Computational Geosciences (2021) 25:701–714
https://doi.org/10.1007/s10596-020-09981-2

ORIGINAL PAPER

# Approximate inverse-based block preconditioners in poroelasticity

**Andrea Franceschini**[1] · **Nicola Castelletto**[2] · **Massimiliano Ferronato**[3]

Received: 14 June 2019 / Accepted: 28 May 2020 / Published online: 25 June 2020


**Abstract**
We focus on the fully implicit solution of the linear systems arising from a three-field mixed finite element approximation of Biot's poroleasticity equations. The objective is to develop algebraic block preconditioners for the efficient solution of such systems by Krylov subspace methods. In this work, we investigate the use of approximate inverse-based techniques to decouple the native system of equations and obtain explicit sparse approximations of the Schur complements related to the physics-based partitioning of the unknowns by field type. The proposed methods are tested in various numerical experiments including real-world applications dealing with petroleum and geotechnical engineering.

**Keywords** Approximate inverses · Poroelasticity · Iterative methods · Preconditioning

**Mathematics Subject Classification (2010)** 65F08 · 65F10

## 1 Introduction

Poroelasticity theory describes the fully coupled processes of three-dimensional (3D) deformation and fluid flow through the void space in a porous medium [6, 14]. Originally developed as a generalization of the one-dimensional consolidation theory [48], poroelasticity is used today in a broad number of applications, ranging from the geosciences, to petroleum, geotechnical, environmental, and also biomedical engineering. Our focus here is on the solution of fully implicit linear systems arising from the mixed finite element discretization of poroelasticity equations using a three-field formulation with displacement, Darcy's velocity, and pore pressure as primary unknowns.

✉ Massimiliano Ferronato
massimiliano.ferronato@unipd.it

Andrea Franceschini
af1990@stanford.edu

Nicola Castelletto
castelletto1@llnl.gov

[1] Energy Resources Engineering, Stanford University, Stanford, CA, USA

[2] Atmospheric, Earth and Energy Division, Lawrence Livermore National Laboratory, Livermore, CA, USA

[3] Department of Civil, Environmental and Architectural Engineering, University of Padova, Padova, Italy

In 3D large-sized applications, the use of sparse direct solvers is often not an option. Thus, iterative schemes become essential. In recent years, many works have been devoted to the development and implementation of efficient iterative solvers for the Biot problem. Most efforts have considered preconditioned Krylov solvers for a two-field, namely displacement-pressure, formulation, e.g., [4, 11, 22, 25, 40, 42, 50, 51]. As to the three-field formulation considered here, sequential-implicit approaches are often advanced, e.g., [3, 7, 8, 15, 23, 34], where the poromechanical equilibrium and the Darcy flow sub-problem are addressed independently iterating until convergence. In such approaches, suitable splittings are necessary to warrant unconditional convergence, such as the algorithms built on the so-called fixed-stress split scheme [36, 37, 44]. Fewer works are available for unconditionally stable fully implicit simulations. Spectrally equivalent block diagonal preconditioners were first proposed in [41], further discussed in [27], and recently generalized to block triangular approaches [1, 2], while families of block preconditioners were advanced in [10, 16, 21].

Recently, Ferronato et al. [17] have proposed a general algebraic framework to classify block preconditioners for coupled multi-physics simulations. Three-field poroelasticity is a classical example of such a problem, giving rise to a $3 \times 3$ block discrete system. The framework is based on the algebraic definition of a decoupling operator which reduces the fully coupled problem to a block diagonal matrix containing the matrix Schur complements related

to the physics-based separation of the unknowns by field type. In this work, we focus on the use of sparse approximate inverses to compute explicitly the decoupling factors and the resulting Schur complements. Other similar works are currently ongoing, such as the idea of using the approximate ideal restriction (AIR) in a reduction-based algebraic multigrid framework [43].

The paper is organized as follows. The three-field poroelasticity equations are briefly reviewed, deriving the structure of the discrete system of interest. After introducing the general framework used for the preconditioner definition, two methods are advanced for computing an explicit approximation of the decoupling operators. Potential and drawbacks of the proposed strategies are investigated, leading to the development of a third approach able to combine the most attractive features of the two methods. Numerical results are presented for real-world large-sized applications, dealing with geotechnical and petroleum engineering. A final discussion on the lessons learned from the numerical experiments closes the presentation.

## 2 Model and methods

### 2.1 Mathematical model

For a given bounded closed domain $\overline{\Omega} = \Omega \cup \Gamma \in \mathbb{R}^3$, with $\Omega$ an open set and $\Gamma$ its boundary, time interval $\mathbb{I} = (0, t_{\max}]$, and volumetric fluid source $s$, we consider the following strong form of Biot's poroelasticity equations [16, 26, 45, 46]: find the displacement vector $\boldsymbol{u} : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}^3$, the Darcy velocity $\boldsymbol{q} : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}^3$, and the excess pore pressure $p : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}$ such that:

$$-\operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}, p) = \mathbf{0} \quad \text{in } \Omega \times \mathbb{I} \quad \text{(equilibrium)}, \tag{1a}$$

$$\mu_f \boldsymbol{\kappa}^{-1} \cdot \boldsymbol{q} + \operatorname{grad} p = 0 \quad \text{in } \Omega \times \mathbb{I} \quad \text{(Darcy's law)}, \tag{1b}$$

$$\dot{\xi}(\boldsymbol{u}, p) + \operatorname{div} \boldsymbol{q} = s \quad \text{in } \Omega \times \mathbb{I} \quad \text{(mass balance)}, \tag{1c}$$

In Eq. (1), $\boldsymbol{\sigma} = (\mathbf{C} : \boldsymbol{\epsilon}(\boldsymbol{u}) - bp\mathbf{1})$ is the total Cauchy stress tensor, with $\boldsymbol{\epsilon}(\boldsymbol{u}) = \operatorname{sym}(\operatorname{grad} \boldsymbol{u})$ the infinitesimal strain tensor, $\mathbf{C}$ the rank-4 elasticity tensor, $b$ the Biot coefficient, and $\mathbf{1}$ the rank-2 identity tensor; $\mu_f$ and $\boldsymbol{\kappa}$ are the fluid viscosity and the rank-2 permeability tensor, respectively; $\xi = \left(b \operatorname{div} \boldsymbol{u} + \frac{1}{M} p\right)$ is the fluid content increment, with $M$ the Biot modulus. In Eq. 1c the superposed dot, $\dot{()}$, is used to denote a material time derivative with respect to the motion of the solid phase. Introducing two disjoint partitions of the domain boundary such that $\Gamma = \overline{\Gamma_u^D \cup \Gamma_\sigma^N} = \overline{\Gamma_p^D \cup \Gamma_q^N}$, we assume, without loss of generality, homogeneous Dirichlet boundary conditions $\boldsymbol{u} = \mathbf{0}$ on $\Gamma_u^D \times \mathbb{I}$ and homogeneous Neumann conditions $\boldsymbol{q} \cdot \boldsymbol{n} = 0$ on $\Gamma_q^N \times \mathbb{I}$, along with Neumann conditions $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}^N$ on $\Gamma_\sigma^N \times \mathbb{I}$ and Dirichlet conditions $p = p^D$ on $\Gamma_p^D \times \mathbb{I}$, where $\boldsymbol{n}$ is the outer normal vector for $\Gamma$. Appropriate initial conditions $\boldsymbol{u}_0$, $\boldsymbol{q}_0$, and $p_0$ complete the formulation.

Let $\boldsymbol{H}_0^1(\Omega) = [H_0^1(\Omega)]^3$ denote the Sobolev space of vector functions satisfying displacement homogeneous Dirichlet conditions on $\Gamma_u^D$ and with first derivatives in $L^2(\Omega)$; let $\boldsymbol{H}_0(\operatorname{div}; \Omega)$ be the Sobolev space of vector functions in $\boldsymbol{L}^2(\Omega) = [L^2(\Omega)]^3$ whose divergence is in $L^2(\Omega)$ satisfying homogeneous flux conditions on $\Gamma_q^N$; and let $\boldsymbol{\mathcal{U}}^h \subset \boldsymbol{H}_0^1(\Omega)$, $\boldsymbol{\mathcal{Q}}^h \subset \boldsymbol{H}_0(\operatorname{div}; \Omega)$, $\mathcal{P}^h \subset L^2(\Omega)$ denote finite-dimensional subspaces. The semidiscrete Galerkin variational statement of Eq. 1 reads: find $\{\boldsymbol{u}^h(t), \boldsymbol{q}^h(t), p^h(t)\} \in \boldsymbol{\mathcal{U}}^h \times \boldsymbol{\mathcal{Q}}^h \times \mathcal{P}^h$, $t \in \mathbb{I}$ such that for all $\{\boldsymbol{\eta}, \boldsymbol{\psi}, \chi\} \in \boldsymbol{\mathcal{U}}^h \times \boldsymbol{\mathcal{Q}}^h \times \mathcal{P}^h$:

$$(\boldsymbol{\epsilon}(\boldsymbol{\eta}), \mathbf{C}_{\text{dr}} : \boldsymbol{\epsilon}(\boldsymbol{u}^h))_\Omega - (\operatorname{div} \boldsymbol{\eta}, bp^h)_\Omega = \int_{\Gamma_\sigma^N} \boldsymbol{\eta} \cdot \boldsymbol{t}^N \, \mathrm{d}\Gamma, \tag{2a}$$

$$(\boldsymbol{\psi}, \mu_f \boldsymbol{\kappa}^{-1} \cdot \boldsymbol{q}^h)_\Omega - (\operatorname{div} \boldsymbol{\psi}, p^h)_\Omega = -\int_{\Gamma_p^D} \boldsymbol{\psi} \cdot \boldsymbol{n} p^D \, \mathrm{d}\Gamma, \tag{2b}$$

$$(\chi, b \operatorname{div} \dot{\boldsymbol{u}}^h)_\Omega + (\chi, \operatorname{div} \boldsymbol{q}^h)_\Omega + (\chi, \frac{1}{M} \dot{p}^h)_\Omega = (\chi, s)_\Omega. \tag{2c}$$

In Eq. 2, the notation $(\cdot, \cdot)_\Omega$ denotes the $L^2$-inner products of scalar functions in $L^2(\Omega)$, vector functions in $[L^2(\Omega)]^3$, or second-order tensor functions in $[L^2(\Omega)]^{3\times 3}$, as appropriate. Introducing the bases $\{\boldsymbol{\eta}_i\}$, $\{\boldsymbol{\psi}_j\}$, and $\{\chi_k\}$ for $\boldsymbol{\mathcal{U}}^h$, $\boldsymbol{\mathcal{Q}}^h$, and $\mathcal{P}^h$, respectively, we define the discrete approximations to displacement, Darcy's velocity, and pressure as:

$$\boldsymbol{u}^h = \sum_i u_i \boldsymbol{\eta}_i, \quad \boldsymbol{q}^h = \sum_j q_j \boldsymbol{\psi}_j, \quad p^h = \sum_k p_k \chi_k. \tag{3}$$

Substituting the above expressions in Eq. 2 and requiring that the semidiscrete variational equations are satisfied for each function of the bases themselves produce the following system of differential algebraic equations:

$$\begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ 0 & B^T & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \\ \mathbf{p} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ Q^T & 0 & P \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_\sigma^N \\ \mathbf{g}_p^D \\ \mathbf{h}_s \end{Bmatrix}, \tag{4}$$

with $\mathbf{u} = \{u_i\}$, $\mathbf{q} = \{q_j\}$, and $\mathbf{p} = \{p_k\}$. In Eq. 4, $K$ is the classical small displacement stiffness matrix, $A$ is the (scaled) velocity mass matrix, $P$ is the (scaled) pressure mass matrix, $Q$ is the poromechanical coupling block, $B$ is the Gram matrix, and vectors $\mathbf{f}_\sigma^N$, $\mathbf{g}_p^D$, and $\mathbf{h}_s$ come from the integration of the total traction Neumann conditions, the pressure Dirichlet boundary conditions, and the source term, respectively. Finally, the system in Eq. 4 is numerically integrated in time by the classic $\theta$-method. Advancing the

solution from discrete time level $t_n$ to $t_{n+1}$ requires the solution of the linear algebraic system $\mathbf{Ax} = \mathbf{b}$, with:

$$\mathbf{A} = \begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ Q^T & \gamma B^T & P \end{bmatrix}, \tag{5}$$

where $\mathbf{x}$ collects $\mathbf{u}_{n+1}$, $\mathbf{q}_{n+1}$, and $\mathbf{p}_{n+1}$, $\mathbf{b}$ is readily obtained as a function of the right-hand side vectors and the solution at step $n$, and $\gamma = \theta \Delta t_n$, being $\Delta t_n$ the time integration step size $(t_{n+1} - t_n)$ and $\theta$ a real parameter $(1/2 \leq \theta \leq 1)$.

In this work, we focus on the preconditioning of the three-field poroelastic matrix $\mathbf{A}$ in the form of Eq. 5 produced using piecewise trilinear ($\mathbb{Q}_1$), lowest order Raviart-Thomas ($\mathbb{RT}_0$), and piecewise constant ($\mathbb{P}_0$) finite element spaces for $\mathcal{U}^h$, $\mathcal{Q}^h$, and $\mathcal{P}^h$, respectively. We observe that the $\mathbb{Q}_1$-$\mathbb{RT}_0$-$\mathbb{P}_0$ discretization does not not intrinsically satisfy the inf-sup stability in the undrained limit [41, 46] and requires some kind of stabilization, e.g., [46]. Nevertheless, as far as the algebraic structure of $\mathbf{A}$ remains the one of Eq. 5, the proposed preconditioning strategy can be applied seamlessly.

### 2.2 Preconditioning framework

Matrix in Eq. 5 is the discrete outcome of a three-field coupled multi-physics problem. A fully algebraic preconditioner can be developed for such matrices following the framework recently introduced in [17]. Let us define the decoupling operators $\mathbf{G}$ and $\mathbf{F}$:

$$\mathbf{G} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ G_1 & G_2 & I \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} I & 0 & F_1 \\ 0 & I & F_2 \\ 0 & 0 & I \end{bmatrix}, \tag{6}$$

such that

$$\mathbf{GAF} = \mathbf{S}, \tag{7}$$

with $\mathbf{S}$ a $3 \times 3$ block diagonal matrix. It is easy to verify that Eq. 7 holds true with $\mathbf{G}$ and $\mathbf{F}$ defined as in Eq. 6 if and only if:

$$\begin{cases} F_1 = K^{-1} Q \\ F_2 = A^{-1} B \end{cases}, \qquad \begin{cases} G_1 = -F_1^T \\ G_2 = -\gamma F_2^T \end{cases}. \tag{8}$$

The resulting diagonal blocks of the decoupled matrix $\mathbf{S}$ are the Schur complements with respect to the physics-based partitioning by field type of $\mathbf{A}$, namely:

$$S_1 = K, \qquad S_2 = A, \qquad S_3 = P + F_1^T K F_1 + \gamma F_2^T A F_2. \tag{9}$$

The exact inverse of $\mathbf{A}$ is therefore:

$$\mathbf{A}^{-1} = \mathbf{F} \mathbf{S}^{-1} \mathbf{G}. \tag{10}$$

Quite obviously, $\mathbf{A}^{-1}$ in Eq. 10 is dense and cannot be computed exactly. However, approximating $F_1$, $F_2$, and the application of the inverses of $S_1$, $S_2$, and $S_3$, allows for defining an algebraic preconditioner for $\mathbf{A}$.

The decoupling blocks $F_1$ and $F_2$ can be approximated either *implicitly* or *explicitly*. Using an implicit approximation requires the definition of two inner preconditioners, $M_K^{-1}$ and $M_A^{-1}$, for $K$ and $A$, respectively:

$$M_K^{-1} \simeq K^{-1}, \qquad M_A^{-1} \simeq A^{-1}. \tag{11}$$

This way $F_1$ and $F_2$ do not need to be formed, but their application to a vector is computed by a matrix-by-vector product and an inner preconditioner application. An advantage of this approach relies on the fact that the inner preconditioners can be selected independently by taking into account the specific properties of $K$ and $A$. For instance, inner AMG algorithms can be used to exploit the ellipticity of the elastic and divergence operators. Another effective choice for $A$ only, if arising from structured grids, may be a simple diagonal or ILU(0) preconditioner. The same inner preconditioners are also used to replace the application of $S_1^{-1}$ and $S_2^{-1}$ (see Eq. 9). A drawback of such an approach, however, is that the third-level Schur complement $S_3$ now reads:

$$\begin{aligned} S_3 &= P + Q^T M_K^{-1} K M_K^{-1} Q + \gamma B^T M_A^{-1} A M_A^{-1} B \\ &\simeq P + Q^T M_K^{-1} Q + \gamma B^T M_A^{-1} B, \end{aligned} \tag{12}$$

and can be neither computed nor inverted in simple and efficient ways. This limitation can be sometimes bypassed by defining physics-based Schur complement approximations preventing the direct use of Eq. 12. For example, Castelletto et al. [10] suggest using the so-called fixed-stress matrix to replace $Q^T M_K^{-1} Q$, i.e., a diagonal mass matrix weighted with the inverse of the local mechanical parameters, and the inverse of a lumped variant of $A$ in the computation of $B^T M_A^{-1} B$. This implies using what is sometimes defined as a "mixed" approach, e.g., see [5, 18, 32], where different approximations are used for the inner blocks and the related Schur complement.

Alternatively, the decoupling blocks $F_1$ and $F_2$ can be computed explicitly. Since these blocks are usually dense, a sparse approximation must be enforced, then the resulting Schur complement $S_3$ of Eq. 9 can be computed by a sparse matrix-by-matrix product. In this work, we want to investigate the effectiveness of this approach, where the explicit approximation of $F_1$ and $F_2$ is computed by using approximate inverse-based techniques.

## 3 Approximate inverse-based approach

According to Eq. 8, the decoupling blocks $F_1$ and $F_2$ are defined as the solution of multiple right-hand side systems

with the SPD matrices $K$ and $A$. Denoting as $C[:,k]$ the $k$th column of a matrix $C$, Eq. 8 can be recast as:

$$\begin{cases} K F_1[:,k] = Q[:,k] \\ A F_2[:,k] = B[:,k] \end{cases}, \qquad k = 1,\dots,n_p, \tag{13}$$

being $n_p$ the number of pressure unknowns. Every column of $F_1$ and $F_2$ is computed so as to satisfy some sparsity constraints.

For the sake of generality, let us consider the solution to the system $C\mathbf{f} = \mathbf{b}$, where $C \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are sparse. If we want to retain a prescribed sparsity for the solution vector $\mathbf{f} \in \mathbb{R}^n$, we can define a set $\mathcal{J} \subset \{1,2,\dots,n\}$ of positions of the non-zeroes of $\mathbf{f}$ (Fig. 1), with $|\mathcal{J}| \ll n$. Hence, only the restriction $C[:,\mathcal{J}] \in \mathbb{R}^{n\times|\mathcal{J}|}$ of $C$, consisting of the columns of $C$ whose index $j \in \mathcal{J}$, is needed to compute the product $C\mathbf{f}$. The matrix $C[:,\mathcal{J}]$ can be further restricted only to the rows where at least a non-zero entry lies in one of the selected columns. Denoting as $\mathcal{I} \subset \{1,2,\dots,n\}$ the set of indices of such rows (Fig. 1), the native system $C\mathbf{f} = \mathbf{b}$ can be reduced to:

$$C[\mathcal{I},\mathcal{J}]\mathbf{f}[\mathcal{J}] = \mathbf{b}[\mathcal{I}], \tag{14}$$

where $\mathbf{f}[\mathcal{J}]$ and $\mathbf{b}[\mathcal{I}]$ are the restrictions of $\mathbf{f}$ and $\mathbf{b}$ to the sets of components with indices in $\mathcal{J}$ and $\mathcal{I}$, respectively. Since $|\mathcal{J}|, |\mathcal{I}| \ll n$, the solution to the system in Eq. 14 is practically inexpensive if compared with the full problem $C\mathbf{f} = \mathbf{b}$.

This idea, which was originally introduced for the algebraic computation of a sparse approximate inverse by keeping the identity as a multiple right-hand side [20, 24, 31, 38], is used to obtain an explicit approximation of the decoupling blocks $F_1$ and $F_2$. To avoid the solution to over- or under-determined rectangular systems, we define the sets:

$$\mathcal{K}_r^{(k)} = \mathcal{I}_r^{(k)} \cup \mathcal{J}_r^{(k)}, \qquad r = 1,2, \qquad k = 1,\dots,n_p, \tag{15}$$

where $\mathcal{J}_r^{(k)} \subset \{1,2,\dots,n_r\}$ is the set of positions of the non-zero entries retained in the $k$th column of the decoupling block $F_r$, $\mathcal{I}_r^{(k)} \subset \{1,2,\dots,n_r\}$ is the corresponding set of row indices with at least a matrix non-zero entry in the columns $j \in \mathcal{J}_r^{(k)}$, and $n_r$ is either $n_u$ or $n_q$ for $r = 1$ or 2, respectively, being $n_u$ and $n_q$ the number of displacement and velocity unknowns. With the definition of Eq. 15, the systems Eq. 13 can be replaced by:

$$\begin{cases} K[\mathcal{K}_1^{(k)},\mathcal{K}_1^{(k)}]\, F_1[\mathcal{K}_1^{(k)},k] = Q[\mathcal{K}_1^{(k)},k] \\ A[\mathcal{K}_2^{(k)},\mathcal{K}_2^{(k)}]\, F_2[\mathcal{K}_2^{(k)},k] = B[\mathcal{K}_2^{(k)},k] \end{cases}, \quad k = 1,\dots,n_p, \tag{16}$$

thus obtaining a sparse explicit approximation of $F_1$ and $F_2$. Notice that the systems in Eq. 16 can be solved independently column by column. Once $F_1$ and $F_2$ are available from Eq. 16, the third-level Schur complement $S_3$ of Eq. 9 can be computed explicitly as well. Three inner preconditioners for $S_1$, $S_2$, and $S_3$, namely $M_1^{-1}$, $M_2^{-1}$, and $M_3^{-1}$, are eventually introduced to complete the algorithm. The final approximate inverse-based block preconditioner $\mathbf{M}^{-1}$ for $\mathbf{A}$ reads:

$$\mathbf{M}^{-1} = \begin{bmatrix} I & 0 & F_1 \\ 0 & I & F_2 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} M_1^{-1} & 0 & 0 \\ 0 & M_2^{-1} & 0 \\ 0 & 0 & M_3^{-1} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -F_1^T & -\gamma F_2^T & I \end{bmatrix}. \tag{17}$$

The key factor for the quality and effectiveness of $\mathbf{M}^{-1}$ in Eq. 17 is the selection of the sets $\mathcal{K}_r^{(k)}$ for the sparse explicit computation of $F_1$ and $F_2$. In the sequel, we investigate three options:

1. Static selection of $\mathcal{K}_r^{(k)}$ based on the structure of $Q$ and $B$;
2a. Dynamic selection of $\mathcal{K}_r^{(k)}$ during the $F_r$ computation;
2b. Dynamic selection of $\mathcal{K}_r^{(k)}$ starting from an initial static pattern provided by $Q$ and $B$.

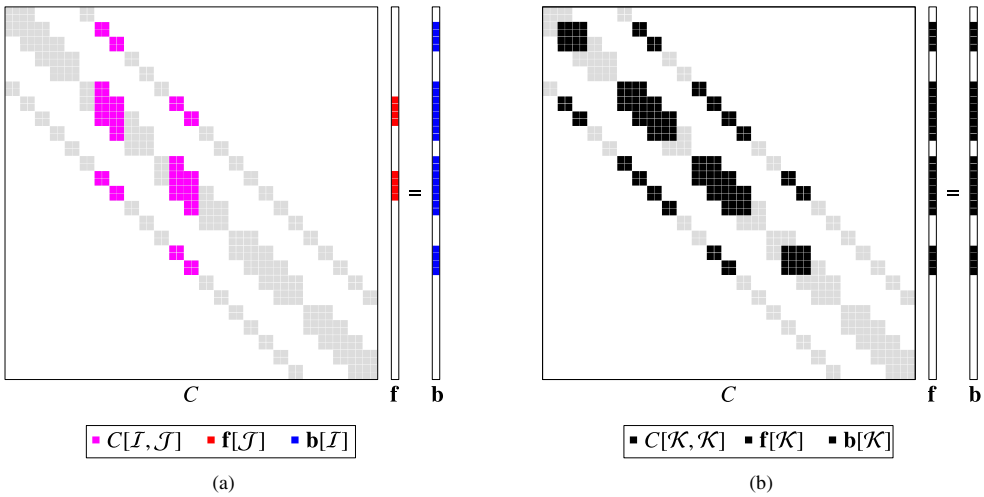


**Fig. 1** Schematic representation of a linear system solution subject to sparsity constraints, before (**a**) and after (**b**) pattern symmetrization, i.e., forming $\mathcal{K} = \mathcal{I} \cup \mathcal{J}$. Here, $\mathcal{I}$ and $\mathcal{J}$ are the sets for rows and columns, respectively. The pattern of matrix $C$ is shown in gray

### 3.1 Method 1: static approach

The decoupling factors $F_r$, $r = 1, 2$, are computed using a prescribed column-wise non-zero pattern $\mathcal{K}_r^{(k)}$, $k = 1, \ldots, n_p$. An effective a priori pattern for the $k$th column of $F_r$ can be defined by considering the physical meaning of the systems in Eq. 13. Let us consider, for instance, the first set of equations in Eq. 13. It states that the $k$th column of $F_1$ can be regarded as the discrete displacement vector in the solid body with stiffness matrix $K$ induced by loads defined in the $k$th column of $Q$—namely, loads produced by a unit pressure change at the $k$th pressure degree of freedom (DoF) $p_k$ and applied to mesh nodes associated with displacement DoFs coupled to $p_k$. Similarly, the second set of equations in Eq. 13 states that the $k$th column of $F_2$ is the discrete Darcy velocity vector in the porous volume with conductivity matrix $A$ generated by the pressure gradients defined in the $k$th column of $B$—namely, pressure gradients due to a unit pressure change at $p_k$ and applied to mesh faces associated with velocity DoFs coupled to $p_k$. A quite natural choice for $\mathcal{K}_r^{(k)}$ is therefore the set of non-zero entries of $Q[:, k]$ and $B[:, k]$, which can be immediately derived from the grid topology and the relation tables *node-to-element* and *face-to-element* [49].

However, the vectors $F_1[:, k]$ and $F_2[:, k]$ are actually expected to be non-zero in all entries, with decaying values as we move farther from the loaded nodes. To include a larger number of possibly significant entries into $F_1[:, k]$ and $F_2[:, k]$, denser non-zero patterns can be built. The expansion of $\mathcal{K}_r^{(k)}$ from the non-zero pattern of the $k$th column of either $Q$ or $B$ is topologically based; i.e., we collect neighbors of neighbors and so on for a given number of levels. The advantage of this approach is twofold:

1. The static patterns are simply defined by selecting an integer value associated to the expansion level, i.e., 1 for the original pattern, 2 for the neighbors, 3 for the neighbors of the neighbors, etc.;
2. The pattern computation is straightforward, very cheap, and can be defined as a function of the grid only.

As a drawback, the number of connections can become soon quite large, also for a small number of steps, say 2, in the graph. The entries of $F_1$ and $F_2$ are computed column-wise by solving the relatively small dense linear systems in Eq. 16, formed by gathering rows and columns of $K$ and $A$ belonging to the selected pattern $\mathcal{K}_r^{(k)}$.

### 3.2 Method 2: dynamic approach

The decoupling factors $F_r$, $r = 1, 2$, are computed without any a priori definition of the column-wise non-zero patterns $\mathcal{K}_r^{(k)}$, $k = 1, \ldots, n_p$. These are the outcome of an adaptive algorithm selecting and computing the most important entries in each column. Several dynamic algorithms have been already advanced in the context of sparse approximate inverse computation, where the multiple right-hand sides of systems are the columns of the identity matrix. For instance, dynamic pattern selection procedures have been proposed for both symmetric and non-symmetric matrices in [24, 28, 29, 35].

In this work, we use an adaptive algorithm for the non-zero pattern selection inspired by the one introduced in [30, 33]. Starting from an arbitrary initial guess $\mathcal{K}_r^{(k),0}$, we compute $F_r[\mathcal{K}_r^{(k),0}, k]$ by solving the systems Eq. 16 and then obtain the residuals:

$$\begin{cases} \mathbf{r}_1^{(k),0} = K[:, \mathcal{K}_1^{(k),0}]\, F_1[\mathcal{K}_1^{(k),0}, k] - Q[:, k] \\ \mathbf{r}_2^{(k),0} = A[:, \mathcal{K}_2^{(k),0}]\, F_2[\mathcal{K}_2^{(k),0}, k] - B[:, k] \end{cases}, \quad k = 1, \ldots, n_p, \tag{18}$$

The non-zero pattern $\mathcal{K}_r^{(k),0}$ is enlarged by adding $\rho_{F,r}$ positions corresponding to the largest components of $\mathbf{r}_r^{(k)}$, thus obtaining the augmented pattern $\mathcal{K}_r^{(k),1}$. The procedure is iterated by computing the new columns $F_r[\mathcal{K}_r^{(k),1}, k]$ and the new residuals, so as to build $\mathcal{K}_r^{(k),2}$, and so on. The column-wise search can be stopped when either a maximum number of entries are added or some norm of $\mathbf{r}_r^{(k)}$ is smaller than a prescribed tolerance.

The advantage of this approach is that the density of the decoupling factors can be automatically adapted column by column to obtain a prescribed quality by selecting and computing the most significant entries only. By contrast, its cost can rapidly increase with the number of computed entries. It can be easily shown that the computational cost of this dynamic approach grows with the fourth power of the density of each column, while with the static approach previously described it increases with the third power only [33]. This can become a major drawback when a big number of entries is required to obtain the prescribed quality for the sparse approximation of $F_r$.

The proposed adaptive approach needs an initial non-zero pattern $\mathcal{K}_r^{(k),0}$. A fully dynamic approach can start from the empty pattern (Method 2a). Another option, however, relies on using a non-empty initial guess, such as for instance the non-zero pattern of $Q[:, k]$ and $B[:, k]$ (method 2b). The latter can be therefore regarded as a combination of the static and dynamic approaches, where the first-level non-zero pattern of $Q[:, k]$ and $B[:, k]$ is dynamically improved column-wise only where necessary.

## 4 Numerical results

To assess the numerical performance of the presented methods, we use three test cases:

- `Mandel80`: this is a standard benchmark test case in poroelasticity. It consists of a porous slab bounded by rigid, frictionless, impermeable plates, with sizes $a = b = 1.0$ m and $c = 0.1$ m (Fig. 2). Outer boundaries are traction-free, drained, and kept to ambient pressure at all times. A structured grid is used with spacing equal to $a/80$. Due to symmetry, a quarter only of the $\hat{x} - \hat{z}$ plane is modeled. The medium is characterized by homogeneous elastic properties, namely Young's modulus $E = 1.0 \times 10^6$ Pa, Poisson's ratio $\nu = 0.2$, Biot's coefficient $b = 1.0$, incompressible fluid, and solid constituents—i.e., Biot's modulus is such that $M^{-1} = 0$ Pa$^{-1}$—, isotropic permeability $\kappa = 1.0 \times 10^{-12}$ m$^2$, and fluid viscosity $\mu_f = 1.0 \times 10^{-3}$ Pa·s.
- `Treporti`: this is a real-world problem dealing with the consolidation of a shallow heterogeneous formation due to the construction of a trial embankment. A 5-year long loading/unloading test is simulated with the aim at characterizing the geomechanical properties of the sedimentary deposits at the Venice coastland [9]. Alternating sandy, silty, and clayey layers down to 60-m depth is implemented according to the available stratigraphic information (Fig. 3), with an intrinsic permeability and uniaxial compressibility varying in the ranges $[5.1 \times 10^{-16}, 5.1 \times 10^{-15}]$ m$^2$ and $[2.05 \times 10^{-2}, 1.05 \times 10^{0}]$ MPa$^{-1}$, respectively.
- `Reservoir`: the top 16 layers of the SPE10 dataset [13] are used to characterize a compacting reservoir subject to single-phase flow. The computational mesh consists of $60 \times 220 \times 16$ hexahedral elements in $x$-, $y$- and $z$-direction, respectively (Fig. 4). The porous medium is populated with homogeneous elastic properties, namely $E = 8.3 \times 10^3$ MPa, $\nu = 0.3$, $b = 1.0$, and $M^{-1} = 0$ Pa$^{-1}$. All boundaries are constrained by roller boundary conditions, except for the top one which is traction-free.

Table 1 summarizes the size and number of non-zeros of the resulting matrix blocks.

First, a detailed sensitivity analysis is carried out for methods 1 and 2a using the `Mandel80` test case with $\gamma = 9.0$ s, i.e., $\theta = 1$ and $\Delta t = 10^{-2} t_c$, being $t_c$ the characteristic consolidation time. Then, the preconditioner robustness to a variation of the physical parameters characterizing the material properties is investigated in the same test case addressed for the sensitivity analysis. Finally, the computational performance obtained in all test cases is analyzed for different values of $\gamma$, i.e., the time-step size. Reported CPU times are the best performance obtained by properly tuning the setup parameters for each combination of $\gamma$, matrix, and method. We elected to use the restarted GMRES(250) algorithm [47] as Krylov method, with the iterations stopped whenever either the 2-norm of the residual is reduced by 8 orders of magnitude or the number of iterations is larger than 1000. In the latter case, a solver failure is accounted for. The null vector is taken as the initial guess. The code is implemented in Fortran90 exploiting a shared-memory parallelism through OpenMP directives [12]. The reported times are obtained on a machine equipped with one Intel(R) Core(TM) i7-8700 processor at 3.20 GHz and 32 Gbyte of RAM. All configurations are run using 4 threads. Results are compared in terms of:

- $\mu$: density of the preconditioner, i.e., the sum of non-zeroes for every matrix involved in the preconditioner application divided by the number of non-zero entries of the original matrix. This parameter gives an indication of the application cost per iteration;

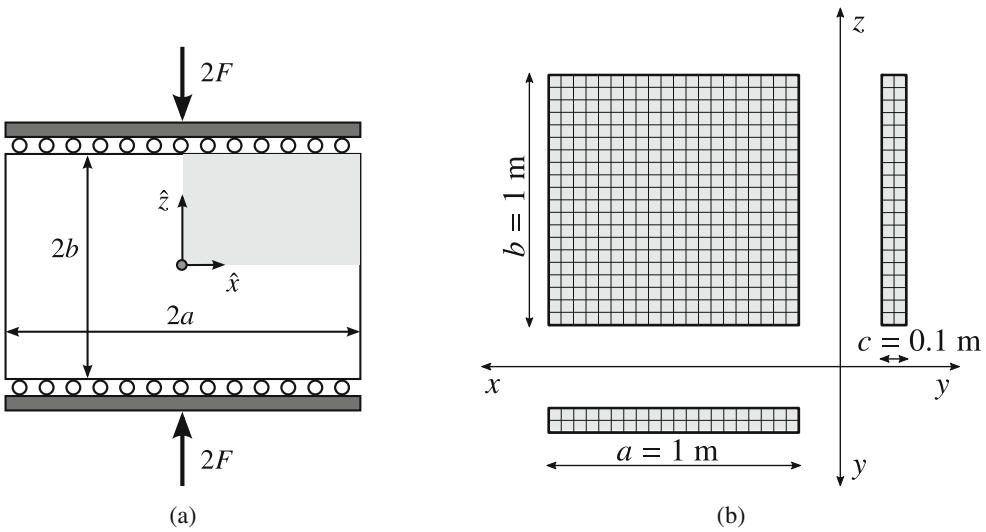


**Fig. 2** `Mandel80` test case: sketch and computational grid

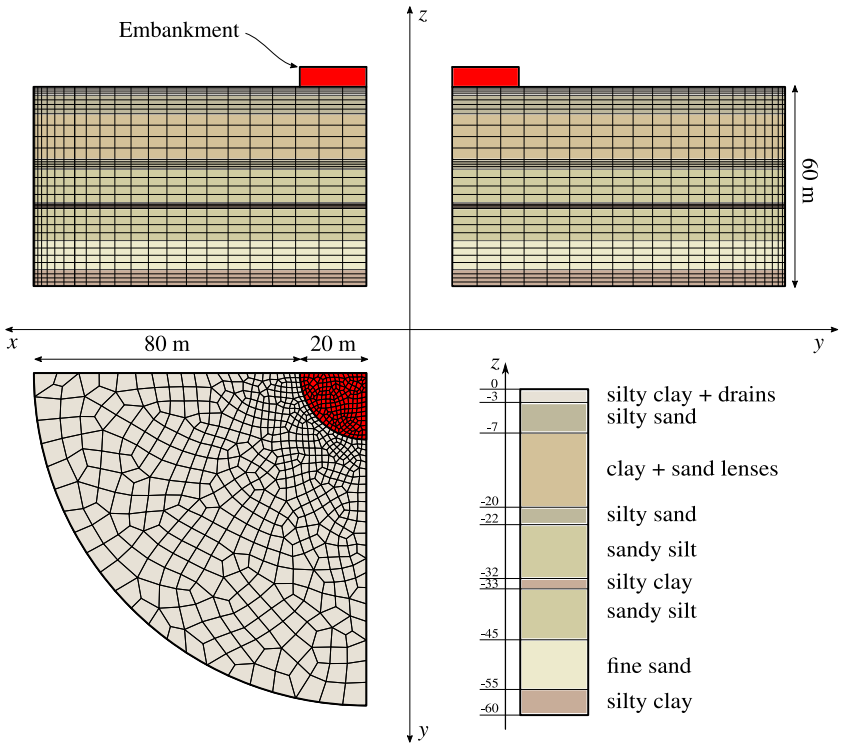


**Fig. 3** `Treporti` test case: sketch of the simulated domain

– $n_{it}$: number of iterations to convergence;
– $T_p^{(1)}$: setup time in seconds for all the stages that are independent on $\gamma$. In a full-transient poroelastic simulation, such a setup is needed just once at the beginning of the process and can be recycled at every system solution;

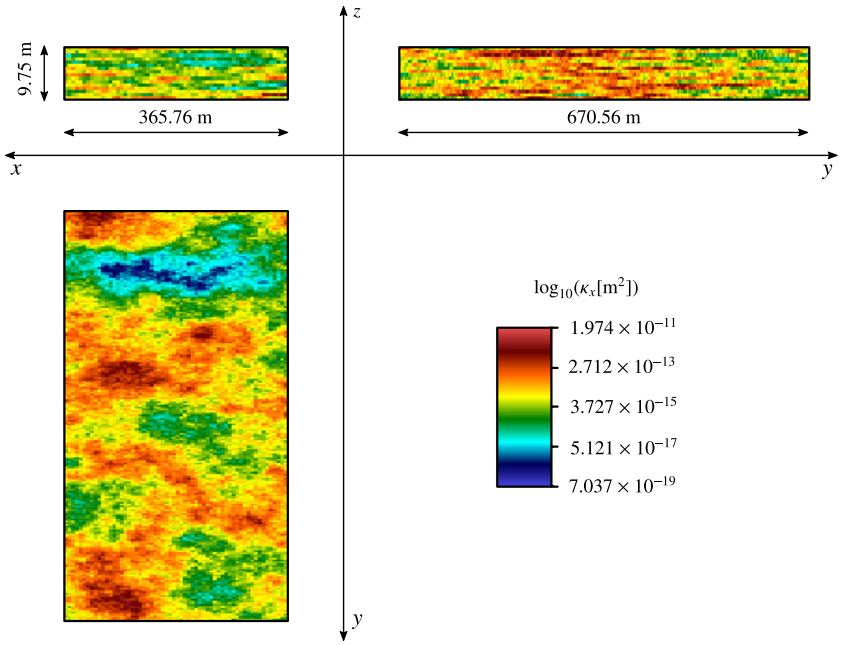


**Fig. 4** `Reservoir` test case: sketch of the simulated domain

**Table 1** Test cases: size and non-zeroes for all matrices

| | `Mandel80` | | | `Treporti` | | | `Reservoir` | | |
|---|---|---|---|---|---|---|---|---|---|
| | # rows | # cols | # non-zeroes | # rows | # cols | # non-zeroes | # rows | # cols | # non-zeroes |
| $K$ | 177,147 | 177,147 | 13,068,225 | 178,923 | 178,923 | 13,794,039 | 687,531 | 687,531 | 52,761,681 |
| $Q$ | 177,147 | 51,200 | 1,228,800 | 178,923 | 55,368 | 1,328,832 | 687,531 | 211,200 | 5,068,800 |
| $A$ | 161,280 | 161,280 | 1,697,280 | 170,257 | 170,257 | 1,831,297 | 651,280 | 651,280 | 6,987,280 |
| $B$ | 161,280 | 51,200 | 307,200 | 170,257 | 55,368 | 332,208 | 651,280 | 211,200 | 1,267,200 |
| $P$ | 51,200 | 51,200 | 51,200 | 55,368 | 55,368 | 55,368 | 211,200 | 211,200 | 211,200 |
| Total | 389,627 | 389,627 | 17,888,705 | 404,548 | 404,548 | 19,002,784 | 1,550,011 | 1,550,011 | 72,632,161 |

- $T_p^{(2)}$: setup time in seconds for all the stages that depend on $\gamma$. This cost has to be spent whenever $\gamma$ changes during a transient simulation;
- $T_s$: time in seconds spent to iterate until convergence.

To conclude this section, we notice that all matrix computations for the preconditioner setup, such as products and merges, are performed approximately by using fill-in and/or dropping tolerance thresholds. This allows to retain the more significant entries and at the same time maximize the preconditioner sparsity. In particular, a magnitude-based dropping is applied row- or column-wise to $F_1$ and $F_2$ (for the Schur complement $S_3$ computation only), and $S_3$ (before computing $M_3^{-1}$). Usually, the dropping threshold is selected in the range $[10^{-3}, 10^{-2}]$ relative to the 2-norm of the full row or column.

### 4.1 Sensitivity analysis

Methods 1 and 2a are tested in the `Mandel80` test case with $\gamma = 9.0$ s. The objective of this analysis is to identify the role played by the setup parameters required in each method, evaluating the algorithmic robustness to their variation and the possible difficulty in their selection. To this aim, the same inner preconditioners $M_1^{-1}$, $M_2^{-1}$, and $M_3^{-1}$ are used, namely an adaptive Block FSAI-Incomplete Cholesky preconditioner (ABF-IC, [30]), which has proved an effective tool for SPD matrices. However, the proposed algorithms are flexible enough to allow any other effective choice for $M_1^{-1}$, $M_2^{-1}$, and $M_3^{-1}$, and the results of the sensitivity analysis that follows are independent on that.

The setup of method 1 (see Section 3.1) requires the expansion level used to define the static pattern of $F_1$ and $F_2$. In particular, denoting by $\mathcal{P}(M)$ the non-zero pattern of a sparse matrix $M$, from an algebraic point of view the pattern of level $\eta$ for $F_1$ and $F_2$ can be computed as:

$$\mathcal{P}(F_1) = \mathcal{P}((K)^{\eta-1})\mathcal{P}(Q), \tag{19}$$

$$\mathcal{P}(F_2) = \mathcal{P}((A)^{\eta-1})\mathcal{P}(B). \tag{20}$$

where we point out that the pattern of $K$ is the same as $QQ^T$, as well as for $A$ and $BB^T$. As already observed in Section 3.1, $\eta = 1$ coincides with the original *node-to-element* and *face-to-element* connections, $\eta = 2$ includes the neighbors, $\eta = 3$ the neighbors of the neighbors, and so on.

Table 2 reports the performance of method 1 for different levels $\eta$. We can observe that the number of iterations is more than halved when $\eta$ moves from 1 to 2, but then it practically flattens for $\eta = 3$. The smallest total time for the $\gamma$-dependent stage, i.e., $T_p^{(2)} + T_s$, is reached for $\eta = 1$. The density $\mu$ grows very quickly with $\eta$, soon becoming unacceptable for practical computations. Therefore, although most of the time is actually spent in the initial setup stage, $T_p^{(1)}$, the variability range for $\eta$ can be limited to 1 or 2. Figure 5 shows the results of Table 2 relative to the performance obtained with $\eta = 2$.

Method 2a requires the selection of the control parameters for the dynamic construction of $\mathcal{P}(F_r)$, $r = 1, 2$. Following the algorithm sketched in Section 3.2, for each decoupling block we need two user-specified parameters:

- $s_{\max}$, an integer value indicating the total number of residual evaluations;
- $\rho_F$, an integer value denoting the number of entries added at each residual evaluation to enlarge the current pattern.

Hence, the number of entries added to $\mathcal{K}_r^{(k),0}$ is $|\mathcal{K}| = s_{\max} \cdot \rho_F$. The performance obtained by method 2a varying $(s_{\max}, \rho_F)$ in the range $[5, 10] \times [10, 20]$ for both $F_1$ and $F_2$ is provided in Table 3. We can note that, for the same value

**Table 2** `Mandel80` with $\gamma = 9.0$ s: performance of method 1 (static approach) with respect to level of expansion for $\mathcal{P}(F_1)$ and $\mathcal{P}(F_2)$

| $\eta$ | $\mu$ | $n_{it}$ | $T_p^{(1)}$ (s) | $T_p^{(2)}$ (s) | $T_s$ (s) |
|---|---|---|---|---|---|
| 1 | 0.56 | 149 | 0.47 | 0.14 | 4.13 |
| 2 | 3.66 | 66 | 23.05 | 2.39 | 4.03 |
| 3 | 6.43 | 48 | 208.79 | 3.63 | 4.50 |

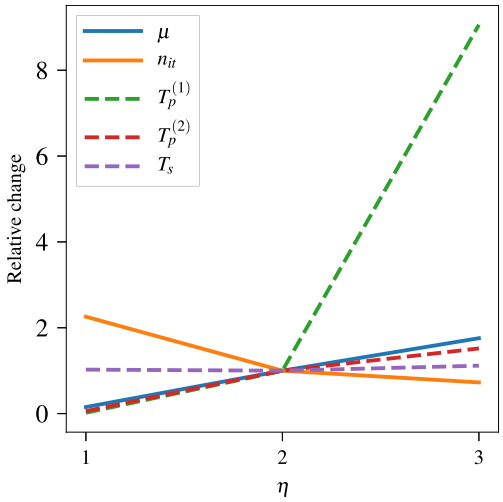


**Fig. 5** `Mandel80` with $\gamma = 9.0$ s: relative change of the performance of method 1 (static approach) with respect to $\eta = 2$

of $|\mathcal{K}|$, the best performance is generally obtained with few large steps instead of several smaller ones, i.e., small $s_{\max}$ and large $\rho_F$. Table 3 also reveals that the $\gamma$-dependent setup, $T_p^{(2)}$, and the solution stage, $T_s$, have approximately the same cost for all the tested combinations, while the $\gamma$-independent setup stage, $T_p^{(1)}$, is minimized by using few residual evaluations with more entries added at each time, i.e., again for small $s_{\max}$ and large $\rho_F$. Therefore, this choice not only improves the preconditioner effectiveness, but also provides a faster setup stage.

### 4.2 Preconditioner robustness

Biot's equations depend on several different physical parameters and it is important to investigate the robustness of the proposed preconditioners within the range of their variations. Specifically, the elastic stiffness matrix $K$ varies with the rock compressibility $c_b$, which is related to Young's modulus $E$ and Poisson's ratio $\nu$, the velocity mass matrix $A$ with the hydraulic conductivity $k$, which depends on the permeability $\kappa$ and the fluid viscosity $\mu_f$, and the pressure mass matrix $P$ with the fluid compressibility $c_f$. Since in real-world problems the compressibility of the fluid, i.e., groundwater in our test cases, cannot deviate significantly from 0, we focus on $c_b$ and $k$ only. The same test case as in the sensitivity analysis is used (`Mandel80` with $\gamma = 10^{-2}t_c$, see Section 4.1) with both $c_b$ and $k$ varied by 2 and 3 orders of magnitude, respectively, with respect to the original values $c_{b,0}$ and $k_0$ corresponding to $E$, $\nu$, $\kappa$, and $\mu_f$ reported in Section 4.

In order to emphasize the effects of the physical parameters only, we scale the global matrix $\mathbf{A}$ so that the scaled diagonal blocks have either unitary or zero Frobenius norm. Given $\mathbf{A}$ as defined in Eq. 5, the scaled matrix $\mathbf{A}_S$ reads:

$$\mathbf{A}_S = \begin{bmatrix} \alpha_K I & 0 & 0 \\ 0 & \alpha_A I & 0 \\ 0 & 0 & \alpha_P I \end{bmatrix} \begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ Q^T & \gamma B^T & P \end{bmatrix} \begin{bmatrix} \alpha_K I & 0 & 0 \\ 0 & \alpha_A I & 0 \\ 0 & 0 & \alpha_P I \end{bmatrix}, \tag{21}$$

**Table 3** `Mandel80` with $\gamma = 9.0$ s: performance of method 2a with respect to the parameters $s_{\max}$ and $\rho_F$ controlling the quality and fill-in of $F_1$ and $F_2$

| $s_{\max,1}$ | $\rho_{F,1}$ | $\lvert\mathcal{K}\rvert_1$ | $s_{\max,2}$ | $\rho_{F,2}$ | $\lvert\mathcal{K}\rvert_2$ | $\mu$ | $n_{it}$ | $T_p^{(1)}$ (s) | $T_p^{(2)}$ (s) | $T_s$ (s) |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 10 | 50 | 5 | 10 | 50 | 0.68 | 17 | 1.30 | 0.09 | 0.34 |
| | | 50 | 5 | 20 | 100 | 0.68 | 17 | 1.43 | 0.09 | 0.34 |
| | | 50 | 10 | 10 | 100 | 0.69 | 31 | 1.32 | 0.18 | 0.63 |
| | | 50 | 10 | 20 | 200 | 0.69 | 31 | 1.44 | 0.18 | 0.63 |
| 5 | 20 | 100 | 5 | 10 | 50 | 0.84 | 17 | 2.66 | 0.09 | 0.38 |
| | | 100 | 5 | 20 | 100 | 0.84 | 17 | 3.13 | 0.10 | 0.38 |
| | | 100 | 10 | 10 | 100 | 0.86 | 28 | 2.77 | 0.18 | 0.63 |
| | | 100 | 10 | 20 | 200 | 0.86 | 28 | 2.92 | 0.20 | 0.63 |
| 10 | 10 | 100 | 5 | 10 | 50 | 0.87 | 18 | 4.68 | 0.12 | 0.40 |
| | | 100 | 5 | 20 | 100 | 0.87 | 18 | 4.90 | 0.12 | 0.40 |
| | | 100 | 10 | 10 | 100 | 0.88 | 31 | 4.61 | 0.22 | 0.71 |
| | | 100 | 10 | 20 | 200 | 0.88 | 31 | 5.03 | 0.22 | 0.70 |
| 10 | 20 | 200 | 5 | 10 | 50 | 1.20 | 18 | 10.85 | 0.12 | 0.48 |
| | | 200 | 5 | 20 | 100 | 1.20 | 18 | 11.32 | 0.12 | 0.48 |
| | | 200 | 10 | 10 | 100 | 1.21 | 30 | 10.90 | 0.22 | 0.81 |
| | | 200 | 10 | 20 | 200 | 1.21 | 30 | 11.83 | 0.22 | 0.80 |

**Table 4** `Mandel80` with $\gamma = 10^{-2} t_c$: preconditioner robustness to a variation of the rock compressibility $c_b$ and the hydraulic conductivity $k$ with respect to the original values $c_{b,0}$ and $k_0$, respectively

| | | Method 1 | | Method 2a | | Method 2b | |
|---|---|---|---|---|---|---|---|
| $c_b/c_{b,0}$ | $k/k_0$ | $\mu$ | $n_{it}$ | $\mu$ | $n_{it}$ | $\mu$ | $n_{it}$ |
| 1 | 1 | 0.70 | 135 | 1.35 | 102 | 0.90 | 117 |
| $10^2$ | 1 | 0.70 | 139 | 1.35 | 103 | 0.90 | 118 |
| $10^{-2}$ | 1 | 0.70 | 138 | 1.35 | 103 | 0.90 | 117 |
| 1 | $10^3$ | 0.70 | 142 | 1.35 | 103 | 0.90 | 119 |
| 1 | $10^{-3}$ | 0.70 | 162 | 1.35 | 104 | 0.90 | 120 |

**Table 5** Computational performance obtained with method 1 (static approach)

| | $\gamma$ [s] | $\mu$ | $n_{it}$ | $T_p^{(1)}$ (s) | $T_p^{(2)}$ (s) | $T_s$ (s) |
|---|---|---|---|---|---|---|
| | $9 \cdot 10^{-2}$ | 0.56 | 145 | 0.46 | 0.13 | 3.91 |
| `Mandel80` | $9 \cdot 10^{0}$ | 0.56 | 149 | 0.47 | 0.13 | 4.22 |
| | $9 \cdot 10^{2}$ | 0.56 | 149 | 0.47 | 0.12 | 4.11 |
| | $10^{-4}$ | 4.07 | 241 | 38.36 | 4.82 | 20.92 |
| `Treporti` | $10^{-2}$ | 4.07 | 233 | 38.18 | 4.99 | 20.58 |
| | $10^{0}$ | 0.65 | 211 | 0.60 | 0.23 | 7.86 |
| | $8.64 \cdot 10^{0}$ | 0.53 | 235 | 2.36 | 0.57 | 49.90 |
| `Reservoir` | $8.64 \cdot 10^{2}$ | 0.53 | 132 | 2.35 | 0.59 | 20.62 |
| | $8.64 \cdot 10^{4}$ | 0.53 | 159 | 2.38 | 0.57 | 26.68 |

**Table 6** Computational performance obtained with method 2a (dynamic approach starting from scratch)

| | $\gamma$ (s) | $\mu$ | $n_{it}$ | $T_p^{(1)}$ (s) | $T_p^{(2)}$ (s) | $T_s$ (s) |
|---|---|---|---|---|---|---|
| | $9 \cdot 10^{-2}$ | 1.21 | 65 | 11.63 | 0.34 | 1.86 |
| `Mandel80` | $9 \cdot 10^{0}$ | 1.21 | 30 | 11.51 | 0.22 | 0.80 |
| | $9 \cdot 10^{2}$ | 1.21 | 19 | 11.73 | 0.22 | 0.51 |
| | $10^{-4}$ | 1.40 | 485 | 12.23 | 0.44 | 25.04 |
| `Treporti` | $10^{-2}$ | 1.34 | 384 | 12.19 | 0.29 | 17.90 |
| | $10^{0}$ | 1.34 | 160 | 12.25 | 0.31 | 6.73 |
| | $8.64 \cdot 10^{0}$ | 1.18 | 237 | 44.53 | 0.78 | 62.74 |
| `Reservoir` | $8.64 \cdot 10^{2}$ | 1.18 | 117 | 47.77 | 0.80 | 21.55 |
| | $8.64 \cdot 10^{4}$ | 1.18 | 85 | 43.88 | 0.82 | 13.62 |

**Table 7** Computational performance obtained with Method 2b (dynamic approach starting from static initial guess)

| | $\gamma$ [s] | $\mu$ | $n_{it}$ | $T_p^{(1)}$ [s] | $T_p^{(2)}$ [s] | $T_s$ [s] |
|---|---|---|---|---|---|---|
| | $9 \cdot 10^{-2}$ | 0.71 | 82 | 1.10 | 0.13 | 1.98 |
| `Mandel80` | $9 \cdot 10^{0}$ | 0.72 | 35 | 1.14 | 0.29 | 0.74 |
| | $9 \cdot 10^{2}$ | 0.72 | 17 | 1.06 | 0.29 | 0.36 |
| | $10^{-4}$ | 1.30 | 241 | 2.49 | 1.42 | 11.96 |
| `Treporti` | $10^{-2}$ | 1.29 | 236 | 2.47 | 1.60 | 11.70 |
| | $10^{0}$ | 0.98 | 161 | 2.96 | 0.58 | 5.95 |
| | $8.64 \cdot 10^{0}$ | 0.88 | 126 | 12.77 | 2.03 | 22.49 |
| `Reservoir` | $8.64 \cdot 10^{2}$ | 0.81 | 99 | 3.99 | 5.32 | 15.49 |
| | $8.64 \cdot 10^{4}$ | 0.79 | 80 | 4.02 | 4.81 | 11.12 |

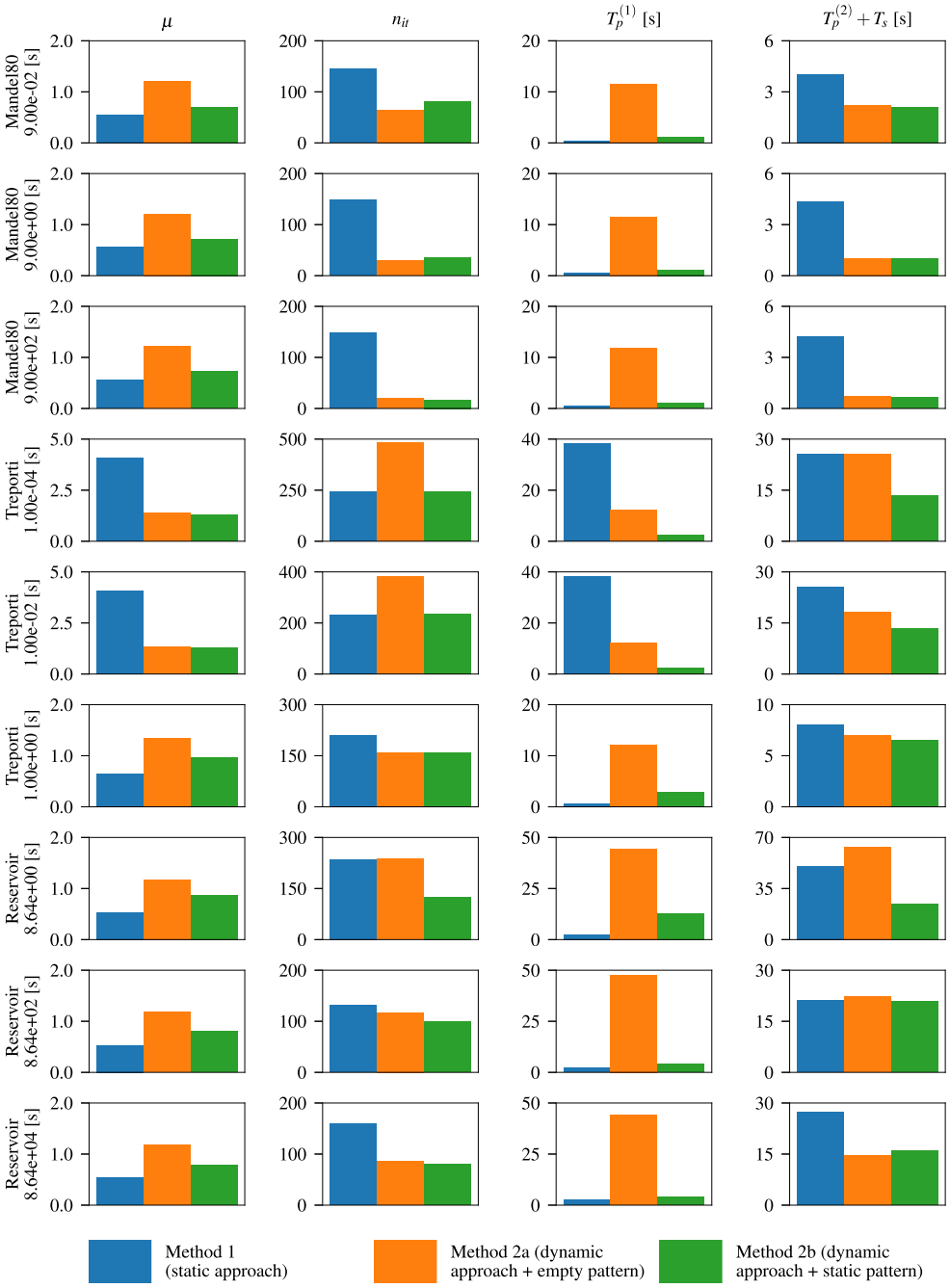


**Fig. 6** Density, iteration count, and CPU times in seconds for all the test cases of Table 1 and different $\gamma$ values. The plotted quantity is indicated on top, while on the left test case and $\gamma$ value are reported

where $\alpha_M$, $M = K, A, P$, is:

$$\alpha_M = \begin{cases} \dfrac{1}{\sqrt{\|M\|_F}} & \text{if } \|M\|_F > 0 \\ 1 & \text{if } \|M\|_F = 0 \end{cases} . \tag{22}$$

Table 4 shows that the proposed methods prove quite robust with respect to a variation of the governing physical parameters. In particular, methods 2a and 2b are completely insensitive, with iteration counts varying by less than 3%, while method 1 is slightly more sensitive, with a maximum variation of about 16% for one parameter combination. Hence, the presented approaches appear to be quite robust to wide variations in the space of the physical parameters.

An additional source of difficulties may arise also from a variable mesh and time discretization size, as well as from heterogeneous material properties, with possible significant variations of $c_b$ and $k$ in space and jumps between adjacent elements. This a typical situation met in real-world problems, which will be addressed in the test cases shown in the following section.

### 4.3 Computational performance

The computational performance of the proposed methods is finally investigated for the three test cases of Table 1 and different values of $\gamma$. Table 5, 6, and 7 show the results obtained for methods 1, 2a, and 2b, respectively. Figure 6 shows a global comparison of all significant quantities, i.e., density, iteration count, and times, for all approaches and test cases.

It can be noticed that the cost of the $\gamma$-dependent setup stage, $T_p^{(2)}$, is usually much cheaper than that of the other stages, so that it can be neglected for a general analysis. On the other hand, the cost of the $\gamma$-independent setup stage, $T_p^{(1)}$, can be usually performed just once in a full-transient poroelastic simulation consisting of hundreds or thousands time steps; hence, it is easily amortized. For this reason, the most significant performance measure is the solution cost, $T_s$, or the sum of the $\gamma$-dependent efforts, $T_s + T_p^{(2)}$.

Inspection of Fig. 6 shows that the most competitive strategy is usually method 2b, i.e., the dynamic approach starting from the first-level static initial guess. Only in the `Reservoir` case with the largest time step is method 2a slightly better than method 2b in the $\gamma$-dependent phase. Method 1 alone is never optimal; nevertheless, it is always competitive. As far as the $\gamma$-independent setup is concerned, method 1 with $\eta = 1$ and method 2b are similar and quite faster than method 2a. Finally, considering the memory footprint measured by $\mu$, method 1 with $\eta = 1$ provides the lowest density, while method 2b is generally less demanding than method 2a.

With the exception of test case `Mandel80`, which is homogeneous and discretized by a structured grid, method 2b requires the minimum iteration count to converge. As to the robustness, all the proposed methods behave similarly good, with no failures for any combination of problem and time-step size. Generally speaking, for small $\gamma$ values, the system requires more iterations to converge. In these cases, method 2b appears to be the most efficient one. For larger $\gamma$ values, the number of iterations generally decreases for all methods.

## 5 Discussion and conclusions

In this work, we investigated a family of methods based on factorized sparse approximate inverse techniques to accelerate the iterative convergence of the poroelasticity equations using a mixed three-field formulation. Taking advantage of the inherent block structure of the global system, the common idea among all approaches is to compute explicitly algebraic preconditioning operators in a black box fashion such that the preconditioned matrix tends to a three-block diagonal pattern. Overall, we compare three approaches.

- Method 1 relies on a static approach for defining the sparse non-zero pattern of the decoupling factors. Based on physical arguments, the patterns of the matrices $Q$ and $B$, and their topologically based expansions including neighbors of neighbors, are used to compute the factors decoupling the displacements and the flux unknowns from the pressure unknowns, respectively. The main benefit of this method is the very cheap and easy setup phase that requires one user-defined parameter only taking value 1 or 2. This approach proves quite robust, achieving convergence in all the experimented test cases. The setup simplicity can be regarded, however, also as the main drawback of method 1 because the preconditioner cannot be finely tuned in ill-conditioned problems. This is because the operator density, which is a measure of the cost of preconditioner setup and application, grows very quickly when increasing the level of the expansion. Such a drawback can be, however, limited very easily by introducing a threshold parameter for post-filtering the smallest entries computed in the decoupling factors, as usually done with static approximate inverses, e.g., [19, 31, 39].
- Method 2a is in principle more flexible than method 1 because the dynamic computation of the non-zero pattern of the decoupling factors ensures a much better management of the memory occupation. The adaptive algorithm proposed in this work enables to select the most significant entries and to stop the computation whenever a prescribed quality of the decoupling factor is obtained.The procedure is computationally more demanding than for method 1, but its cost is actually part of the pre-processing stage that is easily amortized in a full-transient simulation. The main reason for this behavior relies on the fact that we are trying to find explicitly a sparse approximation of a matrix that is actually dense, with a similar size for most of the entries. In these cases, the adaptive algorithm fails to recognize the most significant entries simply because they do not exist and all the coefficients are almost equally important. As a consequence, the residuals $\mathbf{r}_1^{(k)}$ and $\mathbf{r}_2^{(k)}$ (Eq. 18) are not effectively reduced. In some occurrences, the computed coefficients have also a small size, thus adding the detrimental effect of a possible disruptive accumulation of round-off errors. In these situations, method 1 appears to be much more stable, even though not as efficient. Since the main reason for such a behavior is the theoretical absence

of a good sparse approximation for a dense decoupling block, it can be advisable to switch to an implicit approximation of $F_1$ and $F_2$, as done for instance in [10].

– Method 2b is a combination of the two previous methods (1 and 2a), enriching with few adaptive steps the static pattern provided by $\eta = 1$. In this case, the physically available information, i.e., the original node-to-element and face-to-element connection patterns, are used to guide the residual computation. On the other hand, even few steps of the dynamic approach can improve the quality of the decoupling factors, thus accelerating in any case the solver convergence. Generally speaking, this approach appears to be the most effective among the tested algorithms, with a limited setup cost and a reduced iteration count to convergence.

**Acknowledgments** Portions of this work are developed within the 2019 GNCS project "Innovative and parallel techniques for large size linear and non-linear systems, matrix functions and equations, with applications." Portions of this work were performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

## References

1. Adler, J., Gaspar, F., Hu, X., Ohm, P., Rodrigo, C., Zikatanov, L.: Robust block preconditioners for a new stabilized discretization of the poroelastic equations. arXiv:1905.10353 (2020)
2. Adler, J., Gaspar, F., Hu, X., Rodrigo, C., Zikatanov, L.: Robust block preconditioners for Biot's model. In: Domain decomposition methods in science and engineering, Lecture Notes in Computational Science and Engineering. Springer (2018). https://doi.org/10.1007/978-3-319-93873-8_1
3. Almani, T., Kumar, K., Dogru, A., Singh, G., Wheeler, M.: Convergence analysis of multirate fixed-stress split interative schemes for coupling flow with geomechanics. Comput. Methods Appl. Mech. Eng. **311**, 180–207 (2016). https://doi.org/10.1016/j.cma.2016.07.036
4. Bergamaschi, L., Ferronato, M., Gambolati, G.: Novel preconditioners for the iterative solution to FE-discretized coupled consolidation equations. Comput. Methods Appl. Mech. Eng. **196**, 2647–2656 (2007). https://doi.org/10.1016/j.cma.2007.01.13
5. Bergamaschi, L., Ferronato, M., Gambolati, G.: Mixed constraint preconditioners for the iterative solution to FE coupled consolidation equations. J. Comput. Phys. **227**, 9885–9897 (2008). https://doi.org/10.1016/j.jcp.2008.08.002
6. Biot, M.: General theory of three-dimensional consolidation. J. Appl. Phys. **12**, 155–164 (1941). https://doi.org/10.1063/1.1712886
7. Borregales, M., Radu, F.A., Kumar, K., Nordbotten, J.M.: Robust iterative schemes for non-linear poromechanics. Comput. Geosci. **22**(4), 1021–1038 (2018). https://doi.org/10.1007/s10596-018-9736-6
8. Both, J.W., Borregales, M., Nordbotten, J.M., Kumar, K., Radu, F.A.: Robust fixed stress splitting for Biot's equations in heterogeneous media. Appl. Math. Lett. **68**, 101–108 (2017). https://doi.org/10.1016/j.aml.2016.12.019
9. Castelletto, N., Gambolati, G., Teatini, P.: A coupled MFE poromechanical model of a large-scale load experiment at the coastland of Venice. Comput. Geosci. **19**, 17–29 (2015). https://doi.org/10.1007/s10596-014-9450-y
10. Castelletto, N., White, J., Ferronato, M.: Scalable algorithms for three-field mixed finite element coupled poromechanics. J. Comput. Phys. **327**, 894–918 (2016). https://doi.org/10.1016/j.jcp.2016.09.063
11. Castelletto, N., White, J.A., Tchelepi, H.A.: 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 (2015). https://doi.org/10.1002/nag.2400
12. Chapman, B., Jost, G., Van Der Pas, R.: Using openMP: portable shared memory parallel programming, vol. 10, MIT press (2008)
13. Christie, M., Blunt, M., et al.: Tenth SPE comparative solution project: a comparison of upscaling techniques. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (2001). https://doi.org/10.2118/66599-ms
14. Coussy, O.: Mechanics and physics of porous solids. J. Wiley & Sons ltd (2010)
15. Dana, S., Ganis, B., Wheeler, M.F.: A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. J. Comput. Phys. **352**, 1–22 (2018). https://doi.org/10.1016/j.jcp.2017.09.049
16. Ferronato, M., Castelletto, N., Gambolati, G.: A fully coupled 3-D mixed finite element model of Biot consolidation. J. Comput. Phys. **229**(12), 4813–4830 (2010). https://doi.org/10.1016/j.jcp.2010.03.018
17. Ferronato, M., Franceschini, A., Janna, C., Castelletto, N., Tchelepi, H.: A general preconditioning framework for coupled multiphysics problems with application to contact- and poro-mechanics. J. Comput. Phys. **398**(108887), 1–23 (2019). https://doi.org/10.1016/j.jcp.2019.108887
18. Ferronato, M., Janna, C., Gambolati, G.: Mixed constraint preconditioning in computational contact mechanics. Comput. Methods Appl. Mech. Eng. **197**, 3922–3931 (2008). https://doi.org/10.1016/j.cma.2008.03.008
19. Ferronato, M., Janna, C., Pini, G.: Parallel solution to ill-conditioned FE geomechanical problems. Int. J. Numer. Anal. Methods Geomech. **36**, 422–437 (2012). https://doi.org/10.1002/nag.1012
20. Ferronato, M., Janna, C., Pini, G.: A generalized block FSAI preconditioner for nonsymmetric linear systems. J. Comput. Appl. Math. **256**, 230–241 (2014). https://doi.org/10.1016/j.cam.2013.07.049
21. Frigo, M., Castelletto, N., Ferronato, M.: A relaxed physical factorization preconditioner for mixed finite element coupled poromechanics. SIAM J. Sci. Comp. **41**, B694–B720 (2019). https://doi.org/10.1137/18m120645x
22. Gaspar, F.J., Rodrigo, C.: On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics. Comput. Methods Appl. Mech. Eng. **326**, 526–540 (2017). https://doi.org/10.1016/j.cma.2017.08.025
23. Girault, V., Kumar, K., Wheeler, M.F.: Convergence of iterative coupling of geomechanics with flow in a fractured poroelastic medium. Comput. Geosci. **20**(5), 997–1011 (2016). https://doi.org/10.1007/s10596-016-9573-4
24. Grote, M., Huckle, T.: Parallel preconditioning with sparse approximate inverses. SIAM J. Sci. Comp. **18**, 838–853 (1997). https://doi.org/10.1137/S1064827594276552
25. Haga, J., Osnes, H., Langtangen, H.: A parallel block preconditioner for large-scale poroelasticity with highly heterogeneous material parameters. Comput. Geosci. **16**, 723–734 (2012). https://doi.org/10.1007/s10596-012-9284-4
26. Haga, J.B., Osnes, H., Langtangen, H.P.: On the causes of pressure oscillations in low-permeable and low-compressible

porous media. Int. J. Numer. Anal. Methods Geomech. **36**(12), 1507–1522 (2012). https://doi.org/10.1002/nag.1062
27. Hong, Q., Kraus, J.: Parameter-robust stability of classical three-field formulation of Biot's consolidation model. Electron. Trans. Numer. Anal. **48**, 202–226 (2018). https://doi.org/10.1553/etna_vol48s202
28. Huckle, T.: Approximate sparsity patterns for the inverse of a matrix and preconditioning. Appl. Numer. Math. **30**, 291–303 (1999). https://doi.org/10.1016/s0168-9274(98)00117-2
29. Huckle, T.: Factorized sparse approximate inverses for preconditioning. J. Supercomp. **25**, 109–117 (2003). https://doi.org/10.1023/A:1023988426844
30. Janna, C., Ferronato, M.: Adaptive pattern research for block FSAI preconditioning. SIAM J. Sci. Comp. **33**, 3357–3380 (2011). https://doi.org/10.1137/100810368
31. Janna, C., Ferronato, M., Gambolati, G.: A block FSAI-ILU parallel preconditioner for symmetric positive definite linear systems. SIAM J. Sci. Comp. **32**, 2468–2484 (2010). https://doi.org/10.1137/090779760
32. Janna, C., Ferronato, M., Gambolati, G.: Parallel inexact constraint preconditioning for ill-conditioned consolidation problems. Comput. Geosci. **16**, 661–675 (2012). https://doi.org/10.1007/s10596-012-9276-4
33. Janna, C., Ferronato, M., Sartoretto, F., Gambolati, G.: FSAIPACK: A software package for high-perfomance factored sparse approximate inverse preconditioning. ACM Trans. Math. Softw. **41**(Article 10), 1–26 (2015). https://doi.org/10.1145/2629475
34. Jha, B., Juanes, R.: A locally conservative finite element framework for the simulation of coupled flow and reservoir geomechanics. Acta Geotech. **2**(3), 139–153 (2007). https://doi.org/10.1007/s11440-007-0033-0
35. Jia, Z., Zhu, B.: A power sparse approximate inverse preconditioning procedure for large sparse linear systems. Numer. Linear Algebra Appl. **16**, 259–299 (2009). https://doi.org/10.1002/nla.614
36. Kim, J., Tchelepi, H.A., Juanes, R.: Stability, accuracy and efficiency of sequential methods for coupled flow and geomechanics. SPE J. **16**(2), 249–262 (2011). https://doi.org/10.2118/119084-PA
37. Kim, J., Tchelepi, H.A., Juanes, R.: Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. Comput. Methods Appl. Mech. Eng. **200**(13), 1591–1606 (2011). https://doi.org/10.1016/j.cma.2010.12.022
38. Kolotilina, L., Yeremin, A.: Factorized sparse approximate inverse preconditionings I. Theory. SIAM J. Matrix Anal. Appl. **14**, 45–58 (1993). https://doi.org/10.1137/0614004
39. Kolotilina, L., Yeremin, A.: Factorized sparse approximate inverse preconditioning. IV. Simple approaches to rising efficiency. Numer. Linear Algebra Appl. **6**, 515–531 (1999). https://doi.org/10.1002/(SICI)1099-1506(199910/11)6:7<515::AID-NLA176>3.0.CO;2-0
40. Lee, J.J., Mardal, K.A., Winther, R.: Parameter-robust discretization and preconditioning of biot's consolidation model. SIAM J. Sci. Comp. **39**(1), A1–A24 (2010). https://doi.org/10.1137/15M1029473
41. Lipnikov, K.: Numerical methods for the Biot model in poroelasticity. PhD thesis University of Houston (2002)
42. Luo, P., Rodrigo, C., Gaspar, F.J., Oosterlee, C.W.: On an Uzawa smoother in multigrid for poroelasticity equations. Numer. Linear Algebra Appl. **24**(1), e2074 (2017). https://doi.org/10.1002/nla.2074
43. Manteuffel, T.A., Munzenmaier, S., Ruge, J., Southworth, B.: Nonsymmetric reduction-based algebraic multigrid. SIAM J. Sci. Comp. **41**(5), S242–S268 (2019). https://doi.org/10.1137/18m1193761
44. Mikelič, A., Wheeler, M.: Convergence of iterative coupling for coupled flow and geomechanics. Comput. Geosci. **17**, 455–461 (2013). https://doi.org/10.1007/s.10596-012-9318-y
45. Phillips, P.J., Wheeler, M.F.: A coupling of mixed and continuous Galerkin finite element methods for poroelasticity i: the continuous in time case. Comput. Geosci. **11**(2), 131–144 (2007a). https://doi.org/10.1007/s10596-007-9045-y
46. Rodrigo, C., Hu, X., Ohm, P., Adler, J.H., Gaspar, F.J., Zikatanov, L.T.: New stabilized discretizations for poroelasticity and the Stokes' equations. Comput. Methods Appl. Mech. Eng. **341**, 467–484 (2018). https://doi.org/10.1016/j.cma.2018.07.003
47. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comp. **7**(3), 856–869 (1986). https://doi.org/10.1137/0907058
48. Terzaghi, K.: Erdbaumechanik auf Bodenphysikalischer Grundlage. F. Deuticke Wien (1925)
49. Vassilevski, P.S.: Sparse matrix element topology with application to AMG(e) and preconditioning. Numer. Linear Algebra Appl. **9**(6-7), 429–444 (2002). https://doi.org/10.1002/nla.300
50. White, J.A., Borja, R.: Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics. Comput. Geosci. **15**(4), 647–659 (2011). https://doi.org/10.1007/s10596-011-9233-7
51. White, J.A., Castelletto, N., Tchelepi, H.A.: Block-partitioned solvers for coupled poromechanics: a unified framework. Comput. Methods Appl. Mech. Eng. **303**, 55–74 (2016). https://doi.org/10.1016/j.cma.2016.01.008