

GEOSCIENTIFIC MODEL DEVELOPMENT

Response to Reviewer 2

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 #2 | RC2, 16 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-RC2>

Summary

R2 Reviewer comment

The manuscript presents PorousLab, an open-source, object-oriented MATLAB finite-element framework for coupled mechanical, hydraulic, and hydro-mechanical problems in porous media. Pre-existing fractures and faults are represented using an embedded finite-element formulation.

Overall, the manuscript is well written and clearly structured. The software architecture is described in useful detail, and the different physics modules are compared with analytical or literature benchmarks in Section 6, generally showing good agreement. I think the paper is suitable for publication after the authors address the following points. Most of my comments are related to fracture-specific verification and the lack of information on solver performance and computational cost.

Author response

We thank the reviewer for the careful reading of the manuscript and the constructive comments. We have added frictional sliding and opening benchmarks with mesh-refinement studies and extended the transient pressure and aperture comparisons in the Wijesinghe example, including quantitative error measures. We have also clarified the limitations of the current formulation for fracture intersections and coalescence, stated that the solver is serial, and added element and degree-of-freedom counts for the numerical examples. Our detailed responses and the corresponding manuscript changes are presented below.

Major comments

R2.1 Reviewer comment

The Wijesinghe fracture-flow benchmark only shows the pressure/aperture response at final time. It would be useful to also include a time-history plot along the fracture so that the transient response can be seen more clearly.

Author response

We have extended the comparison in Fig. 21 to include the numerical and semi-analytical pore-pressure and fracture-aperture profiles along the fracture at 100, 250, 500, and 2000 s after the inlet pressure step. These successive

profiles show the propagation of the pressure disturbance and the associated fracture opening during the transient response. To quantify the agreement, we also report the root mean squared error (RMSE) of each field at each comparison time. The definition added at the beginning of Section 6 is reproduced in Eq. (R1) below, and the errors are given in Table R1. The corresponding revisions to Section 6.4 and Fig. 21 are reproduced below, including the specification of the relative rotation enrichment used in this example.

Changes in the manuscript

1. **Section 6, p. 26, line 511:** We have inserted the following paragraph and equation after the sentence ending “reference solutions from the literature.”

For quantitative comparisons with reference solutions, the discrepancy in a scalar quantity q is measured using the root mean squared error (RMSE),

$$E_{\text{RMSE},q} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_i^{\text{num}} - q_i^{\text{ref}})^2}, \quad (\text{R1})$$

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.

2. **Section 6.4, p. 32, line 573:** We have added the following sentence after the fracture material parameters and before the mesh description.

The displacement enrichment includes the constant normal and tangential jump modes and the relative rotation mode, allowing the normal displacement jump to vary linearly along each embedded fracture segment.

3. **Section 6.4, p. 32, lines 579–581:** We have replaced the results paragraph with the following text and added the error table immediately after it.

Figure R1 compares the pore-pressure and fracture-aperture profiles obtained with PorousLab against the semi-analytical solution of Wijesinghe (1986) at $t = 100, 250, 500$, and 2000 s after application of the inlet pressure step. The close agreement at all four times supports the verification of the transient hydro-mechanical coupling between fracture pressure and aperture variation. For a quantitative comparison, both fields are sampled at the same fracture integration points, and their root mean squared errors are computed separately at each selected time using Eq. (R1). The resulting errors are reported in Table R1.

Table R1: RMSE in the pore-pressure and fracture-aperture profiles with respect to the semi-analytical solution of Wijesinghe (1986).

t (s)	$E_{\text{RMSE},p}$ (MPa)	$E_{\text{RMSE},b}$ (mm)
100	7.518E-3	9.221E-5
250	4.668E-3	6.498E-5
500	4.404E-3	5.928E-5
2000	7.840E-3	8.443E-5

4. **Figure 21 and its caption, p. 33:** We have updated the figure and its caption to show the profiles at four comparison times. The updated figure is reproduced as Fig. R1 below.

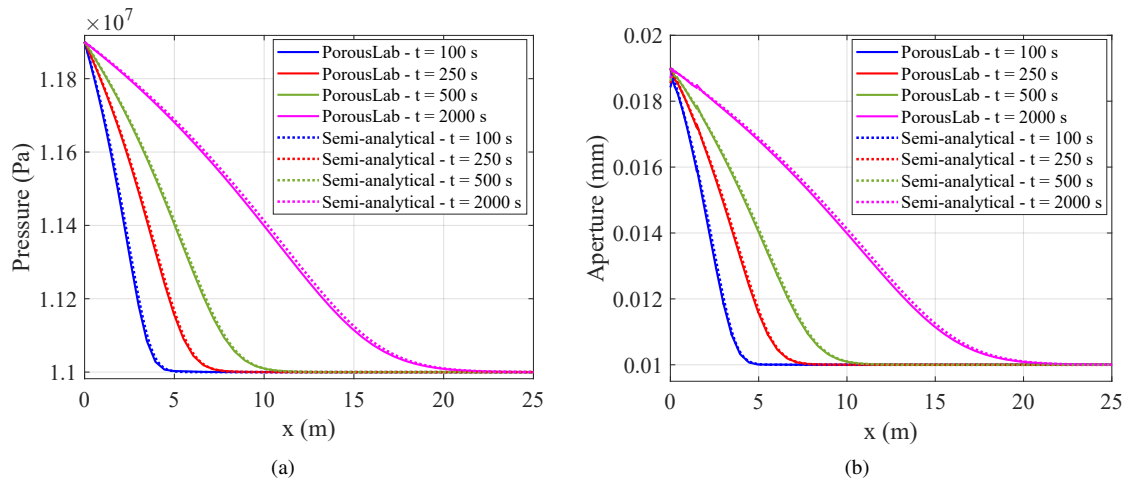


Figure R1: Wijesinghe problem: profiles of (a) pore pressure and (b) fracture aperture at 100, 250, 500, and 2000 s.

R2.2 Reviewer comment

No slip response is shown for the embedded fracture formulation. Since frictional slip appears to be an important capability of the framework, I suggest adding a classical stick-slip benchmark, for example, based on Oden & Pires (1983) or Simo & Laursen (1992).

Author response

We have added a frictional-sliding benchmark in Section 6, in the new subsection “Sliding and opening of a finite fracture”. The example follows the single-fracture benchmark presented by An et al. (2024), for which an analytical solution is available. The same geometry is used for the mechanical-opening benchmark presented in Case 2, which addresses R2.5.

Changes in the manuscript

Section 6, new subsection “Sliding and opening of a finite fracture”: We have added this subsection between the pressurized-reservoir and dam-foundation examples. The shared problem description and Case 1, including its analytical reference, results, and figure, are reproduced below. The shared description also provides the setup for the opening case associated with R2.5.

Sliding and opening of a finite fracture

This example assesses the mechanical embedded-discontinuity formulation through two loading cases: frictional sliding under compression and opening under tension. The setup is based on the single-fracture benchmarks presented by An et al. (2024). A straight fracture of length $L_f = 10$ m is located at the centre of a 200×200 m square domain under plane-strain conditions, as shown in Fig. R2. The surrounding material is isotropic and linear-elastic, with Young’s modulus $E = 10$ GPa and Poisson’s ratio $\nu = 0.25$. The fracture inclination β is measured from the horizontal direction. Vertical displacement is restrained at the midpoints of the left and right boundaries, and horizontal displacement is restrained at the midpoints of the top and bottom boundaries.

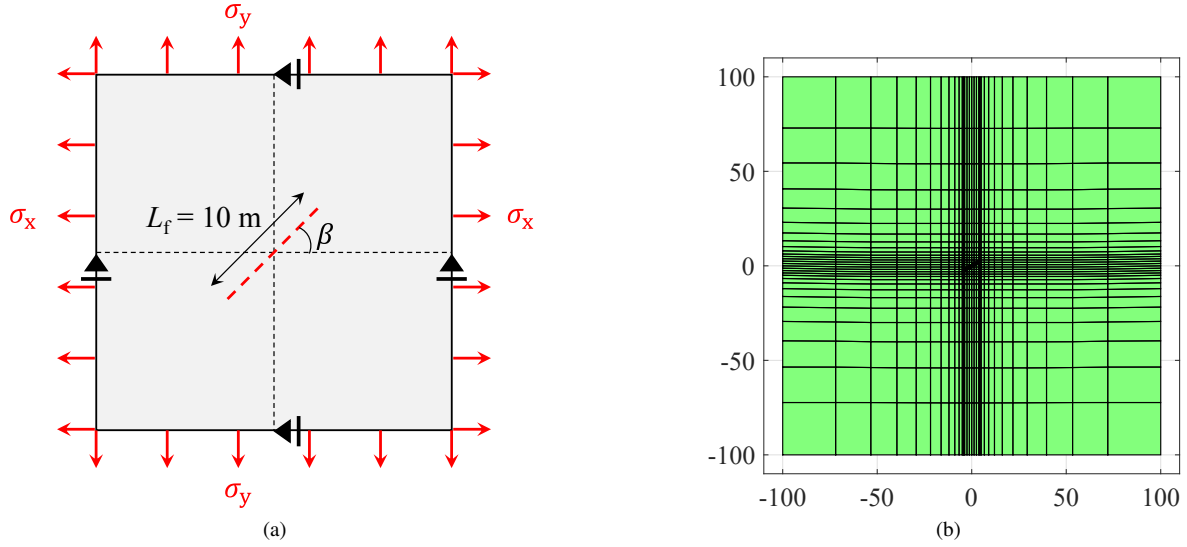


Figure R2: Sliding and opening of a finite fracture: (a) geometry and boundary conditions; (b) graded finite element mesh, with coordinates in metres. The outward arrows in (a) define the positive (tensile) directions of the boundary stresses. The applied values are $\sigma_x = -\sigma_0$, $\sigma_y = 0$ for Case 1 (horizontal compression) and $\sigma_x = \sigma_y = \sigma_0$ for Case 2 (biaxial tension).

The fracture follows a Mohr–Coulomb contact law with friction angle $\varphi_f = 30^\circ$, zero dilation, zero cohesion, zero tensile strength, and zero initial aperture. The domain is discretized with graded meshes of bilinear quadrilateral elements, refined around the fracture, as illustrated in Fig. R2b. Five refinement levels are considered, with a characteristic mesh size near the fracture of $\ell = L_f/n_{\text{ref}}$, where $n_{\text{ref}} = 8, 16, 32, 64$, and 128. The normal and shear interface stiffnesses are set to $K_n = K_m = 10^3 E/\ell$.

Case 1: Frictional sliding

Horizontal compressive tractions of magnitude $\sigma_0 = 10$ MPa are applied to the left and right boundaries, while the top and bottom boundaries are traction-free apart from the point constraints described above. Fracture inclinations of $\beta = 15^\circ, 30^\circ$, and 45° are considered. The analytical tangential displacement jump reported by An et al. (2024) is

$$u_d^m(s) = \frac{4(1-\nu^2)}{E} \sigma_0 \sin \beta (\cos \beta - \mu_f \sin \beta) \sqrt{\left(\frac{L_f}{2}\right)^2 - \left(s - \frac{L_f}{2}\right)^2}, \quad 0 \leq s \leq L_f, \quad (\text{R2})$$

where s is the distance along the fracture from its first tip and $\mu_f = \tan \varphi_f$ is the friction coefficient. For $\varphi_f = 30^\circ$, the analytical slip profiles at $\beta = 15^\circ$ and 45° coincide.

Figure R3 compares the numerical and analytical profiles and presents the refinement study. The RMSE is computed using Eq. (R1), with $q = u_d^m$, at the same $N = 10\,000$ uniformly spaced interior positions, $s_i = (i - 1/2)L_f/N$, for every mesh and inclination. The five meshes contain 870, 1722, 4032, 9900, and 28 056 continuum elements.

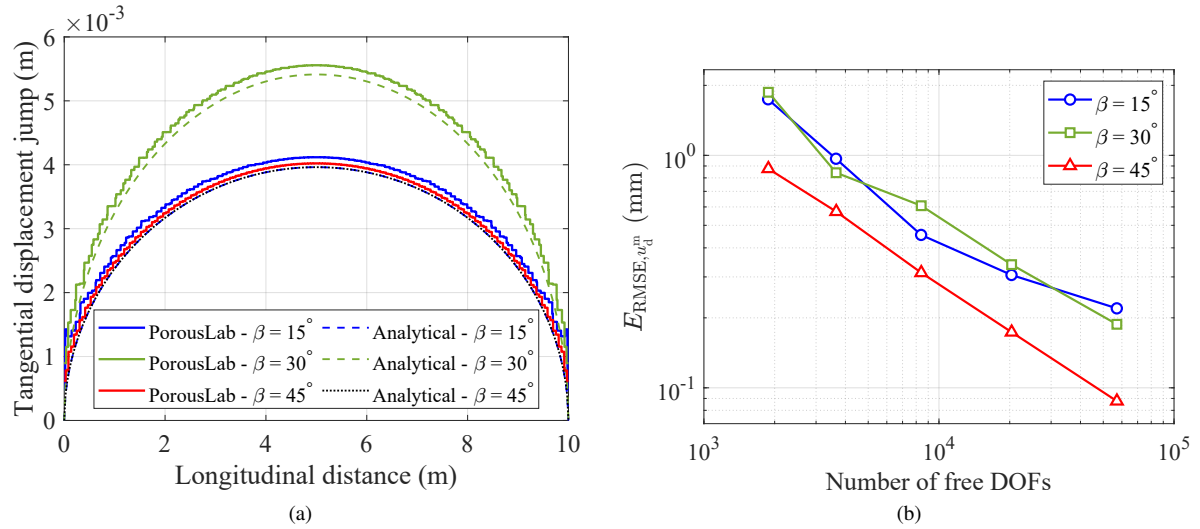


Figure R3: Case 1, frictional sliding: (a) numerical and analytical tangential displacement-jump profiles on the finest meshes, with 28 056 continuum elements; (b) RMSE as a function of the number of free DOFs for the three fracture inclinations.

The RMSE decreases with refinement for all three inclinations. On the finest meshes, its values are 0.220, 0.188, and 0.0876 mm for $\beta = 15^\circ$, 30° , and 45° , respectively. Despite the coincident analytical profiles at 15° and 45° , the numerical errors differ. At 45° , the fracture produces a nearly uniform sequence of embedded segments, whereas the other inclinations generate more variable segment lengths and small cut regions within some intersected elements. The largest local discrepancies occur near the fracture tips, where zero jump is not explicitly enforced.

R2.3 Reviewer comment

There is currently no benchmark involving intersecting or coalescing fractures, either for the mechanical or hydro-mechanical formulation. I think at least one such example would be useful to demonstrate how the method behaves when fractures interact.

Author response

We agree that an example involving intersecting or coalescing fractures would be useful to assess the behaviour of the formulation when fractures interact. PorousLab currently cannot simulate fracture coalescence, since fracture propagation is outside the capabilities of the present implementation.

Pre-existing intersecting strong discontinuities present a separate limitation. PorousLab supports multiple discontinuities within an element, provided that each discontinuity fully crosses the element. Their contributions are introduced by superposing the enrichment fields associated with the individual discontinuity segments, using the deformation modes illustrated in Fig. 5. However, this superposition does not, in general, provide all the independent displacement modes required around an intersection. Missing separation modes can impose artificial kinematic constraints and lead to spurious stiffness. Consequently, the ability to introduce multiple discontinuities should not be interpreted as establishing the accuracy of their mechanical response at intersections.

In XFEM, junction enrichment functions and their associated degrees of freedom provide the additional kinematic modes required at fracture intersections (Cruz et al., 2018, 2019; Vahab et al., 2019). Within the Strong Discontinuity Approach, Linder and Armero (2009) developed embedded branching modes for a particular T-shaped configuration. More recently, An et al. (2024) proposed a treatment for two intersecting fractures that avoids redundant traction constraints and prevents interpenetration of the surrounding rock during slip. Therefore, incorporating a general treatment of intersections into the present formulation would require extending its kinematic approximation and the associated equilibrium equations.

We therefore acknowledge that the mechanical and hydro-mechanical behaviour of general fracture intersections is not an established capability of the current implementation. We have clarified this restriction in Section 3.7.2 and the Conclusions.

Changes in the manuscript

1. **Section 3.7.2, “Mechanical behavior”:** We have added the following paragraph at the end of this subsection, before “Fluid flow inside the discontinuity.”

PorousLab supports multiple pre-existing discontinuities within an element, provided that each discontinuity fully crosses the element. The enrichment fields associated with the individual discontinuity segments, Eq. 37, are superposed. However, this approximation does not, in general, provide all the independent displacement modes required around an intersection. Missing separation modes can impose artificial kinematic constraints and lead to spurious stiffness. XFEM formulations commonly address intersection kinematics using junction enrichment functions and associated DOFs (Cruz et al., 2018, 2019; Vahab et al., 2019). Within the Strong Discontinuity Approach, Linder and Armero (2009) developed embedded branching modes for a particular T-shaped configuration. More recently, An et al. (2024) proposed a treatment for two intersecting fractures that avoids redundant traction constraints and prevents interpenetration of the surrounding rock during slip. A general treatment of intersections would require extending the present kinematic approximation and the associated equilibrium equations.

2. **Section 7, “Conclusions”:** We have expanded the limitations paragraph as follows.

The current implementation is intended primarily for transparent formulation development, verification, and small- to medium-scale prototyping studies rather than large-scale production simulations. Its main limitations are the restriction to two-dimensional problems, the absence of fracture propagation, the requirement that embedded discontinuities fully cross the intersected elements, and the limited computational performance. Although multiple pre-existing discontinuities can be represented, accurate mechanical and hydro-mechanical behaviour at general fracture intersections is not an established capability of the current implementation.

R2.4 Reviewer comment

If a fracture passes very close to an element corner, it may create a highly uneven or nearly degenerate split of the element, for example, e.g. 99%/1%. How does the "node-repelling" procedure decide when a fracture is too close to a node? Also, how much is the mesh actually perturbed? A simple sensitivity or convergence test would be helpful to show that the results remain stable as the fracture approaches this limiting situation.

Author response

The node-repelling procedure is an optional geometric preprocessing step controlled by a dimensionless tolerance τ , whose default value is 0.1. A mesh node is selected for adjustment when its distance to a mesh–fracture intersection point is smaller than τh_i , where h_i is the mean characteristic length of its adjacent elements. The characteristic length is $\sqrt{A_e}$ for quadrilateral elements and $\sqrt{2A_e}$ for triangular elements, where A_e is the element area. Intersection points coinciding exactly with a corner of the domain bounding box are excluded from this check.

Each selected node is displaced once per discontinuity pass by τh_i . Interior nodes move along the locally computed fracture normal, while nodes on the sides of an axis-aligned rectangular domain move parallel to the corresponding boundary direction. The intersections are subsequently recomputed using the updated mesh. The criterion is based on distance and does not impose a prescribed minimum ratio between the areas on either side of the fracture. We have replaced the qualitative description in Section 4.1 with an explicit explanation of the selection criterion and perturbation magnitude.

A sensitivity study would provide useful additional information. To limit further expansion of the revised manuscript, which already includes a broad set of numerical examples, we have not added a dedicated study of the near-corner limit. A more detailed investigation could help clarify the roles of the stress state, active deformation modes, and the numerical formulation. We consider this a useful direction for future work.

Changes in the manuscript

Section 4.1, “Model-related classes”: We have replaced the sentence beginning “When necessary, a node-repelling procedure” with the following paragraph.

The optional node-repelling procedure is controlled by a dimensionless tolerance τ , whose default value is 0.1. A mesh node is selected for adjustment when its distance to a mesh–discontinuity intersection point is smaller than τh_i , where h_i is the mean characteristic length of its adjacent elements. The characteristic length is $\sqrt{A_e}$ for quadrilateral elements and $\sqrt{2A_e}$ for triangular elements, where A_e is the element area. Intersection points coinciding exactly with a corner of the domain bounding box are excluded from this check. Each selected node is displaced once per discontinuity pass by τh_i , corresponding to 10% of the local characteristic length at the default tolerance. Interior nodes move along the locally computed discontinuity normal; nodes on the sides of an axis-aligned rectangular domain move parallel to the corresponding boundary direction. The intersections are subsequently recomputed using the updated mesh. The criterion is based on distance and does not impose a prescribed minimum ratio between the areas on either side of the discontinuity.

R2.5 Reviewer comment

Fracture aperture is only tested through the flow-coupled Wijesinghe example. I suggest adding a static Sneddon benchmark (Sneddon, 1946) to verify the mechanical opening response of the fracture independently of the hydraulic coupling.

Author response

We have added a static Sneddon benchmark as Case 2 in Section 6.3, “Sliding and opening of a finite fracture”, to assess the mechanical opening response independently of hydraulic coupling. It uses the geometry, material parameters and discretization described in our response to R2.2, with equal biaxial tensile loading. The normal displacement jump is compared with the analytical solution of Sneddon (1946) for three fracture inclinations and five refinement levels. The normal-jump RMSE decreases with refinement for all three inclinations.

Changes in the manuscript

Section 6.3, Case 2: Mechanical opening. We have added the analytical reference, numerical profiles and refinement study reproduced below. The common geometry, material parameters and discretization are reproduced under R2.2.

Case 2: Mechanical opening

Equal tensile tractions of magnitude $\sigma_0 = 10$ MPa are applied normally to all four boundaries.

For a straight fracture in an infinite isotropic medium under plane-strain conditions, the analytical normal displacement jump is given by Sneddon (1946) as

$$u_d^n(s) = \frac{4(1-\nu^2)}{E} \sigma_0 \sqrt{\left(\frac{L_f}{2}\right)^2 - \left(s - \frac{L_f}{2}\right)^2}, \quad 0 \leq s \leq L_f. \quad (\text{R3})$$

Sneddon’s solution for an internally pressurized fracture also gives the opening of a traction-free fracture under equal biaxial tension. Under this loading, the analytical normal displacement jump is independent of β , and the analytical tangential jump is zero. The maximum analytical opening is 18.75 mm for the parameters considered here. The three inclinations assess the orientation sensitivity of the numerical solution.

The numerical normal-jump profiles on the finest meshes are compared with the common analytical solution in Fig. R4. The refinement study uses the RMSE defined in Eq. (R1), with $q = u_d^n$, evaluated at the same 10 000 uniformly spaced interior positions used in Case 1.

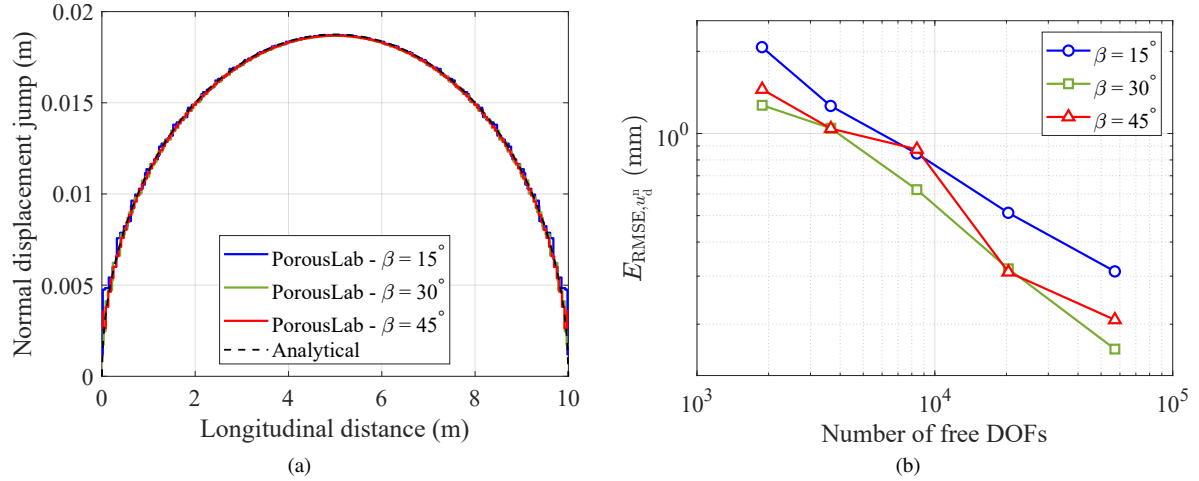


Figure R4: Case 2, mechanical opening: (a) numerical normal displacement-jump profiles on the finest meshes, with 28 056 continuum elements, and the common analytical solution; (b) normal-jump RMSE as a function of the number of free DOFs for the three fracture inclinations.

The numerical profiles closely reproduce the analytical opening over most of the fracture, with the largest local discrepancies occurring near the tips. The normal-jump RMSE decreases with refinement for all three inclinations.

R2.6 Reviewer comment

As I understand it, the current framework does not include MPI parallelization or mesh partitioning. It would be better to state clearly that the present solver is serial. The authors could also briefly discuss what the main challenges would be in extending the framework to parallel computation.

Author response

We have added an explicit statement that the solver is serial at the end of Section 4.3.

The current assembly and state-update procedures require more substantial changes than replacing `for` with `parfor` to support parallelization. Parallelizing the element calculations requires reorganizing how their results are assembled and how the material state is updated. Currently, the assembly loops use counters that advance from one element to the next, so the iterations cannot be directly executed independently using `parfor`. Each iteration would instead need to store its element matrices and vectors separately, with these contributions assembled afterwards. A second difficulty concerns the stresses and internal variables stored in element and integration-point objects. These quantities are updated during element calculations, but changes made to handle objects on parallel workers are not automatically returned to the main MATLAB process. The parallel implementation would therefore need to return the updated stresses and internal variables from each worker and store them in the corresponding element and integration-point objects in the main MATLAB process.

Changes in the manuscript

Section 4.3: We have added the following paragraph at the end of this section.

The present solver is serial. Parallelization of the element calculations would require reorganizing the assembly routines and explicitly collecting the updated constitutive states from parallel workers.

R2.7 Reviewer comment

No wall-clock time or number of degrees of freedom (DOFs) is reported for the numerical examples. It would be useful to provide these quantities at least for the most computationally demanding case, for example the H2M example. If possible, some comparison with other software used for similar problems would also help readers understand the computational performance of PorousLab.

Author response

We have added a “Computational cost” subsection at the end of Section 6. Table 7 reports element counts, free DOFs, total global nonlinear iterations, and mean wall-clock times for all 18 configurations (reproduced as Table R2 below). Times are the arithmetic mean of three runs for each configuration, measured using MATLAB R2022a Update 3 with at most four computational threads on an Intel Core i7-11390H computer with 16 GB of RAM.

The largest mean wall-clock time is 1138.80 s for the Wijesinghe problem, with 3349 free DOFs during injection and 4036 global nonlinear iterations over the full analysis. The two-phase hydro-mechanical Liakopoulos problem requires 15.22 s and 516 iterations with 598 free DOFs.

A systematic cross-software performance benchmark is beyond the scope of this study, which focuses on the architecture, implemented formulations, and verification of PorousLab. The reported computational costs provide a reference for the resources needed to reproduce the presented examples.

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.

R2.8 Reviewer comment

Table 1 lists GOLEM as supporting THM coupling only. However, a 2026 paper by De Lucia et al. reports coupling GOLEM with PHREEQC for THMC applications. Please double-check whether the classification in Table 1 is still up to date.

Author response

We have updated GOLEM's physics entry to "THMC" and added De Lucia et al. (2026). Note a below Table 1 identifies the PHREEQC extension and its reported source availability upon registration.

Changes in the manuscript

1. **Table 1, GOLEM row:** We have replaced "THM" in the Physics column with the following entry.

THMC

2. **Table 1, note a:** We have added the following qualification of the GOLEM extension.

^a GOLEM's chemical processes are provided through its PHREEQC extension; the authors report source availability upon registration (De Lucia et al., 2026).

R2.9 Reviewer comment

The H2M formulation is verified, but there is no simple single-phase HM benchmark, such as a Terzaghi consolidation problem. I think this would be useful for readers who are interested in hydro-mechanical coupling but do not need two-phase flow. This could also be placed in an appendix if the authors want to keep the main text shorter.

Author response

We have added the Terzaghi one-dimensional consolidation problem in Appendix D to provide a simple verification of the single-phase hydro-mechanical continuum formulation.

The numerical pore-pressure and vertical-displacement profiles are compared with the analytical solution (Verruijt, 2016) at 10, 100, 200, 300, 400, and 500 s after load application. The agreement supports verification of pore-pressure dissipation and the associated deformation. For a quantitative comparison, we report the root mean squared error (RMSE) separately for both fields at each comparison time, using Eq. (R1) and evaluating both solutions at the same 51 mesh nodes along the left boundary.

We have also added a cross-reference in Section 6.4 to direct readers to this benchmark.

Changes in the manuscript

1. **Appendix D, "Terzaghi consolidation problem":** We have added the problem description, material parameters, discretization, and solution procedure, together with a figure comparing the numerical and analytical profiles at six times and a table reporting the corresponding RMSE values. The added figure is reproduced as Fig. R5 below.

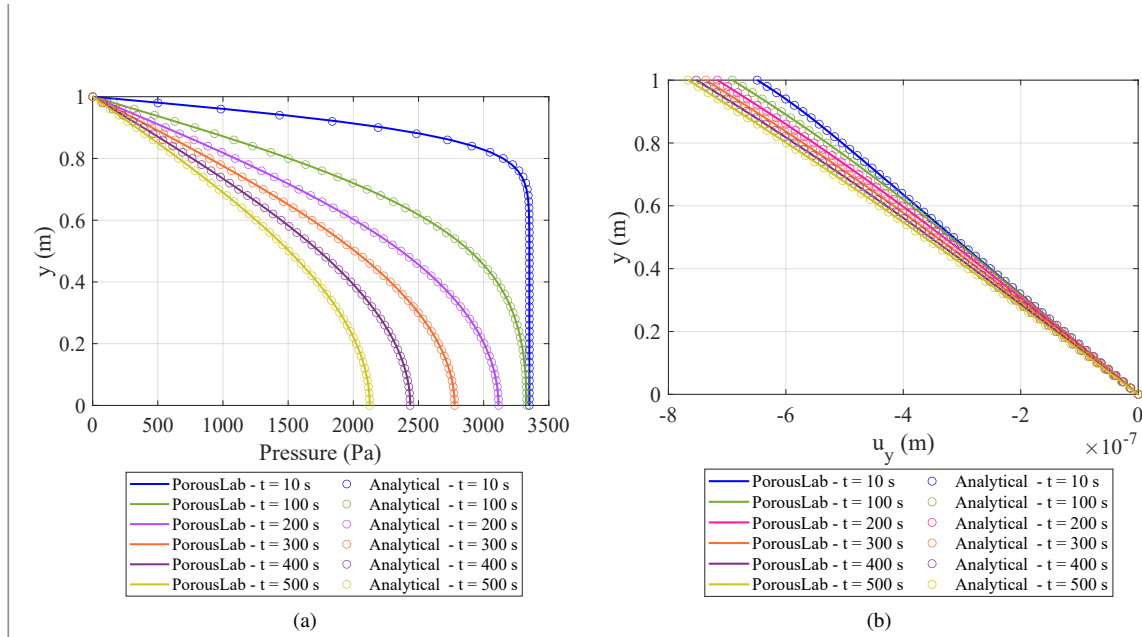


Figure R5: Terzaghi consolidation problem: profiles of (a) pore pressure and (b) vertical displacement at 10, 100, 200, 300, 400, and 500 s. Solid lines represent PorousLab results and circles represent the analytical solution.

- Section 6.4, “Fluid injection into a single fracture”:** We have added the following sentence to the opening paragraph.

The single-phase hydro-mechanical continuum formulation is verified separately using the Terzaghi consolidation problem described in Appendix D.

R2.10 Reviewer comment

Some parts of the manuscript contain fairly general textbook material, for example the discussion of arc-length and control methods. I suggest moving some of this material to an appendix and keeping Section 3 more focused on the parts that are specific to PorousLab.

Author response

We have moved the incremental-iterative derivation, the control-constraint equations and parameter table, and the load-increment adjustment formula to Appendix C. Section 3.8 retains the basic formulations and an overview of the solution procedures implemented in PorousLab, as these provide the mathematical context for the analysis-related classes described in Section 4.3. We have also made the connection between the continuation constraints and the ControlMethod hierarchy explicit.

Changes in the manuscript

- Section 3.8.2, “Nonlinear quasi-static analysis”:** We have retained the equilibrium residual and replaced the detailed continuation-method development with the following overview, including an explicit connection to the analysis-related classes in Section 4.3.

In purely mechanical problems with nonlinear behavior and time-independent conditions, a nonlinear quasi-static analysis is required. It aims to minimize the residual between the unbalanced internal, $\mathbf{f}_i$, and external, $\mathbf{f}_e$, force vectors, expressed as

$$\mathbf{r}(\mathbf{x}, \lambda) = \lambda \mathbf{f}_e - \mathbf{f}_i(\mathbf{x}), \quad (\text{R4})$$

where λ is a load factor. PorousLab solves this equilibrium problem using an incremental-iterative procedure in which both the unknown DOFs and the load factor are updated. A continuation constraint determines the load increment at each iteration. The implemented control methods include load, work, generalized displacement, minimum-norm, orthogonal-residual, and arc-length-type controls. The sep-

aration between the equilibrium solution procedure and the continuation constraint is reflected in the `ControlMethod` hierarchy described in Section 4.3.

Appendix C presents the incremental-iterative derivation, the control-constraint equations and parameters, and the load-increment adjustment formula.

2. **Section 3.8.3, “Nonlinear transient analysis”:** We have retained the notation definitions needed for the transient formulation by adding the following sentence after the definition of the time step size.

Here, subscript n denotes the time step, superscript (k) denotes the iteration, and δ denotes an iterative increment.

3. **Appendix C, “Continuation methods for nonlinear quasi-static analysis”:** We have relocated the incremental-iterative updates, the partitioned linear systems, the control-constraint equations and parameter table, and the load-increment adjustment formula from Section 3.8.2 to this new appendix. The relocated equations, table, and references have been retained, and an introductory cross-reference connects the appendix to the equilibrium residual in Section 3.8.2.

References

- Jaewoo An, Igor Shovkun, and Hamdi Tchelepi. Fully implicit scheme for coupled flow and geomechanics with embedded discrete fracture intersection. *Geoenergy Science and Engineering*, 242:213200, 2024. doi: 10.1016/j.geoen.2024.213200.
- Francisco Cruz, Deane Roehl, and Eurípedes do Amaral Vargas, Jr. An XFEM element to model intersections between hydraulic and natural fractures in porous rocks. *International Journal of Rock Mechanics and Mining Sciences*, 112:385–397, 2018. doi: 10.1016/j.ijrmms.2018.10.001.
- Francisco Cruz, Deane Roehl, and Eurípedes do Amaral Vargas, Jr. An XFEM implementation in Abaqus to model intersections between fractures in porous rocks. *Computers and Geotechnics*, 112:135–146, 2019. doi: 10.1016/j.compgeo.2019.04.014.
- Marco De Lucia, Samuele Frigo, Max Lübke, Mauro Cacace, Elena Petrova, Hannes Hofmann, Magdalena Scheck-Wenderoth, and Guido Blöcher. Towards fully coupled Thermo-Hydro-Mechanical-Chemical (THMC) modelling in advanced reservoir engineering: GOLEM-PHREEQC. *Advances in Geosciences*, 67:117–127, 2026. doi: 10.5194/adgeo-67-117-2026. URL <https://adgeo.copernicus.org/articles/67/117/2026/>.
- Christian Linder and Francisco Armero. Finite elements with embedded branching. *Finite Elements in Analysis and Design*, 45:280–293, 2009. doi: 10.1016/j.finel.2008.10.012.
- I. N. Sneddon. The distribution of stress in the neighbourhood of a crack in an elastic solid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 187(1009):229–260, 1946. doi: 10.1098/rspa.1946.0077.
- M. Vahab, A. R. Khoei, and N. Khalili. An X-FEM technique in modeling hydro-fracture interaction with naturally-cemented faults. *Engineering Fracture Mechanics*, 212:269–290, 2019. doi: 10.1016/j.engfracmech.2019.03.020.