https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


# `PorousLab` 1.1: a modular finite element framework for simulation of fractured porous media

Danilo Cavalcanti[1,2,3], Rafael L. Rangel[4], Carlos A. Mendes[3], Sebastia Olivella[1,2], Victor Vilarrasa[5], Luiz F. Martha[3], Deane Roehl[3], Ignasi de-Pouplana[1,2], and Guillermo Casas[1]

[1]Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Barcelona, Spain
[2]Universitat Politècnica de Catalunya - BarcelonaTech (UPC), Barcelona, Spain
[3]Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil
[4]University of Twente, Enschede, The Netherlands
[5]Global Change Research Group (GCRG), IMEDEA, CSIC-UIB, Esporles, Spain

**Correspondence:** Danilo Cavalcanti (dborges@cimne.upc.edu) and Victor Vilarrasa (victor.vilarrasa@csic.es )

**Abstract.** Numerical modeling of fractured porous media requires the combined treatment of coupled processes, nonlinear geomechanical behavior, and strong discontinuities such as fractures and faults. Although several open-source simulators are available for porous-media applications, many are oriented toward large-scale production analyses and rely on mixed discretizations or complex software architectures. This paper presents `PorousLab`, an open-source, object-oriented finite-element framework implemented in MATLAB for transparent formulation development, verification, and rapid prototyping of coupled porous-media models. The current release supports mechanical, single-phase flow, two-phase flow, hydro-mechanical single-phase, and hydro-mechanical two-phase formulations. Nonlinear geomechanical analyses are supported through different constitutive models and solution procedures, including quasi-static continuation schemes and nonlinear transient analyses. Pre-existing fractures and faults can be represented independently of the background mesh using recent embedded finite-element formulations, which simplify model generation by avoiding the need for mesh conformity with discontinuity geometries. The framework stands out for integrating these features within a single FEM-based discretization setting. Furthermore, the code architecture is developed around abstract OOP classes designed to support modularity and extensibility. Verification examples covering the implemented features are presented and compared with reference solutions.

## 1 Introduction

Numerical simulation supports the design, operation, and monitoring of subsurface engineering applications involving geological media. It is used, for instance, to assess $CO_2$ storage (Rutqvist, 2012; Rutqvist et al., 2014; Vilarrasa et al., 2019; Song et al., 2023; Li and Laloui, 2017; Rutqvist et al., 2016; Ajayi et al., 2019), flow and transport in fractured porous media (Mejia et al., 2021; Rueda et al., 2021; Cavalcanti et al., 2024a; Berre et al., 2019), geothermal operations (Kivi et al., 2025; Tangirala et al., 2024; White et al., 2018), nuclear-waste isolation (Birkholzer et al., 2019; Alonso et al., 2024; Vu et al., 2025a, b), dam stability (Arroyo and Gens, 2021; Karthik et al., 2022; Oberhollenzer et al., 2018), and salt-cavern integrity (Firme et al., 2023; Dias et al., 2023; Firme et al., 2024). These applications are often governed by coupled mechanical, hydraulic, thermal, and

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


chemical processes (Helmig et al., 1997; Rutqvist et al., 2002; Kolditz et al., 2016; Zareidarmiyan et al., 2020; Ogata et al., 2022; Pitz et al., 2023). Their complexity is further increased by the presence of strong discontinuities such as fractures and faults, which create anisotropic, discontinuous pathways that localize flow and deformation (Vaezi et al., 2025).

These discontinuities can be represented in the numerical model with different levels of accuracy and computational cost (Cervera et al., 2022; Vaezi et al., 2024). The most simple and computationally efficient strategy is to use equivalent-continuum or multi-continuum models, in which their effect is incorporated through effective properties or transfer functions (Barenblatt et al., 1960; Warren and Root, 1963; Sánchez-Vila et al., 1995; Wu and Qin, 2009; Renard and Ababou, 2022). However, the applicability of these approaches is conditioned by the existence of a representative elementary volume or by assumptions on the fracture–matrix exchange process (Berre et al., 2019; Berkowitz, 2002). At the opposite end, discrete-fracture and mixed-dimensional models explicitly represent discontinuities within the porous medium (Segura and Carol, 2004; Watanabe et al., 2012; Cerfontaine et al., 2015). This explicit representation allows preferential flow paths, hydraulic barriers, and fracture–matrix exchange to be described more directly (Segura and Carol, 2008; Manzoli et al., 2019; Fabbri et al., 2023). However, it usually requires the computational grid to conform to the discontinuity geometry, making mesh generation a major difficulty for complex fracture networks (Jing, 2003; Berre et al., 2019; Raguenel et al., 2025).

To reduce the meshing constraints, non-conforming approaches have been proposed in both finite volume and finite element contexts, allowing discontinuities to be introduced independently of the porous matrix discretization. The finite volume method (FVM) is widely adopted for flow and heat transfer problems due to its local conservation properties. In this context, standard embedded discrete fracture model formulations are mainly suited to conductive fractures, while projection-based or modified formulations extend the method to low-permeability barriers (Fumagalli et al., 2016, 2019; Ţene et al., 2017; Rao et al., 2020; Olorode et al., 2020; HosseiniMehr et al., 2022; Wong et al., 2021). For problems involving geomechanics, the finite element method (FEM) is commonly adopted, either for the entire coupled problem or combined with FVM for the flow/heat equations (Grunwald et al., 2022; Olivella et al., 1996; Cusini et al., 2021). In the FEM context, the generalized/extended finite element method (GFEM/XFEM) incorporate discontinuities by enriching the primary fields through a partition-of-unity framework (Babuška and Melenk, 1997; Duarte and Oden, 1996; Moës et al., 1999; Belytschko et al., 2001; Dolbow et al., 2000), and have been applied to coupled multiphysics problems (Hirmand et al., 2015; Khoei et al., 2015; Hosseini and Khoei, 2020). Embedded finite element formulations follow a similar enrichment strategy, but introduce the discontinuity effects through element-level enhanced modes, which may allow static condensation and a less intrusive implementation in standard FEM frameworks (Simo and Rifai, 1990; Simo et al., 1993; Wu, 2011; Cavalcanti et al., 2024a, b, 2026).

These modeling choices are reflected in the design of existing porous-media simulators, which differ in the physical processes considered, the treatment of fractures and faults, and the discretization adopted for each governing equation. Table 1 presents representative open-source frameworks with multiphysics capabilities for modeling porous media. The comparison is restricted to frameworks with publicly available source code and with at least one documented strategy for strong discontinuity modeling in their current versions. Therefore, commercial software such as COMSOL Multiphysics (COMSOL AB, 2026) and Abaqus (Dassault Systèmes, 2026), free-to-use but non-open-source multiphysics frameworks such as GeMA (Mendes et al.,

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


**Table 1.** Representative open-source frameworks for multiphysics modeling of geological systems (`M` = mechanical, `H` = hydraulic, `T` = thermal, `C` = chemical).

| Software and references | Platform | Physics | Multiphase flow | Bulk behavior | PDE scheme |
|---|---|---|---|---|---|
| OpenGeoSys (Kolditz et al., 2012) | C++ | THMC | ✓ | General | FEM |
| PorousFlow (Wilkins et al., 2020; Giudicelli et al., 2024) | C++ | THMC | ✓ | General | FEM |
| FALCON (Podgorney et al., 2021; Giudicelli et al., 2024) | C++ | THMC | ✓ | General | FEM |
| GOLEM (Cacace and Jacquey, 2017; Giudicelli et al., 2024) | C++ | THM | – | General | FEM |
| Kratos Multiphysics (Dadvand et al., 2010; de Pouplana and Oñate, 2018, 2017) | C++/Python | HM | – | General | FEM |
| OPM Flow (Rasmussen et al., 2021) | C++/Python | HT | ✓ | - | FVM |
| open-DARTS (Voskov et al., 2024) | C++/Python | THMC | ✓ | Linear Elastic | FVM |
| FEHM (Zyvoloski et al., 1988) | Fortran | THMC | ✓ | General | Hybrid |
| GEOS (Settgast et al., 2024) | C++/Python | THM | ✓ | General | Hybrid |
| DuMux (Flemisch et al., 2011; Koch et al., 2021) | C++/Python | THMC | ✓ | Linear Elastic | Hybrid |
| PFLOTRAN (Hammond et al., 2014; PFLOTRAN) | Fortran | THMC | ✓ | Linear Elastic | Hybrid |
| Flow123d (Březina and Stebel, 2016) | C++/Python | THM | – | Linear Elastic | Hybrid |
| PorePy (Keilegavlen et al., 2021) | Python | THM | ✓ | Linear Elastic | Hybrid |
| MRST (Lie et al., 2012; Krogstad et al., 2015; Lie, 2019) | MATLAB | THMC | ✓ | Linear Elastic | Hybrid |
| `PorousLab` | MATLAB | HM | ✓ | General | FEM |

2016) and CODE_BRIGHT (Olivella et al., 1996), and open-source codes without documented fracture-modeling capabilities such as GeomechX (Yoo and Min, 2025) and JutulDarcy.jl (Møyner, 2024), are not included.

The table summarizes the development platform, supported physics, multiphase-flow capabilities, geomechanical bulk behavior, and main PDE discretization strategy of each framework. For geomechanics, we distinguish between codes limited to linear elasticity and those supporting more general constitutive behavior, such as plasticity or damage. Frameworks adopting more than one discretization scheme, for instance FVM for flow and heat transfer and FEM for mechanics, are classified as hybrid.

As shown in Table 1, several open-source frameworks cover broad THMC physics. In addition, many of them provide advanced capabilities for large-scale applications and benefit from active communities and documentation. However, the comparison also shows that frameworks combining broad multiphysics capabilities with general geomechanical behavior are often performance-oriented, relying on C++ or Fortran implementations and architectures optimized for scalability and parallel execution. Although these choices are advantageous for large-scale simulations, they can make small modifications more dependent on build systems, low-level implementation details, and nontrivial framework overhead. Besides, hybrid discretization schemes are common, requiring proficiency in multiple numerical methods. Exceptions to these limitations are MRST and PorePy, which offer higher-level environments suited to rapid prototyping and lower entry costs. However, their documented

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


geomechanical capabilities are more limited in scope, particularly with respect to nonlinear constitutive behavior and nonlinear geomechanical analysis procedures, and their coupled formulations with mechanics rely on more than one numerical discretization strategy.

To address this gap, this paper presents `PorousLab`, an open-source, object-oriented FEM framework written in MATLAB for the development, verification, and rapid prototyping of porous-media formulations. The framework is designed around a single FEM-based discretization setting and currently covers mechanical, hydraulic, and coupled hydro-mechanical formulations in deformable porous media, including single- and two-phase flow problems. In contrast to prototyping environments mainly centered on flow and transport, `PorousLab` gives particular emphasis to geomechanics by including different mechanical constitutive models and nonlinear analysis procedures, including transient analyses and quasi-static analyses with continuation schemes. The contribution of this work lies in integrating coupled porous-media formulations, nonlinear geomechanical analyses, and embedded strong-discontinuity modeling within a transparent, open-source architecture based on a single FEM discretization. This integration is supported by an object-oriented class structure that separates model definition, material behavior, element-level formulations, and solution procedures, facilitating code readability, reuse, and extension. The resulting framework provides a self-contained scripted workflow with built-in pre- and post-processing, reducing the need for external tools and facilitating model implementation, verification, and prototyping. Beyond these features, `PorousLab` also provides an accessible framework for computational geomechanics researchers interested in understanding additional physical processes, solution schemes, and implementation aspects within a FEM-based setting.

With respect to fracture and fault representation, the frameworks listed in Table 1 follow different strategies, and the availability of each strategy is not always uniform across all the physics supported by a given code. The most common approach is some form of explicit fracture representation, including discrete-fracture, mixed-dimensional, or lower-dimensional fracture models, as documented in different forms for OpenGeoSys, PorousFlow, FALCON, GOLEM, Kratos Multiphysics, open-DARTS, GEOS, DuMux, Flow123d, PorePy, and MRST. Some frameworks also provide alternative representations, including multi-continuum models, embedded-fracture formulations, discrete-fracture-network models, and phase-field fracture formulations. For modeling pre-existing strong discontinuities in the currently supported physics, `PorousLab` adopts an Embedded Finite Element Method (E-FEM) (Cavalcanti et al., 2024a, b). This choice allows discontinuities to be introduced independently of the background finite element mesh, simplifying model generation and avoiding the need for mesh conformity with fracture or fault geometries.

The paper is structured as follows. Section 2 presents the framework modeling and simulation capabilities, and the main simplifying assumptions. Section 3 explains the theoretical concepts behind the modeling and simulation procedures, including the PDE in strong and weak forms, the finite element discretization, and the key aspects of embedded discontinuities. Section 4 describes the framework's implementation structure by detailing its class diagrams. Section 5 presents a brief description of the general simulation workflow. Section 6 provides validation examples, compared against the literature, to demonstrate both the capabilities and the accuracy of the framework. Finally, Section 7 presents concluding remarks, discusses current limitations, and suggests directions for future developments.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


**Table 2.** Available physical models, primary variables, balance equations, and analysis types in `PorousLab`.

| | | | Balance equations | | | Analysis types | | |
|---|---|---|---|---|---|---|---|---|
| Physics types | Tag | Primary variables | Linear momentum | Liquid phase mass | Gas phase mass | Linear | Nonlinear quasi-static | Nonlinear transient |
| Mechanical | `M` | $\mathbf{u}$ | ✓ | | | ✓ | ✓ | ✓ |
| Single-phase flow | `H` | $p_\mathrm{l}$ | | ✓ | | ✓ | | ✓ |
| Two-phase flow | `H2` | $p_\mathrm{l}$, $p_\mathrm{g}$ | | ✓ | ✓ | | | ✓ |
| Hydro-mechanical (single-phase) | `HM` | $\mathbf{u}$, $p_\mathrm{l}$ | ✓ | ✓ | | | | ✓ |
| Hydro-mechanical (two-phase) | `H2M` | $\mathbf{u}$, $p_\mathrm{l}$, $p_\mathrm{g}$ | ✓ | ✓ | ✓ | | | ✓ |

## 2 Framework scope

This section describes the modeling assumptions and simulation capabilities of the current `PorousLab` implementation. The framework is organized around a set of governing equations, each corresponding to a particular physical model, referred to here as physics types. These physics types share the same finite-element modeling structure, while differing in the primary variables, governing equations, constitutive relationships, and analysis procedures required by each problem.

The current release includes five physics types, summarized in Table 2—mechanical (`M`), single-phase flow (`H`), two-phase flow (`H2`), hydro-mechanical single-phase flow (`HM`), and hydro-mechanical two-phase flow (`H2M`)—together with their primary variables, balance equations, and possible analysis procedures. The mechanical component solves the linear momentum balance using the displacement field, $\mathbf{u}$, as the primary variable, whereas the fluid-flow components solve phase mass balance equations using pore-pressure fields as primary variables. In single-phase flow, only the liquid pressure field, $p_\mathrm{l}$, is considered, while in two-phase formulations, both the liquid and gas pressure fields, $p_\mathrm{l}$ and $p_\mathrm{g}$, are used. Linear analysis is available for mechanical and single-phase hydraulic problems. Nonlinear quasi-static analysis is available for mechanical problems and include several continuation strategies, such as load, work, and arc-length control. All implemented physics types support nonlinear transient analysis, solved using either Newton–Raphson or Picard iterations.

The current implementation is restricted to two-dimensional problems. Plane-strain and plane-stress assumptions are available for mechanical and hydro-mechanical models, while axisymmetric conditions are supported for all physics types. The spatial discretization is based on isoparametric finite elements, including triangular and quadrilateral elements with linear or quadratic interpolation: constant-strain triangles (CST), linear-strain triangles (LST), four-node quadrilaterals (ISOQ4), and eight-node quadrilaterals (ISOQ8). For mechanical physics types, the implemented constitutive models include linear elasticity, isotropic damage, and elastoplastic laws.

Pre-existing fractures and faults can be represented through an Embedded Finite Element Method (E-FEM). In this approach, discontinuities are introduced independently of the background mesh and are approximated by polylines whose segments cross finite elements. This capability is currently available for the `M`, `H`, and `HM` physics types. The present implementation does not address fracture propagation, and embedded discontinuities are required to fully cross the intersected finite elements.

The current implementation is based on the following modeling assumptions:

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


- **Kinematics:** small displacements and strains.
- **Dynamics:** quasi–static conditions with inertial forces neglected.
- **Fluid system:** two-phase, immiscible, single-component dispersed fluids with no interphase mass transfer (i.e., no dissolution or evaporation).
- **Embedded discontinuity geometry:** the discontinuity surface is smooth, and only the ones that fully cross a finite element are considered.

The current release is distributed as an open-source project under the MIT licence and requires only base MATLAB, with no additional toolbox dependencies. It has been validated on MATLAB R2022a and later versions, and is not compatible with GNU Octave. 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; at present, this interface is available only for mechanical analyses.

## 3 Theoretical background

This section begins by presenting the FEM formulation of the hydro-mechanical model with two-phase fluid flow (`H2M`) (Schrefler and Scotta, 2001), as it represents the most general physics currently implemented. The FEM formulations for the other physics can mostly be derived as specific cases of this general physical model. Subsequently, the E-FEM is formulated for modeling embedded strong discontinuities in the physics where it is available (`M`, `H`, and `HM`) (Cavalcanti et al., 2024a, b). Lastly, it introduces the formulation behind the different analysis types supported by `PorousLab`.

### 3.1 General physical model

We consider a deformable porous medium composed of two fluid phases: liquid (l) and gas (g), as shown in Fig. 1.

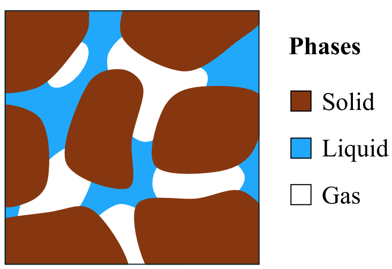



**Figure 1.** Representative elementary volume of a porous material with two fluid phases

Within the representative elementary volume, the average bulk density is

$$\rho = (1-\phi)\,\rho_{\mathrm{s}} + \phi\, S_{\mathrm{l}}\, \rho_{\mathrm{l}} + \phi\, S_{\mathrm{g}}\, \rho_{\mathrm{g}}, \tag{1}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


where $\phi$ is the porosity; $\rho_\mathrm{s}$, $\rho_\mathrm{l}$, and $\rho_\mathrm{g}$ are the densities of the solid, liquid, and gas phases, respectively; and $S_\mathrm{l}$ and $S_\mathrm{g}$ are the corresponding liquid and gas phase saturation, respectively.

The liquid saturation is governed by the capillary pressure, $p_\mathrm{c}$, defined as

$$p_\mathrm{c} = p_\mathrm{g} - p_\mathrm{l}, \tag{2}$$

with $p_\mathrm{g}$ and $p_\mathrm{l}$ being the gas and liquid pressures, respectively. The saturation condition for the two fluid phases imposes that

$$S_\mathrm{l} + S_\mathrm{g} = 1. \tag{3}$$

### 3.2 Linear momentum balance equation

By neglecting inertial forces, the linear momentum balance is given by

$$\nabla \cdot \boldsymbol{\sigma} + \rho\,\mathbf{g} = \mathbf{0}, \tag{4}$$

where $\mathbf{g}$ is the gravitational acceleration vector and $\boldsymbol{\sigma}$ is the total Cauchy stress tensor.

The stress-strain relationship is described in terms of the effective stress tensor, $\boldsymbol{\sigma}'$, which, adopting a Solid Mechanics convention (compression is negative), is expressed as

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} + \alpha\, p_\mathrm{s}\,\mathbf{I}, \tag{5}$$

where $\mathbf{I}$ denotes the identity tensor, $\alpha$ is Biot's coefficient, and $p_\mathrm{s}$ is the pore-pressure acting at the solid skeleton, taken as

$$p_\mathrm{s} = S_\mathrm{l}\, p_\mathrm{l} + S_\mathrm{g}\, p_\mathrm{g}. \tag{6}$$

The effective stress evolves according to the mechanical constitutive law, expressed in rate form as

$$\dot{\boldsymbol{\sigma}}' = \mathbf{D}\,\dot{\boldsymbol{\varepsilon}}, \tag{7}$$

where $\mathbf{D}$ is the tangent constitutive tensor and $\boldsymbol{\varepsilon}$ is the strain tensor, with $\dot{(\,)}$ indicating the rate of change. Assuming small displacements and strains, the strain rate is given by

$$\dot{\boldsymbol{\varepsilon}} = \nabla^\mathrm{s}\dot{\mathbf{u}}, \tag{8}$$

where $\mathbf{u}$ is the displacement field vector and $\nabla$ denotes the gradient operator, with superscript s indicating the symmetric part.

This equation must be supplemented with appropriate boundary conditions to define a well-posed problem. Let $\Gamma = \partial\Omega$ denote the boundary of the domain, which is partitioned into two disjoint subsets $\Gamma_\mathrm{u}$ and $\Gamma_\mathrm{t}$, such that $\Gamma = \Gamma_\mathrm{u} \cup \Gamma_\mathrm{t}$ and $\Gamma_\mathrm{u} \cap \Gamma_\mathrm{t} = \emptyset$. On the Dirichlet boundary, $\Gamma_\mathrm{u}$, the displacement field is prescribed as

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_\mathrm{u}, \tag{9}$$

where $\bar{\mathbf{u}}$ is the imposed displacement vector. On the Neumann boundary, $\Gamma_\mathrm{t}$, the traction vector is prescribed as

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_\mathrm{t}, \tag{10}$$

where $\bar{\mathbf{t}}$ is the applied traction vector and $\mathbf{n}$ is the outward unit normal to the boundary.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


### 3.3 Solid mass balance equation

For the solid matrix, the mass conservation is expressed as

$$\frac{\partial\left(\rho_{\mathrm{s}}\left(1-\phi\right)\right)}{\partial t}+\nabla\cdot\left(\rho_{\mathrm{s}}\left(1-\phi\right)\dot{\mathbf{u}}\right)=0. \tag{11}$$

By expanding this equation, the porosity rate of change can be defined as

$$\frac{D_{\mathrm{s}}\phi}{Dt}=\left(1-\phi\right)\left(\nabla\cdot\dot{\mathbf{u}}+\frac{1}{\rho_{\mathrm{s}}}\frac{D_{\mathrm{s}}\rho_{\mathrm{s}}}{Dt}\right), \tag{12}$$

where

$$\frac{D_{\mathrm{s}}\left(\cdot\right)}{Dt}=\frac{\partial\left(\cdot\right)}{\partial t}+\nabla\left(\cdot\right)\cdot\dot{\mathbf{u}}, \tag{13}$$

is the material derivative with respect to the solid phase.

The material derivative of the solid density is (Schrefler et al., 1990)

$$\frac{1}{\rho_{\mathrm{s}}}\frac{D_{\mathrm{s}}\rho_{\mathrm{s}}}{Dt}=\frac{1}{1-\phi}\left(\frac{\alpha-\phi}{K_{\mathrm{s}}}\frac{D_{\mathrm{s}}p_{\mathrm{s}}}{Dt}-\left(1-\alpha\right)\nabla\cdot\dot{\mathbf{u}}\right), \tag{14}$$

with $K_{\mathrm{s}}$ being the bulk modulus of the solid matrix.

Under small strains, we may approximate the material derivative with respect to the solid with its Eulerian form (Olivella et al., 1996)

$$\frac{D_{\mathrm{s}}\left(\cdot\right)}{Dt}\approx\frac{\partial\left(\cdot\right)}{\partial t}. \tag{15}$$

### 3.4 Fluid mass balance equations

Assuming that each fluid phase, $\pi\in\{\mathrm{l},\mathrm{g}\}$, consists of a single component, where l and g stand for liquid and gas, repectively, the mass conservation equation for each fluid phase is written as

$$\frac{\partial\left(\rho_{\pi}\,\phi\,S_{\pi}\right)}{\partial t}+\nabla\cdot\left(\rho_{\pi}\,\phi\,S_{\pi}\,\mathbf{v}_{\pi}\right)=0, \tag{16}$$

where $\mathbf{v}_{\pi}$ is the absolute velocity vector of phase $\pi$, which is decomposed into a relative velocity with respect to the solid matrix, $\mathbf{v}_{\pi\mathrm{s}}$, and the solid motion itself:

$$\mathbf{v}_{\pi}=\mathbf{v}_{\pi\mathrm{s}}+\dot{\mathbf{u}}, \tag{17}$$

Additionally, the advective flux of fluid phase $\pi$, defined per unit total cross-sectional area, is given by

$$\mathbf{q}_{\pi}=\phi\,S_{\pi}\,\mathbf{v}_{\pi\mathrm{s}}. \tag{18}$$

By expanding Eq. (16), incorporating Eqs. (12), (14), (15), (17) and (18), and dividing by $\rho_{\pi}$, we obtain,

$$\frac{\phi}{\rho_{\pi}}\frac{\partial\left(\rho_{\pi}\,S_{\pi}\right)}{\partial t}+S_{\pi}\left(\frac{\alpha-\phi}{K_{\mathrm{s}}}\frac{\partial p_{\mathrm{s}}}{\partial t}\right)+\nabla\cdot\mathbf{q}_{\pi}+\alpha\,S_{\pi}\,\nabla\cdot\dot{\mathbf{u}}=0. \tag{19}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


The advective flux $\mathbf{q}_\pi$ is governed by the generalized Darcy's law as

$$\mathbf{q}_\pi = \frac{k_{\mathrm{r}\pi}}{\mu_\pi}\mathbf{k}\left(-\nabla p_\pi + \rho_\pi \mathbf{g}\right), \tag{20}$$

where $\mathbf{k}$ is the intrinsic permeability tensor, $\mu_\pi$ is the dynamic viscosity of phase $\pi$, and $k_{\mathrm{r}\pi}$ is the relative permeability of phase $\pi$, which is typically a function of the liquid saturation $S_\mathrm{l}$.

By default, the fluids are weakly compressible as

$$\rho_\pi = \rho_{\pi 0}\left(1 + \frac{1}{K_\pi}(p_\pi - p_{\pi 0})\right), \tag{21}$$

where $\rho_{\pi 0}$ is the reference density, $p_{\pi 0}$ is the reference pressure, and $K_\pi$ is the bulk modulus of phase $\pi$.

To ensure a well-posed problem, Eq. (19) must be complemented with appropriate boundary conditions for each fluid phase $\pi \in \{\mathrm{l},\mathrm{g}\}$. The boundary $\Gamma = \partial\Omega$ is partitioned into two disjoint subsets $\Gamma_{p_\pi}$ and $\Gamma_{q_\pi}$, such that $\Gamma = \Gamma_{p_\pi} \cup \Gamma_{q_\pi}$ and $\Gamma_{p_\pi} \cap \Gamma_{q_\pi} = \emptyset$. On the Dirichlet boundary, $\Gamma_{p_\pi}$, the pressure field of fluid phase $\pi$ is prescribed as

$$p_\pi = \bar{p}_\pi \quad \text{on } \Gamma_{p_\pi}, \tag{22}$$

where $\bar{p}_\pi$ denotes the prescribed fluid pressure. On the Neumann boundary, $\Gamma_{q_\pi}$, the normal component of the phase-specific Darcy flux is imposed as

$$\mathbf{q}_\pi \cdot \mathbf{n} = \bar{q}_\pi \quad \text{on } \Gamma_{q_\pi}, \tag{23}$$

where $\bar{q}_\pi$ is the prescribed volumetric flux of phase $\pi$.

### 3.5 Weak form

The weak form of the coupled problem, comprising the linear momentum balance given by Eq. (4) and the fluid mass balances in Eq. (19), is derived using the standard Galerkin method followed by the application of the Divergence Theorem.

For the linear momentum balance, the weak form is expressed as

$$\int_\Omega \nabla^\mathrm{s}\delta\mathbf{u} : \boldsymbol{\sigma}'\,\mathrm{d}\Omega - \int_\Omega (\nabla^\mathrm{s}\delta\mathbf{u} : \mathbf{I})\,\alpha\, S_\mathrm{l}\, p_\mathrm{l}\,\mathrm{d}\Omega - \int_\Omega (\nabla^\mathrm{s}\delta\mathbf{u} : \mathbf{I})\,\alpha\, S_\mathrm{g}\, p_\mathrm{g}\,\mathrm{d}\Omega - \int_{\Gamma_\mathrm{t}} \delta\mathbf{u}\cdot\bar{\mathbf{t}}\,\mathrm{d}\Gamma - \int_\Omega \rho\,\delta\mathbf{u}\cdot\mathbf{g}\,\mathrm{d}\Omega = 0, \tag{24}$$

where $\delta\mathbf{u}$ is the admissible variation of the displacement field.

The weak form of the fluid mass balance equation for each phase, $\pi \in \{\mathrm{l},\mathrm{g}\}$, is expressed as

$$\begin{aligned}
&\int_\Omega \delta p_\pi \frac{\phi}{\rho_\pi}\frac{\partial(\rho_\pi S_\pi)}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \delta p_\pi S_\pi \left(\frac{\alpha-\phi}{K_\mathrm{s}}\frac{\partial p_\mathrm{s}}{\partial t}\right)\mathrm{d}\Omega + \int_{\Gamma_{q_\pi}} \delta p_\pi\,\bar{q}_\pi\,\mathrm{d}\Gamma \\
&+ \int_\Omega \nabla\delta p_\pi \cdot \frac{\mathbf{k}\,k_{\mathrm{r}\pi}}{\mu_\pi}\nabla p_\pi\,\mathrm{d}\Omega - \int_\Omega \nabla\delta p_\pi \cdot \frac{\mathbf{k}\,k_{\mathrm{r}\pi}}{\mu_\pi}\rho_\pi\,\mathbf{g}\,\mathrm{d}\Omega + \int_\Omega \delta p_\pi\,\alpha\, S_\pi \nabla\cdot\dot{\mathbf{u}}\,\mathrm{d}\Omega = 0,
\end{aligned} \tag{25}$$

where $\delta p_\pi$ is the admissible variation of the pressure field of phase $\pi$.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


### 3.6 Finite element discretization

Following an isoparametric finite-element formulation, the approximated displacement field, $\mathbf{u}^h$, is interpolated from nodal values using a standard shape function matrix, $\mathbf{N}_\mathrm{u}$, as

$$\mathbf{u}^h(\mathbf{x},t) = \mathbf{N}_\mathrm{u}(\mathbf{x})\;\mathbf{d}(t), \tag{26}$$

where $\mathbf{d}$ is the vector of nodal displacements degrees of freedom (DOFs). The symmetric part of the displacement field gradient and the divergence of the displacement rate are discretized, respectively, as

$$\nabla^\mathrm{s}\mathbf{u}^h(\mathbf{x},t) = \mathbf{B}_\mathrm{u}(\mathbf{x})\;\mathbf{d}(t), \tag{27}$$

$$\nabla\cdot\dot{\mathbf{u}}^h(\mathbf{x},t) = \nabla^\mathrm{s}\dot{\mathbf{u}}^h(\mathbf{x},t):\mathbf{I} = \mathbf{m}^\top\mathbf{B}_\mathrm{u}(\mathbf{x})\;\dot{\mathbf{d}}(t), \tag{28}$$

where $\mathbf{B}_\mathrm{u}$ is the standard finite element strain-displacement matrix and $\mathbf{m} = [1\;1\;0]^\top$ for bi-dimensional problems.

The pore-pressure fields associated with each fluid phase $\pi \in \{\mathrm{l},\mathrm{g}\}$ are adopted as primary variables for the hydraulic behavior. These fields and their spatial gradients are approximated using standard finite-element interpolation as

$$p_\pi^h(\mathbf{x},t) = \mathbf{N}_\mathrm{p}(\mathbf{x})\;\mathbf{p}_\pi(t), \tag{29}$$

$$\nabla p_\pi^h(\mathbf{x},t) = \mathbf{B}_\mathrm{p}(\mathbf{x})\;\mathbf{p}_\pi(t), \tag{30}$$

where $\mathbf{p}_\pi$ is the vector of nodal pore-pressure DOFs for phase $\pi$, $\mathbf{N}_\mathrm{p}$ is the row vector of shape functions associated with the pore-pressure field, and $\mathbf{B}_\mathrm{p}$ is the matrix containing the gradients of the shape functions in $\mathbf{N}_\mathrm{p}$.

By substituting the spatially-discretized fields of displacements and pore-pressures into Eqs. (24) and (25), we obtain the discretized residual vectors for the `H2M` physical model as

$$\mathbf{r}_\mathrm{u}(t) = \int_\Omega \mathbf{B}_\mathrm{u}^\top\boldsymbol{\sigma}'\,\mathrm{d}\Omega - \int_\Omega \mathbf{B}_\mathrm{u}^\top\mathbf{m}\,\alpha\,S_\mathrm{l}\,\mathbf{N}_\mathrm{p}\,\mathrm{d}\Omega\,\mathbf{p}_\mathrm{l} - \int_\Omega \mathbf{B}_\mathrm{u}^\top\mathbf{m}\,\alpha\,S_\mathrm{g}\,\mathbf{N}_\mathrm{p}\,\mathrm{d}\Omega\,\mathbf{p}_\mathrm{g} - \int_{\Gamma_\mathrm{t}} \mathbf{N}_\mathrm{u}^\top\,\bar{\mathbf{t}}\,\mathrm{d}\Gamma - \int_\Omega \mathbf{N}_\mathrm{u}^\top\,\rho\,\mathbf{g}\,\mathrm{d}\Omega, \tag{31}$$

$$\begin{aligned}\mathbf{r}_\pi(t) = &\int_\Omega \mathbf{N}_\mathrm{p}^\top\,\frac{\phi}{\rho_\pi}\,\frac{\partial(\rho_\pi S_\pi)}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \mathbf{N}_\mathrm{p}^\top\,S_\pi\left(\frac{\alpha-\phi}{K_\mathrm{s}}\frac{\partial p_\mathrm{s}}{\partial t}\right)\mathrm{d}\Omega + \int_{\Gamma_{q_\pi}} \mathbf{N}_\mathrm{p}^\top\,\bar{q}_\pi\,\mathrm{d}\Gamma\\ &+ \int_\Omega \mathbf{B}_\mathrm{p}^\top\,\frac{\mathbf{k}\,k_{\mathrm{r}\pi}}{\mu_\pi}\,\mathbf{B}_\mathrm{p}\,\mathrm{d}\Omega\,\mathbf{p}_\pi - \int_\Omega \mathbf{B}_\mathrm{p}^\top\,\frac{\mathbf{k}\,k_{\mathrm{r}\pi}}{\mu_\pi}\rho_\pi\,\mathbf{g}\,\mathrm{d}\Omega + \int_\Omega \mathbf{N}_\mathrm{p}^\top\,\alpha\,S_\pi\,\mathbf{m}^\top\mathbf{B}_\mathrm{u}\,\mathrm{d}\Omega\,\dot{\mathbf{d}}.\end{aligned} \tag{32}$$

All these integral expressions are computed using Gaussian quadrature in `PorousLab`. Furthermore, Appendix A details additional numerical treatments adopted for the two-phase fluid flow physics (`H2`).

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


### 3.7 Embedded strong discontinuities

#### 3.7.1 Strong Discontinuity Approach

255 Strong discontinuities, such as fractures and faults, are modeled through the Embedded Finite Element Method (E-FEM) following a formulation based on the Strong Discontinuity Approach (Cavalcanti et al., 2024a, b). It considers the problem illustrated in Fig. 2, where the domain is crossed by a discontinuity $\Gamma_\mathrm{d}$ that partitions it into two subdomains, $\Omega^+$ and $\Omega^-$. These subdomains are labelled according to the unit normal vector $\mathbf{n}_\mathrm{d}$ to the discontinuity, which points towards $\Omega^+$. Assuming the discontinuity surface is smooth, the unit normal vectors on both sides satisfy $\mathbf{n}_\mathrm{d}^- = -\mathbf{n}_\mathrm{d}^+ = \mathbf{n}_\mathrm{d}$.

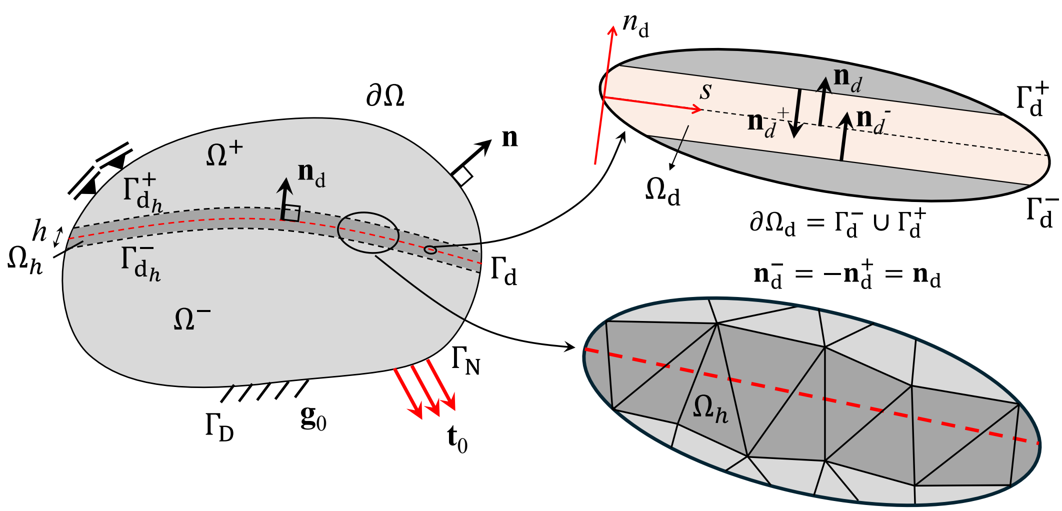



**Figure 2.** Domain with the presence of a strong discontinuity.

260 To implicitly represent a discontinuous arbitrary field across the discontinuity surface within a finite-element framework, a generic field $\mathbf{g}(\mathbf{x})$ defined over the subdomain $\Omega_h$ surrounding the discontinuity (see Fig. 2) is decomposed as

$$\mathbf{g}(\mathbf{x}) = \widehat{\mathbf{g}}(\mathbf{x}) + \bar{\mathbf{g}}(\mathbf{x}), \tag{33}$$

where $\widehat{\mathbf{g}}$ and $\bar{\mathbf{g}}$ are the regular and enhanced parts of the field, respectively. The enhanced part is (Linder and Armero, 2007; Wu, 2011)

265 $$\bar{\mathbf{g}}(\mathbf{x}) = \mathcal{H}_{\Gamma_\mathrm{d}}(\mathbf{x})\widetilde{\mathbf{g}}(\mathbf{x}) - \sum_{i=1}^{n_\mathrm{n}} \mathbf{N}^{(i)}(\mathbf{x})\mathcal{H}_{\Gamma_\mathrm{d}}(\mathbf{x}_i)\widetilde{\mathbf{g}}(\mathbf{x}_i), \tag{34}$$

where $\widetilde{\mathbf{g}}(\mathbf{x})$ is a relative field, $n_\mathrm{n}$ is the number of nodes from the finite element discretization, and $\mathcal{H}_{\Gamma_\mathrm{d}}$ is the Heaviside function defined as

$$\mathcal{H}_{\Gamma_\mathrm{d}}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega^+ \\ 0, & \mathbf{x} \in \Omega^- \end{cases}, \tag{35}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


Figure 3 illustrates the construction of a generic scalar field $g(x)$ in a one-dimensional problem following eqs. (33) and (34).

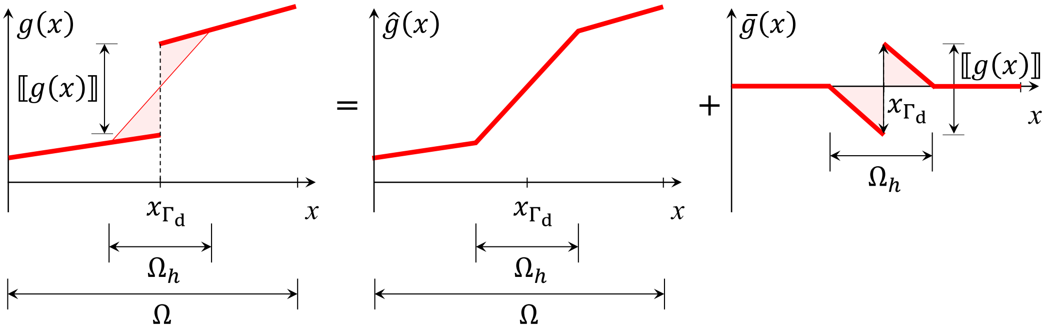



**Figure 3.** Construction of the field composed of a regular and an enhanced part (adapted from Wu (2011)). $[\![g(x)]\!]$ denotes the jump at the discontinuity.

In the present formulation, only discontinuities that fully cross a finite element are considered. Each discontinuity is geometrically characterized within the element by a reference point, $\mathbf{x}_\mathrm{ref}$, located at the midpoint of the segment, and by a pair of orthonormal vectors: the normal direction $\mathbf{n}_\mathrm{d}$ and the tangential direction $\mathbf{m}_\mathrm{d}$, as depicted in Fig. 4.

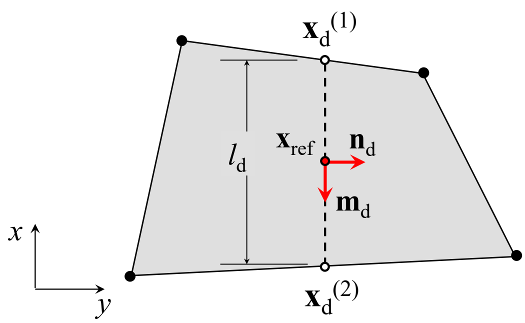



**Figure 4.** Linear quadrilateral finite element with embedded strong discontinuity (dashed line).

### 3.7.2 Mechanical behavior

In the mechanical physical model (physics $\mathtt{M}$), the displacement jump across the discontinuity, expressed in the local coordinate system aligned with the discontinuity, is approximated as

$$\mathbf{u}_\mathrm{d}^h(s) = \begin{bmatrix} 1 & 0 & s & 0 \\ 0 & 1 & 0 & s \end{bmatrix} \begin{bmatrix} \alpha_0^\mathrm{m} \\ \alpha_0^\mathrm{n} \\ \alpha_1^\mathrm{m} \\ \alpha_1^\mathrm{n} \end{bmatrix} = \mathbf{N}_\mathrm{d,u}(s)\,\boldsymbol{\alpha}, \tag{36}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


where $s$ is the local coordinate along the discontinuity. The DOFs $\alpha_0^{\mathrm{m}}$ and $\alpha_0^{\mathrm{n}}$ represent the relative translations in the tangential and normal directions, respectively, while $\alpha_1^{\mathrm{m}}$ and $\alpha_1^{\mathrm{n}}$ represent the linear variations of the displacement jump along the discontinuity, corresponding to relative tangential stretching and relative rotation, respectively. Based on these DOFs, the enhanced
280 part of the displacement field over the continuum can be defined as

$$\tilde{\mathbf{u}}^h(\mathbf{x}) = \alpha_0^{\mathrm{m}}\,\mathbf{m}_{\mathrm{d}} + \alpha_0^{\mathrm{n}}\,\mathbf{n}_{\mathrm{d}} + \alpha_1^{\mathrm{m}}\left(\mathbf{m}_{\mathrm{d}}\,\mathbf{m}_{\mathrm{d}}{}^{\top}\right)\Delta\mathbf{x} + \alpha_1^{\mathrm{n}}\left(\mathbf{n}_{\mathrm{d}}\,\mathbf{m}_{\mathrm{d}}{}^{\top} - \mathbf{m}_{\mathrm{d}}\,\mathbf{n}_{\mathrm{d}}{}^{\top}\right)\Delta\mathbf{x}, \tag{37}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{\mathrm{ref}}$. The additional deformation modes associated with each additional DOF are illustrated in Fig. 5.

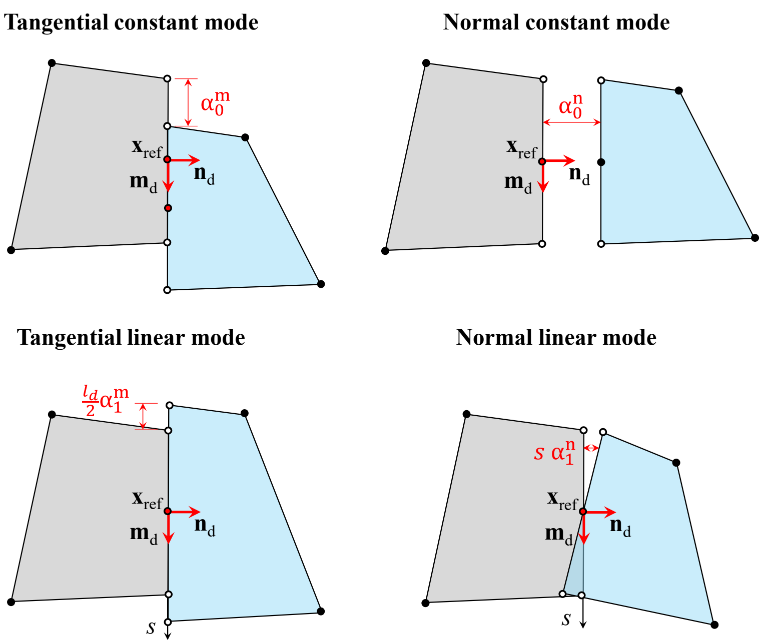



**Figure 5.** Additional deformation modes in a linear quadrilateral finite element with an embedded strong discontinuity.

In Eq. (37), the enriched displacement field is being discretized with local variables which are attached to the discontinuity segment inside each element, and represent the deformation modes illustrated in Fig. 5. These local unknowns are discontinuous
285 across elements and can be condensed at element level. Alternatively, following Dias-da Costa et al. (2009a, b), the nodal variant enforces continuity of the displacement jumps across neighboring cut elements by placing the enrichment DOFs at the mesh nodes where the discontinuity intersects the element edges. Figure 6 compares both choices, and Eq. (38) presents the relationship between them.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

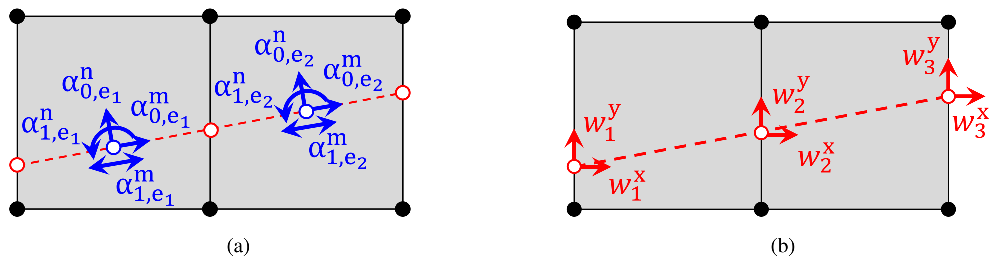




**Figure 6.** Discretizations of the enriched displacement field: (a) local, element-level DOFs; (b) nodal jump DOFs at cut nodes.

$$\begin{Bmatrix} \alpha_0^{\mathrm{m}} \\ \alpha_0^{\mathrm{n}} \\ \alpha_1^{\mathrm{m}} \\ \alpha_1^{\mathrm{n}} \end{Bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{1}{l_d} & 0 & \frac{1}{l_d} & 0 \\ 0 & -\frac{1}{l_d} & 0 & \frac{1}{l_d} \end{bmatrix} \begin{bmatrix} \mathbf{m_d}^\top & \mathbf{0} \\ \mathbf{n_d}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{m_d}^\top \\ \mathbf{0} & \mathbf{n_d}^\top \end{bmatrix} \begin{Bmatrix} w_1^{\mathrm{x}} \\ w_1^{\mathrm{y}} \\ w_2^{\mathrm{x}} \\ w_2^{\mathrm{y}} \end{Bmatrix}. \tag{38}$$

The discretized approximation of the symmetric part of the total displacement field is obtained using Eqs. (33), (34), and (37),

$$\nabla^{\mathrm{s}} \mathbf{u}^h(\mathbf{x}) = \mathbf{B}_{\mathrm{u}}(\mathbf{x})\,\mathbf{d} + \tilde{\mathbf{B}}_{\mathrm{u}}(\mathbf{x})\,\boldsymbol{\alpha}, \tag{39}$$

where,

$$\tilde{\mathbf{B}}(\mathbf{x}) = \mathcal{H}_{\Gamma_{\mathrm{d}}}(\mathbf{x}) \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ (\mathbf{M_d}\,\mathbf{m_d})^\top \\ \mathbf{0} \end{bmatrix}^\top - \sum_{i \in \Omega^+} \begin{bmatrix} \left(\mathbf{B}^{(i)}(\mathbf{x})\mathbf{m_d}\right)^\top \\ \left(\mathbf{B}^{(i)}(\mathbf{x})\mathbf{n_d}\right)^\top \\ \left(\mathbf{B}^{(i)}(\mathbf{x})\,\mathbf{m_d}\,\mathbf{m_d}^\top\left(\mathbf{x}^{(i)} - \mathbf{x}_{\mathrm{ref}}\right)\right)^\top \\ \left(\mathbf{B}^{(i)}(\mathbf{x})\left(\mathbf{n_d}\,\mathbf{m_d}^\top - \mathbf{m_d}\,\mathbf{n_d}^\top\right)\left(\mathbf{x}^{(i)} - \mathbf{x}_{\mathrm{ref}}\right)\right)^\top \end{bmatrix}^\top, \tag{40}$$

and

$$\mathbf{M_d} = \begin{bmatrix} m_{\mathrm{d}}^1 & 0 \\ 0 & m_{\mathrm{d}}^2 \\ m_{\mathrm{d}}^2 & m_{\mathrm{d}}^1 \end{bmatrix}. \tag{41}$$

The equilibrium between the Cauchy stresses in the surrounding continuum and the cohesive tractions acting along the discontinuity is enforced as

$$\int_\Omega \mathbf{G}^\top \boldsymbol{\sigma}^h \, d\Omega + \int_{\Gamma_{\mathrm{d}}} \mathbf{N_{d,u}}^\top \, \mathbf{t}_{\mathrm{d}}^h \, d\Gamma = \mathbf{0}, \tag{42}$$

where $\boldsymbol{\sigma}^h = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}^\top$ is the stress tensor in Voigt notation, $\mathbf{t}_{\mathrm{d}} = \mathbf{t}_{\mathrm{d}}(\mathbf{u}_{\mathrm{d}})$ denotes the cohesive traction vector as a function of the displacement jump, and $\mathbf{G}$ is a stress projection operator (Linder and Armero, 2007).

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


For elements with embedded strong discontinuities, standard Gauss quadrature is sufficient when only rigid-body enrichment modes are active. When the linear tangential mode is included, the enrichment introduces a Heaviside term in Eq. (40), yielding a discontinuous strain field. Consistent integration then requires a sub-cell scheme aligned with the embedded surface. Here, each cut element is partitioned by a Delaunay triangulation into conforming sub-triangles on both sides of the interface, and Gauss points are assigned per sub-cell using the native triangular rule; see Fig. 7.

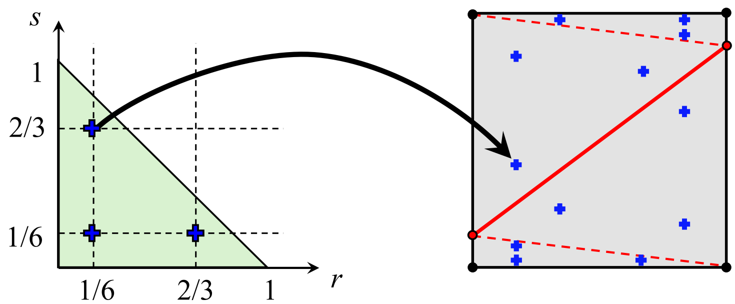



**Figure 7.** Sub-cell integration for an enriched element: the element is split by the discontinuity, and Gauss points are set per sub-triangle cells.

### 3.7.3 Fluid flow inside the discontinuity

The strong form of the fluid continuity equation is

$$\frac{\phi_\mathrm{d}}{\rho_\pi}\frac{\partial(\rho_\pi S_\pi)}{\partial t} + \nabla_\mathrm{d}\cdot\mathbf{q}_{\mathrm{d},\pi} = 0, \tag{43}$$

where $\nabla_\mathrm{d} = [\partial/\partial s \quad \partial/\partial n]^\top$ is the divergence operator defined in the discontinuity local system, and $\mathbf{q}_{\mathrm{d},\pi}$ is the advective flux vector of phase $\pi$ inside the discontinuity, given by:

$$\mathbf{q}_{\mathrm{d},\pi} = \frac{k_{\mathrm{r}\pi}}{\mu_\pi}\mathbf{k}_\mathrm{d}\left(-\nabla_\mathrm{d}\, p_{\mathrm{d},\pi} + \rho_\pi \begin{bmatrix}\mathbf{m}_\mathrm{d}{}^\top\\ \mathbf{n}_\mathrm{d}{}^\top\end{bmatrix}\mathbf{g}\right), \tag{44}$$

where $p_{\mathrm{d},\pi}$ is the pore-pressure of phase $\pi$ inside the discontinuity. Considering that the pressure gradient is negligible in the normal direction, by taking a zero permeability in the normal direction, the advective flux can be simplified to

$$\mathbf{q}_{\mathrm{d},\pi} = k_\mathrm{d}^\mathrm{L}\frac{k_{\mathrm{r}\pi}}{\mu_\pi}\left(-\frac{\partial p_{\mathrm{d},\pi}}{\partial s} + \rho_\pi\,\mathbf{m}_\mathrm{d}{}^\top\mathbf{g}\right)\begin{bmatrix}1\\0\end{bmatrix}, \tag{45}$$

where $k_\mathrm{d}^\mathrm{L}$ is the longitudinal permeability, defined by the cubic law

$$k_\mathrm{d}^\mathrm{L} = \frac{w^2}{12}, \tag{46}$$

where $w$ is the discontinuity aperture.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


The pore-pressure field of phase $\pi$ inside the discontinuity is approximated with

$$p^h_{\mathrm{d},\pi} = \mathbf{N}_{\mathrm{d,p}}(s)\,\mathbf{p}_{\mathrm{d},\pi}, \tag{47}$$

where $\mathbf{N}_{\mathrm{d,p}}(s)$ is a vector with the Lagrange functions in terms of the tangential coordinate of the discontinuity and $\mathbf{p}_{\mathrm{d}}$ 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,

$$\mathbf{r}_{\mathrm{d},\pi}(t) = \int\limits_{\Gamma_{\mathrm{d}}} \mathbf{N}_{\mathrm{d,p}}^{\top}\,\phi_{\mathrm{d}}\,\frac{w}{\rho_\pi}\,\frac{\partial(\rho_\pi\, S_\pi)}{\partial t}\mathrm{d}\Gamma + \int\limits_{\Gamma_{\mathrm{d}}} \frac{\partial \mathbf{N}_{\mathrm{d,p}}^{\top}}{\partial s}\,k_{\mathrm{d}}^{\mathrm{L}}\,w\,\frac{k_{\mathrm{r}\pi}}{\mu_\pi}\,\frac{\partial \mathbf{N}_{\mathrm{d,p}}}{\partial s}\,\mathrm{d}\Gamma\,\mathbf{p}_{\mathrm{d},\pi} - \int\limits_{\Gamma_{\mathrm{d}}} \frac{\partial \mathbf{N}_{\mathrm{d,p}}^{\top}}{\partial s}\,k_{\mathrm{d}}^{\mathrm{L}}\,w\,\frac{k_{\mathrm{r}\pi}}{\mu_\pi}\rho_\pi\,\mathbf{m}_{\mathrm{d}}{}^{\top}\mathbf{g}\,\mathrm{d}\Gamma. \tag{48}$$

The same numerical treatments described in Appendices A and B are applied.

To condense these DOFs and compatibilize the fluid flow between the discontinuity and the porous media, we follow the approach proposed by Mejia et al. (Mejia et al., 2021). Their formulation consists of writing the discontinuity pressure DOFs and their admissible variation as a linear combination of the porous media pore-pressure DOFs as

$$\mathbf{p}_{\mathrm{d},\pi} = \begin{bmatrix} \mathbf{N}\left(\mathbf{x}_{\mathrm{d}}^{(1)}\right) \\ \mathbf{N}\left(\mathbf{x}_{\mathrm{d}}^{(2)}\right) \end{bmatrix} \mathbf{p}_\pi. \tag{49}$$

With this reduction of the space of the DOFs, the contribution of the discontinuities to the fluid flow can be directly summed to the ones computed from the continuum.

The extension of this formulation to cases in which the pore-pressure field can present a jump across the discontinuity, as well as the coupling with the mechanical deformations, is presented and discussed in (Cavalcanti et al., 2024a, b).

### 3.8 Solution process

This section presents the different types of analysis available in `PorousLab`.

#### 3.8.1 Linear analysis

In problems with linear constitutive behavior and time-independent conditions, such as linear elasticity in physics `M` or steady-state flow in physics `H`, the finite element formulation reduces to a linear system of the form

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}, \tag{50}$$

where $\mathbf{A}$ is the global coefficient matrix (stiffness matrix from discretized momentum balance in mechanical analyses or transmissibility matrix from Darcy's law in hydraulic problems), $\mathbf{x}$ is the vector of unknown DOFs (displacements in physics `M` or pore-pressure in physics `H`), and $\mathbf{b}$ is the vector of forcing terms (applied forces, initial stresses, fluxes, and source terms).

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


#### 3.8.2 Nonlinear quasi-static analysis

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, $\mathbf{f}_\mathrm{i}$, and external, $\mathbf{f}_\mathrm{e}$, force vectors, expressed as

$$\mathbf{r}(\mathbf{x},\lambda) = \mathbf{f}_\mathrm{i}(\mathbf{x}) - \lambda\,\mathbf{f}_\mathrm{e}, \tag{51}$$

where $\lambda$ is a load factor. Using an incremental-iterative approach, the vector of unknown DOFs and the load factor are updated as

$$\mathbf{x}_n^{(k)} = \mathbf{x}_{n-1} + \Delta\mathbf{x}_n^{(k-1)} + \delta\mathbf{x}_n^{(k)}, \tag{52}$$

$$\lambda_n^{(k)} = \lambda_{n-1} + \Delta\lambda_n^{(k-1)} + \delta\lambda_n^{(k)}, \tag{53}$$

where subscript $n$ indicates the $n$-th analysis step, superscript $(k)$ indicates the $k$-th iteration of that step, $\delta$ refers to an iterative increment, and $\Delta$ refers to the increment accumulated throughout the step.

Linearization of the residual force vector relative to the DOFs vector leads to the system of equations

$$\frac{\partial\mathbf{f}_{\mathrm{i},n}^{(k-1)}}{\partial\mathbf{x}}\delta\mathbf{x}_n^{(k)} = \delta\lambda_n^{(k)}\,\mathbf{f}_\mathrm{e} + \mathbf{r}_n^{(k-1)}, \tag{54}$$

where the derivatives of internal forces with respect to the DOFs correspond to the tangent stiffness matrix, $\mathbf{K}_\mathrm{t}$. By decomposing the vector of iterative DOF increments into (Batoz and Dhatt, 1979)

$$\delta\mathbf{x}_n^{(k)} = \delta\lambda_n^{(k)}\,\delta\mathbf{x}_{\mathrm{p},n}^{(k)} + \delta\mathbf{x}_{\mathrm{r},n}^{(k)}, \tag{55}$$

the solution of the linearized system can be partitioned as

$$\mathbf{K}_{\mathrm{t},n}^{(k-1)}\delta\mathbf{x}_{\mathrm{p},n}^{(k)} = \mathbf{f}_e, \tag{56}$$

$$\mathbf{K}_{\mathrm{t},n}^{(k-1)}\delta\mathbf{x}_{\mathrm{r},n}^{(k)} = \mathbf{r}_n^{(k-1)}. \tag{57}$$

Moreover, the iterative increment of the load factor is determined by enforcing a constraint equation, which dictates the solution algorithm (control method) employed (Rangel, 2019; Leon et al., 2012). Several algorithms adopt a linear constraint of the form

$$\delta\lambda_n^{(k)} = \frac{c_n^{(k)} - \mathbf{a}_n^{(k)}\cdot\delta\mathbf{x}_{\mathrm{r},n}^{(k)}}{b_n^{(k)} + \mathbf{a}_n^{(k)}\cdot\delta\mathbf{x}_{\mathrm{p},n}^{(k)}}, \tag{58}$$

where the parameters $\mathbf{a}$, $b$, and $c$ vary between algorithms. The linear control methods implemented in `PorousLab` are listed in Table 3, along with their respective constraint parameter values (Rangel and Martha, 2019a; Cavalcanti et al., 2022). Other

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


**Table 3.** Constraint parameters used in Eq. (58) for different nonlinear solution algorithms implemented in `PorousLab`.

| | Constraint parameters | | |
|---|---|---|---|
| Algorithm (control method) | $\mathbf{a}_n^{(k)}$ | $b_n^{(k)}$ | $c_n^{(k)}$ |
| Load | 0 | 1 | $\begin{cases} \Delta\bar{\lambda}_n, k=1 \\ 0, k>1 \end{cases}$ |
| Work | $\delta\lambda_n^{(k)}\,\mathbf{f}_e$ | 0 | $\begin{cases} \Delta\bar{W}, k=1 \\ 0, k>1 \end{cases}$ |
| Arc-length with fixed normal plane* | $\delta\mathbf{x}_n^{(1)}$ | $\delta\lambda_n^{(1)}\,\mathbf{f}_e\cdot\mathbf{f}_e$ | 0 |
| Arc-length with updated normal plane* | $\delta\mathbf{x}_n^{(k-1)}$ | $\delta\lambda_n^{(k-1)}\,\mathbf{f}_e\cdot\mathbf{f}_e$ | 0 |
| Minimum norm* | $\delta\mathbf{x}_{\mathrm{p},n}^{(k)}$ | 0 | 0 |
| Orthogonal residual* | $\mathbf{0}$ | $\Delta\mathbf{x}_n^{(k-1)}\cdot\mathbf{f}_e$ | $-\Delta\mathbf{x}_n^{(k-1)}\mathbf{r}_n^{(k-1)}$ |
| Generalized displacement | $\delta\lambda_n^{(1)}\delta\mathbf{x}_{p,n-1}^{(1)}$ | 0 | $\begin{cases} \left(\delta\lambda_1^{(1)}\right)^2 \left(\delta\mathbf{x}_{p,1}^{(1)}\cdot\delta\mathbf{x}_{p,1}^{(1)}\right), \; k=1 \\ 0, \; k>1 \end{cases}$ |

$\Delta\bar{\lambda}_n$ and $\Delta\bar{W}$ are prescribed load and work increments, respectively.

* Applies when $k>1$, while the cylindrical arc-length constraint of Eq. (59) is used when $k=1$.

widely used solution algorithms implemented are the cylindrical and spherical arc-length methods, which introduce a nonlinear constraint of the form

$$\left(\Delta\mathbf{x}_n^{(k-1)}+\delta\mathbf{x}_n^{(k)}\right)\cdot\left(\Delta\mathbf{x}_n^{(k-1)}+\delta\mathbf{x}_n^{(k)}\right)+\eta\left(\Delta\lambda_n^{(k-1)}+\delta\lambda_n^{(k)}\right)^2-s^2=0, \tag{59}$$

where $s$ denotes the arc length and $\eta$ defines the constraint region: $\eta=0$ for cylindrical and $\eta=1$ for spherical.

In addition, to automatically adapt the load ratio increment according to the degree of nonlinearity of the solution, the increment of the first iteration of each step, $\delta\lambda_n^{(1)}$, can be multiplied by the adjustment factor proposed by Ramm (Ramm, 1981) as

$$J=\sqrt{\frac{N_d}{N_{n-1}}}, \tag{60}$$

where $N_d$ is the desired number of iterations for each step and $N_{n-1}$ is the number of iterations performed in the previous step.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


#### 3.8.3 Nonlinear transient analysis

Under time-dependent conditions, the residual vector can be written as

$$\mathbf{r}\left(\mathbf{x}(t),t\right)=\mathbf{f}_i\left(\mathbf{x}(t),t\right)+\mathbf{K}(t)\,\mathbf{x}(t)+\mathbf{C}(t)\,\dot{\mathbf{x}}(t)-\mathbf{f}_e(t), \tag{61}$$

where $\mathbf{C}$ is the global capacity matrix.

To obtain the solution over time, we employ a fully implicit time discretization. Therefore, the residual vector of Eq. (61) can be written in the *n*-th analysis step as

$$\mathbf{r}_n=\mathbf{f}_{i,n}+\mathbf{K}_n\,\mathbf{x}_n+\frac{1}{\Delta t}\mathbf{C}_n\left(\mathbf{x}_n-\mathbf{x}_{n-1}\right)-\mathbf{f}_{e,n}, \tag{62}$$

where $\Delta t$ is the time step size. To minimize the residual iteratively, two schemes are implemented in `PorousLab`: Newton-Raphson and Picard.

In the Newton-Raphson scheme, the vector of unknown DOFs is updated in each iteration with the solution of a linear system as

$$\mathbf{J}_n^{(k)}\,\delta\mathbf{x}_n^{(k)}=-\mathbf{r}_n^{(k)}\rightarrow\mathbf{x}_n^{(k+1)}=\mathbf{x}_n^{(k)}+\delta\mathbf{x}_n^{(k)}, \tag{63}$$

where $\mathbf{J}_n$ is the Jacobian matrix, which is computed as

$$\mathbf{J}_n=\frac{\partial\mathbf{f}_{i,n}}{\partial\mathbf{x}}+\mathbf{K}_n+\frac{1}{\Delta t}\mathbf{C}_n. \tag{64}$$

For the purpose of completeness, Appendix B presents the Jacobian matrix of the most general physics currently implemented, i.e., the hydro-mechanical model with two-phase fluid flow (`H2M`).

In the Picard scheme, nonlinear coefficients are frozen at the previous iteration. This avoids forming the exact Jacobian and gives only linear convergence. At iteration $k$ one solves

$$\mathbf{A}_n^{(k)}\,\mathbf{x}_n^{(k+1)}=\mathbf{b}_n^{(k)}, \tag{65}$$

where

$$\mathbf{A}_n=\mathbf{K}_n+\frac{1}{\Delta t}\mathbf{C}_n, \tag{66}$$

$$\mathbf{b}_n=\frac{1}{\Delta t}\mathbf{C}_n\,\mathbf{x}_{n-1}+\mathbf{f}_{e,n}. \tag{67}$$

## 4 Software architecture and design

`PorousLab` was developed as a modular MATLAB framework for rapid prototyping and verification of finite-element formulations for fractured porous media. The implementation uses object-oriented programming to separate model definition,

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


material behavior, element-level formulations, and solution procedures. This architecture supports code reuse across the implemented physics while keeping formulation-specific operations localized in specialized classes. The choice of a high-level MATLAB implementation, with no additional toolbox dependencies, is intended to reduce the entry barrier for users interested in inspecting, modifying, and extending the implemented formulations.

Following the work in (Martha et al., 1996; Martha and Parente Jr, 2002; Rangel and Martha, 2019b; Wang and Kolditz, 2007), the object-oriented class organization of `PorousLab` is built around three abstract superclasses: `Model`, `Material`, and `Analysis`. These superclasses define the main responsibilities of the framework: finite-element model construction, material and constitutive behavior, and solution procedures, respectively. This organization provides a common structure for the implemented physics while allowing formulation-specific operations to be specialized in derived classes. The following subsections describe the main class families of `PorousLab`, with UML class diagrams (Booch, 2005) being used to summarize their relationships and to clarify how the framework separates model definition, material behavior, and analysis procedures.

### 4.1 Model-related classes

Figure 8 shows the UML class diagram for the model-related classes in `PorousLab`. This class family is built around the abstract superclass `Model`, which defines the common structure required to construct finite-element models independently of the physics being solved. The derived subclasses `Model_M`, `Model_H`, `Model_H2`, `Model_HM`, and `Model_H2M` specialize this structure for the physics types described in Section 2.

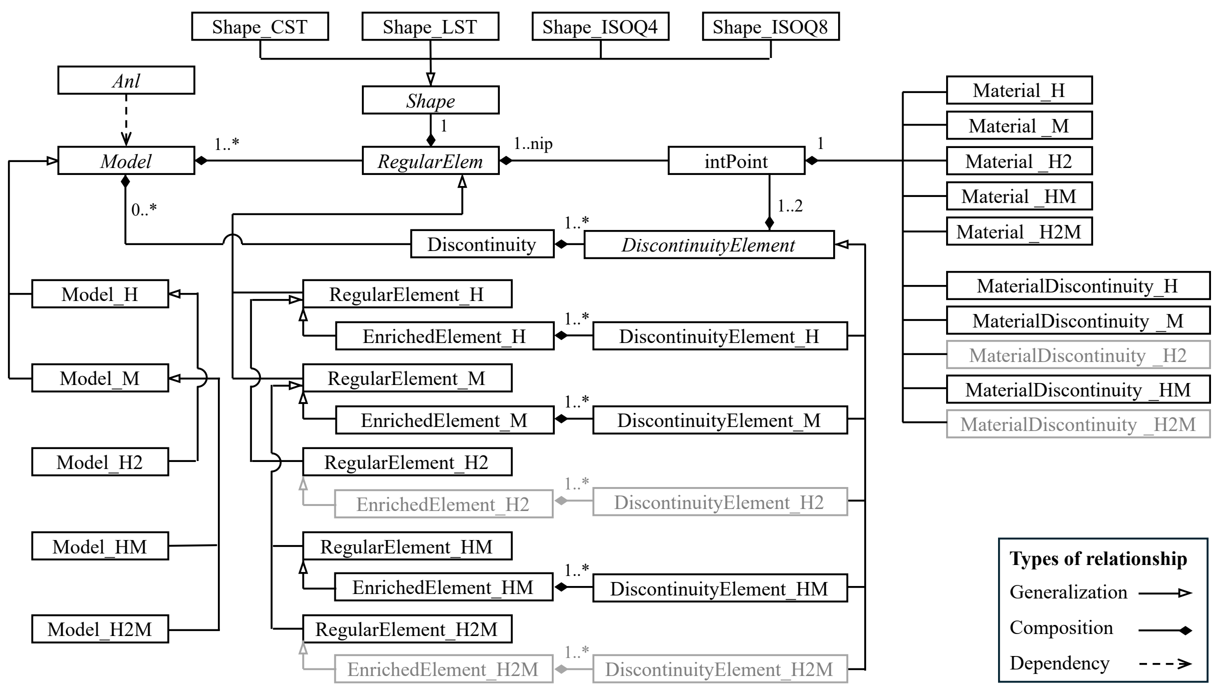



**Figure 8.** UML diagram of model-related classes in `PorousLab` (faded classes are not currently available).

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

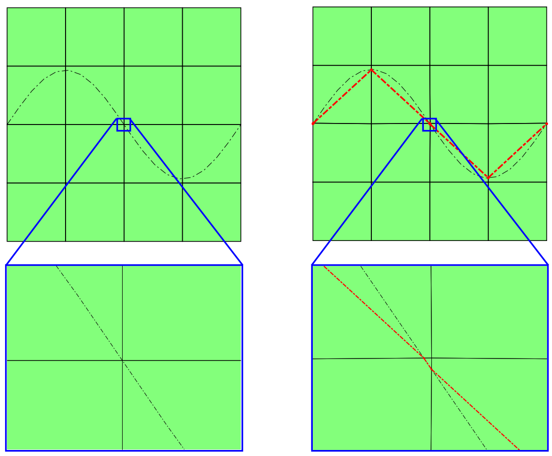

The `Model` superclass centralizes operations that are common to all formulations, including mesh handling, DOF initialization, material assignment, boundary and initial condition management, incorporation of embedded discontinuities, assembly of global matrices and residual vectors, and post-processing of field results. Physics-dependent operations are implemented in the derived subclasses, mainly through the definition of the active DOFs, the validation of material properties, and the initialization of the appropriate element objects. This design follows the physics abstraction used in GeMA (Mendes et al., 2016), while avoiding a monolithic model class with extensive conditional logic.

This specialization is first reflected in the DOF layout. `Model_M` defines mechanical models with two displacement DOFs per node, whereas `Model_H` and `Model_H2` define hydraulic models with one and two pore-pressure DOFs per node, respectively. The coupled classes `Model_HM` and `Model_H2M` combine the corresponding displacement and pore-pressure DOFs to define hydro-mechanical single- and two-phase models.

The same specialization appears at the element level. Each `Model_X` subclass initializes element objects consistent with its physics type, either as standard continuum elements derived from `RegularElement_X` or, when embedded discontinuities are present, as enriched variants derived from `EnrichedElement_X`. The `RegularElement_X` classes implement the formulation-specific element matrices and residual vectors, which are later assembled into the global system. Integration-point objects store local state variables and provide access to the material response evaluated by the corresponding material classes. Geometric and interpolation operations are separated into the `Shape` class hierarchy, whose subclasses define the finite-element interpolation for CST, LST, ISOQ4, and ISOQ8 elements.

Pre-existing strong discontinuities are represented by instances of the `Discontinuity` class. This class stores the geometric and physical properties associated with a fracture or fault, including its curve, initial aperture, cohesive law, normal and shear stiffnesses, and contact-penalty parameters. It also computes the intersections between the discontinuity geometry and the finite-element mesh, providing a linearized representation of the discontinuity as a polyline, consisting of segments generated during the intersection step and aligned with the underlying mesh. When necessary, a node-repelling procedure slightly perturbs mesh nodes located too close to the discontinuity path, reducing numerical artifacts associated with intersections near mesh vertices. Figure 9 illustrates this procedure, showing how a smooth discontinuity curve is intersected with the background mesh to generate the corresponding embedded segments.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

(a) (b)

**Figure 9.** Finite element mesh over a $3\,\mathrm{m} \times 3\,\mathrm{m}$ quadrilateral domain containing a discontinuity defined by $y = 0.7\sin\left(\frac{2\pi}{3}x\right) + 1.5$. (a) Original discontinuity geometry. (b) Linearized approximation of the discontinuity obtained after its intersection with the mesh. The zoomed-in views highlight the node-repelling procedure.

Elements crossed by a discontinuity are defined as enriched element objects derived from `EnrichedElement_X`. These classes extend the corresponding `RegularElement_X` formulations with additional operations required by the embedded representation. The local contribution of each discontinuity segment is handled by `DiscontinuityElement` objects, which store segment-level geometry and evaluate the discontinuity terms to be added to the element residuals and tangent matrices.

For mechanical discontinuities, the enrichment can reproduce different variants of the Strong Discontinuity Approach by selecting the corresponding set of enhanced deformation modes and numerical treatments. The default formulation includes the constant normal and tangential displacement-jump modes, following Oliver (1996) and subsequent works (Wells and Sluys, 2000; Cazes et al., 2016; de Borst et al., 2001; Oliver et al., 2003; Jirásek, 2000). Additional modes, such as tangential stretching and relative rotation, can be activated to recover other formulations proposed in the literature. Table 4 summarizes the combinations currently implemented and the validation tests used to verify each configuration.

**Table 4.** Implemented configurations of mechanical embedded strong-discontinuity formulations and corresponding validation tests.

| | Reference formulation | | |
|---|---|---|---|
| Formulation feature | Linder and Armero (2007) | Dias-da Costa et al. (2009a) | Dias-da Costa et al. (2009b) |
| Tangential stretching mode | ✓ | ✓ | ✓ |
| Relative rotation mode | ✓ | ✓ | ✓ |
| Symmetric formulation | | ✓ | ✓ |
| Nodal enrichment DOFs | | ✓ | ✓ |
| Sub-cell integration | | | ✓ |
| Validation test | Partial tension | Oblique mixed-mode | Stretching |

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


## 4.2 Material-related classes

The material-related classes define the physical properties and constitutive models. For each physics type, `PorousLab` provides a corresponding `Material_X` class for continuum elements and, when embedded discontinuities are considered, a `MaterialDiscontinuity_X` class for discontinuity contributions. This separation allows continuum and discontinuity behavior to be specified independently while preserving a common material structure across the implemented physics. Figure 10 shows the UML class diagram for the continuum material classes.

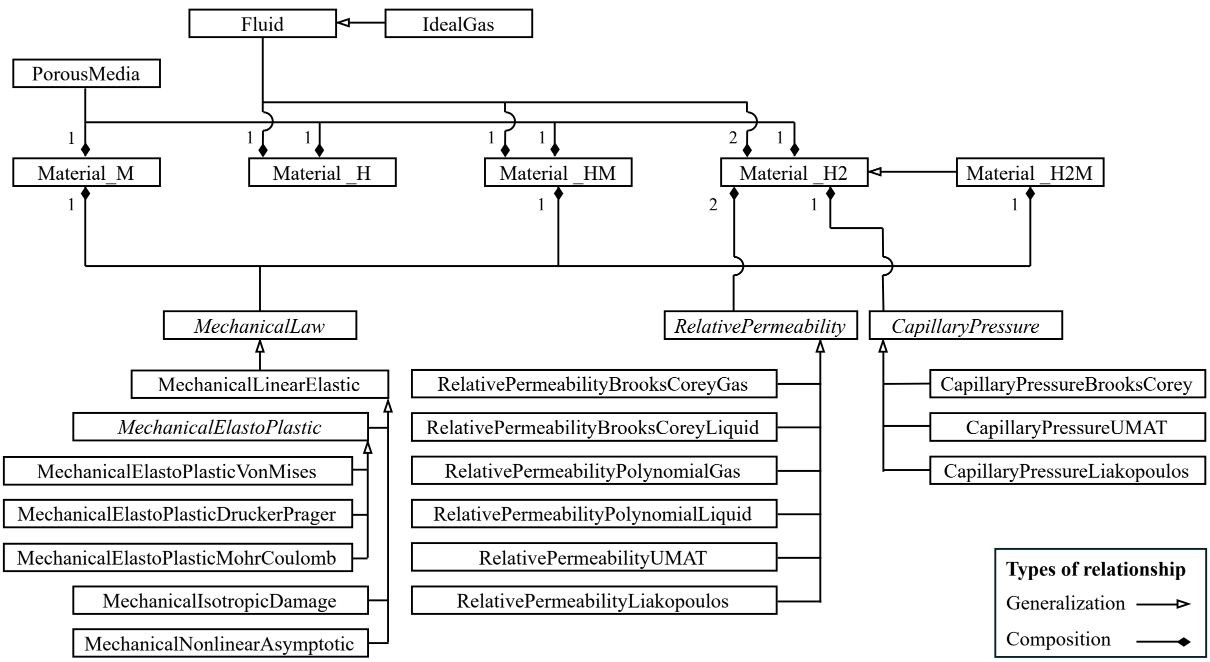



**Figure 10.** UML diagram of continuum material-related classes in `PorousLab`.

All continuum material classes contain a `PorousMedia` object, which stores the basic properties of the porous matrix, including porosity, intrinsic permeability, Biot's coefficient, solid bulk modulus, Young's modulus, and Poisson's ratio. For mechanical formulations, the porous-media object is associated with a constitutive law derived from the abstract superclass `MechanicalLaw`. The implemented laws include linear elasticity, isotropic damage, and elastoplastic models such as von Mises, Drucker–Prager, and Mohr–Coulomb. The elastoplastic models share a common return-mapping structure through the abstract class `MechanicalElastoPlastic`, while the specific yield functions, flow directions, and derivatives are defined in derived classes.

For hydraulic and hydro-mechanical formulations, the material classes include fluid properties and, in two-phase flow models, additional constitutive relationships for capillary pressure (retention curve) and relative permeability. These relationships are represented by dedicated class hierarchies, allowing analytical laws, polynomial expressions, and user-defined tabulated curves to be used within the same material structure.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


A similar organization is used for discontinuity materials, as summarized in Fig. 11. For the physics types that support embedded discontinuities, the `MaterialDiscontinuity_X` classes define the constitutive behavior of fractures and faults. In mechanical and hydro-mechanical models, this includes cohesive laws relating displacement jumps to tractions. The current implementation provides a linear elastic law and an elastoplastic Mohr-Coulomb law with tensile cutoff. For hydraulic discontinuities, the material definition includes aperture, longitudinal permeability, leak-off parameters, and fluid properties.

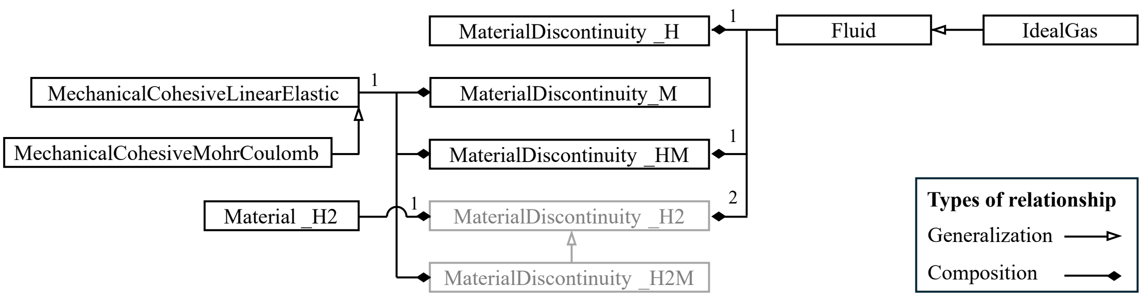



**Figure 11.** UML diagram of discontinuity material-related classes in `PorousLab` (faded classes are not currently available).

### 4.3 Analysis-related classes

Figure 12 shows the UML class diagram for analysis-related classes, which are structured around the abstract superclass `Anl` and its specialized derivatives corresponding to the analysis types supported by `PorousLab`, namely linear, nonlinear quasi-static, and nonlinear transient.

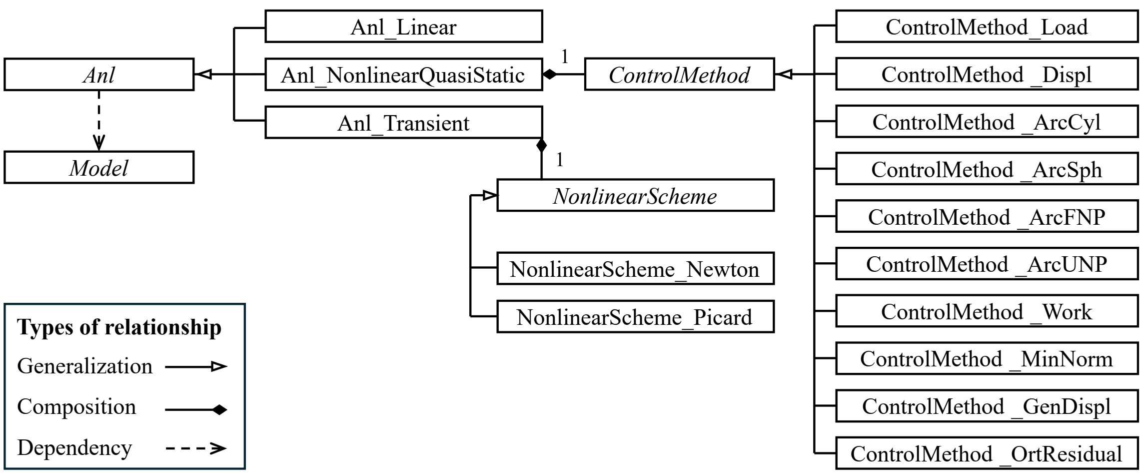



**Figure 12.** UML diagram of analysis-related classes in `PorousLab`.

A relevant feature of this class family is the support for nonlinear quasi-static mechanical analyses with different continuation strategies. These strategies are represented through the `ControlMethod` hierarchy and include load, work, generalized

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


displacement, minimum-norm, orthogonal-residual, and arc-length-type controls. This capability is particularly useful for geomechanical problems involving material nonlinearity, softening, or limit-point behavior, where simple load control may be insufficient.

For nonlinear transient analyses, the iterative solution is handled through the `NonlinearScheme` hierarchy, which currently provides Newton-Raphson and Picard schemes. In all analysis types, the resulting global linear system is solved using MATLAB's built-in direct solvers.

## 5 Simulation Workflow

The use of `PorousLab` follows a scripted workflow consistent with the object-oriented organization described in Section 4. A simulation is defined through a model object, which stores the mesh, material data, initial conditions, boundary conditions, and physics-specific finite-element information, and an analysis object, which controls the solution procedure. This separation makes the same workflow applicable to the different physics types implemented in the framework.

Algorithm 1 summarizes the main steps required to define and execute a simulation. The model is first instantiated according to the selected physics type. The mesh is then generated internally or imported from an external meshing tool, and material objects are assigned to the corresponding element sets. Boundary and initial conditions are prescribed at the model level, while the analysis object defines the type of solution procedure, nonlinear scheme, time-stepping parameters, and convergence tolerances.

**Algorithm 1** Workflow for setting up a simulation in `PorousLab`.

- **Create and configure the model object `mdl`:**
  - Instantiate model class (e.g., `Model_M`, `Model_H`)
  - Load or generate mesh data: `node`, `elem`
  - Set mesh with `mdl.setMesh(node, elem)`
  - Define and configure material objects (e.g., `PorousMedia`, `Fluid`)
  - Assign materials using `mdl.setMaterial(...)`
  - Apply boundary and initial conditions
- **Create and configure the analysis object `anl`:**
  - Instantiate class (e.g., `Anl_Linear`, `Anl_Transient`)
  - Set solver parameters
- **Run the simulation:** `anl.run(mdl)`
- **Post-process results:** Use built-in tools such as `mdl.plotField(...)` for field visualization.

The workflow covers the complete sequence of a simulation within a single MATLAB environment. In the pre-processing stage, the framework provides built-in utilities for mesh generation or import, material assignment, initial condition definition, and boundary condition prescription. The analysis object then executes the selected linear, nonlinear quasi-static, or nonlinear transient procedure. After the solution, built-in visualization and extraction tools allow primary and secondary fields to be

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


inspected directly from the model object. Since all modeling choices are explicitly defined in MATLAB scripts, the workflow is reproducible and reduces the dependence on external pre- and post-processing software.

Additional details on the pre-processing, post-processing, and graphical user interface tools are provided in Supplement Sect. S1.

## 6 Verification examples

This section presents selected verification examples used to assess the formulations implemented in `PorousLab`. The examples cover the main physics currently available in the framework, including mechanical, single-phase flow, two-phase flow, and coupled hydro-mechanical problems. They demonstrate the correctness of the implementation by comparison with analytical, semi-analytical, or reference solutions from the literature. The complete scripts required to reproduce the examples are included in the archived `PorousLab` release cited in the Code availability section and are also maintained in the public GitHub repository. The corresponding input scripts and the main `PorousLab` commands used to reproduce these examples are documented in Supplement Sect. S2. Additional benchmark documentation is provided in the project wiki linked from the repository.

### 6.1 Strip footing bearing capacity

The elastoplastic implementation is verified with the classical strip-footing problem in plane strain (de Souza Neto et al., 2011). The soil is weightless, and a rigid footing applies a uniform vertical pressure over a finite width at the ground surface. Geometry and boundary conditions are given in Fig. 13.

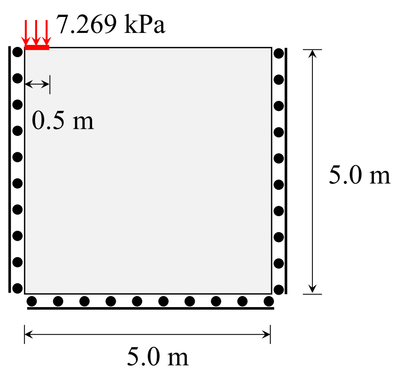


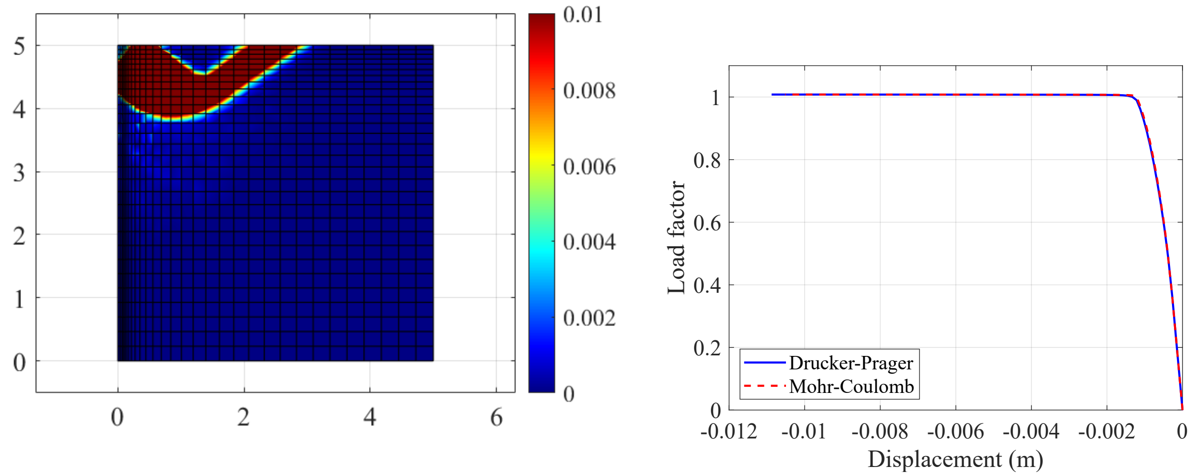

**Figure 13.** Strip footing: geometry and boundary conditions.

The value of uniform pressure prescribed over the footing region corresponds to Prandtl's analytical bearing-capacity solution. The soil is modeled with Young's modulus of $10^7$ kPa, Poisson's ratio of 0.48, cohesion of 490 kPa, and friction and dilation angles of 20°. The problem is analyzed with both the Drucker–Prager and Mohr–Coulomb plasticity models. Spatial

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


discretization is performed with a $30 \times 30$ mesh of eight-noded quadrilateral elements, graded toward the loaded region to provide a finer resolution beneath the footing. A quasi-static nonlinear analysis is performed using the Generalized Displacement control method.

Figure 14a presents the collapse mechanism through the visualization of the plastic strain magnitude over the domain. The load-displacement curves of the upper left node's vertical DOF obtained using both Drucker-Prager and Mohr-Coulomb are presented in Fig. 14b. They provide the same results, converging to the limit pressure estimated by Prandtl's solution.




(a) (b)

**Figure 14.** Strip footing collapse: (a) Plastic strain magnitude at the final step using the Drucker-Prager plasticity model; (b) Load-factor evolution concerning the vertical displacement at the top left point.

### 6.2 Stress changes along a fault in a pressurized reservoir

This example verifies three aspects of the mechanical formulation: the pore-pressure loading, the initialization of an in-situ stress state, and the recovery of cohesive tractions along an embedded fault. A two-dimensional elastic domain containing an inclined pre-existing fault with a dip angle of 60° is considered, as shown in Fig. 15. The model represents a horizontal reservoir subjected to a pore-pressure increase. The initial pore-pressure is $P_0 = 35$ MPa, and an increment $\Delta P = 20$ MPa is applied within the reservoir layer.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

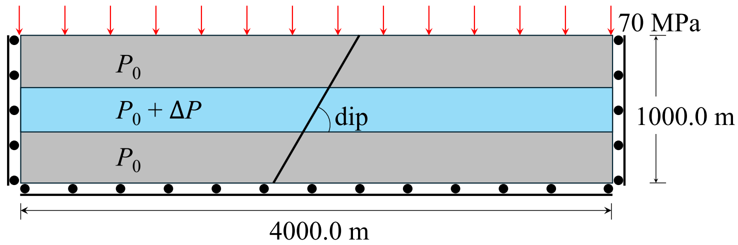




**Figure 15.** Pressurized reservoir: geometry and boundary conditions.

535 The continuum is modeled as a linear-elastic porous medium under plane-strain conditions, and the embedded fault is represented with the mechanical E-FEM formulation, allowing the normal and shear cohesive tractions to be extracted directly along the discontinuity. The simulation is performed in two stages. First, an initial equilibrium problem is solved using the prescribed boundary conditions and the initial pore-pressure field, providing the in-situ effective stress state. The displacement field is then reset while retaining this stress state as the initial condition for the subsequent loading stage. Second, the pore-
540 pressure increment is applied only inside the reservoir layer, and the model is solved again to obtain the incremental stress changes caused by reservoir pressurization.

The numerical solution is compared with the elastic at-rest relations for an isotropic medium under laterally constrained deformation (Fjær et al., 2008; Novikov et al., 2023),

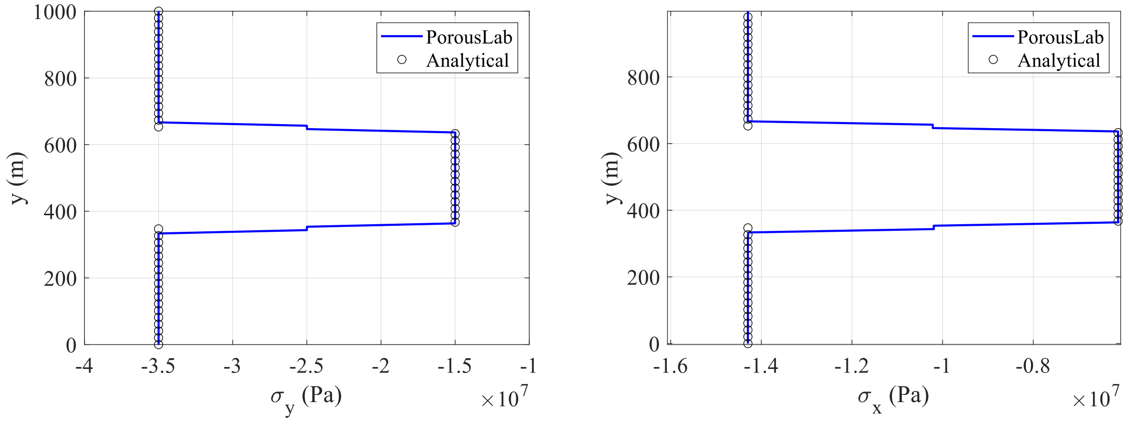
$$\sigma'_{\mathrm{h}} = \frac{\nu}{1-\nu}\sigma'_{\mathrm{v}}, \qquad \Delta\sigma'_{\mathrm{h}} = \frac{\nu}{1-\nu}\Delta\sigma'_{\mathrm{v}}, \qquad \Delta\sigma'_{\mathrm{v}} = \alpha\,\Delta p.$$

545 The corresponding tractions acting on the fault are obtained by projecting the effective stress tensor onto the local normal and tangential directions of the discontinuity,

$$\begin{pmatrix} t_{\mathrm{s}} \\ t_{\mathrm{n}} \end{pmatrix} = \begin{pmatrix} \mathbf{m}_{\mathrm{d}}^{\top} \\ \mathbf{n}_{\mathrm{d}}^{\top} \end{pmatrix} \begin{bmatrix} \sigma'_{\mathrm{h}} & 0 \\ 0 & \sigma'_{\mathrm{v}} \end{bmatrix} \mathbf{n}_{\mathrm{d}}.$$

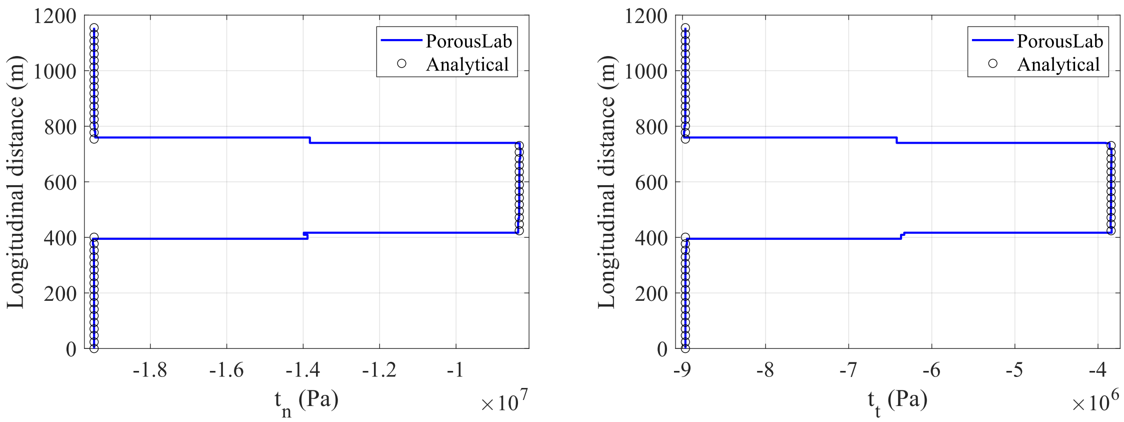
Figures 16 and 17 compare the continuum stress profiles and the cohesive tractions along the fault with the analytical values. The results show close agreement, confirming the implementation of pore-pressure loading in the mechanical formulation and
550 the stress components along embedded discontinuities.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


(a) (b)

**Figure 16.** Continuum effective stresses along $y$ at $x = 2000$ m: (a) vertical component, $\sigma'_y$; (b) horizontal component, $\sigma'_x$.




(a) (b)

**Figure 17.** Cohesive traction along the fault: (a) normal component, $t_\mathrm{n}$; (b) shear component, $t_\mathrm{s}$.

### 6.3 Fluid flow through the foundation of a dam

This example verifies the single-phase hydraulic formulation and the embedded representation of hydraulic discontinuities in a steady seepage problem. A gravity-dam foundation is analyzed under prescribed upstream and downstream hydraulic heads, as shown in Fig. 18. The porous matrix has intrinsic permeability $1.0194 \times 10^{-14}$ m$^2$ and porosity 0.3.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

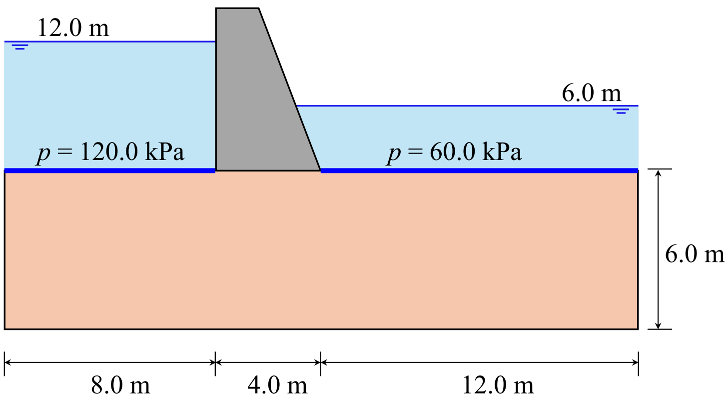




**Figure 18.** Fluid flow through a dam foundation: geometry and boundary conditions.

The porous medium domain is discretized with a structured mesh of $48 \times 12$ linear quadrilateral elements. Three configurations are considered using the same background mesh: an intact foundation, a foundation crossed by a regular set of parallel discontinuities, and a foundation containing an irregular set of discontinuities. This comparison illustrates one of the main advantages of the embedded formulation: different discontinuity geometries can be introduced without modifying the background mesh.

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 regular set of parallel fractures case, the results are compared with reference solutions obtained using discrete-fracture models based on zero-thickness interface elements (Segura and Carol, 2004). Figure 19 shows that the pore-pressure profiles obtained with `PorousLab` agree closely with the reference solutions, confirming the implementation of the single-phase flow formulation and the hydraulic embedded-discontinuity contribution.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

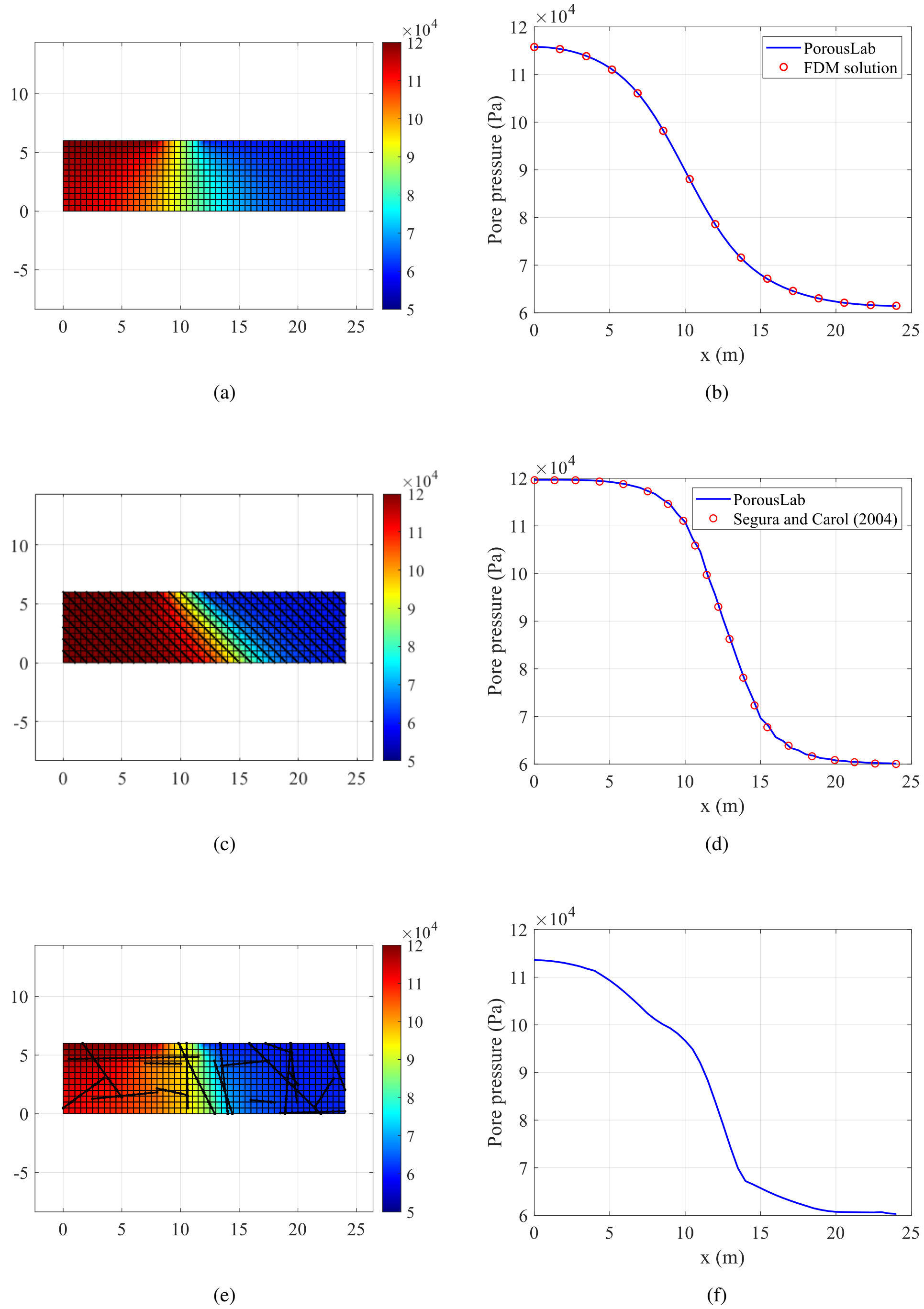




**Figure 19.** Fluid flow through a dam foundation: pore-pressure fields (left) and horizontal profiles along $x$ at $y = 3$m (right). (a)–(b) intact domain (no fractures); (c)–(d) domain with parallel fractures; (e)–(f) domain with random fractures.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


### 6.4 Fluid injection into a single fracture

This example verifies the coupled hydro-mechanical formulation with an embedded hydraulic-mechanical discontinuity. The problem is based on the semi-analytical solution derived by Wijesinghe (1986) for fluid injection into a single fracture embedded in an elastic porous medium. The geometry and boundary conditions are shown in Fig. 20.

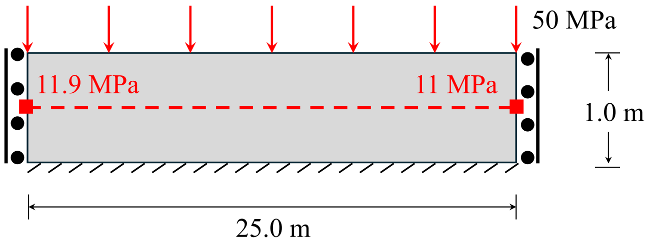



**Figure 20.** Wijesinghe problem: geometry and boundary conditions. The fractured is represented by the red dashed line.

The porous medium is modeled as linear-elastic following the parameters adopted by Watanabe et al. (2012): Young's modulus of $60\,\mathrm{GPa}$, Poisson's ratio of $0.0$, intrinsic permeability of $1.0 \times 10^{-21}\,\mathrm{m}^2$, and porosity of $0.001$. The embedded fracture is represented with an elastic cohesive law, normal and shear stiffnesses of $100\,\mathrm{GPa/m}$, and initial aperture of $1.0 \times 10^{-5}\,\mathrm{m}$. The domain is discretized with a structured mesh of $50 \times 21$ linear quadrilateral elements, with finer refinement in the vicinity of the fracture.

The analysis is performed in two stages. First, an initial equilibrium problem is solved with a uniform pore-pressure of $11.0\,\mathrm{MPa}$, providing the initial stress and deformation state of the porous medium and fracture. Second, the hydraulic boundary conditions are modified to impose fluid injection at the left fracture mouth, with a pressure of $11.9\,\mathrm{MPa}$, while the right boundary is kept at $11.0\,\mathrm{MPa}$. During this transient stage, the fracture aperture is updated from the mechanical deformation.

The numerical solution is compared with the semi-analytical profiles of Wijesinghe (1986). Figure 21 shows the pore-pressure and aperture profiles along the fracture at $t = 500\,\mathrm{s}$. The results show good agreement with the reference solution, thus verifying the hydro-mechanical coupling between fracture pressure and aperture variation in `PorousLab`.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

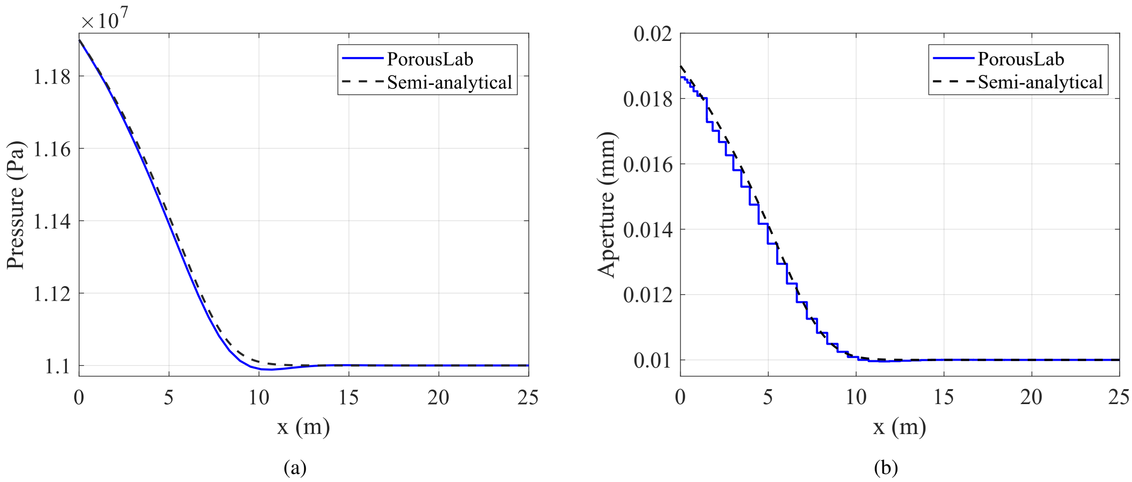




**Figure 21.** Wijesinghe problem: solution at $t = 500$ s for (a) pore-pressure and (b) aperture.

### 6.5 McWhorter-Sunada infiltration problem

This example verifies the two-phase flow formulation using the classical McWhorter-Sunada infiltration problem (McWhorter and Sunada, 1990). Figure 22 shows the 2.6 $\mathrm{m} \times 0.5$ $\mathrm{m}$ domain and boundary conditions.

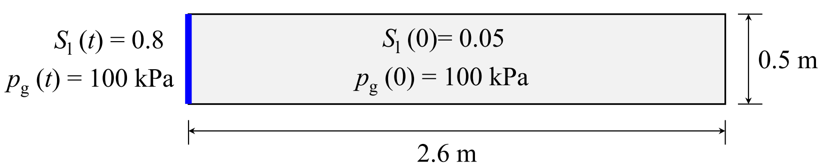



**Figure 22.** McWhorter and Sunada infiltration problem: geometry and boundary conditions.

The porous medium has intrinsic permeability of $1.0 \times 10^{-10}$ $\mathrm{m}^2$ and porosity of $0.15$. The capillary-pressure and relative-permeability curves follow Brooks-Corey laws, with residual liquid saturation of $0.02$, residual gas saturation of $0.001$, gas-entry pressure of $5.0\,\mathrm{kPa}$, and curve-fitting parameter of $3.0$. The domain is discretized with a structured mesh of $100 \times 1$ linear quadrilateral elements.

Although the initial and boundary conditions of the benchmark are defined in terms of liquid saturation and gas pressure, the implemented two-phase formulation adopts the liquid and gas pressures as primary variables. Therefore, the prescribed and initial saturation values are converted into liquid pressures through the capillary-pressure relation given in Eq. (2).

A nonlinear transient analysis is then performed using Newton iterations with adaptive time stepping. The numerical solution is compared with the semi-analytical reference solution of Fučík (Fučík, 2021). Figure 23 shows the liquid-saturation profile along the bottom boundary at $t = 1000\,\mathrm{s}$. The results show good agreement with the reference solution, thus verifying the implementation of the two-phase flow in `PorousLab`.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

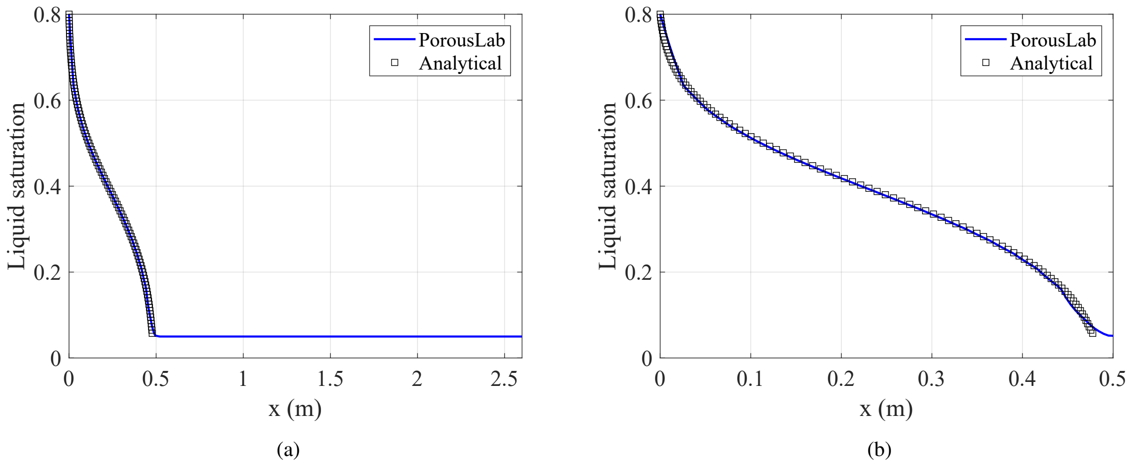




**Figure 23.** McWhorter and Sunada infiltration problem: (a) liquid saturation profile at 1000.0 s and (b) zoomed view.

### 6.6 Liakopoulos test

This example verifies the coupled hydro-mechanical formulation with two-phase flow using the Liakopoulos vertical column test. The benchmark consists of a deformable porous column subjected to gravity drainage, as shown in Fig. 24.

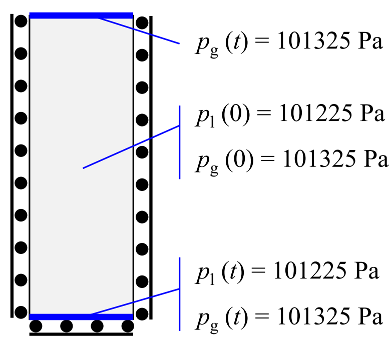



**Figure 24.** Liakopoulos test: geometry and boundary conditions.

The domain is discretized with a structured mesh of $1 \times 100$ linear quadrilateral elements. The liquid phase (water) is modeled as nearly incompressible, with bulk modulus of $1.0 \times 10^{25}$ Pa. The gas phase is modeled as an ideal gas, with dynamic viscosity of $1.8 \times 10^{-5}$ Pa s, molar mass of $0.028949$ kg/mol, and temperature of 300 K. The porous medium has intrinsic permeability of $4.5 \times 10^{-13}\,\mathrm{m}^2$, porosity of 0.2975, solid bulk modulus of $1.0 \times 10^{25}$ Pa, Young's modulus of 1.3 MPa, Poisson's ratio of 0.4, and density of $2000\,\mathrm{kg/m}^3$. The capillary-pressure curve and the liquid relative-permeability law follow the empirical relations adopted in the Liakopoulos benchmark. The gas relative permeability is represented by a Brooks-Corey law, with residual liquid saturation of 0.2, curve-fitting parameter of 3.0, and minimum gas relative permeability of $1.0 \times 10^{-4}$.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026

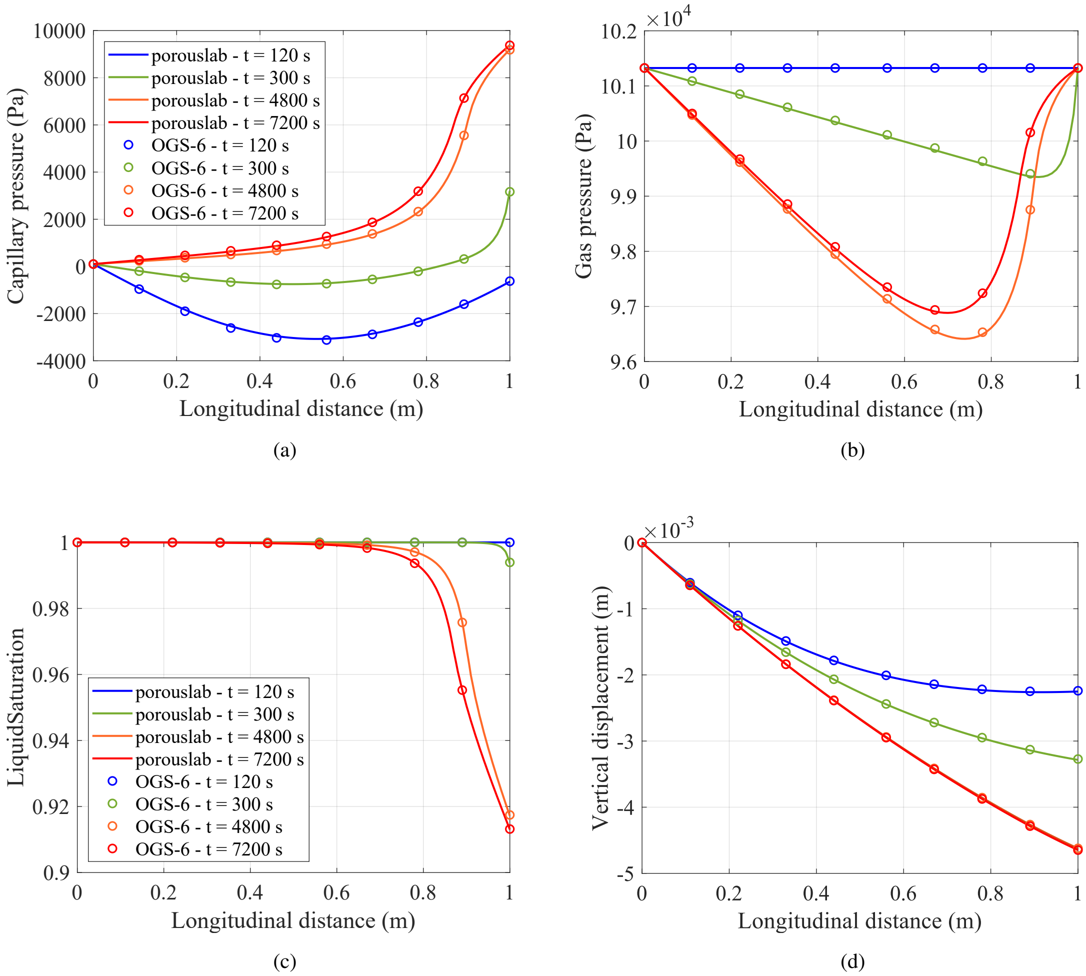




**Figure 25.** Liakopoulos test: comparison between PorousLab and OGS-6 results. Profiles of (a) capillary pressure, (b) gas pressure, (c) liquid saturation, and (d) vertical displacement.

The initial stress state is prescribed before the transient calculation, so that the subsequent response represents the coupled evolution of pressure, saturation, and deformation during drainage. A nonlinear transient analysis is performed using Newton iterations with adaptive time stepping.

The numerical solution is compared with reference results obtained with OGS-6 (Grunwald et al., 2022). Figure 25 presents the profiles of capillary pressure, gas pressure, liquid saturation, and vertical displacement along the column at selected times. The results show close agreement with the reference solution, thus verifying the implementation of the coupled hydro-mechanical two-phase formulation in `PorousLab`.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


## 7 Conclusions

This work presented `PorousLab`, an open-source, object-oriented finite-element framework implemented in MATLAB for the simulation of porous-media problems involving coupled hydromechanical formulations with two-phase flow. The contribution of the framework lies in integrating these formulations, nonlinear geomechanical analysis procedures, and embedded strong-discontinuity modeling within a single FEM-based and scriptable computational environment. In this sense, `PorousLab` is intended as an accessible platform for computational geomechanics researchers interested in formulation development, verification studies, and transparent implementation of coupled porous-media models.

The architecture is organized around model, material, and analysis classes, which separate model definition, constitutive behavior, and solution procedures. This organization supports code readability, reuse, and extension, while preserving a complete workflow within the MATLAB environment, including model construction, solution, orchestration, and post-processing. Embedded discontinuities are represented independently of the background mesh through an E-FEM formulation, allowing pre-existing fractures and faults to be introduced without requiring mesh conformity.

The implemented formulations were verified through benchmark examples covering the main physics currently available in the framework. The comparisons with reference solutions showed good agreement, supporting the correctness of the implemented formulations.

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.

Future developments will focus on extending the range of coupled processes and improving computational capability while preserving the transparent structure of the framework. Relevant directions include the incorporation of thermal effects and the extension of the two-phase flow formulation to a two-component/two-phase version.

*Code and data availability.* The current development version of `PorousLab` is available from GitHub at https://github.com/dbcavalcanti/porousLab/ under the MIT licence. The exact archived release of `PorousLab` v1.1.0 is available from Zenodo at https://doi.org/10.5281/zenodo.15624961 (Cavalcanti, 2026). The archive includes the source code, benchmark scripts, documentation, and regression tests.

### Appendix A: Residual numerical treatments in physics `H2`

To guarantee the mass conservation in time, we follow the same numerical treatments done by CODE_BRIGHT (Olivella et al., 1994, 1996) and OGS-6 (Grunwald et al., 2022). In their formulation, the time derivatives in Eq. (32) are approximated numerically with a fully-implicit scheme:

$$\frac{\partial ()}{\partial t} \approx \frac{()_n - ()_{n-1}}{\Delta t} \tag{A1}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


where $\Delta t$ is the time step size.

We assume that the solid grains are incompressible, which implies $K_s \to \infty$ and $\alpha = 1$. The porosity is taken from the previous converged step, $\phi = \phi_{n-1}$, and is considered an element-wise variable, having a constant value over the domain of each finite element. The residual is then written as

$$\mathbf{r}_\pi(t_n) = \mathbf{r}_\pi^{\mathrm{m}}(t_n) + \mathbf{r}_\pi^{\mathrm{a}}(t_n) + \mathbf{r}_\pi^{\mathrm{v}}(t_n) + \mathbf{r}_\pi^{\mathrm{s}}(t_n), \quad \pi = \mathrm{l}, \mathrm{g}, \tag{A2}$$

where $\mathbf{r}_\pi^{\mathrm{m}}$ corresponds to the mass storage terms,

$$\mathbf{r}_\pi^{\mathrm{m}}(t_n) = \int_\Omega \mathbf{N}_{\mathrm{p}}^\top \frac{\phi_{n-1}}{\rho_\pi} \frac{(\rho_\pi S_\pi)_n - (\rho_\pi S_\pi)_{n-1}}{\Delta t} \, \mathrm{d}\Omega, \quad \pi = \mathrm{l}, \mathrm{g}, \tag{A3}$$

$\mathbf{r}_\pi^{\mathrm{a}}$ to the advective fluxes,

$$\mathbf{r}_\pi^{\mathrm{a}}(t_n) = \int_\Omega \mathbf{B}_{\mathrm{p}}^\top \frac{\mathbf{k}\, k_{\mathrm{r}\pi}}{\mu_\pi} \mathbf{B}_{\mathrm{p}} \, \mathrm{d}\Omega \, \mathbf{p}_\pi - \int_\Omega \mathbf{B}_{\mathrm{p}}^\top \frac{\mathbf{k}\, k_{\mathrm{r}\pi}}{\mu_\pi} \rho_\pi \, \mathbf{g} \, \mathrm{d}\Omega, \quad \pi = \mathrm{l}, \mathrm{g}, \tag{A4}$$

$\mathbf{r}_\pi^{\mathrm{v}}$ to the volumetric strain terms,

$$\mathbf{r}_\pi^{\mathrm{v}}(t_n) = \frac{1}{\Delta t} \int_\Omega \mathbf{N}_{\mathrm{p}}^\top S_\pi \, \mathbf{m}^\top \mathbf{B}_{\mathrm{u}} \, (\mathbf{d}_n - \mathbf{d}_{n-1}) \, \mathrm{d}\Omega, \quad \pi = \mathrm{l}, \mathrm{g}, \tag{A5}$$

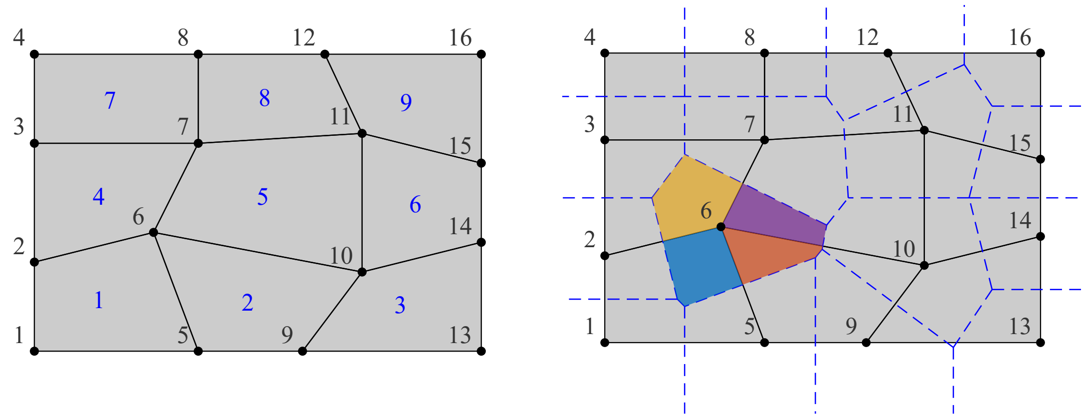
and $\mathbf{r}_\pi^{\mathrm{s}}$ to the sink/source terms,

$$\mathbf{r}_\pi^{\mathrm{s}}(t_n) = \int_{\Gamma_{q_\pi}} \mathbf{N}_{\mathrm{p}}^\top \, \bar{q}_\pi \, \mathrm{d}\Gamma, \quad \pi = \mathrm{l}, \mathrm{g}. \tag{A6}$$

### A1 Treatment of the mass storage terms

To guarantee local mass conservation, the saturation degree and the density are treated as cell-wise variables (Olivella et al., 1996). However, they are not strictly cell-wise, since they depend not only on pore-pressure, but also on material properties, such as permeability, which are defined element-wise. Figure A1 illustrates the concept by showing the node-centered cells associated with a finite element mesh. This cell-wise definition does not alter the assembly procedure, which is still carried out by looping over the elements. For instance, the cell associated with node 6 receives contributions from elements 1, 2, 4, and 5 (Fig. A1).

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


(a) (b)

**Figure A1.** Cell-definition: (a) Finite-element mesh (node labels in black and element labels in blue); (b) Cells with a highlight to the cell associated with node 6.

For an element with $n_{\text{en}}$ nodes with a linear geometry, $\mathbf{r}_{\pi}^{\text{m}}$ is

$$\mathbf{r}_{\pi}^{\text{m}}(t_n) = \frac{\phi_{n-1}}{\Delta t}\begin{Bmatrix} \int_{\Omega} N_1 \,\mathrm{d}\Omega \, \frac{(\rho_\pi S_\pi)_n^1 - (\rho_\pi S_\pi)_{n-1}^1}{\rho_\pi^1} \\ \int_{\Omega} N_2 \,\mathrm{d}\Omega \, \frac{(\rho_\pi S_\pi)_n^2 - (\rho_\pi S_\pi)_{n-1}^2}{\rho_\pi^2} \\ \vdots \\ \int_{\Omega} N_{n_{\text{en}}} \,\mathrm{d}\Omega \, \frac{(\rho_\pi S_\pi)_n^{n_{\text{en}}} - (\rho_\pi S_\pi)_{n-1}^{n_{\text{en}}}}{\rho_\pi^{n_{\text{en}}}} \end{Bmatrix} = \frac{\phi_{n-1}}{\Delta t}\frac{V}{n_{\text{en}}}\begin{Bmatrix} \frac{(\rho_\pi S_\pi)_n^1 - (\rho_\pi S_\pi)_{n-1}^1}{\rho_\pi^1} \\ \frac{(\rho_\pi S_\pi)_n^2 - (\rho_\pi S_\pi)_{n-1}^2}{\rho_\pi^2} \\ \vdots \\ \frac{(\rho_\pi S_\pi)_n^{n_{\text{en}}} - (\rho_\pi S_\pi)_{n-1}^{n_{\text{en}}}}{\rho_\pi^{n_{\text{en}}}} \end{Bmatrix}, \quad \pi = \text{l}, \text{g}, \tag{A7}$$

where $N_i$ is the shape function associated with node $i$, $\rho_\pi^i$ is the density of phase $\pi$ at node $i$, and $S_\pi^i$ is the saturation of phase $\pi$ at node $i$. For a finite element with linear geometry, the integral of the shape functions over the element domain is the volume of the element, $V_{\text{e}}$, divided by the number of nodes. This integral represents the area of influence the element has in the cell. For example, it corresponds to the blue area in element 1 associated with the cell of node 6.

The Jacobian associated with this residual is naturally diagonal, since the storage term in node $i$ is only a function of the pore-pressures of that node, as presented in Appendix B.

### A2 Treatment of the advective and volumetric strain terms

We consider the relative permeabilities and the densities to be element-wise variables (Olivella Pastallé, 1995), being calculated using the mean nodal liquid saturation degree.

$$\mathbf{r}_{\pi}^{\text{a,p}} = \frac{k_{\text{r}\pi}(\bar{S}_{\text{l}})}{\mu_\pi}\int_{\Omega} \mathbf{B}_{\text{p}}^{\top}\,\mathbf{k}\,\mathbf{B}_{\text{p}}\,\mathrm{d}\Omega\,\mathbf{p}_\pi = \frac{k_{\text{r}\pi}(\bar{S}_{\text{l}})}{\mu_\pi}\mathbf{K}\,\mathbf{p}_\pi, \quad \pi = \text{l}, \text{g}. \tag{A8}$$

$$\mathbf{r}_{\pi}^{\text{a,g}} = -\frac{k_{\text{r}\pi}(\bar{S}_{\text{l}})}{\mu_\pi}\bar{\rho}_\pi\int_{\Omega} \mathbf{B}_{\text{p}}^{\top}\,\mathbf{k}\,\mathbf{g}\,\mathrm{d}\Omega = -\frac{k_{\text{r}\pi}(\bar{S}_{\text{l}})}{\mu_\pi}\bar{\rho}_\pi\,\mathbf{f}_{\text{G}}, \quad \pi = \text{l}, \text{g}. \tag{A9}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


$$\mathbf{r}^{\mathrm{v}}_{\pi}(t_n) = \bar{S}_{\pi} \int\limits_{\Omega} \mathbf{N}_{\mathrm{p}}^{\top}\, \mathbf{m}^{\top} \mathbf{B}_{\mathrm{u}} \frac{(\mathbf{d}_n - \mathbf{d}_{n-1})}{\Delta t}\, \mathrm{d}\Omega = \bar{S}_{\pi}\, \mathbf{f}_{\mathrm{v}}, \quad \pi = \mathrm{l}, \mathrm{g}, \tag{A10}$$

with,

$$\bar{S}_{\pi} = \frac{1}{n_{\mathrm{en}}} \sum_{i=1}^{n_{\mathrm{en}}} S^{i}_{\pi}, \tag{A11}$$

$$\bar{\rho}_{\pi} = \frac{1}{n_{\mathrm{en}}} \sum_{i=1}^{n_{\mathrm{en}}} \rho^{i}_{\pi}. \tag{A12}$$

## Appendix B: Jacobian matrix of physics `H2M`

The following expressions correspond to the Jacobian of the implemented `H2M` physics after applying the numerical treatments described in Appendix A. The Jacobian matrix of this system is obtained by differentiating the residual equations in Eqs. (31) and (32) with respect to the DOFs:

$$\mathbf{J}^{(k)}_{n} = \begin{bmatrix} \frac{\partial \mathbf{r}_{\mathrm{u}}(t_n)}{\partial \mathbf{d}} & \frac{\partial \mathbf{r}_{\mathrm{u}}(t_n)}{\partial \mathbf{p}_{\mathrm{l}}} & \frac{\partial \mathbf{r}_{\mathrm{u}}(t_n)}{\partial \mathbf{p}_{\mathrm{g}}} \\ \frac{\partial \mathbf{r}_{\mathrm{l}}(t_n)}{\partial \mathbf{d}} & \frac{\partial \mathbf{r}_{\mathrm{l}}(t_n)}{\partial \mathbf{p}_{\mathrm{l}}} & \frac{\partial \mathbf{r}_{\mathrm{l}}(t_n)}{\partial \mathbf{p}_{\mathrm{g}}} \\ \frac{\partial \mathbf{r}_{\mathrm{g}}(t_n)}{\partial \mathbf{d}} & \frac{\partial \mathbf{r}_{\mathrm{g}}(t_n)}{\partial \mathbf{p}_{\mathrm{l}}} & \frac{\partial \mathbf{r}_{\mathrm{g}}(t_n)}{\partial \mathbf{p}_{\mathrm{g}}} \end{bmatrix}^{(k)}. \tag{B1}$$

### B1 Auxiliary derivatives

To define and compute the derivatives that compose the Jacobian matrix, it is useful to define some auxiliary derivatives beforehand. In the following equations, $\pi$ and $\beta$ are used as phase indices, denoting either the liquid (l) or the gas (g) phase.

$$\frac{\partial p_{\pi}}{\partial p_{\beta}} = \delta_{\pi\beta} = \begin{cases} 1, & \pi = \beta \\ 0, & \pi \neq \beta \end{cases}, \tag{B2}$$

$$\frac{\partial p_{\mathrm{c}}}{\partial p_{\mathrm{l}}} = -1, \tag{B3}$$

$$\frac{\partial p_{\mathrm{c}}}{\partial p_{\mathrm{g}}} = 1, \tag{B4}$$

$$\frac{\partial S_{\mathrm{g}}}{\partial p_{\mathrm{c}}} = -\frac{\partial S_{\mathrm{l}}}{\partial p_{\mathrm{c}}}, \tag{B5}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


$$\frac{\partial \bar{S}_{\mathrm{l}}}{\partial \mathbf{p}_{\beta}} = \begin{Bmatrix} \frac{\partial \bar{S}_{\mathrm{l}}}{\partial p_{\beta}^{1}} \\ \frac{\partial \bar{S}_{\mathrm{l}}}{\partial p_{\beta}^{2}} \\ \vdots \\ \frac{\partial \bar{S}_{\mathrm{l}}}{\partial p_{\beta}^{n_{\mathrm{en}}}} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \bar{S}_{\mathrm{l}}}{\partial S_{\mathrm{l}}^{1}} \frac{\partial S_{\mathrm{l}}^{1}}{\partial p_{\mathrm{c}}^{1}} \frac{\partial p_{\mathrm{c}}^{1}}{\partial p_{\beta}^{1}} \\ \frac{\partial \bar{S}_{\mathrm{l}}}{\partial S_{\mathrm{l}}^{2}} \frac{\partial S_{\mathrm{l}}^{2}}{\partial p_{\mathrm{c}}^{2}} \frac{\partial p_{\mathrm{c}}^{2}}{\partial p_{\beta}^{2}} \\ \vdots \\ \frac{\partial \bar{S}_{\mathrm{l}}}{\partial S_{\mathrm{l}}^{n_{\mathrm{en}}}} \frac{\partial S_{\mathrm{l}}^{n_{\mathrm{en}}}}{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}} \frac{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}}{\partial p_{\beta}^{n_{\mathrm{en}}}} \end{Bmatrix} = \frac{1}{n_{\mathrm{en}}} \begin{Bmatrix} \frac{\partial S_{\mathrm{l}}^{1}}{\partial p_{\mathrm{c}}^{1}} \frac{\partial p_{\mathrm{c}}^{1}}{\partial p_{\beta}^{1}} \\ \frac{\partial S_{\mathrm{l}}^{2}}{\partial p_{\mathrm{c}}^{2}} \frac{\partial p_{\mathrm{c}}^{2}}{\partial p_{\beta}^{2}} \\ \vdots \\ \frac{\partial S_{\mathrm{l}}^{n_{\mathrm{en}}}}{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}} \frac{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}}{\partial p_{\beta}^{n_{\mathrm{en}}}} \end{Bmatrix}, \tag{B6}$$

$$\frac{\partial \bar{\rho}_{\pi}}{\partial \mathbf{p}_{\pi}} = \begin{Bmatrix} \frac{\partial \bar{\rho}_{\pi}}{\partial p_{\pi}^{1}} \\ \frac{\partial \bar{\rho}_{\pi}}{\partial p_{\pi}^{2}} \\ \vdots \\ \frac{\partial \bar{\rho}_{\pi}}{\partial p_{\pi}^{n_{\mathrm{en}}}} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \bar{\rho}_{\pi}}{\partial \rho_{\pi}^{1}} \frac{\partial \rho_{\pi}^{1}}{\partial p_{\pi}^{1}} \\ \frac{\partial \bar{\rho}_{\pi}}{\partial \rho_{\pi}^{2}} \frac{\partial \rho_{\pi}^{2}}{\partial p_{\pi}^{2}} \\ \vdots \\ \frac{\partial \bar{\rho}_{\pi}}{\partial \rho_{\pi}^{n_{\mathrm{en}}}} \frac{\partial \rho_{\pi}^{n_{\mathrm{en}}}}{\partial p_{\pi}^{n_{\mathrm{en}}}} \end{Bmatrix} = \frac{1}{n_{\mathrm{en}}} \frac{1}{K_{\pi}} \begin{Bmatrix} \rho_{\pi}^{1} \\ \rho_{\pi}^{2} \\ \vdots \\ \rho_{\pi}^{n_{\mathrm{en}}} \end{Bmatrix}, \tag{B7}$$

$$\frac{\partial}{\partial \mathbf{p}_{\beta}} \begin{Bmatrix} \frac{(\rho_{\pi} S_{\pi})_{n}^{1} - (\rho_{\pi} S_{\pi})_{n-1}^{1}}{\rho_{\pi}^{1}} \\ \frac{(\rho_{\pi} S_{\pi})_{n}^{2} - (\rho_{\pi} S_{\pi})_{n-1}^{2}}{\rho_{\pi}^{2}} \\ \vdots \\ \frac{(\rho_{\pi} S_{\pi})_{n}^{n_{\mathrm{en}}} - (\rho_{\pi} S_{\pi})_{n-1}^{n_{\mathrm{en}}}}{\rho_{\pi}^{n_{\mathrm{en}}}} \end{Bmatrix} = \begin{bmatrix} \frac{\partial S_{\pi}^{1}}{\partial p_{\mathrm{c}}^{1}} \frac{\partial p_{\mathrm{c}}^{1}}{\partial p_{\beta}^{1}} + \delta_{\pi\beta} \frac{(\rho_{\pi} S_{\pi})_{n-1}^{1}}{\rho_{\pi}^{1} K_{\pi}} & 0 & \dots & 0 \\ 0 & \frac{\partial S_{\pi}^{2}}{\partial p_{\mathrm{c}}^{2}} \frac{\partial p_{\mathrm{c}}^{2}}{\partial p_{\beta}^{2}} + \delta_{\pi\beta} \frac{(\rho_{\pi} S_{\pi})_{n-1}^{2}}{\rho_{\pi}^{2} K_{\pi}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial S_{\pi}^{n_{\mathrm{en}}}}{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}} \frac{\partial p_{\mathrm{c}}^{n_{\mathrm{en}}}}{\partial p_{\beta}^{n_{\mathrm{en}}}} + \delta_{\pi\beta} \frac{(\rho_{\pi} S_{\pi})_{n-1}^{n_{\mathrm{en}}}}{\rho_{\pi}^{n_{\mathrm{en}}} K_{\pi}} \end{bmatrix}. \tag{B8}$$

### B2 Derivatives of the mechanical equilibrium equation residual

The derivative of Eq. (31) with respect to the displacement DOF vector is

$$\frac{\partial \mathbf{r}_{\mathrm{u}}}{\partial \mathbf{d}} = \int_{\Omega} \mathbf{B}_{\mathrm{u}}^{\top} \mathbf{D} \mathbf{B}_{\mathrm{u}} \, \mathrm{d}\Omega, \tag{B9}$$

where $\mathbf{D}$ is the tangent constitutive matrix.

The derivative with respect to the pore-pressure DOFs vector is

$$\frac{\partial \mathbf{r}_{\mathrm{u}}}{\partial \mathbf{p}_{\beta}} = \mathbf{Q} \mathbf{p}_{\mathrm{c}} \left( \frac{\partial \bar{S}_{\mathrm{l}}}{\partial \mathbf{p}_{\beta}} \right)^{\top} - \delta_{\mathrm{l}\beta} \bar{S}_{\mathrm{l}} \mathbf{Q} - \delta_{\mathrm{g}\beta} \bar{S}_{\mathrm{g}} \mathbf{Q}, \tag{B10}$$

with,

$$\mathbf{Q} = \int_{\Omega} \mathbf{B}_{\mathrm{u}}^{\top} \mathbf{m} \mathbf{N}_{\mathrm{p}} \, \mathrm{d}\Omega. \tag{B11}$$

### B3 Derivatives of the fluid continuity equation residuals

The derivative of Eq. (32) with respect to the displacement DOF vector is

$$\frac{\partial \mathbf{r}_{\pi}}{\partial \mathbf{d}} = \frac{\partial \mathbf{r}_{\pi}^{\mathrm{v}}}{\partial \mathbf{d}} = \frac{\bar{S}_{\pi}}{\Delta t} \mathbf{Q}^{\top}. \tag{B12}$$

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


715 The derivative of Eq. (32) with respect to the pore-pressure DOFs vector is

$$\frac{\partial \mathbf{r}_\pi}{\partial \mathbf{p}_\beta} = \frac{\partial \mathbf{r}^{\mathrm{m}}_\pi}{\partial \mathbf{p}_\beta} + \frac{\partial \mathbf{r}^{\mathrm{a,p}}_\pi}{\partial \mathbf{p}_\beta} + \frac{\partial \mathbf{r}^{\mathrm{a,g}}_\pi}{\partial \mathbf{p}_\beta} + \frac{\partial \mathbf{r}^{\mathrm{v}}_\pi}{\partial \mathbf{p}_\beta}, \tag{B13}$$

with,

$$\frac{\partial \mathbf{r}^{\mathrm{m}}_\pi}{\partial \mathbf{p}_\beta} = \frac{\phi_{n-1}}{\Delta t}\frac{V}{N}\begin{bmatrix} \frac{\partial S^1_\pi}{\partial p^1_{\mathrm{c}}}\frac{\partial p^1_{\mathrm{c}}}{\partial p^1_\beta} + \delta_{\pi\beta}\frac{(\rho_\pi S_\pi)^1_{n-1}}{\rho^1_\pi K_\pi} & 0 & \dots & 0 \\ 0 & \frac{\partial S^2_\pi}{\partial p^2_{\mathrm{c}}}\frac{\partial p^2_{\mathrm{c}}}{\partial p^2_\beta} + \delta_{\pi\beta}\frac{(\rho_\pi S_\pi)^2_{n-1}}{\rho^2_\pi K_\pi} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial S^{n_{\mathrm{en}}}_\pi}{\partial p^{n_{\mathrm{en}}}_{\mathrm{c}}}\frac{\partial p^{n_{\mathrm{en}}}_{\mathrm{c}}}{\partial p^{n_{\mathrm{en}}}_\beta} + \delta_{\pi\beta}\frac{(\rho_\pi S_\pi)^{n_{\mathrm{en}}}_{n-1}}{\rho^{n_{\mathrm{en}}}_\pi K_\pi} \end{bmatrix}, \tag{B14}$$

720
$$\frac{\partial \mathbf{r}^{\mathrm{a,p}}_\pi}{\partial \mathbf{p}_\beta} = \frac{1}{\mu_\pi}\left(\mathbf{K}\,\mathbf{p}_\pi\left(\frac{\partial k_{\mathrm{r}\pi}(\bar{S}_{\mathrm{l}})}{\partial \bar{S}_{\mathrm{l}}}\frac{\partial \bar{S}_{\mathrm{l}}}{\partial \mathbf{p}_\beta}\right)^{\top} + \delta_{\pi\beta}\,k_{\mathrm{r}\pi}(\bar{S}_{\mathrm{l}})\,\mathbf{K}\right), \tag{B15}$$

$$\frac{\partial \mathbf{r}^{\mathrm{a,g}}_\pi}{\partial \mathbf{p}_\beta} = -\frac{1}{\mu_\pi}\mathbf{f}_{\mathrm{G}}\left(\frac{\partial k_{\mathrm{r}\pi}(\bar{S}_{\mathrm{l}})}{\partial \bar{S}_{\mathrm{l}}}\frac{\partial \bar{S}_{\mathrm{l}}}{\partial \mathbf{p}_\beta}\bar{\rho}_\pi + \delta_{\pi\beta}\,k_{\mathrm{r}\pi}(\bar{S}_{\mathrm{l}})\frac{\partial \bar{\rho}_\pi}{\partial \mathbf{p}_\pi}\right)^{\top}, \tag{B16}$$

$$\frac{\partial \mathbf{r}^{\mathrm{v}}_\pi}{\partial \mathbf{p}_\beta} = \mathbf{f}_{\mathrm{v}}\left(\frac{\partial \bar{S}_\pi}{\partial \mathbf{p}_\beta}\right)^{\top}. \tag{B17}$$

725 *Author contributions.* DC was responsible for conceptualization, software development, implementation, verification, and writing the original draft. RR contributed to software development and writing the original draft. CM, SO, VV and LFM contributed to reviewing and editing the manuscript. DR, IP and GC were responsible for funding acquisition, project administration, and supervision, and also contributed to reviewing and editing the manuscript.

*Competing interests.* The authors declare that they have no conflict of interest.

730 *Acknowledgements.* The authors acknowledge the financial support from the grant TED2021-130510A-I00 funded by the “European Union NextGenerationEU/PRTR” and by MCIN/AEI/10.13039/501100011033. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors also thank Xavier Tort for testing early versions of `PorousLab` during his master’s studies.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


## References

Ajayi, T., Gomes, J. S., and Bera, A.: A review of CO2 storage in geological formations emphasizing modeling, monitoring and capacity estimation approaches, Petroleum Science, 16, 1028–1063, 2019.

Alonso, M., Vaunat, J., Vu, M.-N., Talandier, J., Olivella, S., and Gens, A.: Three-dimensional modelling of a large-diameter sealing concept in a deep geological radioactive waste disposal, Rock Mechanics and Rock Engineering, 57, 4133–4158, 2024.

Arroyo, M. and Gens, A.: Computational analyses of Dam I failure at the Corrego de Feijao mine in Brumadinho, Final Report for VALE SA, 2021.

Babuška, I. and Melenk, J. M.: The Partition of Unity Method, International Journal for Numerical Methods in Engineering, 40, 727–758, https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N, 1997.

Barenblatt, G. I., Zheltov, I. P., and Kochina, I. N.: Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks, Journal of Applied Mathematics and Mechanics, 24, 1286–1303, https://doi.org/10.1016/0021-8928(60)90107-6, 1960.

Batoz, J.-L. and Dhatt, G.: Incremental displacement algorithms for nonlinear problems, International Journal for Numerical Methods in Engineering, 14, 1262–1267, 1979.

Belytschko, T., Moës, N., Usui, S., and Parimi, C.: Arbitrary discontinuities in finite elements, International Journal for Numerical Methods in Engineering, 50, 993–1013, https://doi.org/10.1002/1097-0207(20010210)50:4<993::AID-NME164>3.0.CO;2-M, 2001.

Berkowitz, B.: Characterizing flow and transport in fractured geological media: A review, Advances in Water Resources, 25, 861–884, https://doi.org/10.1016/S0309-1708(02)00042-8, 2002.

Berre, I., Doster, F., and Keilegavlen, E.: Flow in fractured porous media: A review of conceptual models and discretization approaches, Transport in Porous Media, 130, 215–236, 2019.

Birkholzer, J. T., Tsang, C.-F., Bond, A. E., Hudson, J. A., Jing, L., and Stephansson, O.: 25 years of DECOVALEX-Scientific advances and lessons learned from an international research collaboration in coupled subsurface processes, International Journal of Rock Mechanics and Mining Sciences, 122, 103 995, 2019.

Booch, G.: The Unified Modeling Language User Guide, Pearson Education India, 2005.

Březina, J. and Stebel, J.: Analysis of Model Error for a Continuum-Fracture Model of Porous Media Flow, in: High Performance Computing in Science and Engineering, pp. 152–160, https://doi.org/10.1007/978-3-319-40361-8_11, 2016.

Cacace, M. and Jacquey, A. B.: Flexible parallel implicit modelling of coupled thermal–hydraulic–mechanical processes in fractured rocks, Solid Earth, 8, 921–941, 2017.

Cavalcanti, D.: PorousLab v1.1.0, Zenodo, https://doi.org/10.5281/zenodo.15624961, software, version v1.1.0, MIT license, 2026.

Cavalcanti, D., Mejia, C., Roehl, D., de Pouplana, I., Casas, G., and Martha, L. F.: Embedded finite element formulation for fluid flow in fractured porous medium, Computers and Geotechnics, 171, 106 384, https://doi.org/10.1016/j.compgeo.2024.106384, 2024a.

Cavalcanti, D., Mejia, C., Roehl, D., de Pouplana, I., and Oñate, E.: Hydromechanical embedded finite element for conductive and impermeable strong discontinuities in porous media, Computers and Geotechnics, 172, 106 427, https://doi.org/10.1016/j.compgeo.2024.106427, 2024b.

Cavalcanti, D., Mejia, C., Souza, C., Mendes, C. A., de Pouplana, I., Casas, G., and Roehl, D.: Behavior of cohesive stresses in embedded finite elements based on the strong discontinuity approach, Finite Elements in Analysis and Design, 253, 104 485, 2026.

Cavalcanti, D. B., Rangel, R. L., and Martha, L. F.: Numerical experiments to assess the performance of different formulations and solution algorithms for geometrically nonlinear analysis of two-dimensional frames, in: XLIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE), Foz do Iguaçu/PR, Brazil, 2022.

Cazes, F., Meschke, G., and Zhou, M.-M.: Strong discontinuity approaches: An algorithm for robust performance and comparative assessment of accuracy, International Journal of Solids and Structures, 96, 355–379, 2016.

Cerfontaine, B., Dieudonné, A.-C., Radu, J.-P., Collin, F., and Charlier, R.: 3D zero-thickness coupled interface finite element: Formulation and application, Computers and Geotechnics, 69, 124–140, 2015.

Cervera, M., Barbat, G., Chiumenti, M., and Wu, J.-Y.: A comparative review of XFEM, mixed FEM and phase-field models for quasi-brittle cracking, Archives of Computational Methods in Engineering, 29, 1009–1083, 2022.

COMSOL AB: COMSOL Multiphysics®, Software, https://www.comsol.com/comsol-multiphysics, 2026.

Cusini, M., White, J. A., Castelletto, N., and Settgast, R. R.: Simulation of coupled multiphase flow and geomechanics in porous media with embedded discrete fractures, International Journal for Numerical and Analytical Methods in Geomechanics, 45, 563–584, https://doi.org/10.1002/nag.3168, 2021.

Dadvand, P., Rossi, R., and Oñate, E.: An object-oriented environment for developing finite element codes for multi-disciplinary applications, Archives of Computational Methods in Engineering, 17, 253–297, https://doi.org/10.1007/s11831-010-9045-2, 2010.

Dassault Systèmes: Abaqus, Software, https://www.3ds.com/products/simulia/abaqus, 2026.

de Borst, R., Wells, G., and Sluys, L.: Some observations on embedded discontinuity models, Engineering Computations, 18, 241–254, https://doi.org/10.1108/02644400110365897, publisher: MCB UP Ltd, 2001.

de Pouplana, I. and Oñate, E.: A FIC-based stabilized mixed finite element method with equal order interpolation for solid–pore fluid interaction problems, International Journal for Numerical and Analytical Methods in Geomechanics, 41, 110–134, https://doi.org/10.1002/nag.2550, 2017.

de Pouplana, I. and Oñate, E.: Finite element modelling of fracture propagation in saturated media using quasi-zero-thickness interface elements, Computers and Geotechnics, 96, 103–117, https://doi.org/10.1016/j.compgeo.2017.10.016, 2018.

de Souza Neto, E. A., Peric, D., and Owen, D. R.: Computational Methods for Plasticity: Theory and Applications, John Wiley & Sons, 2011.

Dias, W., Roehl, D., Mejia, C., and Sotomayor, P.: Cavern integrity for underground hydrogen storage in the Brazilian pre-salt fields, International Journal of Hydrogen Energy, 48, 26 853–26 869, 2023.

Dias-da Costa, D., Alfaiate, J., Sluys, L., and Júlio, E.: A discrete strong discontinuity approach, Engineering Fracture Mechanics, 76, 1176–1201, 2009a.

Dias-da Costa, D., Alfaiate, J., Sluys, L. J., and Júlio, E.: Towards a generalization of a discrete strong discontinuity approach, Computer Methods in Applied Mechanics and Engineering, 198, 3670–3681, 2009b.

Dolbow, J., Moës, N., and Belytschko, T.: Discontinuous enrichment in finite elements with a partition of unity method, Finite Elements in Analysis and Design, 36, 235–260, https://doi.org/10.1016/S0168-874X(00)00035-4, 2000.

Duarte, C. A. and Oden, J. T.: An *h-p* adaptive method using clouds, Computer Methods in Applied Mechanics and Engineering, 139, 237–262, https://doi.org/10.1016/S0045-7825(96)01085-7, 1996.

Fabbri, H., Sánchez, M., Maedo, M., Cleto, P., and Manzoli, O.: Modeling gas breakthrough and flow phenomena through engineered barrier systems using a discrete fracture approach, Computers and Geotechnics, 154, 105 148, https://doi.org/10.1016/j.compgeo.2022.105148, 2023.

Firme, P. A., Roehl, D., Mejia, C., and Romanel, C.: Geomechanics of salt towards natural barriers for the abandonment of pre-salt wells in Brazil, Geoenergy Science and Engineering, 228, 211 849, 2023.

Firme, P. A., Roehl, D., Mejia, C., and Romanel, C.: An assessment of the salt caprock creep impact on Pre-salt reservoir geomechanics, Geomechanics for Energy and the Environment, p. 100588, 2024.

Fjær, E., Holt, R. M., Horsrud, P., and Raaen, A. M.: Petroleum Related Rock Mechanics, Elsevier, ISBN 978-0-08-055709-0, 2008.

Flemisch, B., Darcis, M., Erbertseder, K., Faigle, B., Lauser, A., Mosthaf, K., Müthing, S., Nuske, P., Tatomir, A., Wolff, M., et al.: DuMux: DUNE for multi-{phase, component, scale, physics,...} flow and transport in porous media, Advances in Water Resources, 34, 1102–1112, 2011.

Fumagalli, A., Pasquale, L., Zonca, S., and Micheletti, S.: An upscaling procedure for fractured reservoirs with embedded grids, Water Resources Research, 52, 6506–6525, 2016.

Fumagalli, A., Keilegavlen, E., and Scialò, S.: Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations, Journal of Computational Physics, 376, 694–712, 2019.

Fučík, J.: Exact and semi-analytical solutions of immiscible two-phase flow equations in porous media, https://mmg.fjfi.cvut.cz/~fucik/index.php?page=exact, accessed: July 31, 2025, 2021.

Giudicelli, G., Lindsay, A., Harbour, L., Icenhour, C., Li, M., Hansel, J. E., German, P., Behne, P., Marin, O., Stogner, R. H., et al.: 3.0-MOOSE: Enabling massively parallel multiphysics simulations, SoftwareX, 26, 101 690, https://doi.org/10.1016/j.softx.2024.101690, 2024.

Grunwald, N., Lehmann, C., Maßmann, J., Naumov, D., Kolditz, O., and Nagel, T.: Non-isothermal two-phase flow in deformable porous media: systematic open-source implementation and verification procedure, Geomechanics and Geophysics for Geo-Energy and Geo-Resources, 8, 1–29, https://doi.org/10.1007/s40948-022-00394-2, 2022.

Hammond, G. E., Lichtner, P. C., and Mills, R.: Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN, Water Resources Research, 50, 208–228, https://doi.org/10.1002/2012WR013483, 2014.

Helmig, R. et al.: Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems, vol. 1, Springer, 1997.

Hirmand, M., Vahab, M., and Khoei, A. R.: An augmented Lagrangian contact formulation for frictional discontinuities with the extended finite element method, Finite Elements in Analysis and Design, 107, 28–43, https://doi.org/10.1016/j.finel.2015.08.003, 2015.

Hosseini, S. M. and Khoei, A. R.: Numerical simulation of proppant transport and tip screen-out in hydraulic fracturing with the extended finite element method, International Journal of Rock Mechanics and Mining Sciences, 128, 104 247, https://doi.org/10.1016/j.ijrmms.2019.104247, 2020.

HosseiniMehr, M., Tomala, J. P., Vuik, C., Al Kobaisi, M., and Hajibeygi, H.: Projection-based embedded discrete fracture model (pEDFM) for flow and heat transfer in real-field geological formations with hexahedral corner-point grids, Advances in Water Resources, 159, 104 091, 2022.

Jing, L.: A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering, International Journal of Rock Mechanics and Mining Sciences, 40, 283–353, 2003.

Jirásek, M.: Comparative study on finite elements with embedded discontinuities, Computer methods in applied mechanics and engineering, 188, 307–330, 2000.

Karthik, A., Manideep, R., and Chavda, J. T.: Sensitivity analysis of slope stability using finite element method, Innovative Infrastructure Solutions, 7, 184, 2022.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026



Keilegavlen, E., Berge, R., Fumagalli, A., Starnoni, M., Stefansson, I., Varela, J., and Berre, I.: PorePy: an open-source software for simulation of multiphysics processes in fractured porous media, Computational Geosciences, 25, 1–23, https://doi.org/10.1007/s10596-020-10002-5, 2021.

Khoei, A. R., Hosseini, N., and Mohammadnejad, T.: Numerical modeling of two-phase fluid flow in deformable fractured porous media using the extended finite element method and an equivalent continuum model, Advances in Water Resources, 94, 510–528, https://doi.org/10.1016/j.advwatres.2016.02.017, 2015.

Kivi, I. R., Vilarrasa, V., Kim, K.-I., Yoo, H., and Min, K.-B.: Bleed-off control on post-injection seismicity in enhanced geothermal systems, Underground Space, 22, 21–38, 2025.

Koch, T., Gläser, D., Weishaupt, K., Ackermann, S., Beck, M., Becker, B., Burbulla, S., Class, H., Coltman, E., Emmert, S., et al.: DuMux 3–An open-source simulator for solving flow and transport problems in porous media with a focus on model coupling, Computers & Mathematics with Applications, 81, 423–443, https://doi.org/10.1016/j.camwa.2020.02.012, 2021.

Kolditz, O., Bauer, S., Bilke, L., Böttcher, N., Delfs, J.-O., Fischer, T., Görke, U. J., Kalbacher, T., Kosakowski, G., McDermott, C., 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, https://doi.org/10.1007/s12665-012-1546-x, 2012.

Kolditz, O., Görke, U.-J., Shao, H., Wang, W., and Bauer, S.: Thermo-hydro-mechanical chemical processes in fractured porous media: modelling and benchmarking, vol. 25, Springer, 2016.

Krogstad, S., Lie, K.-A., Møyner, O., Nilsen, H. M., Raynaud, X., and Skaflestad, B.: MRST-AD–An open-source framework for rapid prototyping and evaluation of reservoir simulation problems, in: SPE Reservoir Simulation Conference, p. D022S002R004, https://doi.org/10.2118/173317-MS, 2015.

Leon, S. E., Paulino, G. H., Pereira, A., Menezes, I. F. M., and Lages, E. N.: A Unified Library of Nonlinear Solution Schemes, Applied Mechanics Reviews, 64, 040 803, https://doi.org/10.1115/1.4006992, 2012.

Li, C. and Laloui, L.: Coupled thermo-hydro-mechanical effects on caprock stability during carbon dioxide injection, Energy Procedia, 114, 3202–3209, 2017.

Lie, K.-A.: An introduction to reservoir simulation using MATLAB/GNU Octave: User guide for the MATLAB Reservoir Simulation Toolbox (MRST), Cambridge University Press, 2019.

Lie, K.-A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M., and Skaflestad, B.: Open-source MATLAB implementation of consistent discretisations on complex grids, Computational Geosciences, 16, 297–322, 2012.

Linder, C. and Armero, F.: Finite elements with embedded strong discontinuities for the modeling of failure in solids, International Journal for Numerical Methods in Engineering, 72, 1391–1433, 2007.

Manzoli, O. L., Cleto, P. R., Sánchez, M., Guimarães, L. J., and Maedo, M. A.: On the use of high aspect ratio finite elements to model hydraulic fracturing in deformable porous media, Computer Methods in Applied Mechanics and Engineering, 350, 57–80, https://doi.org/10.1016/j.cma.2019.03.006, 2019.

Martha, L., Menezes, I., Lages, E., Parente Jr, E., and Pitangueira, R.: An OOP class organization for materially nonlinear finite element analysis, in: Joint Conference of Italian Group of Computational Mechanics and Ibero-Latin American Association of Computational Methods in Engineering, Padova, Italy, pp. 229–232, 1996.

Martha, L. F. and Parente Jr, E.: An object-oriented framework for finite element programming, in: World Congress on Computational Mechanics, 2002.

McWhorter, D. B. and Sunada, D. K.: Exact integral solutions for two-phase flow, Water Resources Research, 26, 399–413, 1990.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


Mejia, C., Roehl, D., Rueda, J., and Quevedo, R.: A new approach for modeling three-dimensional fractured reservoirs with embedded complex fracture networks, Computers and Geotechnics, 130, https://doi.org/10.1016/j.compgeo.2020.103928, 2021.

Mendes, C. A., Gattass, M., and Roehl, D.: The GeMA framework–An innovative framework for the development of multiphysics and multiscale simulations, in: VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Proceedia), vol. 2383, pp. 7886–7894, https://doi.org/10.7712/100016.2383.6771, 2016.

Moës, N., Dolbow, J., and Belytschko, T.: A finite element method for crack growth without remeshing, International Journal for Numerical Methods in Engineering, 46, 131–150, https://doi.org/10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J, 1999.

Møyner, O.: JutulDarcy.jl – a Fully Differentiable High-Performance Reservoir Simulator based on Automatic Differentiation, in: ECMOR 2024, pp. 1–37, https://doi.org/10.3997/2214-4609.202437111, 2024.

Novikov, A. V., Shokrollahzadeh Behbahani, S., Voskov, D., Hajibeygi, H., and Jansen, J. D.: Benchmarking Analytical and Numerical Simulation of Induced Fault Slip, in: 57th U.S. Rock Mechanics/Geomechanics Symposium, U.S. Rock Mechanics/Geomechanics Symposium, pp. ARMA–2023–0695, https://doi.org/10.56952/ARMA-2023-0695, 2023.

Oberhollenzer, S., Tschuchnigg, F., and Schweiger, H. F.: Finite element analyses of slope stability problems using non-associated plasticity, Journal of Rock Mechanics and Geotechnical Engineering, 10, 1091–1101, 2018.

Ogata, S., Yasuhara, H., Kinoshita, N., Inui, T., Nishira, E., and Kishida, K.: Numerical analyses of coupled thermal–hydraulic–mechanical–chemical processes for estimating permeability change in fractured rock induced by alkaline solution, Geomechanics for Energy and the Environment, 31, 100 372, 2022.

Olivella, S., Carrera, J., Gens, A., and Alonso, E. E.: Nonisothermal multiphase flow of brine and gas through saline media, Transport in Porous Media, 15, 271–293, https://doi.org/10.1007/BF00613282, 1994.

Olivella, S., Gens, A., Carrera, J., and Alonso, E.: Numerical formulation for a simulator (CODE_BRIGHT) for the coupled analysis of saline media, Engineering Computations, 13, 87–112, https://doi.org/10.1108/02644409610151575, 1996.

Olivella Pastallé, S.: Nonisothermal Multiphase Flow of Brine and Gas through Saline Media, Phd thesis, Universitat Politècnica de Catalunya, Departament d'Enginyeria del Terreny, Cartogràfica i Geofísica, ISBN 9788469263433, https://doi.org/10.5821/dissertation-2117-93573, tesi doctoral, 1995.

Oliver, J.: Modelling strong discontinuities in solid mechanics via strain softening constitutive equations. Part II: Numerical simulation, International Journal for Numerical Methods in Engineering, 39, 3601–3623, https://doi.org/10.1002/(SICI)1097-0207(19961115)39:21<3601::AID-NME64>3.0.CO;2-4, 1996.

Oliver, J., Huespe, A. E., and Samaniego, E.: A study on finite elements for capturing strong discontinuities, International journal for numerical methods in engineering, 56, 2135–2161, 2003.

Olorode, O., Wang, B., and Rashid, H. U.: Three-dimensional projection-based embedded discrete-fracture model for compositional simulation of fractured reservoirs, SPE Journal, 25, 2143–2161, 2020.

PFLOTRAN: http://www.pflotran.org, (Accessed: 24-Mar-2025), 2025.

Pitz, M., Grunwald, N., Graupner, B., Kurgyis, K., Radeisen, E., Maßmann, J., Ziefle, G., Thiedau, J., and Nagel, T.: Benchmarking a new TH2M implementation in OGS-6 with regard to processes relevant for nuclear waste disposal, Environmental Earth Sciences, 82, 319, https://doi.org/10.1007/s12665-023-10971-7, 2023.

Podgorney, R., Finnila, A., Simmons, S., and McLennan, J.: A reference thermal-hydrologic-mechanical native state model of the Utah FORGE enhanced geothermal site, Energies, 14, 4758, 2021.

Raguenel, M., Lopez, D., Borgese, C., and Li, W.-C.: Use of Locally Refined Unstructured Meshes for $CO_2$ Storage Flow and Poromechanical Simulations, in: Proceedings of the SPE Reservoir Simulation Conference 2025, p. D021S007R005, Society of Petroleum Engineers, Society of Petroleum Engineers, Galveston, Texas, USA, ISBN 979-8-3313-1960-1, https://doi.org/10.2118/223847-MS, 2025.

Ramm, E.: Strategies for tracing the nonlinear response near limit points, in: Nonlinear Finite Element Analysis in Structural Mechanics, pp. 63–89, https://doi.org/10.1007/978-3-642-81589-8_5, 1981.

Rangel, R. L.: Educational Tool for Structural Analysis of Plane Frame Models with Geometric Nonlinearity, Ph.D. thesis, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), https://doi.org/10.17771/PUCRio.acad.47858, 2019.

Rangel, R. L. and Martha, L. F.: Implementation of a user-controlled structural analysis module with geometric nonlinearity, in: 25th International Congress of Mechanical Engineering (COBEM), Uberlândia/MG, Brazil, 2019a.

Rangel, R. L. and Martha, L. F.: LESM—An object-oriented MATLAB program for structural analysis of linear element models, Computer Applications in Engineering Education, 27, 553–571, https://doi.org/10.1002/cae.22097, 2019b.

Rao, X., Cheng, L., Cao, R., Jia, P., Liu, H., and Du, X.: A modified projection-based embedded discrete fracture model (pEDFM) for practical and accurate numerical simulation of fractured reservoir, Journal of Petroleum Science and Engineering, 187, 106 852, 2020.

Rasmussen, A. F., Sandve, T. H., Bao, K., Lauser, A., Hove, J., Skaflestad, B., Klöfkorn, R., Blatt, M., Rustad, A. B., Sævareid, O., et al.: The open porous media flow reservoir simulator, Computers & Mathematics with Applications, 81, 159–185, https://doi.org/10.1016/j.camwa.2020.05.014, 2021.

Renard, P. and Ababou, R.: Equivalent permeability tensor of heterogeneous media: Upscaling methods and criteria (review and analyses), Geosciences, 12, 269, https://doi.org/10.3390/geosciences12070269, 2022.

Rueda, J., Mejia, C., Noreña, N., and Roehl, D.: A three-dimensional enhanced dual-porosity and dual-permeability approach for hydromechanical modeling of naturally fractured rocks, International Journal for Numerical Methods in Engineering, 122, 1663–1686, 2021.

Rutqvist, J.: The geomechanics of CO2 storage in deep sedimentary formations, Geotechnical and Geological Engineering, 30, 525–551, 2012.

Rutqvist, J., Wu, Y.-S., Tsang, C.-F., and Bodvarsson, G.: A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock, International Journal of Rock Mechanics and Mining Sciences, 39, 429–442, https://doi.org/10.1016/S1365-1609(02)00022-9, 2002.

Rutqvist, J., Cappa, F., Rinaldi, A. P., and Godano, M.: Modeling of induced seismicity and ground vibrations associated with geologic CO2 storage, and assessing their effects on surface structures and human perception, International Journal of Greenhouse Gas Control, 24, 64–77, 2014.

Rutqvist, J., Rinaldi, A. P., Cappa, F., Jeanne, P., Mazzoldi, A., Urpi, L., Guglielmi, Y., and Vilarrasa, V.: Fault activation and induced seismicity in geological carbon storage–Lessons learned from recent modeling studies, Journal of Rock Mechanics and Geotechnical Engineering, 8, 789–804, 2016.

Sánchez-Vila, X., Girardi, J. P., and Carrera, J.: A synthesis of approaches to upscaling of hydraulic conductivities, Water Resources Research, 31, 867–882, https://doi.org/10.1029/94WR02754, 1995.

Schrefler, B. A. and Scotta, R.: A fully coupled dynamic model for two-phase fluid flow in deformable porous media, Computer Methods in Applied Mechanics and Engineering, 190, 3223–3246, https://doi.org/10.1016/S0045-7825(00)00390-X, 2001.

Schrefler, B. A., Simoni, L., Xikui, L., and Zienkiewicz, O. C.: Mechanics of Partially Saturated Porous Media, in: Numerical Methods and Constitutive Modelling in Geomechanics, pp. 169–209, Springer, Vienna, ISBN 978-3-7091-2832-9, https://doi.org/10.1007/978-3-7091-2832-9_2, iSSN: 2309-3706, 1990.

Segura, J. M. and Carol, I.: On zero-thickness interface elements for diffusion problems, International Journal for Numerical and Analytical Methods in Geomechanics, 28, 947–962, https://doi.org/10.1002/nag.358, 2004.

Segura, J. M. and Carol, I.: Coupled HM analysis using zero-thickness interface elements with double nodes. Part I: Theoretical model, International Journal for Numerical and Analytical Methods in Geomechanics, 32, 2083–2101, https://doi.org/10.1002/nag.735, 2008.

Settgast, R. R., Aronson, R. M., Besset, J. R., Borio, A., Bui, Q. M., Byer, T. J., Castelletto, N., Citrain, A., Corbett, B. C., Corbett, J., et al.: GEOS: A performance portable multi-physics simulation framework for subsurface applications, Journal of Open Source Software, 9, 6973, https://doi.org/10.21105/joss.06973, 2024.

Simo, J. C. and Rifai, M. S.: A class of mixed assumed strain methods and the method of incompatible modes, International Journal for Numerical Methods in Engineering, 29, 1595–1638, https://doi.org/10.1002/nme.1620290802, 1990.

Simo, J. C., Oliver, J., and Armero, F.: An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids, Computational mechanics, 12, 277–296, 1993.

Smith, G. D.: Numerical solution of partial differential equations: finite difference methods, Oxford university press, 1985.

Song, Y., Jun, S., Na, Y., Kim, K., Jang, Y., and Wang, J.: Geomechanical challenges during geological CO2 storage: A review, Chemical Engineering Journal, 456, 140 968, 2023.

Tangirala, S. K., Parisio, F., Vaezi, I., and Vilarrasa, V.: Effectiveness of injection protocols for hydraulic stimulation in enhanced geothermal systems, Geothermics, 120, 103 018, 2024.

Ţene, M., Bosma, S. B., Al Kobaisi, M. S., and Hajibeygi, H.: Projection-based embedded discrete fracture model (pEDFM), Advances in Water Resources, 105, 205–216, 2017.

Vaezi, I., Parisio, F., Yoshioka, K., Alcolea, A., Meier, P., Carrera, J., Olivella, S., and Vilarrasa, V.: Implicit hydromechanical representation of fractures using a continuum approach, International Journal of Rock Mechanics and Mining Sciences, 183, 105 916, 2024.

Vaezi, I., Yoshioka, K., De Simone, S., Gómez-Castro, B. M., Paluszny, A., Jalali, M., Berre, I., Rutqvist, J., Min, K.-B., Lei, Q., et al.: A review of thermo-hydro-mechanical modeling of coupled processes in fractured rock: From continuum to discontinuum perspective, Journal of Rock Mechanics and Geotechnical Engineering, 2025.

Vilarrasa, V., Carrera, J., Olivella, S., Rutqvist, J., and Laloui, L.: Induced seismicity in geologic carbon storage, Solid Earth, 10, 871–892, 2019.

Voskov, D., Saifullin, I., Novikov, A., Wapperom, M., Orozco, L., Seabra, G. S., Chen, Y., Khait, M., Lyu, X., Tian, X., de Hoop, S., and Palha, A.: open Delft Advanced Research Terra Simulator (open-DARTS), Journal of Open Source Software, 9, 6737, https://doi.org/10.21105/joss.06737, 2024.

Vu, M., Plúa, C., Armand, G., Bumbieler, F., De Lesquen, C., Souley, M., Alonso, M., Vaunat, J., Gens, A., Yu, Z., et al.: Thermo-hydro-mechanical responses of a high-level radioactive waste repository: Effects of short-and long-term nonlinear behavior of the host rock, International Journal of Rock Mechanics and Mining Sciences, 195, 106 281, 2025a.

Vu, M.-N., Armand, G., Zghondi, J., Souley, M., Renaud, V., Alonso, M., Vaunat, J., Gens, A., Yu, Z., Shao, J. F., et al.: A comparative analysis of different approaches to large-scale modeling for a radioactive waste repository in the Callovo-Oxfordian claystone and sensitivity analysis, Rock Mechanics and Rock Engineering, pp. 1–23, 2025b.

Wang, W. and Kolditz, O.: Object-oriented finite element analysis of thermo-hydro-mechanical (THM) problems in porous media, International Journal for Numerical Methods in Engineering, 69, 162–201, https://doi.org/10.1002/nme.1770, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1770, 2007.

https://doi.org/10.5194/egusphere-2026-3322
Preprint. Discussion started: 31 July 2026


Warren, J. E. and Root, P. J.: The behavior of naturally fractured reservoirs, Society of Petroleum Engineers Journal, 3, 245–255, https://doi.org/10.2118/426-PA, 1963.

Watanabe, N., Wang, W., Taron, J., Görke, U., and Kolditz, O.: Lower-dimensional interface elements with local enrichment: application to coupled hydro-mechanical problems in discretely fractured porous media, International Journal for Numerical Methods in Engineering, 90, 1010–1034, 2012.

Wells, G. and Sluys, L.: Application of embedded discontinuities for softening solids, Engineering fracture mechanics, 65, 263–281, 2000.

White, M., Fu, P., McClure, M., Danko, G., Elsworth, D., Sonnenthal, E., Kelkar, S., and Podgorney, R.: A suite of benchmark and challenge problems for enhanced geothermal systems, Geomechanics and Geophysics for Geo-Energy and Geo-Resources, 4, 79–117, https://doi.org/10.1007/s40948-017-0076-0, 2018.

Wijesinghe, A.: Exact similarity solution for coupled deformation and fluid flow in discrete fractures, Tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 1986.

Wilkins, A., Green, C. P., and Ennis-King, J.: PorousFlow: a multiphysics simulation code for coupled problems in porous media, Journal of Open Source Software, 5, 2176, 2020.

Wong, D. L. Y., Doster, F., and Geiger, S.: Embedded Discrete Fracture Models, in: Advanced Modeling with the MATLAB Reservoir Simulation Toolbox, edited by Lie, K.-A. and Møyner, O., chap. 9, pp. 375–408, Cambridge University Press, Cambridge, ISBN 9781316519967, https://doi.org/10.1017/9781009019781.015, 2021.

Wu, J.-Y.: Unified analysis of enriched finite elements for modeling cohesive cracks, Computer methods in applied mechanics and engineering, 200, 3031–3050, 2011.

Wu, Y. S. and Qin, G.: A generalized numerical approach for modeling multiphase flow and transport in fractured porous media, Communications in Computational Physics, 6, 85–108, https://doi.org/10.4208/cicp.2009.v6.p85, 2009.

Yoo, H. and Min, K.-B.: GeomechX: A finite element code for thermo-hydro-mechanical coupled processes in geomechanics, SoftwareX, 32, 102 390, 2025.

Zareidarmiyan, A., Salarirad, H., Vilarrasa, V., Kim, K.-I., Lee, J., and Min, K.-B.: Comparison of numerical codes for coupled thermo-hydro-mechanical simulations of fractured media, Journal of Rock Mechanics and Geotechnical Engineering, 12, 850–865, 2020.

Zyvoloski, G., Dash, Z., and Kelkar, S.: FEHM: Finite element heat and mass transfer code, Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1988.