

GEOSCIENTIFIC MODEL DEVELOPMENT

Response to Reviewer 1

PorousLab v1.1: a modular finite element framework for simulation of fractured porous media

Danilo Cavalcanti on behalf of the co-authors

Manuscript: egusphere-2026-3322

Anonymous Referee #1 | RC1, 11 September 2026

Reviewer comments appear in shaded boxes. Author responses appear in **dark blue**; manuscript changes are listed separately. Reproduced manuscript text is indented and marked with a vertical line. Addressed comments are labelled and struck through. Comment identifiers are added for navigation.

Page and line references refer to the published preprint. The listed changes have been prepared for the revised manuscript. Equations, tables, and figures reproduced in this response are numbered with the prefix R; cross-references within the reproduced material follow this numbering.

Review source: <https://doi.org/10.5194/egusphere-2026-3322-RC1>

Summary

R1 Reviewer comment

PorousLab 1.1 is an MIT-licensed MATLAB finite element framework for two-dimensional coupled hydro-mechanical problems with single- and two-phase flow, nonlinear geomechanics, continuation methods and embedded strong discontinuities (E-FEM). The formulation is standard and, as far as I checked (balance equations, weak forms, residuals) seems correct; the code is archived with example scripts and regression tests. The contribution is in integration and accessibility rather than in new methods: the E-FEM formulations and the two-phase treatment come from published work. Two aspects require substantial revision. First, the comparison with existing frameworks is not appropriate/balanced: design choices of other codes are presented as deficiencies, in the case of PorePy I'm quite certain it is misclassified, and the dependence on a proprietary runtime is not acknowledged. Second, the verification is qualitative throughout: no error measures or refinement studies, no fracture-network case with a reference solution (these exist elsewhere, see below), no blocking fractures, and no example in which the embedded fault slips or opens (although fault mechanics is a main application). Two examples cannot be reproduced from the paper alone (missing material and fracture parameters). I recommend major revisions. I'm all for new frameworks, there is always something new to learn! The requests are about demonstrating the framework fairly and quantitatively, not the method itself.

Author response

We thank the reviewer for the detailed assessment and constructive suggestions. We have revised the Introduction and software comparison to present existing frameworks neutrally, clarify PorousLab's contribution, and state the MATLAB licence requirement explicitly. The numerical assessment now includes quantitative error measures, mesh-refinement studies, fracture sliding and opening cases, and a fracture-network benchmark with conductive and blocking configurations. We have also extended the parameter descriptions of the reservoir and dam-foundation examples. Detailed responses and the corresponding manuscript changes are provided below.

Major comments

R1.1 Reviewer comment

The introduction motivates PorousLab largely by presenting properties of other frameworks as disadvantages: hybrid discretizations (l. 60–62, 68–69, 72–73), compiled languages and performance orientation (l. 65–68). Most of these are deliberate design choices rather than deficiencies so using a negative connotation is inappropriate. Also the claim are actually not substantiated, and several statements do not hold as written.

Author response

We have rewritten the Introduction to present existing frameworks neutrally and explain PorousLab's contribution through the capabilities it integrates.

Changes in the manuscript

Section 1, paragraph following Table 1: We have replaced the paragraph beginning “As shown in Table 1” with the following discussion.

Table 1 illustrates the range of capabilities supported by existing porous-media frameworks, including broad multiphysics coverage and a variety of spatial discretization families. When a framework supports multiple spatial discretization methods, these may be used either as alternative formulations or for different governing equations within a coupled model. Alongside these modeling capabilities, several frameworks provide parallel computing facilities for large-scale applications (e.g., Settgaest et al., 2024; Hammond et al., 2014), and modular research environments support the development and prototyping of numerical models (e.g., Krogstad et al., 2015; Lie, 2019; Keilegavlen et al., 2021). The use and continued development of established projects are further supported by active developer and user communities, documentation, and shared examples (e.g., Kolditz et al., 2012; Flemisch et al., 2011).

R1.2 Reviewer comment

(i) Example of a mistake: PorePy is classified as "Hybrid" in Table 1, and l. 72–73 state that its coupled formulation with mechanics relies on more than one discretization strategy. PorePy discretizes flow and mechanics with cell-centred finite volume schemes throughout (MPFA for flow, MPSA for mechanics; see Keilegavlen et al., 2021), i.e. a single finite-volume setting. It offers RT0 mixed finite elements and a virtual element method for flow in addition, as options and this is great.

Author response

We have replaced PorePy's “Hybrid” entry with “FVM, FEM, VEM” and removed the statement that its coupled flow–mechanics formulation requires different discretization families. The accompanying text clarifies that listed methods may be alternatives rather than mandatory combinations.

Changes in the manuscript

1. **Table 1, PorePy row:** We have replaced “Hybrid” in the discretization column with the following entry.

FVM, FEM, VEM

2. **Section 1, paragraph following Table 1:** We have replaced the statement about the required combination of methods with the following explanation.

When a framework supports multiple spatial discretization methods, these may be used either as alternative formulations or for different governing equations within a coupled model.

R1.3 Reviewer comment

(ii) The "PDE scheme" column conflates the method family (FEM/FVM) with the number of schemes a framework offers. Example: in DuMux, geomechanics is handled by a combination of a vertex-centered FV (for displacements) and a cell-centred TPFA/MPFA FV scheme for flow. Both are finite-volume schemes implemented in one assembly framework, and the user selects one per sub-problem; that different schemes can be combined is a feature of these frameworks, not a limitation. The claim that hybrid schemes require "proficiency in multiple numerical methods" (l. 69) is not substantiated in this case they just need to know finite volumes and the software abstractions hide the details.

Author response

We have renamed the column "Main spatial discretization", classified DuMux as FVM, and removed the claim about proficiency in multiple numerical methods. The table now groups schemes by method family.

Changes in the manuscript

1. **Table 1, discretization-column heading:** We have replaced "PDE scheme" with the following heading.

Main spatial discretization

2. **Table 1, DuMux row:** We have replaced "Hybrid" with the following entry.

FVM

R1.4 Reviewer comment

(iii) Interestingly, in this work: Appendix A introduces node-centred cells (Fig. A1), cell-wise saturation and density, and element-wise averaged mobilities in order to obtain local mass conservation. These are clearly control-volume treatments inside the finite element method (and there is a good reason for this). But this means the manuscript's own H2 physics is not a pure Galerkin FEM either. The "single FEM discretization" argument (abstract l. 11, l. 75–76, 80–82) should be removed accordingly.

Author response

We have removed the "single FEM discretization" argument from the abstract and Introduction. The contribution is now stated in terms of integrated capabilities and a common software architecture. Appendix A describes the nodal storage treatment and element-wise coefficients used alongside the Galerkin flux discretization.

Changes in the manuscript

1. **Abstract:** We have replaced the sentences beginning "The framework stands out" and "Furthermore, the code architecture" with the following sentence.

These capabilities are integrated within a common software architecture that separates model definition, material behavior, element formulations, and solution procedures to support code reuse and extension.

2. **Section 1, contribution paragraph:** We have replaced the paragraph beginning "To address this gap" with the following paragraph.

This paper presents PorousLab, an open-source, object-oriented finite-element framework implemented in MATLAB for computational geomechanics researchers developing, verifying, and prototyping porous-media formulations. Its contribution is the integration of coupled flow and deformation, nonlinear bulk constitutive models, continuation procedures for quasi-static mechanical analysis, and embedded-discontinuity formulations within a common software architecture. The architecture separates model definition, material behavior, element formulations, and solution procedures. This organization supports the development and assessment of individual formulations while reusing model-construction, assembly, and post-processing components. The resulting scripted workflow connects formulation development with numerical verification within the same computational environment.

R1.5 Reviewer comment

I recommend removing the entire "PDE scheme" column. List neutrally what schemes are available elsewhere and rewrite l. 63–74 so that the contribution is stated positively (what PorousLab offers: general constitutive models, continuation methods, E-FEM, a scripted MATLAB workflow) rather than by discounting other codes.

Author response

We have retained the revised column to provide a concise overview of available numerical methods, with their flexibility explained in the accompanying text. The contribution paragraph now focuses on PorousLab's constitutive models, continuation procedures, embedded discontinuities, and scripted MATLAB workflow.

Changes in the manuscript

1. **Section 1, paragraph following Table 1:** We have added the following explanation of the available discretization families.

When a framework supports multiple spatial discretization methods, these may be used either as alternative formulations or for different governing equations within a coupled model.

2. **Section 1, contribution paragraph:** We have replaced the contribution statement beginning "The contribution of this work lies" with the following sentence.

Its contribution is the integration of coupled flow and deformation, nonlinear bulk constitutive models, continuation procedures for quasi-static mechanical analysis, and embedded-discontinuity formulations within a common software architecture.

R1.6 Reviewer comment

Several authors seem to be core developers of Kratos Multiphysics and of GeMA, whose physics abstraction is adopted. A paragraph on why a separate MATLAB framework was developed, and whether PorousLab is intended as a prototyping front-end for those codes, would position the contribution more convincingly than the comparison table.

Author response

PorousLab was developed to provide a MATLAB finite-element environment for developing, verifying, and prototyping porous-media formulations, as described in the Introduction. Its development was influenced by the authors' experience with Kratos Multiphysics and GeMA, and it has also been used to prototype developments for those frameworks. This prototyping role is not restricted to implementations for these codes: PorousLab is intended for the broader computational geomechanics research community.

R1.7 Reviewer comment

PorousLab is released under the MIT licence but requires a MATLAB licence, and it is stated that it is not compatible with GNU Octave. At the same time the manuscript excludes COMSOL and Abaqus from the comparison as commercial software and builds on "open-source" and "lower entry barrier" arguments (abstract, l. 74, 82–85, 405–407). The exclusion is a legitimate scoping decision, but the text should say explicitly that the framework is open-source code on a proprietary runtime, and that this touches the accessibility argument; at least in comparison with PorePy (Python, also easy to use) and MRST, which (as far as I know) actually makes some effort to run on Octave. Please also state which MATLAB features prevent Octave use (App Designer, specific classdef features, ...) and whether Octave support of the non-GUI core is planned; a table note on the "Platform" column would be appropriate.

Author response

The incompatibility with GNU Octave is not confined to the GUI. Tests with GNU Octave 11.3.0 encountered incompatibilities with `matlab.mixin.Copyable` and the abstract-method declaration syntax used in the non-GUI core. The GUI additionally depends on MATLAB App Designer. GNU Octave support for PorousLab's

non-GUI core remains a possible future development. The Introduction now distinguishes the MIT source licence from the MATLAB licence required for execution and states that Table 1 uses public source-code availability as a selection criterion. We have replaced its comparative accessibility argument with a description of PorousLab's intended research use and extended Section 2 to identify the Octave limitations.

Changes in the manuscript

1. **Section 1, paragraph introducing Table 1:** We have clarified the source-access criterion used to select the frameworks.

The comparison is restricted to frameworks with publicly available source code and at least one documented strategy for strong discontinuity modeling in the versions considered.

2. **Section 1, paragraph beginning “The current release supports”:** We have added the following sentence at the end of the paragraph to distinguish the source licence from the execution requirements.

The source code is distributed under the MIT license, and execution requires a MATLAB license, as the current release does not support GNU Octave.

3. **Section 2, final paragraph:** We have replaced the first two sentences to state the MATLAB licence requirement and identify the Octave incompatibilities. The revised paragraph is reproduced below.

The current release is distributed under the MIT licence and requires a MATLAB licence, with no additional toolbox dependencies. It has been tested on MATLAB R2022a and later versions. Tests with GNU Octave 11.3.0 identified incompatibilities with `matlab.mixin.Copyable` and the abstract-method declaration syntax used in the non-GUI core. The source code, benchmark scripts, documentation, and regression tests are available through the public repository and the archived release cited in the Code availability section. A simple graphical user interface, developed with MATLAB App Designer, is also provided for pre- and post-processing. This interface is currently available only for mechanical analyses.

R1.8 Reviewer comment

All verification cases are one-dimensional, single-fracture, or parallel-fracture problems, and the only network case (Fig. 19e–f, “random fractures”) has no reference solution. The authors state that “Figure 19 shows that the pore-pressure profiles obtained with PorousLab agree closely with the reference solutions”. Community benchmarks for exactly this setting exist, with reference results from a range of codes: Flemisch et al. (2018, Adv. Water Resour. 111, 239–258) define four two-dimensional single-phase flow cases (Hydrocoin; a regular fracture network with permeable and blocking variants; a complex network with mixed permeable and blocking fractures; a realistic outcrop-based network of 64 fractures), with geometries, meshes and reference solutions publicly available. At least one of these cases should be added. This should (a) verify the hydraulic E-FEM on intersecting fractures rather than on isolated or parallel ones, (b) exercise blocking fractures, which the abstract and Sect. 1 mention but no example tests, and (c) place PorousLab on the same comparison as some of the other codes in Table 1 already went through. The “random fractures” case should either receive a reference or be labelled as a show case rather than a verification.

Author response

We have added Benchmark 1 of Flemisch et al. (2018), including both conductive and blocking configurations, in the new subsection “Single-phase flow through a regular fracture network”, immediately after the dam-foundation example. We have also revised Section 3.7.3 to explain the pressure-jump representation and matrix–fracture fluid exchange used in this example.

Finally, we have restricted the reference-comparison statement in the dam-foundation example to the intact and parallel-fracture configurations and quantified these comparisons using the root mean squared error (RMSE), defined in Eq. (R8). The random-fracture configuration is explicitly identified as an illustration without a reference solution and is not presented as a verification test.

Changes in the manuscript

1. **Section 3.7.3, “Fluid flow inside the discontinuity”:** We have revised this subsection to specify its single-phase scope and explain the reduced hydraulic balance, longitudinal flow, pressure-jump representation, and

matrix–fracture exchange. The revised subsection is reproduced below. Equations (33)–(34) refer to the existing SDA definitions in the manuscript.

The embedded formulation for hydraulic problems is currently limited to saturated single-phase flow (Cavalcanti et al., 2024a). Integrating the hydraulic balance across the fracture aperture gives

$$\frac{\phi_d w}{\rho_l} \frac{\partial \rho_l}{\partial t} + \frac{\partial}{\partial s} (w q_{d,l}^L) + q_{d,l}^t = 0, \quad (R1)$$

where ϕ_d is the discontinuity porosity, with $\phi_d = 1$ for open fractures, w is the aperture, s is the tangential coordinate, and $q_{d,l}^t$ is the net outward flux through the fracture faces. The aperture-averaged longitudinal volumetric flux follows Darcy's law,

$$q_{d,l}^L = \frac{k_d^L}{\mu_l} \left(-\frac{\partial p_{d,l}}{\partial s} + \rho_l \mathbf{m}_d^\top \mathbf{g} \right), \quad (R2)$$

where $p_{d,l}$ is the aperture-averaged liquid pressure and k_d^L is the longitudinal permeability, defined by the cubic law

$$k_d^L = \frac{w^2}{12}. \quad (R3)$$

In hydro-mechanical analyses with evolving aperture, the value used in the cubic law is taken from the previous converged time step.

To represent pressure jumps associated with fractures of low normal permeability, the SDA defined in Eqs. (33)–(34) of the manuscript is applied to the pore-pressure field, adopting a constant jump over each embedded segment. Fluid exchange between the discontinuity and the surrounding porous matrix is described by the leak-off law (Segura and Carol, 2004)

$$q_{d,l}^{t\pm} = \lambda_d^\pm (p_{d,l} - p_l^\pm), \quad q_{d,l}^t = q_{d,l}^{t+} + q_{d,l}^{t-}, \quad (R4)$$

where $p_l^\pm$ are the matrix pressures on the fracture faces, and $\lambda_d^\pm$ are the leak-off coefficients. Each face flux is positive from the discontinuity towards the porous matrix. The total leak-off represents the net outflow from the discontinuity. At each face, fluid leaving the discontinuity enters the adjacent porous matrix, so the same flux appears with opposite signs in their balance equations.

The liquid pressure field inside the discontinuity is approximated with

$$p_{d,l}^h = \mathbf{N}_{d,p}(s) \mathbf{p}_{d,l}, \quad (R5)$$

where $\mathbf{N}_{d,p}(s)$ is a vector with the Lagrange functions in terms of the tangential coordinate of the discontinuity and $\mathbf{p}_{d,l}$ is a vector with the DOFs corresponding to the pore-pressure at the discontinuity nodes.

The weak form is obtained by applying the Galerkin method followed by the divergence theorem,

$$\begin{aligned} \mathbf{r}_{d,l}(t) = & \int_{\Gamma_d} \mathbf{N}_{d,p}^\top \phi_d \frac{w}{\rho_l} \frac{\partial \rho_l}{\partial t} d\Gamma + \int_{\Gamma_d} \frac{\partial \mathbf{N}_{d,p}^\top}{\partial s} w \frac{k_d^L}{\mu_l} \frac{\partial \mathbf{N}_{d,p}}{\partial s} d\Gamma \mathbf{p}_{d,l} \\ & - \int_{\Gamma_d} \frac{\partial \mathbf{N}_{d,p}^\top}{\partial s} w \frac{k_d^L}{\mu_l} \rho_l \mathbf{m}_d^\top \mathbf{g} d\Gamma + \int_{\Gamma_d} \mathbf{N}_{d,p}^\top q_{d,l}^t d\Gamma. \end{aligned} \quad (R6)$$

The discontinuity pressure DOFs can be condensed by prescribing the internal pressure at each segment endpoint $\mathbf{x}_a$ as

$$p_{d,l}^h(\mathbf{x}_a) = \frac{p_l^{h+}(\mathbf{x}_a) + p_l^{h-}(\mathbf{x}_a)}{2}. \quad (R7)$$

Further details of the derivation of the weak form and its coupling with mechanical deformation are provided in Cavalcanti et al. (2024a,b) and Cavalcanti (2026).

2. **Section 6, opening paragraphs:** We have added the following error definition after the sentence ending “reference solutions from the literature.” It is used in the new benchmark reproduced below.

For quantitative comparisons with reference solutions, the discrepancy in a scalar quantity q is measured

using the root mean squared error (RMSE),

$$E_{\text{RMSE},q} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_i^{\text{num}} - q_i^{\text{ref}})^2}, \quad (\text{R8})$$

where q_i^{num} and q_i^{ref} are the numerical and reference values at the same sampling locations, and N is the number of samples. The compared quantities, units, and sampling locations are specified in the corresponding examples.

3. **Section 6, new subsection “Single-phase flow through a regular fracture network”:** We have added this subsection immediately after the dam-foundation example. The problem definition, results, figures, and error table are reproduced below.

Single-phase flow through a regular fracture network

This example assesses the single-phase hydraulic formulation for conductive and blocking fractures using the regular fracture network of Geiger et al. (2013), with the boundary conditions and material properties specified in Benchmark 1 of Flemisch et al. (2018). The problem consists of a unit square containing six intersecting fractures, with the geometry and boundary conditions shown in Fig. R1.

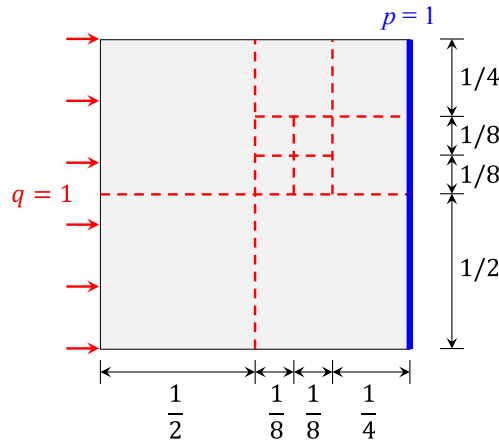


Figure R1: Regular fracture network: geometry and boundary conditions, with $q = 1$ denoting the prescribed inward flux. The dashed lines represent the fractures.

Following the presentation of Benchmark 1 in Flemisch et al. (2018), the parameter values and pressure results are reported without specifying physical units. The matrix permeability is isotropic, $\mathbf{k} = \mathbf{I}$, the fluid viscosity is $\mu_1 = 1$, and the physical fracture aperture is $b = 10^{-4}$. Two configurations are considered: conductive fractures with tangential and normal permeabilities $k_{f,t} = k_{f,n} = 10^4$, and blocking fractures with $k_{f,t} = k_{f,n} = 10^{-4}$. A unit inward flux is prescribed along the left boundary, corresponding to $\mathbf{q}_1 \cdot \mathbf{n} = -1$ for the outward normal $\mathbf{n}$, while $p = 1$ is imposed on the right boundary. The top and bottom boundaries are impermeable. Gravity is neglected, and a linear analysis is performed to obtain the steady-state pressure field.

To reproduce the prescribed tangential conductance and normal fluid transfer, the computational aperture w and the leak-off coefficients $\lambda_d^\pm$ are set through

$$\frac{w^3}{12\mu_1} = \frac{b k_{f,t}}{\mu_1}, \quad \lambda_d^\pm = \frac{2k_{f,n}}{\mu_1 b}. \quad (\text{R9})$$

The domain is discretized with three structured meshes of linear triangular elements, containing 1250, 4802, and 18 818 elements and 824, 2792, and 10 184 free DOFs, respectively. The coarsest mesh and a local zoom are shown in Fig. R2. The interior Cartesian grid lines are shifted by half a cell, placing internal fracture endpoints and junctions on diagonal element edges while preserving the prescribed fracture geometry. The same meshes are used for both permeability configurations.

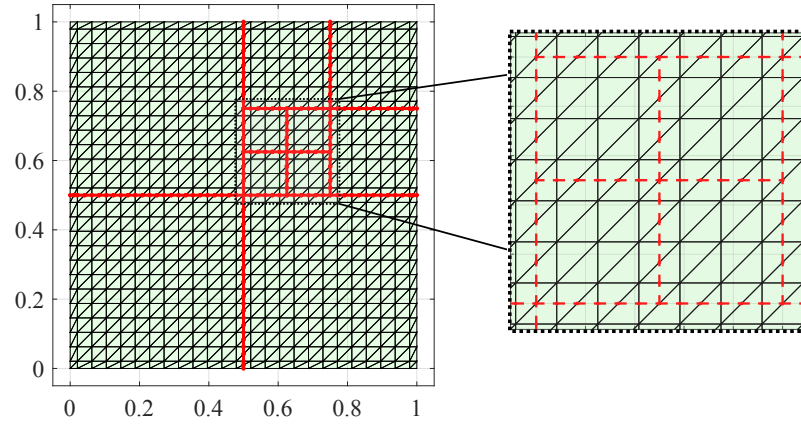


Figure R2: Regular fracture network: coarsest mesh, with 1250 triangular elements, and local enlargement of the highlighted region. The red lines represent the fractures.

Figure R3 presents the pressure fields on the coarsest mesh, with 1250 triangular elements. In Fig. R3b, the pressure jumps across the fractures appear smooth because the current plotting routine in PorousLab interpolates the mesh-node values directly.

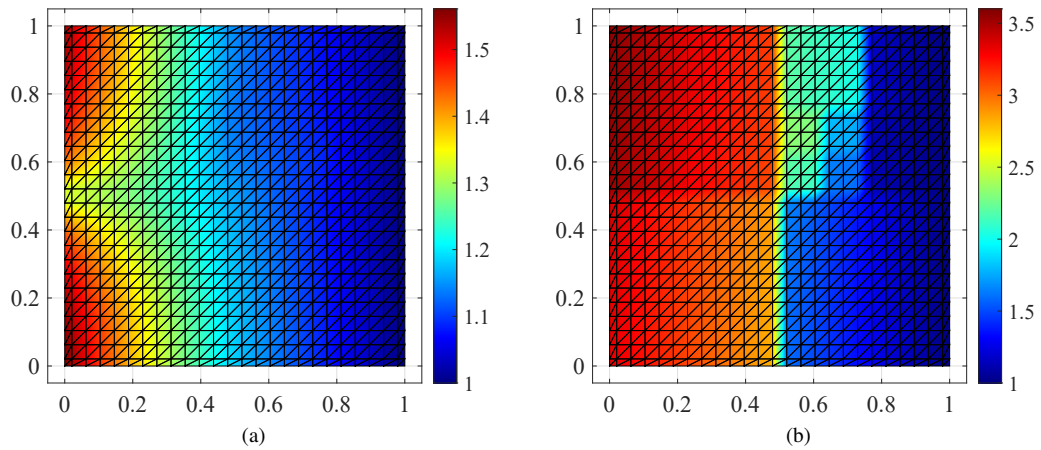


Figure R3: Regular fracture network: pressure fields on the coarsest mesh, with 1250 triangular elements, for (a) conductive fractures and (b) blocking fractures.

Figure R4 compares the numerical profiles with the mimetic finite difference (MFD) reference solution of Flemisch et al. (2018). For the conductive configuration, the profiles along the horizontal line $y = 0.7$ and the vertical fracture at $x = 0.5$ are shown in Figs. R4a and R4b, respectively. The numerical curves approach the MFD reference profiles as the mesh is refined. Figure R4c presents the blocking configuration along the diagonal segment from $(0, 0.1)$ to $(0.9, 1)$. The pressure jumps across the fractures are reproduced on all three meshes, whose profiles closely overlap the MFD reference solution at the displayed scale.

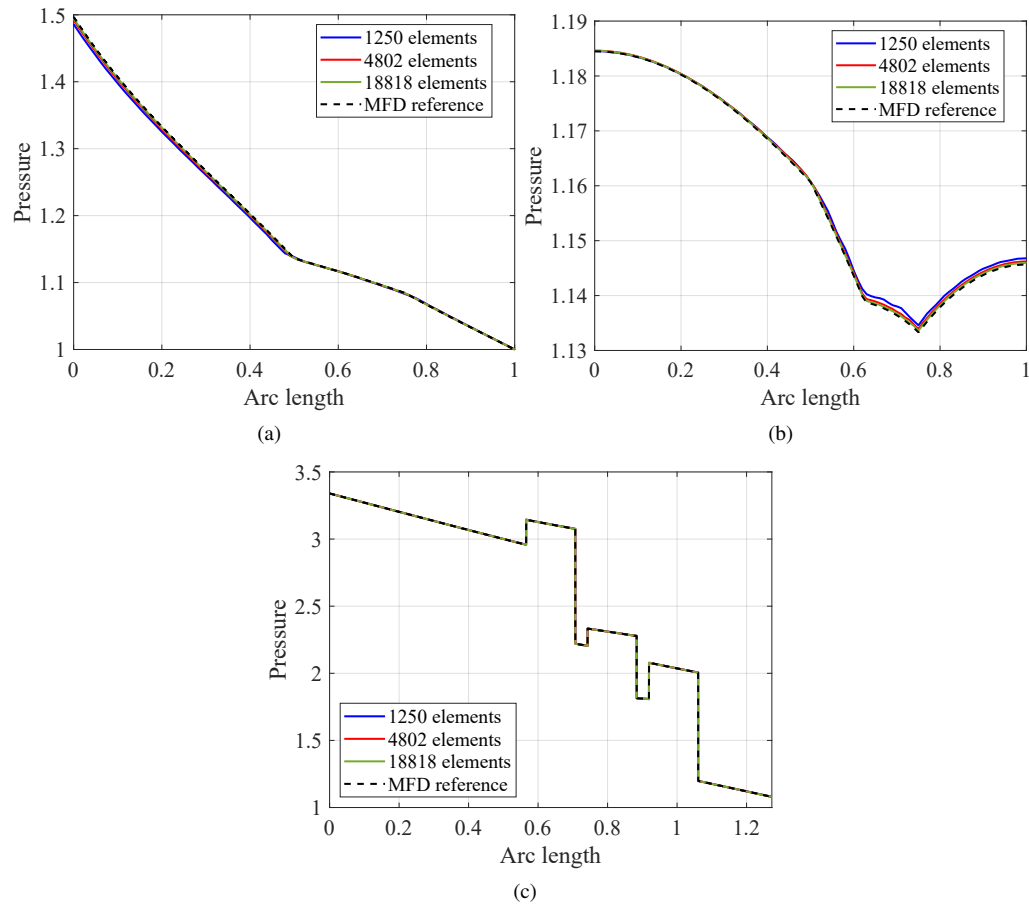


Figure R4: Regular fracture network: pressure profiles on the three meshes and the MFD reference solution. Conductive configuration: (a) horizontal profile at $y = 0.7$ and (b) vertical profile at $x = 0.5$. Blocking configuration: (c) diagonal profile from (0, 0.1) to (0.9, 1). Arc length is measured from (0, 0.7), (0.5, 0), and (0, 0.1) in (a), (b), and (c), respectively.

The RMSE is evaluated using Eq. (R8), with $q = p$, at fixed uniformly spaced positions corresponding to the reference samples. Table R1 shows that the RMSE decreases with refinement for all three profiles. The comparisons support the representation of both conductive and blocking hydraulic behaviour for this fracture network.

Table R1: Regular fracture network: pressure-profile RMSE for the three meshes.

Elements	Free DOFs	Conductive		Blocking Diagonal
		$y = 0.7$	$x = 0.5$	
1250	824	5.379E-3	8.875E-4	4.387E-4
4802	2792	2.447E-3	4.449E-4	3.326E-4
18 818	10 184	1.148E-3	2.228E-4	2.508E-4

4. **Section 6.3, “Fluid flow through the foundation of a dam”:** We have revised the opening sentence to identify the configurations used for the reference comparisons.

The intact and parallel-fracture configurations in this example assess the single-phase hydraulic formulation and the embedded representation of hydraulic discontinuities in a steady seepage problem.

We have replaced the paragraph beginning “A linear analysis is performed” to quantify the intact and parallel-fracture comparisons and identify the random-fracture configuration as illustrative. The revised paragraph is reproduced below.

A linear analysis is performed to obtain the steady-state pore-pressure field. For the intact configuration, the numerical solution is compared with a structured-grid finite-difference solution of the same Laplace problem (Smith, 1985). For the parallel-fracture configuration, the results are compared with reference solutions obtained using discrete-fracture models based on zero-thickness interface elements (Segura and Carol, 2004). Figure 22 shows close agreement with the reference solutions for the intact and parallel-fracture configurations. The pressure-profile RMSEs, computed using Eq. (R8), are 0.039 kPa for the intact foundation and 0.30 kPa for the parallel-fracture configuration. The random-fracture configuration is not a verification test and is included only to illustrate flow through an irregular fracture network.

R1.9 Reviewer comment

"Good agreement" or "close agreement" is stated for every example, but no error measure, no mesh or time-step sizes for the transient cases ("adaptive time stepping" without parameters), and no convergence under refinement are reported. For a model description paper, please give for each example the discretization parameters and DOF count, typically some error norm value with respect to the reference, and for at least one example a refinement study. In the same spirit, the "limited computational performance" and "small- to medium-scale" statements should be accompanied by wall-clock times (just to get a rough idea and be able to compare this to other codes or for someone that develops their own code later as a reference), and DOF counts of the examples.

Author response

We have extended the quantitative assessment of the examples in Section 6 and Appendix D. The root mean squared error (RMSE) is now defined at the beginning of Section 6 (reproduced in Eq. (R8) above), and the corresponding comparisons specify the evaluated quantities and sampling locations. RMSE results are reported for the reservoir, fracture sliding and opening, dam-foundation reference cases, regular fracture network, Wijesinghe, McWhorter-Sunada, Liakopoulos and Terzaghi problems. For the strip-footing example, the final computed pressures are approximately 0.79% above Prandtl's analytical limit pressure. The irregular dam-fracture configuration is explicitly identified as an illustration rather than a verification test.

We have also added spatial-refinement studies. The fracture sliding and opening examples use five refinement levels and three fracture inclinations, with the displacement-jump RMSE plotted against the number of free DOFs. The regular fracture-network benchmark includes three meshes for both conductive and blocking fractures, with pressure-profile RMSEs reported for each mesh. These studies show decreasing errors with refinement for the reported quantities.

The new “Computational cost” subsection at the end of Section 6 reports element types, element counts, free DOFs, total global nonlinear iterations, and mean wall-clock times for all 18 configurations in Table 7 of the revised manuscript (reproduced as Table R2 below). Free DOFs include the globally retained enrichment variables and exclude prescribed and locally condensed variables. For the refinement examples, this summary reports the finest meshes.

The supplementary input listings provide the initial, minimum and maximum time steps for the adaptive transient analyses: Supplement Sections S2.6–S2.8, Listings S30, S35 and S39. The continuation parameters and convergence tolerance for the strip-footing problem are provided in Supplement Section S2.1, Listing S4. Appendix D specifies the fixed time step of 1 s used for the Terzaghi problem.

The measurements were obtained using MATLAB R2022a Update 3 on a 64-bit Windows 11 computer with an Intel Core i7-11390H processor and 16 GB of RAM, using at most four computational threads. Reported wall-clock times are the arithmetic mean of three runs for each configuration and cover numerical initialization and solution over all analysis stages. The subsection specifies the excluded preparation and post-processing tasks and defines the

iteration count as the total number of global nonlinear iterations, including rejected increments.

Mean wall-clock times range from 0.42 s for the intact dam foundation to 1138.80 s for the Wijesinghe problem. The latter has 3349 free DOFs during injection and requires 4036 global nonlinear iterations over the full analysis. Iteration counts were identical across the three runs for every configuration; linear analyses have zero nonlinear iterations.

Changes in the manuscript

Section 6, “Computational cost” subsection and Table 7: We have completed the table with mean wall-clock times and total global nonlinear iteration counts for all 18 configurations. The subsection and completed table are reproduced below.

Table R2 summarizes the discretization sizes and computational costs of the examples. The finest meshes are selected for the fracture sliding, fracture opening, and regular fracture-network problems. Transient calculations extend to the latest comparison time reported for each example. Free DOFs include the globally retained enrichment variables and exclude prescribed and locally condensed variables.

All computations were performed using MATLAB R2022a Update 3 on a computer equipped with an Intel Core i7-11390H processor (3.40 GHz, four physical cores and eight logical processors) and 16 GB of RAM, running 64-bit Windows 11. MATLAB was used with a maximum of four computational threads.

Wall-clock times cover numerical initialization and solution over all analysis stages. Mesh generation or loading, model construction, plotting, reference-solution calculations, and error evaluation are excluded. The iteration count is the total number of global nonlinear iterations over the full analysis, including rejected increments.

Reported wall-clock times are the arithmetic mean of three runs for each configuration.

Table R2: Discretization sizes, total global nonlinear iterations (N_{it}), and mean wall-clock times of the numerical examples.

Example / configuration	Elements (type)	free DOFs	N_{it}	Wall time (s)
Strip footing: Drucker–Prager	900 (Q8)	5459	1054	260.67
Strip footing: Mohr–Coulomb	900 (Q8)	5459	224	91.41
Reservoir fault, 60°	3600 (Q4)	7413	0	3.92
Fracture sliding, 15°	28 056 (Q4)	57 096	16	85.44
Fracture sliding, 30°	28 056 (Q4)	57 132	13	74.17
Fracture sliding, 45°	28 056 (Q4)	57 144	8	54.54
Fracture opening, 15°	28 056 (Q4)	57 096	8	54.73
Fracture opening, 30°	28 056 (Q4)	57 132	8	54.83
Fracture opening, 45°	28 056 (Q4)	57 144	8	56.53
Dam: intact	576 (Q4)	595	0	0.42
Dam: parallel fractures	576 (Q4)	1170	0	0.99
Dam: irregular fractures	576 (Q4)	827	0	0.57
Fracture network: conductive	18 818 (T3)	10 184	0	7.84
Fracture network: blocking	18 818 (T3)	10 184	0	7.82
Wijesinghe, $t_f = 2000$ s	1050 (Q4)	3349 ^a	4036	1138.80
McWhorter–Sunada, $t_f = 1000$ s	100 (Q4)	400	511	8.33
Liakopoulos, $t_f = 7200$ s	100 (Q4)	598	516	15.22
Terzaghi, $t_f = 500$ s; Appendix D	250 (Q4)	804	1000	52.47

Zero values of N_{it} indicate linear analyses. ^aFree DOFs during injection stage. The initial equilibrium stage has 2253 free DOFs.

Specific comments

R1.10 Reviewer comment

Sect. 6.1 probably MPa instead kPa for load?

Author response

We have corrected the pressure label in the geometry figure from 7.269 kPa to 7.269 MPa.

Changes in the manuscript

Section 6.1, geometry figure: We have replaced the pressure label “7.269 kPa” with “7.269 MPa”. The updated

figure is reproduced below.

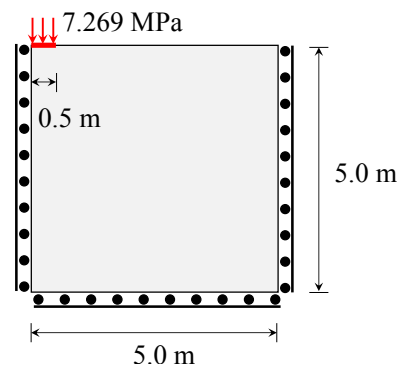


Figure R5: Strip footing: geometry and boundary conditions.

R1.11 Reviewer comment

also Sect. 6.1 text says "a rigid footing applies a uniform vertical pressure" — but a rigid footing imposes a uniform displacement; a uniform pressure would correspond to a flexible footing. Clarify.

Author response

The numerical model applies a uniform vertical traction over a surface strip, corresponding to a perfectly flexible footing. The description as a rigid footing was therefore incorrect. Prandtl's limit pressure also applies to uniform strip loading on a homogeneous, weightless Mohr–Coulomb half-space with associated flow, as considered by Muñoz et al. (2009). Consequently, the comparison remains appropriate for assessing the predicted ultimate pressure, but does not verify the load–settlement response of a rigid footing. We have revised Sect. 6.1 to clarify the loading conditions and the scope of the comparison. We also report the final computed load factors to provide a quantitative comparison with Prandtl's analytical limit pressure. The corresponding relative pressure errors are approximately 0.79% for both plasticity models.

Changes in the manuscript

1. **Section 6.1, opening paragraph:** We have replaced the first two sentences with the following text.

The elastoplastic implementation is assessed using the classical strip-footing bearing-capacity problem in plane strain, with the material parameters adopted by de Souza Neto et al. (2011). The soil is weightless, and a uniform vertical pressure is applied over a finite strip at the ground surface, corresponding to a perfectly flexible footing.

2. **Section 6.1, paragraph following the geometry figure:** We have replaced the opening sentence to identify the reference pressure.

The applied pressure corresponds to Prandtl's analytical limit pressure for uniform strip loading on a homogeneous, weightless Mohr–Coulomb half-space with associated flow (Muñoz et al., 2009).

3. **Section 6.1, constitutive-model description:** We have replaced the sentence introducing the plasticity models to specify the parameter matching.

The problem is analyzed with both the Drucker–Prager and Mohr–Coulomb plasticity models, with the Drucker–Prager parameters matched to Mohr–Coulomb under plane-strain conditions.

4. **Section 6.1, results paragraph:** We have replaced the final sentence describing the comparison with Prandtl's solution and added the final load factors and relative pressure errors.

The two models give closely matching load–displacement curves and approach an ultimate pressure close to Prandtl's analytical value. At the final computed step, the load factors are 1.00791 for Drucker–Prager and

1.00792 for Mohr–Coulomb, corresponding to pressures approximately 0.79% above Prandtl’s analytical value.

R1.12 Reviewer comment

Sect. 6.2: material parameters are missing.

Author response

The continuum material parameters and fault stiffnesses were provided in the original supplementary material (Sect. S2.2, Listings S6 and S8), as part of the input script for reproducing this example. We have now also included the material parameters in Sect. 6.2 to make the example description self-contained.

Changes in the manuscript

Section 6.2, “Stress changes along a fault in a pressurized reservoir”: We have replaced the opening sentence of the paragraph following the geometry figure with the following material description.

The continuum is modeled as a linear-elastic porous medium under plane-strain conditions, with Young’s modulus $E = 11.868$ GPa, Poisson’s ratio $\nu = 0.29$, and Biot’s coefficient $\alpha = 1$. The embedded fault is represented with the mechanical E-FEM formulation and an elastic cohesive law, with normal stiffness of 10^{15} Pa/m and shear stiffness of approximately 2.96×10^{14} Pa/m.

R1.13 Reviewer comment

Sect. 6.3: geometry, aperture and permeability of the parallel and random fracture sets are not given

Author response

The fracture-generation settings and aperture are provided in Supplement Sect. S2.4, Listing S19 and its accompanying text. The longitudinal fracture permeability is calculated from the aperture using the cubic law presented in Sect. 3.7.3. We have now stated the geometry parameters and aperture explicitly in the dam-foundation example, together with a direct reference to the supplementary listing.

Changes in the manuscript

Section 6.3, “Fluid flow through the foundation of a dam”: We have added the following paragraph after the paragraph describing the mesh and the three configurations.

The parallel fractures are inclined at -45° to the horizontal, with a horizontal spacing of 1 m. The irregular configuration contains 20 randomly generated fractures with a minimum length of 1 m, with a given random seed. In both configurations, all fractures have a constant aperture of $w = 10^{-4}$ m. The commands defining both fracture configurations are provided in Supplement Sect. S2.4, Listing S19 and its accompanying text.

R1.14 Reviewer comment

the Jacobian in Eq. (64) omits derivatives of K and C , although K and C depend on x in the two-phase physics (relative permeability, saturation, density); Appendix B includes these derivatives (B15–B17). Either state that Eq. (64) applies when K and C do not depend on x , or write it consistently with Appendix B.

Author response

We have made this dependence explicit in the residual notation and revised the Jacobian expression to differentiate the complete matrix–vector products. The revised expression includes the solution dependence of $\mathbf{K}$ and $\mathbf{C}$, together with any solution-dependent force terms.

Changes in the manuscript

1. **Section 3.8.3, “Nonlinear transient analysis”:** We have replaced the residual expression, Eq. (61), with the following, making the dependence on the unknown DOFs explicit.

$$\mathbf{r}(\mathbf{x}(t), t) = \mathbf{f}_i(\mathbf{x}(t), t) + \mathbf{K}(\mathbf{x}(t), t) \mathbf{x}(t) + \mathbf{C}(\mathbf{x}(t), t) \dot{\mathbf{x}}(t) - \mathbf{f}_e(\mathbf{x}(t), t), \quad (\text{R10})$$

2. **Section 3.8.3, “Nonlinear transient analysis”:** We have replaced the Jacobian expression, Eq. (64), with the following.

$$\mathbf{J}_n = \frac{\partial \mathbf{f}_{i,n}}{\partial \mathbf{x}_n} + \frac{\partial (\mathbf{K}_n \mathbf{x}_n)}{\partial \mathbf{x}_n} + \frac{1}{\Delta t} \frac{\partial [\mathbf{C}_n (\mathbf{x}_n - \mathbf{x}_{n-1})]}{\partial \mathbf{x}_n} - \frac{\partial \mathbf{f}_{e,n}}{\partial \mathbf{x}_n}. \quad (\text{R11})$$

R1.15 Reviewer comment

I think you should cite the specific version on Zenodo <https://doi.org/10.5281/zenodo.20612078>

Author response

We have updated the Code and data availability statement and the corresponding bibliographic reference to cite the version-specific DOI of the revised PorousLab v1.1.1 release, <https://doi.org/10.5281/zenodo.23565647>.

Changes in the manuscript

Code and data availability and References: We have updated the archived-release sentence and the corresponding bibliographic reference to identify PorousLab v1.1.1 and its version-specific DOI.

The exact archived release of PorousLab v1.1.1 is available from Zenodo at <https://doi.org/10.5281/zenodo.23565647> (Cavalcanti, 2026).

R1.16 Reviewer comment

As far as I know, PorePy’s bulk model is linear elastic, but its fracture treatment is a frictional contact-mechanics formulation solved with a semismooth Newton method, which is more general than the penalty-based cohesive treatment used here ("contact-penalty parameters"). "More limited in scope" seems to be a mischaracterization.

Author response

We have removed the characterization of PorePy’s geomechanical capabilities as “more limited in scope”. The Introduction now explicitly states that Table 1’s bulk-behavior classification excludes fracture-contact laws.

Changes in the manuscript

Section 1, paragraph defining the Table 1 classifications: We have replaced the explanation beginning “For geomechanics, we distinguish” with the following definition of bulk behavior.

“General” denotes bulk constitutive behavior beyond linear elasticity, such as plasticity or damage; this classification excludes fracture-contact laws.

R1.17 Reviewer comment

Appendix A2: Relative permeabilities are arithmetic nodal averages of saturation without upwinding. This is adequate for the capillary-driven McWhorter problem but is known to lose monotonicity for advection-dominated displacement; a Buckley–Leverett test would maybe show the behaviour, otherwise the limitation should be stated.

Author response

We have stated explicitly at the end of Appendix A2 that evaluating relative permeabilities at the mean nodal saturation does not include upwinding and does not guarantee monotone saturation profiles in advection-dominated displacement.

Changes in the manuscript

Appendix A2, final paragraph: We have added the following paragraph after the expression for the mean nodal density.

The evaluation of relative permeabilities at the mean nodal saturation does not include upwinding and does not guarantee monotone saturation profiles in advection-dominated displacement. When capillary diffusion is weak, resolving the saturation transition may require substantial mesh refinement.

R1.18 Reviewer comment

Local mass conservation: "Guarantee local mass conservation" (l. 639, 657)" the flux terms are standard Galerkin -> please clarify what "local" refers to, and report the global mass-balance error for the transient examples.

Author response

We have removed the claim that this treatment guarantees local mass conservation. Nodal evaluation and lumping of the storage terms do not, by themselves, establish local conservation of the Galerkin flux discretization. Appendix A now describes the direct time discretization of the storage quantities and their nodal evaluation and lumping. The flux terms retain the standard Galerkin discretization.

Regarding the requested global mass-balance errors, the current version of PorousLab does not provide the diagnostics needed to track phase storage and boundary mass exchanges at each accepted time step. These errors cannot be recovered from the existing final-state outputs alone. Their evaluation would require additional implementation and verification, followed by rerunning the transient examples. This assessment has not been undertaken in the present revision, and the magnitude of the global mass-balance errors therefore remains unquantified.

Changes in the manuscript

1. **Appendix A, introductory paragraph:** We have replaced the opening paragraph beginning “To guarantee the mass conservation in time” with the following text. The existing backward-difference expression is reproduced for context.

The time derivative of $\rho_\pi S_\pi$ is approximated directly by a backward difference, without expanding it in terms of the primary variables. This follows the direct treatment of accumulation terms in Olivella et al. (1996) and the principle of directly differencing secondary quantities described by Grunwald et al. (2022)

$$\frac{\partial()}{\partial t} \approx \frac{()_n - ()_{n-1}}{\Delta t} \quad (\text{R12})$$

where Δt is the time step size.

2. **Appendix A1, “Treatment of the mass storage terms”:** We have replaced the paragraph beginning “To guarantee local mass conservation” with the following description of the nodal storage treatment.

For the storage terms, saturation and density are evaluated from the nodal pressures using the constitutive properties of each contributing element, following Olivella et al. (1996). The resulting storage contributions are lumped at the corresponding nodes and assembled element by element. Figure A1 illustrates the associated node-centered cells. For instance, the cell associated with node 6 receives contributions from elements 1, 2, 4, and 5.

R1.19 Reviewer comment

the HM discontinuity terms: the aperture is updated from the mechanical deformation and enters the cubic law (Eq. (46)) so the fracture transmissivity depends on the jump DOFs. Appendix B lists the H2M Jacobian as I see it without discontinuities only. Is the aperture lagged? Is it really the full Jacobian.

Author response

The aperture used in the cubic law is taken from the previous converged time step and held fixed during the current

time step. Its dependence on the current displacement-jump DOFs therefore does not enter the Jacobian. Appendix B concerns the H2M formulation, which currently does not support embedded discontinuities. We have clarified the aperture treatment immediately after the cubic-law equation and the scope of Appendix B in its introductory paragraph.

Changes in the manuscript

1. **Section 3.7.3, “Fluid flow inside the discontinuity”:** We have added the following sentence immediately after the cubic-law equation.

In hydro-mechanical analyses with evolving aperture, the value used in the cubic law is taken from the previous converged time step.

2. **Appendix B, introductory paragraph:** We have added the following sentence after the opening sentence referring to Appendix A.

Embedded discontinuities are not currently implemented for the H2M physics.

R1.20 Reviewer comment

Abstract, l. 3: "rely on mixed discretizations" suggests mixed finite elements; use "combine different discretizations" or drop the phrase.

Author response

We have replaced the sentence containing “rely on mixed discretizations” with a neutral description of existing open-source frameworks, avoiding confusion with mixed finite-element formulations.

Changes in the manuscript

Abstract, second sentence: We have replaced the sentence containing “rely on mixed discretizations” with the following text.

Existing open-source frameworks address a wide range of porous-media applications and support large-scale simulation and numerical prototyping.

R1.21 Reviewer comment

Addressed

l. 116: "include several continuation strategies" → "includes".

R1.22 Reviewer comment

Addressed

l. 137: "validated on MATLAB R2022a" → "tested on".

R1.23 Reviewer comment

Addressed

l. 195: "repectively" → "respectively".

R1.24 Reviewer comment

Fig. 2: the labels n , s , h , g_0 , t_0 are not explained in the caption or text.

Author response

We have revised the paragraph preceding Fig. 2 in Section 3.7.1 to define n , s , h , g_0 , and t_0 . We have also clarified that Ω_h is the union of the finite elements crossed by the discontinuity and that its characteristic width h depends on the local mesh size.

Changes in the manuscript

1. **Section 3.7.1, “Strong Discontinuity Approach”:** We have replaced the description beginning “It considers the problem illustrated in Fig. 2” with the following text, retaining the opening sentence and its citations.

Figure 2 illustrates a domain Ω whose external boundary has outward unit normal $\mathbf{n}$. Field values $\mathbf{g}_0$ and tractions $\mathbf{t}_0$ are prescribed on the Dirichlet and Neumann boundary portions, Γ_D and Γ_N , respectively. A discontinuity Γ_d , with local tangential coordinate s , partitions the domain into two subdomains, Ω^+ and Ω^- . These subdomains are labelled according to the unit normal vector $\mathbf{n}_d$ to the discontinuity, which points towards Ω^+ . Assuming the discontinuity surface is smooth, the unit normal vectors on both sides satisfy $\mathbf{n}_d^- = -\mathbf{n}_d^+ = \mathbf{n}_d$. In the finite-element discretization, Ω_h denotes the union of the elements crossed by the discontinuity Γ_d . The characteristic width h of this region depends on the local mesh size.

2. **Section 3.7.1, immediately after Fig. 2:** We have shortened the sentence introducing the field decomposition to avoid repeating the definition of Ω_h . The decomposition equation remains unchanged.

To represent a discontinuous field within the finite-element framework, a generic field $\mathbf{g}(\mathbf{x})$ defined over Ω_h is decomposed as

R1.25 Reviewer comment

Eq. (43): ϕ_d is not defined; state its meaning (and value) for an open fracture.

Author response

We have clarified that ϕ_d denotes the porosity of the discontinuity domain and equals unity for an open fracture without solid infill. This clarification appears immediately after the fluid continuity equation in Section 3.7.3, “Fluid flow inside the discontinuity.”

Changes in the manuscript

Section 3.7.3, “Fluid flow inside the discontinuity”: We have revised the paragraph immediately following the fluid continuity equation to define ϕ_d and state its value for an open fracture without solid infill.

where ϕ_d is the discontinuity porosity, with $\phi_d = 1$ for open fractures, w is the aperture, s is the tangential coordinate, and $q_{d,1}^t$ is the net outward flux through the fracture faces.

R1.26 Reviewer comment

1. 305: a Delaunay triangulation is only "conforming" to the interface if it is constrained or performed per side; rephrase.

Author response

The implementation uses a constrained Delaunay triangulation, with the embedded discontinuity segments explicitly imposed as edges. These constraints ensure that the sub-triangles conform to the interface. We have clarified this point in Section 3.7.2 by specifying the constrained triangulation and its edge constraints.

Changes in the manuscript

Section 3.7.2, “Mechanical behavior”: We have replaced the sentence beginning “Here, each cut element” with the following text.

Here, each cut element is partitioned into sub-triangles using a constrained Delaunay triangulation, with the embedded discontinuity segments imposed as edges, and Gauss points are assigned per sub-cell using the native triangular rule; see Fig. 7 of the manuscript.

R1.27 Reviewer comment**Addressed**

Fig. 20 caption: "The fractured is represented" → fracture

R1.28 Reviewer comment

Fig. 23 legend says "Analytical", the text says semi-analytical.

Author response

We have replaced "Analytical" with "Semi-analytical" in the legend of Fig. 23 to make it consistent with the text. The reference is based on the exact integral solution of McWhorter and Sunada, whose evaluation requires numerical integration and iteration.

Changes in the manuscript

Figure 23, panels (a) and (b): We have replaced "Analytical" with "Semi-analytical" in both legends. The updated figure is reproduced as Fig. R6 below.

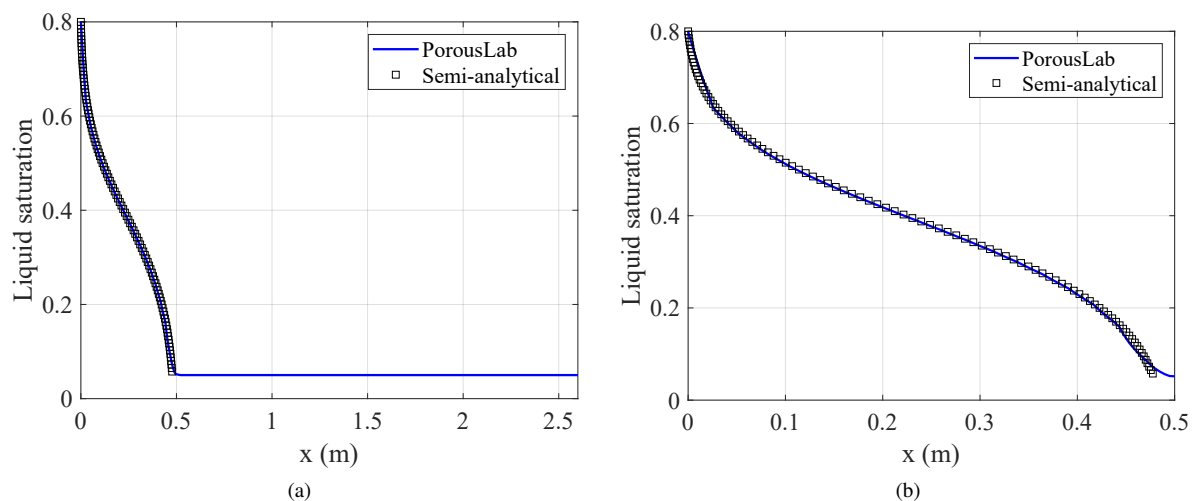


Figure R6: McWhorter and Sunada infiltration problem: (a) liquid saturation profile at 1000.0 s and (b) zoomed view.

R1.29 Reviewer comment**Addressed**

Helmig (1997) is a single-author book, remove "et al."

References

- Danilo Cavalcanti, Cristian Mejia, Deane Roehl, Ignasi de Pouplana, Guillermo Casas, and Luiz F Martha. Embedded finite element formulation for fluid flow in fractured porous medium. *Computers and Geotechnics*, 171:106384, 2024a. doi: 10.1016/j.compgeo.2024.106384.
- Danilo Cavalcanti, Cristian Mejia, Deane Roehl, Ignasi de Pouplana, and Eugenio Oñate. Hydromechanical embedded finite element for conductive and impermeable strong discontinuities in porous media. *Computers and Geotechnics*, 172:106427, 2024b. doi: 10.1016/j.compgeo.2024.106427.
- Danilo Borges Cavalcanti. *Finite element method with embedded strong discontinuities for coupled hydro-mechanical problems*. PhD thesis, Universitat Politècnica de Catalunya, 2026. URL <https://hdl.handle.net/2117/460975>.
- Eduardo A de Souza Neto, Djordje Peric, and David RJ Owen. *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons, 2011.

- Bernd Flemisch, Melanie Darcis, K Erbertseder, B Faigle, A Lauser, Klaus Mosthaf, S Müthing, Philipp Nuske, A Tatomir, M Wolff, et al. Dumux: Dune for multi-{phase, component, scale, physics,...} flow and transport in porous media. *Advances in Water Resources*, 34(9):1102–1112, 2011.
- Bernd Flemisch, Inga Berre, Wietse Boon, Alessio Fumagalli, Nicolas Schwenck, Anna Scotti, Ivar Stefansson, and Alexandru Tatomir. Benchmarks for single-phase flow in fractured porous media. *Advances in Water Resources*, 111:239–258, 2018. doi: 10.1016/j.advwatres.2017.10.036.
- S. Geiger, M. Dentz, and I. Neuweiler. A novel multirate dual-porosity model for improved simulation of fractured and multiporosity reservoirs. *SPE Journal*, 18(4):670–684, 2013. doi: 10.2118/148130-PA.
- Glenn E Hammond, Peter C Lichtner, and RT Mills. Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN. *Water Resources Research*, 50(1):208–228, 2014. doi: 10.1002/2012WR013483.
- Eirik Keilegavlen, Runar Berge, Alessio Fumagalli, Michele Starnoni, Ivar Stefansson, Jhabriel Varela, and Inga Berre. Porepy: an open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences*, 25:1–23, 2021. doi: 10.1007/s10596-020-10002-5.
- Olaf Kolditz, Sebastian Bauer, Lars Bilke, Norbert Böttcher, Jens-Olaf Delfs, Thomas Fischer, Uwe J Görke, Thomas Kalbacher, Georg Kosakowski, CI McDermott, et al. OpenGeoSys: An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media. *Environmental Earth Sciences*, 67:589–599, 2012. doi: 10.1007/s12665-012-1546-x.
- Stein Krogstad, Knut-Andreas Lie, Olav Møyner, Halvor Møll Nilsen, Xavier Raynaud, and Bård Skaflestad. MRST-AD—an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In *SPE Reservoir Simulation Conference*, page D022S002R004, 2015. doi: 10.2118/173317-MS.
- Knut-Andreas Lie. *An introduction to reservoir simulation using MATLAB/GNU Octave: User guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge University Press, 2019.
- J. J. Muñoz, J. Bonet, A. Huerta, and J. Peraire. Upper and lower bounds in limit analysis: Adaptive meshing strategies and discontinuous loading. *International Journal for Numerical Methods in Engineering*, 77(4): 471–501, 2009. doi: 10.1002/nme.2421.
- J. M. Segura and Ignacio Carol. On zero-thickness interface elements for diffusion problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(9):947–962, 2004. ISSN 03639061. doi: 10.1002/nag.358.
- Randolph R Settgaest, Ryan M Aronson, Julien R Besset, Andrea Borio, Quan M Bui, Thomas J Byer, Nicola Castelletto, Aurélien Citrain, Benjamin C Corbett, James Corbett, et al. GEOS: A performance portable multi-physics simulation framework for subsurface applications. *Journal of Open Source Software*, 9(102):6973, 2024. doi: 10.21105/joss.06973.
- Gordon D Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.