

A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data

Bahador Bahmani^a, WaiChing Sun^{b,*}

^a *Department of Civil Engineering and Engineering Mechanics, Columbia University, 500 West 120 Street Mudd 610, New York, NY 10027, United States of America*

^b *Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code: 4709, New York, NY 10027, United States of America*

Received 26 November 2020; received in revised form 24 February 2021; accepted 12 April 2021

Available online 5 May 2021

Abstract

We present a hybrid model/model-free data-driven approach to solve poroelasticity problems. Extending the data-driven modeling framework originated from Kirchdoerfer and Ortiz (2016), we introduce one model-free and two hybrid model-based/data-driven formulations capable of simulating the coupled diffusion-deformation of fluid-infiltrating porous media with different amounts of available data. To improve the efficiency of the model-free data search, we introduce a distance-minimized algorithm accelerated by a k-dimensional tree search. To handle the different fidelities of the solid elasticity and fluid hydraulic constitutive responses, we introduce a hybridized model in which either the solid and the fluid solver can switch from a model-based to a model-free approach depending on the availability and the properties of the data. Numerical experiments are designed to verify the implementation and compare the performance of the proposed model to other alternatives.

© 2021 Elsevier B.V. All rights reserved.

Keywords: Model-free poromechanics; Data-driven modeling; Multi-fidelity data; Geomechanics

1. Introduction

The theory of poromechanics attempts to capture how infiltrating pore fluid interacts with the solid skeleton formed by the solid constituents at the scale of the representative elementary volume where an effective medium can be established [1–3]. Poroelasticity is a sub-discipline of poromechanics problem that focuses on the path independent response of porous media. It has important applications across multiple disciplines including seismology [4,5], hydraulic fracture [6,7], petroleum engineering, reservoir management, geological disposal [8,9], and biomechanics modeling for soft tissues and bones [10].

Due to the multiphysics nature of the coupled diffusion-deformation process, poromechanics models must combine field equations which constraints field variables to obey the balance principle with coupled constitutive models which characterize material laws for pore fluid and solid skeleton [11–15]. One important issue that affects the practicality, accuracy, and robustness of the poroelasticity model is the difference in fidelity of the solid and fluid

* Corresponding author.

E-mail address: wsun@columbia.edu (W. Sun).

constitutive laws. In particular, elastic responses of a variety of porous media such as sandstone [16], clay [17–19], sand [20] and bone [10] can be captured quite adequately with the existing state-of-the-art models [21] such that the error of a well-calibrated prediction is often within a few percentages. Hence, a simple elasticity model calibrated with simple compression or shear tests is often sufficient to make forecasting predictions with a narrow confidence interval.

However, owing to the difficulty to conduct highly precise experiments and the lack of parametric space to characterize the hydraulic responses, a typical prediction of permeability based on the porosity–permeability model is expected to have a much higher variance and, in many cases, is considered accurate even if predicted benchmark permeability is just within the same order of magnitude [9,22–28]. In this case, a calibrated hydraulic model that minimizes the mean square error of the Darcy’s velocity or pressure gradient does not yield a reliable forecast due to the much wider confidence intervals. This disparity in the fidelity of the elasticity and hydraulic models for porous media has also been consistently observed in large-scale multi-research-group benchmark studies such as Andrä et al. [25,29] and has become a major bottleneck for poromechanics models.

An alternative to handle this disparity is to introduce a variational model-free approach for the poromechanics problem. First introduced by Kirchdoerfer and Ortiz [30] and later extended for the constitutive manifold [31,32], incorporated with digital image correlation [33] and adopted to diffusion [34], the data-driven approach enables one to make predictions on the most plausible constitutive responses via a distance minimization algorithm. This distance minimization algorithm then chooses either a data point from a material point database or a linear embedding manifold that minimizes the error from the conservation laws and compatibility conditions such that a physics simulation can be carried out without an explicitly derived constitutive law. While this approach holds great promise when a large amount of data is available especially for a low-dimensional prediction (e.g., heat flux in 2D, axial stress of a truss element), the predictions in three-dimensional space for arbitrary loading paths can be difficult if the amount of data is insufficient or not distributed with a sufficient density in the parametric space.

Recent work, such as Leygue et al. [33] and Karapiperis et al. [35] have explored options to address the demand of data. In [33], collections of non-homogeneous strain data, and the applied face of a specimen, which could be obtained from digital image correlation, are used to constitute database. This database can then be used to compute admissible stress–strain pairs from experimental data. Meanwhile, Karapiperis et al. [35] explore the possibility of introducing an on-the-fly sampling technique to generate data from sub-scale computations with no prior information needed as well as employ an offline goal-oriented sampling technique to incorporate high-fidelity simulations and high-resolution experiments in the material database.

In this work, we introduce a more flexible hybrid approach where the solid constitutive law can be either model-based (when the fidelity of the elasticity model is sufficiently high) or model-free (when there are sufficient data points). Meanwhile, the hydraulic model is replaced by the model-free approach to avoid the usage of models with high deviations to make predictions. To cut the CPU time required to conduct the search for the closest data point, we introduce a k -dimensional tree search that helps organizing data points to accelerate the simulations. Numerical experiments are then conducted to examine and compare the fully model-based, the fully data-driven, and the hybrid models.

The remaining parts of the paper will proceed as follows. We first introduce the formulations that enable data-driven algorithm to replace parts or all of the constitutive laws required to generate incremental solution updates for the poroelasticity problems (Section 2). We then introduce a search strategy that enables us to accelerate the time used to search for the optimal data points which are often the bottleneck of the speed for the data-driven models (Section 3). Section 4 overviews the data-driven algorithm and its implementation aspects. Numerical experiments are then conducted to verify the formulation and test the accuracy, robustness, and fidelity of the hybridized and data-driven models (Section 5) followed by a brief conclusion that summarized the major findings.

As for notations and symbols, bold-faced letters denote tensors (including vectors which are rank-one tensors); the symbol ‘ \cdot ’ denotes a single contraction of adjacent indices of two tensors (e.g. $\mathbf{a} \cdot \mathbf{b} = a_i b_i$ or $(\mathbf{c} \cdot \mathbf{d})_{ik} = c_{ij} d_{jk}$); the symbol ‘ $:$ ’ denotes a double contraction of adjacent indices of tensor of rank two or higher (e.g. $(\mathbf{C} : \boldsymbol{\varepsilon}^e)_{ij} = C_{ijkl} \varepsilon_{kl}^e$); the symbol ‘ \otimes ’ denotes a juxtaposition of two vectors (e.g. $(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j$) or two symmetric second-order tensors (e.g. $(\boldsymbol{\alpha} \otimes \boldsymbol{\beta})_{ijkl} = \alpha_{ij} \beta_{kl}$).

2. Hybridized data-driven/model-based poromechanics problem

This section presents formulations that solve poroelasticity problems either in a fully model-free fashion or in a hybridized model where either the solid or fluid constitutive law is replaced by a model-free data-driven

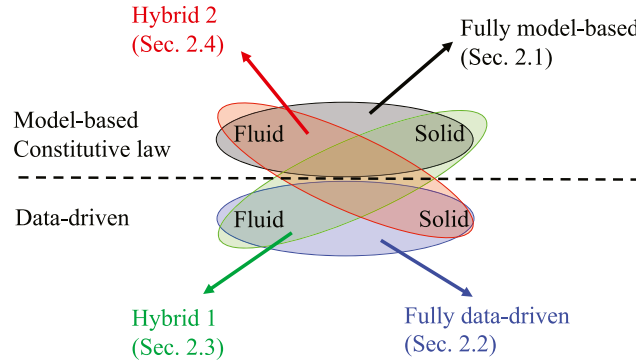


Fig. 1. Subsections arrangement based on the constitutive assumptions for solid and fluid phases. The first row indicates the fully model-based assumption. The second row indicates the fully data-driven assumption.

approach. For completeness, we first review the classical poromechanics problem in which the solid displacement and pore pressure are the primary unknown variables. We then introduce a data-driven model-free algorithm to completely replace classical solid and fluid constitutive laws with the corresponding data-driven algorithm searched for the optimal data points. Finally, two hybridized formulations are provided in each of which either solid or fluid constitutive law follows the classical model-based approach. The pros and cons of each formulation for different situations (e.g., availability of data, fidelity consideration, robustness) are discussed and will be further elaborated in the numerical experiments shown in Section 5. Fig. 1 overviews the main topic of each subsection of the current section.

2.1. A brief review of field equations for poroelasticity problems

For completeness, we provide a concise review of the poroelasticity model, which consists of two major components, i.e., the field theory that provides the necessary constraints for the field variables in the space–time domain and the material laws that provides the local constitutive updates for both the solid skeleton and the fluid constituents. Interested readers may refer to Prevost [36], Borja and Alarcón [37], Zienkiewicz et al. [11], Coussy [3], Sun et al. [38,39], Na and Sun [9] and Na et al. [19] for a more comprehensive treatment for the topic.

For simplicity, we assume that the deformation of the solid skeleton is, infinitesimal, path independent and elastic such that $\epsilon = \epsilon^e$ and $\sigma = \sigma(\epsilon^e, p) = \sigma(\epsilon, p)$. We also assume that the flow in the pore space is a function of the pore pressure gradient such that the Darcy's velocity $\mathbf{q} = \mathbf{q}(\nabla^x p)$ is a function of the pore pressure p . In this formulation, we follow the trajectory of the solid constituent and express the balance principles as a function of solid displacement $\mathbf{u} = \mathbf{u}^s$ and pore pressure p . By neglecting the inertial term, the balance of linear momentum and mass on the spatial domain Ω over time $t \in T$ read (cf. [39]),

$$\nabla^x \cdot \sigma(\epsilon, p) + \gamma = \mathbf{0} \text{ in } \Omega \times T, \quad (1)$$

$$\frac{\dot{p}}{M} + B\dot{\epsilon}_{\text{vol}} + \nabla^x \cdot \mathbf{q}(\nabla^x p) + s = 0 \text{ in } \Omega \times T, \quad (2)$$

where σ is the total stress and $\epsilon_{\text{vol}} = \text{tr } \epsilon$ is the volumetric strain. We postulate that the total stress can be partitioned into the effective stress of the solid skeleton and the pore fluid pressure according to the effective stress principle, i.e.,

$$\sigma = \sigma'(\epsilon) - Bp\mathbf{I} \text{ in } \Omega \times T. \quad (3)$$

Furthermore, γ is the body force, M is the Biot modulus, B is the Biot coefficient, \mathbf{q} is the Darcy's velocity, s is the source (sink) term. The definitions of these physical quantities are listed below.

$$B = 1 - \frac{K}{K_s}, \quad (4)$$

$$\gamma = \gamma^s + \gamma^f = (1 - \phi^f)\rho_s \mathbf{g} + \phi^f \rho_f \mathbf{g}, \quad (5)$$

$$M = \frac{K_s K_f}{K_f(B - \phi^f) + K_s \phi^f}, \quad (6)$$

$$\mathbf{q} = \phi^f(\mathbf{v}^f - \mathbf{v}), \quad (7)$$

where K and K_s are the bulk moduli of the solid skeleton and the solid constituent, respectively, γ^s and γ^f are the partial density of the solid and fluid constituents, while ρ_s and ρ_f are the intrinsic density of the solid and fluid constituents. ϕ^f is the porosity, K_f is the bulk modulus of the fluid constituent, and \mathbf{v}^f and \mathbf{v} are the velocity of the fluid constituent and the solid skeleton, respectively.

To compute the boundary value problem, the initial and boundary conditions are specified as follows. The initial conditions are $\mathbf{u} = \mathbf{u}_0$ in Ω at $t = 0$ and $p = p_0$ in Ω at $t = 0$. Meanwhile, the boundary conditions are $\mathbf{u} = \bar{\mathbf{u}}$ on $\partial\Omega_u \times T$ (prescribed displacement) and $\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}}$ on $\partial\Omega_\sigma \times T$ (prescribed traction), $p = \bar{p}$ on $\partial\Omega_p \times T$ (prescribed pore pressure), and $\mathbf{q} \cdot \mathbf{n} = \bar{q}$ on $\partial\Omega_q \times T$ (prescribed fluid flux) where \mathbf{n} is a unit normal vector pointing outward to the boundary $\partial\Omega$. Furthermore, the following conditions must hold: $\partial\Omega_\sigma \cup \partial\Omega_u = \partial\Omega$, $\partial\Omega_\sigma \cap \partial\Omega_u = \emptyset$, $\partial\Omega_q \cup \partial\Omega_p = \partial\Omega$ and $\partial\Omega_q \cap \partial\Omega_p = \emptyset$.

For convenience, we first discretize the governing equations, Eqs. (1) and (2), in time via the implicit Euler scheme. Given the displacement and pore pressure at time t_n , the time-discretized balance principle within the time interval $T \in [t_n, t_{n+1}]$ can be expressed as,

$$\nabla^x \cdot \boldsymbol{\sigma}_{n+1}(\boldsymbol{\epsilon}_{n+1}, p_{n+1}) + \boldsymbol{\gamma}_{n+1} = \mathbf{0} \text{ in } \Omega, \quad (8)$$

$$\frac{p_{n+1}}{M} + B\epsilon_{\text{vol}n+1} + \nabla^x \cdot \mathbf{q}_{n+1}(\nabla^x p_{n+1})\Delta t + s_{n+1}\Delta t = \frac{p_n}{M} + B\epsilon_{\text{vol}n} \text{ in } \Omega. \quad (9)$$

To complete the initial boundary value problem, the classical model-based approach for poroelasticity problem required us to define material constitutive laws such that, given the strain and pore pressure gradient, we may obtain updated effective stress and Darcy's velocity; the first one maps a given strain to the effective stress $(\boldsymbol{\epsilon}, \boldsymbol{\sigma}')$, another one maps a given pore pressure gradient to the Darcy's velocity $(\nabla^x p, \mathbf{q})$. In what follows, our goal is to introduce a new formulation such that *at least one* of these two constitutive laws, $(\boldsymbol{\epsilon}, \boldsymbol{\sigma}')$ and $(\nabla^x p, \mathbf{q})$ can be replaced by the model-free data-driven approach originally designed for elasticity problem in [30].

Remark 1. Notice that, for porous media with compressible constituents, both the Biot coefficient and the Biot modulus may depend on the bulk modulus of the skeleton and hence may evolve if the elastic response of the solid skeleton is nonlinear. Here we limit our scope to the types of porous media with the bulk modulus of both constituents significantly larger than the effective bulk modulus of the solid skeleton. As such, the evolution of both the Biot coefficient and the Biot modulus, B and M , are neglected [2,40,41]. This assumption and the effective stress principle together enable us to treat the constitutive laws for the solid deformation and fluid flow as two independent systems.

2.2. Option 1: Pure data-driven poroelasticity

Here, our goal is to present a new formulation constrained by the time-discretized balance principle listed in Eqs. (8) and (9) without employing any constitutive laws. To do so, we assume that there exist two databases, each have a finite number of data points for $(\boldsymbol{\epsilon}, \boldsymbol{\sigma}')$ and $(\nabla^x p, \mathbf{q})$. We further assume that there are sufficient data points distributed in the parametric space for both material laws such that a complete model-free approach is feasible. Notice that the data-driven model-free method developed in this research is categorized into non-parametric learning methods [42] which are basically considered data-demanding since they have minimum assumptions about the data-driven model. This is in contrast to the parametric methods such as neural networks [43–45] where there is a stronger pre-assumption about the data-driven model.

In the following subsections, we first express the mathematical statement for the data-driven scheme as a double-minimization problem. Then, we provide a numerical strategy based on the fixed-point (staggered) method to solve this minimization problem in two steps: global and local minimization steps. Finally, we provide a numerical solution for the global minimization based on the Lagrange multiplier method. The local minimization step is not covered in this section and will be discussed in Section 3 separately since it has its specific treatment which is common between all the three formulations.

2.2.1. Problem statement

In this section, we present the fully data-driven, constitutive-model-free formulation for the poroelasticity problem. Our goal is to introduce a minimization problem that constitutes a model-free poroelasticity solver. As such, we seek solutions from data sets of material responses that weakly satisfy the time-discretized balance principle listed in Eqs. (8) and (9) without the necessity introducing constitutive laws for the solid skeleton and the pore fluid constituents.

If there are an infinite number of error-free data points populating the database, then a constitutive manifold can be identified such that one may select elements of the manifold that satisfy the constitutive laws. However, material databases are rarely populated with enough data and often contains data with noise that makes it impractical to impose such a strict requirement. As an alternative, we follow the idea of Kirchdoerfer and Ortiz [30] where we merely seek solutions that satisfy the balance principles while the resultant constitutive responses are the ones closest to but not necessary elements of the set of points in the databases. As such, we need to introduce the notion of “distance” via an appropriate norm we selected for the space of admissible solution (ϵ, σ') and $(\nabla^x p, q)$.

In other words, we regard the balance principles as the universal law that should not be violated. The data-driven method is then designed to generate solutions that fulfill the balance principle while the material response at the integration points (ϵ, σ') and $(\nabla^x p, q)$ are all closest to existing data points in the material database. The distance between an admissible response satisfied balance laws and an existing data point is then measured by an appropriate norm. We will express the above-mentioned statement as a double-minimization problem [30,32,34,46]. For brevity, we assign a new variable for pressure gradient as $\mathbf{r} = \nabla^x p$.

We then define the phase space for poroelasticity at time t_{n+1} as all $\mathbf{z}_{n+1}^{\text{sf}} = (\epsilon_{n+1}, \sigma'_{n+1}, \mathbf{r}_{n+1}, \mathbf{q}_{n+1}) \in \mathcal{Z}^{\text{sf}}$ where $\mathcal{Z}^{\text{sf}} = \mathcal{Z}^s \times \mathcal{Z}^f$ is the product space of the solid phase space $\mathcal{Z}^s = V_\epsilon \times V_{\sigma'}$ and the fluid phase space $\mathcal{Z}^f = V_r \times V_q$; in which $V_\epsilon, V_{\sigma'} = [L^2(\Omega)]^{\sum_{A=1}^{\text{ndim}} A}$ are spaces of real-valued symmetric 2nd order tensor fields with square integrable components, and $V_r, V_q = [L^2(\Omega)]^{\text{ndim}}$ are spaces of real-valued vector fields with square integrable components. We define the space $\mathcal{C}_{n+1}^{\text{momentum}}$ as follows:

$$\mathcal{C}_{n+1}^{\text{momentum}} = \{(\epsilon_{n+1}(\mathbf{u}_{n+1}), \sigma'_{n+1}, p_{n+1}) \in \mathcal{Z}^s \times V_p \mid \nabla^x \cdot \sigma_{n+1} + \gamma_{n+1} = \mathbf{0} \text{ in } \Omega, \mathbf{u}_{n+1} = \bar{\mathbf{u}}_{n+1} \text{ on } \partial\Omega_u, \\ \sigma_{n+1} \cdot \mathbf{n} = \bar{\mathbf{t}}_{n+1} \text{ on } \partial\Omega_\sigma, \sigma_{n+1} = \sigma'_{n+1} - Bp_{n+1}\mathbf{I} \text{ in } \Omega\},$$

where $\mathbf{u}_{n+1} \in V_u$, $V_u = [H^1(\Omega)]^{\text{ndim}}$, $V_p = H^1(\Omega)$, and H^1 denotes the Sobolev space of square-integrable functions with square-integrable first derivative and ndim denotes the dimension of the spatial domain. Here, strain tensor $\epsilon_{n+1}(\mathbf{u}_{n+1})$ is a derived quantity of displacement vector \mathbf{u}_{n+1} through small strain relation. All members in continuous set $\mathcal{C}_{n+1}^{\text{momentum}}$ satisfy time discretized conservation of linear momentum equation, mechanical-related boundary conditions, small deformation relation for strain tensor, and effective stress principle at time t_{n+1} . The compatibility conditions of the strain field are automatically satisfied since the small strain tensor is the symmetric part of the displacement field gradient. The space of admissible solutions that satisfy the conservation of mass is denoted by $\mathcal{C}_{n+1}^{\text{mass}}$ and defined as follows:

$$\mathcal{C}_{n+1}^{\text{mass}} = \left\{(\mathbf{r}_{n+1}(p_{n+1}), \mathbf{q}_{n+1}, \epsilon_{n+1}(\mathbf{u}_{n+1})) \in \mathcal{Z}^f \times V_\epsilon \mid \frac{p_{n+1}}{M} + B\epsilon_{\text{vol}n+1} + \nabla^x \cdot \mathbf{q}_{n+1} \Delta t + s_{n+1} \Delta t = \frac{p_n}{M} + B\epsilon_{\text{vol}n} \text{ in } \Omega, p_{n+1} = \bar{p}_{n+1} \text{ on } \partial\Omega_p, \mathbf{q}_{n+1} \cdot \mathbf{n} = \bar{q}_{n+1} \text{ on } \partial\Omega_q\right\},$$

where the gradient of pore pressure $\mathbf{r}_{n+1}(p_{n+1})$ is a derived quantity of pressure field herein. The admissible solution space $\mathcal{C}_{n+1}^{\text{coupled}}$ for the poroelasticity problem at time t_{n+1} includes field variables $\epsilon_{n+1}(\mathbf{u}_{n+1})$, σ'_{n+1} , $\mathbf{r}_{n+1}(p_{n+1})$, and \mathbf{q}_{n+1} that satisfy all physical constraints at time t_{n+1} . This space exists at the intersection of above defined continuous sets $\mathcal{C}_{n+1}^{\text{coupled}} = \mathcal{C}_{n+1}^{\text{momentum}} \cap \mathcal{C}_{n+1}^{\text{mass}}$. The poroelasticity database at time t_{n+1} is denoted by $\mathcal{D}_{n+1}^{\text{sf}} \subset \mathcal{Z}^{\text{sf}}$. This discrete set contains a finite number of elements that stores experimental data points corresponding to the poroelasticity constitutive laws at time t_{n+1} . The poroelasticity database format and its properties will be clarified in detail later.

The data-driven solution $\bar{\mathbf{z}}_{n+1}^{\text{sf}} \in \mathcal{Z}^{\text{sf}}$ at time t_{n+1} is the solution of the following double-minimization problem:

$$\bar{\mathbf{z}}_{n+1}^{\text{sf}} = \arg \min_{\mathcal{C}_{n+1}^{\text{coupled}}} \left\{ \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{D}_{n+1}^{\text{sf}}} \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{C}_{n+1}^{\text{coupled}}} \left\| \mathbf{z}_{n+1}^{\text{sf}*} - \mathbf{z}_{n+1}^{\text{sf}} \right\|_{\mathcal{Z}^{\text{sf}}}^2 \right\}, \quad (10)$$

where $\mathbf{z}_{n+1}^{\text{sf}*} = (\epsilon_{n+1}^*, \sigma'_{n+1}^*, \mathbf{r}_{n+1}^*, \mathbf{q}_{n+1}^*)$ is an element of poroelasticity database,

$\mathbf{z}_{n+1}^{\text{sf}} = (\boldsymbol{\epsilon}_{n+1}(\mathbf{u}_{n+1}), \boldsymbol{\sigma}'_{n+1}, \mathbf{r}_{n+1}(p_{n+1}), \mathbf{q}_{n+1})$ is an element of admissible poroelasticity solution space, and $\|\cdot\|_{\mathcal{Z}^{\text{sf}}}$ is a norm associated with the space \mathcal{Z}^{sf} which measures closeness of $\mathbf{z}_{n+1}^{\text{sf}*}$ and $\mathbf{z}_{n+1}^{\text{sf}}$. We define the subtract operation on space \mathcal{Z}^{sf} as follows:

$$\mathbf{z}_{n+1}^{\text{sf}*} - \mathbf{z}_{n+1}^{\text{sf}} = (\boldsymbol{\epsilon}_{n+1}^* - \boldsymbol{\epsilon}_{n+1}, \boldsymbol{\sigma}'_{n+1}^* - \boldsymbol{\sigma}'_{n+1}, \mathbf{r}_{n+1}^* - \mathbf{r}_{n+1}, \mathbf{q}_{n+1}^* - \mathbf{q}_{n+1}) \in \mathcal{Z}^{\text{sf}}. \quad (11)$$

We define the product norm for the poroelasticity product phase space \mathcal{Z}^{sf} as follows:

$$\|\mathbf{z}_{n+1}^{\text{sf}}\|_{\mathcal{Z}^{\text{sf}}} = \sqrt{\|\mathbf{z}_{n+1}^{\text{s}}\|_{\mathcal{Z}^{\text{s}}}^2 + \Delta t \|\mathbf{z}_{n+1}^{\text{f}}\|_{\mathcal{Z}^{\text{f}}}^2} \quad (12)$$

where $\mathbf{z}_{n+1}^{\text{s}} = (\boldsymbol{\epsilon}_{n+1}, \boldsymbol{\sigma}'_{n+1}) \in \mathcal{Z}^{\text{s}}$, $\mathbf{z}_{n+1}^{\text{f}} = (\mathbf{r}_{n+1}, \mathbf{q}_{n+1}) \in \mathcal{Z}^{\text{f}}$ and $\|\cdot\|_{\mathcal{Z}^{\text{s}}}$ and $\|\cdot\|_{\mathcal{Z}^{\text{f}}}$ are the norms for the respectively solid and fluid phase spaces. The time step size Δt in the second term of Eq. (12) is introduced to make the unit consistent. We define the following norm for \mathcal{Z}^{s} :

$$\|\mathbf{z}_{n+1}^{\text{s}}\|_{\mathcal{Z}^{\text{s}}}^2 = \|(\boldsymbol{\epsilon}_{n+1}, \boldsymbol{\sigma}'_{n+1})\|_{\mathcal{Z}^{\text{s}}}^2 = \int_{\Omega} \frac{1}{2} \boldsymbol{\epsilon}_{n+1} : \mathbb{C}_{\text{s}} : \boldsymbol{\epsilon}_{n+1} + \frac{1}{2} \boldsymbol{\sigma}'_{n+1} : \mathbb{S}_{\text{s}} : \boldsymbol{\sigma}'_{n+1} d\Omega, \quad (13)$$

where \mathbb{C}_{s} and \mathbb{S}_{s} are 4th order symmetric positive definite tensors. As shown in [30,32,34], specific \mathbb{C}_{s} and \mathbb{S}_{s} can be chosen from different equivalent norms, provided that both tensors remain positive definite. The choices of \mathbb{C}_{s} and \mathbb{S}_{s} may affect the values of the norm due to the weighting but the resultant normed space is topologically identical to a Euclidean space.

The weighting tensor \mathbb{C}_{s} share the same unit as the elasticity tensor, i.e., Force/Length², while the unit of \mathbb{S}_{s} is the reciprocal of that of \mathbb{C}_{s} . Both the eigenvalues and the spectral directions of these tensors affect the values of norms and therefore change how distance is measured and could affect the efficiency of the search problems (cf. [47,48]). It is suggested in [32–34,46] to select $\mathbb{S}_{\text{s}} = \mathbb{C}_{\text{s}}^{-1}$ for solid mechanics applications. On the other hand, the norm for \mathcal{Z}^{f} is defined as,

$$\|\mathbf{z}_{n+1}^{\text{f}}\|_{\mathcal{Z}^{\text{f}}}^2 = \|(\mathbf{r}_{n+1}, \mathbf{q}_{n+1})\|_{\mathcal{Z}^{\text{f}}}^2 = \int_{\Omega} \frac{1}{2} \mathbf{r}_{n+1} \cdot \mathbf{C}_{\text{f}} \cdot \mathbf{r}_{n+1} + \frac{1}{2} \mathbf{q}_{n+1} \cdot \mathbf{S}_{\text{f}} \cdot \mathbf{q}_{n+1} d\Omega, \quad (14)$$

where \mathbf{C}_{f} and \mathbf{S}_{f} are 2nd order symmetric positive definite tensors. Similarly, these numerical parameters control the importance of pressure gradient and Darcy's velocity vectors in the norm calculations. The unit for \mathbf{C}_{f} is the same as hydraulic conductivity unit ($\frac{\text{Length}^4}{\text{Force} \times \text{Time}}$). This norm, which in our case is of the unit of power, has been introduced in [34] for Poisson's equation.

Similar to the norm equipped by the solid phase space, the implication of the choice of the specific weighting effect for \mathbf{S}_{f} and \mathbf{C}_{f} has not been previously studied. Nguyen et al. [34] suggests $\mathbf{S}_{\text{f}} = \mathbf{C}_{\text{f}}^{-1}$. The time increment Δt used in Eq. (12) is the scaling factor to make both terms with the same unit as energy. This factor could be directly included in \mathbf{C}_{f} and \mathbf{S}_{f} , but we preferred to be consistent with power-like definition of fluid phase space metric.

To find the stationary points for the double minimization defined in Eq. (10) at time $t = t_{n+1}$, a global–local iteration (which is our method of preference and will be described later) is needed to find both the admissible solution $\mathbf{z}_{n+1}^{\text{sf}}$ and the discrete data points that minimize the distance defined by the norm in Eq. (12). The latter can be done by comparing every point from the material databases to identify the optimized data point from $\mathcal{D}_{n+1}^{\text{sf}}$ for each integration point. However, searching the optimized data points from the entire data set can be inefficient for a large database. As such, we consider an adaptive poroelasticity database where a subset of the plausible data points is collected for each time step to constitute a temporal-varying material database constituted by the union of an adaptive solid effective-stress–strain database \mathcal{D}_i^{s} and an adaptive fluid pressure–gradient–Darcy-velocity database \mathcal{D}_i^{f}

$$\mathcal{D}_i^{\text{sf}} = \{(\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}'^*, \mathbf{r}^*, \mathbf{q}^*) | (\boldsymbol{\epsilon}^*, \boldsymbol{\sigma}'^*) \in \mathcal{D}_i^{\text{s}}, (\mathbf{r}^*, \mathbf{q}^*) \in \mathcal{D}_i^{\text{f}}\}, \quad (15)$$

where the subscript i indicates the snapshot taken at a discrete time step t_i . For instance, an admissible subset of data can be identified via prior knowledge (e.g., upper and lower bounds of the porosity–permeability relationship, correlation structures, etc.) and deductive reasoning. In our last numerical experiment, we use porosity (the ratio between the void and solid phase volume) to filter out the implausible data points. This treatment enables the data-driven solver to narrow down the search of possible solutions and therefore enhance the efficiency and reduce the memory requirement. Furthermore, the trade-off between computational time and memory will be discussed later in Section 3. Note that the adaptive database design can also be used as a mean to incorporate an active

learning algorithm that generates new data points on demand [49,50]. The active learning approach as well as other algorithms that may identify the feasible subset of data through clustering [51,52] or other techniques are not discussed in this work but will further be explored in the future.

Here, the solid data set \mathcal{D}_{n+1}^s is the set of experimental strain and effective stress pairs (ϵ^*, σ'^*) ; one point in this data set is a bundle of strain and effective stress together corresponding to one experimental observation. Fluid data set \mathcal{D}_{n+1}^f is the set of all experimental pressure gradient and Darcy's velocity pairs $(\mathbf{r}^*, \mathbf{q}^*)$; one point in this data set is a bundle of pressure gradient and Darcy's velocity together corresponding to one experimental observation. The union of separate databases for stress-strain and hydraulic responses is designed for practical reasons because experiments that obtain the stress-strain curves and the effective permeability of a specimen are often conducted separately [22,26,53].

2.2.2. Solution strategy: fixed-point iteration

Here, we use the fixed-point method to numerically solve the double-minimization statement Eq. (10) associated with the fully data-driven poroelasticity problem. The use case of this method is initially shown by Kirchdoerfer and Ortiz [30] for data-driven elasticity problems.

The data-driven solution must minimize an objective, i.e., the norm defined over poroelasticity phase space Eq. (12), with two different sets of constraints. One set of constraints belongs to a discrete set, $\mathcal{D}_{n+1}^{\text{sf}}$, with a finite number of members, but the other one belongs to continuous space, i.e., $\mathcal{C}_{n+1}^{\text{coupled}}$, that satisfies conservation laws. This minimization statement is categorized into combinatorial optimization problems due to the discrete nature of $\mathcal{D}_{n+1}^{\text{sf}}$, making the problem NP-hard. The fixed point method, or staggered method, reduces complexities by proposing a sequential solution algorithm, and it has been used in many applications [54–56]. Here, it is used to break down the double-minimization into two separate, simpler minimization problems. We solve for one minimization problem by assuming the solution for the other one is fixed. In this way, we iteratively solve a minimization problem and update for another one until convergence of the solution.

According to the fixed-point method, we assume that the optimal data points $\mathbf{z}_{n+1}^{\text{sf}*}$ are known in the objective function Eq. (10), therefore we just need to minimize the objective function for unknowns $\mathbf{z}_{n+1}^{\text{sf}}$ as follows:

$$\mathbf{z}_{n+1}^{\text{sf}} = \arg \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{C}_{n+1}^{\text{coupled}}} \left\| \mathbf{z}_{n+1}^{\text{sf}*} - \mathbf{z}_{n+1}^{\text{sf}} \right\|_{\mathcal{Z}^{\text{sf}}}^2, \text{ given } \mathbf{z}_{n+1}^{\text{sf}*} \in \mathcal{D}_{n+1}^{\text{sf}}. \quad (16)$$

We call this minimization step global minimization. The global solution field $\mathbf{z}_{n+1}^{\text{sf}}$ is a product field consists of global fields as $\mathbf{z}_{n+1}^{\text{sf}} = (\epsilon_{n+1}(\mathbf{u}_{n+1}), \sigma'_{n+1}, \mathbf{r}_{n+1}(p_{n+1}), \mathbf{q}_{n+1})$. The global minimization step, geometrically, project discrete points $\mathbf{z}_{n+1}^{\text{sf}*}$ onto the continuous space $\mathcal{C}_{n+1}^{\text{coupled}}$ according to the defined norm in Eq. (12); see red dash lines in Fig. 2. In other words, this step finds solutions belong to the physical space $\mathcal{C}_{n+1}^{\text{coupled}}$ that are closest points to the selected data points (from material space).

In the next step of fixed-point method, we assume the solutions $\mathbf{z}_{n+1}^{\text{sf}}$ are known, then we find optimal data points $\mathbf{z}_{n+1}^{\text{sf}*}$ that minimize the objective function Eq. (10) as follows:

$$\mathbf{z}_{n+1}^{\text{sf}*} = \arg \min_{\mathbf{z}_{n+1}^{\text{sf}*} \in \mathcal{D}_{n+1}^{\text{sf}}} \left\| \mathbf{z}_{n+1}^{\text{sf}*} - \mathbf{z}_{n+1}^{\text{sf}} \right\|_{\mathcal{Z}^{\text{sf}}}^2, \text{ given } \mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{C}_{n+1}^{\text{coupled}}. \quad (17)$$

This minimization is defined over the discrete space $\mathcal{D}_{n+1}^{\text{sf}}$. Since choices for $\mathbf{z}_{n+1}^{\text{sf}*}$ are finite and there is no constraint on data points in the database, the global objective function Eq. (17) defined over Ω is minimized if the integrand is locally minimized. In this work, Gaussian quadrature is used to approximate the spatial integration. Let all the integration points be elements of a finite set $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_{n_{\text{int}}}\}$, then the local minimization problem for an integration point $\bar{\mathbf{x}}_a$ reads,

$$\mathbf{z}_{n+1}^{\text{sf}*}(\bar{\mathbf{x}}_a) = \arg \min_{\mathbf{z}_{n+1}^{\text{sf}*} \in \mathcal{D}_{n+1}^{\text{sf}}} [d^{\text{sf}}(\mathbf{z}_{n+1}^{\text{sf}*}, \mathbf{z}_{n+1}^{\text{sf}}(\bar{\mathbf{x}}_a))]^2, \text{ given } \mathbf{z}_{n+1}^{\text{sf}}(\bar{\mathbf{x}}_a) \in \mathcal{C}_{n+1}^{\text{coupled}}, \quad (18)$$

where $d^{\text{sf}}(\cdot)$ is a local distance function between $\mathbf{z}_{n+1}^{\text{sf}*}$ and $\mathbf{z}_{n+1}^{\text{sf}}$, which is defined as, Eqs. (11), (12), (13), and (14):

$$d^{\text{sf}}(\mathbf{z}_{n+1}^{\text{sf}*}, \mathbf{z}_{n+1}^{\text{sf}}) = [d^s(\mathbf{z}_{n+1}^{\text{s}*}, \mathbf{z}_{n+1}^{\text{s}})^2 + \Delta t d^f(\mathbf{z}_{n+1}^{\text{f}*}, \mathbf{z}_{n+1}^{\text{f}})^2]^{1/2}, \quad (19)$$

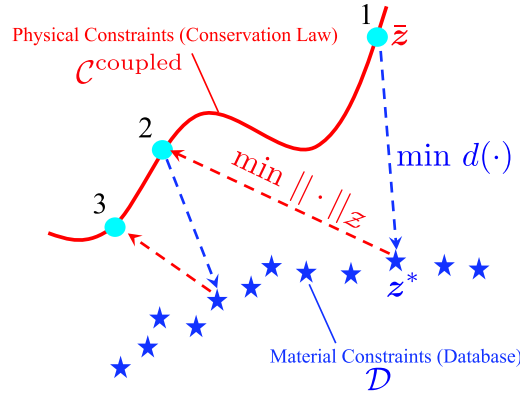


Fig. 2. Schematic representation of material z^* and physical \bar{z} responses at a quadrature point during fixed-point iterations. Numbers show iteration numbers. Star points present the entire database. The solid line describes the continuous admissible solution space that respects conservation laws. Dash blue arrow lines depict the local minimization step from the physical manifold to the material manifold for the quadrature point. Dash red arrow lines depict the global minimization step from the material manifold to the physical manifold. Note that the global step minimizes the norm over the entire domain (for all quadrature points). However, the local step minimizes the distance locally at each quadrature point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$d^s(z_{n+1}^{s*}, z_{n+1}^s)^2 = \frac{1}{2}(\epsilon_{n+1}^* - \epsilon_{n+1}) : \mathbb{C}_s : (\epsilon_{n+1}^* - \epsilon_{n+1}) + \frac{1}{2}(\sigma_{n+1}' - \sigma_{n+1}') : \mathbb{S}_s : (\sigma_{n+1}' - \sigma_{n+1}'), \quad (20)$$

$$d^f(z_{n+1}^{f*}, z_{n+1}^f)^2 = \frac{1}{2}(\mathbf{r}_{n+1}^* - \mathbf{r}_{n+1}) \cdot \mathbf{C}_f \cdot (\mathbf{r}_{n+1}^* - \mathbf{r}_{n+1}) + \frac{1}{2}(\mathbf{q}_{n+1}^* - \mathbf{q}_{n+1}) \cdot \mathbf{S}_f \cdot (\mathbf{q}_{n+1}^* - \mathbf{q}_{n+1}), \quad (21)$$

where $d^s(\cdot)$ and $d^f(\cdot)$ denote the local distances between solid-related and fluid-related components, respectively. The technique that solves the local minimization will be discussed in Section 3. Geometrically, the local minimization step project a point in $\mathcal{C}_{n+1}^{\text{coupled}}$ onto the data set $\mathcal{D}_{n+1}^{\text{sf}}$, see blue dash lines in Fig. 2. By minimizing the distance defined in Eq. (19), we determine points in the data set (material space) that are closest to the conservation laws.

Each fixed-point iteration consists of two steps [32]. First, we solve the global minimization Eq. (17) to project the solution coming from material space (database) $\mathcal{D}_{n+1}^{\text{sf}}$ onto physical space $\mathcal{C}_{n+1}^{\text{coupled}}$; see red dash arrow lines in Fig. 2. Second, we solve local minimization problems to project the most recent solutions belong to the physical space onto the material space; see blue dash arrow lines in Fig. 2. Fixed-point iterations continue until there is no change more than a user-defined tolerance in optimal solutions at time step t_{n+1} . In summary, we solve the double-minimization problem in two consecutive steps:

Global Step: find the physical field $\mathbf{z}_{n+1}^{\text{sf}} = (\epsilon, \sigma', \mathbf{r}, \mathbf{q})_{n+1}$ for a given material field $\mathbf{z}_{n+1}^{\text{sf}*} \in \mathcal{D}_{n+1}^{\text{sf}}$ via:

$$\min_{\mathbf{z}_{n+1}^{\text{sf}}} \left\| \mathbf{z}_{n+1}^{\text{sf}*} - \mathbf{z}_{n+1}^{\text{sf}} \right\|_{\mathcal{Z}^{\text{sf}}}^2, \text{ such that } \mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{C}_{n+1}^{\text{coupled}}.$$

Local Step: at each local spatial point $\bar{\mathbf{x}}_i$ (quadrature), find the local material point $\mathbf{z}_{i, n+1}^{\text{sf}*} = (\epsilon^*, \sigma'^*, \mathbf{r}^*, \mathbf{q}^*)_{i, n+1}$ for a given physical point $\mathbf{z}_{i, n+1}^{\text{sf}} \in \mathcal{C}_{n+1}^{\text{coupled}}$ via:

$$\arg \min_{\mathbf{z}_{i, n+1}^{\text{sf}*}} [d^{\text{sf}}(\mathbf{z}_{i, n+1}^{\text{sf}*}, \mathbf{z}_{i, n+1}^{\text{sf}})]^2, \text{ such that } \mathbf{z}_{i, n+1}^{\text{sf}*} \in \mathcal{D}_{n+1}^{\text{sf}}.$$

We will explain how the global and local steps can be formulated for numerical solutions in Sections 2.2.3 and 3, respectively.

2.2.3. Global minimization

We introduce the functional associated with the global constrained optimization for the fully data-driven poroelasticity problem. The objective function in Eq. (16) is minimized along with the set of constraints defined in $\mathcal{C}_{n+1}^{\text{coupled}}$. As such, the trial spaces V_u and V_p for the u/p poromechanics formulation are chosen to strongly satisfy

Dirichlet boundary conditions,

$$V_u = \left\{ \mathbf{u} : \Omega \rightarrow \mathbb{R}^3 \mid \mathbf{u} \in [H^1(\Omega)]^3, \mathbf{u} = \bar{\mathbf{u}} \text{ on } \partial\Omega_u \right\}, \quad (22)$$

$$V_p = \left\{ p : \Omega \rightarrow \mathbb{R} \mid p \in H^1(\Omega), p = \bar{p} \text{ on } \partial\Omega_p \right\}. \quad (23)$$

The optimal solutions $\mathbf{u}_{n+1} \in V_u$, $\boldsymbol{\sigma}'_{n+1} \in V_{\sigma'}$, $p_{n+1} \in V_p$, and $\mathbf{q}_{n+1} \in V_q$ are the stationary points of the following functional:

$$\mathcal{L}_{\text{tot}}^{\text{DD}}(\mathbf{z}_{n+1}^{\text{sf}}, \mathcal{B}_{n+1}; \mathbf{z}_{n+1}^{\text{sf}*}) = \mathcal{L}_{\text{loss}}^{\text{DD}} + \mathcal{L}_{\text{momentum}}^{\text{DD}} + \mathcal{L}_{\text{mass}}^{\text{DD}}, \quad (24)$$

where $\mathcal{L}_{\text{loss}}^{\text{DD}}$ is the original objective (loss) function, and $\mathcal{L}_{\text{momentum}}^{\text{DD}}$ and $\mathcal{L}_{\text{mass}}^{\text{DD}}$ are contributions from constraints defined in $\mathcal{C}_{n+1}^{\text{momentum}}$ and $\mathcal{C}_{n+1}^{\text{mass}}$, respectively. We group all the Lagrange multipliers in $\mathcal{B}_{n+1} = \{\boldsymbol{\beta}_{n+1}^{\text{mon}}, \boldsymbol{\beta}_{n+1}^{\sigma}, \beta_{n+1}^{\text{mass}}, \beta_{n+1}^q\}$ where $\boldsymbol{\beta}_{n+1}^{\text{mon}}$ and $\boldsymbol{\beta}_{n+1}^{\sigma}$ are real-valued vector fields to weakly enforce the balance of linear momentum and traction boundary conditions, respectively, and $\beta_{n+1}^{\text{mass}}$ and β_{n+1}^q are real-valued scalar fields to weakly enforce conservation of mass, and normal Darcy's velocity boundary conditions. These terms are obtained as follows:

$$\mathcal{L}_{\text{loss}}^{\text{DD}} = \int_{\Omega} (d^{\text{sf}}(\mathbf{z}_{n+1}^{\text{sf}*}, \mathbf{z}_{n+1}^{\text{sf}}))^2 d\Omega, \quad (25)$$

$$\begin{aligned} \mathcal{L}_{\text{momentum}}^{\text{DD}} &= \int_{\Omega} \boldsymbol{\beta}_{n+1}^{\text{mom}} \cdot (\nabla^x \cdot (\boldsymbol{\sigma}'_{n+1} - Bp_{n+1}\mathbf{I}) + \boldsymbol{\gamma}_{n+1}) d\Omega \\ &\quad + \int_{\partial\Omega_{\sigma}} \boldsymbol{\beta}_{n+1}^{\sigma} \cdot ((\boldsymbol{\sigma}'_{n+1} - Bp_{n+1}\mathbf{I}) \cdot \mathbf{n} - \bar{\mathbf{t}}_{n+1}) d\Gamma, \end{aligned} \quad (26)$$

$$\mathcal{L}_{\text{mass}}^{\text{DD}} = \int_{\Omega} \beta_{n+1}^{\text{mass}} \left(\frac{p_{n+1}}{M} + B\epsilon_{\text{vol}n+1} + \nabla^x \cdot \mathbf{q}_{n+1} \Delta t + s_{n+1} \Delta t - \frac{p_n}{M} - B\epsilon_{\text{vol}n} \right) d\Omega \quad (27)$$

$$+ \int_{\partial\Omega_q} \beta_{n+1}^q (\mathbf{q}_{n+1} \cdot \mathbf{n} - \bar{q}_{n+1}) d\Gamma. \quad (28)$$

Taking the first variation of Eq. (24), using common rules of the calculus of variations [34,57], after applying the divergence theorem leads to

$$\begin{aligned} \delta\mathcal{L}_{\text{tot}}^{\text{DD}} &= \delta\mathcal{L}_{\text{loss}}^{\text{DD}} + \delta\mathcal{L}_{\text{momentum}}^{\text{DD}} + \delta\mathcal{L}_{\text{mass}}^{\text{DD}} \\ &= \delta\mathcal{L}_u^{\text{DD}} + \delta\mathcal{L}_p^{\text{DD}} + \delta\mathcal{L}_{\beta^{\text{mom}}}^{\text{DD}} + \delta\mathcal{L}_{\beta^{\text{mass}}}^{\text{DD}} + \delta\mathcal{L}_{\sigma'}^{\text{DD}} + \delta\mathcal{L}_q^{\text{DD}} + \delta\mathcal{L}_{\beta^{\sigma}}^{\text{DD}} + \delta\mathcal{L}_{\beta^q}^{\text{DD}} = 0, \end{aligned}$$

where each contribution is as follows:

$$\begin{aligned} \delta\mathcal{L}_u^{\text{DD}} &= \int_{\Omega} \delta\mathbf{u}_{n+1} \cdot \frac{\partial \boldsymbol{\epsilon}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} : \mathbb{C}_s : (\boldsymbol{\epsilon}_{n+1} - \boldsymbol{\epsilon}_{n+1}^*) d\Omega + \int_{\Omega} B \delta\mathbf{u}_{n+1} \cdot \frac{\partial \epsilon_{\text{vol}}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} \beta_{n+1}^{\text{mass}} d\Omega, \\ \delta\mathcal{L}_p^{\text{DD}} &= \int_{\Omega} \nabla^x \delta p_{n+1} \cdot \mathbf{C}_f \cdot (\mathbf{r}_{n+1} - \mathbf{r}_{n+1}^*) \Delta t d\Omega + \int_{\Omega} \frac{1}{M} \delta p_{n+1} \beta_{n+1}^{\text{mass}} d\Omega + \int_{\Omega} B \delta p_{n+1} \nabla^x \boldsymbol{\beta}_{n+1}^{\text{mom}} : \mathbf{I} d\Omega \\ &\quad - \int_{\partial\Omega_{\sigma}} \delta p_{n+1} B \boldsymbol{\beta}_{n+1}^{\sigma} \cdot \mathbf{n} d\Gamma - \int_{\partial\Omega} \delta p_{n+1} B \boldsymbol{\beta}_{n+1}^{\text{mom}} \cdot \mathbf{n} d\Gamma, \\ \delta\mathcal{L}_{\beta^{\text{mom}}}^{\text{DD}} &= - \int_{\Omega} \nabla^x \delta \boldsymbol{\beta}_{n+1}^{\text{mom}} : (\boldsymbol{\sigma}'_{n+1} - Bp_{n+1}\mathbf{I}) d\Omega + \int_{\partial\Omega} \delta \boldsymbol{\beta}_{n+1}^{\text{mom}} \cdot (\boldsymbol{\sigma}'_{n+1} - Bp_{n+1}\mathbf{I}) \cdot \mathbf{n} d\Gamma \\ &\quad + \int_{\Omega} \delta \boldsymbol{\beta}_{n+1}^{\text{mom}} \cdot \boldsymbol{\gamma}_{n+1} d\Omega, \\ \delta\mathcal{L}_{\beta^{\text{mass}}}^{\text{DD}} &= \int_{\Omega} \delta \beta_{n+1}^{\text{mass}} \left[\frac{1}{M} (p_{n+1} - p_n) + B(\epsilon_{\text{vol}n+1} - \epsilon_{\text{vol}n}) + s_{n+1} \Delta t \right] d\Omega - \int_{\Omega} \nabla^x \delta \beta_{n+1}^{\text{mass}} \cdot \mathbf{q}_{n+1} \Delta t d\Omega \\ &\quad + \int_{\partial\Omega} \delta \beta_{n+1}^{\text{mass}} \mathbf{q}_{n+1} \cdot \mathbf{n} \Delta t d\Gamma, \\ \delta\mathcal{L}_{\sigma'}^{\text{DD}} &= \int_{\Omega} \delta \boldsymbol{\sigma}'_{n+1} : \mathbb{S}_s : (\boldsymbol{\sigma}'_{n+1} - \boldsymbol{\sigma}'_{n+1}^*) d\Omega - \int_{\Omega} \delta \boldsymbol{\sigma}'_{n+1} : \nabla^x \boldsymbol{\beta}_{n+1}^{\text{mom}} d\Omega + \int_{\partial\Omega_{\sigma}} \delta \boldsymbol{\sigma}'_{n+1} : (\boldsymbol{\beta}_{n+1}^{\sigma} \otimes \mathbf{n}) d\Gamma \\ &\quad + \int_{\partial\Omega} \delta \boldsymbol{\sigma}'_{n+1} : (\boldsymbol{\beta}_{n+1}^{\text{mom}} \otimes \mathbf{n}) d\Gamma, \end{aligned}$$

$$\begin{aligned}
\delta \mathcal{L}_q^{\text{DD}} &= \int_{\Omega} \delta \mathbf{q}_{n+1} \cdot \mathbf{S}_f \cdot (\mathbf{q}_{n+1} - \mathbf{q}_{n+1}^*) \Delta t d\Omega - \int_{\Omega} \delta \mathbf{q}_{n+1} \cdot \nabla^x \beta^{\text{mass}} \Delta t d\Omega + \int_{\partial \Omega_q} \delta \mathbf{q}_{n+1} \cdot \beta^q \mathbf{n} d\Gamma \\
&\quad + \int_{\partial \Omega} \delta \mathbf{q}_{n+1} \cdot \beta^{\text{mass}} \mathbf{n} \Delta t d\Gamma, \\
\delta \mathcal{L}_{\beta^\sigma}^{\text{DD}} &= \int_{\partial \Omega_\sigma} \delta \beta_{n+1}^\sigma \cdot ((\boldsymbol{\sigma}'_{n+1} - B p_{n+1} \mathbf{I}) \cdot \mathbf{n} - \bar{\mathbf{t}}_{n+1}) d\Gamma, \\
\delta \mathcal{L}_{\beta^q}^{\text{DD}} &= \int_{\partial \Omega_q} \delta \beta_{n+1}^q (\mathbf{q}_{n+1} \cdot \mathbf{n} - \bar{q}_{n+1}) d\Gamma.
\end{aligned}$$

We reduce the number of field variables by setting $\beta_{n+1}^\sigma = -\beta_{n+1}^{\text{mom}}$ defined on the boundary $\partial \Omega_\sigma$ and $\beta_{n+1}^q = -\Delta t \beta_{n+1}^{\text{mass}}$ defined on the boundary $\partial \Omega_q$. After some mathematical manipulations, we obtain the following residuals (corresponding to Euler–Lagrange equations) along with the additional restrictions on fields β_{n+1}^{mom} and $\beta_{n+1}^{\text{mass}}$ as extra boundary conditions $\beta_{n+1}^{\text{mom}} = \mathbf{0}$ on $\partial \Omega_u$ and $\beta_{n+1}^{\text{mass}} = 0$ on $\partial \Omega_p$:

$$\mathcal{R}_{n+1}^u = \int_{\Omega} \delta \mathbf{u}_{n+1} \cdot \frac{\partial \boldsymbol{\epsilon}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} : \mathbb{C}_s : (\boldsymbol{\epsilon}_{n+1} - \boldsymbol{\epsilon}_{n+1}^*) d\Omega + \int_{\Omega} B \delta \mathbf{u}_{n+1} \cdot \frac{\partial \epsilon_{\text{vol}}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} \beta_{n+1}^{\text{mass}} d\Omega = 0, \quad (29)$$

$$\begin{aligned}
\mathcal{R}_{n+1}^p &= \int_{\Omega} \nabla^x \delta p_{n+1} \cdot \mathbf{C}_f \cdot (\mathbf{r}_{n+1} - \mathbf{r}_{n+1}^*) \Delta t d\Omega + \int_{\Omega} \frac{1}{M} \delta p_{n+1} \beta_{n+1}^{\text{mass}} d\Omega \\
&\quad + \int_{\Omega} B \delta p_{n+1} \nabla^x \beta_{n+1}^{\text{mom}} : \mathbf{I} d\Omega = 0,
\end{aligned} \quad (30)$$

$$\mathcal{R}_{n+1}^{\beta^{\text{mom}}} = - \int_{\Omega} \nabla^x \delta \beta_{n+1}^{\text{mom}} : (\boldsymbol{\sigma}'_{n+1} - B p_{n+1} \mathbf{I}) d\Omega + \int_{\partial \Omega_\sigma} \delta \beta_{n+1}^{\text{mom}} \cdot \bar{\mathbf{t}}_{n+1} d\Gamma + \int_{\Omega} \delta \beta_{n+1}^{\text{mom}} \cdot \boldsymbol{\gamma}_{n+1} d\Omega = 0, \quad (31)$$

$$\begin{aligned}
\mathcal{R}_{n+1}^{\beta^{\text{mass}}} &= \int_{\Omega} \delta \beta_{n+1}^{\text{mass}} \left[\frac{1}{M} (p_{n+1} - p_n) + B(\epsilon_{\text{vol}n+1} - \epsilon_{\text{vol}n}) + s_{n+1} \Delta t \right] d\Omega \\
&\quad - \int_{\Omega} \nabla^x \delta \beta_{n+1}^{\text{mass}} \cdot \mathbf{q}_{n+1} \Delta t d\Omega + \int_{\partial \Omega_q} \delta \beta_{n+1}^{\text{mass}} \bar{q}_{n+1} \Delta t d\Gamma = 0,
\end{aligned} \quad (32)$$

$$\mathcal{R}_{n+1}^{\sigma'} = \int_{\Omega} \delta \boldsymbol{\sigma}'_{n+1} : (\mathbb{S}_s : (\boldsymbol{\sigma}'_{n+1} - \boldsymbol{\sigma}'_{n+1}^*) - \nabla^x \beta_{n+1}^{\text{mom}}) d\Omega = 0, \quad (33)$$

$$\mathcal{R}_{n+1}^q = \int_{\Omega} \delta \mathbf{q}_{n+1} \cdot (\mathbf{S}_f \cdot (\mathbf{q}_{n+1} - \mathbf{q}_{n+1}^*) - \nabla^x \beta_{n+1}^{\text{mass}}) \Delta t d\Omega = 0. \quad (34)$$

We further reduce number of independent fields (and equations) by the local (point-wise) satisfaction of Eqs. (33) and (34) via:

$$\boldsymbol{\sigma}'_{n+1} = \boldsymbol{\sigma}'_{n+1}^* + \mathbb{S}_s^{-1} : \nabla^x \beta_{n+1}^{\text{mom}} \text{ in } \Omega, \quad (35)$$

$$\mathbf{q}_{n+1} = \mathbf{q}_{n+1}^* + \mathbf{S}_f^{-1} \cdot \nabla^x \beta_{n+1}^{\text{mass}} \text{ in } \Omega. \quad (36)$$

Remark 2. The coupled system (Eqs. (29), (30), (31), and (32)) is linear even if there is any hidden non-linearity in the database. As such, one may simply store the LU factorization (with pivoting) of the tangential matrix at the beginning and use the decomposition to facilitate the Gaussian elimination and therefore improve the efficiency by avoiding the use of a linear solver at each iteration.

2.3. Option 2: Hybrid data-driven poroelasticity 1 (model-based solid + data-driven fluid solver)

In this section, our goal is to introduce an alternative formulation where the fluid constitutive responses are determined from the data-driven approach whereas the solid constitutive responses are determined from a material model. This treatment is appropriate for a large variety of poroelasticity problems where the confidence interval for any given hydraulic model is expected to be significantly larger than the solid elasticity counterpart after normalization, e.g., see Fig. 3. Examples of these materials include sandstone, clay, rock, and biological tissues where the estimated effective permeability is often considered accurate if it is within the same order of the benchmark values whereas the elasticity error is expected to be much smaller [22].

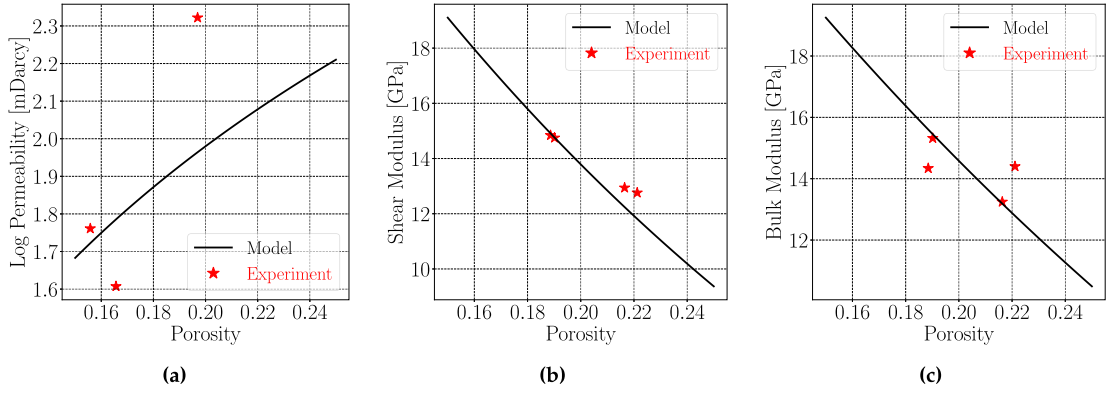


Fig. 3. Black solid lines are model predictions reported in [25]. Red star points are experimental observations reported in [25] for Berea sandstone. The errors between model predictions and experimental data are considerably less for shear and bulk moduli than permeability. Notice that the permeability is plotted in the Log scale, and so the difference between model and experiments is even greater in the linear scale.

Source: These plots are reproduced from [25], see Figs. 2(a), 4(a), and 6(a) in the original reference.

The idea is that we know an appropriate constitutive law for solid deformation, and there is a database \mathcal{D}_{n+1}^f for flow constitutive behavior which is a set of finite pairs of pressure gradient and Darcy's velocity. The following derivation is not restricted to a specific solid constitutive law. The only assumption is that the constitutive law is derived from an energy potential in the context of hyperelasticity for the small deformation limit.

Since the data-driven part is only accounted for the hydraulic constitutive law, the defined norm for the poroelasticity phase space Eq. (12) includes only the fluid contribution. We incorporate solid constitutive model as additional constraint in the set $\mathcal{C}_{n+1}^{\text{momentum}}$ defined in the fully data-driven formulation, and we designate the new set by $\bar{\mathcal{C}}_{n+1}^{\text{momentum}}$ to distinguish them. The resultant problem statement for this hybrid option reads,

$$\bar{\mathbf{z}}_{n+1}^{\text{sf}} = \arg \left\{ \min_{\bar{\mathcal{C}}_{n+1}^{\text{coupled}}} \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{D}_{n+1}^f} \left\| \mathbf{z}_{n+1}^{\text{sf}} - \mathbf{z}_{n+1}^f \right\|_{\mathcal{Z}^f}^2 \right\}, \quad (37)$$

where $\bar{\mathcal{C}}_{n+1}^{\text{coupled}} = \bar{\mathcal{C}}_{n+1}^{\text{momentum}} \cap \mathcal{C}_{n+1}^{\text{mass}}$. Recall that $\mathbf{z}_{n+1}^{\text{sf}}$ encodes the fluid-related variables of $\mathbf{z}_{n+1}^{\text{sf}}$, i.e., pressure gradient and Darcy's velocity. The norm $\|\cdot\|_{\mathcal{Z}^f}$ defined over the fluid phase space \mathcal{Z}^f is the same as Eq. (14). Note that the unit of the objective function in Eq. (37) is power, but it is energy for the fully data-driven Eq. (10). Following the same procedure described in the fully data-driven formulation, we solve the above double-minimization by the fixed-point method consisting of global and local steps.

For the global minimization step we have:

$$\mathcal{L}_{\text{tot}}^{\text{HYB1}}(\mathbf{z}_{n+1}^{\text{sf}}, \mathcal{B}_{n+1}; \mathbf{z}_{n+1}^{\text{sf}*}) = \mathcal{L}_{\text{loss}}^{\text{HYB1}} + \mathcal{L}_{\text{momentum}}^{\text{HYB1}} + \mathcal{L}_{\text{mass}}^{\text{HYB1}}, \quad (38)$$

where $\mathcal{L}_{\text{momentum}}^{\text{HYB1}}$ is almost the same as $\mathcal{L}_{\text{momentum}}^{\text{DD}}$ defined in Eq. (26) with the only difference that the effective stress term is replaced by the constitutive relation; there exists a potential $\psi(\epsilon)$ such that $\sigma'(\epsilon) = \frac{\partial \psi}{\partial \epsilon}$. In other words, the constitutive relation is imposed strongly (point-wise) herein. Notice that if there is a need to define an effective stress field as an independent field, similar to mixed formulations for elasticity [58], one could weakly impose the constitutive relation by adding its contribution through a tensorial Lagrange multiplier defined over the whole domain. Here, we do not intend to arrive at formulations with strain or stress fields as independent fields. Because such formulations increase the number of unknowns significantly, also they require solution spaces with higher regularity such as Hilbert space $H(\text{div}, \Omega)$ [59–62]; this regularity is needed to fulfill the continuity condition of normal traction between elements while tangential traction can be discontinuous. The term $\mathcal{L}_{\text{mass}}^{\text{HYB1}}$ is exactly the same as $\mathcal{L}_{\text{mass}}^{\text{DD}}$ defined in Eq. (28) since nothing related to the mass balance is changed. According to the norm defined in Eq. (14), the original objective (loss) function is as follows:

$$\mathcal{L}_{\text{loss}}^{\text{HYB1}} = \int_{\Omega} (d^f(\mathbf{z}_{n+1}^{\text{sf}*}, \mathbf{z}_{n+1}^f))^2 d\Omega, \quad (39)$$

where the fluid distance function $d^f(\cdot)$ is defined in Eq. (21). In this hybrid formulation, the effective stress σ'_{n+1} is obtained from a constitutive model, such as an hyperelastic strain–energy functional $\psi(\epsilon(\mathbf{u}))$, a key departure from the fully data-driven formulation in Eq. (24). The corresponding first variation of Eq. (38) reads,

$$\begin{aligned}\delta\mathcal{L}_{\text{tot}}^{\text{HYB1}} &= \delta\mathcal{L}_{\text{loss}}^{\text{HYB1}} + \delta\mathcal{L}_{\text{momentum}}^{\text{HYB1}} + \delta\mathcal{L}_{\text{mass}}^{\text{HYB1}} \\ &= \delta\mathcal{L}_u^{\text{HYB1}} + \delta\mathcal{L}_p^{\text{HYB1}} + \delta\mathcal{L}_{\beta^{\text{mom}}}^{\text{HYB1}} + \delta\mathcal{L}_{\beta^{\text{mass}}}^{\text{HYB1}} + \delta\mathcal{L}_q^{\text{HYB1}} + \delta\mathcal{L}_{\beta^\sigma}^{\text{HYB1}} + \delta\mathcal{L}_{\beta^q}^{\text{HYB1}} = 0,\end{aligned}\quad (40)$$

where:

$$\begin{aligned}\delta\mathcal{L}_u^{\text{HYB1}} &= \int_{\Omega} B\delta\mathbf{u}_{n+1} \cdot \frac{\partial\epsilon_{\text{vol}}(\mathbf{u}_{n+1})}{\partial\mathbf{u}}\beta_{n+1}^{\text{mass}}d\Omega - \int_{\Omega} \delta\mathbf{u}_{n+1} \cdot \nabla^x\beta^{\text{mom}} : \frac{\partial\sigma'(\epsilon_{n+1})}{\partial\epsilon} : \frac{\partial\epsilon(\mathbf{u}_{n+1})}{\partial\mathbf{u}}d\Omega \\ &\quad + \int_{\partial\Omega} \delta\mathbf{u}_{n+1} \cdot \frac{\partial\sigma'(\epsilon_{n+1})}{\partial\epsilon} : \frac{\partial\epsilon(\mathbf{u}_{n+1})}{\partial\mathbf{u}} : (\beta^{\text{mom}} \otimes \mathbf{n})d\Gamma \\ &\quad - \int_{\partial\Omega_\sigma} \delta\mathbf{u}_{n+1} \cdot \frac{\partial\sigma'(\epsilon_{n+1})}{\partial\epsilon} : \frac{\partial\epsilon(\mathbf{u}_{n+1})}{\partial\mathbf{u}} : (\beta^\sigma \otimes \mathbf{n})d\Gamma, \\ \delta\mathcal{L}_p^{\text{HYB1}} &= \int_{\Omega} \nabla^x\delta p_{n+1} \cdot \mathbf{C}_f \cdot (\mathbf{r}_{n+1} - \mathbf{r}_{n+1}^*)d\Omega + \int_{\Omega} \frac{1}{M}\delta p_{n+1}\beta_{n+1}^{\text{mass}}d\Omega + \int_{\Omega} B\delta p_{n+1}\nabla^x\beta_{n+1}^{\text{mom}} : \mathbf{I}d\Omega \\ &\quad - \int_{\partial\Omega_\sigma} \delta p_{n+1}B\beta_{n+1}^\sigma \cdot \mathbf{n}d\Gamma - \int_{\partial\Omega} \delta p_{n+1}B\beta_{n+1}^{\text{mom}} \cdot \mathbf{n}d\Gamma, \\ \delta\mathcal{L}_q^{\text{HYB1}} &= \int_{\Omega} \delta\mathbf{q}_{n+1} \cdot \mathbf{S}_f \cdot (\mathbf{q}_{n+1} - \mathbf{q}_{n+1}^*)d\Omega - \int_{\Omega} \delta\mathbf{q}_{n+1} \cdot \nabla^x\beta^{\text{mass}}\Delta t d\Omega + \int_{\partial\Omega_q} \delta\mathbf{q}_{n+1} \cdot \beta^q \mathbf{n}d\Gamma \\ &\quad + \int_{\partial\Omega} \delta\mathbf{q}_{n+1} \cdot \beta^{\text{mass}}\mathbf{n}\Delta t d\Gamma.\end{aligned}$$

The remaining terms are identical to the fully data-driven counterparts, i.e., $\delta\mathcal{L}_{\beta^{\text{mom}}}^{\text{HYB1}} = \delta\mathcal{L}_{\beta^{\text{mom}}}^{\text{DD}}$, $\delta\mathcal{L}_{\beta^{\text{mass}}}^{\text{HYB1}} = \delta\mathcal{L}_{\beta^{\text{mass}}}^{\text{DD}}$, $\delta\mathcal{L}_{\beta^\sigma}^{\text{HYB1}} = \delta\mathcal{L}_{\beta^\sigma}^{\text{DD}}$, and $\delta\mathcal{L}_{\beta^q}^{\text{HYB1}} = \delta\mathcal{L}_{\beta^q}^{\text{DD}}$. For brevity, we refer to those terms defined in the fully data-driven section.

Similar to the fully data driven formulation, we first reduce the number of unknown fields by setting $\beta_{n+1}^\sigma = -\beta_{n+1}^{\text{mom}}$ defined on boundary $\partial\Omega_\sigma$ and $\beta_{n+1}^q = -\Delta t\beta_{n+1}^{\text{mass}}$ defined on boundary $\partial\Omega_q$. After some mathematical manipulations, we obtain the following residuals (corresponding to the Euler–Lagrange equations of Eq. (38)) along with the additional restrictions on fields β_{n+1}^{mom} and $\beta_{n+1}^{\text{mass}}$ as extra boundary conditions $\beta_{n+1}^{\text{mom}} = \mathbf{0}$ on $\partial\Omega_u$ and $\beta_{n+1}^{\text{mass}} = 0$ on $\partial\Omega_p$:

$$\mathcal{R}_{n+1}^u = \int_{\Omega} B\delta\mathbf{u}_{n+1} \cdot \frac{\partial\epsilon_{\text{vol}}(\mathbf{u}_{n+1})}{\partial\mathbf{u}}\beta_{n+1}^{\text{mass}}d\Omega - \int_{\Omega} \delta\mathbf{u}_{n+1} \cdot \nabla^x\beta^{\text{mom}} : \frac{\partial\sigma'(\epsilon_{n+1})}{\partial\epsilon} : \frac{\partial\epsilon(\mathbf{u}_{n+1})}{\partial\mathbf{u}}d\Omega = 0, \quad (41)$$

$$\begin{aligned}\mathcal{R}_{n+1}^p &= \int_{\Omega} \nabla^x\delta p_{n+1} \cdot \mathbf{C}_f \cdot (\mathbf{r}_{n+1} - \mathbf{r}_{n+1}^*)\Delta t d\Omega + \int_{\Omega} \frac{1}{M}\delta p_{n+1}\beta_{n+1}^{\text{mass}}d\Omega \\ &\quad + \int_{\Omega} B\delta p_{n+1}\nabla^x\beta_{n+1}^{\text{mom}} : \mathbf{I}d\Omega = 0,\end{aligned}\quad (42)$$

$$\begin{aligned}\mathcal{R}_{n+1}^{\beta^{\text{mom}}} &= - \int_{\Omega} \nabla^x\delta\beta_{n+1}^{\text{mom}} : \left(\frac{\partial\psi(\epsilon_{n+1})}{\partial\epsilon} - Bp_{n+1}\mathbf{I}\right)d\Omega + \int_{\partial\Omega_\sigma} \delta\beta_{n+1}^{\text{mom}} \cdot \bar{\mathbf{t}}_{n+1}d\Gamma \\ &\quad + \int_{\Omega} \delta\beta_{n+1}^{\text{mom}} \cdot \boldsymbol{\gamma}_{n+1}d\Omega = 0,\end{aligned}\quad (43)$$

$$\begin{aligned}\mathcal{R}_{n+1}^{\beta^{\text{mass}}} &= \int_{\Omega} \delta\beta_{n+1}^{\text{mass}} \left[\frac{1}{M}(p_{n+1} - p_n) + B(\epsilon_{\text{vol},n+1} - \epsilon_{\text{vol},n}) + s_{n+1}\Delta t \right]d\Omega \\ &\quad - \int_{\Omega} \nabla^x\delta\beta_{n+1}^{\text{mass}} \cdot \mathbf{q}_{n+1}\Delta t d\Omega + \int_{\partial\Omega_q} \delta\beta_{n+1}^{\text{mass}}\bar{\mathbf{q}}_{n+1}\Delta t d\Gamma = 0,\end{aligned}\quad (44)$$

$$\mathcal{R}_{n+1}^q = \int_{\Omega} \delta\mathbf{q}_{n+1} \cdot (\mathbf{S}_f \cdot (\mathbf{q}_{n+1} - \mathbf{q}_{n+1}^*) - \Delta t\nabla^x\beta^{\text{mass}})d\Omega = 0. \quad (45)$$

We further reduce number of independent fields (and equations) by the local (point-wise) satisfaction of Eq. (45) via:

$$\mathbf{q}_{n+1} = \mathbf{q}_{n+1}^* + \Delta t\mathbf{S}_f^{-1} \cdot \nabla^x\beta_{n+1}^{\text{mass}} \text{ in } \Omega. \quad (46)$$

As mentioned in the previous section, the next step in the fixed-point method is to solve the local minimization problem. According to Eq. (37), the local minimization for this hybrid scheme reads as follows,

$$\mathbf{z}_{n+1}^{f*}(\bar{\mathbf{x}}_a) = \arg \min_{\mathbf{z}_{n+1}^f \in \mathcal{D}_{n+1}^f} (d^f(\mathbf{z}_{n+1}^{f*}, \mathbf{z}_{n+1}^f(\bar{\mathbf{x}}_a))^2, \text{ given } \mathbf{z}_{n+1}^f(\bar{\mathbf{x}}_a) \in \bar{\mathcal{C}}_{n+1}^{\text{coupled}}, \quad (47)$$

where $\bar{\mathbf{x}}_a \in \Omega^h$ is an integration point in Ω^h . We will discuss how to solve the local sub-problem defined at each integration point in Section 3.

Notice that, in the data-driven formulation there are three major factors that determine computational costs: data availability (gathering data), global optimization (solving a system of equations), local optimization (searching inside a database). The local optimization part is an NP-hard problem, and its computational cost grows exponentially by increasing the database size. Therefore, if the number of unknowns (degree of freedoms) is not so high while the database required for data-driven schemes is large, local optimization is the dominant source of computational cost. As a result, the hybrid formulation could be more efficient than the fully data-driven counterpart if the solid behavior can be accurately captured by a constitutive law with an acceptable standard deviation from the ground-truth. We will discuss the computational issues concerning the local optimization step in Section 3.

2.4. Option 3: Hybrid data-driven poroelasticity 2 (data-driven solid + model-based fluid solver)

For comparison purposes, we present another hybrid formulation that includes Darcy's law as the fluid constitutive model, whereas the solid deformation is predicted directly from a collection of data points. We start with the fully data-driven problem defined in Eq. (10) and adjust it to be compatible with this hybrid formulation that considers just solid part as data-driven. To this end, we add Darcy's equation as an additional constraint in the set $\mathcal{C}_{n+1}^{\text{mass}}$ defined for the fully data-driven formulation. Darcy's law reads as follows:

$$\mathbf{q}_{n+1} = -\frac{1}{\mu^f} \mathbf{k}(\nabla^x p_{n+1} + \mathbf{y}_{n+1}^f), \quad (48)$$

where μ^f and \mathbf{k} are dynamic viscosity of pore fluid and intrinsic permeability tensor with unit $[\text{Length}^2]$, respectively. Besides, the contribution of the fluid-related term in the norm defined in Eq. (12) should be excluded since only solid constitutive behavior is data-driven. Therefore, the problem definition in this hybrid formulation is as follows:

$$\bar{\mathbf{z}}_{n+1}^{\text{sf}} = \arg \min_{\hat{\mathcal{C}}_{n+1}^{\text{coupled}}} \left\{ \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \mathcal{D}_{n+1}^{\text{sf}}} \min_{\mathbf{z}_{n+1}^{\text{sf}} \in \hat{\mathcal{C}}_{n+1}^{\text{coupled}}} \|\mathbf{z}_{n+1}^{s*} - \mathbf{z}_{n+1}^{\text{sf}}\|_{\mathcal{Z}^s}^2 \right\}, \quad (49)$$

where $\hat{\mathcal{C}}_{n+1}^{\text{coupled}} = \mathcal{C}_{n+1}^{\text{momentum}} \cap \hat{\mathcal{C}}_{n+1}^{\text{mass}}$ and $\hat{\mathcal{C}}_{n+1}^{\text{mass}}$ is almost the same as $\mathcal{C}_{n+1}^{\text{mass}}$ defined for fully data-driven formulation except it has an additional constraint from the hydraulic constitutive law (Darcy's law). Recall that \mathbf{z}_{n+1}^{s*} encodes the solid-related variables of $\mathbf{z}_{n+1}^{\text{sf}}$, i.e., strain and effective stress. The norm $\|\cdot\|_{\mathcal{Z}^s}$ defined over the solid phase space \mathcal{Z}^s is the same as Eq. (13). Similarly, we solve the above double-minimization by the fixed-point method consisting global and local steps.

According to Eq. (49), the total objective function in the global minimization reads as:

$$\mathcal{L}_{\text{tot}}^{\text{HYB2}}(\mathbf{z}_{n+1}^{\text{sf}}, \mathcal{B}_{n+1}; \mathbf{z}_{n+1}^{s*}) = \mathcal{L}_{\text{loss}}^{\text{HYB2}} + \mathcal{L}_{\text{momentum}}^{\text{HYB2}} + \mathcal{L}_{\text{mass}}^{\text{HYB2}}, \quad (50)$$

where $\mathcal{L}_{\text{mass}}^{\text{HYB2}}$ is almost the same as $\mathcal{L}_{\text{mass}}^{\text{DD}}$ defined in Eq. (28) with the only difference that the Darcy's velocity term is replaced by Eq. (48). The term $\mathcal{L}_{\text{momentum}}^{\text{HYB2}}$ is exactly the same as $\mathcal{L}_{\text{momentum}}^{\text{DD}}$ defined in Eq. (26) since nothing related to the balance of linear-momentum is changed. The original objective function $\mathcal{L}_{\text{loss}}^{\text{HYB2}}$ in this hybrid formulation is the norm defined over solid phase space Eq. (13) as follows:

$$\mathcal{L}_{\text{loss}}^{\text{HYB2}} = \int_{\Omega} (d^s(\mathbf{z}_{n+1}^{s*}, \mathbf{z}_{n+1}^s))^2 d\Omega, \quad (51)$$

where solid distance function $d^s(\cdot)$ is defined in Eq. (20).

Recall that, here, the Darcy velocity \mathbf{q}_{n+1} is not an independent field in Eq. (50) which is not the case in Eq. (24). It relates to the pressure gradient field via the Darcy's law. Taking the first variation of Eq. (50) after applying the divergence theorem leads to:

$$\delta \mathcal{L}_{\text{tot}}^{\text{HYB2}} = \delta \mathcal{L}_{\text{loss}}^{\text{HYB2}} + \delta \mathcal{L}_{\text{momentum}}^{\text{HYB2}} + \delta \mathcal{L}_{\text{mass}}^{\text{HYB2}}$$

$$= \delta \mathcal{L}_u^{\text{HYB2}} + \delta \mathcal{L}_p^{\text{HYB2}} + \delta \mathcal{L}_{\beta^{\text{mom}}}^{\text{HYB2}} + \delta \mathcal{L}_{\beta^{\text{mass}}}^{\text{HYB2}} + \delta \mathcal{L}_{\sigma'}^{\text{HYB2}} + \delta \mathcal{L}_{\beta^\sigma}^{\text{HYB2}} + \delta \mathcal{L}_{\beta^q}^{\text{HYB2}} = 0,$$

where:

$$\begin{aligned} \delta \mathcal{L}_p^{\text{HYB2}} = & \int_{\Omega} \nabla^x \delta p \cdot \mathbf{k} \cdot \nabla^x \beta^{\text{mass}} \frac{\Delta t}{\mu^f} d\Omega - \int_{\partial\Omega} \nabla^x \delta p \cdot \mathbf{k} \cdot \beta^{\text{mass}} \mathbf{n} \frac{\Delta t}{\mu^f} d\Omega - \int_{\partial\Omega_q} \nabla^x \delta p \cdot \mathbf{k} \cdot \beta^q \mathbf{n} \frac{1}{\mu^f} d\Omega \\ & + \int_{\Omega} \frac{1}{M} \delta p_{n+1} \beta_{n+1}^{\text{mass}} d\Omega + \int_{\Omega} B \delta p_{n+1} \nabla^x \beta_{n+1}^{\text{mom}} : \mathbf{I} d\Omega - \int_{\partial\Omega_\sigma} \delta p_{n+1} B \beta_{n+1}^\sigma \cdot \mathbf{n} d\Gamma \\ & - \int_{\partial\Omega} \delta p_{n+1} B \beta_{n+1}^{\text{mom}} \cdot \mathbf{n} d\Gamma, \end{aligned}$$

and other terms remain similar to the fully data-driven formulation, i.e., $\delta \mathcal{L}_u^{\text{HYB2}} = \delta \mathcal{L}_u^{\text{DD}}$, $\delta \mathcal{L}_{\beta^{\text{mom}}}^{\text{HYB2}} = \delta \mathcal{L}_{\beta^{\text{mom}}}^{\text{DD}}$, $\delta \mathcal{L}_{\beta^{\text{mass}}}^{\text{HYB2}} = \delta \mathcal{L}_{\beta^{\text{mass}}}^{\text{DD}}$, $\delta \mathcal{L}_{\sigma'}^{\text{HYB2}} = \delta \mathcal{L}_{\sigma'}^{\text{DD}}$, $\delta \mathcal{L}_{\beta^\sigma}^{\text{HYB2}} = \delta \mathcal{L}_{\beta^\sigma}^{\text{DD}}$, and $\delta \mathcal{L}_{\beta^q}^{\text{HYB2}} = \delta \mathcal{L}_{\beta^q}^{\text{DD}}$.

Similar to the fully data driven formulation, we first reduce the number of unknown fields by setting $\beta_{n+1}^\sigma = -\beta_{n+1}^{\text{mom}}$ defined on boundary $\partial\Omega_\sigma$ and $\beta_{n+1}^q = -\Delta t \beta_{n+1}^{\text{mass}}$ defined on boundary $\partial\Omega_q$. After some mathematical manipulations, we obtain the following residuals (corresponding to Euler–Lagrange equations) along with the additional restrictions on fields β_{n+1}^{mom} and $\beta_{n+1}^{\text{mass}}$ as extra boundary conditions $\beta_{n+1}^{\text{mom}} = \mathbf{0}$ on $\partial\Omega_u$ and $\beta_{n+1}^{\text{mass}} = 0$ on $\partial\Omega_p$:

$$\mathcal{R}_{n+1}^u = \int_{\Omega} \delta \mathbf{u}_{n+1} \cdot \frac{\partial \boldsymbol{\epsilon}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} : \mathbb{C}_s : (\boldsymbol{\epsilon}_{n+1} - \boldsymbol{\epsilon}_{n+1}^*) d\Omega + \int_{\Omega} B \delta \mathbf{u}_{n+1} \cdot \frac{\partial \epsilon_{\text{vol}}(\mathbf{u}_{n+1})}{\partial \mathbf{u}} \beta_{n+1}^{\text{mass}} d\Omega = 0, \quad (52)$$

$$\begin{aligned} \mathcal{R}_{n+1}^p = & \int_{\Omega} \frac{1}{\mu^f} \nabla^x \delta p_{n+1} \cdot \mathbf{k} \cdot \nabla^x \beta_{n+1}^{\text{mass}} \Delta t d\Omega + \int_{\Omega} \frac{1}{M} \delta p_{n+1} \beta_{n+1}^{\text{mass}} d\Omega \\ & + \int_{\Omega} B \delta p_{n+1} \nabla^x \beta_{n+1}^{\text{mom}} : \mathbf{I} d\Omega = 0, \end{aligned} \quad (53)$$

$$\mathcal{R}_{n+1}^{\beta^{\text{mom}}} = - \int_{\Omega} \nabla^x \delta \beta_{n+1}^{\text{mom}} : (\boldsymbol{\sigma}'_{n+1} - B p_{n+1} \mathbf{I}) d\Omega + \int_{\partial\Omega_\sigma} \delta \beta_{n+1}^{\text{mom}} \cdot \bar{\mathbf{t}}_{n+1} d\Gamma + \int_{\Omega} \delta \beta_{n+1}^{\text{mom}} \cdot \boldsymbol{\gamma}_{n+1} d\Omega = 0, \quad (54)$$

$$\begin{aligned} \mathcal{R}_{n+1}^{\beta^{\text{mass}}} = & \int_{\Omega} \delta \beta_{n+1}^{\text{mass}} \left[\frac{1}{M} (p_{n+1} - p_n) + B (\epsilon_{\text{vol},n+1} - \epsilon_{\text{vol},n}) + s_{n+1} \Delta t \right] d\Omega \\ & - \int_{\Omega} \nabla^x \delta \beta_{n+1}^{\text{mass}} \cdot \mathbf{q}_{n+1} \Delta t d\Omega + \int_{\partial\Omega_q} \delta \beta_{n+1}^{\text{mass}} \bar{q}_{n+1} \Delta t d\Gamma = 0, \end{aligned} \quad (55)$$

$$\mathcal{R}_{n+1}^{\sigma'} = \int_{\Omega} \delta \boldsymbol{\sigma}'_{n+1} : (\mathbb{S}_s : (\boldsymbol{\sigma}'_{n+1} - \boldsymbol{\sigma}'_{n+1}^*) - \nabla^x \beta_{n+1}^{\text{mom}}) d\Omega = 0. \quad (56)$$

We further reduce number of independent fields (and equations) by the local (point-wise) satisfaction of Eq. (56) via:

$$\boldsymbol{\sigma}'_{n+1} = \boldsymbol{\sigma}'_{n+1}^* + \mathbb{S}_s^{-1} : \nabla^x \beta_{n+1}^{\text{mom}} \text{ in } \Omega. \quad (57)$$

The next step in the fixed-point method after solving the above coupled equations is to solve the local minimization problem. According to Eq. (49), the local minimization for this hybrid scheme reads as follow,

$$\mathbf{z}_{n+1}^{s*}(\bar{\mathbf{x}}_a) = \arg \min_{\mathbf{z}_{n+1}^s \in \mathcal{D}_{n+1}^s} (d^s(\mathbf{z}_{n+1}^{s*}, \mathbf{z}_{n+1}^s(\bar{\mathbf{x}}_a)))^2, \text{ given } \mathbf{z}_{n+1}^s(\bar{\mathbf{x}}_a) \in \hat{\mathcal{C}}_{n+1}^{\text{coupled}}, \quad (58)$$

where $\bar{\mathbf{x}}_a \in \Omega^h$ is an integration point in the domain Ω^h , i.e., Gauss quadrature point. We will discuss how to solve this local problem in Section 3.

Remark 3. In the global minimization problem, the combination of the finite-dimensional spaces used to interpolate the weighting function and the solution fields is significant for the stability and convergence of the solution. In this work, we employ only the linear basis function for all solution fields. This choice is mainly attributed to the ease of implementation. However, like other multi-field saddle point problems, the usage of equal-order discretization may lead to spurious oscillation for both the pore pressure and the Lagrange Multipliers [63–68]. Our numerical experiments have indicated that the spurious oscillation of pore pressure usually disappears in the diffusion-dominated regime or when the time step is large, whereas the oscillation of the Lagrange multiplier may

persist. Potential remedies to circumvent this problem may include deriving proper procedures for analysis on the multi-fold inf-sup condition for the multi-field problems (cf. [68]), deriving inf-sup tests for the product norm space (cf. [63], introducing proper discretization or composite element to ensure stability and convergence (e.g. [69]) or introducing rigorous analysis to introduce stabilization procedure to guarantee stability and convergence. A comprehensive study to evaluate some of these options are important for the global minimization step for both the single- and multi-physics data-driven problems but nevertheless out of the scope of this study.

3. Local minimization: physical response projection onto material space

This section describes the procedure to project the hydro-mechanical responses obtained in the global minimization step onto the points that belong to the material phase space for each integration point. The data-driven solver searches for the nearest neighbor point to a hydro-mechanical response inside the material phase space (database). Since the local minimization problems for the three formulations defined in Sections 2.2–2.4 are similar (except for the distance measure and the basis of the material space), the local minimization description is generic.

In this research, the nearest neighbor search (NNS) is used to facilitate this minimization step. In Section 3.1, we discuss alternative schemes for the fast NNS and explain why the k-d tree is an appropriate choice. We then show how the energy metric can be rewritten as a Euclidean metric through a linear projection. This mapping provides a straightforward way of using available open-source packages for the fast NNS since these packages are generally developed and optimized for Minkowski's metric functions.

3.1. Nearest neighbor search

Here, we first describe the Nearest Neighbor Search (NNS), which enables us to locate the optimal data points in the phase space (ϵ, σ') and $(\nabla^x p, q)$ equipped with different distance metrics. We also provide an overview and compare different NNS approaches and justify our choice of using the k-d tree data structure for the data-driven poroelasticity problem.

Assume P is a set of n points $p_i \in \mathbb{R}^k$ with k dimensions, i.e., $P = \{p_1, p_2, \dots, p_n\}$. Nearest Neighbor Search (NNS) in a metric space is an optimization problem that determines the closest point $p_i \in P$ to a query point $q \in \mathbb{R}^k$ where P and q belong to a metric space M equipped with a well-defined distance function $d: M \times M \rightarrow \mathbb{R}^+$ [70,71], i.e.,

$$\text{NNS}(q, P; M) = \arg \min_{p_i \in P} d(q, p_i), \quad (59)$$

where a distance function must possess the following properties:

$$d(q, p_i) = 0 \Leftrightarrow q \equiv p_i, \quad (60)$$

$$d(q, p_i) = d(p_i, q), \quad (61)$$

$$d(q, p_j) \leq d(q, p_i) + d(p_i, p_j). \quad (62)$$

In our application, for example, point p_i encodes components of strain–stress pair. In three-dimensional cases, $k = 12$ since both strain and stress are symmetric. A Brute-force algorithm that compares every element in the set is easy to implement but is inefficient for large sets P with a huge amount of queries, due to the resultant linear query time $O(nk)$. There are algorithms developed to make the search fast enough to be close to a logarithmic query time for a variety of applications and circumstances [70–72]. A fast algorithm for NNS relies heavily on leveraging efficient data structures. However, more efficient data structures could also be more complex and less memory-efficient, known as time–space trade-offs [73]. For instance, if one intends to query the distance between two points of a set of n points, a naive approach is to store n points in an array with the space complexity $O(n)$ and each time computes the distance of two points with the time complexity $O(n)$ (access time complexity). However, the distance between all points can be tabulated with the space complexity $O(n^2)$ in the pre-processing step, and the query time becomes $O(1)$ in this case. Consequently, an efficient algorithm for the search must strike a balance among speed, memory efficiency, robustness, and accuracy.

For any specific application number of data points in P , dimensionality of points, metric structure, and number of query points are among the first most important factors needed to be considered for choosing an efficient algorithm

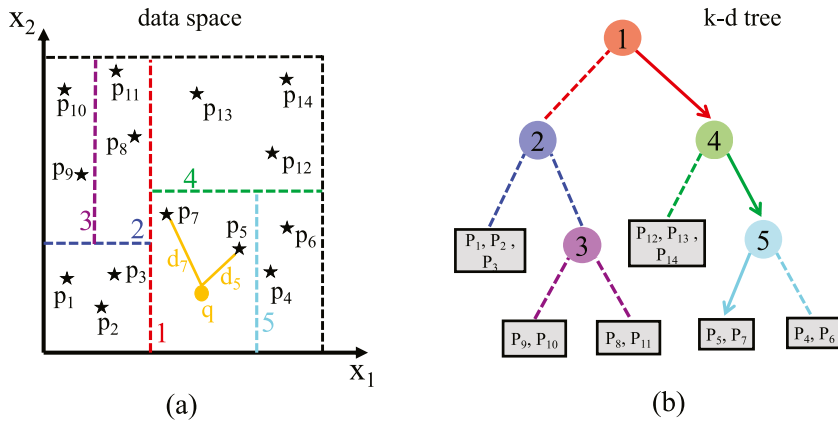


Fig. 4. (a) partitioned 2D data space by some orthogonal lines; (b) equivalent representation of (a) by a k-d tree data structure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(cf. [71]). For example, brute-force is the most efficient approach when data size is small [71]. As the number of query points or size of data set P is growing the need for other efficient NNS increases.

There exist two types of fast algorithms: exact NNS and approximate NNS (ANNS). Exact NNS are based on the hierarchical space partitioning [74,75]. There are many variants of exact NNS, each of which proposes a different approach to split high dimensional space. Tree data structures are recognized as the efficient data structures for these type of algorithms, and k-d tree [74,76], vp-tree [77], R-tree [78], and X-tree [79] are among popular ones. Exact NNS and in general fast algorithms loose their efficiency in dealing with high dimensional spaces, known as the curse of dimensionality. Roughly speaking, sub-linear or logarithmic time complexity of fast algorithms increases as n exceeds 2^k limit for high dimensional spaces [80,81]. Note that in computer science $k = 20$ is still not considered as a high dimensional space [80,81] whereas in mechanics the space of strain–stress pairs with dimensionality 12 for 3D geometry is assessed as a high dimensional space. On the other side, it is proven that speed can be significantly improved by ANNS comparing with exact NNS if we slightly sacrifice the accuracy [70]. This means ANNS finds a point which is in the ϵ -ball of the exact closest point. Algorithms based on hashing [82], randomized tree [83], and hierarchical graphs [72] are widely used for ANNS.

One efficient data structure suitable for low dimensional points ($k < 20$) is the k -dimensional tree known as k-d tree [76,81]. Bentley's k-d tree as the standard k-d tree is a binary tree that recursively partitions the \mathbb{R}^k space along some orthogonal hyperplanes into at most $O(\log n)$ hypercubes. Basically, each hyperplane is chosen to be orthogonal to a data axis and bisects the axis in a way that two resulting partitions have almost the same size. In the constructed tree, eventually, a maximum predefined number of data points are stored at each leaf node, and other nodes including root save information of each hyperplane. Once the tree structure is built, for a query point the algorithm traverses the tree from the root, and based on the underlying metric structure and hyperplane configuration narrows down the search into one half of the sub-tree until arrives at a leaf node. At the leaf node, it switches to the brute force distance calculation to find the closest point among the points inside the leaf node. What we intended to describe is a big picture of the idea behind k-d tree family of algorithms, and interested readers should refer to Friedman et al. [74], Bentley [76] and Maneewongvatana and Mount [75] for more details. Note that Ball tree is suggested as an efficient alternative for high dimensional spaces $k > 20$ [81,84] which is not studied in this research.

Fig. 4 illustrates the process of constructing a k-d tree for a data set $P = \{p_1, \dots, p_{14}\}$ with $k = 2$ and querying a new point q . In Fig. 4(a), line 1 (red dashed line) which plays the role of a hyperplane in 2D data space partitions the data set into two rectangles, hypercubes in 2D space, with seven points in each. Next, line 2 splits the left rectangle constructed by line 1 into two smaller rectangles where the bottom one has three points, and the top one has four points. This process is continuing until we end up with some small rectangles which include at most three points, the maximum number of data points in leaf nodes. In Fig. 4(b), we show the constructed k-d tree that describes Fig. 4(a). Geometrical information of each hyperplane is encoded hierarchically from root node towards a level before leaf nodes, e.g., lines 1–5 in Fig. 4(a) are encoded in nodes 1–5 in Fig. 4(b). Leaf nodes store

those data points inside each final hypercube. Note that k-d tree is constructed in an off-line manner. For an online query \mathbf{q} shown as an orange circle in Fig. 4(a), the algorithm traverses the tree from the root node numbered 1 in Fig. 4(b) and at each node it determines which sub-tree holds the closest point to \mathbf{q} , based on the metric structure and hyperplane configuration, until it reaches to one of the leaf nodes. Solid arrows in Fig. 4(b) declare the path from the root to leaf node having the closest point to \mathbf{q} , $1 \rightarrow 4 \rightarrow 5 \rightarrow \text{leaf}$. At the founded leaf node the algorithm calculates the distance function $d_i = d(\mathbf{q}, \mathbf{p}_i)$ to find the nearest point to \mathbf{q} , distances are shown in orange solid lines in Fig. 4(a). Note that the maximum number of points at each leaf node is a hyper-parameter and should not be chosen too large or too small. In this work, we set it equal to 10. Larger values make height of tree shorter and so faster traverse, but it has more cost during the distance calculation over all leaf nodes.

Minkowski's l_p distance metrics are among widely used distance functions in several open-source packages SciPy [84], FLANN [85], Faiss [86], and Annoy [87], due to the well-documented theoretical and practical performance. It is noteworthy that more complex distance functions may increase time complexity for the tree construction and query. For instance, Friedman et al. [74] empirically show that the simplest case $l_p = \infty$, i.e., maximum coordinate distance, outperforms Euclidean distance $l_p = 2$.

Herein, we do not intend to develop a data-structure effective for the energy-like distance functions Eqs. (19), (20), and (21) in the data-driven formulations. Instead, we will introduce an isomorphism that maps the energy-like metric to Euclidean metric, so all the data-structures established for Euclidean metric can be easily utilized in our application.

Remark 4. Bentley's k-d tree has average height $O(\log n)$, average construction time $O(kn \log n)$, average query time $O(\log n)$ for Minkowski's l_p metric [74,75]. The query complexity in worst case scenario is $O(kn^{1-1/k})$ [88,89].

Remark 5. In this work, we utilize k-d tree data structure implemented in class `scipy.spatial.cKDTree` of open source package SciPy [84].

3.2. Local metric minimization

Here, we describe how to perform the local minimization efficiently to utilize techniques available in the NNS literature. First, we focus on the fully data-driven case and demonstrate how the NNS may simplify the local distance calculations. Then, we will formulate the solid and fluid local distance calculation in a unified and compact formulation. Recall the local minimization problem corresponding to the distance function in Eq. (18). Since we assume that the solid and fluid phase spaces are orthogonal to each other, the local minimization can be done separately for each phase space, i.e.,

$$\mathbf{z}^{s*}(\bar{\mathbf{x}}_a) = \arg \min_{\mathbf{z}^{s*} \in \mathcal{D}^s} [d^s(\mathbf{z}^{s*}, \mathbf{z}^s(\bar{\mathbf{x}}_a))]^2; \text{ given } \mathbf{z}^s(\bar{\mathbf{x}}_a), \quad (63)$$

$$\mathbf{z}^{f*}(\bar{\mathbf{x}}_a) = \arg \min_{\mathbf{z}^{f*} \in \mathcal{D}^f} [d^f(\mathbf{z}^{f*}, \mathbf{z}^f(\bar{\mathbf{x}}_a))]^2; \text{ given } \mathbf{z}^f(\bar{\mathbf{x}}_a), \quad (64)$$

where each sub-problem is solved by an independent nearest neighbor search inside the corresponding database. The designated metric for the fully data-driven formulation reduces the phase space dimensionality by splitting a phase space that contains 18 bases (12 for solid and 6 for fluid in 3D cases) into two smaller phase spaces. The advantage of this metric is two-fold. First, the NNS algorithms may perform more efficiently in the two lower-dimensional phase space. Second, these two NNS tasks can be executed in parallel to further reduce computational time.

These local minimization problems should be solved for every quadrature point $\bar{\mathbf{x}}_a$ where $a = 1, 2, \dots, n_{\text{int}}$ where n_{int} is the total number of integration points. These global-local iterations continue until updated variables during the fixed-point iterations converge or their variations remain within a small tolerance. The data-driven solver may perform several fixed-point iterations within a time step. Hence the efficiency of the local minimization is a dominant factor for the speed of the simulations, especially for large-scale problems. In the following, we switch to conventional matrix notation in Linear Algebra. Consequently, the summation convention is no longer hold, in this section, unless we specify the other way around.

Note that both terms in solid distance function Eq. (20) have the same structure as $\boldsymbol{\tau} : \mathbb{K} : \boldsymbol{\tau}$ where $\boldsymbol{\tau}$ is a symmetric 2nd order tensor which is either $\boldsymbol{\tau} = \boldsymbol{\epsilon} - \boldsymbol{\epsilon}^*$ or $\boldsymbol{\tau} = \boldsymbol{\sigma}' - \boldsymbol{\sigma}'^*$, and \mathbb{K} is a symmetric positive definite

4th order tensor which is either \mathbb{C}_s or \mathbb{S}_s . This energy-like scalar variable can be represented in a full vector–matrix notation:

$$\boldsymbol{\tau} : \mathbb{K} : \boldsymbol{\tau} = \mathbf{t}^T \mathbf{K} \mathbf{t}, \quad (65)$$

where \mathbf{t} is a column vector that stores information about $\boldsymbol{\tau}$ and \mathbf{K} is a symmetric positive definite matrix that encodes information of \mathbb{K} . There are multiple ways to obtain such vector–matrix quantities that preserve the left-hand side in Eq. (65), e.g., Kelvin and Voigt notations. Mehrabadi and Cowin [90] and Itskov [91] have shown that \mathbb{K} and \mathbf{K} share common spectral characteristics such as eigenvalues and eigenbases in the case of Kelvin and Voigt representations.

Fluid distance function Eq. (21) can be represented in a vector–matrix format without any further action, since \mathbf{C}_f and \mathbf{S}_f are 2nd order tensors which are also matrix. Hence, they have already followed the format in the right-hand side of Eq. (65). In the view of vector–matrix notation both solid and fluid distances have the same format but with different sizes, e.g., in 3D set-up \mathbf{t} has 6 components for the solid distance and 3 components for the fluid part. For the rest of the discussion in this section, we develop the framework for a general, unified metric as follows:

$$d(\mathbf{t}, \mathbf{r}) = \mathbf{t}^T \mathbf{K} \mathbf{t} + \mathbf{r}^T \mathbf{M} \mathbf{r}, \quad (66)$$

where \mathbf{t} and \mathbf{r} are column vectors with size n , and \mathbf{K} and \mathbf{M} are constant symmetric positive definite matrices. For solid distance function, we incorporate Voigt notation to store elements of tensors $\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^*$ and $\boldsymbol{\sigma}' - \boldsymbol{\sigma}'^*$ into vectors \mathbf{t} and \mathbf{r} , respectively, and build matrices \mathbf{K} and \mathbf{M} equivalent to tensors \mathbb{C}_s and \mathbb{S}_s , respectively. For fluid distance function, vectors \mathbf{t} and \mathbf{r} are equal to vectors $\nabla^x p - \nabla^x p^*$ and $\mathbf{q} - \mathbf{q}^*$, respectively, and matrices \mathbf{K} and \mathbf{M} are exactly equal to tensors \mathbf{C}_f and \mathbf{S}_f . Since any symmetric positive definite matrix \mathbf{K} possess decomposition $\mathbf{K} = \mathbf{C}^T \mathbf{C}$ (e.g., via Cholesky factorization and matrix square root) we can write:

$$d(\mathbf{t}, \mathbf{r}) = \mathbf{t}^T \mathbf{K} \mathbf{t} + \mathbf{r}^T \mathbf{M} \mathbf{r} = \mathbf{t}^T \mathbf{C}^T \mathbf{C} \mathbf{t} + \mathbf{r}^T \mathbf{S}^T \mathbf{S} \mathbf{r} = \tilde{\mathbf{t}}^T \tilde{\mathbf{t}} + \tilde{\mathbf{r}}^T \tilde{\mathbf{r}}, \quad (67)$$

where $\tilde{\mathbf{t}} = \mathbf{C} \mathbf{t}$ and $\tilde{\mathbf{r}} = \mathbf{S} \mathbf{r}$ are projected vectors of \mathbf{t} and \mathbf{r} , also $\mathbf{M} = \mathbf{S}^T \mathbf{S}$. The distance function can be further reduced to a more compact form by concatenation of vectors $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{r}}$ into $\tilde{\mathbf{w}} = \begin{bmatrix} \tilde{\mathbf{t}} \\ \tilde{\mathbf{r}} \end{bmatrix} \in \mathbb{R}^{2n}$ as follows:

$$d(\mathbf{t}, \mathbf{r}) = \tilde{\mathbf{t}}^T \tilde{\mathbf{t}} + \tilde{\mathbf{r}}^T \tilde{\mathbf{r}} = \tilde{\mathbf{w}}^T \tilde{\mathbf{w}}. \quad (68)$$

The linear projection via the matrices \mathbf{C} and \mathbf{S} for the corresponding variables is introduced for convenience. This simple treatment enables us to adopt open-source libraries developed by the NNS community for Euclidean space (and in a more general sense Minkowski space) with minimal implementation efforts.

Algorithm 1 Off-line kd-tree construction

- 1: **Input:** Database of $\{(\boldsymbol{\epsilon}_i^*, \boldsymbol{\sigma}'_i^*)\}_{i=1}^N$ strain-effective stress tensor pairs (2nd order symmetric tensors) in $m = 1, 2, 3$ dimensions, numerical parameters \mathbb{C}_s and \mathbb{S}_s (4th order super symmetric positive definite tensors)
 - 2: **Output:** Tree object \mathcal{T}^s , $\mathbf{C} \in \mathbb{R}^{n \times n}$, $\mathbf{S} \in \mathbb{R}^{n \times n}$, $\mathbf{K}^{n \times n}$, and $\mathbf{M}^{n \times n}$
 - 3: $\mathbf{K} \leftarrow \text{Voigt}(\mathbb{C}_s)$, $\mathbf{M} \leftarrow \text{Voigt}(\mathbb{S}_s)$ ▷ Apply Voigt notation (or Kelvin notation)
 - 4: $\mathbf{C} \leftarrow \text{Fact}(\mathbf{K})$, $\mathbf{S} \leftarrow \text{Fact}(\mathbf{M})$ ▷ Matrix factorization, e.g. Cholesky or square root
 - 5: $\mathbf{D} \leftarrow \mathbf{0}$ ▷ Zeros matrix of size $N \times n$ where $n = m(m+1)/2$
 - 6: **for** $i = 1 : N$ **do** ▷ Loop over N data points (pairs of strain–stress)
 - 7: $\mathbf{t} \leftarrow \text{Voigt}(\boldsymbol{\epsilon}_i^*)$, $\mathbf{r} \leftarrow \text{Voigt}(\boldsymbol{\sigma}'_i^*)$
 - 8: $\mathbf{D}_{i,:} \leftarrow [(\mathbf{C} \mathbf{t})^T \mid (\mathbf{S} \mathbf{r})^T]$
 - 9: $\mathcal{T}^s \leftarrow \text{KDTree}(\mathbf{D})$ ▷ Apply the kd-tree algorithm `scipy.spatial.cKDTree`
-

We provide Pseudo-codes 1 and 2 to summarize this section. In Pseudo-code 1, we first project database and then construct the kd-tree object for the projected database by utilizing the functionality of `scipy.spatial.cKDTree`. This task is executed once before the simulation. We present this pseudo-code for brevity only for solid database, but the same procedure should be applied for the fluid database. The resultant tree object of this pseudo-code stores projected data points in an efficient way for fast nearest neighbor search during the simulation. Recall that the global minimization step does not have any projected quantities, so the input for the nearest neighbor search is in the real

Algorithm 2 Local minimization: nearest data point calculation

-
- 1: **Input:** Physical strain-effective stress pair (ϵ, σ') at a quadrature point, k-d tree object of solid database \mathcal{T}^s
 - 2: **Output:** Closest material strain-effective stress pair $(\epsilon_i^*, \sigma_i'^*)$ from database, distance function d_i
 - 3: $\mathbf{t} \leftarrow \text{Voigt}(\epsilon)$, $\mathbf{r} \leftarrow \text{Voigt}(\sigma')$
 - 4: $\bar{\mathbf{w}}^T \leftarrow [(\mathbf{C}\mathbf{t})^T | (\mathbf{S}\mathbf{r})^T]$ ▷ \mathbf{C} and \mathbf{S} are stored once before the on-line simulation
 - 5: $d_i, \bar{\mathbf{w}}_i^* \leftarrow \text{NNS}(\bar{\mathbf{w}}, \mathcal{T}^s)$ ▷ A new query $\bar{\mathbf{w}}$ for k-d tree nearest neighbor search
 - 6: $\bar{\mathbf{t}}_i^* \leftarrow \bar{\mathbf{w}}_i^*|_{1:n}$, $\bar{\mathbf{r}}_i^* \leftarrow \bar{\mathbf{w}}_i^*|_{n+1:2n}$ ▷ Separate strain and stress related components
 - 7: $\mathbf{t}_i^* \leftarrow \mathbf{K}^{-1}\mathbf{C}^T\bar{\mathbf{t}}_i^*$, $\mathbf{r}_i^* \leftarrow \mathbf{M}^{-1}\mathbf{S}^T\bar{\mathbf{r}}_i^*$ ▷ inverse projection onto real configuration
 - 8: $\epsilon_i^* \leftarrow \text{Voigt}^{-1}(\mathbf{t}_i^*)$, $\sigma_i'^* \leftarrow \text{Voigt}^{-1}(\mathbf{r}_i^*)$ ▷ From vector representation to tensor
-

configuration of databases. In Pseudo-code 2, we first project the new query point, and then we perform the nearest neighbor search. Finally, the nearest point should be projected back to the real configuration; otherwise, it cannot be used in the global optimization step. Instead of projecting back, the real database could be stored as well. In this case, the real data point could be tracked by the data index without any extra matrix multiplication due to the inverse projection. This latter approach could be faster if the original database is stored in a hash table, but it may also exhibit memory deficiency. Notice that the local minimization step is the execution of Pseudo-code 2 at each fixed-point iteration over all quadrature points separately.

Remark 6. Symmetric square matrix $\mathbf{K}_{n \times n}$, with factorization $\mathbf{K} = \mathbf{C}^T \mathbf{C}$ where $\mathbf{C}_{m \times n}$, is positive definite if and only if \mathbf{C} has full column rank; see Theorem 7.2.7 in [92].

Remark 7. Cholesky factorization and square root matrix are unique for symmetric positive definite matrix; see Corollary 7.2.9 and Theorem 7.2.6 in [92].

Remark 8. Note that if \mathbf{C} satisfies $\mathbf{K} = \mathbf{C}^T \mathbf{C}$ then for any unitary matrix (rotation) \mathbf{Q} the new matrix \mathbf{QC} also factorizes $\mathbf{K} = (\mathbf{QC})^T \mathbf{QC}$. Therefore, symmetric positive definite matrix \mathbf{K} can be factorized by infinite matrices unless \mathbf{K} is identity matrix.

Remark 9. Note that the Cholesky factorized matrix \mathbf{L} and its inverse are lower triangular matrices which can be optimized to reduce operations approximately by half in case of matrix multiplication or summation in comparing with operations involving square root matrix $\mathbf{K}^{1/2}$ which is a full matrix. Also, this reduction of basic operations reduces the chance of over floating. In this work, we do not compare computational performance of available decomposition schemes for symmetric positive definite matrices, and we use square root factorization.

4. Algorithm and numerical implementation

This section intends to bundle all the discussed ingredients for solving poroelasticity with/out constitutive laws. For brevity, the overall algorithm is presented for the fully data-driven framework, but it is kept general and can be used for hybrid formulations.

Algorithm 3 provides general steps used in the fully data-driven formulation. In this research, we first solve global minimization, so the first fixed-point iteration of the first time step needs an initialization for the distribution of material states at each quadrature point (see line 5 of Algorithm 3). This initialization can be performed by a random assignment of data points from the database. The initial assignment affects the fixed-point method convergence, and different assignments may lead to slightly different final solution [30]. For nonlinear cases, hybrid formulation 1 with nonlinear model-based constitutive law, such a random initialization may result in non-convergence of the Newton–Raphson algorithm. Hence, in nonlinear cases, a homogeneous assignment with zero values (or residual states) is more meaningful. Notice that there is no need to do more than one Newton–Raphson iteration for the fully data-driven formulation regardless of any hidden non-linearity in the material database since this formulation is always linear, as discussed earlier. Although the material states (line 5 of Algorithm 3) can be re-initialized for other time steps we use values from the previous time step.

Algorithm 3 Fully data-driven poroelasticity

```

1: Input: solid database  $\{(\epsilon_k^*, \sigma_k'^*)\}_{k=1}^N$ , fluid database  $\{(\nabla^x p_l^*, q_l^*)\}_{l=1}^M$ , numerical parameters  $\mathbb{C}_s$ ,  $\mathbb{S}_s$ ,  $\mathbf{C}_f$ , and  $\mathbf{S}_f$ 
2: construct k-d tree object  $\mathcal{T}^s$  for solid database via Algorithm 1
3: construct k-d tree object  $\mathcal{T}^f$  for fluid database via a fluid version of Algorithm 1
4: for  $i = 1 : N_{ts}$  do ▷ loop over  $N_{ts}$  time steps
5:   initialize  $(\epsilon^*, \sigma'^*)$  and  $(\nabla^x p^*, q^*)$  at each quadrature point from database. ▷ required for the first time step
    $i = 1$  and it can be a random assignment.
6:   while Fixed-Point not converged do
7:     initialize degree of freedoms corresponding to fields  $u_i, \beta_i^{\text{mom}}, p_i, \beta_i^{\text{mass}}$ .
8:     while Newton–Raphson not converged do ▷ global minimization
9:       solve linearized Eqs. (29), (30), (31), and (32).
10:      update degree of freedoms corresponding to fields  $u_i, \beta_i^{\text{mom}}, p_i, \beta_i^{\text{mass}}$ .
11:      for  $j = 1 : N_{qp}$  do ▷ loop over  $N_{qp}$  quadrature points
12:        update physical stress by Eq. (35).
13:        local minimization for solid part via Algorithm 2.
14:        update physical Darcy’s velocity by Eq. (36).
15:        local minimization for fluid part via a fluid version of Algorithm 2.
16:      record solution for next time marching.

```

5. Numerical examples

Now we examine different aspects of the proposed data-driven formulations through six examples. Before presenting the actual poroelasticity problems, in [Appendix A](#), we show how effective is the k-d tree search for even a very simple problem. Then we will verify data-driven poroelasticity formulations by solving two well-known problems in the literature of geomechanics and biomechanics with available analytical solutions. Convergence with respect to the amount of data and fixed-point iterations will be studied. Three other problems will be investigated to demonstrate the capability and robustness of formulations in more complex circumstances such as non-linearity of material behavior. In the last two examples, we will discuss the fidelity and availability of data and the effectiveness of hybrid data-driven formulations. In the last example, we showcase the application of the hybrid data-driven formulation in dealing with an extreme condition where fluid flow database possesses a highly oscillatory porosity–permeability behavior. At the same time, the solid skeleton deformation follows a nonlinear porosity and pressure-dependent behavior. In all of the poroelastic examples, we utilize the k-d tree search method unless otherwise is noted. These examples are solved with our in-house data-driven solver developed in the Python programming language. The k-d tree data structure is adopted from `scipy.spatial.cKDTree` class of the open-source package SciPy [84]. Linear systems are solved by the LU factorization scheme implemented in the class `scipy.linalg.lu` of the SciPy package. We perform this factorization once for linear cases when material behavior is linear or the fully data-driven scheme is used. Numerical experiments were run on a computer with a Quad-Core Intel Core i5 processor running at 1.4 GHz using 16 GB of RAM.

5.1. Verification exercise 1: Terzaghi’s consolidation problem

In this problem, we verify the three proposed formulations in a two-dimensional computational set-up and compare the accuracy and convergence of the three formulations.

Terzaghi’s consolidation problem has been previously used to verify Finite Element codes for modeling the consolidation process of two-phase porous media [11,60,61,93–99]. The analytical solution is derived for the one-dimensional problem. However, we solve the problem in a 2D domain (see [Fig. 5](#)) to validate the formulation and implementation for two-dimensional elements.

According to [Fig. 5](#), symmetry boundary conditions are applied laterally. Traction loads $\bar{\mathbf{t}} = (0, -0.9)$ GPa are suddenly applied (at $t = 0$ s) over the drained boundary and remained constant during the simulation. The bottom boundary is clamped. Initial conditions are at rest, i.e., $p_0 = 0$ and $u_0 = 0$. We use 1×20 quadrilateral elements

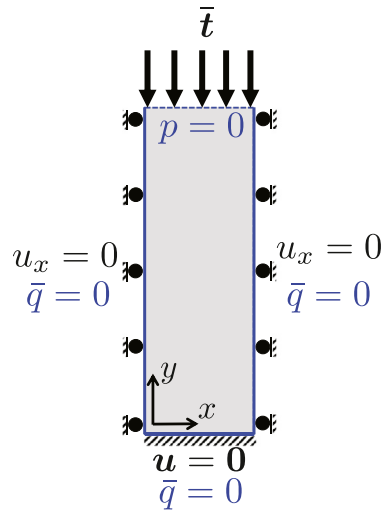


Fig. 5. Terzaghi consolidation problem.

Table 1

Material parameters for Terzaghi problem.

Physical parameter	Unit	Value
Young's modulus (E)	GPa	70
Poisson's ratio (ν)	–	0
Plane condition	–	Plane strain
Intrinsic permeability (k)	m^2	3.0612×10^{-9}
Fluid dynamics viscosity (μ)	Pa s	0.001
Biot coefficient (B)	–	1
Biot modulus (M)	GPa	266.667

to mesh a domain of $0.1 \text{ m} \times 1 \text{ m}$ (width \times height). Numerical integration is performed by the 4-point Gaussian quadrature rule. Time increment is set to $\Delta t = 0.1 \text{ s}$, and simulations end at $t_{\text{end}} = 10 \text{ s}$. We will use material parameters listed in Table 1 to generate artificial data needed for each data-driven scheme.

5.1.1. Hybrid formulation 1: data-driven fluid phase

Hooke's law in plane strain is used as the constitutive law for the hybrid formulation, used in the first term of Eq. (43), but the flow part relies on data. To compare data-driven solution with model-based approach, we generate a database with N equidistant points sampled from linear Darcy's law by setting $\partial p / \partial x = 0$ and $-8.6 \text{ GPa/m} \leq \partial p / \partial y \leq 4.3 \text{ GPa/m}$. In this way, we expect to recover the model-based solution within the limit of infinite data points. According to the problem's one-dimensional nature, pressure gradient and fluid velocity are zero in the x direction. Hence, we can generate a database without including any variations in the x direction, the same idea used in [30]. This simplification helps to generate smaller databases for this particular problem. In one case, we will show that even if we include some variations of $\partial p / \partial x$ into the database, the solver will find the closest data points with $\partial p / \partial x \rightarrow 0$ as final solutions. The prior knowledge about model-based FEM solution (i.e., conventional u-p formulation with known constitutive laws) indicates $-4.3 \text{ GPa/m} \leq \frac{\partial p}{\partial y} |^{\text{FEM}} \leq 0$. Hence, we have sufficient out of solution candidates in the database to testify the accuracy of data-driven solvers. Also, to make the situation difficult for the solver, fixed-point iteration starts with initial homogeneous assignment $\partial p / \partial y = 4.3 \text{ GPa/m}$ at every quadrature point which is completely out of the solution space. The same configuration will be used for the fluid part of the fully data-driven formulation, as well.

The effect of numerical parameters in metric functions, see Eqs. (19), (20), and (21), on the overall performance of data-driven scheme is not negligible and needs a separate study to find a systematic way of fine-tuning of these values [33]. Based on a suggestion in [32], we assume optimal values for C_s and C_f are equal to elasticity and

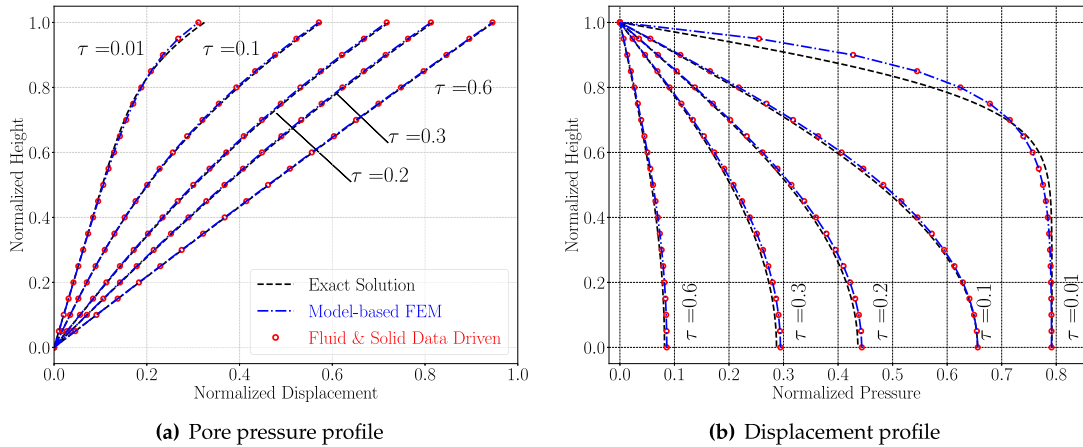


Fig. 6. A comparison between results of the fully data-driven formulation, conventional model-based formulation, and exact solution in (a) normalized displacement and (b) normalized pore pressure. Normalized time is $\tau = t/t_{\text{end}}$.

permeability tensors, respectively. Although this is not a fair decision, since it means we have already known the underlying structure in data, it helps us compare the performance of different data-driven formulations by freezing artifacts of these numerical parameters, assuming their numerical parameters are optimal.

5.1.2. Hybrid formulation 2: data-driven solid phase

In this case, the Darcy's law serves as the constitutive law in the hybrid formulation 2 (see Section 2.4). Due to the one-dimensional characteristic, we sample N equidistant points for $-0.026 \leq \epsilon_{yy} \leq 0.013$ and set $\epsilon_{xx} = \epsilon_{xy} = 0$ in the database. Note that the prior knowledge shows $-0.013 \leq \epsilon_{yy}^{\text{FEM}} \leq -0.0027$ base on the model-based FEM solution. To testify the robustness of the formulation, we force the fixed-point iteration starts with an initially homogeneous data assignment $\epsilon_{yy} = 0.013$ which is completely out of the solution space, instead of random assignment.

5.1.3. Fully data-driven formulation: data-driven fluid and solid phases

We employ the formulation described in Section 2.2 which requires two separate data sets for solid and fluid parts. These data sets are generated as described above within the same ranges and conditions for pressure gradient and strain tensor. Note that, in the fully data-driven scheme, when we say N data points, it means we have two separate data sets with N points in each for solid and fluid databases. To compare convergence, the initial assignment for the fixed-point iteration in the fully data-driven scheme is the same as the hybrid simulations.

5.1.4. Comparisons of numerical results

Fig. 6 compares profiles of pressure and displacement along the sample height at various time steps. Two data sets each with $N = 16385$ data points is used for the fully data-driven simulation shown by red circles in Fig. 6. There is a good agreement between results obtained by the fully data-driven formulation and exact solutions; see Appendix B for the analytical expression. Moreover, the error between fully data-driven and conventional model-based FEM results is almost negligible. We observed similar trends for hybrid formulations as well which are not included in this paper for the sake of brevity.

Figs. 7–10 compare the predicted time histories of pore pressure, fluid velocity, displacement, and effective stress predicted by the fully data-driven and the two hybrid schemes at $y = 0.91$ m. In each case, the simulations are conducted with a database of different numbers of data points ($N = 65, 1025$ and 16385) to examine how the size of the database affects the accuracy and robustness of the predictions. For Terzaghi's problem, the three simulations do not exhibit a significant difference in accuracy in the last two cases where $N = 1025$ and 16385 , respectively. Nevertheless, the three schemes do exhibit slightly different transient responses when the database is small at $N = 65$. In Figs. 7–8, all three schemes exhibit some level of oscillations in time. These oscillations has also been observed in other data-driven problems as such as Carrara et al. [100] and Leygue et al. [33] where the discrete

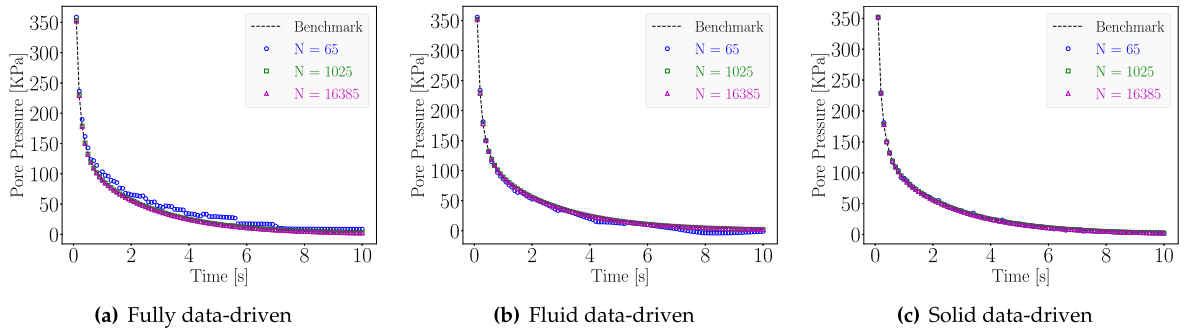


Fig. 7. Pore pressure time-history response for the quadrature point located at $y = 0.91$ m. (a) fully data-driven, (b) fluid data-driven, and (c) solid data-driven solutions are compared with the benchmark solution (model-based). N shows the size of fluid and solid databases, e.g., $N = 65$ means each of the solid and fluid databases has 65 samples.

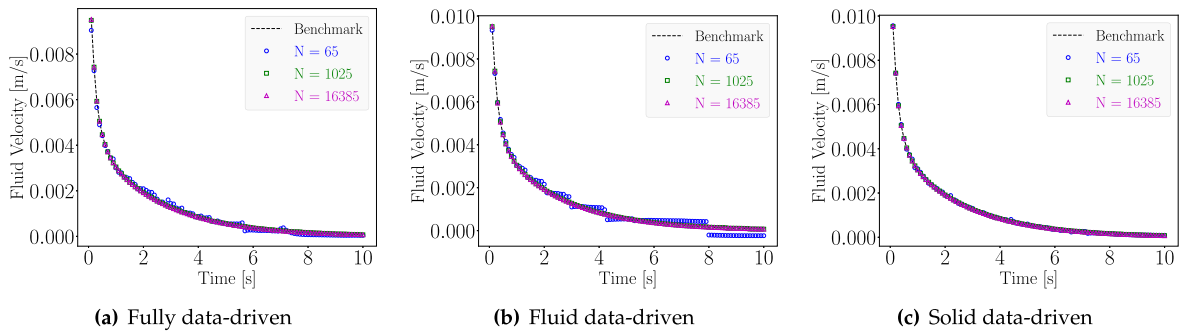


Fig. 8. Fluid velocity time-history response for the quadrature point located at $y = 0.91$ m. (a) fully data-driven, (b) fluid data-driven, and (c) solid data-driven solutions are compared with the benchmark solution (model-based). N shows the size of fluid and solid databases, e.g., $N = 65$ means each of the solid and fluid databases has 65 samples.

nature of the local minimization problem may lead to a sudden jump in the predicted responses when a new data point is selected. This issue is most obvious in Fig. 8(b) where the case with $N = 65$ shows the fluid velocity exhibits a stepwise decrease as the porous medium approaching the drained state. This spurious stepwise pattern in fluid velocity is not observed in Fig. 8(c) where the hydraulic constitutive responses is model-based. In the fully data-driven case shown in Fig. 8(a), the fluid velocity does exhibit spurious oscillation when the database is small ($N = 65$), but the pattern is not stepwise. This may be due to the fact that the choices of the admissible effective stress–strain and fluid velocity–pressure gradient pairs in the local minimization may affect each other.

Due to the hydromechanical coupling, this stepwise pattern also influences the accuracy of the displacement and stress predictions for the fluid-driven case where the sudden increase of the fluid velocity at time $t = 8$ s may lead to significant error in displacement and effective stress in the fluid data-driven scheme as depicted in Figs. 9–10. For Terzaghi's problem with limited data, the solid data-driven scheme seems to have the best results, and the fluid data-driven scheme performs poorly. This is not surprising, as Terzaghi's consolidation begins with a pore pressure jump at the top and finishes with the excess pore pressure vanished everywhere. To replicate this pattern, the fluid database must include data points with a wide range of pressure gradient (from infinite to zero), whereas the upper and lower bounds of the magnitude of the effective stress (of the exact solution) are equal to the applied traction and zero, respectively.

Table 2 compares the computational time of different data-driven formulations solved with brute-force and kd-tree algorithms. The computational time for the fully data-driven scheme is higher than the hybrid alternatives. This is expected as it requires two local minimization steps (one for each data set) instead of one. The fluid data-driven formulation is found to be slightly more efficient (about 10% difference in CPU time) than the solid counterpart in both the brute-force and kd-tree cases. This latter trend is nevertheless sensitive to the data availability, noises, and the patterns of the solutions (as demonstrated in another numerical test summarized in Table 5). Note that

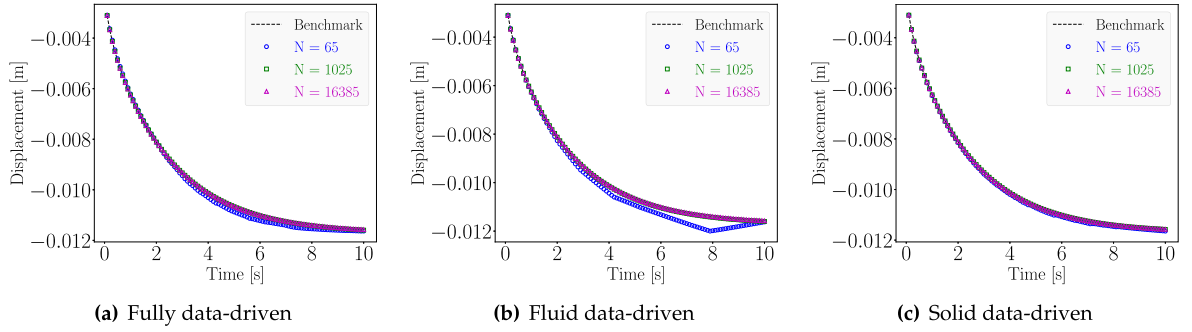


Fig. 9. Displacement time-history response for the quadrature point located at $y = 0.91$ m. (a) fully data-driven, (b) fluid data-driven, and (c) solid data-driven solutions are compared with the benchmark solution (model-based). N shows the size of fluid and solid databases, e.g., $N = 65$ means each of the solid and fluid databases has 65 samples.

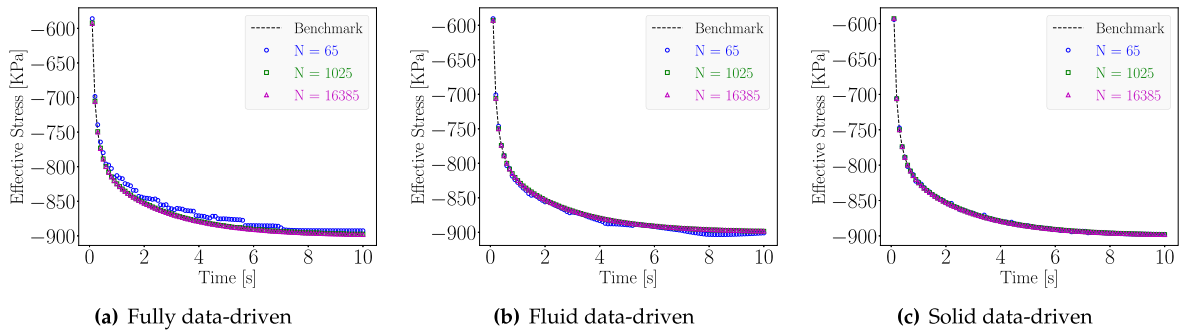


Fig. 10. Effective stress time-history response for the quadrature point located at $y = 0.91$ m. (a) fully data-driven, (b) fluid data-driven, and (c) solid data-driven solutions are compared with the benchmark solution (model-based). N shows the size of fluid and solid databases, e.g., $N = 65$ means each of the solid and fluid databases has 65 samples.

Table 2

A comparison of computational time for different schemes. For each scheme, we run the same simulation setup five times. The average and the standard deviation of CPU time are shown in the last two columns.

Method	Number of data points	CPU time [s] (brute-force)	CPU time [s] (kd-tree)
Hybrid 1: fluid data-driven	65	4.19 ± 0.03	4.86 ± 0.09
	257	5.43 ± 0.03	5.98 ± 0.04
	1025	8.65 ± 0.08	7.93 ± 0.07
	4097	19.24 ± 0.07	10.01 ± 0.1
	16385	73.04 ± 0.61	13.1 ± 0.18
Hybrid 2: solid data-driven	65	4.73 ± 0.04	5.43 ± 0.05
	257	6.57 ± 0.11	7.22 ± 0.05
	1025	9.95 ± 0.08	8.93 ± 0.04
	4097	22.5 ± 0.25	11.52 ± 0.06
	16385	80.38 ± 0.67	14.58 ± 0.22
Fully data-driven	65	6.22 ± 0.18	7.38 ± 0.07
	257	7.60 ± 0.11	9.30 ± 0.12
	1025	12.45 ± 0.10	11.20 ± 0.32
	4097	31.33 ± 0.16	13.68 ± 0.34
	16385	122.30 ± 0.94	16.37 ± 0.18

the difference among different data-driven schemes becomes much less significant once the kd-tree is employed. This result indicates that the kd-tree is capable of improving the efficiency of the local minimization steps of the data-driven simulations and is consistent with the benchmark study presented in [Appendix A](#).

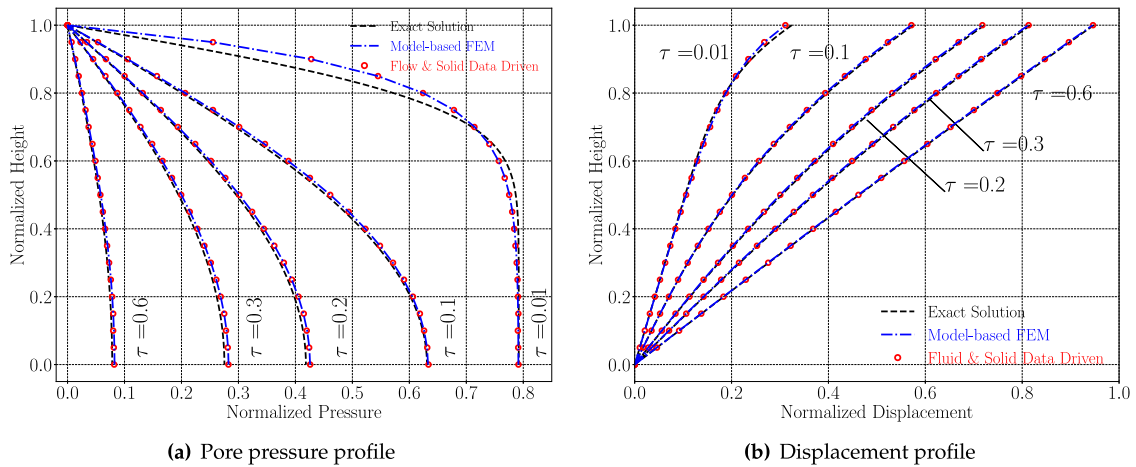


Fig. 11. A comparison between results of the fully data-driven formulation, conventional model-based formulation, and exact solution with a multi-dimensional data set. Normalized time is $\tau = t/t_{\text{end}}$ (a) pore pressure profile; (b) displacement profile.

To make sure the accuracy is not limited to specialized data sets, we showcase the pressure and displacement profiles for the fully data-driven formulation with more general data sets in Fig. 11. In this figure, fluid data set is constructed by uniform discretization of data axis $\partial p/\partial x$ with 65 points in range $[-0.4, 0.4]$ GPa and axis $\partial p/\partial y$ with 4096 points in range $[-8.6, 4.3]$ GPa. In solid data set ϵ_{xx} , ϵ_{yy} , and ϵ_{xy} are uniformly sampled by 65, 4096, and 65 points within intervals $[-0.002, 0.002]$, $[-0.026, 0.013]$, and $[-0.001, 0.001]$, respectively. In the initial fixed-point iteration, data points are randomly assigned (as opposed to the uniform assignment in previous cases). Moreover, numerical parameters for fluid and solid metrics are set to $2.45 \times 10^{-6} \text{ I m}^2/\text{Pa s}$ and plane strain elasticity tensor with Poisson's ratio 0.1 and Young's modulus 56 GPa respectively, as opposed to previous cases where they were chosen to their potential optimal values. The reason we changed the previous set-ups is to show the robustness of formulation in more general settings. As it is clear from this figure, the results are in a good agreement with the ground truths, similar to previous cases.

Remark 10. The setup of the global minimization problem is a complex multi-field problem that requires a proper combination of discretization of solid displacement, fluid pressure, effective stress, Darcy's velocity, and the corresponding Lagrangian multipliers to achieve stability and convergence. In Terzaghi's consolidation problem, pressure oscillation has been observed when the simulation is near the undrained limit with a small time step employed. This oscillation is more severe when the displacement and pore pressure fields are interpolated by the same basis function (cf. [96,101]). To examine this issue, we have conducted a preliminary study in which a pressure projection term is added in Eq. (30) to eliminate the pore pressure oscillation in equal-order finite element (based on the bounds for the real-valued growth/decay rate, cf. [102]). The numerical experiments indicate that this stabilized procedure is capable of eliminating the pore pressure oscillations for the data-driven poromechanics problem (see Fig. 12 which compares the numerical solutions taken near the undrained limit with normalized time = 0.00005 without (a) and with (b) the stabilization term. Nevertheless, a systematic study that explores the feasibility of deriving a stabilization procedure or identifying the compatible finite-dimensional space for the global minimization is required to further analyze the effects of the approximation spaces on the stability and convergence of all solution fields. Such a study is currently ongoing but out of the scope of this study.

5.2. Verification exercise 2: Stress relaxation

In this problem, we validate formulations and implementation for three-dimensional elements. Also, we will study the convergence behavior of the metric minimization part of the algorithm.

Fig. 13 depicts geometry and mesh used herein. All external faces of the cylinder are impermeable except the top face. The only free displacement degrees of freedom for boundary faces are in the z direction except the bottom

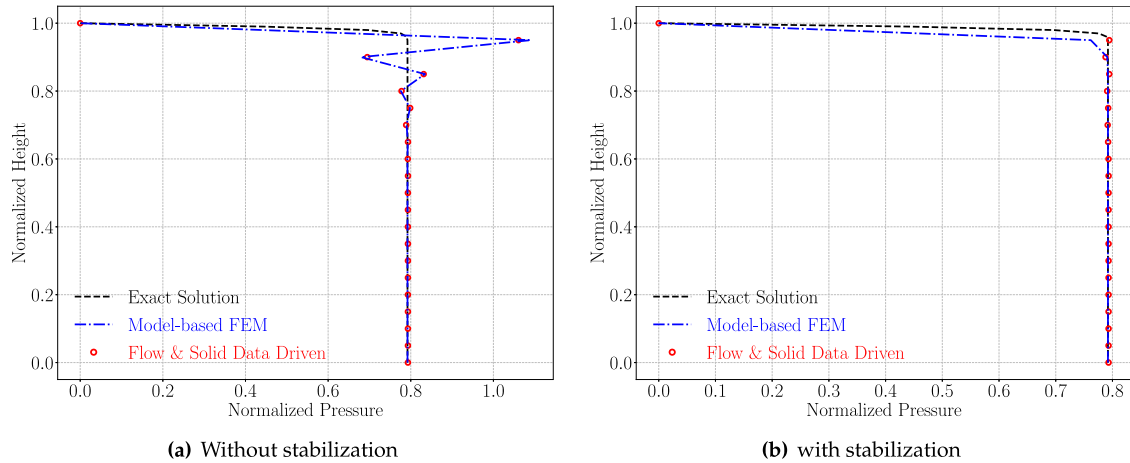


Fig. 12. Numerical solutions of Terzaghi's problem obtained from the standard equal-order finite element and fully data-driven model-free simulations compared against the exact solution. (a) equal-order discretization without any stabilization term; (b) equal-order discretization with pressure-project stabilization term for the pore pressure field.

Table 3

Material parameters for stress relaxation problem.

Physical parameter	Unit	Value
Young's modulus (E)	GPa	100
Poisson's ratio (ν)	–	0.25
Intrinsic permeability (k)	m^2	8.33×10^{-8}
Fluid dynamics viscosity (μ)	Pa s	0.001
Biot coefficient (B)	–	1
Biot modulus (M)	GPa	2×10^{10}

boundary which is completely clamped from movement. Pressure is kept zero at the top boundary, and a compressive displacement with constant rate $\dot{u}_z = -0.005 \text{ m/s}$ is applied for $t_{\text{ramp}} = 2 \text{ s}$ and remained constant towards the simulation end $t_{\text{end}} = 10 \text{ s}$. This problem is known as stress relaxation test which is another benchmark for validation of numerical codes [38,103]. Material properties are listed in Table 3 which are used for the data generation or conventional model-based FEM. The system is in rest at $t = 0$, and time increment is set to $\Delta t = 0.1 \text{ s}$. We use trilinear Lagrangian basis functions to discretize all unknown fields. Numerical integration is performed by the 8-points Gaussian quadrature rule.

We generate fluid data set using equidistant points sampled from Darcy's law with $\partial p / \partial x = \partial p / \partial y = 0$ and $-58 \text{ MPa} \leq \partial p / \partial z \leq 0.0005 \text{ MPa}$. The only nonzero component of strain in solid data set is $-0.0024 \leq \epsilon_{zz} \leq 0$, and data is sampled from Hooke's law. Again, we have used our prior knowledge about the problem's one-dimensional characteristic to reduce the amount of data needed to complete the simulation. For the following results, 1000 data points are used in fluid data-driven and solid data-driven simulations. The fully data-driven simulation has two separate databases with 1000 data points in each for solid and fluid parts. Numerical parameters for fluid and solid metric functions are set to $6.66 \times 10^{-5} \text{ I m}^2/\text{Pa s}$ and elasticity tensor with elastic modulus 80 GPa and Poisson's ratio 0.225, respectively.

As illustrated in Fig. 14, traction history at the top nodal point, colored with red star in Fig. 13, is in good agreement with the analytical solution (see Appendix C) for all three data-driven formulations. Additionally, Fig. 15 confirms a satisfactory match for the lateral effective stress between model-based response and data-driven formulations. Although there is no constitutive relation in solid and fully data-driven schemes to impose the Poisson effect this phenomenon is captured correctly even with a small database consists of 1000 data points. Note that this observation reveals the capacity of the proposed framework in transferring hidden information from data to modeling without human intervention. Fig. 16 compares pressure predictions at different times along the height of cylinder which indicates solution accuracy for the continuity equation.

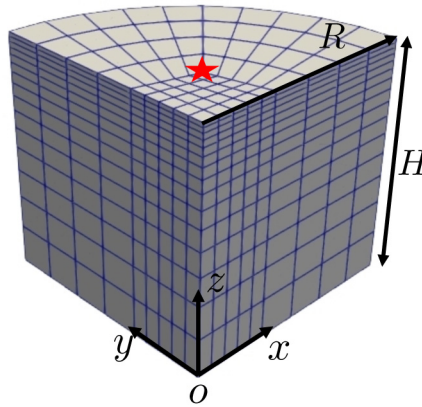


Fig. 13. Geometry and mesh for stress relaxation problem with $R = H = 10$ m. Red star point with coordinates (7.89, 7.89, 10)m shows an observation location for plotting purposes.

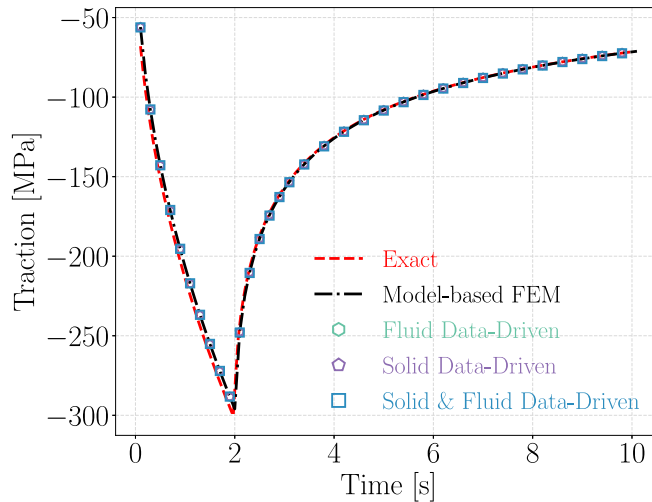


Fig. 14. Normal traction history for different data-driven schemes at the observation point shown in Fig. 13.

Initial material assignments for the fixed-point iteration algorithm is random at the first time step. Within each time step, fixed-point iterations continue until the total number of material projections becomes zero. Fig. 17 depicts the number of fixed-point iterations at each time step for all data-driven schemes. This figure shows that data-driven formulations behave almost similar, in terms of the number of fixed-point iterations. Total values for metric functions (spatial integration of distance functions over the whole domain) at each fixed-point iteration for entire simulation are plotted in Fig. 18. Each of those peaks corresponds to the first iteration of a new time step. Within each time step, we observe a decay behavior in metric function which confirms the convergence performance of fixed-point iteration, see zoom snapshots in Fig. 18(a) and (b).

5.3. Two-dimensional footing

This problem demonstrates the capability of formulation in capturing rather a complex response field with singular characteristics. The boundary condition at the footing edge (W_l, H), see Fig. 19, induces a singular response in stress and consequently fluid velocity fields.

Fig. 19 depicts boundary conditions, geometry, and mesh used herein. Normal traction over the footing area is applied with the rate -50 MPa/s. We run the simulation for 20 time steps with time increment $\Delta t = 1$ s. Material

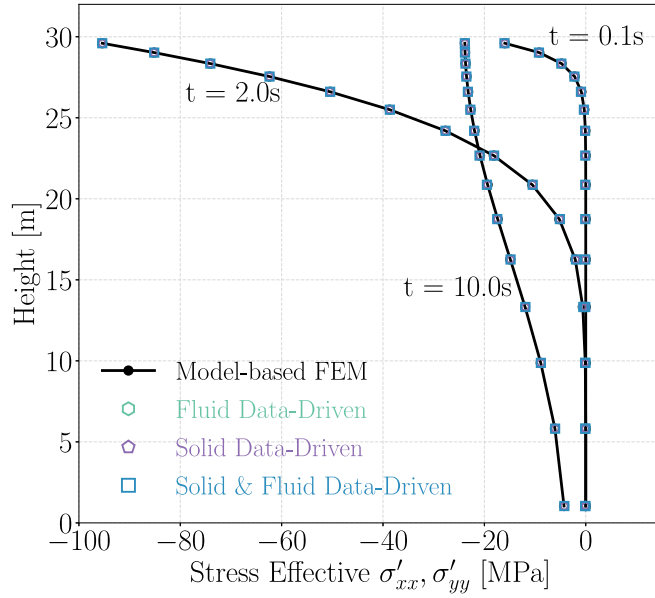


Fig. 15. Lateral effective stress profile along the sample height at different times.

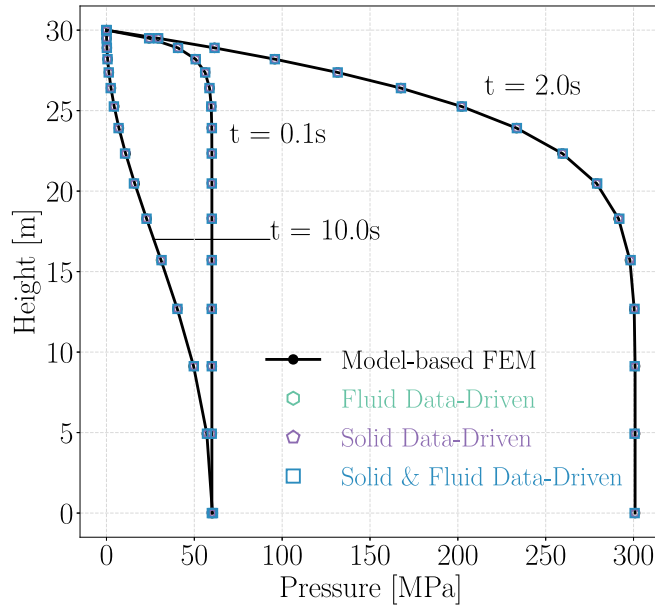


Fig. 16. Pressure profile along the sample height at different times.

properties listed in Table 4 are utilized for a synthetic data generation or model-based simulation. We set numerical parameters of metric functions equal to $8 \times 10^{-8} \text{ I m}^2/\text{Pa s}$ and elasticity tensor calculated for Young's modulus 24 GPa and Poisson's ratio 0.3. The initial data-assignment for the first fixed-point iteration is random. Bilinear Lagrangian basis functions are used for all unknown fields, and the 4-point Gaussian quadrature rule calculates numerical integration.

Fluid constitutive law follows Darcy's law, and synthetic data is generated for $-864 \text{ MPa} \leq \partial p / \partial x \leq 984 \text{ MPa}$ and $-3740 \text{ MPa} \leq \partial p / \partial y \leq 288 \text{ MPa}$. To generate solid database, Hooke's law is assumed for the effective stress-strain relations, and solid database is sampled within strain ranges $-5.04 \times 10^{-3} \leq \epsilon_{xx} \leq 14.3 \times 10^{-3}$,

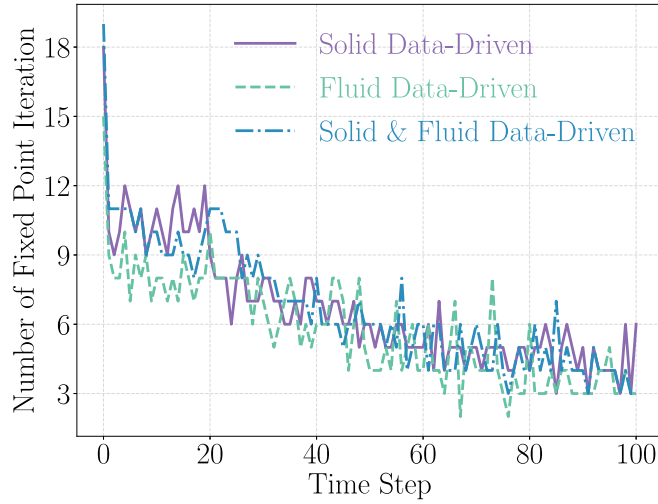


Fig. 17. Number of fixed-point iterations at each time step for all data-driven formulations.

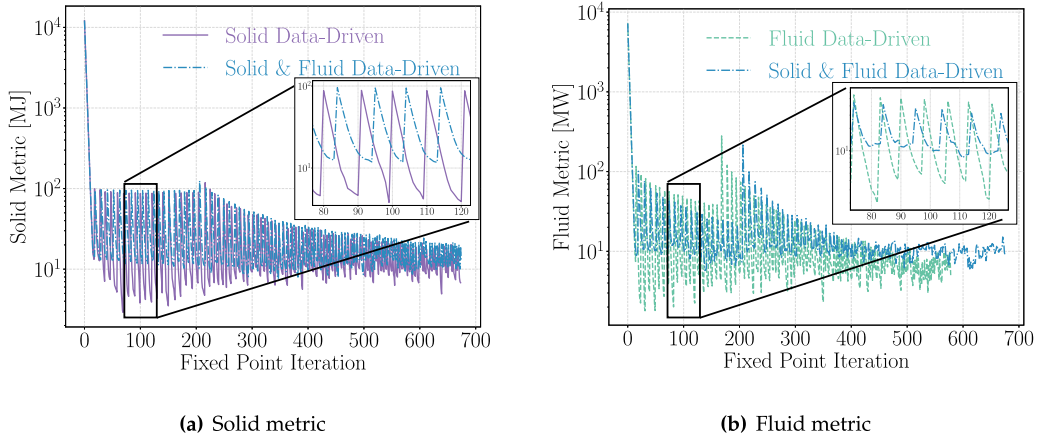


Fig. 18. Comparing (a) the total solid metric and (b) the total fluid metric at each fixed-point iteration for different schemes.

Table 4

Material parameters for Footing problem.

Physical parameter	Unit	Value
Young's modulus (E)	GPa	30
Poisson's ratio (ν)	–	0.2
Plane condition	–	Plane strain
Intrinsic permeability (k)	m ²	1×10^{-10}
Fluid dynamics viscosity (μ)	Pa s	0.001
Biot coefficient (B)	–	1
Biot modulus (M)	GPa	∞

$-28.7 \times 10^{-3} \leq \epsilon_{yy} \leq 5.04 \times 10^{-3}$, and $-0.72 \times 10^{-3} \leq \epsilon_{xy} \leq 12.1 \times 10^{-3}$. In the following results, to generate data, we divide each active dimension of sample space into N equidistant points. For example, database for solid part will have N^3 data points in total, since its sample space has only three active dimensions corresponding to symmetric strain tensor components.

We show pressure and von-Mises total stress contours at end of simulations in Figs. 20 and 21, respectively. The database for fluid data-driven simulation has 401^2 samples. The fully data-driven simulation has two separate data

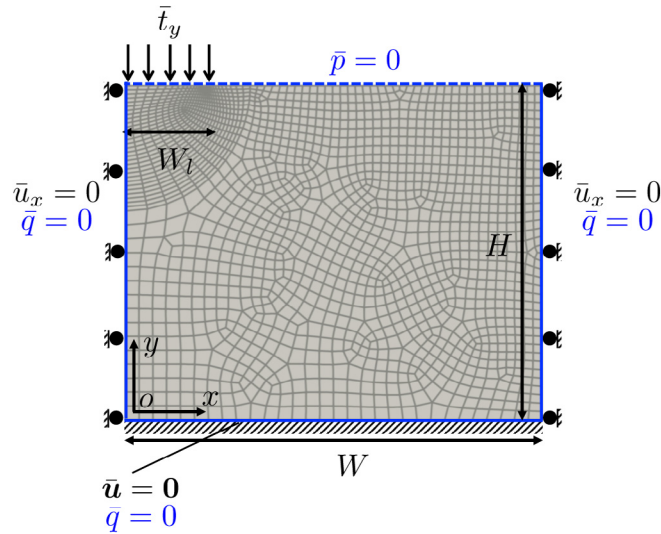


Fig. 19. Geometry, mesh, and boundary conditions for the footing problem with $W = 5W_l = 1.25H = 10$ m.

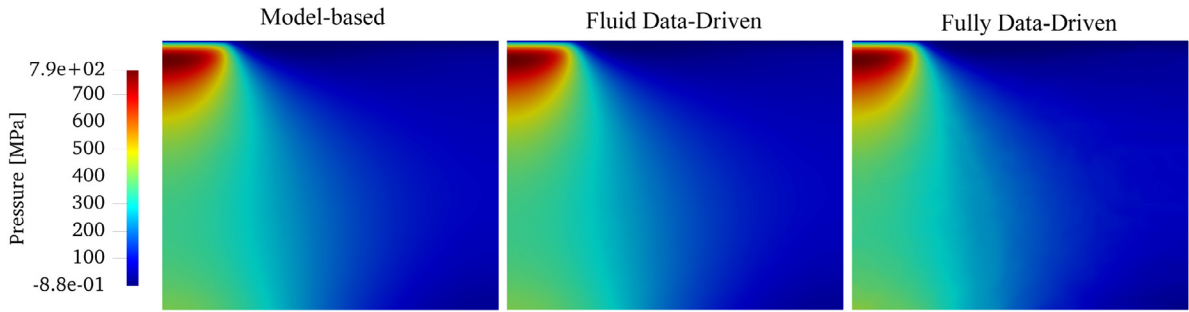


Fig. 20. Fluid pressure contour.

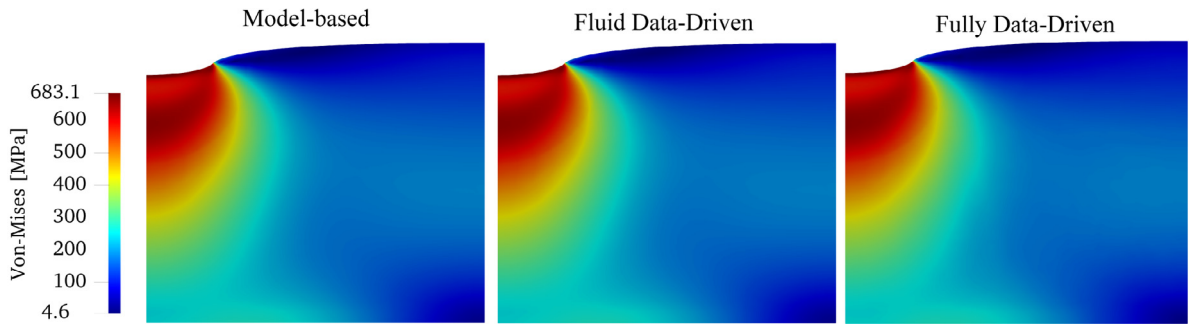


Fig. 21. Von Mises stress contour on the deformed geometry. Deformation is magnified by ten times.

sets for fluid and solid parts with 401^2 and 101^3 samples, respectively. These figures confirm a good agreement between model-based and data-driven solutions.

Table 5 compares the simulation time for the fully data-driven and hybrid formulations. Since the system of equations for all formulations has the same size, the significant source of computational time changes is the local optimization task (fixed-point iterations), which is affected by the quality and size of the database. Again, the fully data-driven formulation has higher computational time since its database consists of two sets; hence the local search is performed twice. Comparing the computational time between hybrid formulations is not straightforward since

Table 5

A comparison of computational time for different schemes. For each scheme, we run five times the same simulation setup with a different random seed corresponding to the data assignment for the first fixed-point iteration. The average and the standard deviation of CPU time are reported.

Formulation	CPU time [s] (kd-tree)
Hybrid 1: solid data-driven	360.31 \pm 34.44
Hybrid 2: fluid data-driven	404.40 \pm 12.4
Fully data-driven	621.44 \pm 6.01

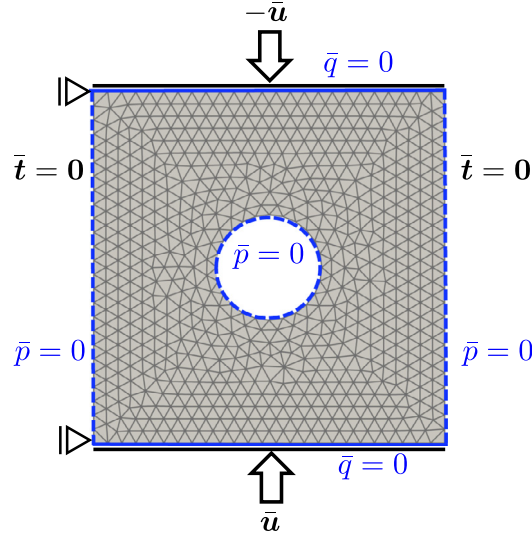


Fig. 22. Geometry, mesh, and boundary conditions for the plate problem with a hole. Hole radius is 0.3 m, and square has length of 2 m.

it depends on many factors such as the number of data points in their associated database, data density around admissible solutions, data coverage and distribution, etc. In this particular problem, although the solid database size is almost 6.5 times the fluid database size, the solid data-driven formulation requires less computational time. This difference in performance may be related to how the data and admissible material states distribute in the corresponding parametric spaces of the stress–strain and velocity–pressure–gradient databases.

5.4. Unconfined compression on poroelastic domain with a circular hole

This numerical example examines the efficiency of the hybridized model-based/model-free formulations and explores whether stress concentration and material nonlinearity may affect the accuracy and robustness of the data-driven poromechanics simulations. The effective stress–strain relation follows a modified constitutive law originally proposed in [34], which is a combination of Saint-Venant and Neo-Hookean material models expressed in the limit of small deformation kinematics, i.e.,

$$\psi(\epsilon) = \frac{G}{2} [\text{tr}(2\epsilon + \mathbf{I}) - 2 - 2\ln(1 + \text{tr}(\epsilon))] + \frac{\lambda}{2} [\ln(1 + \text{tr}(\epsilon))]^2 + G \text{tr}(\epsilon^2), \quad (69)$$

where $\lambda = \kappa - 2G/3$ and κ and G are bulk and shear moduli, respectively.

As shown in Fig. 22, top and bottom boundaries are rigidly compressed towards each other with rate 0.4 m/s. We run the simulation for $t_{\text{end}} = 1.2$ s with time increment $\Delta t = 0.2$ s. The applied deformation goes beyond the infinitesimal strain regime which might be non-physical, but this is intentional to activate severe nonlinear behavior. Material parameters are listed in Table 6. We discretize all unknown fields with Linear Lagrangian basis functions and use one Gaussian quadrature point to spatially integrate. Data sets for fluid part are synthetically generated using Darcy's law within the ranges $|\partial p / \partial x| \leq 4.1$ GPa and $|\partial p / \partial y| \leq 3.7$ GPa. Solid database is created by sampling

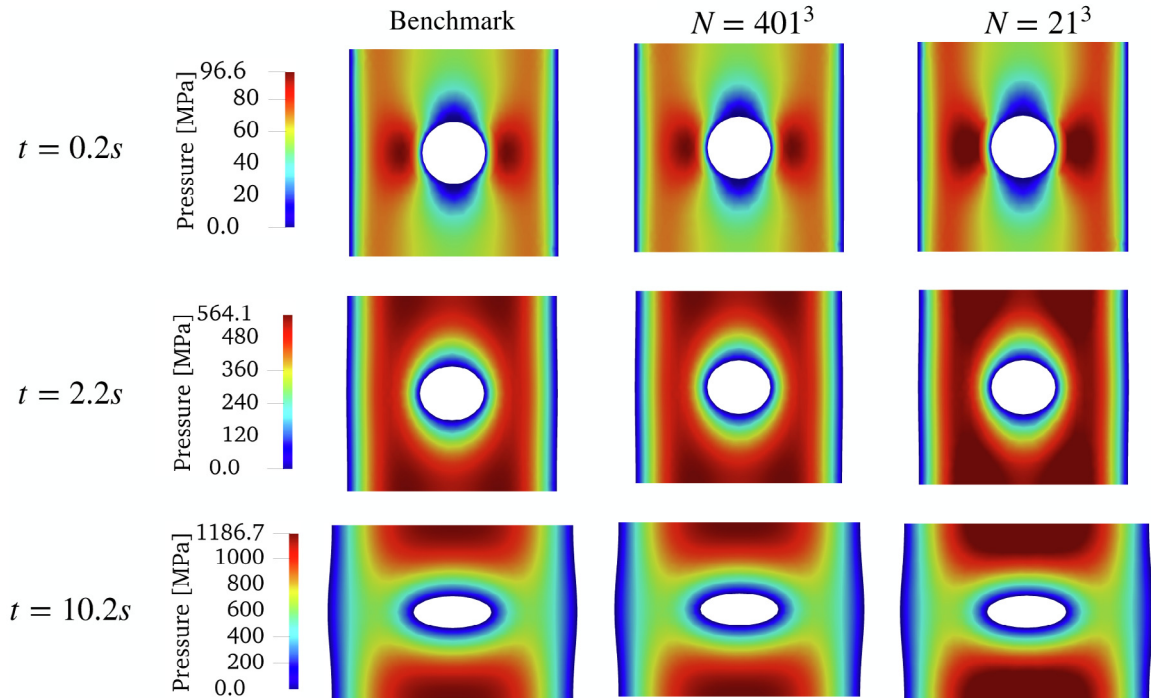


Fig. 23. Pore pressure contour evolution in time for different database sizes used in the fluid data-driven scheme.

Table 6

Material parameters for plate with hole.

Physical parameter	Unit	Value
Young's modulus (E)	GPa	30
Poisson's ratio (ν)	–	0.35
Plane condition	–	Plane strain
Intrinsic permeability (k)	m^2	3.0612×10^{-9}
Fluid dynamics viscosity (μ)	Pa s	0.001
Biot coefficient (B)	–	1
Biot modulus (M)	GPa	600

strain and effective stress pairs for strains within ranges $0 \leq \epsilon_{xx} \leq 0.25$, $-0.42 \leq \epsilon_{yy} \leq 0$, and $|\epsilon_{xy}| \leq 0.135$. Phase spaces for each data set are built by the equidistant sampling along each data axis, same as previous examples. We set numerical parameters for fluid metric equal to $8 \times 10^{-7} \mathbf{I} \text{ m}^2/\text{Pa s}$ and for solid metric equal to the Hessian tensor of functional Eq. (69) at $\epsilon_{xx} = 0.1$, $\epsilon_{yy} = -0.2$, and $\epsilon_{xy} = 0.05$ with the same material properties used for the data generation

In Fig. 23, there is a satisfactory agreement among benchmark solution, conventional model-based FEM solution, and results obtained by the fluid data-driven formulation with $N = 401^2$ data points for the pressure field. The low-density data $N = 21^2$ is still able to track the benchmark solution qualitatively. Fig. 24 presents the same results for solid data-driven formulation. In this figure, pressure fields at time $t = 0.2 \text{ s}$ are not as good as their counterparts in case of fluid data-driven. First and foremost, comparing results between $N = 21^3$ and $N = 401^3$ at time $t = 0.2 \text{ s}$ in Fig. 24 confirms that increasing amount of data improves the accuracy. In other words, case $N = 401^3$ still does not have sufficient data coverage to hit the benchmark's response surface. Second, the numerical parameter for solid metric function might not be appropriately chosen which has a considerable effect on the data-driven performance [33]. Fine-tuning process for numerical parameters in metric functions is out of the scope of this manuscript. Note that solid behavior in this problem is highly nonlinear, while the fluid constitutive behavior is linear. Also, solid phase-space belongs to a higher dimensional space than fluid phase-space. Therefore, in this

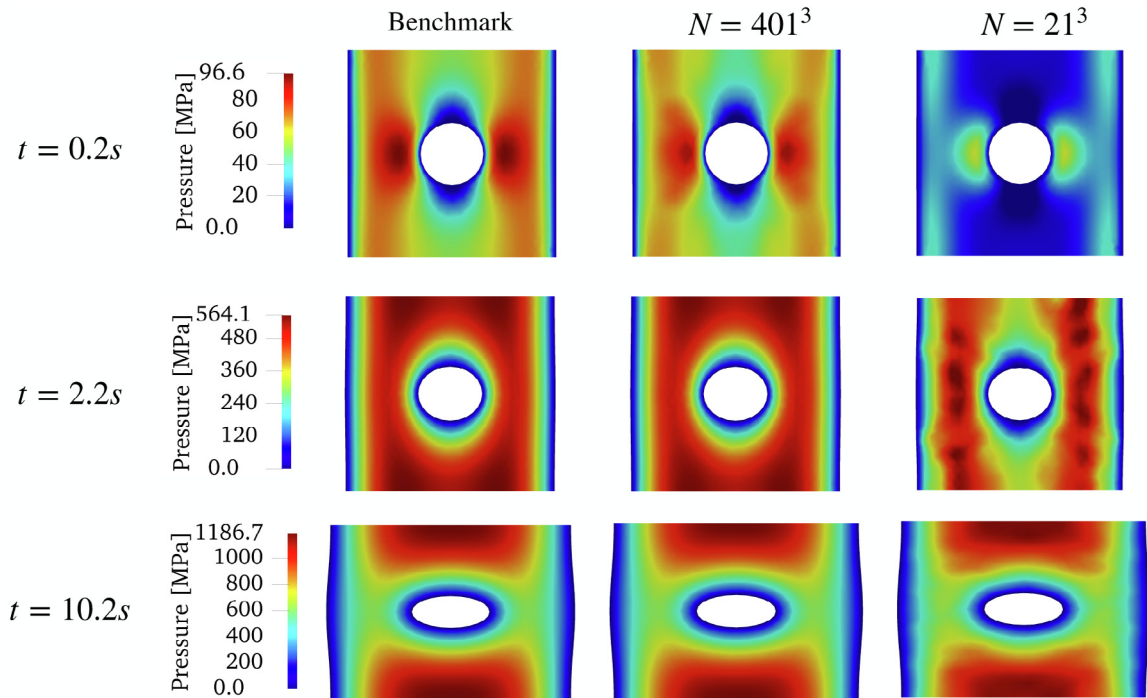


Fig. 24. Pore pressure contour evolution in time for different database sizes used in the solid data-driven scheme.

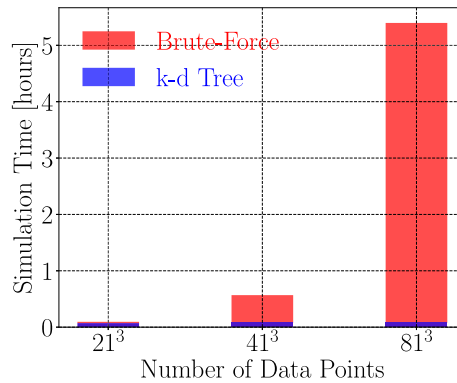


Fig. 25. Total simulation time for k-d tree and brute-force approaches.

particular problem, it is not unexpected to observe that fluid data-driven formulation, which benefits from an explicit model for solid constitutive law, outperforms solid data-driven formulation. In the other side, pressure responses of solid data-driven formulation at times $t = 2.2$ s, 10.2 s for $N = 401^3$ in Fig. 24 are in good agreement with benchmark.

We compare the performance of the kd-tree nearest neighbor search with the brute-force approach in Fig. 25. Not surprisingly, due to the logarithmic time complexity of the kd-tree search, the kd-tree outperforms the naive approach, especially for a large database (e.g., in the case $N = 81^3$, it reduces a five hours simulation to less than 10 min). Note that Newton–Raphson iterations increase the number of NNS requests in nonlinear problems. Consequently, a fast NNS approach may outperform the brute-force one, even for a small database. Recall the conclusion drawn in Appendix A where the brute-force scheme may have a slightly better performance for small data sets in dealing with a linear problem.

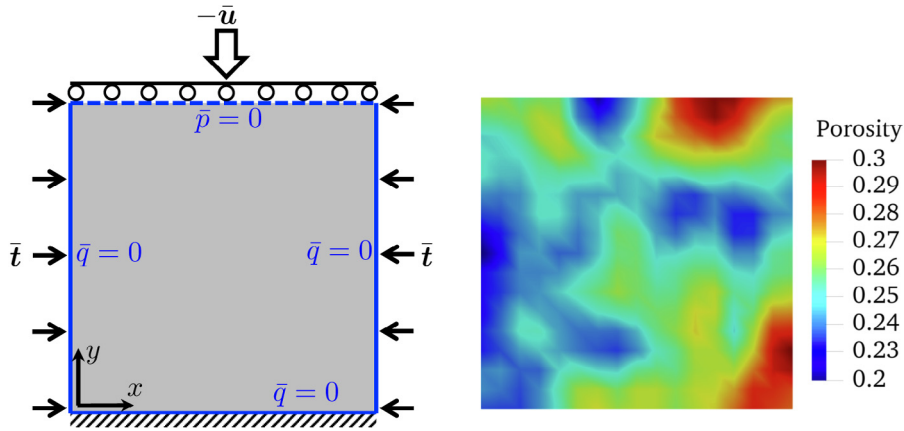


Fig. 26. (left) Geometry and boundary conditions for Berea sandstone and (right) heterogeneous distribution of initial porosity with average 0.25 and standard deviation 0.02.

5.5. Multifidelity simulations of berea sandstone data

This problem is designed to demonstrate how the data-driven formulation can be used to handle the different fidelities of the solid skeleton elasticity and the hydraulic responses of a porous medium. The difference in fidelity is a well-known issue in poromechanics problems where the predictions of the effective permeability are often much less accurate **and** precise than the solid constitutive responses [22,25,26,104]. For instance, a porosity–permeability model capable of predicting effective permeability with less than an order of the discrepancy from the experiment measurement is often considered accurate [105,106]. Meanwhile, experimental tests (e.g., uniaxial test and triaxial test) may routinely calibrate an isotropic linear elasticity model delivering predictions with a few percents of errors in the path-independent regime [22,107–109].

In these highly plausible scenarios, the benefit of replacing the solid constitutive law with the data-driven approach might not outweigh the efficiency of the model-based approach, whereas the data-driven approach may still be favorable due to the inevitable high variance of the predictions made by the deterministic porosity–permeability models. In this example, we constitute the fluid database (see Fig. 27(a)) by running 84 FFT simulations on microCT images of Berea sandstone (provided by collaborator Prof. Teng-fong Wong) and use the data-driven approach in Section 2.3 to run simulations without using hydraulic constitutive law. Meanwhile, the solid constitutive response is obtained from the literature [25].

The boundary value problem is two-dimensional, and the plane strain condition is assumed, as shown in Fig. 26(left). Lateral confining pressure of 6416.6 [MPa] is applied at the beginning of the simulation and kept constant throughout the simulations. A downward displacement is prescribed at the top boundary with the rate 1.2 mm/day. Time increment is set at $\Delta t = 12$ [h]. The spatial domain is discretized by 15×15 regular quadrilateral elements.

To test the versatility of the data-driven approach in handling boundary value problems with a spatially heterogeneous domain, we employ an open-source code developed by Baker et al. [110] to generate a realization of initial porosity distribution. This random porosity field is element-wise and is generated based on the exponential model with a mean value 0.25 and a standard deviation 0.02, shown in Fig. 26(right), which is in the range of values reported in [25]. Solid constitutive law is based on the hyperelastic energy functional introduced in chapter 6.4 of Borja [111] as follows:

$$\psi(\epsilon_v, \epsilon_s) = -p_0 c_r \exp\left(\frac{\epsilon_{v0} - \epsilon_v}{c_r}\right) - \frac{3}{2} c_\mu p_0 \exp\left(\frac{\epsilon_{v0} - \epsilon_v}{c_r}\right) \epsilon_s^2 + \psi_0, \quad (70)$$

where p_0 , c_r , c_μ , and ϵ_{v0} are model parameters, and $\epsilon_v = \text{tr}(\epsilon)$ and $\epsilon_s = \sqrt{2/3} \|\epsilon - 1/3 \epsilon_v \mathbf{I}\|$ are volumetric and shear strains, respectively. This model introduces pressure-dependent bulk and shear moduli which are considered necessary for modeling soils and sands [17,112]. In this model, bulk and shear moduli are related to diagonal

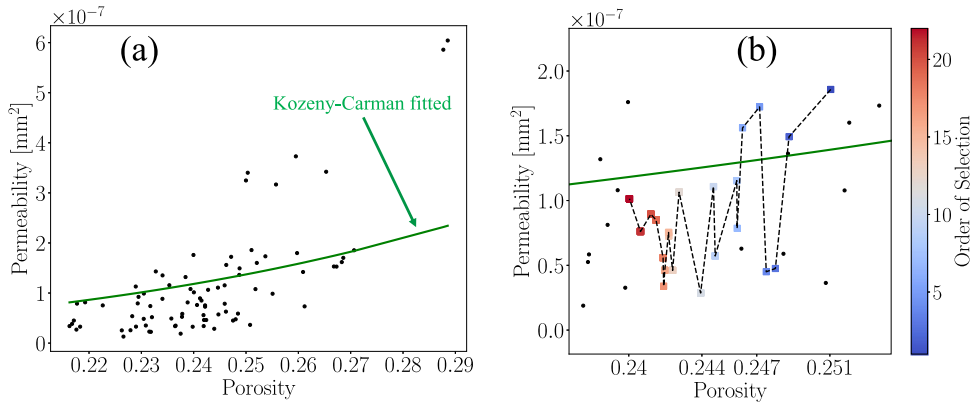


Fig. 27. (a) Fine-scale homogenized porosity–permeability data; (b) trajectory of data assignments for a quadrature point during the macro-scale simulation. The solid green line shows the fitted Kozeny–Carman curve to the points in figure (a). We observe a considerable discrepancy between the Kozeny–Carman model and ground truth points.

terms of Hessian matrix, 2nd derivative of energy functional with respect to volumetric and shear strains, with the following relations:

$$\kappa(\epsilon_v, \epsilon_s) = -\frac{p_0}{c_r} \left(1 + \frac{3c_\mu}{2c_r} \epsilon_s^2 \right) \exp\left(\frac{\epsilon_{v0} - \epsilon_v}{c_r}\right), \quad (71)$$

$$\mu(\epsilon_v, \epsilon_s) = -c_\mu p_0 \exp\left(\frac{\epsilon_{v0} - \epsilon_v}{c_r}\right). \quad (72)$$

Here, we use the bulk and shear moduli reported in [25], see Figs. 2(a) and 4(a) in the reference paper. Notice that, in their study, these moduli are porosity dependent which can be related to volumetric strain via $\phi = (1 + \epsilon_v)\phi_0$ for the small deformation analysis [13]. Exponential functions are fitted to their experimental curves for porosity values in the range [0.18, 0.32]. Therefore, for each computational element with a distinct initial porosity ϕ_0 we are able to calibrate model parameters p_0 , c_r , c_{μ} , and ϵ_{v0} to match bulk and shear moduli Eqs. (71) and (72) with those experimentally calibrated exponential functions. Since we do not have information about shear loading conditions in [25] we set $\epsilon_s = 0$ for just parameter calibration process.

A collection of upscaled effective permeability for 84 Representative Volume Elements (RVEs) corresponding to 3D images of Berea sandstone is obtained by solving Stokes equations at microscale using an in-house developed Fast Fourier Transform (FFT) based solver [113–115]. Fig. 27(a) shows these 84 porosity–permeability pairs. The highly non-smooth behavior in 27(a) indicates that porosity alone is not sufficient to predict permeability. However, without further information available, analytical regression is likely to lead to underfitting predictions. Here the effective permeability is assumed to be isotropic. For each point in Fig. 27(a), a data set of 801^2 pressure gradient and fluid velocity pairs is created and labeled for a specific porosity. All generated data sets cover same range of pressure gradients as $-902 \text{ [MPa/m]} \leq \partial p / \partial x \leq 1540 \text{ [MPa/m]}$ and $-3190 \text{ [MPa/m]} \leq \partial p / \partial x \leq 1650 \text{ [MPa/m]}$. After many trials and errors, these ranges are found by changing the bounds to make sure final numerical pressure gradients belong to them. Numerical parameter for fluid metric function is set to $10^{-11} \mathbf{I}$.

Since data sets have a dependence on porosity we propose a modification to the algorithm for material projection which consists of two steps. First, based on the current estimation of porosity at a quadrature point, the closest data set among 84 options will be selected. Then, inside the founded data set, the closest pressure gradient and fluid velocity pair will be reported similar to earlier problems.

Fig. 28 depicts the performance of solver at each fixed-point iteration. Each of those jumps (peaks) corresponds to an initial iteration at a new time step. The initial guess at a new time step is expected to be off due to the new loading condition, and so we should expect those jumps. We expect to converge to a constant metric value within each time step along with fixed-point iterations without any further material projection. Generally, these figures confirm the same behavior, but some time steps do not converge exactly to a constant metric value. Metric values at those time steps slightly oscillate, see behavior around iteration 300 in the zoomed snapshot of Fig. 28(a).

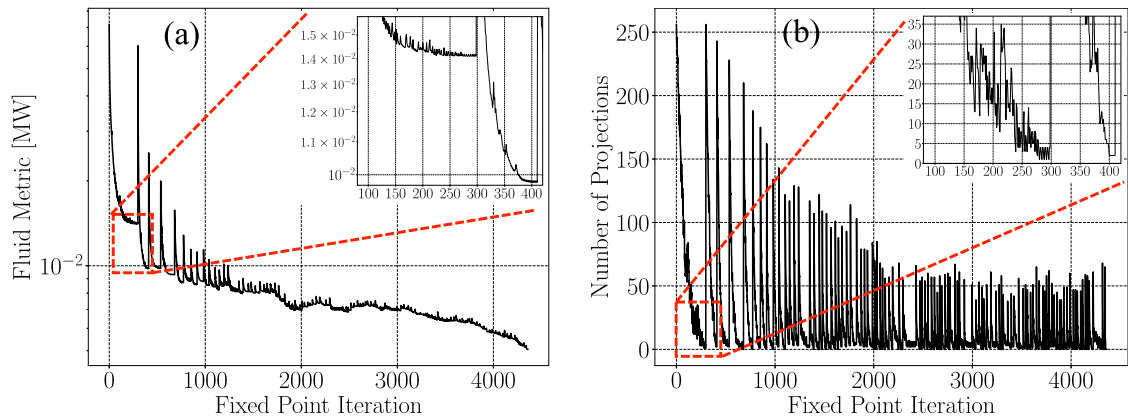


Fig. 28. (a) the total fluid metric and (b) the number of projected material points at the end of each fixed-point iteration for 100 time steps.

However, the total numbers of material projections are close to zero, see corresponding iterations in the zoomed snapshot of Fig. 28(b), and not more than 5 in the worst case happened in the simulation. Therefore, we can say the solver's overall performance is satisfactory because the exact minima for just a few material points (less than 3%) are lost. This oscillatory behavior is not something unexpected since at some regions in Fig. 27(a) porosity values are very close to each other while permeability values differ considerably and so, after some fixed-point iterations, there is a possibility for the solver to be trapped between several choices each selected from a different data set. It is well-known that heuristic approaches, e.g., fixed-point iteration used in our research, may not find the global optima for combinatorial optimization problems [116,117], i.e., trapped between some local minima. As suggested in [116] an alternative optimization problem can be formulated by a relaxation method which makes the optimization problem convex, and consequently, a global optimum is guaranteed. We have not studied such a formulation in the current research.

Porosity and pressure contours of selected data sets at the end of the material projection process are shown at different time stages in Figs. 29 and 30, respectively. As expected, for most of the quadrature points, the data points at lower porosity value are selected as time elapses. Fig. 27(b) showcases history of selected data sets in 100 time steps of the simulation for one of the quadrature points located at position (0.3, 3.7). Initially, a data point with porosity 0.251 is assigned to this point, and a data point with porosity 0.24 is selected at the end of simulation. This search iterates through almost 20 different permeability values with a considerable variance and no consensus trend, even for such a small variation of porosity values. If, instead, we used Kozeny–Carman relation to model porosity-dependent permeability we observed a smooth decay in permeability value by decreasing porosity.

Due to the high variance of the permeability dataset, selecting and calibrating a porosity–permeability model is by no mean a trivial task. While a complicated model fluctuated with porosity is likely to overfit the data, a simple model such as the Kozeny–Carman model may underfit the data and hence lead to significant errors (see solid green line in Fig. 27(a)). While extending the dimensions of the parametric space (e.g., adding features [118] or geometric measures such as fabric tensors [119] of the hydraulic model may improve the bias–variance trade-off, the increased dimensionality may also increase the difficulty of deriving calibrating the constitutive laws or the training of neural network models.

Note that the data-driven approach does not impose or require any smoothness assumption among the data points in the phase space. While the assumptions of smoothness, continuity or convexity are not always physically justified, the lack thereof of the data-driven approach does impose a high demand on the data. To make a prediction, there must be a sufficient population or density of data locally distributed in the parametric space where the data point becomes admissible to the physics constraints. Recent work such as Eggersmann et al. [120] introduces an approach based on tensor voting to make data-driven methods working with less intensive data, but the demand for data is still noticeably higher than the conventional approach. A model-free method that can operate under less data by incorporating physics constraints and statistics will be considered in the future but is out of the scope of this study.

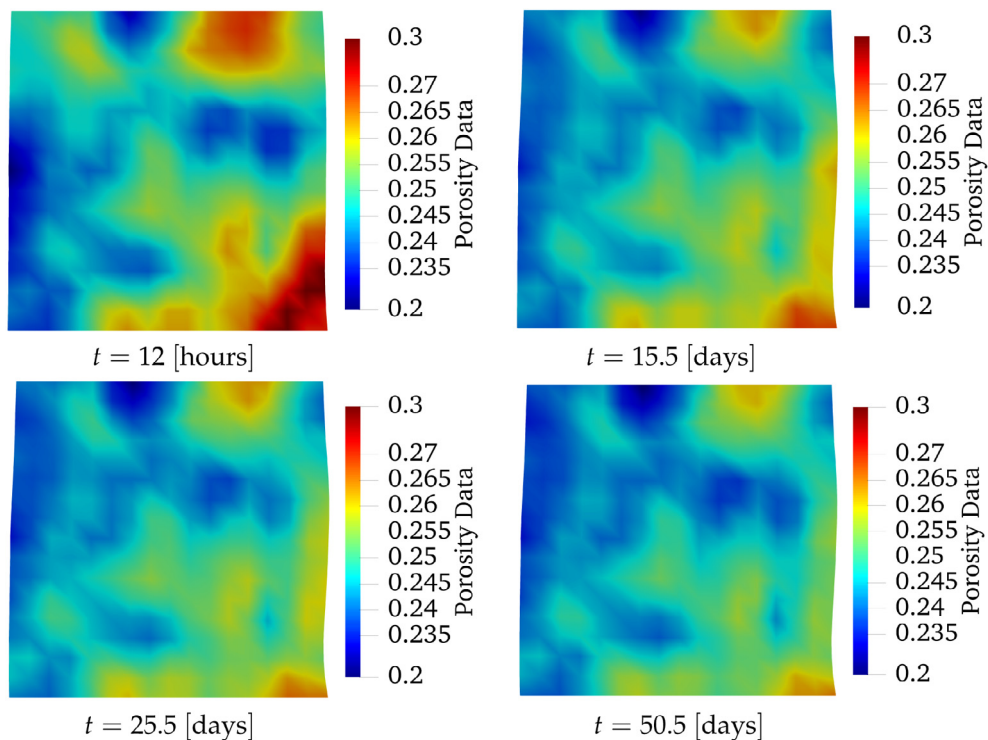


Fig. 29. Contours of assigned porosity on the deformed domain.

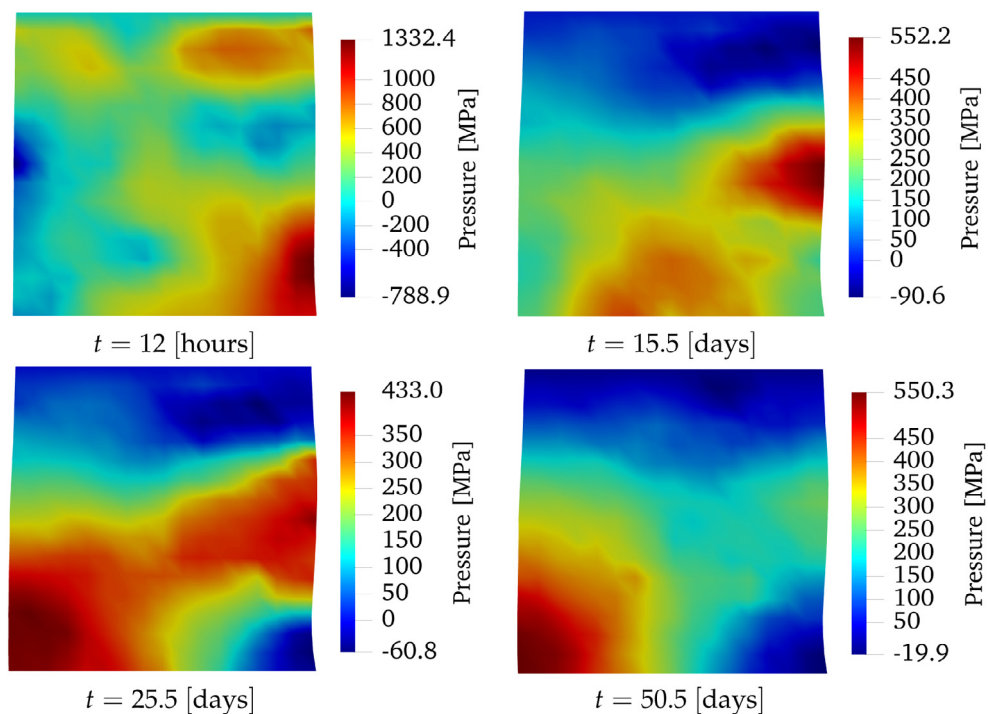


Fig. 30. Pressure contours on deformed body at different stages of loading.

6. Conclusion

We establish fully data-driven and hybridized data-driven/model-base frameworks that solve the poroelasticity problems either completely free of hand-crafted constitutive laws or having them partially replaced by a data-driven scheme. By extending the metric-minimization algorithm originally proposed for single-physics elasticity problems, we bypass the usages of material laws when the higher fluctuation hydraulic data may unavoidably lead to under-fitting. To overcome the significant computational time required to search the optimal data point from the discrete dataset, we introduce a k-d tree based nearest neighbor search for an energy metric for poromechanics problems. Comparisons between the k-d tree search and the brute-force search revealed that the k-d tree search may significantly shorten the search time, especially when the dataset is large.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the two anonymous reviewers for the constructive suggestions. The authors would also like to thank Professor Teng-fong Wong for providing the MicroCT images of the Berea sandstone and Dr. Ran Ma for programming the FFT solver that calculates the effective permeability of the Berea Sandstone and Dr. Guodong Zhang for providing the permeability database in Fig. 27. The authors are supported by the NSF CAREER grant from Mechanics of Materials and Structures program at National Science Foundation, United States of America under grant contract CMMI-1846875, and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research, United States of America under grant contracts FA9550-17-1-0169, and the Earth Materials and Processes program from the US Army Research Office under grant contract W911NF-18-2-0306, as well as the These supports are gratefully acknowledged. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Appendix A. K-D tree performance

Here we aim to clarify, in detail, why k-d tree search is a major step for the model-free formulation to reduce the computational cost and resource. To focus on just the k-d tree performance, we consider a static, single physics problem with a manufactured polynomial type solution which is fairly a trivial problem.

Consider a linear Poisson equation governed steady-state response of fluid flow through a cube of unity length (depicted in Fig. 31) as follows:

$$\nabla^x \cdot \mathbf{q} + s = 0, \quad (73)$$

$$\mathbf{q} = -k \nabla^x p, \quad (74)$$

where \mathbf{q} , s , k , and p are Darcy's velocity, source term, isotropic hydraulic conductivity, and pressures, respectively. The source term is designed to satisfy the solution $p = x^2 + y^2 + z^2$ where (x, y, z) is the spatial coordinate in the cube domain. The data-driven formulation for the Poisson equation is a simplified version of the poromechanics formulation when the solid phase is ignored, and transient effects are not involved. Therefore, we have omitted the extra derivation for the sake of brevity, and interested readers can refer to Nguyen et al. [34].

According to the exact solution for p , Dirichlet boundary conditions are applied over all external faces of the cube. Database consists of $\nabla^x p$ and \mathbf{q} pairs are sampled regularly in intervals $-1.1 \leq \frac{\partial p}{\partial x} \leq 1.1$, $-1.1 \leq \frac{\partial p}{\partial y} \leq 1.1$, and $-1.1 \leq \frac{\partial p}{\partial z} \leq 1.1$. For example, if each direction of the pressure gradient is sampled by 64 equally distance points in the interval $[-1.1, 1.1]$, then a database with 64^3 rows and 6 columns has been artificially generated. In this problem, seven data sets ranging from 2^3 to 128^3 points are created. Hydraulic conductivity k is set to unity. We use 4096 structured hexahedron elements to discretize the cube. Trilinear Lagrangian basis functions are utilized, and numerical integration is calculated by one quadrature point.

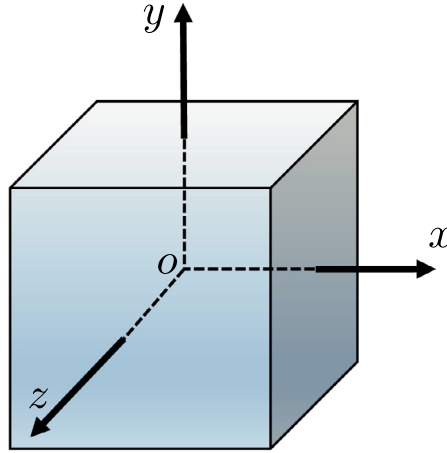


Fig. 31. Cubic domain where the origin is placed at its centroid.

Since the data-driven formulation might be sensitive to the initial random data assignment to each quadrature point at the first fixed-point iteration, we have run the same BVP for each algorithm three times. Therefore, in total, 42 time records are included in Fig. 32 and Fig. 33(b). These figures suggest that the simulation time in this particular problem is relatively insensitive to the random initialization.

As shown in Fig. 32, the brute-force approach is more efficient than the k-d tree for small data sets ($N \leq 16^3$), but after a threshold k-d tree exponentially outperforms the brute-force algorithm. Note that for $N = 128^3$ the k-d tree search is almost two orders of magnitude more efficient. Fig. 33(a) depicts the tree construction time which is an offline process. Time expressed in Fig. 33(b) is the simulation time from beginning to end, including post-processing, subtracted by the tree construction time.

Appendix B. Terzaghi's analytical solution

The exact solution for pressure and displacement profiles used in 5.1 is obtained from the following equations (see [94,121]):

$$p(y, t) = \frac{4bM\bar{t}_y}{\pi(E + b^2M)} \sum_{i=0}^{\infty} \frac{(-1)^{i+1}}{2i+1} \exp\left(-\frac{(2i+1)^2\pi^2c_f}{4H^2}t\right) \cos\left(\frac{(2i+1)\pi}{2H}y\right), \quad (75)$$

$$u_y(y, t) = \frac{y\bar{t}_y}{E + b^2M} + \frac{b^2M\bar{t}_y}{E(E + b^2M)} \left[y - \frac{8H}{\pi^2} \sum_{i=0}^{\infty} \frac{(-1)^{i+1}}{(2i+1)^2} \exp\left(-\frac{(2i+1)^2\pi^2c_f}{4H^2}t\right) \sin\left(\frac{(2i+1)\pi}{2H}y\right) \right], \quad (76)$$

where height $H = 1$ m and $0 \leq y \leq 1$ for the problem in Fig. 6. For the material properties in Table 1, fluid diffusivity coefficient is $c_f = \frac{2kME}{b^2M+E}$. Note that all the above relations are simplified for zero Poisson's ratio.

Appendix C. Stress relaxation analytical solution

The following exact equation for traction history at the top boundary is used in Fig. 14 (see [103]):

$$t_z(z = H, t) = \frac{\dot{u}_z(2G + \lambda)}{H} (t\text{Hev}(t_{\text{ramp}} - t) + t_{\text{ramp}}\text{Hev}(t - t_{\text{ramp}})) + \frac{2\dot{u}_z\mu_f H}{k} \sum_{i=0}^{\infty} \frac{1}{i^2\pi^2} (\text{Hev}(t_{\text{ramp}} - t)(1 - e^{-a_i t}) + \text{Hev}(t - t_{\text{ramp}})(e^{-a_i(t-t_{\text{ramp}})} - e^{-a_i t})), \quad (77)$$

where $a_i = i^2\pi^2k(2G + \lambda)/H^2$ and $\text{Hev}(\cdot)$ is the Heaviside step function, $\text{Hev}(x < 0) = 0$ and $\text{Hev}(x \geq 0) = 1$. G and λ are shear modulus and Lamé's parameter, respectively. \dot{u}_z is the rate of applied displacement.

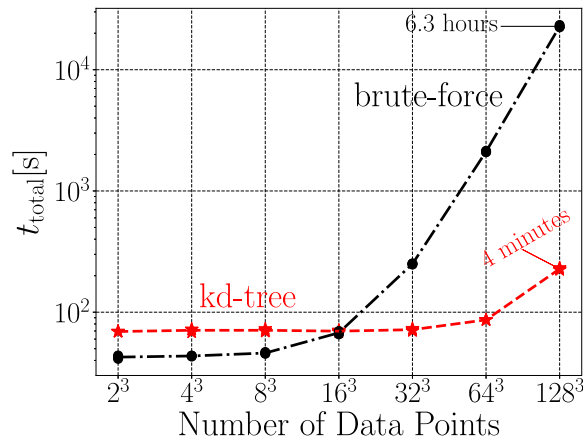


Fig. 32. A comparison of total simulation time (including post-processing) from start to end for seven data sets with different sizes. Each data set, e.g. data set corresponding to $N = 32^3$, is used for three simulations based on the k-d tree search and three simulations based on the brute-force search algorithms. All the 42 time records are included in this figure. Dash and dash-dot lines pass through the mean value of points corresponding to the k-d tree and brute-force categories, respectively. Note that each simulation starts with a random assignment of data at the first fixed-point iteration.

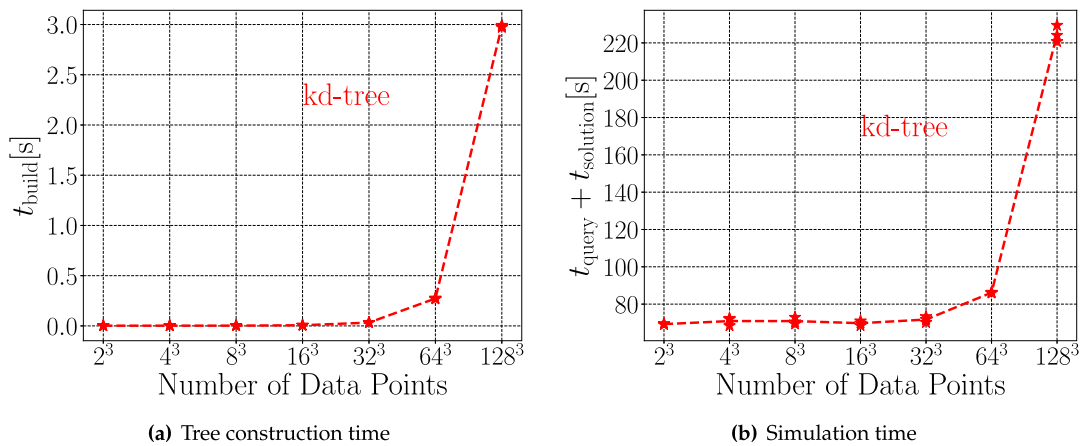


Fig. 33. (a) shows the tree construction time before the simulation for each data set. (b) shows the total time of simulation (including post-processing) subtracted by the tree construction time reported in (a). In (b), dash line passes through the mean value of time records for three trials corresponding to each data set (see caption Fig. 32 for more information).

References

- [1] Karl Terzaghi, Ralph B. Peck, Gholamreza Mesri, *Soil Mechanics in Engineering Practice*, John Wiley & Sons, 1996.
- [2] Maurice A. Biot, General theory of three-dimensional consolidation, *J. Appl. Phys.* 12 (2) (1941) 155–164.
- [3] Olivier Coussy, *Poromechanics*, John Wiley & Sons, 2004.
- [4] Massimo Cocco, James R. Rice, Pore pressure and poroelasticity effects in Coulomb stress analysis of earthquake interactions, *J. Geophys. Res.: Solid Earth* 107 (B2) (2002) ESE-2.
- [5] Guillaume Chambon, John W. Rudnicki, Effects of normal stress variations on frictional stability of a fluid-infiltrated fault, *J. Geophys. Res.: Solid Earth* 106 (B6) (2001) 11353–11372.
- [6] Emmanuel Detournay, Alexander H.-D. Cheng, Fundamentals of poroelasticity, in: *Analysis and Design Methods*, Elsevier, 1993, pp. 113–171.
- [7] Emmanuel Detournay, *Mechanics of hydraulic fractures*, *Annu. Rev. Fluid Mech.* 48 (2016) 311–339.
- [8] WaiChing Sun, A stabilized finite element formulation for monolithic thermo-hydro-mechanical simulations at finite strain, *Internat. J. Numer. Methods Engrg.* 103 (11) (2015) 798–839, <http://dx.doi.org/10.1002/nme.4910>.
- [9] SeonHong Na, WaiChing Sun, Computational thermo-hydro-mechanics for multiphase freezing and thawing porous media in the finite deformation range, *Comput. Methods Appl. Mech. Engrg.* 318 (2017) 667–700.

- [10] Stephen C. Cowin, Bone poroelasticity, *J. Biomech.* 32 (3) (1999) 217–238.
- [11] Olgierd C. Zienkiewicz, A.H.C. Chan, M. Pastor, B.A. Schrefler, T. Shiomi, *Computational Geomechanics*, Vol. 613, Citeseer, 1999.
- [12] WaiChing Sun, A unified method to predict diffuse and localized instabilities in sands, *Geomech. Geoeng.* 8 (2) (2013) 65–75.
- [13] Kun Wang, WaiChing Sun, A unified variational eigen-erosion framework for interacting brittle fractures and compaction bands in fluid-infiltrating porous media, *Comput. Methods Appl. Mech. Engrg.* 318 (2017) 1–32.
- [14] Kun Wang, WaiChing Sun, An updated Lagrangian LBM–DEM–FEM coupling model for dual-permeability fissured porous media with embedded discontinuities, *Comput. Methods Appl. Mech. Engrg.* 344 (2019) 276–305.
- [15] Maria Laura De Bellis, Gabriele Della Vecchia, Michael Ortiz, Anna Pandolfi, A multiscale model of distributed fracture and permeability in solids in all-round compression, *J. Mech. Phys. Solids* 104 (2017) 12–31.
- [16] G. Renaud, Julie Rivière, P.-Y. Le Bas, P.A. Johnson, Hysteretic nonlinear elasticity of Berea sandstone at low-vibrational strain revealed by dynamic acousto-elastic testing, *Geophys. Res. Lett.* 40 (4) (2013) 715–719.
- [17] Ronaldo I. Borja, Claudio Tamagnini, Angelo Amorosi, Coupling plasticity and energy-conserving elasticity models for clays, *J. Geotech. Geoenviron. Eng.* 123 (10) (1997) 948–957.
- [18] Eric C. Bryant, WaiChing Sun, A micromorphically regularized Cam-clay model for capturing size-dependent anisotropy of geomaterials, *Comput. Methods Appl. Mech. Engrg.* 354 (2019) 56–95.
- [19] SeonHong Na, Eric C. Bryant, WaiChing Sun, A configurational force for adaptive re-meshing of gradient-enhanced poromechanics problems with history-dependent variables, *Comput. Methods Appl. Mech. Engrg.* 357 (2019) 112572.
- [20] Donald A. Cameron, John P. Carter, A constitutive model for sand based on non-linear elasticity and the state parameter, *Comput. Geotech.* 36 (7) (2009) 1219–1228.
- [21] Ronaldo I. Borja, *Plasticity: Modeling & Computation*, Springer Science & Business Media, 2013.
- [22] Mervyn S. Paterson, Teng-fong Wong, *Experimental Rock Deformation-The Brittle Field*, Springer Science & Business Media, 2005.
- [23] WaiChing Sun, Jose E. Andrade, John W. Rudnicki, Multiscale method for characterization of porous microstructures and their impact on macroscopic effective permeability, *Internat. J. Numer. Methods Engrg.* 88 (12) (2011) 1260–1279.
- [24] WaiChing Sun, José E. Andrade, John W. Rudnicki, Peter Eichhubl, Connecting microstructural attributes and permeability from 3D tomographic images of in situ shear-enhanced compaction bands using multiscale computations, *Geophys. Res. Lett.* 38 (10) (2011).
- [25] Heiko Andrä, Nicolas Combaret, Jack Dvorkin, Erik Glatt, Junehee Han, Matthias Kabel, Youngseuk Keehm, Fabian Krzikalla, Minhui Lee, Claudio Madonna, et al., Digital rock physics benchmarks—Part II: Computing effective properties, *Comput. Geosci.* 50 (2013) 33–43.
- [26] WaiChing Sun, Teng-fong Wong, Prediction of permeability and formation factor of sandstone with hybrid lattice Boltzmann/finite element simulation on microtomographic images, *Int. J. Rock Mech. Min. Sci.* 106 (2018) 269–277.
- [27] Hyoung Suk Suh, WaiChing Sun, An immersed phase field fracture model for microporomechanics with Darcy–Stokes flow, *Phys. Fluids* 33 (1) (2021) 016603.
- [28] Yousef Heider, Hyoung Suk Suh, WaiChing Sun, An offline multi-scale unsaturated poromechanics model enabled by self-designed/self-improved neural networks, *Int. J. Numer. Anal. Methods Geomech.* (2021).
- [29] Heiko Andrä, Nicolas Combaret, Jack Dvorkin, Erik Glatt, Junehee Han, Matthias Kabel, Youngseuk Keehm, Fabian Krzikalla, Minhui Lee, Claudio Madonna, et al., Digital rock physics benchmarks—Part I: Imaging and segmentation, *Comput. Geosci.* 50 (2013) 25–32.
- [30] Trenton Kirchdoerfer, Michael Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 81–101.
- [31] Ruben Ibañez, Domenico Borzacchiello, Jose Vicente Aguado, Emmanuelle Abisset-Chavanne, Elías Cueto, Pierre Ladevèze, Francisco Chinesta, Data-driven non-linear elasticity: constitutive manifold construction and problem discretization, *Comput. Mech.* 60 (5) (2017) 813–826.
- [32] Qizhi He, Jiun-Shyan Chen, A physics-constrained data-driven approach based on locally convex reconstruction for noisy database, *Comput. Methods Appl. Mech. Engrg.* 363 (2020) 112791.
- [33] Adrien Leygue, Michel Coret, Julien Réthoré, Laurent Stainier, Erwan Verron, Data-based derivation of material response, *Comput. Methods Appl. Mech. Engrg.* 331 (2018) 184–196.
- [34] Lu Trong Khiem Nguyen, Matthias Rambausek, Marc-André Keip, Variational framework for distance-minimizing method in data-driven computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 365 (2020) 112898.
- [35] K. Karapiperis, L. Stainier, M. Ortiz, J.E. Andrade, Data-driven multiscale modeling in mechanics, *J. Mech. Phys. Solids* 147 (2021) 104239.
- [36] Jean H. Prevost, Wave propagation in fluid-saturated porous media: an efficient finite element procedure, *Int. J. Soil Dyn. Earthq. Eng.* 4 (4) (1985) 183–202.
- [37] Ronaldo I. Borja, Enrique Alarcón, A mathematical framework for finite strain elastoplastic consolidation Part 1: Balance laws, variational formulation, and linearization, *Comput. Methods Appl. Mech. Engrg.* 122 (1–2) (1995) 145–171.
- [38] W. Sun, J.T. Ostien, A.G. Salinger, A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain, *Int. J. Numer. Anal. Methods Geomech.* 37 (16) (2013) 2755–2788.
- [39] W. Sun, Q. Chen, J.T. Ostien, Modeling hydro-mechanical responses of strip and circular footings on saturated collapsible geomaterials, *Acta Geotech.* (2014) <http://dx.doi.org/10.1007/s11440-013-0276-x>.
- [40] Karl Terzaghi, Karl Terzaghi, Civil Engineer, Austria Czechoslovakia, Karl Terzaghi, Ingénieur Civil, Autriche Tchecoslovaquie, Etats Unis, *Theoretical Soil Mechanics*, Vol. 18, Wiley, New York, 1943.
- [41] C.W.A. Cryer, A comparison of the three-dimensional consolidation theories of Biot and Terzaghi, *Quart. J. Mech. Appl. Math.* 16 (4) (1963) 401–412.
- [42] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Yoshua Bengio, *Deep Learning*, Vol. 1, MIT press, Cambridge, 2016.
- [43] Jamshid Ghaboussi, David A. Pecknold, Mingfu Zhang, Rami M. Haj-Ali, Autoprogressive training of neural network constitutive models, *Internat. J. Numer. Methods Engrg.* 42 (1) (1998) 105–126.

- [44] Kun Wang, WaiChing Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, *Comput. Methods Appl. Mech. Engrg.* 334 (2018) 337–380.
- [45] A.M. Tartakovsky, C. Ortiz Marrero, Paris Perdikaris, G.D. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, *Water Resour. Res.* 56 (5) (2020) e2019WR026731.
- [46] Trenton Kirchdoerfer, Michael Ortiz, Data-driven computing in dynamics, *Internat. J. Numer. Methods Engrg.* 113 (11) (2018) 1697–1710.
- [47] Alejandro Mota, Qiushi Chen, James W. Foulk III, Jakob T. Ostien, Zhengshou Lai, A cartesian parametrization for the numerical analysis of material instability, *Internat. J. Numer. Methods Engrg.* 108 (2) (2016) 156–180.
- [48] Yousef Heider, Kun Wang, WaiChing Sun, SO (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, *Comput. Methods Appl. Mech. Engrg.* 363 (2020) 112875.
- [49] Turab Lookman, Prasanna V. Balachandran, Dezheng Xue, Ruihao Yuan, Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design, *npj Comput. Mater.* 5 (1) (2019) 1–17.
- [50] Kun Wang, WaiChing Sun, Qiang Du, A non-cooperative meta-modeling game for automated third-party calibrating, validating, and falsifying constitutive laws with parallelized adversarial attacks, *Comput. Methods Appl. Mech. Engrg.* 373 (113514) (2021).
- [51] Zeliang Liu, M.A. Bessa, Wing Kam Liu, Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials, *Comput. Methods Appl. Mech. Engrg.* 306 (2016) 319–341.
- [52] Lei Zhang, Shaoqiang Tang, Cheng Yu, Xi Zhu, Wing Kam Liu, Fast calculation of interaction tensors in clustering-based homogenization, *Comput. Mech.* 64 (2) (2019) 351–364.
- [53] Jean-Pierre Bardet, *Experimental Soil Mechanics*, Prentice Hall, Upper Saddle River, NJ, 1997.
- [54] Carlos A. Felippa, Kwang-Chun Park, Charbel Farhat, Partitioned analysis of coupled mechanical systems, *Comput. Methods Appl. Mech. Engrg.* 190 (24–25) (2001) 3247–3270.
- [55] Michael J. Borden, Clemens V. Verhoosel, Michael A. Scott, Thomas J.R. Hughes, Chad M. Landis, A phase-field description of dynamic brittle fracture, *Comput. Methods Appl. Mech. Engrg.* 217 (2012) 77–95.
- [56] Tianchen Hu, Johann Guilleminot, John E. Dolbow, A phase-field model of fracture with frictionless contact and random fracture properties: Application to thin-film fracture and soil desiccation, *Comput. Methods Appl. Mech. Engrg.* 368 (2020) 113106.
- [57] Carlos A. Felippa, A survey of parametrized variational principles and applications to computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 113 (1–2) (1994) 109–139.
- [58] Kyuichiro Washizu, *Variational Methods in Elasticity and Plasticity*, Vol. 3, Pergamon press, Oxford, 1975.
- [59] Douglas N. Arnold, Richard S. Falk, A new mixed formulation for elasticity, *Numer. Math.* 53 (1–2) (1988) 13–30.
- [60] Johannes Karsawe, Gerhard Starke, Wenqing Wang, Olaf Kolditz, Finite element analysis of poro-elastic consolidation in porous media: Standard and mixed approaches, *Comput. Methods Appl. Mech. Engrg.* 195 (9–12) (2006) 1096–1115.
- [61] S. Teichtmeister, S. Mauthe, C. Miehe, Aspects of finite element formulations for the coupled problem of poroelasticity based on a canonical minimization principle, *Comput. Mech.* 64 (3) (2019) 685–716.
- [62] Frederik Fahrendorf, Simone Morganti, Alessandro Reali, Thomas J.R. Hughes, Laura De Lorenzis, Mixed stress-displacement isogeometric collocation for nearly incompressible elasticity and elastoplasticity, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113112.
- [63] Paul T. Bauman, Hachmi Ben Dhia, Nadia Elkhodja, J. Tinsley Oden, Serge Prudhomme, On the application of the arlequin method to the coupling of particle and continuum models, *Comput. Mech.* 42 (4) (2008) 511–530.
- [64] Éric Béchet, Nicolas Moës, Barbara Wohlmuth, A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method, *Internat. J. Numer. Methods Engrg.* 78 (8) (2009) 931–954.
- [65] Fushen Liu, Ronaldo I. Borja, Stabilized low-order finite elements for frictional contact with the extended finite element method, *Comput. Methods Appl. Mech. Engrg.* 199 (37–40) (2010) 2456–2471.
- [66] WaiChing Sun, Zhijun Cai, Jinhyun Choo, Mixed arlequin method for multiscale poromechanics problems, *Internat. J. Numer. Methods Engrg.* 111 (7) (2017) 624–659.
- [67] Haoyan Wei, Jiun-Shyan Chen, Frank Beckwith, Jonghyuk Baek, A naturally stabilized semi-Lagrangian meshfree formulation for multiphase porous media with application to landslide modeling, *J. Eng. Mech.* 146 (4) (2020) 04020012.
- [68] Jason S. Howell, Noel J. Walkington, Inf-sup conditions for twofold saddle point problems, *Numer. Math.* 118 (4) (2011) 663.
- [69] F. Brezzi, D. Boffi, L. Demkowicz, R.G. Durán, R.S. Falk, M. Fortin, Mixed finite elements, compatibility conditions, and applications, *Springer* 2 (2008) 4–2.
- [70] Marius Muja, David G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (11) (2014) 2227–2240.
- [71] Gregory Shakhnarovich, Trevor Darrell, Piotr Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice* (Neural Information Processing), The MIT Press, ISBN: 026219547X, 2006.
- [72] Yury A. Malkov, Dmitry A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).
- [73] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, Erik Waingarten, Optimal hashing-based time-space trade-offs for approximate near neighbors, in: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2017, pp. 47–66.
- [74] Jerome H. Friedman, Jon Louis Bentley, Raphael Ari Finkel, An algorithm for finding best matches in logarithmic expected time, *ACM Trans. Math. Software* (ISSN: 0098-3500) 3 (3) (1977) 209–226, <http://dx.doi.org/10.1145/355744.355745>.
- [75] Songrit Maneewongvatana, David M. Mount, Analysis of approximate nearest neighbor searching with clustered point sets, 1999, CoRR cs.CG/9901013. URL <https://arxiv.org/abs/cs/9901013>.

- [76] Jon Louis Bentley, Multidimensional binary search trees used for associative searching, *Commun. ACM* 18 (9) (1975) 509–517.
- [77] Peter N. Yianilos, Data structures and algorithms for nearest neighbor search in general metric spaces, in: *Soda*, Vol. 93, 1993, pp. 311–321.
- [78] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger, The R*-tree: an efficient and robust access method for points and rectangles, in: *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, 1990, pp. 322–331.
- [79] Stefan Berchtold, Daniel A. Keim, Hans-Peter Kriegel, The X-tree: An index structure for high-dimensional data, in: *Very Large Data-Bases*, 1996, pp. 28–39.
- [80] Piotr Indyk, *Nearest Neighbors in High-Dimensional Spaces*, Citeseer, 2004.
- [81] Nazneen Rajani, Kate McArdle, Inderjit S. Dhillon, Parallel k nearest neighbor graph construction using tree-based data structures, in: *1st High Performance Graph Mining Workshop*, Vol. 1, 2015, pp. 3–11.
- [82] Alexandr Andoni, Piotr Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, IEEE, 2006, pp. 459–468.
- [83] Chanop Silpa-Anan, Richard Hartley, Optimised KD-trees for fast image descriptor matching, in: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.
- [84] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, *Nature Methods* 17 (3) (2020) 261–272.
- [85] Marius Muja, David G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, *VISAPP* (1) 2 (331–340) (2009) 2.
- [86] Jeff Johnson, Matthijs Douze, Hervé Jégou, Billion-scale similarity search with GPUs, *IEEE Trans. Big Data* (2019).
- [87] Erik Bernhardsson, ANNOY: Approximate nearest neighbors in C++/Python optimized for memory usage and loading/saving to disk, 2013, 2013, URL <https://github.com/spotify/annoy>.
- [88] Der-Tsai Lee, C.K. Wong, Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees, *Acta Inform.* 9 (1) (1977) 23–29.
- [89] Songrit Maneewongvatana, David M. Mount, It's okay to be skinny, if your friends are fat, 1999.
- [90] Morteza M. Mehrabadi, Stephen C. Cowin, Eigentensors of linear anisotropic elastic materials, *Quart. J. Mech. Appl. Math.* 43 (1) (1990) 15–41.
- [91] Mikhail Itskov, On the theory of fourth-order tensors and their applications in computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 189 (2) (2000) 419–438.
- [92] Roger A. Horn, Charles R. Johnson, *Matrix Analysis*, Cambridge university press, 2012.
- [93] Ronaldo I. Borja, One-step and linear multistep methods for nonlinear consolidation, *Comput. Methods Appl. Mech. Engrg.* 85 (3) (1991) 239–272.
- [94] N. Castelletto, J.A. White, H.A. Tchelepi, Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics, *Int. J. Numer. Anal. Methods Geomech.* (ISSN: 03639061) 39 (14) (2015) 1593–1618, <http://dx.doi.org/10.1002/nag.2400>, URL <http://doi.wiley.com/10.1002/nag.2400>.
- [95] J. Kim, H.A. Tchelepi, R. Juanes, Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits, *Comput. Methods Appl. Mech. Engrg.* 200 (13) (2011) 1591–1606.
- [96] Joshua A. White, Ronaldo I. Borja, Stabilized low-order finite elements for coupled solid-deformation/fluid-diffusion and their application to fault zone transients, *Comput. Methods Appl. Mech. Engrg.* 197 (49–50) (2008) 4353–4366.
- [97] WaiChing Sun, Qiushi Chen, Jakob T. Ostien, Modeling the hydro-mechanical responses of strip and circular punch loadings on water-saturated collapsible geomaterials, *Acta Geotech.* 9 (5) (2014) 903–934.
- [98] Haoyan Wei, Jiun-Shyan Chen, Michael Hillman, A stabilized nodally integrated meshfree formulation for fully coupled hydro-mechanical analysis of fluid-saturated porous media, *Comput. & Fluids* 141 (2016) 105–115.
- [99] Kun Wang, WaiChing Sun, A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 546–583.
- [100] Pietro Carrara, Laura De Lorenzis, Laurent Stainier, Michael Ortiz, Data-driven fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113390.
- [101] Isaac Harari, Guillermo Hauke, Semidiscrete formulations for transient transport at small time steps, *Int. J. Numer. Methods Fluids* 54 (6–8) (2007) 731–743.
- [102] WaiChing Sun, Jakob T. Ostien, Andrew G. Salinger, A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain, *Int. J. Numer. Anal. Methods Geomech.* 37 (16) (2013) 2755–2788.
- [103] Mansoor A. Haider, Farshid Guilak, Application of a three-dimensional poroelastic beam to modeling the biphasic mechanics of cell–matrix interactions in articular cartilage, *Comput. Methods Appl. Mech. Engrg.* 196 (31–32) (2007) 2999–3010.
- [104] Peter A. Lock, Xudong Jing, Robert W. Zimmerman, Erika M. Schlueter, Predicting the permeability of sandstone from image analysis of pore structure, *J. Appl. Phys.* 92 (10) (2002) 6311–6319.
- [105] Y. Bernabé, U. Mok, B. Evans, Permeability-porosity relationships in rocks subjected to various evolution processes, *Pure Appl. Geophys.* 160 (5–6) (2003) 937–960.
- [106] Antonio Costa, Permeability-porosity relationship: A reexamination of the Kozeny-Carman equation based on a fractal pore-space geometry assumption, *Geophys. Res. Lett.* 33 (2) (2006).
- [107] Jack Dvorkin, Manika Prasad, Akio Sakai, Dawn Lavoie, Elasticity of marine sediments: Rock physics modeling, *Geophys. Res. Lett.* 26 (12) (1999) 1781–1784.

- [108] Tarek I. Zohdi, P. Wriggers, Computational micro-macro material testing, *Arch. Comput. Methods Eng.* 8 (2) (2001) 131–228.
- [109] L. Pimienta, J. Fortin, Y. Guéguen, Bulk modulus dispersion and attenuation in sandstones, *Geophysics* 80 (2) (2015) D111–D127.
- [110] Jack W. Baker, Andrew Seifried, Jose E. Andrade, Qiushi Chen, Characterization of random fields at multiple scales: an efficient conditional simulation procedure and applications in geomechanics, *Appl. Stat. Probab. Civil Eng.* (2011) 347.
- [111] Ronaldo I. Borja, *Plasticity*, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-38546-9, 2013, <http://dx.doi.org/10.1007/978-3-642-38547-6>, URL <http://link.springer.com/10.1007/978-3-642-38547-6>.
- [112] José E. Andrade, Ronaldo I. Borja, Capturing strain localization in dense sands with random density, *Internat. J. Numer. Methods Engrg.* 67 (11) (2006) 1531–1564.
- [113] Ran Ma, WaiChing Sun, FFT-based solver for higher-order and multi-phase-field fracture models applied to strongly anisotropic brittle materials, *Comput. Methods Appl. Mech. Engrg.* 362 (2020) 112781.
- [114] Ran Ma, WaiChing Sun, Computational thermomechanics for crystalline rock. Part II: Chemo-damage-plasticity and healing in strongly anisotropic polycrystals, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113184.
- [115] Ran Ma, WaiChing Sun, Phase field modeling of coupled crystal plasticity and deformation twinning in polycrystals with monolithic and splitting solvers, *Internat. J. Numer. Methods Engrg.* 112 (2021) 1167–1189.
- [116] Yoshihiro Kanno, Mixed-integer programming formulation of a data-driven solver in computational elasticity, *Optim. Lett.* 13 (7) (2019) 1505–1514.
- [117] Cristian Guillermo Gebhardt, Marc Christian Steinbach, Dominik Schillinger, Raimund Rolfes, Framework for data-driven structural analysis in general elasticity based on nonlinear optimization: The dynamic case, *Internat. J. Numer. Methods Engrg.* (2020).
- [118] Oleg Sudakov, Evgeny Burnaev, Dmitry Koroteev, Driving digital rock towards machine learning: Predicting permeability with gradient boosting and deep neural networks, *Comput. Geosci.* 127 (2019) 91–98.
- [119] WaiChing Sun, Matthew R. Kuhn, John W. Rudnicki, A multiscale DEM-LBM analysis on permeability evolutions inside a dilatant shear band, *Acta Geotech.* 8 (5) (2013) 465–480.
- [120] Robert Eggersmann, Laurent Stainier, Michael Ortiz, Stefanie Reese, Model-free data-driven computational mechanics enhanced by tensor voting, *Computer Methods in Applied Mechanics and Engineering* 373 (2021) 113499.
- [121] Herbert F. Wang, *Theory of Linear Poroelasticity with Applications to Geomechanics and Hydrogeology*, Princeton University Press, 2000.