Math Geosci
https://doi.org/10.1007/s11004-020-09893-y

SPECIAL ISSUE

Finite Element Solvers for Biot's Poroelasticity Equations in Porous Media

T. Kadeethum[1,2] · S. Lee[3] · H. M. Nick[1]

Received: 4 January 2020 / Accepted: 14 September 2020

Abstract We study and compare five different combinations of finite element spaces for approximating the coupled flow and solid deformation system, so-called Biot's equations. The permeability and porosity fields are heterogeneous and depend on solid displacement and fluid pressure. We provide detailed comparisons among the continuous Galerkin, discontinuous Galerkin, enriched Galerkin, and two types of mixed finite element methods. Several advantages and disadvantages for each of the above techniques are investigated by comparing local mass conservation properties, the accuracy of the flux approximation, number of degrees of freedom (DOF), and wall and CPU times. Three-field formulation methods with fluid velocity as an additional primary variable generally require a larger number of DOF, longer wall and CPU times, and a greater number of iterations in the linear solver in order to converge. The two-field formulation, a combination of continuous and enriched Galerkin function space, requires the fewest DOF among the methods that conserve local mass. Moreover, our results illustrate that three out of the five methods conserve local mass and produce similar flux approximations when conductivity alteration is included. These comparisons of the key performance indicators of different combinations of finite ele-

✉ T. Kadeethum
teekad@dtu.dk; tk658@cornell.edu

S. Lee
lee@math.fsu.edu

H. M. Nick
hamid@dtu.dk

[1] Technical University of Denmark, Lyngby, Denmark

[2] Cornell University, Ithaca, NY, USA

[3] Florida State University, Tallahassee, FL, USA

Published online: 01 October 2020

ment methods can be utilized to choose the preferred method based on the required accuracy and the available computational resources.

Keywords Poroelasticity · Biot's equations · Local mass conservative · Finite element · Two-field formulation · Three-field formulation

1 Introduction

Volumetric deformation of a porous medium caused by fluid pressure changes in pore spaces may impact the hydraulic storability and permeability of a porous material, which in turn influences the fluid flow field. This deformation mechanism is the coupled process between fluid flow and solid deformation, and it can be captured through the application of Biot's equations of poroelasticity (Biot 1941; Biot and Willis 1957). Biot's equations can be solved analytically for simple geometries (Terzaghi 1951; Wang 2017), but for complex geometries and boundary conditions, numerical approximations such as finite volume discretization (Nordbotten 2014; Sokolova et al. 2019; Honorio et al. 2018) and finite element methods (Choo and Lee 2018; Deng et al. 2017; Haga et al. 2012; Liu et al. 2018; Murad et al. 2013; Salimzadeh et al. 2018; Wheeler et al. 2014; Bouklas et al. 2015a) are commonly used. Recently, the possibility of solving Biot's equations using deep learning techniques was presented in Kadeethum et al. (2020a, b). Biot's equations are relevant for many applications, including land subsidence by groundwater overexploitation, enhanced geothermal system and fossil fuel production in deep and tight reservoirs, hydraulic fracturing-induced seismicity, and biomedical engineering using porous hyperelastic material (Bisdom et al. 2016; Kadeethum et al. 2019b, 2020d; Juanes et al. 2016; Lee et al. 2016; Nick et al. 2013; Vinje et al. 2018; Bouklas et al. 2015b).

Specifically for the finite element method, Biot's equations can generally be discretized using the two-field or three-field formulations (Haga et al. 2012; Phillips and Wheeler 2007a, b). The two-field formulation uses fluid pressure and bulk deformation as the primary variables, while the three-field formulation adds fluid velocity to the set of primary variables. For problems concerning incompressible material or large strains, a four- or five-field formulation can be implemented (Kumar et al. 2020; Zdunek et al. 2016). In this paper, we limit the scope of our study to two- and three-field formulations.

Each of the formulations mentioned above has its advantages and disadvantages. For example, the two-field formulation, which employs a stable pair of continuous function spaces such as Taylor–Hood elements (Arnold et al. 1984) in its classical form, is unable to provide solutions with local mass conservation or that are free of fluid pressure oscillation at material interfaces with high permeability contrast (Murad and Loula 1994; Scovazzi et al. 2017; Vermeer and Verruijt 1981). The three-field formulation, which explicitly approximates fluid velocity, generally requires more computational resources than the two-field formulation, especially in a three-dimensional domain (Choo and Lee 2018; Kadeethum et al. 2019a, 2020c). Moreover, the three-field formulation involves the inversion of the permeability tensor, leading to an ill-posed problem (Choo and Lee 2018).

There are many ways to compose two- and three-field formulations. For the two-field formulation, both solid displacement and fluid pressure variables can be approximated by the continuous Galerkin (CG) method (Deng et al. 2017; Salimzadeh et al. 2018; Salimzadeh and Nick 2019). For a higher-order approximation, one may employ a CG finite element space for the displacement field and a discontinuous Galerkin (DG) or an enriched Galerkin (EG) finite element space for the fluid pressure variable (Choo and Lee 2018; Kadeethum et al. 2019a; Liu et al. 2009). For three-field formulation, one can use a mixed space of CG, the lowest Raviart–Thomas (RT), and the lowest DG finite element spaces for displacement, velocity, and fluid pressure fields, respectively (Haga et al. 2012; Phillips and Wheeler 2007a).

In this paper, we aim to provide quantitative comparisons of two- and three-field formulations for solving linear and nonlinear Biot's equations. The flux approximation, local mass conservation, number of Newton iterations, and required computational time are used as comparison parameters. For the sake of simplicity, we solve Biot's equations using a monolithic approach instead of a sequential method (White and Borja 2011), and no preconditioner is applied (White et al. 2016). A direct solver with lower–upper decomposition (LU) is used inside each Newton iteration to reduce elimination steps in the Gauss elimination (Balay et al. 2018). We note that this approach may not yield the optimum results with regard to computational resource requirements for each method. Advanced developments targeting each specific approach could result in better computational time and memory management (Adler et al. 2019; Ayuso et al. 2009; Lee and Wheeler 2018; White et al. 2016). An iterative scheme such as a fixed-stress algorithm can also speed up the process and reduce the computational cost significantly (Almani et al. 2016; Castelletto et al. 2015; Girault et al. 2019; Kim et al. 2011).

We begin with model descriptions, mathematical equations, and model discretizations for both two- and three-field formulations. Subsequently, the solution strategy is illustrated, including block structure and Newton iterations. Numerical results from two- and three-field formulations are used to compare local mass conservation, flux approximation, and computer processing time. We then provide guidelines for which methods to use with respect to different model descriptions.

2 Mathematical Model

2.1 Governing Equations

We briefly recapitulate Biot's equations for poroelasticity used in this paper. Let $\Omega \subset \mathbb{R}^d$ $(d \in \{1, 2, 3\})$ be the computational domain, which is bounded by $\partial\Omega$. The time (t) domain is denoted by $\mathbb{T} = (0, T]$, with $T > 0$. The coupling between the fluid flow and solid deformation can be captured through Biot's equations of poroelasticity, which are the coupling between linear momentum and mass balance equations (Biot 1941).

The linear momentum balance equation can be written as

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) = \boldsymbol{f}, \tag{1}$$

where $p(\cdot, t) : \Omega \times (0; T] \rightarrow \mathbb{R}$ is a scalar-valued fluid pressure, $\boldsymbol{u}(\cdot, t) : \Omega \times (0; T] \rightarrow \mathbb{R}^d$ is a vector-valued displacement, and $\boldsymbol{f}$ is a body force. For simplicity, $\boldsymbol{f}$ is neglected in this study. Here, $\boldsymbol{\sigma}$ is total stress, which is defined as follows

$$\boldsymbol{\sigma} := \boldsymbol{\sigma}' - \alpha p \boldsymbol{I}, \tag{2}$$

where $\boldsymbol{I}$ is the identity tensor and α is Biot's coefficient, expressed as (Jaeger et al. 2009)

$$\alpha := 1 - \frac{K}{K_s}, \tag{3}$$

with the bulk modulus of a rock matrix K and the solid grains modulus K_s. In addition, $\boldsymbol{\sigma}'$ is an effective stress

$$\boldsymbol{\sigma}'(\boldsymbol{u}) = \lambda_l(\boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{I} + 2\mu_l \boldsymbol{\varepsilon}(\boldsymbol{u}). \tag{4}$$

where λ_l and μ_l are Lamé constants, and $\boldsymbol{\varepsilon}$ is the strain assuming infinitesimal displacement

$$\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}(\boldsymbol{u}) := \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathsf{T}}\right). \tag{5}$$

Here, $\partial\Omega$ can be decomposed to displacement and traction boundaries, $\partial\Omega_u$ and $\partial\Omega_t$, respectively. Linear momentum balance is supplemented by its boundary and initial conditions, as shown below

$$\begin{aligned}
\nabla \cdot \boldsymbol{\sigma}'(\boldsymbol{u}) - \alpha \nabla \cdot p\boldsymbol{I} &= \boldsymbol{f} \quad \text{in } \Omega \times \mathbb{T}, && (6)\\
\boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega_u \times \mathbb{T}, && (7)\\
\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{n} &= \boldsymbol{t}_D \quad \text{on } \partial\Omega_t \times \mathbb{T}, && (8)\\
\boldsymbol{u} &= \boldsymbol{u}_0 \quad \text{in } \Omega \text{ at } t = 0, && (9)
\end{aligned}$$

where $\boldsymbol{u}_D$ and $\boldsymbol{\sigma}_D$ are the prescribed displacement and traction quantities at boundaries, respectively.

Next, the mass balance equation is given as (Coussy 2004)

$$\rho\left(\phi c_f + \frac{\alpha - \phi}{K_s}\right)\frac{\partial}{\partial t} p + \rho\alpha \frac{\partial}{\partial t} \nabla \cdot \boldsymbol{u} + \nabla \cdot (\rho \boldsymbol{v}) = g \text{ in } \Omega \times \mathbb{T}. \tag{10}$$

Here, ρ is fluid density, ϕ is initial porosity, c_f is fluid compressibility, $\nabla \cdot \boldsymbol{u}$ is the volumetric deformation, and g is a sink/source. Velocity $\boldsymbol{v}$ is defined by Darcy's velocity

$$\boldsymbol{v} := -\frac{\mathscr{N}[\boldsymbol{\kappa}]}{\rho}(\nabla p - \rho \boldsymbol{g}) \text{ in } \Omega \times \mathbb{T}, \tag{11}$$

where $\boldsymbol{g}$ is a gravitational vector, $\mathscr{N}[\cdot]$ represents a nonlinear operator, and $\boldsymbol{\kappa}$ is

$$\boldsymbol{\kappa} := \frac{\rho \boldsymbol{k}_m}{\mu}, \tag{12}$$

where μ represents fluid viscosity and $\boldsymbol{k}_m$ is a matrix permeability tensor defined as

$$\boldsymbol{k}_m = \begin{bmatrix} k_m^{xx} & k_m^{xy} & k_m^{xz} \\ k_m^{yx} & k_m^{yy} & k_m^{yz} \\ k_m^{zx} & k_m^{zy} & k_m^{zz} \end{bmatrix}, \tag{13}$$

and k^{xx}, k^{yy}, and k^{zz} represent matrix permeability in the x-, y-, and z-directions, respectively. In this study, we are dealing with two types of a system of equations using different definitions of $\mathscr{N}[\boldsymbol{\kappa}]$ as follows

$$\mathscr{N}[\boldsymbol{\kappa}] = \begin{cases} = \boldsymbol{\kappa}, & \text{for a system of linear equations} \\ = \boldsymbol{\kappa}(\boldsymbol{u}), & \text{for a system of nonlinear equations.} \end{cases} \tag{14}$$

The boundary of mass balance equation $\partial \Omega$ is decomposed to pressure and flux boundaries, $\partial \Omega_p$ and $\partial \Omega_q$, respectively. The mass balance equation is then supplemented by the following boundary and initial conditions

$$p = p_D \text{ on } \partial \Omega_p \times \mathbb{T}, \tag{15}$$

$$\rho \boldsymbol{v} \cdot \boldsymbol{n} = q_D \text{ on } \partial \Omega_q \times \mathbb{T}, \tag{16}$$

$$p = p_0 \text{ in } \Omega \text{ at } t = 0, \tag{17}$$

where p_D and q_D are the given pressure and flux, respectively.

Remark 1 The nonlinearity arising in Biot's equations studied here is focused only in the matrix permeability tensor, which is expressed as a function of the displacement vector, see Eq. (14). This form of nonlinearity is not the only possible formulation. The elastoplastic or visco-elastoplastic constitutive laws could also be introduced and add sources of nonlinearity (Cao et al. 2016; Choo 2018; Macminn et al. 2016; Lewis and Schrefler 1998).

2.2 Numerical Methods

Let $\mathscr{T}_h$ be the shape-regular triangulation by a family of partitions of Ω into d-simplices T (triangles/squares in $d = 2$ or tetrahedral/hexahedra in $d = 3$). We let h_T be the diameter of T, and we set $h = \max_{T \in \mathscr{T}_h} h_T$. Also, we denote $\mathscr{E}_h$ to be the set of all facets. Then let $\mathscr{E}_h^I$ and $\mathscr{E}_h^\partial$ be the collection of all interior and boundary facets, respectively. The $\mathscr{E}_h^\partial$ is decomposed into $\mathscr{E}_h^D$, the Dirichlet boundary faces, and $\mathscr{E}_h^N$, the Neumann boundary faces. The space $H^s(\mathscr{T}_h)$ ($s \in \mathbb{R}$) is the set of element-wise H^s functions on $\mathscr{T}_h$, and $L^2(\mathscr{E}_h)$ refers to the set of functions whose traces on the elements of $\mathscr{E}_h$ are square integrable. Let $\mathbb{Q}_k(T)$ represent the space of polynomials of partial degree at most k.

Next, we briefly introduce several notations. Let

$$e = \partial T^+ \cap \partial T^-, \quad e \in \mathscr{E}_h, \tag{18}$$

where T^+ and T^- are two neighboring elements. We denote h_e by the length of the facet e, and $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ by the outward normal unit vectors to ∂T^+ and ∂T^-, respectively ($\boldsymbol{n}^\pm := \boldsymbol{n}_{|T^\pm}$). For any given function ξ and vector function $\boldsymbol{\xi}$, defined on the triangulation $\mathscr{T}_h$, let $\xi^\pm$ and $\boldsymbol{\xi}^\pm$ be the restrictions of ξ and $\boldsymbol{\xi}$ to $T^\pm$, respectively. Next, we define the weighted average operator $\{\cdot\}_{\delta_e}$ for $\zeta \in L^2(\mathscr{T}_h)$ and $\boldsymbol{\tau} \in L^2(\mathscr{T}_h)^d$, where $d \geq 2$

$$\{\zeta\}_{\delta e} = \delta_e \zeta^+ + (1 - \delta_e)\,\zeta^-, \quad \text{on } e \in \mathscr{E}_h, \tag{19}$$

and

$$\{\boldsymbol{\tau}\}_{\delta e} = \delta_e \boldsymbol{\tau}^+ + (1 - \delta_e)\,\boldsymbol{\tau}^-, \quad \text{on } e \in \mathscr{E}_h, \tag{20}$$

where δ_e is calculated by Ern et al. (2009), Ern and Stephansen (2008) and Kadeethum et al. (2019a)

$$\delta_e := \frac{\kappa_e^-}{\kappa_e^+ + \kappa_e^-}. \tag{21}$$

Here

$$\kappa_e^+ := \left(\boldsymbol{n}^+\right)^\intercal \cdot \boldsymbol{\kappa}^+ \cdot \boldsymbol{n}^+, \quad \text{and} \quad \kappa_e^- := \left(\boldsymbol{n}^-\right)^\intercal \cdot \boldsymbol{\kappa}^- \cdot \boldsymbol{n}^-, \tag{22}$$

where κ_e is a harmonic average of κ_e^+ and κ_e^- expressed as

$$\kappa_e := \frac{2\kappa_e^+ \kappa_e^-}{\left(\kappa_e^+ + \kappa_e^-\right)}. \tag{23}$$

On the other hand, for $e \in \mathscr{E}_h^\partial$, we set $\{\zeta\}_{\delta_e} := \zeta$ and $\{\boldsymbol{\tau}\}_{\delta_e} := \boldsymbol{\tau}$. Jumps across interior facets will be defined as

$$[\![\zeta]\!] = \zeta^+ \boldsymbol{n}^+ + \zeta^- \boldsymbol{n}^- \quad \text{and} \quad [\![\boldsymbol{\tau}]\!] = \boldsymbol{\tau}^+ \cdot \boldsymbol{n}^+ + \boldsymbol{\tau}^- \cdot \boldsymbol{n}^- \quad \text{on } e \in \mathscr{E}_h.$$

For $e \in \mathscr{E}_h^\partial$, we let $[\![\zeta]\!] := \zeta \boldsymbol{n}$ and $[\![\boldsymbol{\tau}]\!] := \boldsymbol{\tau} \cdot \boldsymbol{n}$.

2.2.1 Finite Element Spaces

We employ five different finite element methods for the spatial discretization and compare the performance for the given problem. The first three methods we consider

Table 1 Two-field ($\boldsymbol{u}\times p$) and three-field ($\boldsymbol{u}\times\boldsymbol{v}\times p$) formulations presented with their choices of function space

Formulation	Mixed space	Designated name	Displacement	Velocity	Pressure
$\boldsymbol{u}\times p$	$\mathrm{CG}_2 \times \mathrm{CG}_1$	CG	CG_2	–	CG_1
	$\mathrm{CG}_2 \times \mathrm{EG}_1$	EG	CG_2	–	EG_1
	$\mathrm{CG}_2 \times \mathrm{DG}_1$	DG	CG_2	–	DG_1
$\boldsymbol{u}\times\boldsymbol{v}\times p$	$\mathrm{CG}_2 \times \mathrm{RT}_1 \times \mathrm{DG}_0$	MFE-RT	CG_2	RT_1	DG_0
	$\mathrm{CG}_2 \times \mathrm{CG}_2 \times \mathrm{CG}_1$	MFE-P2	CG_2	CG_2	CG_1

$\mathrm{CG_k}$, $\mathrm{EG_k}$, $\mathrm{DG_k}$, and $\mathrm{RT_k}$ refer to continuous Galerkin, enriched Galerkin, discontinuous Galerkin, and Raviart–Thomas with polynomial degree k

are continuous Galerkin (CG) (Murad and Loula 1994; Deng et al. 2017), discontinuous Galerkin (DG) (Liu et al. 2009; Riviere et al. 2017), and enriched Galerkin (EG) (Lee and Wheeler 2018; Choo and Lee 2018; Kadeethum et al. 2019a; Lee and Wheeler 2017; Lee and Woocheol 2019) methods, where the primary unknown variables are ($\boldsymbol{u}\times p$), i.e the displacement $\boldsymbol{u}(\cdot, t)$ and the pressure $p(\cdot, t)$. These are known as two-field formulations.

The next two methods are so-called three-field formulations, where the primary unknown variables are ($\boldsymbol{u}\times\boldsymbol{v}\times p$), i.e the displacement $\boldsymbol{u}(\cdot, t)$, the velocity $\boldsymbol{v}(\cdot, t)$, and the pressure $p(\cdot, t)$. For this case, we consider the Raviart–Thomas (RT) mixed finite element method (Pain et al. 2005; Castelletto et al. 2016; Phillips and Wheeler 2007a, b) and continuous mixed finite element method (Haga et al. 2012). The summary of each method and its designated name for simplicity of presentation is presented in Table 1.

We also compare the number of unknowns, or degrees of freedom (DOF), among these methods by setting k as shown in Table 1. Please see Fig. 1(a, b) for an illustration of two- and three-dimensional cases. The CG method has the lowest number of DOF, while MFE-P2 has the greatest. The DG and the MFE-RT methods have almost the same number of DOF, and the EG method has fewer DOF than the DG and the MFE-RT methods.

2.2.2 Temporal Discretization

The time domain, $\mathbb{T} = (0, \tau]$, is partitioned into N open intervals such that $0 =: t^0 < t^1 < \cdots < t^N := \tau$. The length of the interval, Δt^n, is defined as $\Delta t^n = t^n - t^{n-1}$, where n represents the current time step. Δt^0 is an initial Δt, which is defined as $t^1 - t^0$, while the other time steps, Δt^n, are calculated as follows

$$\Delta t^n := \begin{cases} \Delta t_{\mathrm{mult}} \times \Delta t^{n-1} & \text{if } \Delta t^n \leq \Delta t_{\max} \\ \Delta t_{\max} & \text{if } \Delta t^n > \Delta t_{\max}, \end{cases} \tag{24}$$

where Δt_{mult} is a positive constant multiplier, and $\Delta t_{\max}$ is a maximum of Δt. In this study, implicit first-order time discretization is utilized for a time domain as shown

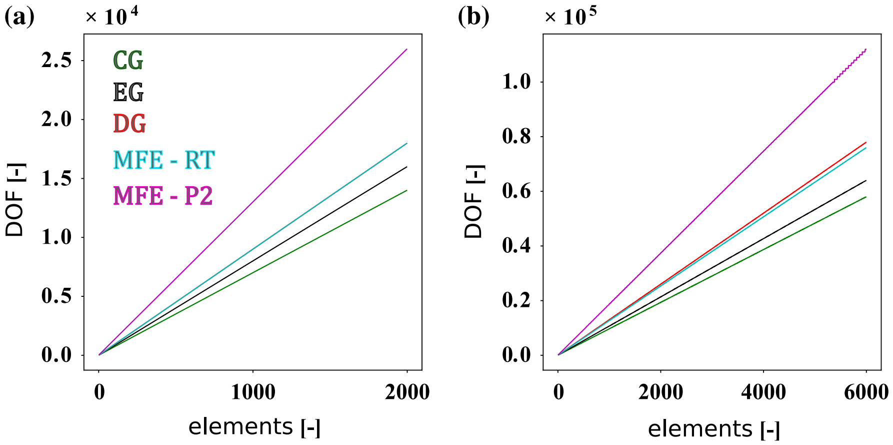

Fig. 1 Comparison of the DOF of the methods presented in Table 1 of (**a**) two-dimensional geometry, (**b**) three-dimensional geometry. Note that the DG and MFE-RT results overlap each other in (**a** and **b**)

below for a given function $\mathbb{X}(\cdot, t)$ at time t^n

$$\frac{\partial \mathbb{X}(\cdot, t^n)}{\partial t} \approx \frac{\mathbb{X}^n - \mathbb{X}^{n-1}}{\Delta t^n}. \tag{25}$$

The fully discretized formulations of each method in Table 1 are presented in the appendices. Please note that to simplify the process used in this paper, we solve Biot's equations using a monolithic method, and only a direct solver is employed without any preconditioner (Balay et al. 2018; White and Borja 2011; White et al. 2016). We note that in recent years, iterative solvers have been introduced that can exploit the block structure of the discrete problem and its intrinsic algebraic properties, which lead to very efficient strategies (Castelletto et al. 2016; White et al. 2016; Ferronato et al. 2010; Lee et al. 2017; Frigo et al. 2019; Ferronato et al. 2019). Additionally, each of the methods presented in Table 1 can be optimized; for example, there is a fast solver technique available for the EG method (Lee and Wheeler 2018). A decomposition of the linear DG space into a Crouzeix–Raviart (C–R) space and a complementary space can be used as a block preconditioner for the DG method (Lee and Wheeler 2018; Ayuso et al. 2009). New robust solvers and preconditioners for three-field formulations have also recently been proposed (Adler et al. 2019; Frigo et al. 2019). Moreover, a fixed-point iteration approach, such as the fixed-stress split, can be employed instead of the monolithic approach in order to speed up the process (Almani et al. 2016; Girault et al. 2019; Castelletto et al. 2015; Kim et al. 2011).

3 Numerical Examples

We compare four factors among the five methods listed in Table 1. The flux approximation, local mass conservation property, DOF, and required computational time are the comparison factors. We begin with a simple one-dimensional linear consolidation problem in homogeneous and heterogeneous porous materials. We then present two- and three-dimensional nonlinear Biot's systems in homogeneous and heterogeneous materials.

Throughout this paper, matrices and vectors are built using the FEniCS form compiler (Alnaes et al. 2015). The block structure is assembled using the multiphenics toolbox (Ballarin and Rozza 2019). Solvers are employed from the PETSc package (Balay et al. 2018). The random fields porosity (ϕ) and permeability ($\boldsymbol{k}_{m0}$) are populated using the SciPy package (Virtanen et al. 2020). The penalty parameter (β) is set at 1.1 and 1.0 for the DG and EG methods, respectively.

3.1 A Consolidation Benchmark Problem with Homogeneous Material

We verify the numerical methods CG, EG, DG, MFE-RT, and MFE-P2 using an analytical solution known as Terzaghi's one-dimensional consolidation problem (Terzaghi 1951). Examples 1 and 2 are similar to examples employed by Choo and Lee (2018) and Kadeethum et al. (2019a), but here we also include MEF-RT and MFE-P2, the mixed finite element methods, for validation and comparison. We consider the linear case with $\mathscr{N}[\boldsymbol{\kappa}] = \boldsymbol{\kappa}$ and assume the domain is homogeneous, isotropic, and saturated with a single-phase fluid. The computational domain and boundary conditions are presented in Fig. 2.

The nondimensional parameters are defined as follows (Terzaghi 1951; Choo and Lee 2018; Kadeethum et al. 2019a)

$$p^* := \frac{p}{\sigma},\ z^* := \frac{z}{H},\ t^* := \frac{c_v}{H^2} t, \tag{26}$$

where p is the fluid pressure, σ is the far-field stress or external load, z is the distance from the drainage boundary, and H is the domain thickness. In addition, c_v is the coefficient of consolidation calculated by

$$c_v = 3K \left(\frac{1-v}{1+v} \right) \frac{k_{m,y}}{\mu}, \tag{27}$$

where $k_{m,y}$ is the matrix permeability in the y-direction. Subsequently, the analytical pressure solution of Terzaghi's one-dimensional consolidation is written as follows

$$p^* \left(z^*, t^* \right) = \sum_{m=0}^{\infty} \frac{2}{M} \sin \left(M z^* \right) \mathrm{e}^{-M^2 t^*}, \tag{28}$$

where $M = \pi(2m+1)/2$. Input parameters are $\boldsymbol{\sigma}_D = [0, 1]$ kPa, $p_D = 0$ Pa, $\boldsymbol{k_m} = 10^{-12} \boldsymbol{I}$ m^2, $\mu = 10^{-6}$ kPa s, $\rho = 1{,}000$ kg/m^3, $K = 1{,}000$ kPa, mesh size$(h) = 0.05\ m$, $K_s \approx \infty$ kPa, which leads to $\alpha \approx 1$, and $v = 0.25$, $\Delta t^0 = 1.0$ s, $\Delta t_{\text{mult}} = 1.0$, $\Delta t_{\max} = 1$; λ_l and μ_l are calculated by the following equations

$$\lambda_l = \frac{3Kv}{1+v}, \quad \text{and} \quad \mu_l = \frac{3K(1-2v)}{2(1+v)}. \tag{29}$$

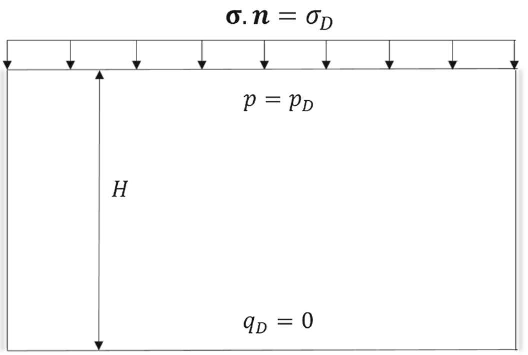

Fig. 2 Example 3.1: A setup for the one-dimensional consolidation problem in a homogeneous material

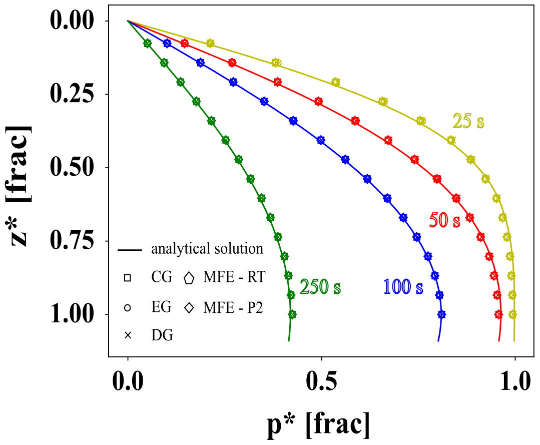

Fig. 3 Example 3.1: The pressure values for the one-dimensional consolidation problem in a homogeneous material. The results from all CG, EG, DG, MFE-RT, and MFE-P2 methods are illustrated. The new results for the mixed methods (MFE-RT, MFE-P2) are similar to CG, DG, and EG, which were also shown in Choo and Lee (2018) and Kadeethum et al. (2019a)

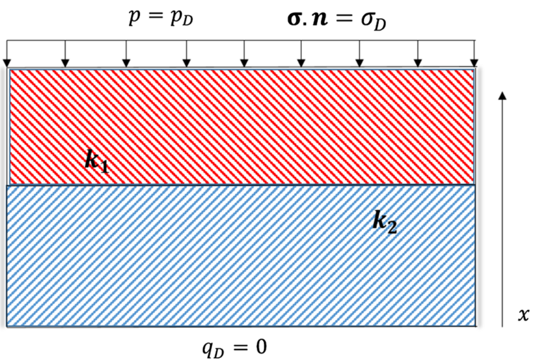

Fig. 4 Example 3.2: A setup for the one-dimensional consolidation problem in a two-layered material

We use the direct solver with lower–upper decomposition (LU) in this problem (Balay et al. 2018). The results of the five different methods are presented in Fig. 3. We observe reasonable matches between analytical and numerical solutions for all time steps, 25, 50, 100, and 250 s. Moreover, the solutions of CG, EG, DG, MFE-RT, and MFE-P2 methods are approximately identical for this problem.

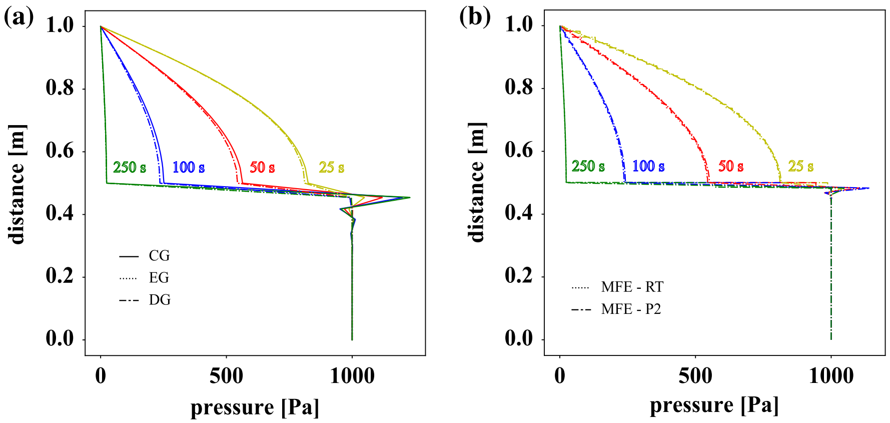

Fig. 5 Example 3.2: The pressure values for the one-dimensional consolidation problem in a two-layered material is illustrated. (**a**) presents the pressure results of the CG, EG, and DG methods, (**b**) shows the pressure results of the MFE-RT and MFE-P2 methods. The new results for the mixed methods (MFE-RT, MFE-P2) are similar to CG, DG, and EG, which were also shown in Choo and Lee (2018); Kadeethum et al. (2019a)

3.2 A Consolidation Problem with Two-Layered Material

Next, we evaluate the fluid pressure, p, in a Two-layered material (Choo and Lee 2018; Kadeethum et al. 2019a). Figure 4 presents a geometry and boundary conditions utilized in this problem. Most of the input parameters, mesh size, and solvers are the same as in the previous example, but in this example, $\boldsymbol{k}_m = k_1\boldsymbol{I}$, if $1 > x > 0.5$ and $\boldsymbol{k}_m = k_2\boldsymbol{I}$, if $0.5 > x > 0$, and $k_1 = 10^{-12}$ m^2 and $k_2 = 10^{-16}$ m^2.

The pressure values, p, are presented in Fig. 5(a, b) for the CG, EG, and DG methods and the MFE-RT and MFE-P2 methods, respectively. In this heterogeneous material, we observe unphysical (spurious) pressure oscillations at the interface when the CG and MFE-P2 methods are used, because it is well known that the classical CG method does not conserve mass locally (Haga et al. 2012; Choo and Lee 2018; Kadeethum et al. 2019a; Choo 2019; Choo and Borja 2015; Hong and Kraus 2017; Honorio et al. 2018; Lee et al. 2017; Rodrigo et al. 2018; White and Borja 2008).This oscillation may lead to incorrect fluid flux calculation and incorrect permeability alteration, since $\boldsymbol{\kappa}$ is a function of $\boldsymbol{u}$ as described in Eq. (14), and $\boldsymbol{u}$ is strongly coupled with p. Hence, suitable modifications are required to overcome this issue. On the other hand, we do not observe any oscillations for the EG, DG, or MFE-RT methods. The pressure solutions of these methods are smooth for all the time steps.

3.3 A Flow Problem in Two-Dimensional Deformable Media

In this example, we consider the system of nonlinear equations by assuming $\mathscr{N}[\boldsymbol{\kappa}] = \boldsymbol{\kappa}(\boldsymbol{u})$ in the computational domain $\Omega = [0, 1]^2$. For the nonlinear case, the volumetric displacement can cause the matrix permeability alteration, and $\boldsymbol{k}_m$ is defined as Du and Wong (2007) and Abou-Kassem et al. (2013)

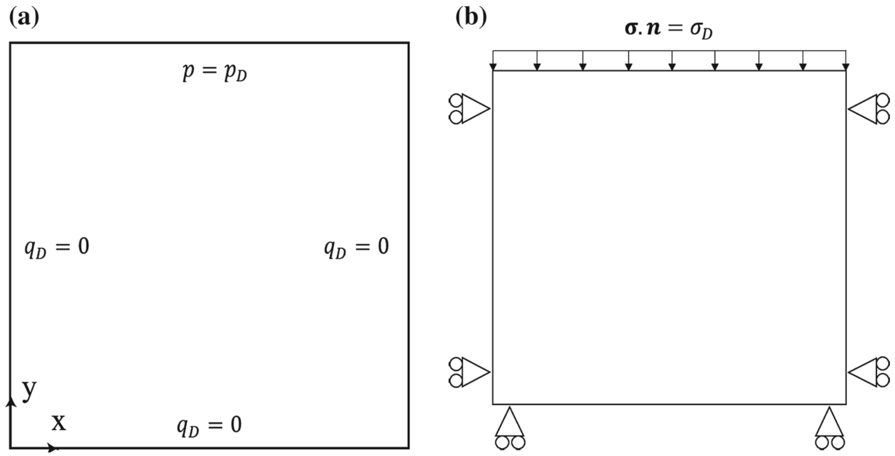

Fig. 6 Example 3.3: A setup for the flow problem in a two-dimensional deformable media. (**a**) Mass boundary conditions and (**b**) force boundary conditions are illustrated

Table 2 Example 3.3: Degrees of freedom (DOF) comparison among CG, EG, DG, MFE-RT, and MFE-P2 methods

Model	Total	Displacement	Velocity	Pressure
CG	19,582	17,370	–	2,212
EG	23,844	17,370	–	6,474
DG	30,156	17,370	–	12,786
MFE-RT	28,105	17,370	6,473	4,262
MFE-P2	36,952	17,370	17,370	2,212

$$\boldsymbol{k}_m = \boldsymbol{k}_m{}^0 \frac{\left(1 + \frac{\varepsilon_v}{\phi}\right)^3}{1 + \varepsilon_v}, \tag{30}$$

where $\boldsymbol{k}_m^0$ represents the initial rock matrix permeability. Here, ε_v is the total volumetric strain

$$\varepsilon_v := \mathrm{tr}(\boldsymbol{\varepsilon}) = \sum_{i=1}^{d} \varepsilon_{ii}. \tag{31}$$

Throughout this problem, we assume that ρ and μ are constants; thus, $\boldsymbol{\kappa}$ varies only as a function of $\boldsymbol{k_m}$. The details for the setup and the boundary conditions are shown in Fig. 6.

The physical parameters are given as $\boldsymbol{\sigma}_D = [0, 20]$ MPa, $p_D = 1$ MPa, $p_0 = 10$ MPa, $\alpha = 0.79$, $K = 8.0$ GPa, K_s is then calculated by Eq. (3), $c_f = 1 \times 10^{-10}$ Pa, $\rho = 1{,}000\,\mathrm{kg/m}^3$, $\mu = 10^{-6}$ kPa s, $h = 0.025$m, and $v = 0.2$. The rock properties are selected for a sandstone example (Jaeger et al. 2009). In addition, λ_l, and μ_l are calculated by Eq. (29). Finally, the numerical parameters are set as $\Delta t^0 = 1.0$ s,

$\Delta t_{\text{mult}} = 1.5$, $\Delta t_{\text{max}} = 2.5$ s. We have two different cases of the ϕ and $\boldsymbol{k_m}$ for this example.

This example is used to compare the flux approximation, property of local mass conservation, and the number of Newton iterations. First, the number of DOF for five different methods is presented in Table 2. As we discussed before, we observe that the CG method has the fewest DOF. The DG and MFE-RT methods have approximately the same number of DOF, while MFE-P2 has the highest number of DOF.

The local mass conservation of each cell at each time step, $\mathrm{r}^{\mathrm{n}}_{\text{mass}}$, is calculated by

$$\begin{aligned} {\mathrm{r}^{\mathrm{n}}}_{\text{mass}} := \int_T \rho \left(\phi c_f + \frac{\alpha - \phi}{K_s} \right) \frac{p^n - p^{n-1}}{\Delta t^n} \\ + \rho \alpha \nabla \cdot \frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t^n} dV + \sum_{e \in \mathscr{E}_h} \int_e \boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}} \, dS, \end{aligned} \tag{32}$$

and the numerical flux, $\boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}}$, is defined by

$$\boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}} = -\left\{ \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \boldsymbol{g} \right) \cdot \boldsymbol{n} \right\} + \frac{\beta}{h_e} \boldsymbol{\kappa}(\boldsymbol{u}_h^n) [\![p_h^n]\!] \quad \forall e \in \mathscr{E}_h^I, \tag{33}$$

$$\boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}} = q_D \quad \forall e \in \mathscr{E}_h^N, \tag{34}$$

$$\boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}} = -\boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \boldsymbol{g} \right) \cdot \boldsymbol{n} + \frac{\beta}{h_e} \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(p_h^n - p_D \right) \quad \forall e \in \mathscr{E}_h^D. \tag{35}$$

For the comparison, we evaluate the maximum value of ${\mathrm{r}^{\mathrm{n}}}_{\text{mass}}$ at each time step, $\max(r_{\text{mass}})$. For the flux approximation, we use a recovery factor (RF), which is an industry standard (Peaceman 2000; Abou-Kassem et al. 2013; Chen 2007), and it is defined at each time step as follows

$$\mathrm{RF}^n := \frac{\sum_{t=0}^{t^n} \sum_{e \in \mathscr{E}_h} \int_e \boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}} \, dS}{\rho V_0 \phi}, \tag{36}$$

where V_0 is an initial reservoir volume. In this example, $\boldsymbol{v}^n \cdot \boldsymbol{n}|_{\mathrm{e}}$ is calculated for all outlet surfaces.

Throughout this example, the Newton iteration is used to solve the system of nonlinear equations, and a direct solver with lower–upper decomposition (LU) is employed for each Newton iteration (Balay et al. 2018). The absolute tolerance is set as 1×10^{-6}, relative tolerance is set as 1×10^{-16}, the maximum number of Newton iterations is 100, and the relaxation parameter is 0.75. All simulations are run on a Xeon E5-2650 v4 processor with a single thread.

3.3.1 Homogeneous Rock Matrix Porosity and Permeability

In this case, we consider that ${\boldsymbol{k_m}}^0$ is isotropic and homogeneous, and the value is set to ${\boldsymbol{k}_m}^0 = 10^{-12} \boldsymbol{I}$. Figure 7 illustrates the comparison of the RF, $\max(r_{\text{mass}})$, and number of Newton iterations for the different methods.

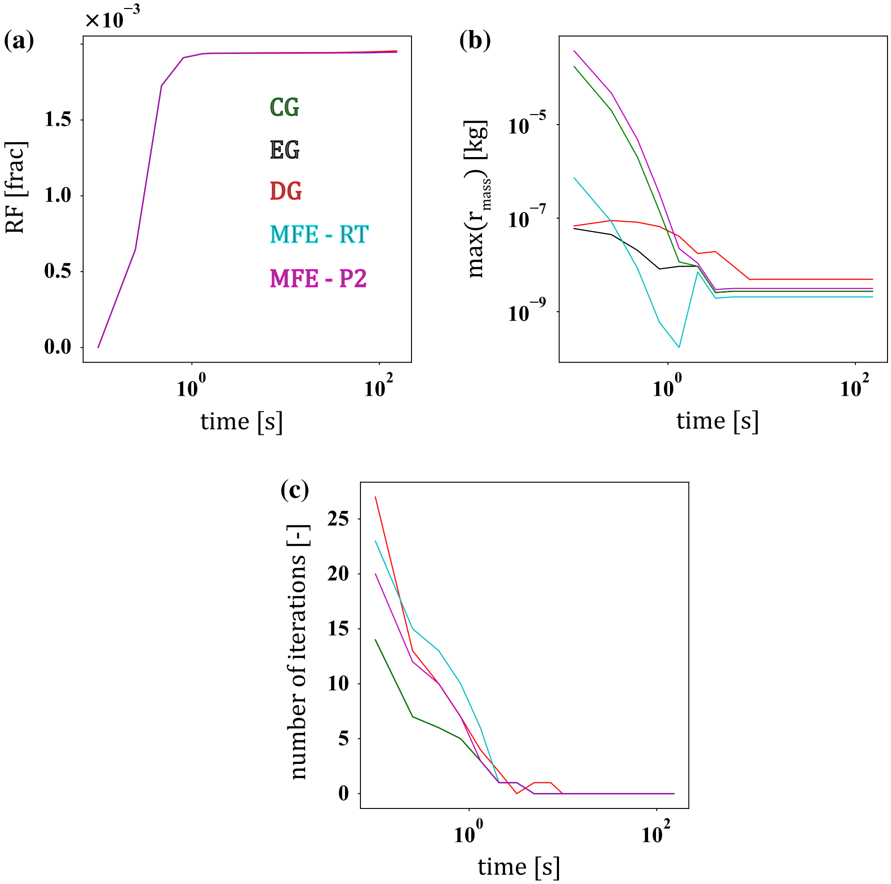

Fig. 7 Example 3.3.1: Comparison of the values of (**a**) RF, (**b**) $\max(r_{\text{mass}})$, and (**c**) number of Newton iterations among the CG (green line), EG (black line), DG (red line), MFE-RT (magenta line), and MFE-P2 (blue line) methods. Note that all results overlap each other in (**a**), and the CG and EG results overlap in (**c**)

First, we note that all the methods present approximately the same RF results. However, the value for $\max(r_{\text{mass}})$ shows different behavior for each method. The EG, DG, and MFE-RT methods conserve the local mass, with the value of $\max(r_{\text{mass}})$ always less than the absolute tolerance of 1×10^{-6}, whereas the CG and MFE-P2 methods do not conserve the local mass, and the $\max(r_{\text{mass}})$ value is not bounded.

Figure 7(c) shows that the CG and EG methods require the fewest Newton iterations, while the DG, MFE-RT, and MFE-P2 require approximately the same number of Newton iterations.

The summaries of total wall and CPU times are presented in Table 3 for all methods. This table shows that the CG method requires the least total wall and CPU time, while the MFE-RT method requires the most. This result indicates that the total computational time is not directly determined by the number of DOF presented in Table 2 and the number of Newton iterations shown in Fig. 7(c). Thus, more detailed information

Table 3 Example 3.3.1: Comparison of the total wall and CPU times

Model	Wall time (s)	CPU time (s)
CG	263	248
EG	280	264
DG	316	308
MFE-RT	389	382
MFE-P2	326	311

Table 4 Example 3.3.1: Comparison of the average wall time for each operation

Summary of timings	Average wall time (s)				
	CG	EG	DG	MFE-RT	MFE-P2
Assemble cells	6.28E−03	5.97E−03	6.31E−03	6.64E−03	7.34E−03
Assemble exterior facets	1.84E−04	1.79E−04	1.77E−04	1.66E−04	1.69E−04
Assemble interior facets	–	7.88E−03	7.87E−03	–	–
Assemble system	8.45E−02	9.18E−02	9.48E−02	1.30E−01	1.92E−01
Build sparsity	1.48E−03	1.26E−03	1.53E−03	1.36E−03	1.44E−03
Delete sparsity	1.51E−06	1.37E−06	1.40E−06	1.38E−06	1.45E−06
DirichletBC compute	5.25E−05	5.12E−05	5.27E−05	4.32E−05	5.78E−05
Init dof vector	4.64E−04	4.11E−04	4.58E−04	4.10E−04	4.85E−04
Init dofmap	7.50E−03	8.07E−03	8.63E−03	5.82E−03	8.75E−03
Init tensor	4.99E−04	4.49E−04	4.80E−04	4.97E−04	5.23E−04
LU solver	1.91E−01	2.33E−01	2.68E−01	6.06E−01	3.08E−01

regarding the required computational time used by each operation is presented in Table 4.

In Table 4, we observe that the LU solver and assemble system operations are the main contributors to the wall time. Note that for the CG, MFE-RT, and MFE-P2, there is no computational time for assembling the interior facets because there is no interior facet integral term in their discretizations. As expected, the more DOF the method has, the more time it needs to assemble the system. The two-field formulation, CG, EG, and DG methods, requires less time to solve a linear system than the three-field formulation, MFE-RT and MFE-P2 methods, because the two-field formulation has one fewer primary variable than the three-field formulation. Even though the assembly time of the MFE-P2 method is higher than that of the MFE-RT method, the LU solver operation takes twice as long for the MFE-RT method as that of the MFE-P2 method. This leads to the longer total wall and CPU time taken by the MFE-RT method.

3.3.2 Heterogeneous Rock Matrix Porosity and Permeability

In this example, the heterogeneous ϕ and $\boldsymbol{\kappa}^0$ values are considered as shown in Fig. 8(a, b), respectively. In particular, the heterogeneous fields are generated using the following specifications: $\bar{\phi} = 0.2$, $\mathrm{var}(\phi) = 0.01$, $\phi_{\mathrm{min}} = 0.001$, $\phi_{\mathrm{max}} = 0.4$,

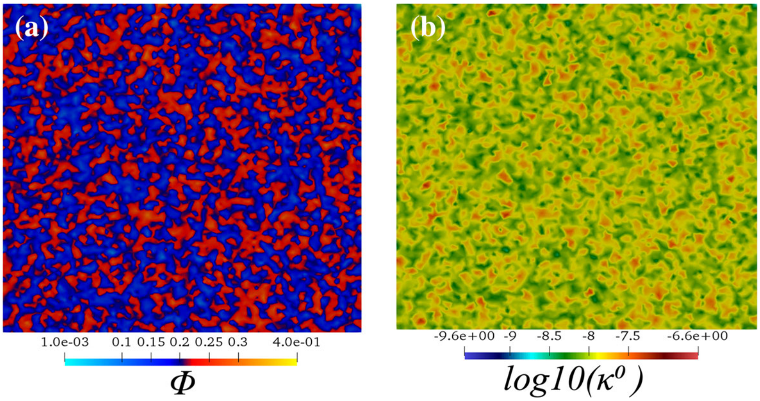

Fig. 8 Example 3.3.2: Heterogeneous rock matrix porosity and permeability values based on normal and log-normal distributions. (**a**) ϕ field and (**b**) log10(κ^0) field

$\bar{\kappa}^0 = 1.2\times10^8$ s, var(κ^0) $= 1.4\times10^{-16}$ s^2, ${\kappa_{\min}}^0 = 1.2\times10^{-13}$ s, and ${\kappa_{\max}}^0 = 1.2\times10^{-6}$ s. Here $\overline{(\cdot)}$ denotes an arithmetic average, var($\cdot$) is the variance, and $(\cdot)_{\min}$ and $(\cdot)_{\max}$ are the minimum and maximum values, respectively. The values $\bar{\phi}$, $\phi_{\min}$, $\phi_{\max}$, $\bar{\kappa}^0$, ${\kappa_{\min}}^0$, and ${\kappa_{\max}}^0$ are selected based on the reported values for shales and sandstones from Jaeger et al. (2009). Note that $\bar{\kappa}^0$, ${\kappa_{\min}}^0$, and ${\kappa_{\max}}^0$ reflect the $\bar{\boldsymbol{k}}_m{}^0$, ${\boldsymbol{k}_{m\,\min}}^0$, and ${\boldsymbol{k}_{m\,\max}}^0$ values of 1.2×10^{-14}, 1.2×10^{-19}, and 1.2×10^{-12} m^2, respectively. Additionally, these two heterogeneous fields are populated independently (i.e., there is no correlation between ϕ and κ^0). All the other input parameters are assigned the same values in the previous homogeneous example (Sect. 3.3.1).

In Fig. 9, the results of the RF, max(r_{mass}), and number of Newton iterations are illustrated. In this case, the EG, DG, and MFE-RT methods provide approximately the same RF result, but the CG and MFE-P2 results are significantly different. Similar to the homogeneous case, the max(r_{mass}) of the EG, DG, and MFE-RT methods is always less than the absolute tolerance of 1×10^{-6}. This shows that the EG, DG, and MFE-RT have the property of local mass conservation. The CG and MFE-P2 methods, on the other hand, do not conserve local mass. Figure 9(c) illustrates that the CG and EG methods require the lowest number of Newton iterations, and the two-field formulation requires fewer Newton iterations than the three-field formulation. The MFE-RT method has the highest number of Newton iterations.

The summary of the total wall and CPU times of the heterogeneous case for all methods is illustrated in Table 5. The result agrees with the previous case in Sect. 3.3.1. The CG method requires the least wall and CPU time, while the MFE-RT method requires the most wall and CPU time.

More detailed analysis of the computational time of each operation is summarized in Table 6. Similar to Sect. 3.3.1, the LU solver and assemble system operations take most of the wall time used by each simulation. The two-field formulation requires less time to solve a linear system than the three-field formulation. As expected, the more DOF the method has (Table 2), the more time it needs to assemble the system.

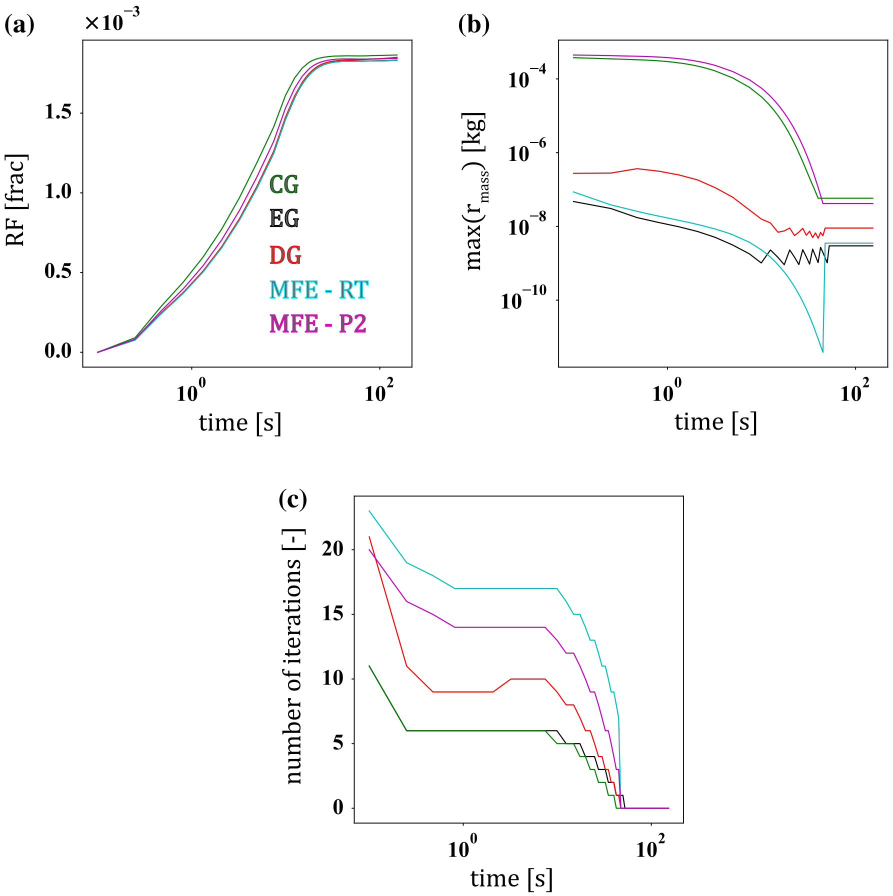

Fig. 9 Example 3.3.2: Comparison of the values of (**a**) RF, (**b**) $\max(r_{\text{mass}})$, and (**c**) number of Newton iterations among the CG (green line), EG (black line), DG (red line), MFE-RT (magenta line), and MFE-P2 (blue line) methods. Note that the EG, DG, and MFE-RT results overlap each other in **a**

Table 5 Example 3.3.2: Comparison of the total wall and CPU times

Model	Wall time (s)	CPU time (s)
CG	357	344
EG	455	442
DG	543	524
MFE-RT	1,221	1,201
MFE-P2	792	773

Moreover, as mentioned in the homogeneous case, the LU solver operation of the MFE-RT method takes twice as much time as the MFE-P2. This leads to the longer total wall and CPU times of the MFE-RT method.

Table 6 Example 3.3.2: Comparison of the average wall time for each operation

Summary of timings	Average wall time (s)				
	CG	EG	DG	MFE-RT	MFE-P2
Assemble cells	6.25E−03	5.95E−03	6.35E−03	6.44E−03	7.44E−03
Assemble exterior facets	1.83E−04	1.81E−04	1.90E−04	1.66E−04	1.72E−04
Assemble interior facets	–	7.79E−03	8.10E−03	–	–
Assemble system	1.28E−01	1.41E−01	1.56E−01	2.45E−01	3.92E−01
Build sparsity	1.47E−03	1.26E−03	1.56E−03	1.21E−03	1.44E−03
Delete sparsity	1.43E−06	1.63E−06	1.59E−06	1.59E−06	1.42E−06
DirichletBC compute	5.26E−05	5.42E−05	5.61E−05	4.33E−05	5.79E−05
Init dof vector	5.13E−04	4.53E−04	5.77E−04	4.13E−04	4.75E−04
Init dofmap	7.78E−03	8.05E−03	9.01E−03	5.97E−03	8.78E−03
Init tensor	5.02E−04	4.84E−04	5.03E−04	4.89E−04	5.05E−04
LU solver	3.05E−01	4.07E−01	4.53E−01	1.41E+00	6.26E−01

3.4 A Flow Problem in Three-Dimensional Deformable Media

This example extends the previous study in the two-dimensional domain to a three-dimensional domain. The computational domain is given as $\Omega = [0, 1]^3$. The geometry and boundary conditions are presented in Fig. 10(a). The input parameters are similar to those of the previous two-dimensional problem. For the heterogeneous case, the ϕ and $\boldsymbol{\kappa}^0$ fields are constructed by the same specifications as the two-dimensional problem, as illustrated in Fig. 10(b, c). Here, the $\boldsymbol{\sigma}_D$ value is set as $\boldsymbol{\sigma}_{Dx} = [10, 0, 0]$, $\boldsymbol{\sigma}_{Dy} = [0, 10, 0]$, and $\boldsymbol{\sigma}_{Dz} = [0, 0, 10]$ MPa.

This problem also assumes the nonlinear relation $\mathscr{N}[\boldsymbol{\kappa}] = \boldsymbol{\kappa}(\boldsymbol{u})$ in Eq. (14). The definition of Eq. (30) used in the two-dimensional case is again utilized in this problem. First, the summary of the number of DOF among different methods is presented in Table 7. Similar to the previous problem, we can observe that the CG method has the fewest DOF. The DG and MFE-RT methods have approximately the same number of DOF, while MFE-P2 has the highest number of DOF.

Again, throughout this problem, the Newton iteration is used to solve the system of nonlinear equations, and a direct solver with lower–upper decomposition (LU) is employed for each Newton iteration (Balay et al. 2018). The absolute tolerance is set as 1×10^{-6}, relative tolerance is set as 1×10^{-16}, the maximum number of Newton iterations is 100, and the relaxation parameter is 0.75. All simulations are computed on a Xeon E5-2650 v4 processor with a single thread.

3.4.1 Homogeneous Rock Matrix Porosity and Permeability

In this example, we consider that $\boldsymbol{k_m}^0$ is isotropic and homogeneous, and its value is assigned as $\boldsymbol{k}_m{}^0 = 10^{-12}\boldsymbol{I}$ m^2. The results of the RF, $\max(r_{\text{mass}})$, and number of Newton iterations are presented in Fig. 11. We observe that all methods except the CG method present approximately the same RF results. Similar to the two-dimensional

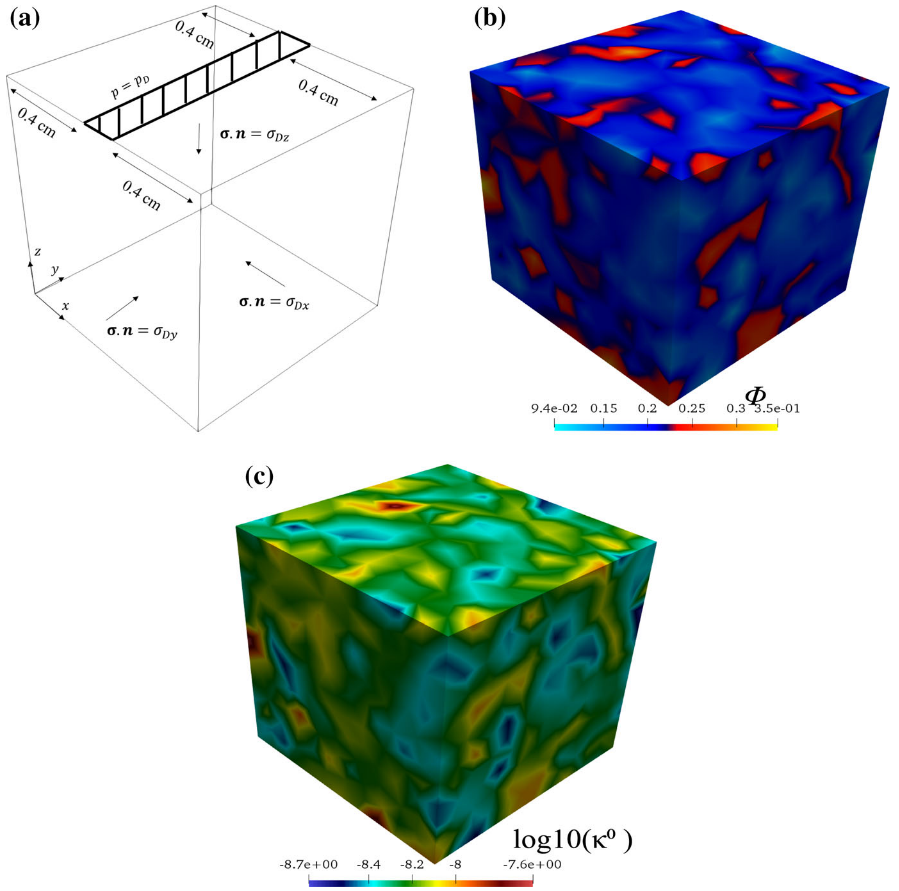
(a)
0.4 cm
0.4 cm
0.4 cm
0.4 cm
$p = p_D$
$\boldsymbol{\sigma}.\boldsymbol{n} = \sigma_{Dz}$
$\boldsymbol{\sigma}.\boldsymbol{n} = \sigma_{Dx}$
$\boldsymbol{\sigma}.\boldsymbol{n} = \sigma_{Dy}$
z
y
x
(b)
Φ
9.4e-02 0.15 0.2 0.25 0.3 3.5e-01
(c)
log10(κ⁰)
-8.7e+00 -8.4 -8.2 -8 -7.6e+00

Fig. 10 Example 3.4: A setup for the three-dimensional problem. **a** Geometry and boundary conditions. Besides the surface that is imposed by the constant pressure boundary, other surfaces are imposed by the zero-flux boundary. Also, the surfaces that are opposite the stress boundaries are imposed with zero normal displacement). **b** Illustration of ϕ and **c** log10($\boldsymbol{\kappa}^0$) fields

Table 7 Example 3.4: Comparison of degrees of freedom (DOF) among CG, EG, DG, MFE-RT, and MFE-P2 methods

Model	Total	Displacement	Velocity	Pressure
CG	21,969	20,901	–	1,068
EG	26,064	20,901	–	5,163
DG	37,281	20,901	–	16,380
MFE-RT	33,923	20,901	8,927	4,095
MFE-P2	42,870	20,901	20,901	1,068

case, the EG, DG, and MFE-RT methods show the property of local mass conservation, while the CG and MFE-P2 methods do not. Figure 11(c) illustrates that the CG and EG methods require the fewest Newton iterations. Unlike the two-dimensional case,

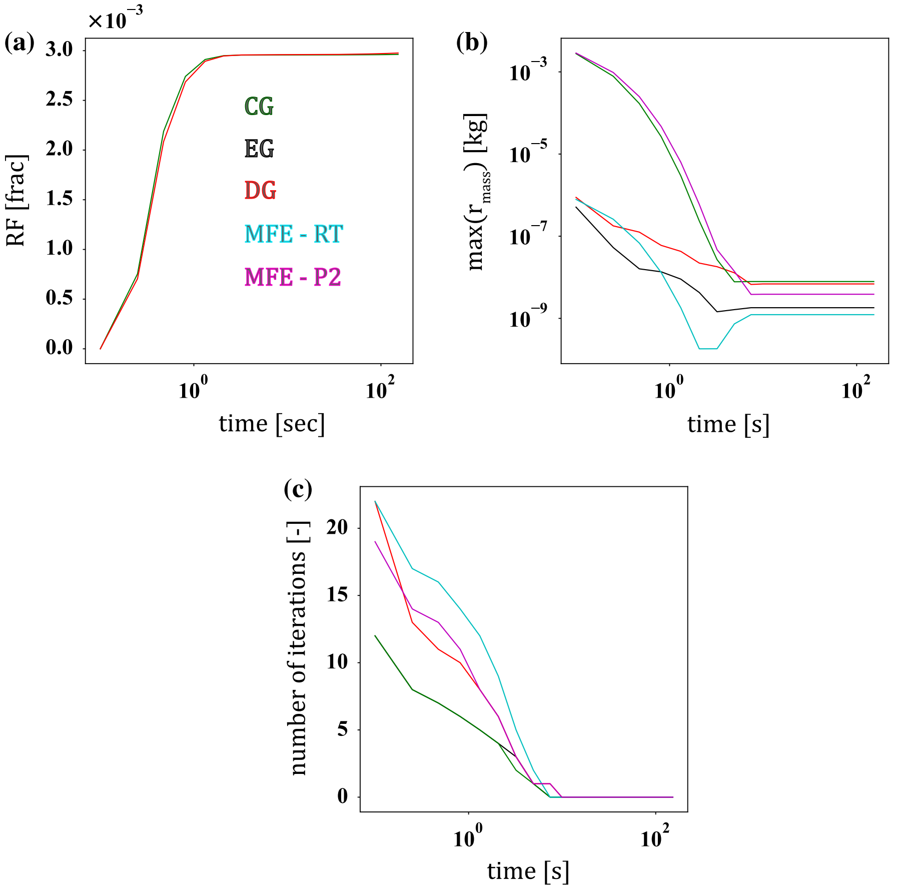

Fig. 11 Example 3.4.1: Comparison of the values of (**a**) RF, (**b**) $\max(r_{\text{mass}})$, and (**c**) number of Newton iterations among the CG (green line), EG (black line), DG (red line), MFE-RT (magenta line), and MFE-P2 (blue line) methods. Note that all results overlap each other in (**a**), and the CG and EG results are overlapped in (**c**)

the DG method requires fewer Newton iterations than both the MFE-RT and MFE-P2 methods, and the MFE-RT method requires the highest number of Newton iterations.

A summary of the total wall and CPU times for all methods is presented in Table 8. We observe that the CG method requires the least wall and CPU time, while the MFE-P2 method requires the most. This result may be related to the number of DOF presented in Table 2 and the number of Newton iterations shown in Fig. 11(c).

A more detailed comparison of the required computational time for each operation is summarized in Table 9. Similar to all previous cases, the main contributors to the wall time are the LU solver and assemble system operations. However, the difference in this three-dimensional case is that the assemble system operation sometimes takes more time than the LU solver for the MFE-RT and MFE-P2 methods. This is a visible result, since it is clear that the greater the number of DOF the method has, the more time it requires to assemble the system. As in the three-dimensional case, the three-

Table 8 Example 3.4.1: Comparison of the total wall and CPU times

Model	Wall time (s)	CPU time (s)
CG	1,481	1,461
EG	2,045	2,018
DG	3,251	3,230
MFE-RT	24,983	24,952
MFE-P2	41,602	41,562

Table 9 Example 3.4.1: Comparison of the average wall time for each operation

Summary of timings	Average wall time (s)				
	CG	EG	DG	MFE-RT	MFE-P2
Assemble cells	8.29E−03	8.19E−03	7.73E−03	7.99E−03	1.02E−02
Assemble exterior facets	2.70E−04	2.64E−04	2.64E−04	2.31E−04	2.43E−04
Assemble interior facets	–	1.10E−02	1.18E−02	–	–
Assemble system	2.17E+00	2.42E+00	2.46E+00	5.47E+01	1.14E+02
Build sparsity	1.06E−02	9.33E−03	1.14E−02	5.44E−03	1.01E−02
Delete sparsity	1.73E−06	1.61E−06	1.56E−06	1.46E−06	1.47E−06
DirichletBC compute	2.76E−04	2.89E−04	2.90E−04	2.18E−04	5.15E−04
Init dof vector	7.66E−04	7.24E−04	6.76E−04	7.76E−04	6.88E−04
Init dofmap	1.66E−02	1.85E−02	1.89E−02	1.26E−02	2.30E−02
Init tensor	8.13E−04	7.85E−04	7.06E−04	8.29E−04	8.07E−04
LU solver	2.76E+00	4.89E+00	7.06E+00	1.33E+01	7.99E+00

formulation DOF increases dramatically compared to the two-dimensional case. The two-field formulation, CG, EG, and DG methods, requires less time to solve a linear system than the three-field formulation MFE-RT and MFE-P2 methods because the two-field formulation has one fewer primary variable than the three-field formulation. The computational time required by the LU solver operation of the MFE-RT is twice that of the MFE-P2 one. Thus, we observe that the LU solver operation is not always proportional to the number of DOF. The computational time required to assemble the system also increases significantly for the MFE-P2 case, which leads to the longer total wall and CPU times relative to the MFE-RT method.

3.4.2 Heterogeneous Rock Matrix Porosity and Permeability

In this example, we illustrate the results obtained with the heterogeneous materials. In Fig. 12, the results of the RF, $\max(r_{\text{mass}})$, and number of Newton iterations are presented. Similar to the heterogeneous case in the two-dimensional domain, the RF results of the EG, DG, and MFE-RT methods are almost identical, while the CG and MFE-P2 RF results are significantly different. Moreover, the EG, DG, and MFE-RT methods show the property of local mass conservation, since ($\max(r_{\text{mass}})$ is always less than the absolute tolerance of 1×10^{-6}). As expected, the CG and MFE-P2 methods do

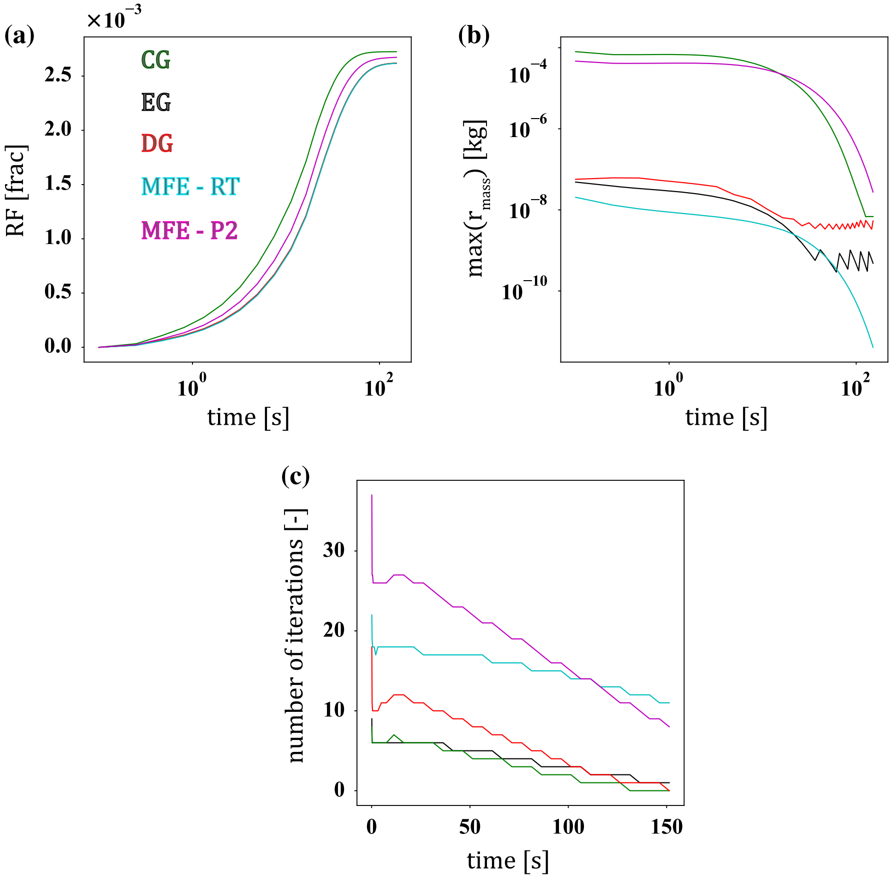

Fig. 12 Example 3.4.2: Comparison of the values of (**a**) RF, (**b**) max(r_{mass}), and (**c**) number of Newton iterations among the CG (green line), EG (black line), DG (red line), MFE-RT (magenta line), and MFE-P2 (blue line) methods. Note that the EG, DG, and MFE-RT results overlap each other in (**a**)

not conserve the local mass. Figure 12(c) shows that the CG and EG methods require the fewest Newton iteration, and the two-field formulation requires fewer Newton iterations than the three-field formulation. In the three-dimensional case, the MFE-RT method has the highest number of Newton iterations instead of the MFE-P2 method.

A summary of the total wall and CPU times for the three-dimensional heterogeneous case for all methods is illustrated in Table 10. The result is similar to the previous homogeneous case in Sect. 3.4.1. The CG method requires the least wall and CPU time, while the MFE-P2 method requires the most wall and CPU time.

The required computational time for each operation is summarized in detail in Table 11. As shown in all the previous cases, the main contributors to the wall time are LU solver and assemble system operations. Also, the assemble system operation for the three-field formulation takes more time than the LU solver operation. As mentioned in the previous case, the LU solver operation of the MFE-RT requires twice the wall time as that of the MFE-P2 method, while the assembly operation of the MFE-P2 method is 2.5 times that of the MFE-RT method.

Table 10 Example 3.4.2: Comparison of the total wall and CPU times

Model	Wall time (s)	CPU time (s)
CG	3,466	3,444
EG	5,449	5,414
DG	9,748	9,733
MFE-RT	141,983	141,960
MFE-P2	351,577	351,547

Table 11 Example 3.4.2: Comparison of the average wall time for each operation

Summary of timings	Average wall time (s)				
	CG	EG	DG	MFE-RT	MFE-P2
Assemble cells	8.18E−03	7.57E−03	7.90E−03	7.98E−03	1.03E−02
Assemble exterior facets	2.56E−04	2.58E−04	2.51E−04	2.36E−04	2.49E−04
Assemble interior facets	–	1.10E−02	1.07E−02	–	–
Assemble system	3.04E+00	3.49E+00	4.07E+00	8.60E+01	1.97E+02
Build sparsity	1.05E−02	9.21E−03	1.11E−02	5.51E−03	1.02E−02
Delete sparsity	1.67E−06	1.45E−06	1.50E−06	1.47E−06	1.56E−06
DirichletBC compute	2.70E−04	2.74E−04	2.70E−04	2.02E−04	4.90E−04
Init dof vector	7.84E−04	6.67E−04	8.08E−04	7.54E−04	8.79E−04
Init dofmap	1.57E−02	1.66E−02	1.69E−02	1.27E−02	2.36E−02
Init tensor	7.78E−04	6.98E−04	7.91E−04	8.14E−04	9.74E−04
LU solver	4.77E+00	8.64E+00	8.86E+00	2.93E+01	1.55E+01

4 Discussion

We compared and illustrated the performance of five finite element approximations, CG, EG, DG, MFE-RT, and MFE-P2, shown in Table 1 using examples that span one- to three-dimensional geometries and homogeneous to heterogeneous materials. Each numerical method gives pressure solutions for linear one-dimensional consolidation models that are almost the same as Terzaghi's analytical solution, with less than 1% difference. Where these methods diverge is when local mass conservation must be enforced across all interfaces, especially those with high conductivity contrasts. In this case, the CG and MFE-P2 methods might produce spurious oscillations at the material interface (see Sect. 3.2), suggesting they may not be good representations of the linear Biot system in a heterogeneous material. It is well known that the CG and MFE-P2 methods without appropriate stabilization produce spurious pressure oscillations, and these oscillations can be eliminated by introducing stabilization terms (Choo 2019; Choo and Borja 2015; Hong and Kraus 2017; Honorio et al. 2018; Lee et al. 2017; Rodrigo et al. 2018; White and Borja 2008). As expected, the spurious oscillations do not occur in EG, DG, or MFE-RT methods (Choo and Lee 2018; Haga et al. 2012; Kadeethum et al. 2019a; Liu et al. 2009; Phillips and Wheeler 2007a).

The recovery factor (RF), Newton iteration, average wall time for assembling a system, and LU solver operations are computed for the suite of two- and three-dimensional numerical examples to be used in quantitative comparisons of all the methods presented in Table 1. Results from each method are compared to those of the CG method using a relative change calculation, presented in Table 12. The CG method is used as the base case because it is the most classic finite element method. There is a small difference between each method for the RF results in a two-dimensional homogeneous case (Sect. 3.3.1). The maximum difference is only 0.07%. For heterogeneous cases, however, the maximum difference in RF results is at least 10%. We find that the RF differences among the EG, DG, and MFE-RT methods are minimal, within 2%. The RF results show that the MFE-P2 method is 10.09% different from the CG method and is approximately 5% different from the group of EG, DG, and MFE-RT methods. There are two main explanations for this behavior. The first is that the EG, DG, and MFE-RT methods do not produce spurious oscillations at material interfaces. Therefore, they provide more accurate flux calculation and conductivity alteration (Eqs. 14, 12, 30). The second reason is that these three methods provide local mass conservation, which leads to more accurate flux approximations.

For the three-dimensional homogeneous material example, the RF results from the EG, DG, MFE-RT, and MFE-P2 methods differ by 7 to 10% from the RF results with the CG method. The differences become more significant for models with heterogeneous material, such that the relative change from the CG method for the other four methods ranges from 28 to 40%. The RF result of the MFE-P2 methods using homogeneous material is similar to the results for the EG, DG, and MFE-RT, but the difference becomes more significant for models with heterogeneous material. We can see that the methods that have the property of local mass conservation (i.e., EG, DG, and MFE-RT) can provide more robust flux approximation, especially for heterogeneous materials.

An examination of the overall trend reveals that the two-field formulation requires fewer Newton iterations than the three-field formulation because it has fewer primary variables, which leads to faster convergence behavior. The EG method does not require a significantly higher number of Newton iterations than the CG method, and EG requires the fewest Newton iterations among the methods with local mass conservation. The number of Newton iterations increases when a material is heterogeneous. The number of Newton iterations can influence the total wall time required by the simulation. As discussed in Sect. 3, the number of Newton iterations is not the only factor that affects the simulation time. The assembly operation is also a major contributor to the wall time. As expected, the higher the number of DOF, the greater the time required to assemble the system. Hence, the assembly time for the three-field formulation is significantly greater than that of the two-field formulation, especially for three-dimensional examples.

For the LU solver operation inside each nonlinear iteration, the three-field formulation methods generally require longer wall time than the two-field formulation methods. The MFE-RT method requires the longest wall time, while the CG method requires the least time. The time required for the LU solver operation is significantly increased when modeling heterogeneous materials. Among the methods that have the local mass conservation property, the EG method requires the least time for the LU solver operation.

Table 12 Summary of comparisons among the methods in Table 1 using the CG method as a base case

	RF difference		Newton iteration difference	Wall time		
	Average (%)	Maximum (%)	Total (%)	Total (%)	Assemble system (%)	Average solver average (%)
2-Dimensional						
homogeneous						
EG	0.00	0.01	0.00	6.46	8.60	21.73
DG	0.05	0.05	75.68	20.15	12.18	40.01
MFE-RT	0.04	0.07	86.49	47.91	53.92	216.50
MFE-P2	0.02	0.07	45.95	23.95	127.43	61.08
heterogeneous						
EG	5.15	15.16	14.43	27.45	10.17	33.19
DG	4.66	14.81	72.16	52.10	22.06	48.37
MFE-RT	5.31	16.46	252.58	242.02	91.55	362.88
MFE-P2	3.02	10.09	160.82	121.85	206.12	105.08
3-Dimensional						
homogeneous						
EG	0.18	6.96	2.22	38.08	11.22	77.16
DG	0.53	6.46	66.67	119.51	13.43	155.39
MFE-RT	0.69	8.81	115.56	1,586.90	2,419.59	381.27
MFE-P2	0.60	9.29	68.89	2,709.05	5,156.27	189.40
heterogeneous						
EG	15.94	40.23	15.71	57.21	14.99	81.33
DG	15.83	40.23	87.14	181.25	34.04	85.95
MFE-RT	16.67	40.69	322.14	3,996.45	2,729.36	514.58
MFE-P2	9.72	28.71	439.29	10,043.59	6,384.93	226.01

5 Conclusions

We evaluate the performance of five different finite element approximations for flow in deformable porous media. Here, we solve Biot's system, which involves multi-physics processes coupling fluid flow and bulk deformation problems. The numerical results and comparisons provided aid users in their selection of an appropriate method that best suits the desired accuracy of their results and their access to computational resources. This approach could be applied to many different studies, including land subsidence by groundwater overexploitation, enhanced geothermal systems and fossil fuel production in deep and tight reservoirs, hydraulic fracturing-induced seismicity, and biomedical engineering involving porous hyperelastic material. In particular, we present quantitative comparisons of flux approximation, local mass conservation property, degrees of freedom, and computation time between five different two- and three-field formulations for solving linear and nonlinear Biot systems. The approximated fluxes from the five methods are similar (less than 10% relative difference) for the linear Biot system with homogeneous material properties. However, at least 10% relative difference in flux approximation is observed in cases with heterogeneous material. Here, the accurate approximation of the fluxes is crucial to conserve the local mass and to avoid spurious oscillations. However, longer wall and CPU times are required due to the larger number of degrees of freedom and Newton iterations for local mass conservation methods.

Acknowledgements This research has received financial support from the Danish Hydrocarbon Research and Technology Centre under the Advanced Water Flooding program. The computational results in this work were produced by the multiphenics library (Ballarin and Rozza 2019), which is an extension of FEniCS (Alnaes et al. 2015) for multiphysics problems. We acknowledge the developers of and contributors to these libraries. SL is supported by the National Science Foundation under Grant No. NSF DMS-1913016. The authors would like to thank Dr. Solomon Seyum for constructive criticism of the manuscript.

Appendices

In these appendices, we describe the fully discretized formulation of each method listed in Table 1.

A Continuous Galerkin (CG) Method

We begin by defining the finite element space for the continuous Galerkin (CG) method for a vector-valued function

$$\mathscr{U}_h^{\mathrm{CG}_k}(\mathscr{T}_h) := \left\{ \boldsymbol{\psi}_{\boldsymbol{u}} \in \mathbb{C}^0(\Omega; \mathbb{R}^d) : \boldsymbol{\psi}_{\boldsymbol{u}}|_T \in \mathbb{Q}_k(T; \mathbb{R}^d), \forall T \in \mathscr{T}_h \right\}, \tag{37}$$

where k indicates the order of polynomials, $\mathbb{C}^0(\Omega; \mathbb{R}^d)$ denotes the space of vector-valued piecewise continuous polynomials, and $\mathbb{Q}_k(T; \mathbb{R}^d)$ is the space of polynomials of degree at most k over each element T. For our problem, the vector-valued displacement field is approximated using CG space. Next, the CG space for scalar-valued functions is defined as

$$\mathscr{P}_h^{\mathrm{CG}_k}(\mathscr{T}_h) := \left\{ \psi_p \in \mathbb{C}^0(\Omega) : \psi_p\big|_T \in \mathbb{Q}_k(T), \forall T \in \mathscr{T}_h \right\}, \tag{38}$$

for the pressure solution.

Thus, we seek the approximated displacement solution ($\boldsymbol{u_h}$) by discretizing the linear momentum balance Eq. (6) employing the above CG finite element spaces as done in Choo and Lee (2018), Kadeethum et al. (2019a) and Vik et al. (2018). The fully discretized linear momentum balance Eq. (6) can be defined using the following forms

$$\mathscr{A}_u\left((\boldsymbol{u}_h^n, p_h^n), \boldsymbol{\psi_u}\right) = \mathscr{L}_u(\boldsymbol{\psi_u}), \quad \forall \boldsymbol{\psi_u} \in \mathscr{U}_h^{\mathrm{CG}_2}(\mathscr{T}_h), \tag{39}$$

at each time step t^n, where

$$\mathscr{A}_u\left((\boldsymbol{u}_h^n, p_h^n), \boldsymbol{\psi_u}\right) = \sum_{T\in\mathscr{T}_h} \int_T \boldsymbol{\sigma}'(\boldsymbol{u}_h) : \nabla^s \boldsymbol{\psi_u}\, dV - \sum_{T\in\mathscr{T}_h} \int_T \alpha p_h \boldsymbol{I} : \nabla^s \boldsymbol{\psi_u}\, dV, \tag{40}$$

and

$$\mathscr{L}_u(\boldsymbol{\psi_u}) = \sum_{T\in\mathscr{T}_h} \int_T \boldsymbol{f}\boldsymbol{\psi_u}\, dV + \sum_{e\in\mathscr{E}_h^N} \int_e \boldsymbol{\sigma}_D \boldsymbol{\psi_u}\, dS, \quad \forall \boldsymbol{\psi_u} \in \mathscr{U}_h^{\mathrm{CG}_2}(\mathscr{T}_h). \tag{41}$$

Here, ∇^s is a symmetric gradient operator. We can split

$$\mathscr{A}_u\left((\boldsymbol{u}_h^n, p_h^n), \boldsymbol{\psi_u}\right) := a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi_u}\right) + b\left(p_h^n, \boldsymbol{\psi_u}\right), \tag{42}$$

by using

$$a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi_u}\right) := \sum_{T\in\mathscr{T}_h} \int_T \boldsymbol{\sigma}'\left(\boldsymbol{u}_h^n\right) : \nabla^s \boldsymbol{\psi_u}\, dV, \tag{43}$$

$$b\left(p_h^n, \boldsymbol{\psi_u}\right) := \sum_{T\in\mathscr{T}_h} \int_T -\alpha p_h^n \boldsymbol{I} : \nabla^s \boldsymbol{\psi_u}\, dV. \tag{44}$$

For the approximated pressure solution (p_h), we solve the mass balance Eq. (10) combined with Eqs. (11) and (25) as follows

$$\mathscr{A}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) = \mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right), \quad \forall \psi_p \in \mathscr{P}_h^{\mathrm{CG}_1}(\mathscr{T}_h), \tag{45}$$

for each time step t^n, where

$$
\begin{aligned}
&\mathscr{A}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right)\\
&\quad := \sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^n\psi_p\, dV + \sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right) p_h^n\psi_p\, dV\\
&\qquad + \Delta t^n \sum_{T\in\mathscr{T}_h}\int_T \boldsymbol{\kappa}(\boldsymbol{u}_h^n)\left(\nabla p_h^n - \rho\mathbf{g}\right)\cdot\nabla\psi_p\, dV\\
&\qquad - \Delta t^n \sum_{e\in\mathscr{E}_h^D}\int_e \left\{\boldsymbol{\kappa}(\boldsymbol{u}_h^n)\left(\nabla p_h^n - \rho\mathbf{g}\right)\right\}_{\delta_e}\cdot [\![\psi_p]\!]\, dS\\
&\qquad - \Delta t^n \sum_{e\in\mathscr{E}_h^D}\int_e \left\{\boldsymbol{\kappa}(\boldsymbol{u}_h^n)\nabla\psi_p\right\}_{\delta_e}\cdot [\![p_h^n]\!]\, dS\\
&\qquad + \Delta t^n \sum_{e\in\mathscr{E}_h^D}\int_e \frac{\beta}{h_e}\boldsymbol{\kappa}(\boldsymbol{u}_h^n)_e [\![p_h^n]\!]\cdot[\![\psi_p]\!]\, dS,
\end{aligned}
\tag{46}
$$

$$
\begin{aligned}
&\mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right)\\
&\quad := \sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^{n-1}\psi_p\, dV + \sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right) p_h^{n-1}\psi_p\, dV\\
&\qquad + \Delta t^n \sum_{T\in\mathscr{T}_h}\int_T g\psi_p\, dV + \Delta t^n \sum_{e\in\mathscr{E}_h^N}\int_e q_D\psi_p\, dS\\
&\qquad - \Delta t^n \sum_{e\in\mathscr{E}_h^D}\int_e \boldsymbol{\kappa}(\boldsymbol{u}_h^n)\nabla\psi_p\cdot p_D\boldsymbol{n}\, dS\\
&\qquad + \Delta t^n \sum_{e\in\mathscr{E}_h^D}\int_e \frac{\beta}{h_e}\boldsymbol{\kappa}(\boldsymbol{u}_h^n)_e [\![\psi_p]\!]\cdot p_D\boldsymbol{n}\, dS.
\end{aligned}
\tag{47}
$$

Here, again, we can split

$$
\mathscr{A}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) := c\left(\boldsymbol{u}_h^n, \psi_p\right) + d\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right),
\tag{48}
$$

with

$$
c\left(\boldsymbol{u}_h^n, \psi_p\right) := \sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^n\psi_p\, dV,
\tag{49}
$$

$$
\begin{aligned}
d\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right) &:= \sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right) p_h^n\psi_p\, dV\\
&\quad + \Delta t^n \sum_{T\in\mathscr{T}_h}\int_T \boldsymbol{\kappa}(\boldsymbol{u}_h^n)\left(\nabla p_h^n - \rho\mathbf{g}\right)\cdot\nabla\psi_p\, dV
\end{aligned}
$$

$$
\begin{aligned}
&- \Delta t^n \sum_{e \in \mathscr{E}_h^D} \int_e \left\{ \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \mathbf{g} \right) \right\}_{\delta_e} \cdot [\![\psi_p]\!] \, dS \\
&- \Delta t^n \sum_{e \in \mathscr{E}_h^D} \int_e \left\{ \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \nabla \psi_p \right\}_{\delta_e} \cdot [\![p_h^n]\!] \, dS \\
&+ \Delta t^n \sum_{e \in \mathscr{E}_h^D} \int_e \frac{\beta}{h_e} \boldsymbol{\kappa}(\boldsymbol{u}_h^n)_e [\![p_h^n]\!] \cdot [\![\psi_p]\!] \, dS. \qquad (50)
\end{aligned}
$$

We note that the symmetric internal penalty approach is employed to weakly impose the Dirichlet boundary condition (Bazilevs and Hughes 2007; Nitsche 1971; Hansbo 2005). The interior penalty parameter β and h_e characteristic length are calculated as

$$
h_e := \frac{\operatorname{meas}\left(T^+\right) + \operatorname{meas}\left(T^-\right)}{2 \operatorname{meas}(e)}. \qquad (51)
$$

The system of equations is nonlinear (the nonlinear variable being reported after a semicolon for the sake of clarity). Thus, Newton's method is employed in this work. The Jacobian matrix that arises from Eqs. (39) and (45) is

$$
\begin{bmatrix} \mathscr{J}_{uu} & \mathscr{J}_{up} \\ \mathscr{J}_{pu} & \mathscr{J}_{pp} \end{bmatrix} \begin{Bmatrix} \delta \boldsymbol{u}_h \\ \delta p_h \end{Bmatrix} = - \begin{Bmatrix} R_u \\ R_p \end{Bmatrix}, \qquad (52)
$$

where

$$
\begin{aligned}
\mathscr{J}_{uu} &:= \frac{\partial a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial \boldsymbol{u}_h^n}, \quad & \mathscr{J}_{up} &:= \frac{\partial b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial p_h^n}, \\
\mathscr{J}_{pu} &:= \frac{\partial c\left(\boldsymbol{u}_h^n, \psi_p\right)}{\partial \boldsymbol{u}_h^n}, \quad & \mathscr{J}_{pp} &:= \frac{\partial d\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right)}{\partial p_h^n},
\end{aligned} \qquad (53)
$$

and the residual vector is defined as

$$
\begin{aligned}
R_u &:= a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) + b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) - \mathscr{L}_u\left(\boldsymbol{\psi}_{\boldsymbol{u}}\right), \\
R_p &:= c\left(\boldsymbol{u}_h^n, \psi_p\right) + d\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right) - \mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right).
\end{aligned} \qquad (54)
$$

Here, $\delta \boldsymbol{u}_h$ and δp_h are Newton increments of the $\boldsymbol{u}_h$ and p_h, respectively. The block structure for the CG method arises as follows

$$
\begin{bmatrix} \mathscr{J}_{uu}^{\mathrm{CG}_2 \times \mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2 \times \mathrm{CG}_1} \\ \mathscr{J}_{pu}^{\mathrm{CG}_1 \times \mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{CG}_1 \times \mathrm{CG}_1} \end{bmatrix} \begin{Bmatrix} \left(\delta \boldsymbol{u}_h^n\right)^{\mathrm{CG}_2} \\ \left(\delta p_h^n\right)^{\mathrm{CG}_1} \end{Bmatrix} = - \begin{Bmatrix} R_u^{\mathrm{CG}_2} \\ R_p^{\mathrm{CG}_1} \end{Bmatrix}. \qquad (55)
$$

B Discontinuous Galerkin (DG) Method

For the discontinuous Galerkin (DG) method presented in Table 1, we approximate the $\boldsymbol{u}_h$ using the CG finite element space, Eqs. (37) and (39), but we utilize the DG finite element space to approximate the p_h. The DG finite element space with the polynomial order k is given as

$$\mathscr{P}_h^{\mathrm{DG}_k}\left(\mathscr{T}_h\right) := \left\{ \psi_p \in L^2(\Omega) : \psi_p\big|_T \in \mathbb{Q}_k(T), \forall T \in \mathscr{T}_h \right\}, \tag{56}$$

where $L^2(\Omega)$ is the space of the square integrable functions. This nonconforming finite element space allows us to consider discontinuous coefficients and preserves the local mass conservation. We then discretize Eqs. (10), (11), and (25) as

$$\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) = \mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right), \quad \forall \psi_p \in \mathscr{P}_h^{\mathrm{DG}_1}\left(\mathscr{T}_h\right), \tag{57}$$

for each time step t^n, where

$$\begin{aligned}
&\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) \\
&\quad := \sum_{T \in \mathscr{T}_h} \int_T \rho\alpha \nabla \cdot \boldsymbol{u}_h^n \psi_p \, dV + \sum_{T \in \mathscr{T}_h} \int_T \rho \left(\phi c_f + \frac{\alpha - \phi}{K_s} \right) p_h^n \psi_p \, dV \\
&\qquad + \Delta t^n \sum_{T \in \mathscr{T}_h} \int_T \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \mathbf{g} \right) \cdot \nabla \psi_p \, dV \\
&\qquad - \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \left\{ \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \mathbf{g} \right) \right\}_{\delta_e} \cdot [\![\psi_p]\!] \, dS \\
&\qquad - \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \left\{ \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \nabla \psi_p \right\}_{\delta_e} \cdot [\![p_h^n]\!] \, dS \\
&\qquad + \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \frac{\beta}{h_e} \boldsymbol{\kappa}(\boldsymbol{u}_h^n)_e [\![p_h^n]\!] \cdot [\![\psi_p]\!] \, dS.
\end{aligned} \tag{58}$$

The $\mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right)$ term is similar to Eq. (47) as presented in the previous section. Next, we split

$$\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) := c\left(\boldsymbol{u}_h^n, \psi_p\right) + e\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right). \tag{59}$$

We utilize the same definition for $c\left(\boldsymbol{u}_h^n, \psi_p\right)$ from the CG method, Eq. (49), but we define $e\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right)$ as

$$
\begin{aligned}
e\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right) :=& \sum_{T \in \mathscr{T}_h} \int_T \rho \left(\phi c_f + \frac{\alpha - \phi}{K_s}\right) p_h^n \psi_p \, dV \\
&+ \Delta t^n \sum_{T \in \mathscr{T}_h} \int_T \boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \mathbf{g}\right) \cdot \nabla \psi_p \, dV \\
&- \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \left\{\boldsymbol{\kappa}(\boldsymbol{u}_h^n) \left(\nabla p_h^n - \rho \mathbf{g}\right)\right\}_{\delta_e} \cdot [\![\psi_p]\!] \, dS \\
&- \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \left\{\boldsymbol{\kappa}(\boldsymbol{u}_h^n) \nabla \psi_p\right\}_{\delta_e} \cdot [\![p_h^n]\!] \, dS \\
&+ \Delta t^n \sum_{e \in \mathscr{E}_h^I \cup \mathscr{E}_h^D} \int_e \frac{\beta}{h_e} \boldsymbol{\kappa}(\boldsymbol{u}_h^n)_e [\![p_h^n]\!] \cdot [\![\psi_p]\!] \, dS.
\end{aligned}
\tag{60}
$$

Compared to $\mathscr{A}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right)$ from the CG method, we note that there are additional surface integral terms $\int_e \cdot \, dS$ on the interior faces ($e \in \mathscr{E}_h^I$) in $\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right)$. This results in more demanding computational time when the matrix is assembled.

Again, we note that this system of equations are nonlinear forms in the displacement $\boldsymbol{u}_h^n$. We form the Jacobian matrix that arises from Eqs. (39) and (57) as follows

$$
\begin{bmatrix} \mathscr{J}_{uu} & \mathscr{J}_{up} \\ \mathscr{J}_{pu} & \mathscr{J}_{pp} \end{bmatrix} \begin{Bmatrix} \delta \boldsymbol{u}_h \\ \delta p_h \end{Bmatrix} = - \begin{Bmatrix} R_u \\ R_p \end{Bmatrix},
\tag{61}
$$

where

$$
\begin{aligned}
\mathscr{J}_{uu} &:= \frac{\partial a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial \boldsymbol{u}_h^n}, \quad \mathscr{J}_{up} := \frac{\partial b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial p_h^n}, \\
\mathscr{J}_{pu} &:= \frac{\partial c\left(\boldsymbol{u}_h^n, \psi_p\right)}{\partial \boldsymbol{u}_h^n}, \quad \mathscr{J}_{pp} := \frac{\partial e\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right)}{\partial p_h^n},
\end{aligned}
\tag{62}
$$

and the residual vector is defined as

$$
\begin{aligned}
R_u &:= a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) + b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) - \mathscr{L}_u\left(\boldsymbol{\psi}_{\boldsymbol{u}}\right), \\
R_p &:= c\left(\boldsymbol{u}_h^n, \psi_p\right) + e\left(p_h^n, \psi_p; \boldsymbol{u}_h^n\right) - \mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right),
\end{aligned}
\tag{63}
$$

at each time step t^n. The block structure for the DG method arises as follows

$$
\begin{bmatrix} \mathscr{J}_{uu}^{\mathrm{CG}_2 \times \mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2 \times \mathrm{DG}_1} \\ \mathscr{J}_{pu}^{\mathrm{DG}_1 \times \mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{DG}_1 \times \mathrm{DG}_1} \end{bmatrix} \begin{Bmatrix} \left(\delta \boldsymbol{u}_h^n\right)^{\mathrm{CG}_2} \\ \left(\delta p_h^n\right)^{\mathrm{DG}_1} \end{Bmatrix} = - \begin{Bmatrix} R_u^{\mathrm{CG}_2} \\ R_p^{\mathrm{DG}_1} \end{Bmatrix}.
\tag{64}
$$

C Enriched Galerkin (EG) Method

For the EG method presented in Table 1, we define the EG finite element space with the polynomial order k as

$$\mathscr{P}_h^{\mathrm{EG}_k}\left(\mathscr{T}_h\right) := \mathscr{P}_h^{\mathrm{CG}_k}\left(\mathscr{T}_h\right) + \mathscr{P}_h^{\mathrm{DG}_0}\left(\mathscr{T}_h\right), \tag{65}$$

where the CG finite element space is enriched by $\mathscr{P}_h^{\mathrm{DG}_0}\left(\mathscr{T}_h\right)$, the piecewise constant functions. As in the previous section, we approximate the displacement ($\boldsymbol{u}_h$) using the CG finite element space Eq. (39), but here the p_h is approximated by the EG finite element space Eq. (65). Thus, we seek p_h by solving

$$\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right) = \mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right), \quad \forall \psi_p \in \mathscr{P}_h^{\mathrm{EG}_1}\left(\mathscr{T}_h\right). \tag{66}$$

We note that the EG method employs the same bilinear $\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right)$ and linear $\mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right)$ forms as the DG method. Since the EG space Eq. (65) only takes piecewise constants (DG$_0$) to enforce the discontinuity, several terms involving the derivatives in $\mathscr{B}_p\left((\boldsymbol{u}_h^n, p_h^n), \psi_p; \boldsymbol{u}_h^n\right)$ and $\mathscr{L}_p\left(\psi_p; \boldsymbol{u}_h^n\right)$ become zero. Thus, not only does the EG method have fewer degrees of freedom than DG, but it requires less computational time for assembling the matrix.

We again note that this system of equations are nonlinear forms in the displacement $\boldsymbol{u}_h^n$. We employ the same Jacobian matrix, Eqs. (61) and (62), and residual vector, Eq. (63), built for the DG method. The block structure used for the DG method Eq. (64), however, is decomposed into

$$\begin{bmatrix} \mathscr{J}_{uu}^{\mathrm{CG}_2\times\mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2\times\mathrm{CG}_1} & \mathscr{J}_{up}^{\mathrm{CG}_2\times\mathrm{DG}_0} \\ \mathscr{J}_{pu}^{\mathrm{CG}_1\times\mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{CG}_1\times\mathrm{CG}_1} & \mathscr{J}_{pp}^{\mathrm{CG}_1\times\mathrm{DG}_0} \\ \mathscr{J}_{pu}^{\mathrm{DG}_0\times\mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{DG}_0\times\mathrm{CG}_1} & \mathscr{J}_{pp}^{\mathrm{DG}_0\times\mathrm{DG}_0} \end{bmatrix} \begin{Bmatrix} \left(\delta\boldsymbol{u}_h^n\right)^{\mathrm{CG}_2} \\ \left(\delta {p_h}^n\right)^{\mathrm{CG}_1} \\ \left(\delta {p_h}^n\right)^{\mathrm{DG}_0} \end{Bmatrix} = -\begin{Bmatrix} R_u^{\mathrm{CG}_2} \\ R_p^{\mathrm{CG}_1} \\ R_p^{\mathrm{DG}_0} \end{Bmatrix}, \tag{67}$$

to satisfy Eq. (66).

D Raviart–Thomas Mixed Finite Element (MFE-RT) Method

For the MFE-RT method presented in Table 1, we also approximate the $\boldsymbol{u}_h$ using the CG finite element space [Eqs. (37) and (39)]. The mass balance Eq. (10) for the MFE-RT method is discretized by using the piecewise constant (i.e., DG finite element space Eq. (56) with $k = 0$).

Thus, we seek p_h by solving

$$\mathscr{C}_p\left((\boldsymbol{u}_h^n, p_h^n, \boldsymbol{v}_h^n), \psi_p\right) = \mathscr{M}_p\left(\psi_p\right), \quad \forall \psi_p \in \mathscr{P}_h^{\mathrm{DG}_0}\left(\mathscr{T}_h\right), \tag{68}$$

for each time step t^n, where

$$\begin{aligned}\mathscr{C}_p\left((\boldsymbol{u}_h^n, p_h^n, \boldsymbol{v}_h^n), \psi_p; \boldsymbol{u}_h^n\right) := &\sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^n\psi_p\, dV\\ &+\sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right)p_h^n\psi_p\, dV\\ &+\Delta t^n\sum_{T\in\mathscr{T}_h}\int_T \nabla\cdot\left(\rho\boldsymbol{v}_h^n\right)\psi_p\, dV,\end{aligned} \tag{69}$$

and

$$\begin{aligned}\mathscr{M}_p\left(\psi_p; \boldsymbol{u}_h^n\right) := &\sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^{n-1}\psi_p\, dV\\ &+\sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right)p_h^{n-1}\psi_p\, dV\\ &+\Delta t^n\sum_{T\in\mathscr{T}_h}\int_T g\psi_p\, dV.\end{aligned} \tag{70}$$

Here, we can split Eq. (69) as

$$\mathscr{C}_p\left((\boldsymbol{u}_h^n, p_h^n, \boldsymbol{v}_h^n), \psi_p\right) := f\left(\boldsymbol{u}_h^n, \psi_p\right) + g\left(p_h^n, \psi_p\right) + h\left(\boldsymbol{v}_h^n, \psi_p\right), \tag{71}$$

where

$$f\left(\boldsymbol{u}_h^n, \psi_p\right) := \sum_{T\in\mathscr{T}_h}\int_T \rho\alpha\nabla\cdot\boldsymbol{u}_h^n\psi_p\, dV, \tag{72}$$

$$g\left(p_h^n, \psi_p\right) := \sum_{T\in\mathscr{T}_h}\int_T \rho\left(\phi c_f + \frac{\alpha-\phi}{K_s}\right)p_h^n\psi_p\, dV, \tag{73}$$

and

$$h\left(\boldsymbol{v}_h^n, \psi_p\right) := \Delta t^n\sum_{T\in\mathscr{T}_h}\int_T \nabla\cdot\left(\rho\boldsymbol{v}_h^n\right)\psi_p\, dV. \tag{74}$$

For the velocity approximation ($\boldsymbol{v}_h$), we use the RT finite element space, which is defined as

$$\mathscr{V}_h^{\mathrm{RT}_k}\left(\mathscr{T}_h\right) := \left\{\boldsymbol{\psi}_{\boldsymbol{v}} \in \mathbb{C}^0(\Omega) : \boldsymbol{\psi}_{\boldsymbol{v}}|_T \in \mathbb{Q}_k(T)^m + x\mathbb{Q}_k(T), \forall T \in \mathscr{T}_h\right\}, \tag{75}$$

$\mathbf{q}(x) = \begin{pmatrix} q_1(x) \\ q_2(x) \\ \vdots \\ q_m(x) \end{pmatrix} + q_0(x) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \quad \forall x := (x_1, x_2, \ldots, x_m)^{\mathrm{T}} \in \Omega, \; \mathbf{q}(x) \subset \mathbb{Q}_k(T)^m$, and $q_0(x) \subset \mathbb{Q}_k(T)$. We then solve the following Darcy velocity Eq. (11) to obtain $\boldsymbol{v}_h^n$ by

$$\mathscr{C}_v\left((\boldsymbol{v}_h^n, p_h^n), \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) = \mathscr{M}_v\left(\boldsymbol{\psi}_{\boldsymbol{v}}\right), \quad \forall \boldsymbol{\psi}_v \in \mathscr{V}_h^{\mathrm{RT}_1}\left(\mathscr{T}_h\right), \tag{76}$$

where

$$\mathscr{C}_v\left((\boldsymbol{v}_h^n, p_h^n), \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) := \sum_{T \in \mathscr{T}_h} \int_T p_h^n \nabla \cdot \boldsymbol{\psi}_{\boldsymbol{v}} \, dV + \sum_{T \in \mathscr{T}_h} \int_T \rho \boldsymbol{\kappa}(\boldsymbol{u}_h^n)^{-1} \boldsymbol{v}_h^n \boldsymbol{\psi}_{\boldsymbol{v}} \, dV, \tag{77}$$

and

$$\mathscr{M}_v\left(\boldsymbol{\psi}_{\boldsymbol{v}}\right) := -\sum_{e \in \mathscr{E}_h^D} \int_e p_D \boldsymbol{\psi}_v \cdot \boldsymbol{n} \, dS. \tag{78}$$

Here, we can split

$$\mathscr{C}_v\left((\boldsymbol{v}_h^n, p_h^n,), \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) := i\left(p_h^n, \boldsymbol{\psi}_v\right) + j\left(\boldsymbol{v}_h^n, \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right), \tag{79}$$

where

$$i\left(p_h^n, \boldsymbol{\psi}_v\right) := \sum_{T \in \mathscr{T}_h} \int_T p_h^n \nabla \cdot \boldsymbol{\psi}_{\boldsymbol{v}} \, dV, \tag{80}$$

and

$$j\left(\boldsymbol{v}_h^n, \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) := \sum_{T \in \mathscr{T}_h} \int_T \rho \boldsymbol{\kappa}(\boldsymbol{u}_h^n)^{-1} \boldsymbol{v}_h^n \boldsymbol{\psi}_{\boldsymbol{v}} \, dV. \tag{81}$$

Similar to the CG, DG, and EG methods, we note that this system of equations are nonlinear forms for the displacement $\boldsymbol{u}_h^n$. We then form the Jacobian matrix that arises from Eqs. (39), (68), and (76) as follows

$$\begin{bmatrix} \mathscr{J}_{uu} & \mathscr{J}_{up} & \mathscr{J}_{uv} \\ \mathscr{J}_{pu} & \mathscr{J}_{pp} & \mathscr{J}_{pv} \\ \mathscr{J}_{vu} & \mathscr{J}_{vp} & \mathscr{J}_{vv} \end{bmatrix} \begin{Bmatrix} \delta \boldsymbol{u}_h \\ \delta p_h \\ \delta \boldsymbol{v}_h \end{Bmatrix} = -\begin{Bmatrix} R_u \\ R_p \\ R_v \end{Bmatrix}, \tag{82}$$

where $\delta \boldsymbol{v}_h$ is the Newton increments of the $\boldsymbol{v}_h$ and

$$
\begin{aligned}
&\mathscr{J}_{uu} := \frac{\partial a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial \boldsymbol{u}_h^n}, \quad \mathscr{J}_{up} := \frac{\partial b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right)}{\partial p_h^n}, \quad \mathscr{J}_{uv} := 0, \\
&\mathscr{J}_{pu} := \frac{\partial f\left(\boldsymbol{u}_h^n, \psi_p\right)}{\partial \boldsymbol{u}_h^n}, \quad \mathscr{J}_{pp} := \frac{\partial g\left(p_h^n, \psi_p\right)}{\partial p_h^n}, \quad \mathscr{J}_{pv} := \frac{\partial h\left(\boldsymbol{v}_h^n, \psi_p\right)}{\partial \boldsymbol{v}_h^n}, \\
&\mathscr{J}_{vu} := 0, \quad \mathscr{J}_{vp} := \frac{\partial i\left(p_h^n, \boldsymbol{\psi}_v\right)}{\partial p_h^n}, \quad \mathscr{J}_{vv} := \frac{\partial j\left(\boldsymbol{v}_h^n, \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right)}{\partial \boldsymbol{v}_h^n},
\end{aligned}
\tag{83}
$$

and the residual vector is defined as

$$
\begin{aligned}
R_u &:= a\left(\boldsymbol{u}_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) + b\left(p_h^n, \boldsymbol{\psi}_{\boldsymbol{u}}\right) - \mathscr{L}_u\left(\boldsymbol{\psi}_{\boldsymbol{u}}\right), \\
R_p &:= f\left(\boldsymbol{u}_h^n, \psi_p\right) + g\left(p_h^n, \psi_p\right) + h\left(\boldsymbol{v}_h^n, \psi_p\right) - \mathscr{M}_p\left(\psi_p; \boldsymbol{u}_h^n\right), \\
R_v &:= i\left(p_h^n, \boldsymbol{\psi}_v\right) + j\left(\boldsymbol{v}_h^n, \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) - \mathscr{M}_v\left(\boldsymbol{\psi}_{\boldsymbol{v}}\right).
\end{aligned}
\tag{84}
$$

The block structure of the MFE-RT method arises as

$$
\begin{bmatrix}
\mathscr{J}_{uu}^{\mathrm{CG}_2 \times \mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2 \times \mathrm{DG}_0} & 0 \\
\mathscr{J}_{pu}^{\mathrm{CG}_2 \times \mathrm{DG}_0} & \mathscr{J}_{pp}^{\mathrm{DG}_0 \times \mathrm{DG}_0} & \mathscr{J}_{pv}^{\mathrm{DG}_0 \times \mathrm{RT}_1} \\
0 & \mathscr{J}_{vp}^{\mathrm{RT}_1 \times \mathrm{DG}_0} & \mathscr{J}_{vv}^{\mathrm{RT}_1 \times \mathrm{RT}_1}
\end{bmatrix}
\begin{Bmatrix}
\left(\delta \boldsymbol{u}_h^n\right)^{\mathrm{CG}_2} \\
\left(\delta p_h{}^n\right)^{\mathrm{DG}_0} \\
\left(\delta \boldsymbol{v}_h{}^n\right)^{\mathrm{RT}_1}
\end{Bmatrix}
= -
\begin{Bmatrix}
R_u^{\mathrm{CG}_2} \\
R_p^{\mathrm{DG}_0} \\
R_v^{\mathrm{RT}_1}
\end{Bmatrix}. \tag{85}
$$

E Continuous Mixed Finite Element (MFE-P2) Method

For the MFE-P2 method presented in Table 1, we utilize a similar system of equations as employed by the MFE-RT method. The displacement $\boldsymbol{u}_h$ is approximated by the CG finite element space, which is the same as the CG method Eq. (39). The p_h in this method is approximated by using the CG finite element space Eq. (38) with $k = 1$

$$
\mathscr{C}_p\left(\left(\boldsymbol{u}_h^n, p_h^n, \boldsymbol{v}_h^n\right), \psi_p\right) = \mathscr{M}_p\left(\psi_p\right), \quad \forall \psi_p \in \mathscr{P}_h^{\mathrm{CG}_1}\left(\mathscr{T}_h\right). \tag{86}
$$

Here, the approximated velocity $\boldsymbol{v}_h$ is also discretized by the vector-valued CG finite element space Eq. (37) but with higher-order $k = 2$. Thus, we obtain the following for $\boldsymbol{v}_h$

$$
\mathscr{C}_v\left(\left(\boldsymbol{v}_h^n, p_h^n\right), \boldsymbol{\psi}_v; \boldsymbol{u}_h^n\right) = \mathscr{M}_v\left(\boldsymbol{\psi}_{\boldsymbol{v}}\right), \quad \forall \boldsymbol{\psi}_v \in \mathscr{U}_h^{\mathrm{CG}_2}\left(\mathscr{T}_h\right). \tag{87}
$$

As a result, the block structure of the MFE-RT method Eq. (85) is modified to

$$
\begin{bmatrix}
\mathscr{J}_{uu}^{\mathrm{CG}_2 \times \mathrm{CG}_2} & \mathscr{J}_{up}^{\mathrm{CG}_2 \times \mathrm{CG}_1} & 0 \\
\mathscr{J}_{pu}^{\mathrm{CG}_1 \times \mathrm{CG}_2} & \mathscr{J}_{pp}^{\mathrm{CG}_1 \times \mathrm{CG}_1} & \mathscr{J}_{pv}^{\mathrm{CG}_1 \times \mathrm{CG}_2} \\
0 & \mathscr{J}_{vp}^{\mathrm{CG}_2 \times \mathrm{CG}_1} & \mathscr{J}_{vv}^{\mathrm{CG}_2 \times \mathrm{CG}_2}
\end{bmatrix}
\begin{Bmatrix}
\left(\delta \boldsymbol{u}_h^n\right)^{\mathrm{CG}_2} \\
\left(\delta p_h{}^n\right)^{\mathrm{CG}_1} \\
\left(\delta \boldsymbol{v}_h{}^n\right)^{\mathrm{CG}_2}
\end{Bmatrix}
= -
\begin{Bmatrix}
R_u^{\mathrm{CG}_2} \\
R_p^{\mathrm{CG}_1} \\
R_v^{\mathrm{CG}_2}
\end{Bmatrix}. \tag{88}
$$

Remark 2 For both two- and three-field formulations, the Neumann boundary condition is naturally applied on the boundary faces, $e \in \mathscr{E}_h^N$. For the two-field formulation, the Dirichlet boundary condition is weakly enforced on $e \in \mathscr{E}_h^D$, for all CG, EG, and DG methods. On the other hand, the three-field formulation strongly applies the Dirichlet boundary conditions.

References

Abou-Kassem J, Islam M, Farouq-Ali S (2013) Petroleum reservoir simulations. Elsevier, London

Adler J, Gaspar F, Hu X, Ohm P, Rodrigo C, Zikatanov L (2019) Robust preconditioners for a new stabilized discretization of the poroelastic equations. Preprint arXiv:1905.10353

Almani T, Kumar K, Dogru A, Singh G, Wheeler M (2016) Convergence analysis of multirate fixed-stress split iterative schemes for coupling flow with geomechanics. Comput Methods Appl Mech Eng 311:180–207

Alnaes M, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes M, Wells G (2015) The FEniCS project version 1.5. Arch Numer Softw 3(100):12

Arnold D, Brezzi F, Fortin M (1984) A stable finite element for the Stokes equations. Calcolo 21(4):337–344

Ayuso B, Georgiev I, Kraus J, Zikatanov L (2009) Preconditioning techniques for discontinuous Galerkin methods discretizations for linear elasticity equations

Balay S, Abhyankar S, Adams M, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp W, Kaushik D, Knepley M, May D, McInnes L, Mills R, Munson T, Rupp K, Sanan P, Smith B, Zampini S, Zhang H, Zhang H (2018) PETSc users manual. Technical report ANL-95/11—revision 3.10, Argonne National Laboratory

Ballarin F, Rozza G (2019) Multiphenics-easy prototyping of multiphysics problems in FEniCS. Version: 0.2.0. https://gitlab.com/multiphenics/multiphenics

Bazilevs Y, Hughes T (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput Fluids 36(1):12–26

Biot M (1941) General theory of three-dimensional consolidation. J Appl Phys 12(2):155–164

Biot M, Willis D (1957) The elastic coefficients of the theory of consolidation. J Appl Mech 15:594–601

Bisdom K, Bertotti G, Nick H (2016) A geometrically based method for predicting stress-induced fracture aperture and flow in discrete fracture networks. AAPG Bull 100(7):1075–1097

Bouklas N, Landis C, Huang R (2015a) Effect of solvent diffusion on crack-tip fields and driving force for fracture of hydrogels. J Appl Mech 82(8):081007

Bouklas N, Landis C, Huang R (2015b) A nonlinear, transient finite element method for coupled solvent diffusion and large deformation of hydrogels. J Mech Phys Solids 79:21–43

Cao T, Sanavia L, Schrefler B (2016) A thermo-hydro-mechanical model for multiphase geomaterials in dynamics with application to strain localization simulation. Int J Numer Methods Eng 107(4):312–337

Castelletto N, White J, Tchelepi H (2015) Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics. Int J Numer Anal Methods Geomech 39(14):1593–1618

Castelletto N, White J, Ferronato M (2016) Scalable algorithms for three-field mixed finite element coupled poromechanics. J Comput Phys 327:894–918

Chen Z (2007) Reservoir simulation: mathematical techniques in oil recovery, vol 77. SIAM, New York

Choo J (2018) Large deformation poromechanics with local mass conservation: an enriched Galerkin finite element framework. Int J Numer Methods Eng 116(1):66–90

Choo J (2019) Stabilized mixed continuous/enriched Galerkin formulations for locally mass conservative poromechanics. Comput Methods Appl Mech Eng 357:112568

Choo J, Borja R (2015) Stabilized mixed finite elements for deformable porous media with double porosity. Comput Methods Appl Mech Eng 293:131–154

Choo J, Lee S (2018) Enriched Galerkin finite elements for coupled poromechanics with local mass conservation. Comput Methods Appl Mech Eng 341:311–332

Coussy O (2004) Poromechanics. Wiley, New York

Deng Q, Ginting V, McCaskill B, Torsu P (2017) A locally conservative stabilized continuous Galerkin finite element method for two-phase flow in poroelastic subsurfaces. J Comput Phys 347:78–98

Du J, Wong R (2007) Application of strain-induced permeability model in a coupled geomechanics–reservoir simulator. J Can Pet Technol 46(12):55–61

Ern A, Stephansen A (2008) A posteriori energy-norm error estimates for advection–diffusion equations approximated by weighted interior penalty methods. J Comput Math 2008:488–510

Ern A, Stephansen A, Zunino P (2009) A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity. IMA J Numer Anal 29(2):235–256

Ferronato M, Castelletto N, Gambolati G (2010) A fully coupled 3-D mixed finite element model of biot consolidation. J Comput Phys 229(12):4813–4830

Ferronato M, Franceschini A, Janna C, Castelletto N, Tchelepi H (2019) A general preconditioning framework for coupled multiphysics problems with application to contact-and poro-mechanics. J Comput Phys 398:108887

Frigo M, Castelletto N, Ferronato M (2019) A relaxed physical factorization preconditioner for mixed finite element coupled poromechanics. SIAM J Sci Comput 41(4):B694–B720

Girault V, Wheeler M, Almani T, Dana S (2019) A priori error estimates for a discretized poro-elastic–elastic system solved by a fixed-stress algorithm. Oil Gas Sci Technol Rev IFP Energ Nouvelles 74:24

Haga J, Osnes H, Langtangen H (2012) On the causes of pressure oscillations in low permeable and low compressible porous media. Int J Numer Anal Methods Geomech 36(12):1507–1522

Hansbo P (2005) Nitsche's method for interface problems in computational mechanics. GAMM Mitteilungen 28(2):183–206

Hong Q, Kraus J (2017) Parameter-robust stability of classical three-field formulation of biot's consolidation model. Preprint arXiv:1706.00724

Honorio H, Maliska C, Ferronato M, Janna C (2018) A stabilized element-based finite volume method for poroelastic problems. J Comput Phys 364:49–72

Jaeger J, Cook NG, Zimmerman R (2009) Fundamentals of rock mechanics. Wiley, Berlin

Juanes R, Jha B, Hager B, Shaw J, Plesch A, Astiz L, Dieterich J, Frohlich C (2016) Were the May 2012 Emilia–Romagna earthquakes induced? A coupled flow-geomechanics modeling assessment. Geophys Res Lett 43(13):6891–6897

Kadeethum T, Nick H, Lee S, Richardson C, Salimzadeh S, Ballarin F (2019a) A novel enriched Galerkin method for modelling coupled flow and mechanical deformation in heterogeneous porous media. In 53rd US rock mechanics/geomechanics symposium, New York, NY, USA. American Rock Mechanics Association

Kadeethum T, Salimzadeh S, Nick H (2019b) An investigation of hydromechanical effect on well productivity in fractured porous media using full factorial experimental design. J Petrol Sci Eng 181:106233

Kadeethum T, Jørgensen T, Nick H (2020a) Physics-informed neural networks for solving inverse problems of nonlinear Biot's equations: batch training. In 54th US rock mechanics/geomechanics symposium, Golden, CO, USA. American Rock Mechanics Association

Kadeethum T, Jørgensen T, Nick H (2020b) Physics-informed neural networks for solving nonlinear diffusivity and Biot's equations. PLoS ONE 15(5):e0232683

Kadeethum T, Nick H, Lee S, Ballarin F (2020c) Flow in porous media with low dimensional fractures by employing Enriched Galerkin method. Adv Water Resour 142:103620

Kadeethum T, Salimzadeh S, Nick H (2020d) Well productivity evaluation in deformable single-fracture media. Geothermics 87:101839

Kim J, Tchelepi H, Juanes R (2011) Stability and convergence of sequential methods for coupled flow and geomechanics: fixed-stress and fixed-strain splits. Comput Methods Appl Mech Eng 200(13–16):1591–1606

Kumar S, Oyarzua R, Ruiz-Baier R, Sandilya R (2020) Conservative discontinuous finite volume and mixed schemes for a new four-field formulation in poroelasticity. ESAIM Math Model Numer Anal 54(1):273–299

Lee S, Wheeler M (2017) Adaptive enriched Galerkin methods for miscible displacement problems with entropy residual stabilization. J Comput Phys 331:19–37

Lee S, Wheeler M (2018) Enriched Galerkin methods for two-phase flow in porous media with capillary pressure. J Comput Phys 367:65–86

Lee S, Woocheol C (2019) Optimal error estimate of elliptic problems with Dirac sources for discontinuous and enriched Galerkin methods. Appl Numer Math 150:76–104

Lee S, Wheeler M, Wick T (2016) Pressure and fluid-driven fracture propagation in porous media using an adaptive finite element phase field model. Comput Methods Appl Mech Eng 305:111–132

Lee J, Mardal K, Winther R (2017) Parameter-robust discretization and preconditioning of Biot's consolidation model. SIAM J Sci Comput 39(1):A1–A24

Lewis R, Schrefler B (1998) The finite element method in the static and dynamic deformation and consolidation of porous media. Wiley, Berlin

Liu R, Wheeler M, Dawson C, Dean R (2009) Modeling of convection-dominated thermoporomechanics problems using incomplete interior penalty Galerkin method. Comput Methods Appl Mech Eng 198(9–12):912–919

Liu J, Tavener S, Wang Z (2018) Lowest-order weak Galerkin finite element method for Darcy flow on convex polygonal meshes. SIAM J Sci Comput 40(5):B1229–B1252

Macminn C, Dufresne E, Wettlaufer J (2016) Large deformations of a soft porous material. Phys Rev Appl 5(4):1–30

Murad M, Loula A (1994) On stability and convergence of finite element approximations of Biot's consolidation problem. Int J Numer Methods Eng 37(4):645–667

Murad M, Borges M, Obregon J, Correa M (2013) A new locally conservative numerical method for two-phase flow in heterogeneous poroelastic media. Comput Geotech 48:192–207

Nick H, Raoof A, Centler F, Thullner M, Regnier P (2013) Reactive dispersive contaminant transport in coastal aquifers: numerical simulation of a reactive Henry problem. J Contam Hydrol 145:90–104

Nitsche J (1971) Uber ein variationsprinzip zur losung von dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind. Abhandlungen aus dem mathematischen Seminar der Universität Hamburg 36(1):9–15

Nordbotten J (2014) Cell-centered finite volume discretizations for deformable porous media. Int J Numer Methods Eng 100(6):399–418

Pain C, Saunders J, Worthington M, Singer J, Stuart-Bruges W, Mason G, Goddard A (2005) A mixed finite-element method for solving the poroelastic Biot equations with electrokinetic coupling. Geophys J Int 160(2):592–608

Peaceman D (2000) Fundamentals of numerical reservoir simulation, vol 6. Elsevier, Berlin

Phillips P, Wheeler M (2007a) A coupling of mixed and continuous Galerkin finite element methods for poroelasticity I: the continuous in time case. Comput Geosci 11(2):131

Phillips P, Wheeler M (2007b) A coupling of mixed and continuous Galerkin finite element methods for poroelasticity II: the discrete-in-time case. Comput Geosci 11(2):145–158

Riviere B, Tan J, Thompson T (2017) Error analysis of primal discontinuous Galerkin methods for a mixed formulation of the Biot equations. Comput Math Appl 73(4):666–683

Rodrigo C, Hu X, Ohm P, Adler J, Gaspar F, Zikatanov L (2018) New stabilized discretizations for poroelasticity and the Stokes' equations. Comput Methods Appl Mech Eng 341:467–484

Salimzadeh S, Nick H (2019) A coupled model for reactive flow through deformable fractures in enhanced geothermal systems. Geothermics 81:88–100

Salimzadeh S, Nick H, Zimmerman R (2018) Thermoporoelastic effects during heat extraction from low-permeability reservoirs. Energy 142:546–558

Scovazzi G, Wheeler M, Mikelic A, Lee S (2017) Analytical and variational numerical methods for unstable miscible displacement flows in porous media. J Comput Phys 335:444–496

Sokolova I, Bastisya M, Hajibeygi H (2019) Multiscale finite volume method for finite-volume-based simulation of poroelasticity. J Comput Phys 379:309–324

Terzaghi K (1951) Theoretical soil mechanics. Chapman And Hall, London

Vermeer P, Verruijt A (1981) An accuracy condition for consolidation by finite elements. Int J Numer Anal Methods Geomech 5(1):1–14

Vik H, Salimzadeh S, Nick H (2018) Heat recovery from multiple-fracture enhanced geothermal systems: the effect of thermoelastic fracture interactions. Renew Energy 121:606–622

Vinje V, Brucker J, Rognes M, Mardal K, Haughton V (2018) Fluid dynamics in syringomyelia cavities: Effects of heart ate, CSF velocity, CSF velocity waveform and craniovertebral decompression. Neuroradiol J 31:1971400918795482

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (2020) SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat Methods 17:261–272. https://doi.org/10.1038/s41592-019-0686-2

Wang H (2017) Theory of linear poroelasticity with applications to geomechanics and hydrogeology. Princeton University Press, Princeton

Wheeler M, Xue G, Yotov I (2014) Coupling multipoint flux mixed finite element methods with continuous Galerkin methods for poroelasticity. Comput Geosci 18(1):57–75

White J, Borja R (2008) Stabilized low-order finite elements for coupled solid-deformation/fluid-diffusion and their application to fault zone transients. Comput Methods Appl Mech Eng 197(49–50):4353–4366

White J, Borja R (2011) Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics. Comput Geosci 15(4):647

White J, Castelletto N, Tchelepi H (2016) Block-partitioned solvers for coupled poromechanics: a unified framework. Comput Methods Appl Mech Eng 303:55–74

Zdunek A, Rachowicz W, Eriksson T (2016) A five-field finite element formulation for nearly inextensible and nearly incompressible finite hyperelasticity. Comput Math Appl 72(1):25–47

Zhao Y, Choo J (2020) Stabilized material point methods for coupled large deformation and fluid flow in porous materials. Comput Methods Appl Mech Eng 362:112742