the porosity response, the following paragraphs show that it reduces the porosity error associated with the new porosity surface.

The key observation motivating the conditionally limited porosity surface is ϕ^{ps} is generally larger than ϕ^{gm} when $p_{\mathrm{f}} > p_{\mathrm{ls}}$ for the new porosity surface. Consider the two specific cases shown in Figs. 2-4c and 2-5b. The uncoupled porosity surface approach gives $\phi^{\mathrm{ps}} > \phi^{\mathrm{gm}}$ when p_{f} first exceeds p_{ls}, but it can also give $\phi^{\mathrm{ps}} < \phi^{\mathrm{gm}}$ when the pressure drops back down again. One must, therefore, consider all 600 cases to determine the most common behavior. To do so, the signed relative error is computed as

$$\breve{e}^{\mathrm{ps}} = \frac{\phi^{\mathrm{ps}} - \phi^{\mathrm{gm}}}{\phi^{\mathrm{gm}}}, \tag{3.1}$$

which is identical to Eq. (2.1), except it does not include the absolute value signs. The quantities mean($\breve{e}^{\mathrm{ps}}$) and range($\breve{e}^{\mathrm{ps}}$) are computed using the procedure described at the start of Section 2.2. Figure 3-1a shows that $\breve{e}^{\mathrm{ps}}$ can be positive or negative for a given p_{f}, but mean($\breve{e}^{\mathrm{ps}}$) > 0 for $p_{\mathrm{f}} > p_{\mathrm{ls}}$, which confirms that ϕ^{ps} is generally larger than ϕ^{gm} when $p_{\mathrm{f}} > p_{\mathrm{ls}}$.

The maximum porosity $\phi_{\max} = 9$ % was selected to roughly minimize the porosity error associated with the conditionally limited porosity surface approach. One could rigorously compute this error using a procedure analogous to that discussed in Section 2.1: one could use the geohydraulic simulations with the conditionally limited porosity surface to compute the conditionally limited porosity ϕ^{clps} and fluid pressure $p_{\mathrm{f}}^{\mathrm{cl}}$ histories; feed the $p_{\mathrm{f}}^{\mathrm{cl}}$ histories into the



(a) Signed Error **(b) Error**

Figure 3-1. The porosity error associated with the new porosity surface and the approximate porosity error associated with the conditionally limited new porosity surface for all 600 cases combined. Note that the error ranges are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}) (or range($\breve{e}^{ps}$)), pink for range(e^{clps}) (or range($\breve{e}^{clps}$)), and lavender where range(e^{ps}) and range(e^{clps}) (or range($\breve{e}^{ps}$) and range($\breve{e}^{clps}$)) overlap.

geomechanical model to compute conditionally limited geomechanical model porosities ϕ^{clgm}; and then compute the error as $|\phi^{clps} - \phi^{clgm}|/\phi^{clgm}$. Those geomechanical simulations, however, would take significant computational resources and this is a preliminary sensitivity study, so a more expeditious method was used. The porosity ϕ^{clps} was computed by enforcing the conditional limit on the existing ϕ^{ps} data, and the porosity error associated with the conditionally limited porosity surface e^{clps} was computed as

$$e^{clps} = \frac{\left|\phi^{clps} - \phi^{gm}\right|}{\phi^{gm}}. \tag{3.2}$$

Similarly, the signed error $\breve{e}^{clps}$ was computed as

$$\breve{e}^{clps} = \frac{\phi^{clps} - \phi^{gm}}{\phi^{gm}}. \tag{3.3}$$

To be clear, one can only guarantee $\phi^{gm} = \phi^{clgm}$ if $p_f < p_{ls}$ or if $p_f \geq p_{ls}$ and $\phi^{gm} \leq \phi_{max}$. On the other hand, only 2.1 % of all data points correspond to $p_f \geq p_{ls}$ and $\phi^{gm} > \phi_{max}$. In addition, the discrepancy between ϕ^{gm} and ϕ^{clgm} in this 2.1 % of all data points should be moderate since both ϕ^{gm} and ϕ^{clgm} respond relatively slowly to high pressures. Eqs. (3.2) and (3.3) are, therefore, considered approximate error metrics. Being able to quickly compute e^{clps} and $\breve{e}^{clps}$ permitted a rapid selection of ϕ_{max}. Values between ϕ_{max} = 5 % and 12 %, in 1 % increments, were tried until ϕ_{max} = 9 % was found to minimize e^{clps} and $\breve{e}^{clps}$. Fig. 3-1a shows that mean($\breve{e}^{clps}$) has both negative and positive values for $p_f > p_{ls}$. The lower plot in Fig. 3-1b shows that mean(e^{clps}) and range(e^{clps}) are significantly smaller than mean(e^{ps}) and range(e^{ps}), respectively. Thus, the conditionally limited porosity surface approach behaves as intended: it predicts porosities that are closer to those expected from the fully coupled approach.

3.2. Comparison of Radionuclide Releases

This section compares the radionuclide releases associated with the new porosity surface and the conditionally limited (new) porosity surface. Geohydraulic behaviors are also shown to better understand the impact of these two porosity surfaces. The RPPCR PA, which used the new porosity surface, serves as the baseline for the comparison. This baseline aligns with the RPPCR geohydraulic model used in Section 2.1 to evaluate the porosity error and Day (2025)'s analysis that quantified the effects of transitioning from the legacy to the new porosity surface. The baseline is compared to a new analysis, referred to as RPPCR_MS. RPPCR_MS used the conditionally limited porosity surface in a Replicate 1 (100-realization) analysis of all six geohydraulic scenarios, while remaining otherwise identical to the RPPCR PA.

Figs. 3-2 to 3-4 depict the Replicate 1 geohydraulic flow results corresponding to the new porosity surface and the conditionally limited porosity surface. This paragraph introduces the content in these figures, while the next paragraph interprets the content. Figs. 3-2 and 3-3 show detailed results for the representative waste panel in the undisturbed (S1-BF) and Castile-intruded (S2-BF) scenarios, respectively, because these combinations typically have higher brine pressures and brine saturations than other locations and scenarios. The plots include brine pressure (WAS_PRES), brine saturation (WAS_SATB), and porosity (WAS_POR) results for individual vectors, along with the associated means. The variable WAS_PRES is equivalent to p_f because capillary pressure is not modeled in the waste areas (gas pressure is equal to brine pressure). Fig. 3-4 gives a more comprehensive view of the flow results. It shows the mean responses for the same metrics within the representative waste panel, the north rest-of-repository, and the west rest-of-repository for S1-BF and S2-BF, as well as for the intrusion scenario without Castile intersection (S4-BF) and the dual intrusion scenario with and without Castile intersection (S6-BF). Scenarios S3-BF and S5-BF were computed as well, but are not shown in Fig. 3-4 because the results for those scenarios are similar to Scenarios S2-BF and S4-BF, respectively.

The Replicate 1 geohydraulic flow results in Figs. 3-2 to 3-4 exhibit low sensitivity to switching to the conditionally limited porosity surface. The rightmost columns of Figs. 3-2 and 3-3 show that the conditionally limited porosity surface successfully limited ϕ^{clps} to ϕ_{max} when $p_f > p_{ls}$ for some realizations. The leftmost and middle columns of Figs. 3-2 and 3-3 also show that

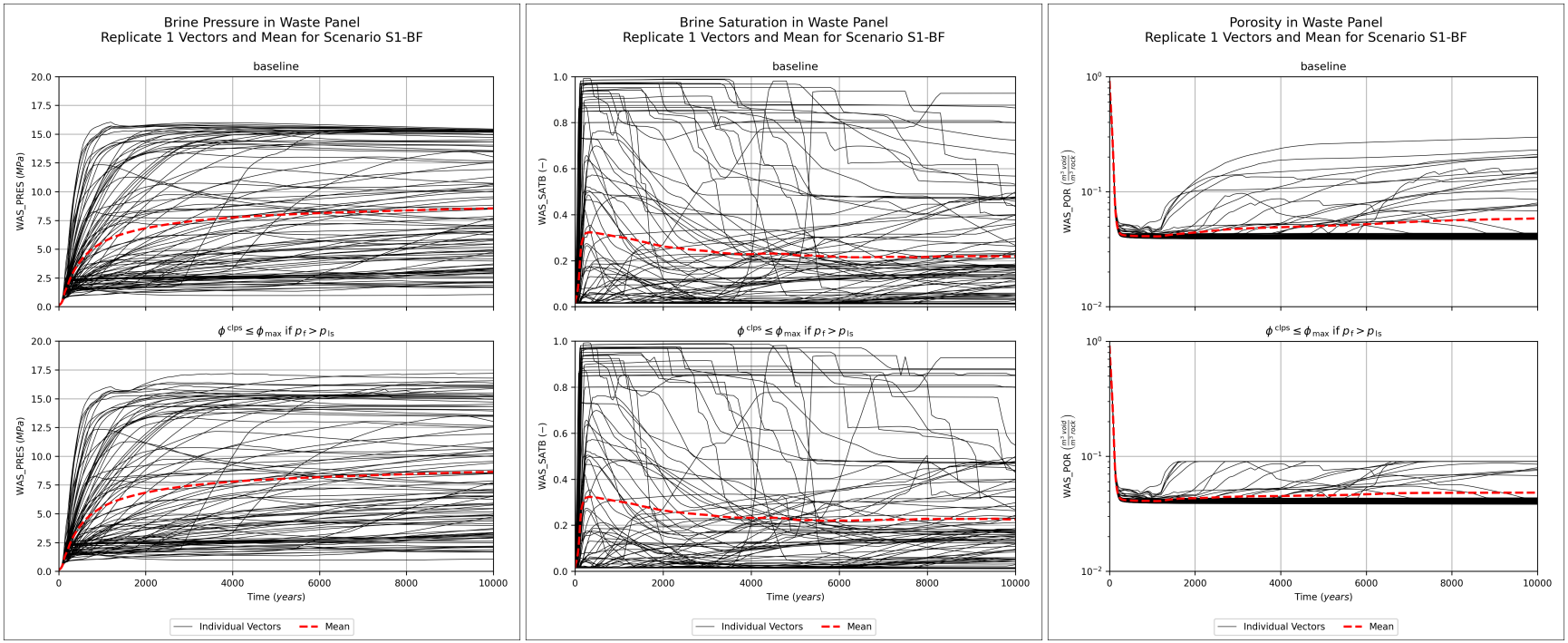


Figure 3-2. Replicate 1 brine pressure, brine saturation, and porosity individual histories and mean in the representative waste panel for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathbf{clps}} \leq \phi_{\mathbf{max}}$ if $p_{\mathbf{f}} > p_{\mathbf{ls}}$) subject to Scenario S1-BF

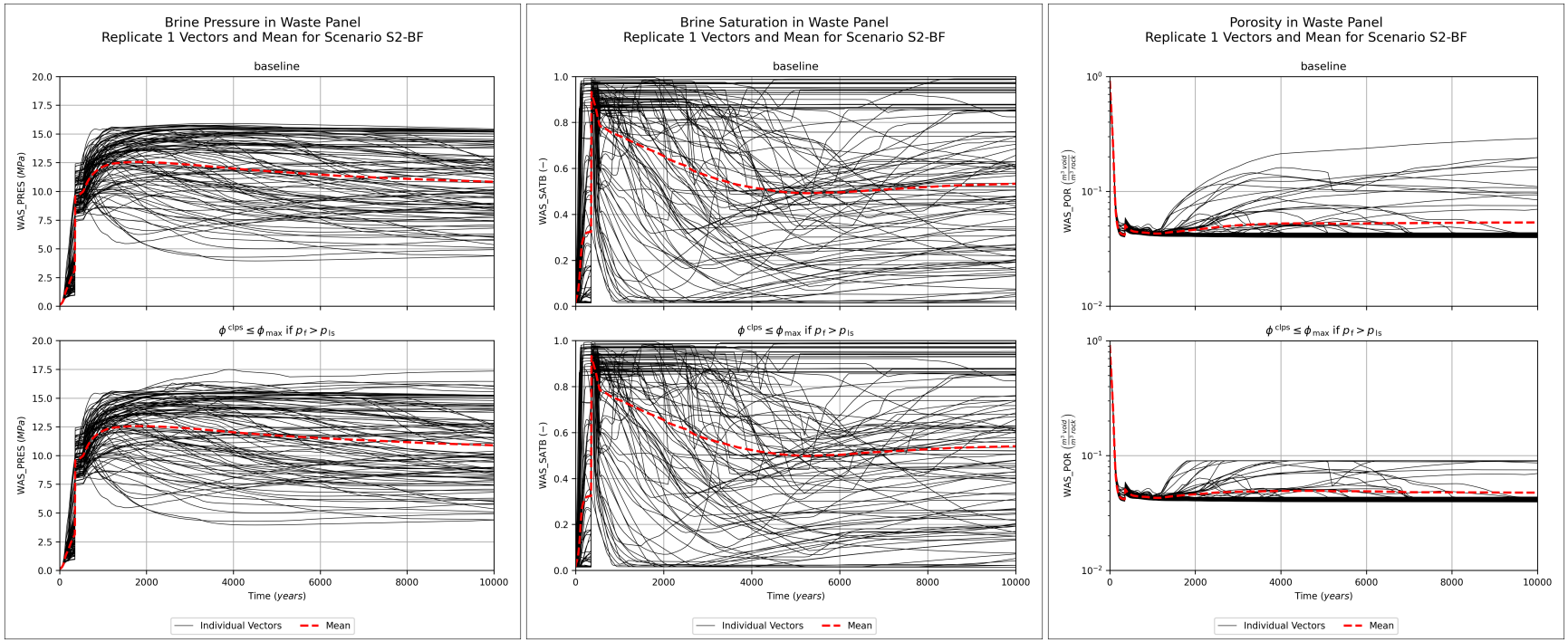


Figure 3-3. Replicate 1 brine pressure, brine saturation, and porosity individual histories and mean in the representative waste panel for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{clps} \leq \phi_{max}$ if $p_f > p_{ls}$) subject to Scenario S2-BF

conditionally limiting the porosity for those few realizations produces moderate brine pressure increases and minor brine saturation increases, as expected. The effects on the mean brine pressures and saturations in Fig. 3-4, however, are nearly negligible because the conditionally limited porosity surface only reduces the pore space for 3.7 % of all data points.

The slight overall increase in brine pressure and saturation cause insignificant increases in direct brine and spalling releases (see Fig. 3-4), which translate to a similarly insignificant increase in total releases (see Fig. 3-5). The percent difference for total release means at the high-probability (P(R) > 0.1) and low-probability (P(R) > 0.001) compliance points are approximately 0.03 % and 1.52 %, respectively, as a result of the conditionally limited porosity surface. This small sensitivity suggests that employing the uncoupled new porosity surface, instead of a fully coupled approach, does not lead to significant errors in radionuclide release predictions.

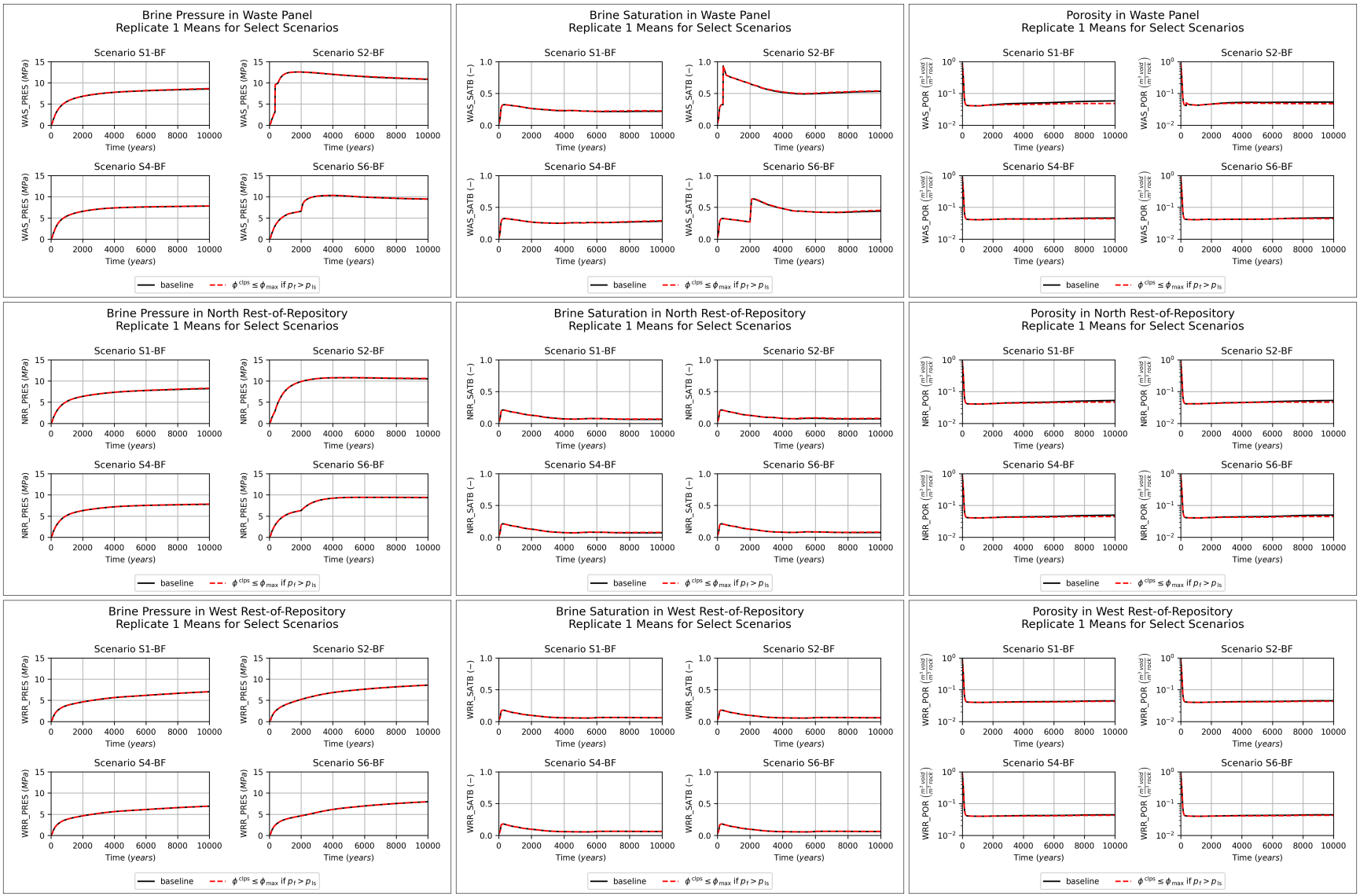
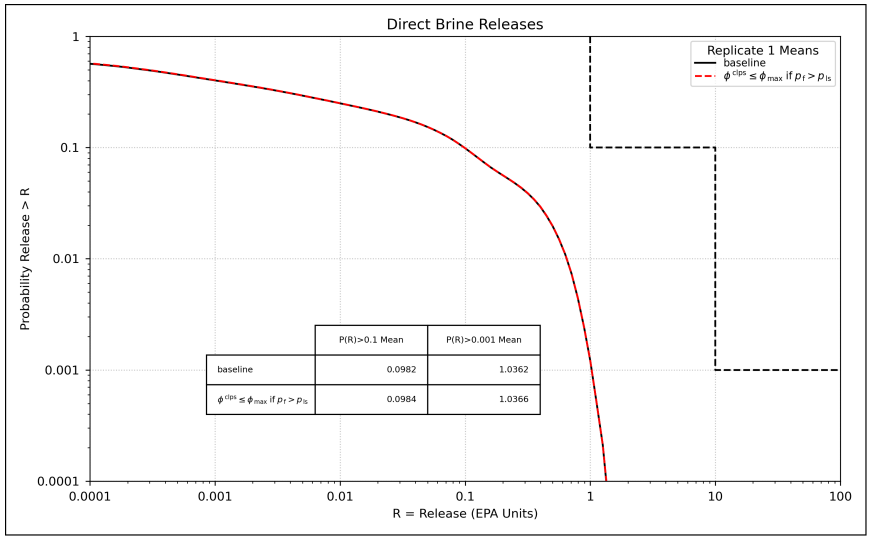
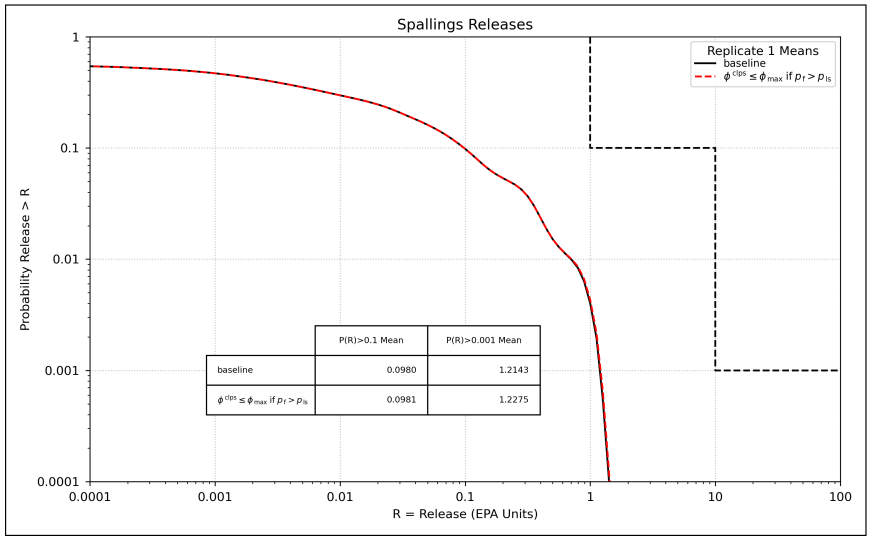


Figure 3-4. Replicate 1 brine pressure, brine saturation, and porosity means in the representative waste panel, north rest-of-repository, and west rest-of-repository for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{clps} \le \phi_{max}$ if $p_f > p_{ls}$) subject to Scenarios S1-BF, S2-BF, S4-BF, and S6-BF





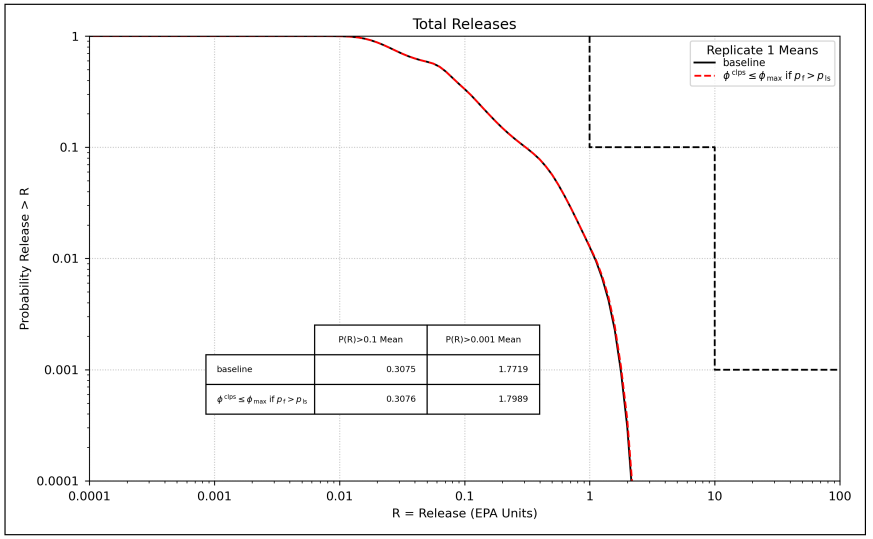
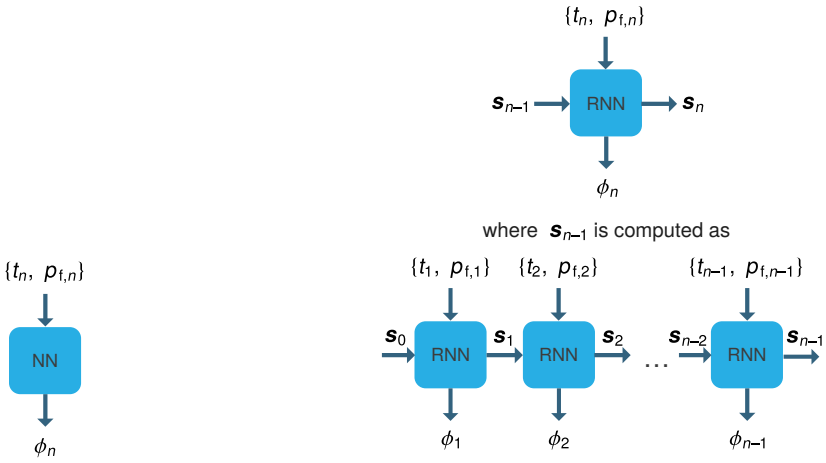


Figure 3-5. Replicate 1 direct brine, spallings, and total mean releases for the new porosity surface (baseline) and a conditionally limited variant ($\phi^{\mathbf{clps}} \le \phi_{\mathbf{max}}$ if $p_{\mathbf{f}} > p_{\mathbf{ls}}$)

4. PRELIMINARY MACHINE LEARNED MODEL FOR POROSITY

Machine-learned (ML) models for porosity have the potential to reasonably approach the accuracy of the fully coupled approach, yet run orders of magnitude faster. If successful, the ML model could eventually replace the porosity surface in the geohydraulic model. To investigate this approach's potential, a preliminary ML model was developed and documented in this chapter.

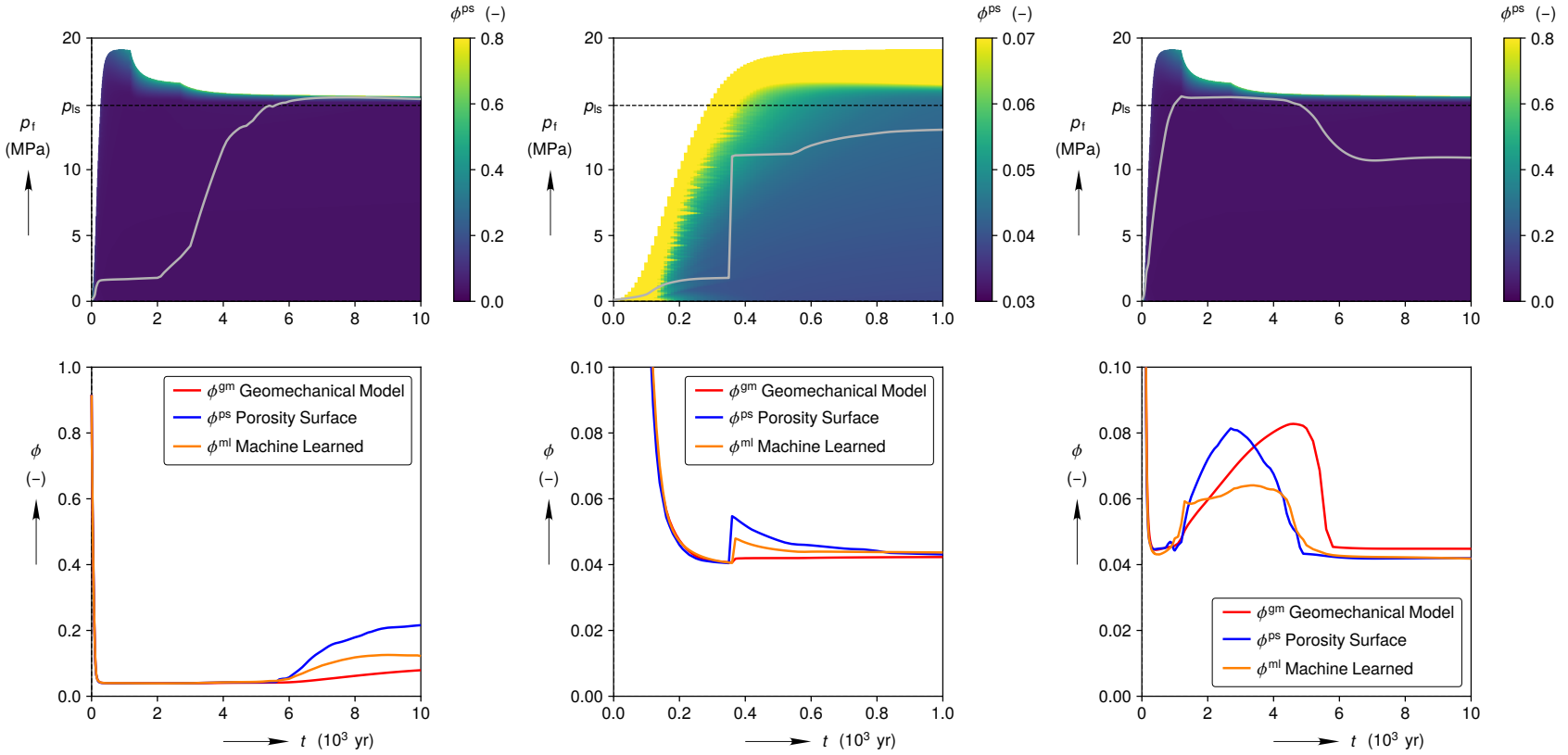


(a) Neural Network **(b) Recurrent Neural Network**

Figure 4-1. Simple schematics of a NN and an RNN.

The preliminary ML model used a Long-Short Term Memory (LSTM) Neural Network, which is a specific type of Recurrent Neural Network (RNN). As shown in Fig. 4-1a, a plain Neural Network (NN) can be set up to take the current time t_n and fluid pressure $p_{\mathrm{f},n}$ at step n as inputs and return the current porosity ϕ_n at step n. This simple approach is sufficient in many applications, but it has no means to make ϕ_n depend on the fluid pressure history prior to $\{t_n,\ p_{\mathrm{f},n}\}$. An RNN, on the other hand, can include path dependence. As shown in Fig. 4-1b, an RNN allows ϕ_n to depend on $\{t_n,\ p_{\mathrm{f},n}\}$ *and* a vector of internal state variables $\boldsymbol{s}_{n-1}$ from step $n-1$. This vector of internal state variables starts as $\boldsymbol{s}_0$ and is updated by the RNN on every step, such that $\boldsymbol{s}_{n-1}$ contains the integrated effects of the pressure history through step $n-1$. The RNN stores $\boldsymbol{s}_{n-1}$, so the geohydraulic code would only need to input $\{t_n,\ p_{\mathrm{f},n}\}$, not the entire fluid pressure history, into the RNN to obtain ϕ_n.

The LSTM parameters were trained, tested, and validated against the p_{f} histories and corresponding new geomechanical model ϕ^{gm} histories discussed in Section 2.1. The training, testing, and validation data consisted of 70 %, 20 %, and 10 % of the 600 cases, respectively. The LSTM was implemented within the JAX Python ecosystem (Bradbury et al. 2018). (JAX uses



(a) Gradual build to a high pressure (Scenario S6-BF, Vector 5)

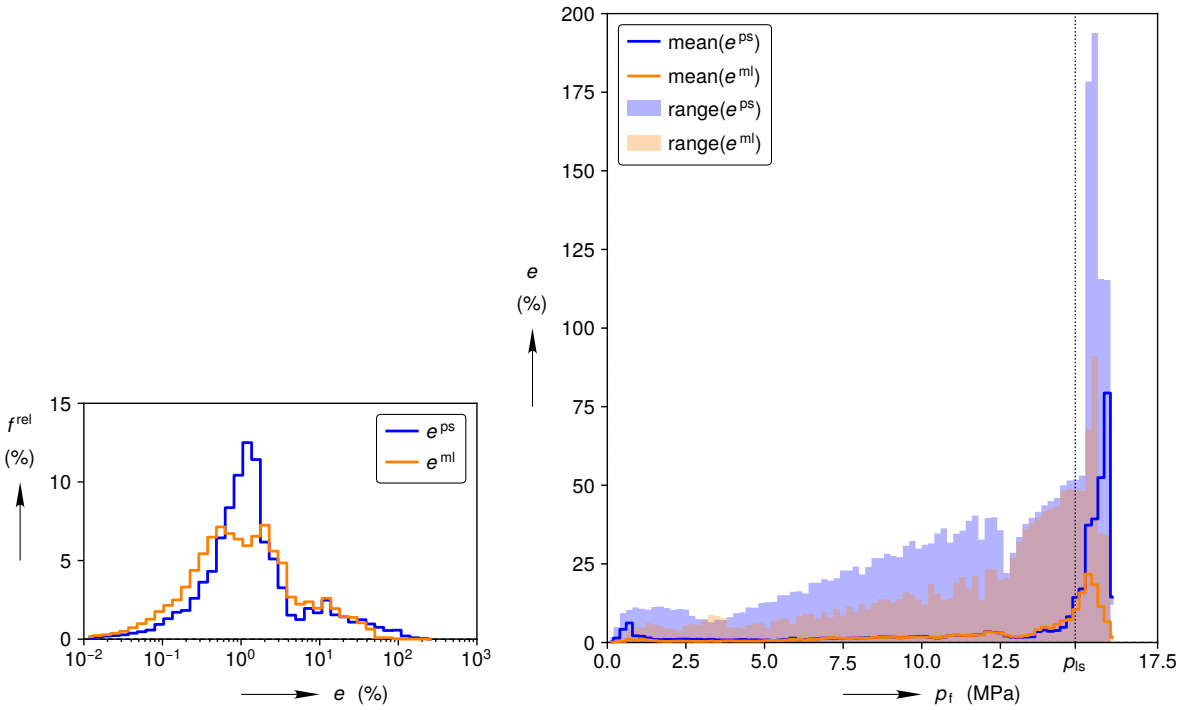
(b) Early abrupt pressure increase (Scenario S2-BF, Vector 7)

(c) Above p_{ls} then below p_{ls} (Scenario S5-BF, Vector 28)

Figure 4-2. Fluid pressure histories overlaid on the new porosity response surface (top row of plots), as well as porosities predicted by the preliminary ML model, the new geomechanical model, and the new legacy porosity surface (bottom row of plots) for select cases.

Just-In-Time compilation to make the LSTM run efficiently.) The Adams optimizer, also implemented in JAX, optimized the LSTM parameters to minimize the differences between the porosity ϕ^{ml} predicted by the ML model and ϕ^{gm}. The optimized LSTM parameters are not documented herein because there are simply too many values to list for a preliminary model, but they can be extracted from the archive listed at the end of Chapter 1. After training the ML model, the ϕ^{ml} histories corresponding to the training, testing, and validation data sets agreed with the ϕ^{gm} histories by roughly the same amount, so no further effort was made to distinguish between the different data sets in this chapter.

Fig. 4-2 shows the performance of the ML model for three specific cases. These three cases were previously studied in Section 2.2.2 because they are good examples of significant discrepancies between ϕ^{gm} and ϕ^{ps}. The gradual pressure increase to $p_{\text{f}} > p_{\text{ls}}$ in Fig. 4-2a causes the ML model to more closely match ϕ^{gm} than the porosity surface approach, but differences between ϕ^{ml} and ϕ^{gm} remain. Similarly, the abrupt pressure increase in Fig. 4-2b evokes a smaller response from the ML model than the porosity surface approach, but the ϕ^{ml} jump is still too large compared to the small ϕ^{gm} jump. The case in Fig. 4-2c, on the other hand, shows that ϕ^{ml} can sometimes be no better at predicting ϕ^{gm} than ϕ^{ps}.



(a) Error Distribution **(b) Pressure Dependence of Error**

Figure 4-3. The porosity error associated with the new porosity surface and the machine learned model for porosity for all 600 cases combined. Note that the error ranges in the lower plot of (b) are shown using transparent colors, which leads to three potential colors for the shaded regions: light blue for range(e^{ps}), light orange for range(e^{ml}), and bluish orange where range(e^{ps}) and range(e^{ml}) overlap.

The ML model porosity error was analyzed in the same manner as the porosity surface's error.

The relative error history for each pressure history was computed as

$$e^{\mathrm{ml}} = \frac{\left|\phi^{\mathrm{ml}} - \phi^{\mathrm{gm}}\right|}{\phi^{\mathrm{gm}}}. \tag{4.1}$$

The methods described at the start of Section 2.2 were again used to pool the 600 cases together, compute the e^{ml} distribution, and calculate the pressure dependence of e^{ml}.

The error distribution in Fig. 4-3a shows that the ML model is more accurate than the porosity surface approach, but room for improvement remains. The ML error distribution in Fig. 4-3a has more small, desirable, errors below 0.5 % and less large, undesirable, errors above 30 % than the porosity surface error distribution. Averaging e^{ml} over all 600 cases gives 3.4 %, which is a decent refinement over the 5.1 % for e^{ps}. Nevertheless, the ML model still has plenty of moderate, undesirable, errors between 5 % and 30 %.

Similar conclusions can be reached upon examining the pressure distribution of error in Fig. 4-3b. The mean ML model error mean(e^{ml}) is smaller than mean(e^{ps}) for both 0.4 MPa $\leq p_{\mathrm{f}} \leq$ 0.8 MPa and 14.6 MPa $\leq p_{\mathrm{f}}$. The reduction at low pressures is due to more accurately predicting the early room closure process ($0 < t \leq 0.12 \times 10^3$ yr), while the reduction at high pressures is due to the ML model responding more slowly than the porosity surface approach, as shown in Fig. 4-2a. The range of ML model error range(e^{ml}) is also smaller than range(e^{ps}) for nearly all pressures. Despite these improvements, mean(e^{ml}) $\geq$ 5 % for 14.2 MPa $\leq p_{\mathrm{f}} \leq$ 16 MPa and range(e^{ml}) reaches as high as 91 % for the bin centered at 15.5 MPa.

The advancements in this chapter are encouraging, but insufficient to justify implementing the ML model into the geohydraulic code at this juncture. The geohydraulic code is written in Fortran while the ML model is written in Python. It may be possible to link these two codes together, but it could be necessary to simply rewrite the ML model in Fortran. It, therefore, seems best to wait to implement a more polished ML model.

5. SUMMARY

The WIPP PA needs porosity estimates to compute the likelihood of direct brine releases and spalling releases. The most accurate method to predict porosity involves coupling geomechanical and geohydralic simulations together, but this fully coupled approach is computationally intractable. As a result, the WIPP PA links the geomechanical and geohydraulic simulations by interpolating the porosity at a given time and fluid pressure from a "look-up table" that is known as the porosity response surface. This uncoupled porosity surface approach is sufficient if the fluid pressure history being considered by the geohydralic model is similar to the pressure histories used to create the porosity response surface. It, however, can lead to significant porosity errors if the fluid pressure history deviates from those histories because the uncoupled porosity surface approach is unable to represent the path dependent effects included in the geomechanical model.

The chief goal of this report was to assess the porosity error associated with the uncoupled porosity surface approach. The porosity error was quantified by constructing a set of 600 representative fluid pressure histories; inputing each pressure history into the porosity surface and the geomechanical model; and comparing the resulting porosities. The porosity errors from this procedure should be representative of the porosity errors that one would compute by comparing the porosity from the porosity surface against the porosity computed by the fully coupled approach. Two different porosity response surfaces were considered using this procedure: the legacy porosity surface and the new porosity surface.

The porosity errors associated with the legacy porosity surface were large. As expected, the porosity error for a given case was related to the deviation of the fluid pressure history from those histories used to construct the legacy porosity surface. Also as expected, the porosity surface approach responded faster to pressure changes than the geomechanical model. When all the 600 cases were pooled together, the average porosity error was 38.6 % and the most common error was $\approx$ 60 %. Sorting the pooled data into fluid pressure bins showed that the porosity error increased significantly with increasing fluid pressure, reaching as high as 125.3 % when averaged within a 200 kPa bin.

The average porosity error associated with the new porosity surface was about 1.3 % for fluid pressures below lithostatic pressure, but it still reached as high as 79.3 %, averaged within a 200 kPa bin, when pressures exceeded lithostatic. The errors at low to medium pressures were small because the initial room closure largely outran the fluid pressurization, which made both the porosity surface approach and geomechanical model only mildly sensitive to low to medium fluid pressures. In contrast, the large porosity errors at high pressures arose because (1) the porosity surface approach and geomechanical model were highly sensitive to fluid pressures above lithostatic and (2) the fluid pressure histories rarely matched those used to create the porosity surface.

The sensitivity of radionuclide releases to porosity errors resulting from the use of the new porosity surface was roughly quantified. To do so, a conditionally limited porosity surface was constructed, which first computed the porosity using the new porosity surface and then capped the resulting porosity to 9 % or less if fluid pressures exceeded lithostatic. This conditionally limited porosity surface produced smaller porosity errors: the largest average (approximate) porosity error within a 200 kPa bin was 27.9 %. Conditionally limiting the porosity increased fluid pressures and saturations for a handful of cases, as one might expect, but these increases translated to negligible increases in radionuclide releases.

Lastly, a preliminary ML model was developed in an effort to reduce the porosity error at high pressures. This ML model involved an RNN because they are commonly used and can capture some degree of path dependence. The ML model was trained against the fluid pressure histories and the corresponding porosity histories from the geomechanical model. The resulting ML model significantly reduced the error at high pressures: the largest average porosity error within a 200 kPa bin was 29.8 %. This reduction, nevertheless, was not enough to warrant adopting the ML model as a replacement for the porosity surface approach.

BIBLIOGRAPHY

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: http://github.com/jax-ml/jax.

Day, B. (May 2025). *WIPP Repository Performance Impact as Attributed to Use of the New vs Legacy Porosity Surface for RPPCR*. Sandia Memorandum.

DOE (1996). *Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant (October). 21 vols.* Tech. rep. DOE/CAO-1996-2184. US Department of Energy: Carlsbad Field Office.

DOE (2004). *Title 40 CFR Part 191, Subparts B and C, Compliance Recertification Application 2004 for the Waste Isolation Pilot Plant*. Tech. rep. United States Department of Energy, Waste Isolation Pilot Plant.

DOE (2009). *Title 40 CFR Part 191, Subparts B and C, Compliance Recertification Application 2009 for the Waste Isolation Pilot Plant*. Tech. rep. United States Department of Energy, Waste Isolation Pilot Plant.

DOE (2014a). *Title 40 CFR Part 191, Subparts B and C, Compliance Recertification Application 2014 for the Waste Isolation Pilot Plant*. Tech. rep. United States Department of Energy, Waste Isolation Pilot Plant.

DOE (2014b). *Title 40 CFR Part 191, Subparts B and C, Compliance Recertification Application 2014 for the Waste Isolation Pilot Plant, Appendix PORSURF-2014 Porosity Surface*. Tech. rep. United States Department of Energy, Waste Isolation Pilot Plant.

DOE (2019). *Title 40 CFR Part 191, Subparts B and C, Compliance Recertification Application 2019 for the Waste Isolation Pilot Plant*. Tech. rep. United States Department of Energy, Waste Isolation Pilot Plant.

DOE (Mar. 2024). *Planned Change Request for the Use of Replacement Panels 11 and 12*. Tech. rep. ERMS 579729. United States Department of Energy, Waste Isolation Pilot Plant.

Freeze, G. A. (Feb. 1996). *Repository Closure–Reasoned Argument for FEP Issue DR12*. Sandia Memorandum. ERMS 413328.

Freeze, G. A., Larson, K. W., and Davies, P. B. (Oct. 1995). *A Summary of Methods for Approximating Salt Creep and Disposal Room Closure in Numerical Models of Multiphase Flow*. Tech. rep. SAND94-0251. Sandia National Laboratories.

Hansen, C. W., Brush, L. H., Gross, M. B., Hansen, F. D., Park, B. Y., Stein, J. S., and Thompson, T. W. (Jan. 2004). *Effects of Supercompacted Waste and Heterogeneous Waste Emplacement on Repository Performance, Revision 2*. Tech. rep. ERMS 533551. Sandia National Laboratories.

Jensen, R., Broome, S., Herrick, C., and Reedlunn, B. (June 2023). *Improved Compaction Experiments and Modeling of Waste Isolation Pilot Plant Standard, Non-degraded, Waste*

Containers. Tech. rep. SAND2023-04823. Sandia National Laboratories. DOI: 10.2172/2430967.

Popp, T., Minkley, W., Wiedemann, M., and Salzer, K. (Nov. 2014). "Gas-pressure-induced integrity of salt formations". In: *Geological Society, London, Special Publications* 415.1, pp. 107–122. ISSN: 2041-4927. DOI: 10.1144/sp415.3. URL: http://dx.doi.org/10.1144/SP415.3.

Reedlunn, B. (2023). *Sierra/SolidMechanics Version 5.4.1 Qualification Package*. ERMS 577341. Sandia National Laboratories.

Reedlunn, B., Bean, J., Wilkes, J., and Bignell, J. (Jan. 2023). *Simulations of Criticality Control Overpack Container Compaction at the Waste Isolation Pilot Plant*. Tech. rep. SAND2023-10477. Sandia National Laboratories. DOI: 10.2172/2432229.

Reedlunn, B. and King, S. (2025). *Analysis Plan to Assess Approaches to Link Geomechanical and Geohydraulic Simulations*. Tech. rep. AP-209 Rev. 0. Sandia National Laboratories.

Sierra/SolidMechanics (Jan. 2022). *Sierra/SolidMechanics 5.4 User's Guide*. 5.4.1. SAND2022-0103. Sandia National Laboratories. Albuquerque, NM, USA; Livermore, CA, USA.

Stone, C. M. (Aug. 1997a). *Final disposal room structural response calculations*. Tech. rep. SAND97-0795. Albuquerque, NM, USA; Livermore, CA, USA: Sandia National Laboratories.

Stone, C. M. (1997b). *SANTOS: a two-dimensional finite element program for the quasistatic, large deformation, inelastic response of solids*. SAND90-0543. Sandia National Laboratories. Albuquerque, NM, USA; Livermore, CA, USA.

Vignes, C., Bean, J., and Reedlunn, B. (June 2023). *Improved Modeling of Waste Isolation Pilot Plant Disposal Room Porosity*. Tech. rep. SAND2023-04826. Sandia National Laboratories. DOI: 10.2172/2430956.

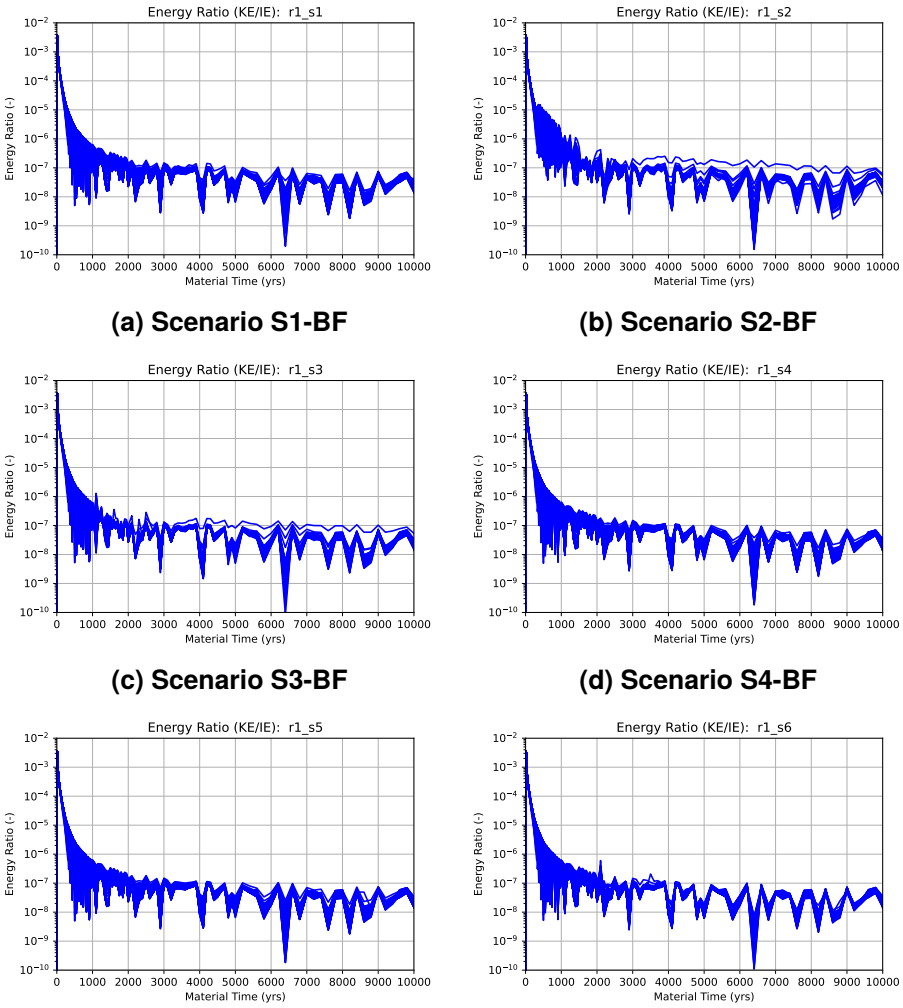
APPENDIX A. Explicit Time Integration With Viscoplastic Scaling

The preferred approach to solve the geomechanical model's quasi-static equilibrium equations would be to use Sierra/SM's implicit solvers. Past experience, however, has shown that these implicit solvers are not robust enough to easily handle hundreds of complex fluid pressure histories. Consequently, explicit time integration with viscoplastic scaling was employed in all the simulations herein.

Vignes et al. (2023, Appendix B.2) explains how the viscoplastic scaling compresses the physical (material) time t into a reduced simulation time, termed the momentum balance time t^{mb}. This compression allows one to solve the momentum balance equations using explicit time integration, which is typically much more robust than implicit time integration. One must be careful, though, to avoid compressing the physical time so much that inertial effects pollute the solution. The rock mass surrounding the room is likely to move fastest immediately after excavating the room. To avoid inertial effects during this relatively rapid room closure period, Reedlunn et al. (2023, Appendix A.2.1) and Vignes et al. (2023, Appendix B.2) gradually ramped up the scaling between an increment $\mathrm{d}t$ and an increment $\mathrm{d}t^{\mathrm{mb}}$. Reedlunn et al. (2023, Appendix A.2.1) found that this scheme produced results that closely approximated a reference implicit quasi-static solution for $p_{\mathrm{f}} = 0$ if the ratio of kinetic energy over internal energy was less than 5×10^{-3}. Small differences between the explicit and implicit solution occurred when the energy ratio was 5×10^{-2}.

All the geomechanical simulations herein used the same scaling scheme as Vignes et al. (2023, Appendix B.2) with the same parameters ($S_0 = 10^{-2}$ and $S_1 = 2 \times 10^9$ 1/s). Nearly all of these simulations did not have $p_{\mathrm{f}} = 0$, so care was taken to ensure that the applied p_{f} histories, which sometimes included abrupt changes in p_{f}, did not significantly affect the accuracy of the solution. Figs. A-1 and A-2 show, separated by model, the ratio of kinetic energy over internal energy versus t for the 100 simulations in each scenario. In all cases, the energy ratio starts at $\approx 5 \times 10^{-3}$ and then decreases to values orders of magnitude beneath 5×10^{-3}, indicating that inertial effects were negligible.

A few other notes about the results in Figs. A-1 and A-2 are interesting, even though they don't have practical impacts on the results in this report. After about 200 yr, the range of energy ratios increases for all scenarios and both geomechanical models. The legacy geomechanical model produces a tighter energy ratio range, at a given time, probably because it tends to respond slower than the new geomechanical model. (The legacy geomechanical model does not have clay seams, has zero friction between the waste and the room surfaces, and does not include salt's low stress creep behavior.) In addition, the new geomechanical model produces more energy ratio outliers after about 200 yr, which is probably related to the faster initial room closure predictions from the new geomechanical model. The legacy geomechanical model predicts room closure is far from a fully static state, even when $p_{\mathrm{f}} = 0$ and $t = 10 \times 10^3$ yr (see the porosity histories in Fig. 1-2). Any p_{f} change, therefore, causes the legacy geomechanical model to predict a change in porosity,



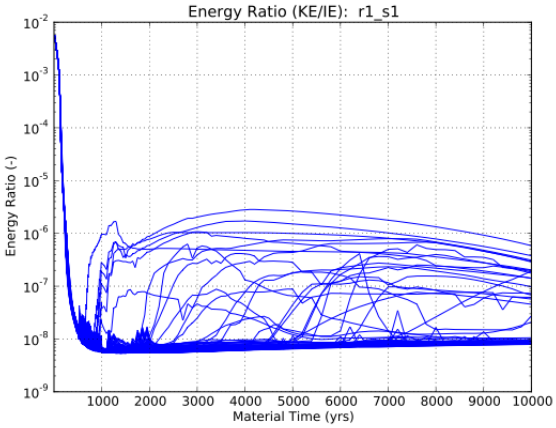
(a) Scenario S1-BF **(b) Scenario S2-BF**

(c) Scenario S3-BF **(d) Scenario S4-BF**

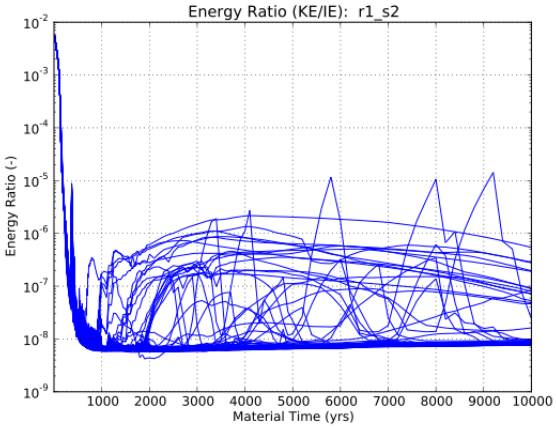
(e) Scenario S5-BF **(f) Scenario S6-BF**

Figure A-1. Ratio of kinetic energy over internal energy for the recreated legacy geomechanical model simulations in each of the six PA scenarios.

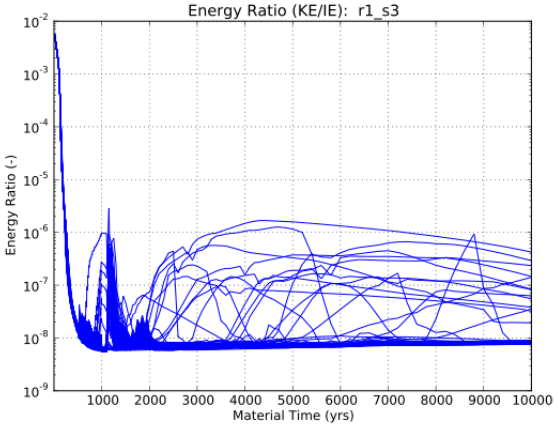
which has an impact on the energy ratio. The new geomechanical model, by contrast, is close to a fully static state after 200 yr for all the pressure histories considered (see Fig. 2-7). When $p_f < p_{ls}$ for $t > 200$ yr, the new geomechanical model predicts the room will remain largely static, so the energy ratio remains very small. In the few cases when $p_f > p_{ls}$ for $t > 200$ yr, the new geomechanical model predicts the room will expand, causing the energy ratio to increase and become an outlier.



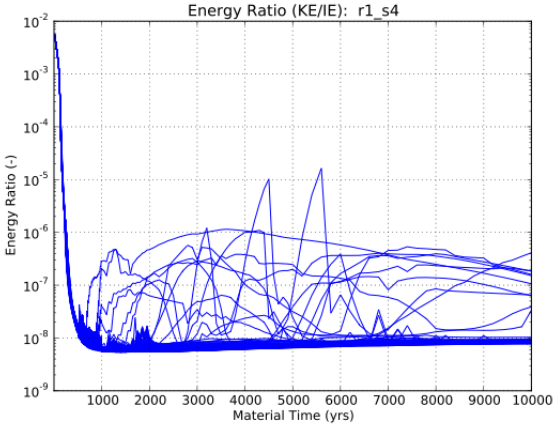
(a) Scenario S1-BF



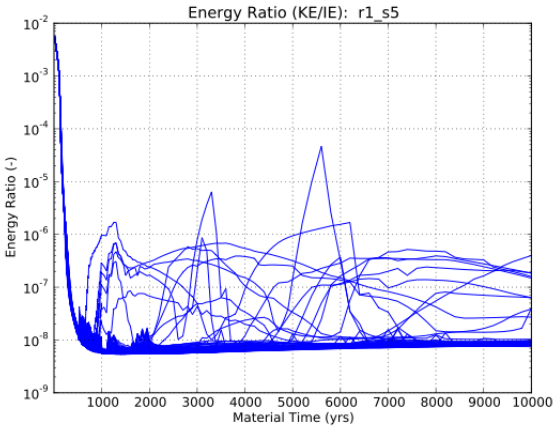
(b) Scenario S2-BF



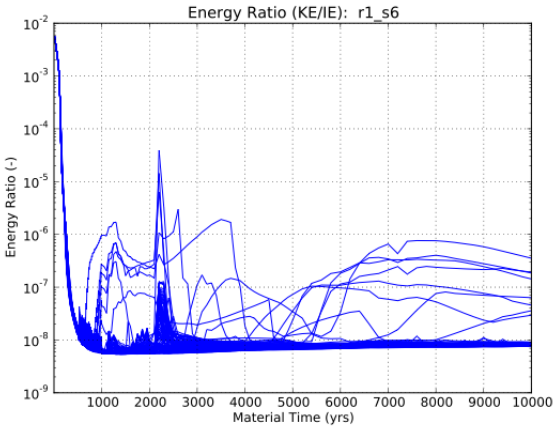
(c) Scenario S3-BF



(d) Scenario S4-BF



(e) Scenario S5-BF



(f) Scenario S6-BF

Figure A-2. Ratio of kinetic energy over internal energy for the new geomechanical model simulations in each of the six geohydraulic scenarios.

This page intentionally left blank.

APPENDIX B. Recreated Legacy Porosity Response Surface

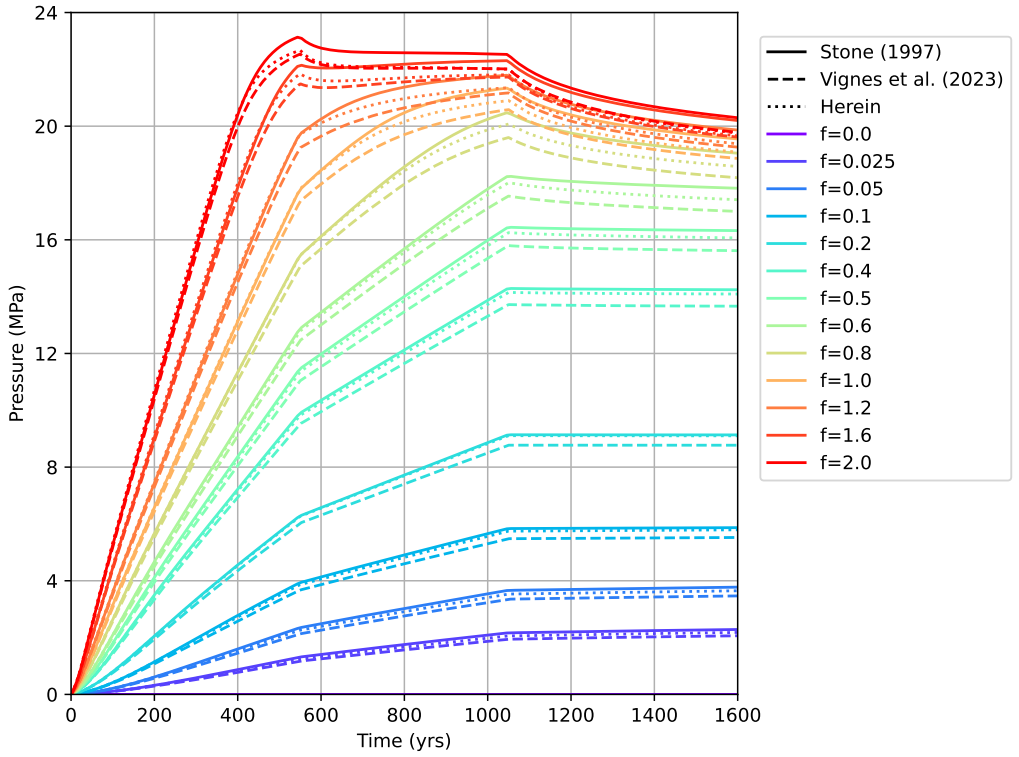
Stone (1997a) created the original legacy porosity response surface using the SANTOS finite element code. SANTOS explicitly integrated the equations of motion (with inertia effects) and then used a dynamic relaxation scheme to approximate the quasi-static solution for a given time step (Stone 1997b, Section 4.6). More recently, Vignes et al. (2023, Sec. 2.4 and App. B) recreated the legacy porosity response surface using Sierra/SM, where the quasi-static equilibrium equations were solved implicitly. As one expect, the original and recreated legacy porosity response surfaces differed slightly, but were largely the same.

The recreated legacy geomechanical simulations herein also used Sierra/SM, but the numerical setup deviated from that in Vignes et al. (2023, Sec. 2.4 and App. B) in two ways. As mentioned in Section 2.1, the recreated legacy geomechanical simulations herein used explicit time integration with viscoplastic scaling (see Appendix A) rather than implicitly solving the equilibrium equations, and the waste block was discretized with the selective deviatoric element instead of the uniform gradient element. Consequently, the legacy porosity response surface was recreated again using this revised numerical setup. The Sierra/SM input files for this second recreation of the legacy porosity response surface simulations can be found in Appendix C.

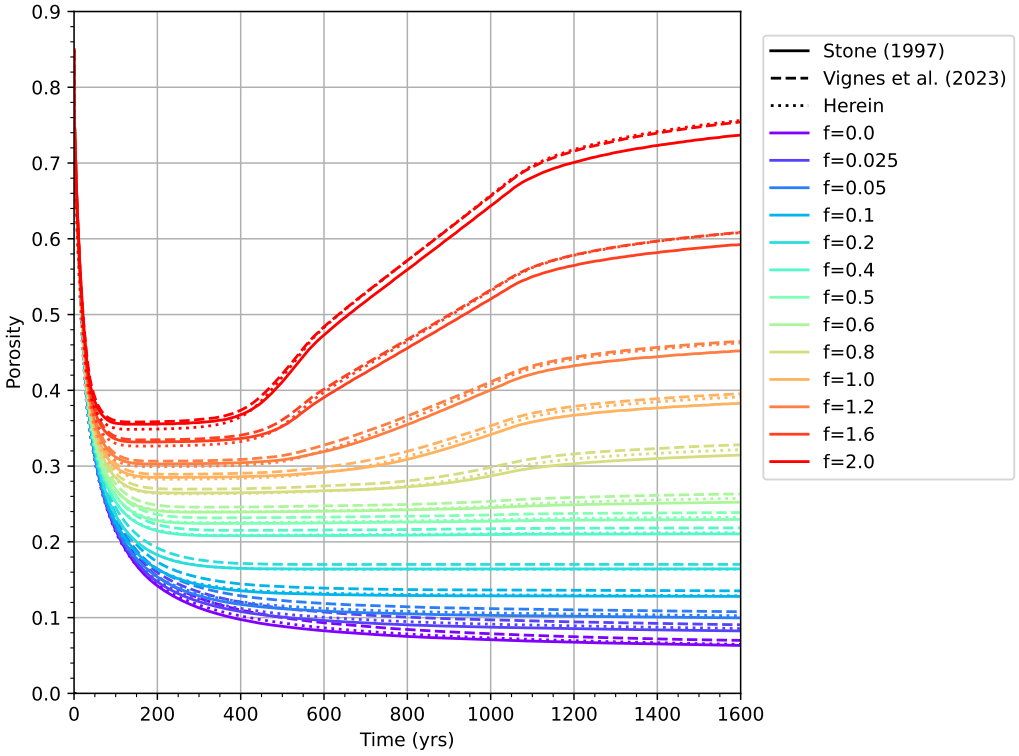
Fig. B-1 compares the p_{f} and ϕ^{ps} histories corresponding to each f value. The original legacy pressure and porosity curves from Stone (1997a), labeled as "Stone (1997)" in the figure legend, are shown as solid lines, while those from Vignes et al. (2023) are shown as dashed, and those computed herein are shown as dotted lines*. The agreement between the Stone (1997a) results and the two recreations is good considering that they were computed about 25 years apart and used different simulation codes. See Vignes et al. (2023, Sec. 2.4 and App. B.1) for discussion of the minor differences between the codes. The agreement between the two recreations is also reasonably good, but not as excellent as might be expected for two sets of results from the same simulation code. Switching from implicit to explicit time integration and/or switching from the UG to the SD element for the waste block affected the solution to a small, yet appreciable, degree. The root causes of these small differences were not tracked down, but one possibility is implicit and explicit time integration use different contact algorithms.

Given that the original (Stone 1997a) and second recreation (herein) results in Fig. B-1 are not identical, care was taken to only use the recreated (herein) legacy porosity surface in the geohydraulic simulations that employed the legacy porosity surface and in all plots of the legacy porosity surface.

*Note that Vignes et al. (2023, Fig. 3-1a) depicts Eulerian porosity, while Fig. B-1 depicts Lagrangian porosity. The transformation between these two porosity metrics is given in Vignes et al. (2023, Eqn. (3.3)), but is not restated in this report because only Lagrangian porosities are discussed herein.

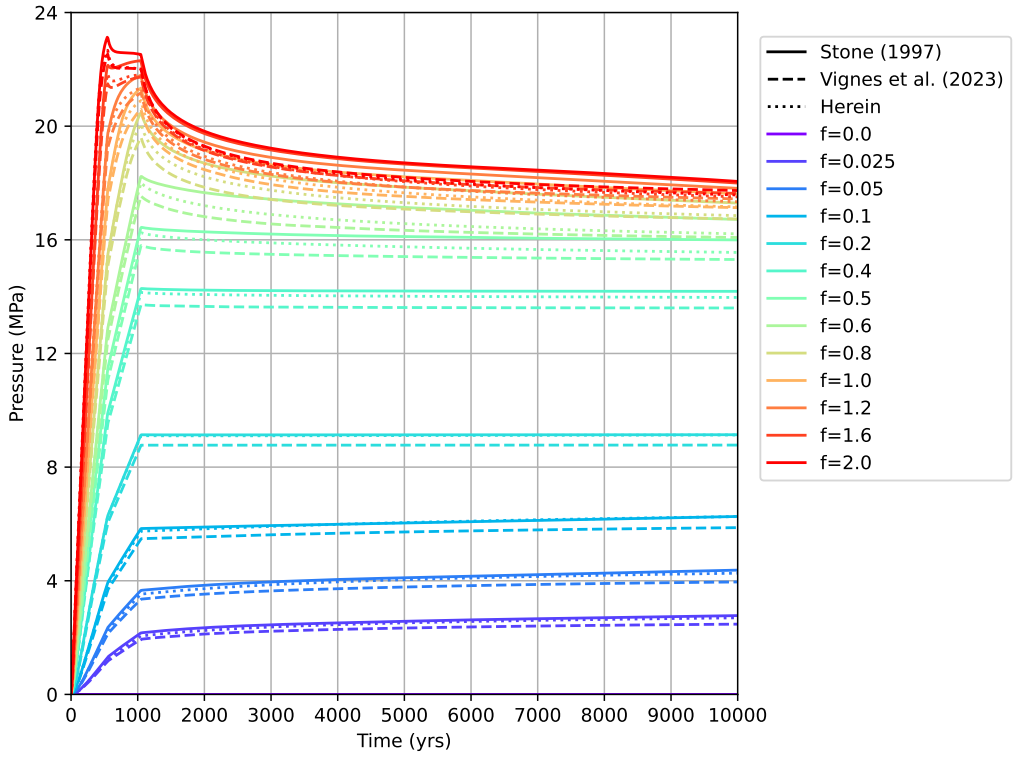


(a) Early pressure histories

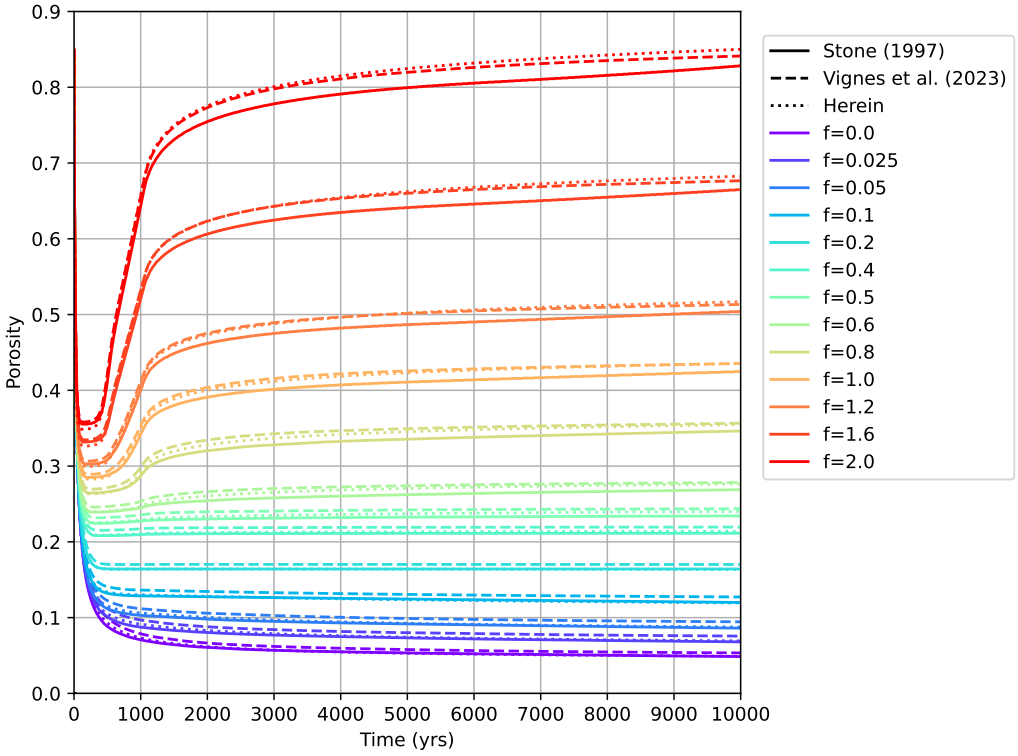


(b) Early porosity histories

Figure B-1. Comparison of porosity and fluid pressure histories from the original legacy model and two recreations of the legacy model for a lower horizon disposal room.



(c) Full pressure histories



(d) Full porosity histories

Figure B-1. (Continued) Comparison of porosity and fluid pressure histories from the original legacy model and two recreations of the legacy model for a lower horizon disposal room.

This page intentionally left blank.

APPENDIX C. Sierra/SM Input Files

C.1. Recreated Legacy Geomechanical Model

```
begin sierra WIPP Isothermal Disposal Room

   title Simulation of WIPP Isothermal Disposal Room

   #restart = automatic

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

   #-------------------------------------
   #---------- Room geometry  ---------
   #-------------------------------------

   #Define mesh name
   ${mesh_name = "disposal_room_CMStone-1_element_deep.g"}

   #Define the disposal room horizon ('lower' or 'upper')
   ${horizon = "lower"}

   #Define if disposal room empty or with waste ('empty' or 'waste')
   ${iswaste = "waste"}

   #Define element thickness (m)
   ${element_thickness = 1.0}

   #-------------------------------------
   #---------- Waste model  -----------
   #-------------------------------------

   #Define waste material model ('soil_foam', 'foam_damage', or 'elastic')
   ${waste_model = "soil_foam"}

   #Define foam damage model ('model_F2', 'model_F3', or 'model_F4')
   ${foam_model = "model_F3"}

   #-------------------------------------
   #---------- Gas generation  --------
   #-------------------------------------

   #Set the gas generation curve ('stone1997' or 'king2021' or 'phistory')
   # Stone 1997: SAND97-0795
   # King: ERMS-575793
   # phistory: predefined pressure history to apply to room surfaces
   ${gasCurve = "stone1997"}

   #Specify the gas generation factor if computing a porosity response surface
   #Set to 1.0 when using a predefined pressure history
   ${gasScale = GASVALUE}

   #Define if gas generation pressure masking active ('mask' or 'nomask')
   ${masking = "nomask"}

   #Define if gas generation pressure applied to clay seams ('yes' or 'no')
   ${clay_pressure = "no"}

   #-------------------------------------
   #---------- Contact options  -------
   #-------------------------------------

   #Define friction coefficients
   ${clay_seam_friction  = 0.2}  # not used
   ${clay_seam_frictionF = 0.42} # not used
   ${salt_salt_friction  = 0.0}
   ${salt_waste_friction = 0.0}

   #Define if individual clay seams are tied ('yes' or 'no')
   ${tied_A = "no"}
   ${tied_B = "no"}
   ${tied_D = "no"}
   ${tied_E = "no"}
   ${tied_F = "no"}
   ${tied_G = "no"}
   ${tied_H = "no"}
   ${tied_I = "no"}
```

```
${tied_J = "no"}
${tied_K = "no"}
${tied_L = "no"}
${tied_M = "no"}

#-----------------------------------
#--- Simulation time and scaling ---
#-----------------------------------

#Define simulation time period (yrs)
${time_period = 10000}

#Define time integration method ('implicit', or 'explicit')
${time_integration = "explicit"}

{if (gasCurve=="stone1997")}
  $yr1 = {yr1 = 550.0}  end of first gas generation period
  $yr2 = {yr2 = 1050.0} end of second gas generation period
{elseif (gasCurve=="king2021")}
  $yr1 = {yr1 = 1200.0} end of first gas generation period
  $yr2 = {yr2 = 2700.0} end of second gas generation period
{endif}

{if (time_integration=="implicit")}

#Define the time periods
${t_0_0 = -0.01}     #s
${t_0_1 = 0.0}       #s
${t_1_0 = t_0_1}     #s
${t_1_1 = 86400.0 * 365.25 * time_period} #s

#Define the gas generation periods
${t1 = yr1 * 86400.0 * 365.25}
${t2 = yr2 * 86400.0 * 365.25}

{else}

#Define the time periods in material time (sec)
${t_m_0_0 = -0.5}    #s
${t_m_0_1 = 0.0}     #s
${t_m_1_0 = t_m_0_1} #s
${t_m_1_1 = 86400.0 * 365.25 * time_period} #s

#Define the scale factors
${s_1_0 = 1.0e-2}
${s_ramp = 2e9}

#Define the time periods in momentum balance time (sec)
$t_mb_m_0_0  = {t_mb_m_0_0 = -0.01} #s
$t_mb_m_0_1  = {t_mb_m_0_1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (t_m_0_1))) / s_ramp} #s
$t_mb_m_1_0  = {t_mb_m_1_0 = t_m_0_1}
$t_mb_m_1_1  = {t_mb_m_1_1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (t_m_1_1))) / s_ramp} #s

#Define the gas generation periods in momentum balance time (sec)
$ t1 = {t1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr1 * 86400.0 * 365.25))) / s_ramp }
$ t2 = {t2 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr2 * 86400.0 * 365.25))) / s_ramp }

{endif}

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

define direction y with vector 0.0 1.0 0.0
define direction x with vector 1.0 0.0 0.0
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 definition for function function_constant
   type is piecewise linear
   begin values
     -1.0e16                1.0
      1.0e16                1.0
   end values
end definition for function function_constant

begin function gas_pressure
  {if (gasCurve=="phistory")}
      type is piecewise linear
  {else}
      type is piecewise analytic
  {endif}
  expression variable: rv        = global room_volume
  {if (time_integration=="implicit")}
  expression variable:  t        = global time
  {else}
  expression variable:  t        = global mat_time
  {endif}
  expression variable: fricModel = nodal contact_friction_index
  {if (masking=="mask")}
    expression variable: cstrn   = nodal contact_strength
```

```
  {else}
    expression variable: cstrn   = global cstrn_off
  {endif}
  {if (gasCurve=="stone1997")}
    begin expressions
     -1.0  "0.0"
      0.0  "fricModel==5 ? (1.0 - min(1.0,cstrn))*13608.0*(t/3.15576e7)*2494.35/(rv-551.2) : 13608.0*(t/3.15576e7)*2494.35/(rv-551.2)"
      {t1} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(3742200.0 + 6804.0*(t/3.15576e7))*2494.35/(rv-551.2) : (3742200.0 +
     ↪  6804.0*(t/3.15576e7))*2494.35/(rv-551.2)"
      {t2} "fricModel==5 ? (1.0 - min(1.0,cstrn))*10886400.0*2494.35/(rv-551.2) : 10886400.0*2494.35/(rv-551.2)"
    end
  {elseif (gasCurve=="king2021")}
    begin expressions
     -1.0  "0.0"
      0.0  "fricModel==5 ? (1.0 - min(1.0,cstrn))*9846.38*(t/3.15576e7)*2494.35/(rv-302.0) : 9846.38*(t/3.15576e7)*2494.35/(rv-302.0)"
      {t1} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(9396876.0 + 2015.65*(t/3.15576e7))*2494.35/(rv-302.0) : (9396876.0 +
     ↪  2015.65*(t/3.15576e7))*2494.35/(rv-302.0)"
      {t2} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(13244376.0 + 590.65*(t/3.15576e7))*2494.35/(rv-302.0) : (13244376.0 +
     ↪  590.65*(t/3.15576e7))*2494.35/(rv-302.0)"
    end
   {elseif (gasCurve=="phistory")}
     begin values
       includefile PCURVE
     end
  {endif}
end function gas_pressure

begin definition for 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
   {if (time_integration=="implicit")}
   evaluate expression is "2300.0 / 2.0 * (1.0 - cos(pi * time / {t_0_1 - t_0_0}))"
   {else}
   evaluate expression is "2300.0 / 2.0 * (1.0 - cos(pi * time / {t_m_0_1 - t_m_0_0}))"
   {endif}
end definition for function fluid_density_ramp

{if (time_integration=="explicit")}
#---------- Viscoplastic Rate Scaling Functions ---------

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
   type is piecewise analytic
   begin expressions
      -0.5              "{s_1_0}"
      1e-10             "{s_1_0} + {s_ramp} * (time - 0)"
   end expressions
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
{endif}

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

includefile include/salt-md_viscoplastic-stone_1997.txt

includefile include/anhydrite-stone_1997.txt

{if (waste_model=="foam_damage")}
  {if (foam_model=="model_F2")}
    includefile include/foamDamageModel_F2.txt
  {elseif (foam_model=="model_F3")}
    includefile include/foamDamageModel_F3.txt
  {elseif (foam_model=="model_F4")}
    includefile include/foamDamageModel_F4.txt
  {endif}
{else}
```

```
  includefile include/soil_and_foam.txt
{endif}

begin solid section solid_1
   formulation = mean_quadrature
   strain incrementation = strongly_objective
end solid section solid_1

begin solid section solid_2
   formulation = selective_deviatoric
   strain incrementation = strongly_objective
end solid section solid_2

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

begin finite element model room
   Database name = {mesh_name}
   Database type = exodusII

   begin block defaults
      hourglass stiffness = 0.005 # default is 0.05
   end

   begin parameters for block b_halite
      material halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_halite

   begin parameters for block b_arg_halite
      material arg_halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_arg_halite

   begin parameters for block b_arg_halite_a
      material arg_halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_arg_halite_a

   begin parameters for block b_arg_halite_b
      material arg_halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_arg_halite_b

   begin parameters for block b_arg_halite_c
      material arg_halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_arg_halite_c

   begin parameters for block b_anhydrite
      material anhydrite
      solid mechanics use model soil_foam
      section = solid_1
   end parameters for block b_anhydrite

   {if (iswaste=="waste")}
   begin parameters for block b_homogenized_can_block
     material homogenized_can_block
     solid mechanics use model {waste_model}
     section = solid_2
    #hourglass stiffness = 0.05 # other blocks are using reduced value
   end parameters for block b_homogenized_can_block
   {endif}

end finite element model room

   {if (time_integration=="implicit")}

   begin adagio procedure The_Procedure.1

   begin time control

       begin time stepping block p0
         start time = {t_0_0}
         begin parameters for adagio region SmRegion.1
           time increment = {(t_0_1 - t_0_0) / 100.0}
         end
       end

       begin time stepping block p1
         start time = {t_1_0}
         begin parameters for adagio region SmRegion.1
           time increment = 1.0e-3
         end
       end

       termination time = {t_1_1}
```

```
{else}

begin presto procedure The_Procedure.1

begin time control

   begin time stepping block p0
      start time = {t_mb_m_0_0}
      begin parameters for presto region SmRegion.1
         time step scale factor = 0.9  #1
         time step increase factor = 5
      end
   end

   begin time stepping block p1
      start time = 0
      begin parameters for presto region SmRegion.1
         time step scale factor = 0.9 #1
         time step increase factor = 5
      end
   end

   termination time = {t_mb_m_1_1}
   {endif}

end time control

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

{if (time_integration=="implicit")}
begin adagio region SmRegion.1

   begin adaptive time stepping time
       method = material during p1
       cutback factor = 0.5
       growth factor  = 1.1
       minimum multiplier = 1.e0 during p0
       minimum multiplier = 1.e-3 during p1
       maximum multiplier = 1e14
       maximum failure cutbacks = 50
   end adaptive time stepping time

{else}
begin presto region SmRegion.1
{endif}

   use finite element model room

   #--------- 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
      function = function_constant
   end gravity

   begin prescribed temperature
      block = b_halite b_arg_halite b_arg_halite_a b_arg_halite_b b_arg_halite_c b_anhydrite # b_polyhalite
      function = function_constant
      scale factor = 300.0
   end prescribed temperature

   {if (iswaste=="waste")}
   begin prescribed temperature
      block = b_homogenized_can_block
      {if (waste_model=="foam_damage" && time_integration=="explicit")}
        function = vp_rate_scale_factor_function
        scale factor = 1.0
      {else}
        function = function_constant
        scale factor = 300.0
     {endif}
   end prescribed temperature
   {endif}

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

   begin pressure
     surface = SS_room_surf
     function = gas_pressure
     scale factor = {gasScale}
     external force contribution output name = pgas_room
   end pressure

   begin fixed displacement
      node set = NS_bot_surf
```

```
   components = z
end fixed displacement

begin fixed displacement
   node set = NS_left_surf
   components = x
end fixed displacement

begin fixed displacement
   node set = NS_right_surf
   components = x
end fixed displacement

begin fixed displacement
   node set = NS_back_surf
   components = y
end fixed displacement

begin fixed displacement
   node set = NS_front_surf
   components = y
end fixed displacement

#The waste front, back, and left side are included in NS_front_surf, NS_back_surf, and NS_left_surf.

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 fluid_pressure

   #      --------- CONTACT MODEL --------------

begin contact definition contact_defn

   begin enforcement options
      {if (time_integration=="explicit")}
      momentum balance iterations = 20
      {endif}
   end

   begin interaction defaults
      general contact = on
      constraint formulation = face_face
   end

   begin constant friction model salt_to_salt_friction_model
      friction coefficient = {salt_salt_friction}
   end

   begin contact surface room_surf
      surface = SS_room_surf
   end

   begin interaction room_surf_to_room_surf
      surfaces = room_surf room_surf
      friction model = salt_to_salt_friction_model
   end

   {if (iswaste=="waste")}
   begin constant friction model salt_to_waste_friction_model
      friction coefficient = {salt_waste_friction}
   end

   begin contact surface homogenized_can_block_surf
      surface = SS_homogenized_can_block_exposed_surf
   end

   begin interaction room_surf_to_waste_surf
      surfaces = room_surf homogenized_can_block_surf
      friction model = salt_to_waste_friction_model
   end
   {endif}

   begin dash options
```

```
      subdivision level = 3
   end

end contact definition contact_defn

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

begin initial condition
   block = b_halite b_arg_halite b_arg_halite_a b_arg_halite_b b_arg_halite_c b_anhydrite # b_polyhalite
   #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 initial condition

{if (time_integration=="explicit")}
#--------- Viscoplastic Rate Scaling ---------

# See note above stored_time_function.
begin user variable stored_time
   type = global real length = 1
   global operator = min
   initial value = -0.5
end

begin user variable mat_time
   type = global real length = 1
   global operator = min
   #edit the following line for restart runs!
   initial value = -0.01
end

begin user variable vp_rate_scale_factor
   type = global real length = 1
   global operator = min
   initial value = 0.02
end
{endif}

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

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

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

begin user output
   node set = NS_right_wall_ctr
   compute global right_wall_disp as average of nodal displacement
end

begin user variable room_volume
   type = global real length = 1
   global operator = sum
   initial value = 3.642750144 0e+03   # W (10.06) * L (91.44) * H (3.96)
end user variable

begin user output
   {if (horizon=="lower")}
   surface = SS_room_surf # SS_clay_F_master SS_clay_F_slave
   {elseif (horizon=="upper")}
   surface = SS_room_surf # SS_clay_F_master SS_clay_F_slave SS_clay_G_master SS_clay_G_slave
   {endif}
   surface subroutine = aupst_cavity_volume
   subroutine string parameter: outputVarName = room_volume
   subroutine real parameter: cavitysymmetrypoint_x = 0.0
   subroutine real parameter: cavitysymmetrypoint_y = 0.0 #0.5
   {if (horizon=="lower")}
   subroutine real parameter: cavitysymmetrypoint_z = -4.41
   {elseif (horizon=="upper")}
   subroutine real parameter: cavitysymmetrypoint_z = -1.98
   {endif}
   subroutine real parameter: volumeScale = {274.32/element_thickness}
```

```
    # volumeScale is 1.5 (correction factor)*2 (for W symmetry)*91.44 (L) * (1/element_thickness) (converts volume per element thickness to
    ↪  volume per unit thickness)
    compute at every step
 end

 begin user variable cstrn_off
    type = global real length = 1
    global operator = sum
    initial value = 0.0
 end user variable

 {if (iswaste=="waste")}
 begin user output
   block = b_homogenized_can_block
   compute global waste_sum as sum of element volume
 end

 begin user output
   compute global waste_volume from expression "waste_sum*2*87.85/{element_thickness}"
   # waste_volume = waste_sum * 2 (for symmetry) * 87.85 (for length of waste in room) * (1/element_thickness) (converts volume per element
   ↪  thickness to volume per unit thickness)
 end
 {endif}

{if (time_integration=="explicit")}
 begin user output
    block = b_halite b_arg_halite b_arg_halite_a b_arg_halite_b b_arg_halite_c
    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(1) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(2) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(3) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(4) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(5) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(6) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(7) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(8) 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
 {endif}

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

 begin results output output_1
    database name = efiles/%B-r.e
    database type = exodusII
    {if (time_integration=="implicit")}
    At time 0, increment = {86400.0 * 365.25 * 10.0}
    At time {86400.0 * 365.25 * 2000.0}, increment = {86400.0 * 365.25 * 500.0}
    global time
    nodal variables = residual
    {else}
    #Output data every 10 years of material time from 0 to 1000 yrs
    ${yr = -10}
    {Loop(101)}
    ${yr += 10} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 50 years of material time from 1000 to 2000 yrs
    {Loop(20)}
    ${yr += 50} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 100 years of material time from 2000 to 5000 yrs
    {Loop(30)}
    ${yr += 100} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 200 years of material time from 5000 to 10000 yrs
    {Loop(25)}
    ${yr += 200} yr
    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
    {endif}
    global variables = room_volume
    nodal variables = displacement as disp
    nodal variables = contact_accumulated_slip
    nodal variables = contact_friction_index as cfi
    nodal variables = force_external as f_ext
    nodal variables = force_contact as f_cont
    nodal variables = contact_strength as cstr
```

```
   nodal variables = contact_status as cstat
   nodal variables = contact_normal_force_magnitude as cnfor
   nodal variables = pgas_room
   {if (clay_pressure=="yes")}
   nodal variables = pgas_clay
   {endif}
   element variables = mean_stress
   element variables = eq_stress
   element variables = eq_tr_strain
   element variables = eq_vp_strain
   element variables = log_strain
   element variables = stress

   element variables = log_strain_invariant_1
   element variables = effective_log_strain
   element variables = green_lagrange_strain_rate
   element variables = damage
   element variables = emax
   element variables = epvol
   element variables = eqps
   element variables = fa
   element variables = fb
   element variables = phi
   element variables = pwork
   element variables = von_mises
   element variables = cauchy_stress
end results output output_1

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

begin history output output_3
   database name = history/%B-h.e
   database type = exodusII
   {if (time_integration=="implicit")}
   at time 0.0 increment = 8.64 #s
   at time 8.64 increment = 86.4 #s
   at time 86.4 increment = 864.0 #s
   at time 864 increment = 8640 #s
   at time 8640 increment = 86400 #s
   at time 86400 increment = 864000 #s
   at time 864000 increment = 8640000 #s
   at time {86400 * 365.25} increment = {86400 * 365.25} #s
   at time {86400 * 365.25 * 10.0} increment = {86400 * 365.25 * 10.0} #s
   at time {86400 * 365.25 * 1000.0} increment = {86400 * 365.25 * 50.0} #s
   global time
   {else}
   #Output data every 0.0864 sec of material time from 0 to 0.864 sec
   ${sec = -0.0864}
   {Loop(11)}
   ${sec += 0.0864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 0.864 sec of material time from 0.864 to 8.64 sec
   {Loop(9)}
   ${sec += 0.864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 8.64 sec of material time from 8.64 to 86.4 sec
   {Loop(9)}
   ${sec += 8.64} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 86.4 sec of material time from 86.4 to 864 sec
   {Loop(9)}
   ${sec += 86.4} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 864 sec of material time from 864 to 8640 sec
   {Loop(9)}
   ${sec += 864.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 8640 sec of material time from 8640 to 86400 sec
   {Loop(9)}
   ${sec += 8640.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every day = 86400 sec of material time from 1 day to 10 days
   {Loop(9)}
   ${sec += 86400.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 10 days of material time from 10 days to 1 yr
   {Loop(34)}
   ${sec += 864000.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data at 1 year
   ${sec = 86400.0*365.25} sec
   ${yr  = 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   #Output data every 1 yrs = 86400.0*365.25 sec of material time from 1 yr to 10 yrs
```

```
   {Loop(9)}
   ${sec += 86400.0*365.25} sec
   ${yr  += 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 10 yrs = 10*86400.0*365.25 sec of material time from 10 yr to 1000 yrs
   {Loop(99)}
   ${sec += 10*86400.0*365.25} sec
   ${yr  += 10} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 50 years of material time from 1000 to 2000 yrs
   {Loop(20)}
   ${yr += 50} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}
   #Output data every 100 years of material time from 2000 to 5000 yrs
   {Loop(30)}
   ${yr += 100} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}
   #Output data every 200 years of material time from 5000 to 10000 yrs
   {Loop(25)}
   ${yr += 200} yr
   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
   global timestep as tstep
   {endif}
   global wall_clock_time as run_time
   global vp_rate_scale_factor
   global roof_disp
   global floor_disp
   global right_wall_disp
   global room_volume
   global hourglass_energy
   {if (iswaste=="waste")}
   global waste_volume
   {endif}
   {if (horizon=="lower")}
   face pressure nearest location 0.19 0.0 -2.43 as roof_pressure
   {elseif (horizon=="upper")}
   face pressure nearest location 0.1934615 0.0 0.0 as roof_pressure
   {endif}
end history output output_3

#--------- Heartbeat Output ---------

begin heartbeat output heartbeat_presto
   stream name = %B.hb
   append = off
   precision = 10
   legend = on
   {if (time_integration=="implicit")}
   at time 0.0 increment = 8.64 #s
   at time 8.64 increment = 86.4 #s
   at time 86.4 increment = 864.0 #s
   at time 864 increment = 8640 #s
   at time 8640 increment = 86400 #s
   at time 86400 increment = 864000 #s
   at time 864000 increment = 8640000 #s
   at time {86400 * 365.25} increment = {86400 * 365.25} #s
   at time {86400 * 365.25 * 10.0} increment = {86400 * 365.25 * 10.0} #s
   at time {86400 * 365.25 * 1000.0} increment = {86400 * 365.25 * 50.0} #s
   {else}
   #Output data every 0.0864 sec of material time from 0 to 0.864 sec
   ${sec = -0.0864}
   {Loop(11)}
   ${sec += 0.0864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 0.864 sec of material time from 0.864 to 8.64 sec
   {Loop(9)}
   ${sec += 0.864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 8.64 sec of material time from 8.64 to 86.4 sec
   {Loop(9)}
   ${sec += 8.64} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 86.4 sec of material time from 86.4 to 864 sec
   {Loop(9)}
   ${sec += 86.4} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 864 sec of material time from 864 to 8640 sec
```

```
         {Loop(9)}
         ${sec += 864.0} sec
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data every 8640 sec of material time from 8640 to 86400 sec
         {Loop(9)}
         ${sec += 8640.0} sec
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data every day = 86400 sec of material time from 1 day to 10 days
         {Loop(9)}
         ${sec += 86400.0} sec
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data every 10 days of material time from 10 days to 1 yr
         {Loop(34)}
         ${sec += 864000.0} sec
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data at 1 year
         ${sec = 86400.0*365.25} sec
         ${yr  = 1} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}

         #Output data every 1 yrs = 86400.0*365.25 sec of material time from 1 yr to 10 yrs
         {Loop(9)}
         ${sec += 86400.0*365.25} sec
         ${yr  += 1} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data every 10 yrs = 10*86400.0*365.25 sec of material time from 10 yr to 1000 yrs
         {Loop(99)}
         ${sec += 10*86400.0*365.25} sec
         ${yr  += 10} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
         {EndLoop}

         #Output data every 50 years of material time from 1000 to 2000 yrs
         {Loop(20)}
         ${yr += 50} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
         {EndLoop}

         #Output data every 100 years of material time from 2000 to 5000 yrs
         {Loop(30)}
         ${yr += 100} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
         {EndLoop}

         #Output data every 200 years of material time from 5000 to 10000 yrs
         {Loop(25)}
         ${yr += 200} yr
         additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
         {EndLoop}
         {endif}
         global time as mom_bal_time
         global mat_time
         global timestep as tstep
         global wall_clock_time as run_time
         global vp_rate_scale_factor
         global room_volume
         {if (iswaste=="waste")}
         global waste_volume
         {endif}
         {if (horizon=="lower")}
         face pressure nearest location 0.19 0.0 -2.43 as roof_pressure
         {elseif (horizon=="upper")}
         face pressure nearest location 0.1934615 0.0 0.0 as roof_pressure
         {endif}
         global roof_disp
         global floor_disp
         global right_wall_disp
         labels = off
         timestamp format = ""
      end heartbeat output heartbeat_presto

      #--------- Restart ---------

      begin restart data restart_1
         database name = restart/%B.rsout
         overwrite = True
         at wall time 0.0h increment = 11.95h
      end restart data restart_1

{if (time_integration=="implicit")}

     includefile include/mySolver.in
```

```
      end adagio region SmRegion.1

   end adagio procedure The_Procedure.1

   includefile include/equationSolvers.in

   {else}
      end presto region SmRegion.1

   end presto procedure The_Procedure.1
   {endif}

end sierra WIPP Isothermal Disposal Room
```

C.1.0.1. salt-md_viscoplastic-stone_1997.txt

```
begin property specification for material halite

    density = 2300.0

    begin parameters for model md_viscoplastic

        #Intact salt parameters
        #(These parameters correspond to the Legacy Clean Salt calibration in
        #SAND2018-12601 and Reedlunn et al. (2022).)
        bulk modulus  = 20.7e9      # Pa  compute bulk modulus from E=31e+09 Pa and nu = 0.25
        shear modulus = 12.4e9      # Pa
        a1            = 8.386e22    # 1/sec
        q1/R          = 12582       # K = 25000 cal / mol / 1.9872035 cal / (mol K)
        n1            = 5.5
        b1            = 6.086e6     # 1/sec
        a2            = 9.672e12    # 4.415e16 # 1/sec
        q2/R          = 5.0338e3    # 5123 # K
        n2            = 5.0         # 6.279
        b2            = 3.034e-2    # 1/sec
        sigma_g       = 20570000.0  # Pa
        q             = 5335.0
        m1            = 3.0
        k1            = 627500.0
        c1            = 9.1967e-3   # 1/K
        alpha_h       = -17.37
        beta_h        = -7.738
        alpha_r       = 0.58
        a             = 1000        # Close approximation of Tresca flow potential
        alpha         = 45.0e-6
    end parameters for model md_viscoplastic

end property specification for material halite

begin property specification for material arg_halite

    density = 2300.0

    begin parameters for model md_viscoplastic

        #Intact salt parameters
        #(These parameters correspond to the Legacy Argillaceous Salt calibration in
        #SAND2018-12601 and Reedlunn et al. (2022).)
        bulk modulus  = 20.7e9      # Pa  compute bulk modulus from E=31e+09 Pa and nu = 0.25
        shear modulus = 12.4e9      # Pa
        a1            = 1.407e23    # 1/sec
        q1/R          = 12582       # K = 25000 cal / mol / 1.9872035 cal / (mol K)
        n1            = 5.5
        b1            = 8998000.0   # 1/sec
        a2            = 1.314e13    # 1/sec
        q2/R          = 5.0328e3    # K
        n2            = 5.0
        b2            = 0.04289     # 1/sec
        sigma_g       = 20570000.0  # Pa
        q             = 5335.0
        m1            = 3.0
        k1            = 2470000.0
        c1            = 9.1967e-3   # 1/K
        alpha_h       = -14.96
        beta_h        = -7.738      # -0.6838
        alpha_r       = 0.58
        a             = 1000        # Close approximation of Tresca flow potential
        alpha         = 45.0e-6
    end parameters for model md_viscoplastic

end property specification for material arg_halite
```

C.1.0.2. anhydrite-stone_1997.txt

```
begin property specification for material anhydrite
    density = 2300.0 # kg / m^3
    #Sierra/SM needs to have elastic parameters defined with these names, so we repeat
    #the values in the "begin parameters ..." block.
    bulk modulus = 83.4444444e9 # Pa = bulk modulus
    shear modulus = 27814814814 # Pa = shear modulus

    #These Drucker-Prager parameters came from SAND97-0795, Table 6, except it does not
    #mention a dilation angle.  They almost certainly used the soil and foam model, which
    #has a dilatation angle of 0 degrees.

    begin parameters for model soil_foam
        #two mu           =    5.5630e10  # Pa
        #bulk modulus     =    8.3444e10  # Pa
        A0               =    2.338e6    # Pa
        A1               =    2.338
        A2               =    0.0
        pressure cutoff  =    0.0 $ Pa
        pressure function = anhydrite_pressure
    end parameters for model soil_foam

end property specification for material anhydrite

    begin definition for function anhydrite_pressure
    type is piecewise linear
    ordinate is volumetricc_strain # unitless
    abscissa is pressure # Pa
    begin values
        0      0
        1.0   8.3e10
    end values
    end definition for function anhydrite_pressure
```

C.1.0.3. soil_and_foam.txt

```
begin definition for function homogenized_can_block_pressure_volstrain_function
    #These parameters came from SAND97-0796, Table 8
    type is piecewise linear
    ordinate is volumetric_strain # unitless
    abscissa is pressure # Pa
    begin values
        0       0
        0.510   1.53e6
        0.631   2.03e6
        0.719   2.53e6
        0.786   3.03e6
        0.838   3.53e6
        0.881   4.03e6
        0.942   4.93e6
        1.14    12.0e6
    end values
end definition for function homogenized_can_block_pressure_volstrain_function

begin property specification for material homogenized_can_block

    density = 559.5 #kg / m^3, see SAND97-0796, Table 5
    #These parameters came from SAND97-0796, Table 7

    #The yield function for this model is
    #phi = sqrt(3*J_2) - (a0 + a1 * p + a2 * p^2)
    #This model has a Drucker-Prager yield surface if a2 = 0
    #The Drucker-Prager yield function is often written as
    #phi = sqrt(J_2) + a * I_1 - C
    #Thus, we can write a0 and a1 in terms of a and C

    begin parameters for model soil_foam
        bulk modulus      = 222.0e6 # Pa, this value matches SAND97-0796, Table 7, but it should be bulk modulus = 333.0e6 according to Weatherby et
        ↪  al. (1991)
        two mu            = 333.0e6 # Pa, this value matches SAND97-0796, Table 7, but it should be mu = 222.0e6 according to Weatherby et al. (1991)
        a0                = 1.0e6 # Pa
        a1                = 3.0 # unitless
        a2                = 0.0
        pressure cutoff   = 0.0 # Pa
        pressure function = homogenized_can_block_pressure_volstrain_function
    end parameters for model soil_foam

    begin parameters for model elastic
        youngs modulus = 0.1e9 # Pa
        poissons ratio = 0.33
    end parameters for model elastic


end property specification for material homogenized_can_block
```

C.2. New Geomechanical Model

```
begin sierra WIPP Isothermal Disposal Room

   title Simulation of WIPP Isothermal Disposal Room

   restart = automatic

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

   #-----------------------------------
   #---------- Room geometry  ---------
   #-----------------------------------

   #Define mesh name
   ${mesh_name = "disposal_room_lower_waste_base.g"}

   #Define the disposal room horizon ('lower' or 'upper')
   ${horizon = "lower"}

   #Define if disposal room empty or with waste ('empty' or 'waste')
   ${iswaste = "waste"}

   #Define element thickness (m)
   ${element_thickness = 1.0}

   #-----------------------------------
   #---------- Waste model  -----------
   #-----------------------------------

   #Define waste material model ('soil_foam', 'foam_damage', or 'elastic')
   ${waste_model = "foam_damage"}

   #Define foam damage model ('model_F2', 'model_F3', or 'model_F4')
   ${foam_model = "model_F3"}

   #-----------------------------------
   #---------- Gas generation  --------
   #-----------------------------------

   #Set the gas generation curve ('stone1997' or 'king2021' or 'phistory')
   # Stone: SAND97-0795
   # King: ERMS-575793
   # phistory: predefined pressure history to apply to room surfaces

   ${gasCurve = "king2021"}

   #Specify the gas generation factor if computing a porosity response surface
   #Set to 1.0 when using a predefined pressure history
   ${gasScale = GASVALUE}

   #Define if gas generation pressure masking active ('mask' or 'nomask')
   ${masking = "nomask"}

   #Define if gas generation pressure applied to clay seams ('yes' or 'no')
   ${clay_pressure = "yes"}

   #-----------------------------------
   #---------- Contact options  -------
   #-----------------------------------

   #Define friction coefficients
   ${clay_seam_friction  = 0.2}
   ${clay_seam_frictionF = 0.42}
   ${salt_salt_friction  = 0.5}
   ${salt_waste_friction = 0.4}

   #Define if individual clay seams are tied ('yes' or 'no')
   ${tied_A = "no"}
   ${tied_B = "no"}
   ${tied_D = "no"}
   ${tied_E = "no"}
   ${tied_F = "no"}
   ${tied_G = "no"}
   ${tied_H = "no"}
   ${tied_I = "no"}
   ${tied_J = "no"}
   ${tied_K = "no"}
   ${tied_L = "no"}
   ${tied_M = "no"}

   #-----------------------------------
   #--- Simulation time and scaling ---
   #-----------------------------------

   #Define simulation time period (yrs)
   ${time_period = 10000}

   #Define time integration method ('implicit', or 'explicit')
   ${time_integration = "explicit"}
```

```
{if (gasCurve=="stone1997")}
  $yr1 = {yr1 = 550.0}  end of first gas generation period
  $yr2 = {yr2 = 1050.0} end of second gas generation period
{elseif (gasCurve=="king2021")}
  $yr1 = {yr1 = 1200.0} end of first gas generation period
  $yr2 = {yr2 = 2700.0} end of second gas generation period
{endif}

{if (time_integration=="implicit")}

#Define the time periods
${t_0_0 = -0.01}     #s
${t_0_1 = 0.0}       #s
${t_1_0 = t_0_1}     #s
${t_1_1 = 86400.0 * 365.25 * time_period} #s

#Define the gas generation periods
${t1 = yr1 * 86400.0 * 365.25}
${t2 = yr2 * 86400.0 * 365.25}

{else}

#Define the time periods in material time (sec)
${t_m_0_0 = -0.5}    #s
${t_m_0_1 = 0.0}     #s
${t_m_1_0 = t_m_0_1} #s
${t_m_1_1 = 86400.0 * 365.25 * time_period} #s

#Define the scale factors
${s_1_0 = 1.0e-2}
${s_ramp = 2e9}

#Define the time periods in momentum balance time (sec)
$t_mb_m_0_0  = {t_mb_m_0_0 = -0.01} #s
$t_mb_m_0_1  = {t_mb_m_0_1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (t_m_0_1))) / s_ramp} #s
$t_mb_m_1_0  = {t_mb_m_1_0 = t_m_0_1}
$t_mb_m_1_1  = {t_mb_m_1_1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (t_m_1_1))) / s_ramp} #s

#Define the gas generation periods in momentum balance time (sec)
$ t1 = {t1 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr1 * 86400.0 * 365.25))) / s_ramp }
$ t2 = {t2 = (-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr2 * 86400.0 * 365.25))) / s_ramp }

{endif}

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

define direction y with vector 0.0 1.0 0.0
define direction x with vector 1.0 0.0 0.0
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 definition for function function_constant
   type is piecewise linear
   begin values
     -1.0e16               1.0
      1.0e16               1.0
   end values
end definition for function function_constant

begin function gas_pressure
  {if (gasCurve=="phistory")}
      type is piecewise linear
  {else}
      type is piecewise analytic
  {endif}
  expression variable: rv        = global room_volume
  {if (time_integration=="implicit")}
  expression variable:  t        = global time
  {else}
  expression variable:  t        = global mat_time
  {endif}
  expression variable: fricModel = nodal contact_friction_index
  {if (masking=="mask")}
    expression variable: cstrn   = nodal contact_strength
  {else}
    expression variable: cstrn   = global cstrn_off
  {endif}
  {if (gasCurve=="stone1997")}
    begin expressions
     -1.0  "0.0"
      0.0  "fricModel==5 ? (1.0 - min(1.0,cstrn))*13608.0*(t/3.15576e7)*2494.35/(rv-302.0) : 13608.0*(t/3.15576e7)*2494.35/(rv-302.0)"
      {t1} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(3742200.0 + 6804.0*(t/3.15576e7))*2494.35/(rv-302.0) : (3742200.0 +
     ↪  6804.0*(t/3.15576e7))*2494.35/(rv-302.0)"
      {t2} "fricModel==5 ? (1.0 - min(1.0,cstrn))*10886400.0*2494.35/(rv-302.0) : 10886400.0*2494.35/(rv-302.0)"
    end
  {elseif (gasCurve=="king2021")}
    begin expressions
     -1.0  "0.0"
```

```
      0.0  "fricModel==5 ? (1.0 - min(1.0,cstrn))*9846.38*(t/3.15576e7)*2494.35/(rv-302.0) : 9846.38*(t/3.15576e7)*2494.35/(rv-302.0)"
      {t1} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(9396876.0 + 2015.65*(t/3.15576e7))*2494.35/(rv-302.0) : (9396876.0 +
     ↪  2015.65*(t/3.15576e7))*2494.35/(rv-302.0)"
      {t2} "fricModel==5 ? (1.0 - min(1.0,cstrn))*(13244376.0 + 590.65*(t/3.15576e7))*2494.35/(rv-302.0) : (13244376.0 +
     ↪  590.65*(t/3.15576e7))*2494.35/(rv-302.0)"
    end
   {elseif (gasCurve=="phistory")}
     begin values
       includefile PCURVE
     end
  {endif}
end function gas_pressure

begin function gas_pressure_claySeams
  type is piecewise analytic
  expression variable: rv = global room_volume
  {if (time_integration=="implicit")}
  expression variable:  t        = global time
  {else}
  expression variable:  t        = global mat_time
  {endif}
  expression variable: cstrn = nodal contact_strength
  {if (gasCurve=="stone1997")}
    begin expressions
       -1.0 "0.0"
       0.0  "(1.0 - min(1.0,cstrn))*13608.0*(t/3.15576e7)*2494.35/(rv-302.0)"
       {t1} "(1.0 - min(1.0,cstrn))*(3742200.0 + 6804.0*(t/3.15576e7))*2494.35/(rv-302.0)"
       {t2} "(1.0 - min(1.0,cstrn))*10886400.0*2494.35/(rv-302.0)"
    end
  {elseif (gasCurve=="king2021")}
    begin expressions
       -1.0 "0.0"
       0.0  "(1.0 - min(1.0,cstrn))*9846.38*(t/3.15576e7)*2494.35/(rv-302.0)"
       {t1} "(1.0 - min(1.0,cstrn))*(9396876.0 + 2015.65*(t/3.15576e7))*2494.35/(rv-302.0)"
       {t2} "(1.0 - min(1.0,cstrn))*(13244376.0 + 590.65*(t/3.15576e7))*2494.35/(rv-302.0)"
    end
  {endif}
end function gas_pressure_claySeams

 begin definition for 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
    {if (time_integration=="implicit")}
    evaluate expression is "2300.0 / 2.0 * (1.0 - cos(pi * time / {t_0_1 - t_0_0}))"
    {else}
    evaluate expression is "2300.0 / 2.0 * (1.0 - cos(pi * time / {t_m_0_1 - t_m_0_0}))"
    {endif}
 end definition for function fluid_density_ramp

 {if (time_integration=="explicit")}
 #---------- Viscoplastic Rate Scaling Functions ---------

 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
    type is piecewise analytic
    begin expressions
       -0.5              "{s_1_0}"
       1e-10             "{s_1_0} + {s_ramp} * (time - 0)"
    end expressions
 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
 {endif}

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

 includefile include/salt-md_viscoplastic-cal_3b.txt
```

```
includefile include/anhydrite-cal_4.txt

includefile include/polyhalite-cal_1.txt

{if (waste_model=="foam_damage")}
  {if (foam_model=="model_F2")}
    includefile include/foamDamageModel_F2.txt
  {elseif (foam_model=="model_F3")}
    includefile include/foamDamageModel_F3.txt
  {elseif (foam_model=="model_F4")}
    includefile include/foamDamageModel_F4.txt
  {endif}
{else}
  includefile include/soil_and_foam.txt
{endif}

begin solid section solid_1
   formulation = selective_deviatoric
   strain incrementation = strongly_objective
end solid section solid_1

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

begin finite element model room
   Database name = {mesh_name}
   Database type = exodusII

   begin parameters for block b_halite
      material halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_halite

   begin parameters for block b_arg_halite
      material halite
      solid mechanics use model md_viscoplastic
      section = solid_1
   end parameters for block b_arg_halite

   begin parameters for block b_anhydrite
      material anhydrite
      solid mechanics use model kayenta
      section = solid_1
   end parameters for block b_anhydrite

   begin parameters for block b_polyhalite
      material polyhalite
      solid mechanics use model soil_foam
      section = solid_1
   end parameters for block b_polyhalite

   {if (iswaste=="waste")}
   begin parameters for block b_homogenized_can_block
     material homogenized_can_block
     solid mechanics use model {waste_model}
     section = solid_1
   end parameters for block b_homogenized_can_block
   {endif}

end finite element model room

   {if (time_integration=="implicit")}

   begin adagio procedure The_Procedure.1

   begin time control

       begin time stepping block p0
         start time = {t_0_0}
         begin parameters for adagio region SmRegion.1
           time increment = {(t_0_1 - t_0_0) / 100.0}
         end
       end

       begin time stepping block p1
         start time = {t_1_0}
         begin parameters for adagio region SmRegion.1
           time increment = 1.0e-3
         end
       end

       termination time = {t_1_1}

   {else}

   begin presto procedure The_Procedure.1

   begin time control

      begin time stepping block p0
         start time = {t_mb_m_0_0}
```

```
      begin parameters for presto region SmRegion.1
         time step scale factor = 1
         time step increase factor = 5
      end
   end

   begin time stepping block p1
      start time = 0
      begin parameters for presto region SmRegion.1
         time step scale factor = 1
         time step increase factor = 5
      end
   end

   termination time = {t_mb_m_1_1}
   {endif}

end time control

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

{if (time_integration=="implicit")}
begin adagio region SmRegion.1

   begin adaptive time stepping time
       method = material during p1
       cutback factor = 0.5
       growth factor  = 1.1
       minimum multiplier = 1.e0 during p0
       minimum multiplier = 1.e-3 during p1
       maximum multiplier = 1e14
       maximum failure cutbacks = 50
   end adaptive time stepping time

{else}
begin presto region SmRegion.1
{endif}

   use finite element model room

   #--------- 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
      function = function_constant
   end gravity

   begin prescribed temperature
      block = b_halite b_arg_halite b_polyhalite b_anhydrite
      function = function_constant
      scale factor = 300.0
   end prescribed temperature

   {if (iswaste=="waste")}
   begin prescribed temperature
      block = b_homogenized_can_block
      {if (waste_model=="foam_damage" && time_integration=="explicit")}
        function = vp_rate_scale_factor_function
        scale factor = 1.0
      {else}
        function = function_constant
        scale factor = 300.0
     {endif}
   end prescribed temperature
   {endif}

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

   begin pressure
     surface = SS_room_surf
     function = gas_pressure
     scale factor = {gasScale}
     external force contribution output name = pgas_room

   end pressure

   {if (clay_pressure=="yes")}
   begin pressure
     {if (horizon=="lower")}
     surface = SS_clay_F_master SS_clay_F_slave
     {elseif (horizon=="upper")}
     surface = SS_clay_F_master SS_clay_F_slave SS_clay_G_master SS_clay_G_slave
     {endif}
     function = gas_pressure_claySeams
     scale factor = {gasScale}
```

```
  external force contribution output name = pgas_clay

end pressure
{endif}

begin fixed displacement
   node set = NS_bot_surf
   components = z
end fixed displacement

begin fixed displacement
   node set = NS_left_surf
   components = x
end fixed displacement

begin fixed displacement
   node set = NS_right_surf
   components = x
end fixed displacement

begin fixed displacement
   node set = NS_back_surf
   components = y
end fixed displacement

begin fixed displacement
   node set = NS_front_surf
   components = y
end fixed displacement

{if (iswaste=="waste")}
begin fixed displacement
   node set = NS_homogenized_can_block_left_surf
   components = x
end fixed displacement

begin fixed displacement
   node set = NS_homogenized_can_block_back_surf
   components = y
end fixed displacement

begin fixed displacement
   node set = NS_homogenized_can_block_front_surf
   components = y
end fixed displacement
{endif}

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 fluid_pressure

   #      --------- CONTACT MODEL --------------

begin contact definition contact_defn

   begin enforcement options
      {if (time_integration=="explicit")}
      momentum balance iterations = 20
      {endif}
   end

   contact surface clay_A_master contains SS_clay_A_master
   contact surface clay_A_slave  contains SS_clay_A_slave
   contact surface clay_B_master contains SS_clay_B_master
   contact surface clay_B_slave  contains SS_clay_B_slave
   contact surface clay_D_master contains SS_clay_D_master
   contact surface clay_D_slave  contains SS_clay_D_slave
   contact surface clay_E_master contains SS_clay_E_master
   contact surface clay_E_slave  contains SS_clay_E_slave
   contact surface clay_F_master contains SS_clay_F_master
   contact surface clay_F_slave  contains SS_clay_F_slave
   contact surface clay_G_master contains SS_clay_G_master
   contact surface clay_G_slave  contains SS_clay_G_slave
```

```
contact surface clay_H_master contains SS_clay_H_master
contact surface clay_H_slave  contains SS_clay_H_slave
contact surface clay_I_master contains SS_clay_I_master
contact surface clay_I_slave  contains SS_clay_I_slave
contact surface clay_J_master contains SS_clay_J_master
contact surface clay_J_slave  contains SS_clay_J_slave
contact surface clay_K_master contains SS_clay_K_master
contact surface clay_K_slave  contains SS_clay_K_slave
contact surface clay_L_master contains SS_clay_L_master
contact surface clay_L_slave  contains SS_clay_L_slave
contact surface clay_M_master contains SS_clay_M_master
contact surface clay_M_slave  contains SS_clay_M_slave

begin interaction defaults
   general contact = on
   constraint formulation = face_face
end

begin constant friction model clay_seam_model
   friction coefficient = {clay_seam_friction}
end

begin constant friction model clay_seam_modelF
   friction coefficient = {clay_seam_frictionF}
end

begin interaction clay_A
   surfaces = clay_A_master clay_A_slave
   {if (tied_A=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_B
   surfaces = clay_B_master clay_B_slave
   {if (tied_B=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_D
   surfaces = clay_D_master clay_D_slave
   {if (tied_D=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_E
   surfaces = clay_E_master clay_E_slave
   {if (tied_E=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_F
   surfaces = clay_F_master clay_F_slave
   {if (tied_F=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_modelF
   {endif}
end interaction

begin interaction clay_G
   surfaces = clay_G_master clay_G_slave
   {if (tied_G=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_H
   surfaces = clay_H_master clay_H_slave
   {if (tied_H=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_I
   surfaces = clay_I_master clay_I_slave
   {if (tied_I=="yes")}
```

```
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_J
   surfaces = clay_J_master clay_J_slave
   {if (tied_J=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_K
   surfaces = clay_K_master clay_K_slave
   {if (tied_K=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_L
   surfaces = clay_L_master clay_L_slave
   {if (tied_L=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin interaction clay_M
   surfaces = clay_M_master clay_M_slave
   {if (tied_M=="yes")}
   friction model = tied
   {else}
   friction model = clay_seam_model
   {endif}
end interaction

begin constant friction model salt_to_salt_friction_model
   friction coefficient = {salt_salt_friction}
end

begin contact surface room_surf
   surface = SS_room_surf
end

begin interaction room_surf_to_room_surf
   surfaces = room_surf room_surf
   friction model = salt_to_salt_friction_model
end

begin interaction room_surf_to_clay_F_master
   surfaces = room_surf clay_F_master
   friction model = salt_to_salt_friction_model
end

begin interaction room_surf_to_clay_F_slave
   surfaces = room_surf clay_F_slave
   friction model = salt_to_salt_friction_model
end

{if (horizon=="upper")}
begin interaction room_surf_to_clay_G_master
   surfaces = room_surf clay_G_master
   friction model = salt_to_salt_friction_model
end

begin interaction room_surf_to_clay_G_slave
   surfaces = room_surf clay_G_slave
   friction model = salt_to_salt_friction_model
end
{endif}

{if (iswaste=="waste")}
begin constant friction model salt_to_waste_friction_model
   friction coefficient = {salt_waste_friction}
end

begin contact surface homogenized_can_block_surf
   surface = SS_homogenized_can_block_exposed_surf
end

begin interaction room_surf_to_waste_surf
   surfaces = room_surf homogenized_can_block_surf
   friction model = salt_to_waste_friction_model
end
{endif}
```

```
   begin dash options
      subdivision level = 3
   end

end contact definition contact_defn

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

begin initial condition
   block = b_halite b_arg_halite b_polyhalite b_anhydrite
   #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 initial condition

{if (time_integration=="explicit")}
#--------- Viscoplastic Rate Scaling ---------

# See note above stored_time_function.
begin user variable stored_time
   type = global real length = 1
   global operator = min
   initial value = -0.5
end

begin user variable mat_time
   type = global real length = 1
   global operator = min
   #edit the following line for restart runs!
   initial value = -0.01
end

begin user variable vp_rate_scale_factor
   type = global real length = 1
   global operator = min
   initial value = 0.02
end
{endif}

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

begin user output
   compute element el_avg_stress as average of element cauchy_stress
   compute element el_avg_mean_stress as average of element hydrostatic_stress
   compute element el_avg_eq_stress as average of element eq_stress
   compute element el_avg_eq_tr_strain as average of element eq_tr_strain
   compute element el_avg_eq_vp_strain as average of element eq_vp_strain
   compute element el_avg_log_strain as average of element unrotated_log_strain
end user output

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

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

begin user output
   node set = NS_right_wall_ctr
   compute global right_wall_disp as average of nodal displacement
end

begin user variable room_volume
   type = global real length = 1
   global operator = sum
   initial value = 3.642750144 0e+03   # W (10.06) * L (91.44) * H (3.96)
end user variable

begin user output
   {if (horizon=="lower")}
   surface = SS_room_surf SS_clay_F_master SS_clay_F_slave
   {elseif (horizon=="upper")}
   surface = SS_room_surf SS_clay_F_master SS_clay_F_slave SS_clay_G_master SS_clay_G_slave
   {endif}
```

```
    surface subroutine = aupst_cavity_volume
    subroutine string parameter: outputVarName = room_volume
    subroutine real parameter: cavitysymmetrypoint_x = 0.0
    subroutine real parameter: cavitysymmetrypoint_y = 0.0 #0.5
    {if (horizon=="lower")}
    subroutine real parameter: cavitysymmetrypoint_z = -4.41
    {elseif (horizon=="upper")}
    subroutine real parameter: cavitysymmetrypoint_z = -1.98
    {endif}
    subroutine real parameter: volumeScale = {274.32/element_thickness}
    # volumeScale is 1.5 (correction factor)*2 (for W symmetry)*91.44 (L) * (1/element_thickness) (converts volume per element thickness to
    ↪  volume per unit thickness)
    compute at every step
 end

 begin user variable cstrn_off
    type = global real length = 1
    global operator = sum
    initial value = 0.0
 end user variable

 {if (iswaste=="waste")}
 begin user output
   block = b_homogenized_can_block
   compute global waste_sum as sum of element volume
 end

 begin user output
   compute global waste_volume from expression "waste_sum*2*87.85/{element_thickness}"
   # waste_volume = waste_sum * 2 (for symmetry) * 87.85 (for length of waste in room) * (1/element_thickness) (converts volume per element
  ↪  thickness to volume per unit thickness)
 end
 {endif}

{if (time_integration=="explicit")}
 begin user output
    block = b_halite b_arg_halite
    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(1) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(2) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(3) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(4) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(5) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(6) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(7) as function vp_rate_scale_factor_broadcast_function
    compute element vp_rate_scale_factor(8) 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
 {endif}

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

 begin results output output_1
    database name = efiles/%B-r.e
    database type = exodusII
    {if (time_integration=="implicit")}
    At time 0, increment = {86400.0 * 365.25 * 10.0}
    At time {86400.0 * 365.25 * 2000.0}, increment = {86400.0 * 365.25 * 500.0}
    global time
    nodal variables = residual
    {else}
    #Output data every 10 years of material time from 0 to 1000 yrs
    ${yr = -10}
    {Loop(101)}
    ${yr += 10} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 50 years of material time from 1000 to 2000 yrs
    {Loop(20)}
    ${yr += 50} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 100 years of material time from 2000 to 5000 yrs
    {Loop(30)}
    ${yr += 100} yr
    additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
    {EndLoop}
    #Output data every 200 years of material time from 5000 to 10000 yrs
    {Loop(25)}
    ${yr += 200} yr
    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
   {endif}
   global variables = room_volume
   nodal variables = displacement as disp
   nodal variables = contact_accumulated_slip
   nodal variables = contact_friction_index as cfi
   nodal variables = force_external as f_ext
   nodal variables = force_contact as f_cont
   nodal variables = contact_strength as cstr
   nodal variables = contact_status as cstat
   nodal variables = contact_normal_force_magnitude as cnfor
   nodal variables = pgas_room
   {if (clay_pressure=="yes")}
   nodal variables = pgas_clay
   {endif}
   element variables = el_avg_stress
   element variables = el_avg_mean_stress
   element variables = el_avg_eq_stress
   element variables = el_avg_eq_tr_strain
   element variables = el_avg_eq_vp_strain
   element variables = el_avg_log_strain
   element variables = log_strain_invariant_1
   element variables = effective_log_strain
   element variables = green_lagrange_strain_rate
   element variables = damage
   element variables = emax
   element variables = epvol
   element variables = eqps
   element variables = fa
   element variables = fb
   element variables = phi
   element variables = pwork
   element variables = von_mises
   element variables = cauchy_stress
end results output output_1

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

begin history output output_3
   database name = history/%B-h.e
   database type = exodusII
   {if (time_integration=="implicit")}
   at time 0.0 increment = 8.64 #s
   at time 8.64 increment = 86.4 #s
   at time 86.4 increment = 864.0 #s
   at time 864 increment = 8640 #s
   at time 8640 increment = 86400 #s
   at time 86400 increment = 864000 #s
   at time 864000 increment = 8640000 #s
   at time {86400 * 365.25} increment = {86400 * 365.25} #s
   at time {86400 * 365.25 * 10.0} increment = {86400 * 365.25 * 10.0} #s
   at time {86400 * 365.25 * 1000.0} increment = {86400 * 365.25 * 50.0} #s
   global time
   {else}
   #Output data every 0.0864 sec of material time from 0 to 0.864 sec
   ${sec = -0.0864}
   {Loop(11)}
   ${sec += 0.0864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 0.864 sec of material time from 0.864 to 8.64 sec
   {Loop(9)}
   ${sec += 0.864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 8.64 sec of material time from 8.64 to 86.4 sec
   {Loop(9)}
   ${sec += 8.64} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 86.4 sec of material time from 86.4 to 864 sec
   {Loop(9)}
   ${sec += 86.4} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 864 sec of material time from 864 to 8640 sec
   {Loop(9)}
   ${sec += 864.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 8640 sec of material time from 8640 to 86400 sec
   {Loop(9)}
   ${sec += 8640.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every day = 86400 sec of material time from 1 day to 10 days
   {Loop(9)}
   ${sec += 86400.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 10 days of material time from 10 days to 1 yr
```

```
   {Loop(34)}
   ${sec += 864000.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data at 1 year
   ${sec = 86400.0*365.25} sec
   ${yr  = 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   #Output data every 1 yrs = 86400.0*365.25 sec of material time from 1 yr to 10 yrs
   {Loop(9)}
   ${sec += 86400.0*365.25} sec
   ${yr  += 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 10 yrs = 10*86400.0*365.25 sec of material time from 10 yr to 1000 yrs
   {Loop(99)}
   ${sec += 10*86400.0*365.25} sec
   ${yr  += 10} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
   #Output data every 50 years of material time from 1000 to 2000 yrs
   {Loop(20)}
   ${yr += 50} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}
   #Output data every 100 years of material time from 2000 to 5000 yrs
   {Loop(30)}
   ${yr += 100} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}
   #Output data every 200 years of material time from 5000 to 10000 yrs
   {Loop(25)}
   ${yr += 200} yr
   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
   global timestep as tstep
   {endif}
   global wall_clock_time as run_time
   global vp_rate_scale_factor
   global roof_disp
   global floor_disp
   global right_wall_disp
   global room_volume
   {if (iswaste=="waste")}
   global waste_volume
   {endif}
   {if (horizon=="lower")}
   face pressure nearest location 0.19 0.0 -2.43 as roof_pressure
   {elseif (horizon=="upper")}
   face pressure nearest location 0.1934615 0.0 0.0 as roof_pressure
   {endif}
end history output output_3

#--------- Heartbeat Output ---------

begin heartbeat output heartbeat_presto
   stream name = %B.hb
   append = off
   precision = 10
   legend = on
   {if (time_integration=="implicit")}
   at time 0.0 increment = 8.64 #s
   at time 8.64 increment = 86.4 #s
   at time 86.4 increment = 864.0 #s
   at time 864 increment = 8640 #s
   at time 8640 increment = 86400 #s
   at time 86400 increment = 864000 #s
   at time 864000 increment = 8640000 #s
   at time {86400 * 365.25} increment = {86400 * 365.25} #s
   at time {86400 * 365.25 * 10.0} increment = {86400 * 365.25 * 10.0} #s
   at time {86400 * 365.25 * 1000.0} increment = {86400 * 365.25 * 50.0} #s
   {else}
   #Output data every 0.0864 sec of material time from 0 to 0.864 sec
   ${sec = -0.0864}
   {Loop(11)}
   ${sec += 0.0864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 0.864 sec of material time from 0.864 to 8.64 sec
   {Loop(9)}
   ${sec += 0.864} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 8.64 sec of material time from 8.64 to 86.4 sec
   {Loop(9)}
   ${sec += 8.64} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}
```

```
   #Output data every 86.4 sec of material time from 86.4 to 864 sec
   {Loop(9)}
   ${sec += 86.4} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 864 sec of material time from 864 to 8640 sec
   {Loop(9)}
   ${sec += 864.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 8640 sec of material time from 8640 to 86400 sec
   {Loop(9)}
   ${sec += 8640.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every day = 86400 sec of material time from 1 day to 10 days
   {Loop(9)}
   ${sec += 86400.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 10 days of material time from 10 days to 1 yr
   {Loop(34)}
   ${sec += 864000.0} sec
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data at 1 year
   ${sec = 86400.0*365.25} sec
   ${yr  = 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}

   #Output data every 1 yrs = 86400.0*365.25 sec of material time from 1 yr to 10 yrs
   {Loop(9)}
   ${sec += 86400.0*365.25} sec
   ${yr  += 1} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 10 yrs = 10*86400.0*365.25 sec of material time from 10 yr to 1000 yrs
   {Loop(99)}
   ${sec += 10*86400.0*365.25} sec
   ${yr  += 10} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * sec)) / s_ramp}
   {EndLoop}

   #Output data every 50 years of material time from 1000 to 2000 yrs
   {Loop(20)}
   ${yr += 50} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}

   #Output data every 100 years of material time from 2000 to 5000 yrs
   {Loop(30)}
   ${yr += 100} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}

   #Output data every 200 years of material time from 5000 to 10000 yrs
   {Loop(25)}
   ${yr += 200} yr
   additional times = {(-s_1_0 + sqrt(s_1_0^2 + 2 * s_ramp * (yr * 86400.0 * 365.25))) / s_ramp}
   {EndLoop}
   {endif}
   global time as mom_bal_time
   global mat_time
   global timestep as tstep
   global wall_clock_time as run_time
   global vp_rate_scale_factor
   global room_volume
   {if (iswaste=="waste")}
   global waste_volume
   {endif}
   {if (horizon=="lower")}
   face pressure nearest location 0.19 0.0 -2.43 as roof_pressure
   {elseif (horizon=="upper")}
   face pressure nearest location 0.1934615 0.0 0.0 as roof_pressure
   {endif}
   global roof_disp
   global floor_disp
   global right_wall_disp
   labels = off
   timestamp format = ""
end heartbeat output heartbeat_presto

#--------- Restart ---------

begin restart data restart_1
```

```
            database name = restart/%B.rsout
            overwrite = True
            at wall time 0.0h increment = 11.95h
         end restart data restart_1

   {if (time_integration=="implicit")}

        includefile include/mySolver.in

      end adagio region SmRegion.1

   end adagio procedure The_Procedure.1

   includefile include/equationSolvers.in

   {else}
      end presto region SmRegion.1

   end presto procedure The_Procedure.1
   {endif}

end sierra WIPP Isothermal Disposal Room
```

C.2.0.1. salt-md_viscoplastic-cal_3b.txt

```
begin property specification for material halite

    density = 2300.0

    begin parameters for model md_viscoplastic

        #Intact salt parameters
        #(These parameters correspond to Calibration 3B in SAND2018-12601 and
        #Reedlunn et al. (2022), except steady state mechanism 3 has been turned off
        #by modifying b1, b2, and q.)
        bulk modulus  = 20.7e9 # Pa  compute bulk modulus from E=31e+09 Pa and nu = 0.25
        shear modulus = 12.4e9  # Pa
        a0            = 5.617e1 # 1/sec
        q0/R          = 5123 # K
        n0            = 1.595
        a1            = 8.386e22 # 1/sec
        q1/R          = 12580.5 # K = 25000 cal / mol / 1.9872035 cal / (mol K)
        n1            = 5.5
        b1            = 0.0 ## 6.086e6 # 1/sec
        a2            = 4.415e16 # 1/sec
        q2/R          = 5123 # K
        n2            = 6.279
        b2            = 0.0 ## 3.034e-2 # 1/sec
        sigma_g       = 20570000.0 # Pa
        q             = 0.0 ## 5335.0
        m0            = 0.9201
        k0            = 5.277e-2
        c0            = 8.882e-3 # 1/K
        m1            = 5.282
        k1            = 3.052e12
        c1            = 8.882e-3 # 1/K
        alpha_h       = 3.367
        beta_h        = -0.6838
        alpha_r       = 0.58
        a             = 16.0
        alpha         = 45.0e-6
    end parameters for model md_viscoplastic

end property specification for material halite
```

C.2.0.2. anhydrite-cal_4.txt

```
begin property specification for material anhydrite
    density = 2300.0 # kg / m^3
    #Sierra/SM needs to have elastic parameters defined with these names, so we repeat
    #the values in the "begin parameters ..." block.
    bulk modulus = 83.4444444e9 # Pa = bulk modulus
    shear modulus = 27814814814 # Pa = shear modulus

    #These Drucker-Prager parameters came from SAND97-0795, Table 6, except it does not
    #mention a dilation angle.  They almost certainly used the soil and foam model, which
    #has a dilatation angle of 0 degrees.  Here we have switched to associated flow.
    #The Drucker-Prager yield function is often written as phi = sqrt(J_2) + a * I_1 - C
    #B0 = bulk modulus = 83.4444444e9 Pa
    #G0 = shear modulus = 27814814814 Pa
    #C = 1.35
    #a = 0.45
```

```
    #alpha = dilatation angle = 24.227745 deg = atan(0.45)

    #Reduce Kayenta to Drucker-Prager
    begin parameters for model kayenta
        B0 = 83.4444444e9   # Pa = bulk modulus
        G0 = 27814814814    # Pa = shear modulus
        J3TYPE = 1          # Sets the dependence on J_3.  J3TYPE = 1 is a Von-Mises dependence.
        A1 = 1350000        # Pa
        A2 = 0.0
        A3 = 0.0
        A4 = 0.45
        RK = 1.0
        P0 = -1.0e99        # Put the compression cap at virtually infinity
        P1 = 0.0            # No cap
        P2 = 0.0            # No cap
        P3 = 0.0            # Zero porosity
        CR = 0.001          # Minimize the size of the curved part of the cap
        HC = 0.0            # Disable kinematic hardening
        RN = 0.0            # Disable kinematic hardening
    end parameters for model kayenta
end property specification for material anhydrite
```

C.2.0.3. polyhalite-cal_1.txt

```
begin definition for function polyhalite_pressure_volstrain_function
    type is piecewise linear
    ordinate is volumetric_strain
    abscissa is pressure
    begin values
        -1  -6.583333333e+10 # -65.83333333 GPa
        0  0
        1  6.583333333e+10 # 65.83333333 GPa = bulk modulus = E / (3*(1-2*nu)), where E = 55.3 GPa and nu = 0.36
    end values
end definition for function polyhalite_pressure_volstrain_function

begin property specification for material polyhalite

    #These parameters came from SAND97-0796, Pages A-40 through A-41, Section 2.5.2

    #The yield function for this model is
    #phi = sqrt(3*J_2) - (a0 + a1 * p + a2 * p^2)
    #This model has a Drucker-Prager yield surface if a2 = 0
    #The Drucker-Prager yield function is often written as
    #phi = sqrt(J_2) + a * I_1 - C
    #Thus, we can write a0 and a1 in terms of a and C

    density = 2300.0
    begin parameters for model soil_foam
        poissons ratio  = 0.36
        youngs modulus  = 5.53e+10 # Pa
        a0              = 2459512.147 # Pa = sqrt(3) * C, where C = 1.42 MPa
        a1              = 2.457780096 # unitless = 3 * sqrt(3) * a, where a = 0.473
        a2              = 0.0
        pressure cutoff = -1000000.0 # Pa
        pressure function = polyhalite_pressure_volstrain_function
    end parameters for model soil_foam
end property specification for material polyhalite
```

C.2.0.4. foamDamageModel_F3.txt

```
   ###
   ### Material property definitions
   ###
   ##
   ##  Input block for Container
   ##  N, kg, m, seconds, C  Stress: Pa,  Temperature: C
   ##
   ##

   begin property specification for material homogenized_can_block
      density = 580.0              ## kg/m^3
      thermal engineering strain function = container_Thermal
      begin parameters for model foam_damage
         youngs modulus    = 4000.0e6   ## locked up Pa
         poissons ratio    =  0.30
         phi               =   0.175   ## volume fraction solids
         flow rate         =  1.000
         power exponent    =  1.000
         tensile strength  =  100.0e6 ## high so no fail in tension
         adam              =  1.000
         bdam              =  0.330
         youngs function   =  container_Modulus
```

```
      poissons function =  container_PR
      rate function     =  container_Rate
      exponent function =  container_Expo
      shear hardening function = container_Shear
      hydro hardening function = container_Hydro
      beta function            = container_Beta
      youngs phi function      = container_E
      poissons phi function    = container_Constant
      damage function          = container_Damage
   end parameters for model foam_damage
end property specification for material homogenized_can_block

begin definition for function container_Damage
   type is piecewise linear
   begin values
       0.00000     0.000
       0.50000     0.000
       0.60000     0.000
     100.00000     0.000
   end values
end definition for function container_Damage

## flow direction 1 = radial, 0 = associated
begin definition for function container_Beta
   type is piecewise linear
   begin values
      0.000   0.7
      0.200   0.7
      0.250   0.65
      0.30    0.55
      0.35    0.4
      0.40    0.22
      0.45    0.15
      0.50    0.075
      0.640   0.0
      1.000   0.0
      10.00   0.0
   end values
end definition for function container_Beta
##
## from unloading slopes (function values are multiplied x reference value above)
##
begin definition for function container_E
   type is piecewise linear
   begin values
       0.000    0.01
       0.250    0.015
       0.400    0.025
       0.500    0.075
       0.650    0.128
       1.000    0.500
       2.000    1.000
     100.000    1.000
   end values
end definition for function container_E

begin definition for function container_PR
   type is constant
   begin values
      1.000
   end values
end definition for function container_PR

begin definition for function container_Constant
   type is constant
   begin values
      1.0
   end values
end definition for function container_Constant

# Eliminated thermal strains and temperature dependence of the Young's
# modulus, so that temperature changes have no effect on the model, except
# to change the viscoplastic rate scaling via the container_Rate value when
# explicit time integration is being used.

begin definition for function container_Thermal
   type is piecewise linear
   ordinate is strain
   abscissa is temperature
   begin values
      -500.0    -0.0
         0.0     0.0
       500.0     0.0
   end values
end definition for function container_Thermal

begin definition for function container_Modulus
   type is piecewise linear
   ordinate is temperature
   abscissa is time
   begin values
       -1.0e5          1.0
```

```
            2.0e2          1.0
            1.3e5          1.0
         end values
      end definition for function container_Modulus

   # parameter "h" in \dot(\lambda) = exp(h) * < \sigma^* / a - 1 >
   # (see equation 9 in SAND2015-4352)
   # (Note we are using h to speed up the viscoplastic strain rate according to
   # d(\lambda) / d(t^(mb)) = s * exp(h_r) * < \sigma^* / a - 1 >,
   # where t^(mb) is the momentum balance time, t is material time,
   # s = d(t)/d(t^(mb)) is the vp_rate_scale_factor, and h_r is the real value
   # for h.  To pull s into exp(), we use s * exp(h_r) = exp(h_r + ln(s)), hence
   # the expression below.)
   # (Note that the expression below appears to use natural log of time, but the
   # foam damage model treats "t" as temperature, and we elsewhere set temperature
   # equal to the vp_rate_scale_factor "s".)
   # (Note that if one tells Sierra/SM to evaluate the container_Rate for output
   # purposes, then the output will treat the "t" as momentum balance time, not
   # temperature.  This means the output evaluation of the container_Rate function
   # and the foam damage model evalation of the container_Rate function will
   # disagree, but it does not mean something went wrong in the simulation.)
      begin definition for function container_Rate
         type is analytic
         {if (time_integration=="explicit" && s_ramp!=0.0)}
           evaluate expression = " -55.0 + ln(t) "
         {else}
           evaluate expression = " -55.0 "
         {endif}
      end definition for function container_Rate

   # parameter "n"
      begin definition for function container_Expo
         type is constant
         begin values
            23.0
        end values
      end definition for function container_Expo

      begin function container_Shear
         type is piecewise linear
         x scale  = 1.12
         begin values
            0.00      0.02e6
            0.10      0.02e6
            0.25      0.02e6
            0.30      0.02e6
            0.35      0.04e6
            0.40      0.07e6
            0.42      0.12e6
            0.44      0.19e6
            0.45      0.22e6
   #         0.46      0.04e6
   #         0.47      0.12e6
            0.50      0.36e6
            0.52      0.46e6
            0.55      0.72e6
            0.60      1.32e6
            0.62      1.74e6
            0.65      2.80e6
            0.70      4.42e6
            0.80     10.52e6
            0.90     14.62e6
            1.00     30.00e6
           10.00    500.00e6
          100.00   1600.00e6
         end values
      end function

      begin function container_Hydro
         type is piecewise linear
         x scale  = 1.12
         begin values
            0.00     0.012e6
            0.25     0.012e6
            0.30     0.036e6
            0.40     0.132e6
            0.45     0.240e6
            0.50     0.480e6
            0.60     1.440e6
            0.70    12.000e6
            0.80    45.600e6
            0.90    84.000e6
            1.00   168.000e6
           10.00   960.000e6
          100.00  9600.000e6
         end values
      end function

      ###
      ### end of input block for container material
      ###
```

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
K. Peterson	1442	kjpeter@sandia.gov
F. Beckwith	1542	fbeckwi@sandia.gov
T. Mattsson	1550	trmatts@sandia.gov
J. Petti	1554	jppetti@sandia.gov
W. Huber	1554	wjhuber@sandia.gov
S. Nelson	1558	smnelso@sandia.gov
D. O'Connor	1558	dtoconn@sandia.gov
C. Vignes	1558	cvignes@sandia.gov
K. Kuhlman	8842	klkuhlm@sandia.gov
R. Rechard	8842	rprecha@sandia.gov
R. C. Camphouse	8842	rccamph@sandia.gov
M. Mills	8842	mmmills@sandia.gov
G. Roselle	8880	gtrosel@sandia.gov
D. Kicker	8880	dckicke@sandia.gov
G. Duran	8880	gasosa@sandia.gov
R. Kirkes	8881	grkirke@sandia.gov
J. Long	8881	jjlong@sandia.gov
S. Wagner	8881	swagner@sandia.gov
B. Park	8912	bypark@sandia.gov
T. Zeitler	8912	tzeitle@sandia.gov
S. Broome	8914	stbroom@sandia.gov
C. Choens	8914	rcchoen@sandia.gov
M. Ingraham	8914	mdingr@sandia.gov
C. Hansen	8924	cwhanse@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

Email—External

Name	Company Email Address	Company Name
Kirk McDaniel	Kirk.McDaniel@cbfo.doe.gov	U.S. Department of Energy Carlsbad Field Office
Anderson Ward	anderson.ward@wipp.ws	U.S. Department of Energy Carlsbad Field Office
Michael Gross	mike_gross@earthlink.net	MG Enterprises
Gordan Gjerapic	ggjerapic@gmail.com	ZG-Geo Ltd.
Frank Hansen	francis.d.hansen@gmail.com	N/A
J. Guadalupe Arguello	llae.arguello@att.net	N/A
Courtney Herrick	cgherri.ch@gmail.com	N/A
Jibril Coulibaly	jcoulibaly@cusatisinc.com	Cusatis Computational Services

This page intentionally left blank.



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.