



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Enhanced Relaxed Physical Factorization Preconditioner for Coupled Poromechanics

M. Frigo, N. Castelletto, M. Ferronato

December 3, 2019

Computers & Mathematics with Applications

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Enhanced Relaxed Physical Factorization preconditioner for coupled poromechanics

Matteo Frigo^a, Nicola Castelletto^b, Massimiliano Ferronato^a

^aDepartment ICEA, University of Padova, Padova, IT

^bAtmospheric, Earth and Energy Division, Lawrence Livermore National Laboratory, United States

Abstract

The relaxed physical factorization (RPF) preconditioner is a recent algorithm allowing for the efficient and robust solution to the block linear systems arising from the three-field displacement-velocity-pressure formulation of coupled poromechanics. For its application, however, it is necessary to invert blocks with the algebraic form $\hat{C} = (C + \beta FF^T)$, where C is a symmetric positive definite matrix, FF^T a rank-deficient term, and β a real non-negative coefficient. The inversion of \hat{C} , performed in an inexact way, can become unstable for large values of β , as it usually occurs at some stages of a full poromechanical simulation. In this work, we propose a family of algebraic techniques to stabilize the inexact solve with \hat{C} . This strategy can prove useful in other problems as well where such an issue might arise, such as augmented Lagrangian preconditioning techniques for Navier-Stokes or incompressible elasticity. First, we introduce an iterative scheme obtained by a natural splitting of matrix \hat{C} . Second, we develop a technique based on the use of a proper projection operator annihilating the near-kernel modes of \hat{C} . Both approaches give rise to a novel class of preconditioners denoted as Enhanced RPF (ERPF). Effectiveness and robustness of the proposed algorithms are demonstrated in both theoretical benchmarks and real-world large-size applications, outperforming the native RPF preconditioner.

Keywords: Preconditioning, Krylov subspace methods, Poromechanics

1. Introduction

Numerical simulation of Darcy's flow in a porous medium coupled with quasi-static mechanical deformation is based on poromechanics theory [1, 2]. We consider saturated single-phase flow of a slightly compressible fluid through a deformable medium. The set of governing equations for the displacement-velocity-pressure formulation of the poroelastic problem, e.g., [3–8] reads:

$$\nabla \cdot (\mathbb{C}_{dr} : \nabla^s \mathbf{u} - b p \mathbf{1}) = \mathbf{0} \quad \text{in } \Omega \times \mathcal{I} \quad (\text{equilibrium}), \quad (1a)$$

$$\mu \boldsymbol{\kappa}^{-1} \cdot \mathbf{q} + \nabla p = \mathbf{0} \quad \text{in } \Omega \times \mathcal{I} \quad (\text{Darcy's law}), \quad (1b)$$

$$b \nabla \cdot \dot{\mathbf{u}} + S_\epsilon \dot{p} + \nabla \cdot \mathbf{q} = f \quad \text{in } \Omega \times \mathcal{I} \quad (\text{continuity}), \quad (1c)$$

with $\Omega \subset \mathbb{R}^3$ a bounded domain and $\mathcal{I} = (0, t_{max})$ an open time interval. In (1), \mathbb{C}_{dr} is the rank-four elasticity tensor, ∇^s is the symmetric gradient operator, b is the Biot coefficient, and $\mathbf{1}$ is the rank-two identity tensor; μ and $\boldsymbol{\kappa}$ are the fluid viscosity and the rank-two permeability tensor, respectively; S_ϵ is the constrained specific storage coefficient, i.e. the reciprocal of Biot's modulus, and f the fluid source term. The primary unknowns are denoted by \mathbf{u} (displacement),

Email addresses: matteo.frigo.3@phd.unipd.it (Matteo Frigo), castelletto1@llnl.gov (Nicola Castelletto), massimiliano.ferronato@unipd.it (Massimiliano Ferronato)

10 \mathbf{q} (Darcy's velocity), and p (excess pore pressure). The problem is completed with proper boundary conditions:

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u \times \mathcal{I}, \quad (\text{prescribed displacement}), \quad (2a)$$

$$(\mathbb{C}_{dr} : \nabla^s \mathbf{u} - b p \mathbf{1}) \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_\sigma \times \mathcal{I}, \quad (\text{prescribed traction}), \quad (2b)$$

$$\mathbf{q} \cdot \mathbf{n} = \bar{q} \quad \text{on } \Gamma_q \times \mathcal{I}, \quad (\text{prescribed discharge}), \quad (2c)$$

$$p = \bar{p} \quad \text{on } \Gamma_p \times \mathcal{I}, \quad (\text{prescribed pressure}), \quad (2d)$$

where $\bar{\mathbf{u}}$, $\bar{\mathbf{t}}$, \bar{q} , and \bar{p} are the boundary displacement, traction, Darcy velocity and excess pore pressure, respectively, on the portions Γ_u , Γ_σ , Γ_q , and Γ_p of the frontier $\partial\Omega$ such that $\Gamma_u \cup \Gamma_\sigma = \Gamma_q \cup \Gamma_p = \partial\Omega$ and $\Gamma_u \cap \Gamma_\sigma = \Gamma_q \cap \Gamma_p = \emptyset$. The initial condition can be given as

$$b \nabla \cdot \mathbf{u}(\mathbf{x}, 0) + S_\epsilon p(\mathbf{x}, 0) = b \nabla \cdot \mathbf{u}_0 + S_\epsilon p_0, \quad \mathbf{x} \in \bar{\Omega}, \quad (3)$$

15 with \mathbf{u}_0 and p_0 the initial displacement and excess pore pressure field. While equation (3) specifies the initial fluid content of the medium [1], in practice p_0 is often directly measured or computed, and \mathbf{u}_0 is then obtained so as to satisfy the momentum balance (1a)—see, e.g. [9, 10]. The focus of this work is the iterative solution of the linear algebraic system arising from the discretization of the initial boundary value problem (1)-(3). In particular, we consider the block linear system $\mathbf{A}\mathbf{x} = \mathbf{F}$ obtained by combining a mixed finite element discretization in space with implicit integration in time using the θ -method [11]:

$$\mathbf{A} = \begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ Q^T & \gamma B^T & P \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{u} \\ \mathbf{q} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_q \\ \mathbf{f}_p \end{bmatrix}, \quad (4)$$

20 where $\mathbf{u} \in \mathbb{R}^{n_u}$, $\mathbf{q} \in \mathbb{R}^{n_q}$ and $\mathbf{p} \in \mathbb{R}^{n_p}$ denote the vectors containing the unknown n_u displacement, n_q Darcy velocity, and n_p pressure degrees of freedom, respectively, at discrete time t_{n+1} , and $\gamma = \theta \Delta t$, with $\Delta t = (t_{n+1} - t_n)$ the time integration step size and θ a real parameter ($1/2 \leq \theta \leq 1$). In (4), K is the classic small-strain structural stiffness matrix, A is the (scaled) velocity mass matrix, P is the (scaled) pressure mass matrix, Q is the poromechanical coupling block and B is the Gram matrix. We employ \mathbb{Q}_1 elements for the displacement field, \mathbb{RT}_0 elements for the velocity field, and \mathbb{P}_0 elements for the pressure field. Assuming a linear elastic law for the mechanical behavior of the porous medium leads to a symmetric positive definite (SPD) matrix K . Matrix A is SPD as well, while P is diagonal with non-negative entries. Additional details and explicit expressions for matrices in \mathbf{A} and vectors in the block right-hand side \mathbf{F} can be found in [11, 12].

30 The two-field displacement-pressure formulation discretized by the continuous Galerkin (CG) finite elements is probably the most common technique for numerical modeling of coupled poromechanics [13]. In contrast to the CG method, the selected three-field formulation yields locally mass-conservative velocity fields and is robust for highly heterogeneous permeability fields, a key requirement in several geoscience applications, e.g., [14, 15]. We note that an enriched Galerkin (EG) method can be also used to fix this drawback of the CG method [16, 17]. If the selected combination of discretization spaces does not intrinsically satisfy the inf-sup stability in the undrained limit [18], proper stabilization strategies can be introduced to eliminate the spurious oscillation modes arising in the pressure solution. These techniques can be subdivided into two classes, based on whether they: (i) enrich the discrete variable spaces so that the Ladyzhenskaya-Babuška-Brezzi (LBB) condition is met, e.g. [19, 20], or (ii) introduce a stabilization term to guarantee the solvability of the resulting saddle-point problem, e.g. [21, 22]. The first strategy is mathematically elegant and robust, but can modify the algebraic structure of problem (4). The selected \mathbb{Q}_1 - \mathbb{RT}_0 - \mathbb{P}_0 mixed discretization can be effectively stabilized following the second strategy by adding a proper term to the mass balance equation [23]. This approach has the advantage of having a minor impact on the algebraic structure of the problem (4), so that the solution algorithms developed in this work can be used independently of the presence of such a contribution.

45 Our goal is the efficient iterative solution of the linear system (4) with the aid of preconditioned Krylov subspace methods. Since \mathbf{A} is non-symmetric, or indefinite if properly symmetrized, a global Bi-CGStab [24] or GMRES [25] algorithm can be used as a solver. Over the past decade, a growing interest has regarded the implicit solution of coupled poromechanics, with the introduction of a number of methods built on both fully-algebraic approaches and

discretization-based strategies. The proposed algorithms can be grouped into two main categories: (i) *sequential-implicit* methods, in which the primary variables are updated one at a time, iterating between governing equations [9, 26–32]; and (ii) *fully-coupled* approaches, which solve the global system of equations simultaneously for all the primary unknowns, with an emphasis on preconditioned Krylov solvers with scalable and parameter-robust methods [11, 12, 18, 33–46]. Sequential-implicit methods exhibit a linear convergence, but can take advantage of using distinct, and specialized, codes for the structural and the flow models. In contrast, fully-coupled approaches ensure unconditional stability with a super-linear convergence, but require the design of dedicated preconditioning operators. This research field, in particular, has attracted an increasing interest from the scientific community, with the introduction of a number of different algorithms in the last few years. Just to cite a few recent examples of preconditioners for coupled poromechanics, we mention spectrally-equivalent block diagonal and block triangular strategies [18, 33, 38, 41, 43, 45], multigrid-based scalable methods [39, 40, 46], or block preconditioners combining physically-based and algebraic tricks to construct different Schur complement approximations [11, 12, 23, 44]. A recent novel approach belonging to the category of fully-coupled methods is the relaxed physical factorization (RPF) preconditioner [47]. The distinctive feature of the RPF algorithm is that it does not rely on accurate sparse approximations of the Schur complements, with the convergence accelerated by setting a nearly-optimal relaxation parameter α . Similar approaches were originally developed for the solution of the linear system arising from the discretization of the Navier-Stokes equations [48, 49].

The RPF technique has been shown very efficient in solving challenging problems with severe material heterogeneities and proved competitive with other well-established block preconditioners [47]. However, a performance degradation can be observed when the time-step approaches 0 (undrained conditions) or $+\infty$ (fully drained conditions). This behavior is due to the fact that the native RPF preconditioner requires the inexact solution of inner systems in the form $\hat{C} = (C + \beta FF^T)$, where C is SPD and FF^T is rank-deficient, and β is a scalar coefficient that may tend to infinity in the limit case of $\frac{\Delta t}{t_c} \ll 1$ (undrained conditions) or $\frac{\Delta t}{t_c} \gg 1$ (uncoupled consolidation), with t_c the characteristic consolidation time [2]. When β grows beyond a certain threshold, the inexact inversion of \hat{C} carried out by well-known approaches, such as incomplete factorizations or algebraic multigrid, may become unstable, thus affecting the overall RPF efficiency and robustness. Problematic issues with a very similar algebraic origin may frequently occur in several other important applications beyond coupled poromechanics and independently of the selected discretization spaces, e.g., in augmented Lagrangian approaches for the Navier-Stokes equations or mixed formulations of incompressible elasticity [50–54]. In this paper, we advance two alternative methods to stabilize the inexact solves with the inner blocks \hat{C} . In the first approach, a natural splitting of \hat{C} is proposed to obtain a convergent sequential iterative scheme, which is stopped at a low precision. In the second approach, a proper projection operator is introduced to approximately annihilate the near-kernel modes of \hat{C} . We would like to emphasize that the proposed two strategies are fully algebraic, hence they are naturally amenable to a generalization to other applications where such an issue might arise.

The paper is organized as follows. In Section 2 a brief review of the RPF preconditioner is provided, with a focus on the numerical challenges associated with large and small time-step values. Section 3 provides the theoretical basis for the two methods introduced for the RPF stabilization, with the related properties, performance and robustness investigated in Section 4 through both academic benchmarks and real-world applications. Finally, a few conclusive remarks close the paper in Section 5.

2. The RPF Preconditioner

In this section, we briefly revisit the RPF preconditioner [47] recalling only the aspects necessary for the remainder of this work. In factored form, the RPF preconditioner \mathbf{M}^{-1} reads:

$$\mathbf{M}^{-1} = \alpha (\mathbf{M}_1 \mathbf{M}_2)^{-1} = \alpha \left(\begin{bmatrix} K & 0 & -Q \\ 0 & \alpha I_q & 0 \\ Q^T & 0 & \alpha I_p \end{bmatrix} \begin{bmatrix} \alpha I_u & 0 & 0 \\ 0 & A & -B \\ 0 & \gamma B^T & \alpha I_p \end{bmatrix} \right)^{-1}, \quad (5)$$

where $\alpha > \|P\|_\infty$ is the relaxation parameter, and I_u , I_q , and I_p are the identity matrix in $\mathbb{R}^{n_u \times n_u}$, $\mathbb{R}^{n_q \times n_q}$, and $\mathbb{R}^{n_p \times n_p}$, respectively. Setting α is key to accelerate the convergence of the Krylov subspace method preconditioned with (5) to solve (4). Though the convergence rate of non-symmetric Krylov solvers does not depend only on the eigenvalue

distribution of the preconditioned matrix, the computational practice suggests that a clustered eigenspectrum rarely leads to slow convergence. Hence, the criterion for setting α relies on clustering as much as possible the eigenvalues of the preconditioned matrix $\mathbf{T} = \mathbf{M}^{-1}\mathbf{A}$ around 1. To this aim, α can be selected such that the trace of \mathbf{T} is as close as possible to the system size:

$$\alpha = \arg \min_{\alpha \geq \|P\|_{\infty}} (n_u + n_q + n_p - \text{tr}[\mathbf{T}]). \quad (6)$$

Following the arguments developed in [47], equation (6) can be rewritten as:

$$\alpha = \arg \min_{\alpha \geq \|P\|_{\infty}} (n_p - \text{tr}[Z_{\alpha}]), \quad (7)$$

where

$$Z_{\alpha} = \alpha (\alpha I_p + S_K)^{-1} S (\alpha I_p + \gamma S_A)^{-1}, \quad (8)$$

and

$$S_K = Q^T K^{-1} Q, \quad (9)$$

$$S_A = B^T A^{-1} B, \quad (10)$$

$$S = P + S_K + \gamma S_A. \quad (11)$$

Finding analytically α as in (7) is not practically feasible, because the matrices (9), (10), (11) are dense. However, for the sake of α estimate only, we can replace S_K and S_A with diagonal matrices, D_K and D_A , respectively. As far as S_K is concerned, it can be effectively approximated by the so-called ‘‘fixed-stress’’ matrix [55, 56], also computed in a fully algebraic way [12]. For the matrix S_A , we follow the approach used in [57] by defining:

$$D_A = \text{diag}(B^T \tilde{A}^{-1} B), \quad (12)$$

where

$$\tilde{A} = \text{diag}(a_1, a_2, \dots, a_{n_q}), \quad a_i = \left(\sum_{j=1}^{n_q} |A_{ij}|^2 \right)^{1/2}, \quad i \in \{1, \dots, n_q\}. \quad (13)$$

With the diagonal approximations D_K and D_A instead of S_K and S_A , the trace of Z_{α} can be computed at a negligible cost. The value of α is therefore obtained from (7) as [47]:

$$\alpha = \frac{\sqrt{\gamma}}{n_p} \sum_{i=1}^{n_p} \sqrt{D_K^{(i)} D_A^{(i)}}. \quad (14)$$

Remark 2.1. Throughout this paper, we overload the operator $\text{diag}(\cdot)$ to create a square diagonal matrix. In particular, if $\mathbf{v} \in \mathbb{R}^n$, then $\text{diag}(\mathbf{v})$ returns a diagonal matrix with the elements of vector \mathbf{v} (Eq. (13)); otherwise, if $V \in \mathbb{R}^{n \times n}$, then $\text{diag}(V)$ returns a diagonal matrix consisting of the main diagonal of V (Eq. (12)).

Remark 2.2. We emphasize that, in many approaches, the design of robust and efficient preconditioners based on accurate approximations of matrices (9), (10), and (11)—i.e., the Schur complement approximation—is the focus. Conversely, in the RPF framework cheap diagonal approximations of such matrices are enough to enable effective estimates for the parameter α , the key component to the success of this preconditioning technique.

To apply \mathbf{M}^{-1} to a vector, it is convenient to rewrite (5) as:

$$\mathbf{M}^{-1} = \left(\begin{bmatrix} I_u & 0 & -\frac{1}{\alpha} Q \\ 0 & I_q & 0 \\ 0 & 0 & I_p \end{bmatrix} \begin{bmatrix} \hat{K} & 0 & 0 \\ 0 & I_q & 0 \\ Q^T & 0 & I_p \end{bmatrix} \begin{bmatrix} I_u & 0 & 0 \\ 0 & \hat{A} & -B \\ 0 & 0 & \alpha I_p \end{bmatrix} \begin{bmatrix} I_u & 0 & 0 \\ 0 & I_q & 0 \\ 0 & \frac{\gamma}{\alpha} B^T & I_p \end{bmatrix} \right)^{-1}, \quad (15)$$

115 with

$$\hat{K} = K + \frac{1}{\alpha} QQ^T, \quad (16)$$

$$\hat{A} = A + \frac{\gamma}{\alpha} BB^T. \quad (17)$$

Equation (15) shows that the RPF application requires two inner solves with matrices \hat{K} and \hat{A} of (16)-(17), which are performed inexactly for the sake of computational efficiency. Notice that such matrices depend on α , whose inverse multiplies the rank-deficient matrices QQ^T and BB^T . The latter could prevail on the SPD contributions K and A for relatively small values of α , thus affecting the accuracy and numerical stability in the inexact application of \hat{K}^{-1} and \hat{A}^{-1} . For this reason, two limiting values, α_K and α_A , are introduced, with the definitions (16)-(17) modified as:

$$\hat{K} = K + \frac{1}{\max(\alpha, \alpha_K)} QQ^T, \quad (18)$$

$$\hat{A} = A + \frac{\gamma}{\max(\alpha, \alpha_A)} BB^T. \quad (19)$$

Following the result of Theorem 3.1 in [47], the bounds α_K and α_A read:

$$\alpha_K = \frac{\lambda_1(S_K)}{\bar{\omega}_K - 1}, \quad \alpha_A = \frac{\gamma \lambda_1(S_A)}{\bar{\omega}_A - 1}, \quad (20)$$

where $\lambda_1(S_K)$ and $\lambda_1(S_A)$ are the spectral radii of S_K and S_A , and $\bar{\omega}_K$ and $\bar{\omega}_A$ are user-specified parameters denoting the maximum acceptable value for the ratios $\kappa(\hat{K})/\kappa(K)$ and $\kappa(\hat{A})/\kappa(A)$, i.e., the maximum increase of the spectral condition number of \hat{K} and \hat{A} with respect to that of K and A , respectively. For the practical computation of α_K and α_A , in equation (20) S_K and S_A can be replaced by D_K and D_A , respectively.

The RPF set-up and application to the vector $\mathbf{r}^T = [\mathbf{r}_u^T, \mathbf{r}_q^T, \mathbf{r}_p^T]$ are summarized in Algorithm 1 and 2, respectively. Of course, lines 3 and 8 of Algorithm 2 are usually replaced by an approximate application of \hat{K}^{-1} and \hat{A}^{-1} .

Algorithm 1 RPF computation: $\mathbf{M} = \text{RPF_SETUP}(\gamma, \bar{\omega}_K, \bar{\omega}_A, D_K, D_A, \mathbf{A})$

- 1: $\lambda_1(D_K) = \max_i D_K^{(i)}$
 - 2: $\lambda_1(D_A) = \max_i D_A^{(i)}$
 - 3: $\alpha_K = (\bar{\omega}_K - 1)^{-1} \lambda_1(D_K)$
 - 4: $\alpha_A = (\bar{\omega}_A - 1)^{-1} \gamma \lambda_1(D_A)$
 - 5: $\alpha = \sqrt{\gamma} n_p^{-1} \sum_i \sqrt{D_K^{(i)} D_A^{(i)}}$
 - 6: $\hat{K} = QQ^T$
 - 7: $\hat{A} = BB^T$
 - 8: $\hat{K} \leftarrow K + \max(\alpha, \alpha_K)^{-1} \hat{K}$
 - 9: $\hat{A} \leftarrow A + \gamma \max(\alpha, \alpha_A)^{-1} \hat{A}$
-

3. Enhanced RPF preconditioner

The use of equations (18)-(19) instead of (16)-(17) often is not sufficient to guarantee the RPF efficiency. Especially for large time-step values, a degradation of the RPF performance can be observed. There are several reasons for this behavior. First, the value of α_K and α_A can be quite far from the optimal value of α , in particular for $\frac{\gamma}{t_c} \gg 1$. This can be heuristically noted by considering that α is proportional to $\sqrt{\gamma}$ (equation (14)), while α_K is constant and α_A varies linearly with γ (equation (20)). The separation between α_K and α is finite for $\gamma \rightarrow 0$, whereas for $\gamma \rightarrow \infty$ the separation between α_A and α grows to infinity. Hence, the larger the value of γ , the greater is the distance between the optimal α and α_A .

Algorithm 2 RPF application: $[\mathbf{t}_u^T, \mathbf{t}_q^T, \mathbf{t}_p^T] = \text{RPF_APPLY}(\mathbf{r}_u, \mathbf{r}_q, \mathbf{r}_p, \mathbf{A}, \mathbf{M})$

- 1: $\mathbf{x}_u = \alpha^{-1} \mathbf{Q} \mathbf{r}_p$
 - 2: $\mathbf{x}_u \leftarrow \mathbf{r}_u + \mathbf{x}_u$
 - 3: **Solve** $\hat{\mathbf{K}} \mathbf{t}_u = \mathbf{x}_u$
 - 4: $\mathbf{y}_p = \mathbf{Q}^T \mathbf{t}_u$
 - 5: $\mathbf{y}_p \leftarrow \mathbf{r}_p - \mathbf{y}_p$
 - 6: $\mathbf{z}_q = \mathbf{B} \mathbf{y}_p$
 - 7: $\mathbf{z}_q \leftarrow \mathbf{r}_q + \alpha^{-1} \mathbf{z}_q$
 - 8: **Solve** $\hat{\mathbf{A}} \mathbf{t}_q = \mathbf{z}_q$
 - 9: $\mathbf{t}_p = \mathbf{B}^T \mathbf{t}_q$
 - 10: $\mathbf{t}_p \leftarrow \alpha^{-1} (\mathbf{y}_p - \gamma \mathbf{t}_p)$
-

The impact of setting a bound to α on the overall quality of the RPF preconditioner can be investigated as follows. For the sake of generality, we use the notation:

$$\hat{\mathbf{C}} \mathbf{w} = \mathbf{b}, \quad (21)$$

with:

$$\hat{\mathbf{C}} = \mathbf{C} + \beta \mathbf{F} \mathbf{F}^T, \quad (22)$$

to denote the inner solves required by the RPF application (lines 3 and 8 of Algorithm 2). The following result holds.

Theorem 3.1. *Let $\mathbf{C} \in \mathbb{R}^{n \times n}$ and $\mathbf{F} \in \mathbb{R}^{n \times n_p}$, with $n_p < n$, be an SPD and a full-rank matrix, respectively. If $\beta > \beta_\ell > 0$, then the n eigenvalues λ_i of the generalized eigenproblem*

$$(\mathbf{C} + \beta \mathbf{F} \mathbf{F}^T) \mathbf{v}_i = (\mathbf{C} + \beta_\ell \mathbf{F} \mathbf{F}^T) \lambda_i \mathbf{v}_i \quad (23)$$

are bounded in the interval $[1, \lambda_1]$, where:

$$\lambda_1 = \frac{\beta \mu_1(S_C) + 1}{\beta_\ell \mu_1(S_C) + 1}, \quad (24)$$

$S_C = \mathbf{F}^T \mathbf{C}^{-1} \mathbf{F}$, and $\mu_1(S_C)$ is the spectral radius of S_C .

Proof. The generalized eigenvalue problem (23) can be rewritten as:

$$\left[\mathbf{C}(1 - \lambda_i) + (\beta - \beta_\ell \lambda_i) \mathbf{F} \mathbf{F}^T \right] \mathbf{v}_i = \mathbf{0}. \quad (25)$$

Setting $\mathbf{H} = \mathbf{C}^{-1} \mathbf{F} \mathbf{F}^T$ and $\mu_i = (\lambda_i - 1) / (\beta - \beta_\ell \lambda_i)$, equation (25) reads:

$$\mathbf{H} \mathbf{v}_i = \mu_i \mathbf{v}_i. \quad (26)$$

The eigenvalues μ_i of \mathbf{H} are either 0, with multiplicity $n - n_p$, or equal to those of S_C . Furthermore, the eigenvalues λ_i are related to μ_i as follows:

$$\lambda_i = \frac{\beta \mu_i + 1}{\beta_\ell \mu_i + 1}. \quad (27)$$

Since $\mu_i \in [0, \mu_1(S_C)]$ and λ_i in (27) increases monotonically with μ_i for $\beta > \beta_\ell$, the eigenvalues λ_i belong to the interval \mathcal{I} :

$$\mathcal{I} = \left\{ \lambda_i \in \mathbb{R} \mid 1 \leq \lambda_i \leq \frac{\beta \mu_1(S_C) + 1}{\beta_\ell \mu_1(S_C) + 1} \right\}. \quad (28)$$

□

Let us denote with $\hat{\mathbf{C}}_\ell$ the matrix

$$\hat{\mathbf{C}}_\ell = \mathbf{C} + \beta_\ell \mathbf{F} \mathbf{F}^T, \quad (29)$$

obtained with the lower-bound value β_ℓ for β , and use \hat{C}_ℓ^{-1} as a preconditioner for the inner solve with \hat{C} in the RPF application (Algorithm 2). Theorem 3.1 shows that the spectral conditioning number of $\hat{C}_\ell^{-1}\hat{C}$ grows indefinitely with β , making the inner solve with \hat{C} more difficult and expensive as $\beta/\beta_\ell \rightarrow \infty$. If the solution to the inner system (21) is obtained inexactly, as usually necessary for the sake of computational efficiency in large-size applications, the accuracy of \mathbf{w} is expected to decrease when the separation between β and the limiting value β_ℓ grows. This may lead to an overall degradation of the RPF performance that might even prevent the outer Krylov solver from converging.

In order to stabilize the global RPF behavior, it is therefore important to restore equations (16) and (17) in the preconditioner construction and apply inexactly \hat{C}^{-1} through alternative methods that are less affected by its large condition number. To this aim, we introduce here two different approaches:

- Method 1: exploits the structure of \hat{C} by defining a natural splitting that can be used to develop an effective preconditioner for the inner problem;
- Method 2: uses an appropriate projection operator in order to get rid of the near-null space of \hat{C} .

The use of Method 1 and Method 2 gives rise to the Enhanced RPF preconditioners (ERPF1 and ERPF2, respectively). We notice here that both Method 1 and Method 2 are fully algebraic and are developed for a general SPD matrix C and rank-deficient term FF^T . Hence, their generalization to applications beyond the one focussed in this work is reasonably straightforward.

3.1. Method 1

A weighted splitting of matrix \hat{C} is adopted to obtain a stationary iterative scheme. Let us write \hat{C} as follows:

$$\hat{C} = \left(1 - \frac{\beta}{\beta_\ell}\right)C + \frac{\beta}{\beta_\ell}(C + \beta_\ell FF^T) = \left(1 - \frac{\beta}{\beta_\ell}\right)C + \frac{\beta}{\beta_\ell}\hat{C}_\ell, \quad (30)$$

and consider the following stationary iteration to solve (21):

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\beta_\ell}{\beta}\hat{C}_\ell^{-1}\mathbf{r}_k, \quad (31)$$

with $\mathbf{r}_k = (\mathbf{b} - \hat{C}\mathbf{w}_k)$ the residual at iteration k . The iteration matrix G associated with scheme (31) reads:

$$G = I - \frac{\beta_\ell}{\beta}\hat{C}_\ell^{-1}\hat{C} = \left(1 - \frac{\beta_\ell}{\beta}\right)\hat{C}_\ell^{-1}C. \quad (32)$$

Lemma 3.2. *The stationary scheme (31) for the solution of the system $\hat{C}\mathbf{w} = \mathbf{b}$ of equation (21), with $\beta > \beta_\ell > 0$, is unconditionally convergent with rate $R = \log(1 - \beta_\ell/\beta)$.*

Proof. The spectral radius $\lambda_1(G)$ of the iteration matrix is given by:

$$\lambda_1(G) = \left(1 - \frac{\beta_\ell}{\beta}\right)\lambda_1(N), \quad (33)$$

with $N = \hat{C}_\ell^{-1}C$ and $\lambda_1(N)$ its spectral radius. Since N reads

$$N = (C + \beta_\ell FF^T)^{-1}C = [C(I + \beta_\ell C^{-1}FF^T)]^{-1}C = (I + \beta_\ell H)^{-1}, \quad (34)$$

its spectral radius is:

$$\lambda_1(N) = \frac{1}{1 + \beta_\ell \lambda_n(H)}, \quad (35)$$

where $\lambda_n(H)$ denotes the smallest eigenvalue of $H = C^{-1}FF^T$. Being H similar to the symmetric positive semidefinite matrix $C^{-\frac{1}{2}}FF^T C^{-\frac{1}{2}}$, $\lambda_n(H) = 0$ and

$$\lambda_1(G) = 1 - \frac{\beta_\ell}{\beta} < 1. \quad (36)$$

Hence, the stationary scheme (31) is unconditionally convergent with rate $R = \log(1 - \beta_\ell/\beta)$. \square

180 The idea of Method 1 is to solve the system of equation (21) with the scheme (31). Since in the RPF preconditioner application such system is solved inexactly, we carry out a pre-defined number of stationary iterations, n_{in} , and replace the exact application of \hat{C}_ℓ^{-1} with an inexact one, say the inner preconditioner $M_{\hat{C}_\ell}^{-1}$. This procedure is summarized in Algorithm 3.

Algorithm 3 Method 1: $\mathbf{w} = \text{MET1_APPLY}(n_{in}, \beta, \beta_\ell, \mathbf{b}, \hat{C}, M_{\hat{C}_\ell}^{-1})$

```

1:  $\mathbf{w} = \mathbf{0}$ 
2: for  $k = 1, \dots, n_{in}$  do
3:    $\mathbf{r} = \mathbf{b} - \hat{C}\mathbf{w}$ 
4:   Apply  $M_{\hat{C}_\ell}^{-1}$  to  $\mathbf{r}$  to get  $\mathbf{v}$ 
5:    $\mathbf{w} \leftarrow \mathbf{w} + (\beta_\ell/\beta)\mathbf{v}$ 
6: end for

```

185 The ERPF1 preconditioner is obtained by replacing lines 3 and 8 of Algorithm 2 with the function in Algorithm 3 whenever $\alpha < \alpha_K$ and $\alpha < \alpha_A$, respectively. For each inner iteration k , Method 1 involves one matrix-vector product, one application of $M_{\hat{C}_\ell}^{-1}$ and two vector updates, hence the algorithm cost can grow up quickly with n_{in} . More precisely, denoting by n the number of rows of \hat{C} , the complexity χ_1 of Algorithm 3 reads:

$$\chi_1 = n_{in} \times [\chi(M_{\hat{C}_\ell}^{-1}) + \mathcal{O}(n)], \quad (37)$$

where $\chi(M_{\hat{C}_\ell}^{-1})$ is the complexity of the inner preconditioner application. For this reason, n_{in} should be kept as low as possible, say 2 or 3.

190 *Remark 3.1.* Equation (36) shows that the convergence rate of the iteration (31) degrades to 0 as $\beta \rightarrow \infty$. In our application, β is either $1/\alpha$ (equation (16)) or γ/α (equation (17)). In a full poromechanical simulation, the condition $\gamma/\alpha \rightarrow \infty$ is more likely to occur and usually happens at the final stage of the process, when the solution approaches steady-state conditions and Δt , hence γ , progressively increases. In this situation, Method 1 is expected to lose efficiency because it may require too many inner iterations to obtain a sufficiently accurate result.

195 *Remark 3.2.* The stationary scheme (31) is a Richardson iteration preconditioned by $(\beta_\ell/\beta)\hat{C}_\ell^{-1}$. Since \hat{C} is SPD, it could be also effectively replaced by a Conjugate Gradient scheme with the same preconditioner. Stopping the inner iterations after $n_{in} = 2$ or 3 steps, however, usually does not provide a significantly different outcome with respect to the use of Algorithm 3.

3.2. Method 2

200 Another strategy to solve (21) relies on using an appropriate projection operator. In particular, the idea is to project the system (21) onto a space \mathcal{Z} such that:

$$\mathbb{R}^n = \ker(F^T) \oplus \mathcal{Z}, \quad (38)$$

so as to annihilate the components of \mathbf{w} lying in the kernel of F^T . Let us define the \hat{C} -orthogonal projector P as:

$$P = Z[Z^T(\hat{C}Z)]^{-1}Z^T\hat{C}, \quad (39)$$

where $Z = [\mathbf{z}_1, \dots, \mathbf{z}_{n_p}] \in \mathbb{R}^{n \times n_p}$ is the generator of \mathcal{Z} :

$$\mathcal{Z} = \text{span}\{\mathbf{z}_1, \dots, \mathbf{z}_{n_p}\}. \quad (40)$$

The operator P is the linear mapping that projects a vector \mathbf{w} onto \mathcal{Z} orthogonally to \mathcal{L} :

$$\mathcal{L} = \{\mathbf{y}_i \in \mathbb{R}^n \mid \mathbf{y}_i = \hat{C}\mathbf{z}_i, \mathbf{z}_i \in \mathcal{Z}\}. \quad (41)$$

205 Notice that the projector P is \hat{C} -orthogonal according to the following definition [58]:

Definition 3.3. A projector P onto a subspace \mathcal{Z} is \hat{C} -orthogonal if and only if it is self-adjoint with respect to the \hat{C} -inner product.

Of course, we have $\text{Ran}(P) = \text{Ran}(Z)$. The solution \mathbf{w} of (21) can be decomposed as the sum of two contributions:

$$\mathbf{w} = P\mathbf{w} + (I - P)\mathbf{w} = \mathbf{w}_Z + \mathbf{w}_S, \quad (42)$$

210 where $\mathbf{w}_Z \in \mathcal{Z}$ and $\mathbf{w}_S \in \mathcal{S} = \text{Ran}(I - P)$. Applying the projector (39), the component \mathbf{w}_Z is given by:

$$\mathbf{w}_Z = P\mathbf{w} = ZE^{-1}Z^T\hat{C}\mathbf{w} = ZE^{-1}Z^T\mathbf{b}, \quad (43)$$

where $E = Z^T\hat{C}Z$ is the projection of \hat{C} onto \mathcal{Z} . In order to obtain \mathbf{w}_S , let us consider the following projected system:

$$(I - P^T)\hat{C}\mathbf{w} = (I - P^T)\mathbf{b}. \quad (44)$$

Recalling that $(I - P^T)\hat{C} = \hat{C}(I - P)$, equation (44) can be rewritten as:

$$\hat{C}\mathbf{w}_S = (I - P^T)\mathbf{b}. \quad (45)$$

Since $\mathbf{w}_S \perp \mathcal{L}$ of equation (41), the following condition holds:

$$Z^T(C + \beta FF^T)\mathbf{w}_S = 0. \quad (46)$$

215 If we set $Z = C^{-1}F$, equation (46) becomes:

$$(I + \beta S_C)F^T\mathbf{w}_S = 0, \quad (47)$$

which implies $\mathbf{w}_S \in \ker(F^T)$ because of the regularity of the matrix $(I + \beta S_C)$, with $S_C = F^T C^{-1}F$ (see proof of Theorem 3.1). Hence, by setting $\mathcal{Z} = \text{Ran}(C^{-1}F)$ equation (45) simplifies to:

$$C\mathbf{w}_S = (I - P^T)\mathbf{b}, \quad (48)$$

which no longer requires the inversion of \hat{C} . Observe that the contribution $P^T\mathbf{b}$ at the right-hand side of (48) reads:

$$\hat{C}ZE^{-1}Z^T\mathbf{b} = \hat{C}\mathbf{w}_Z, \quad (49)$$

220 i.e., $P^T\mathbf{b} = \hat{C}\mathbf{w}_Z = \mathbf{b}_Z$ is the projected right-hand side onto \mathcal{Z} . The solution \mathbf{w} to the system (21) is therefore given by the following set of equations:

$$\begin{aligned} \mathbf{w}_Z &= ZE^{-1}Z^T\mathbf{b}, \\ \mathbf{w}_S &= C^{-1}(\mathbf{b} - \mathbf{b}_Z), \\ \mathbf{w} &= \mathbf{w}_Z + \mathbf{w}_S. \end{aligned} \quad (50)$$

The matrix $ZE^{-1}Z^T$ reads:

$$\begin{aligned} ZE^{-1}Z^T &= C^{-1}F \left[F^T C^{-1} C C^{-1} F + \beta (F^T C^{-1} F F^T C^{-1} F) \right]^{-1} F^T C^{-1} \\ &= C^{-1}F \left[S_C + \beta S_C^2 \right]^{-1} F^T C^{-1} \\ &= C^{-1}F S_C^{-1} [I + \beta S_C]^{-1} F^T C^{-1}. \end{aligned} \quad (51)$$

Introducing (51) in the first equation of (50), we have:

$$\mathbf{w}_Z = C^{-1}F S_C^{-1} \tilde{S}_C^{-1} F^T C^{-1} \mathbf{b}, \quad (52)$$

with $\tilde{S}_C = I + \beta S_C$. The global solution \mathbf{w} can be rewritten as:

$$\begin{aligned}\mathbf{w} &= \mathbf{w}_S + \mathbf{w}_Z = C^{-1}(\mathbf{b} - \mathbf{b}_Z) + \mathbf{w}_Z \\ &= C^{-1}(\mathbf{b} - \hat{C}\mathbf{w}_Z + C\mathbf{w}_Z) = C^{-1}(\mathbf{b} - \beta FF^T \mathbf{w}_Z),\end{aligned}\quad (53)$$

and using \mathbf{w}_Z of equation (52) finally yields:

$$\mathbf{w} = C^{-1}(\mathbf{b} - \beta FF^T \mathbf{w}_Z) = C^{-1}(\mathbf{b} - \beta F \tilde{S}_C^{-1} F^T C^{-1} \mathbf{b}). \quad (54)$$

225 *Remark 3.3.* The result (54) can be also obtained applying directly the Sherman-Morrison-Woodbury identity to equation (21). Therefore, this relationship can be also regarded as the natural outcome of a particular projection operated on the matrix \hat{C} .

230 The solution of the inner system (21) through equation (54) requires two inner solves with C , which is a well-defined SPD matrix, and one with \tilde{S}_C . The latter is not available explicitly, however it can be easily and inexpensively approximated for \hat{K} and \hat{A} by using, for instance, D_K and D_A already adopted to compute α (see Section 2). Hence, an inexact solve with \tilde{S}_C is performed by using some approximation, say $M_{\tilde{S}}^{-1}$. Of course, also the solves with C can be performed inexactly, by means of another inner local preconditioner, M_C^{-1} , for SPD matrices. Algorithm 4 summarizes the procedure resulting from Method 2.

Algorithm 4 Method 2: $\mathbf{w} = \text{MET2_APPLY}(\beta, \mathbf{b}, F, M_C^{-1}, M_{\tilde{S}}^{-1})$

- 1: **Apply** M_C^{-1} to \mathbf{b} to get \mathbf{c}
 - 2: $\mathbf{d} = F^T \mathbf{c}$
 - 3: **Apply** $M_{\tilde{S}}^{-1}$ to \mathbf{d} to get \mathbf{g}
 - 4: $\mathbf{c} = F \mathbf{g}$
 - 5: $\mathbf{c} \leftarrow \mathbf{b} - \beta \mathbf{c}$
 - 6: **Apply** M_C^{-1} to \mathbf{c} to get \mathbf{w}
-

235 The ERPF2 preconditioner is obtained by replacing lines 3 and 8 of Algorithm 2 with the function in Algorithm 4 whenever $\alpha < \alpha_K$ and $\alpha < \alpha_A$, respectively. The complexity χ_2 of Algorithm 2 reads:

$$\chi_2 = 2\chi(M_C^{-1}) + \chi(M_{\tilde{S}}^{-1}) + \mathcal{O}(n), \quad (55)$$

where n is the number of rows of F , and $\chi(M_C^{-1})$ and $\chi(M_{\tilde{S}}^{-1})$ denote the complexity of the inner preconditioner applications. The overall cost for the preconditioner application per iteration increases with respect to the native RPF, however the overall algorithm is expected to take benefit from the acceleration of the global Krylov solver.

240 *Remark 3.4.* The use of Method 2 for solving the inner systems with both \hat{K} and \hat{A} is equivalent to compute the vector $\mathbf{t} = \mathbf{M}^{-1} \mathbf{r}$ using directly equation (5):

$$\begin{cases} \mathbf{y} = \alpha \mathbf{M}_1^{-1} \mathbf{r} \\ \mathbf{t} = \mathbf{M}_2^{-1} \mathbf{y} \end{cases}, \quad (56)$$

where \mathbf{M}_1 and \mathbf{M}_2 are inverted with the aid of the factorizations:

$$\mathbf{M}_1 = \begin{bmatrix} K & 0 & -Q \\ 0 & \alpha I_q & 0 \\ Q^T & 0 & \alpha I_p \end{bmatrix} = \begin{bmatrix} K & 0 & 0 \\ 0 & \alpha I_q & 0 \\ Q^T & 0 & \tilde{S}_K \end{bmatrix} \begin{bmatrix} I_u & 0 & -K^{-1}Q \\ 0 & I_q & 0 \\ 0 & 0 & \alpha I_p \end{bmatrix}, \quad (57)$$

$$\mathbf{M}_2 = \begin{bmatrix} I_u & 0 & 0 \\ 0 & A & -B \\ 0 & \gamma B^T & \alpha I_p \end{bmatrix} = \begin{bmatrix} I_u & 0 & 0 \\ 0 & A & 0 \\ 0 & \gamma B^T & \tilde{S}_A \end{bmatrix} \begin{bmatrix} I_u & 0 & 0 \\ 0 & I_q & -A^{-1}B \\ 0 & 0 & \alpha I_p \end{bmatrix}. \quad (58)$$

This leads to an alternative ERPF2 implementation, which is naturally more prone to a split preconditioning strategy. The use of the factorizations (57)-(58) into (56) whenever $\alpha < \alpha_K$ and $\alpha < \alpha_A$ provides the same numerical outcome

as the application of Algorithm 4. Details of such equivalence along with the alternative ERPF2 algorithm are provided in Appendix A.

4. Numerical Results

Two sets of numerical experiments are discussed in this section. The first set (Test 1) arises from Mandel’s problem [59], i.e., a classic benchmark of linear poroelasticity. This problem is used to verify the theoretical properties of the proposed methods. In the second set (Test 2), two real-world applications are considered in order to test the robustness and efficiency of the preconditioners.

We consider three variants of RPF, ERPF1, and ERPF2 according to the selection of the inner preconditioners. In essence, the first variant (\mathbf{M}_I) relies on applying “exactly” the inner preconditioners using nested direct solvers and aims at investigating the theoretical properties of the different approaches. The second (\mathbf{M}_II) and the third (\mathbf{M}_III) variants introduce further levels of approximation utilizing incomplete Cholesky (ic) and algebraic multigrid (AMG) preconditioners, respectively. Specifically:

- **RPF** (Algorithm 2): all exact/inexact solves involve \hat{K} in (16) and \hat{A} in (17) irrespective of the value of α ;
- **ERPF1** (Algorithm 3): Method 1 requires inner solves with matrices \hat{K}_α or \hat{A}_α that depend on the relaxation parameter α as follows

$$\hat{K}_\alpha = \begin{cases} \hat{K}_\ell, & \text{if } \alpha < \alpha_K \\ \hat{K}, & \text{if } \alpha \geq \alpha_K \end{cases}, \quad \hat{A}_\alpha = \begin{cases} \hat{A}_\ell, & \text{if } \alpha < \alpha_A \\ \hat{A}, & \text{if } \alpha \geq \alpha_A \end{cases}, \quad (59)$$

with \hat{K}_ℓ and \hat{A}_ℓ computed at α_K and α_A , respectively, using expression (29);

- **ERPF2** (Algorithm 4): Method 2 is used to replace the inner solves with \hat{K} or \hat{A} whenever either $\alpha < \alpha_K$ or $\alpha < \alpha_A$. It requires inner solves with K and \tilde{S}_K , or A and \tilde{S}_A . For \tilde{S}_K we use the diagonal matrix D_K employed in the computation of α and α_K (see Section 2) by defining:

$$\tilde{D}_K = I_p + \frac{1}{\alpha} D_K. \quad (60)$$

The application of \tilde{S}_K^{-1} is approximated by \tilde{D}_K^{-1} . For \tilde{S}_A we use the diagonal matrix \tilde{A}^{-1} employed in the computation of α and α_A (see Section 2, equation (13)) by defining:

$$\tilde{S}_A \approx I_p + \frac{\gamma}{\alpha} B^T \tilde{A}^{-1} B. \quad (61)$$

The inverse of \tilde{A} is used to approximate the application of A^{-1} as well.

Table 1 summarizes the different variants illustrated above. For the incomplete Cholesky factorization, we use an algorithm implementing a fill-in technique based on the selection of a user-specified row-wise number of entries in addition to those of the original matrix, as proposed in [60] and [61]. A classic algebraic multigrid method [62] is used as implemented in the HSL_MI20 package [63]. Of course, several other algebraic options for the inner blocks are possible.

The objective of this section is to demonstrate numerically that the proposed enhancements ERPF1 and ERPF2 overcome the drawbacks of the native RPF preconditioner. A comparison with other well-established block preconditioners, which has already been done for RPF in [47], is not performed here. Test 1 shows the stability and robustness of ERPF1 and ERPF2 to a variation of the mesh and timestep size for tightly coupled poromechanical conditions. Tests 2 give the performance in real-world heterogeneous problems, where the robustness to significant jumps in the hydromechanical parameters and the computational efficiency measured in terms of CPU time reduction are the main issues.

In all test cases, Bi-CGStab [24] is selected as a Krylov subspace method. Our focus is the robustness and computational efficiency in the solution of a single linear system. Hence, following standard practice [58], all tests

Table 1: RPF, ERPF1, and ERPF2 preconditioner variants.

Preconditioner	RPF	ERPF1	ERPF2 $\alpha < \alpha_K$	ERPF2 $\alpha < \alpha_A$
\mathbf{M}_I^{-1}	$\mathbf{M}_I^{-1}(\hat{K}^{-1}, \hat{A}^{-1})$	$\mathbf{M}_I^{-1}(\hat{K}_{\alpha}^{-1}, \hat{A}_{\alpha}^{-1})$	$\mathbf{M}_I^{-1}(K^{-1}, \tilde{D}_K^{-1}, \hat{A}^{-1})$	$\mathbf{M}_I^{-1}(\hat{K}^{-1}, \tilde{A}^{-1}, \tilde{S}_A^{-1})$
\mathbf{M}_{II}^{-1}	$\mathbf{M}_{II}^{-1}(\hat{K}_{IC}^{-1}, \hat{A}_{IC}^{-1})$	$\mathbf{M}_{II}^{-1}(\hat{K}_{\alpha, IC}^{-1}, \hat{A}_{\alpha, IC}^{-1})$	$\mathbf{M}_{II}^{-1}(K_{IC}^{-1}, \tilde{D}_K^{-1}, \hat{A}_{IC}^{-1})$	$\mathbf{M}_{II}^{-1}(\hat{K}_{IC}^{-1}, \tilde{A}^{-1}, \tilde{S}_{A, IC}^{-1})$
\mathbf{M}_{III}^{-1}	$\mathbf{M}_{III}^{-1}(\hat{K}_{AMG}^{-1}, \hat{A}_{AMG}^{-1})$	–	$\mathbf{M}_{III}^{-1}(K_{AMG}^{-1}, \tilde{D}_K^{-1}, \hat{A}_{AMG}^{-1})$	$\mathbf{M}_{III}^{-1}(\hat{K}_{AMG}^{-1}, \tilde{A}^{-1}, \tilde{S}_{A, AMG}^{-1})$

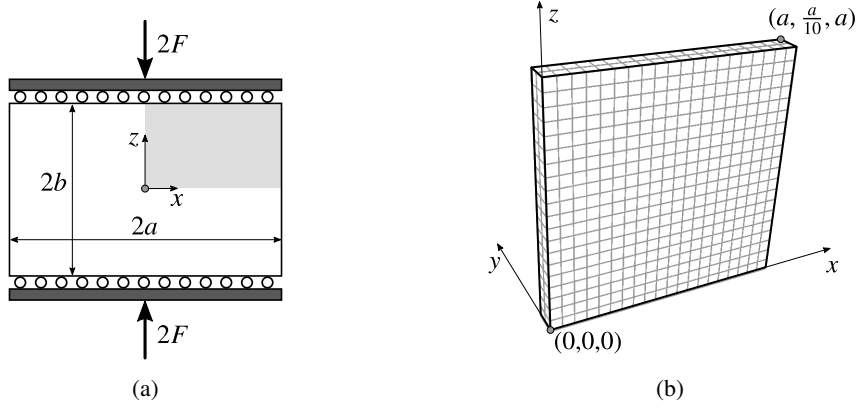


Figure 1: Test 1, Mandel's problem: sketch of the geometry (a) and the computational model (b).

280 are performed by setting a random solution vector \mathbf{x} and computing the corresponding right-hand side $\mathbf{F} = \mathbf{A}\mathbf{x}$. The iterations are stopped when the ratio between the 2-norm of the real residual vector and the 2-norm of the right-hand side is smaller than $\tau = 10^{-6}$. The computational performance is evaluated in terms of number of iterations n_{it} , CPU time in seconds for the preconditioner set-up T_p and for the solver to converge T_s . The total time is denoted by $T_t = T_p + T_s$. All computations are performed using a code written in FORTRAN90 on an Intel(R) Xeon(R) CPU
 285 E5-1620 v4 at 3.5 GHz Quad-Core with 64 GB of RAM memory.

4.1. Test1: Mandel's problem

This is a classical benchmark for validating coupled poromechanical models. The problem consists of a porous slab with section $2a \times 2b$, discretized by a cartesian grid, sandwiched between rigid, frictionless, and impermeable plates (Figure 1). For the sake of simplicity, we set $a = b$. A detailed description of the test case is reported in
 290 [12, 47]. This test case has mainly a theoretical value and is used to investigate the properties of ERPF1 and ERPF2 for a wide range of time step, Δt , and characteristic mesh size, h , values. In particular, four grids progressively refined are considered as shown in Table 2. The time-step size is defined with respect to the characteristic consolidation time t_c of the problem. This allows for testing the solver behavior at any stage of the consolidation process independently of the specific material parameters, i.e., the medium compressibility and permeability. Hence, the analysis carried out
 295 in this test case provides also information about the parameter-robustness of the proposed algorithms.

First of all, we analyze how the eigenspectrum of $\hat{K}_{\ell}^{-1}\hat{K}$ and $\hat{A}_{\ell}^{-1}\hat{A}$ changes with the ratios α/α_K and α/α_A , respectively. Figure 2 provides such eigenspectra for the coarsest discretization ($a/h = 10$). As the separation between α and the limiting values α_K and α_A increases, the largest eigenvalue progressively grows and the eigenspectrum is less and less clustered, as theoretically predicted by the result of Theorem 3.1. This is basically due to the increase
 300 of the condition number of \hat{K} and \hat{A} with respect to that of K and A , respectively. The numerical values for the cases reported in Figure 2 are provided in Table 3, while the general behavior obtained varying α is shown in Figure 3. In other words, \hat{K} and \hat{A} are getting closer and closer to a singular matrix as $\alpha \rightarrow 0$, and consequently the global RPF performance is expected to get worse. The application of Method 1 and Method 2 allows to overcome such an issue.

Table 2: Test 1, Mandel's problem: grid refinement and problem size.

a/h	number of elements	n_u	n_q	n_p
10	$10 \times 1 \times 10$	726	420	100
20	$20 \times 2 \times 20$	3,969	2,880	800
40	$40 \times 4 \times 40$	25,215	21,120	6,400
80	$80 \times 8 \times 80$	177,147	161,280	51,200

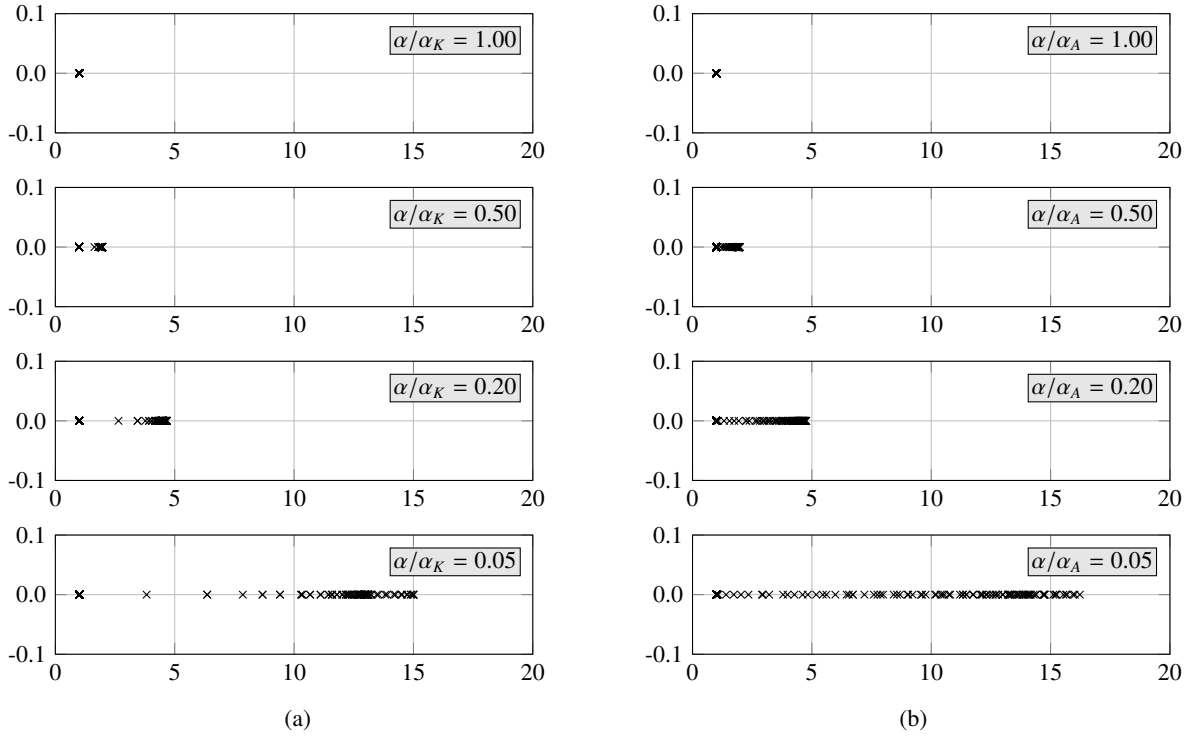


Figure 2: Test 1, Mandel's problem ($a/h = 10$): eigenspectrum of $\hat{K}_\ell^{-1} \hat{K}$ and $\hat{A}_\ell^{-1} \hat{A}$ varying α/α_K (a) and α/α_A (b).

Table 3: Test 1, Mandel's problem ($a/h = 10$): condition number κ of \hat{K} and \hat{A} varying α/α_ℓ , α_ℓ being either α_K or α_A . The condition number of K and A are: $\kappa(K) = 2.14 \cdot 10^5$, $\kappa(A) = 9.84 \cdot 10^9$.

α/α_ℓ	$\kappa(\hat{K})$	$\kappa(\hat{A})$
1.00	$8.47 \cdot 10^6$	$1.21 \cdot 10^{12}$
0.50	$1.67 \cdot 10^7$	$2.42 \cdot 10^{12}$
0.20	$4.15 \cdot 10^7$	$6.05 \cdot 10^{12}$
0.05	$1.65 \cdot 10^8$	$2.41 \cdot 10^{13}$

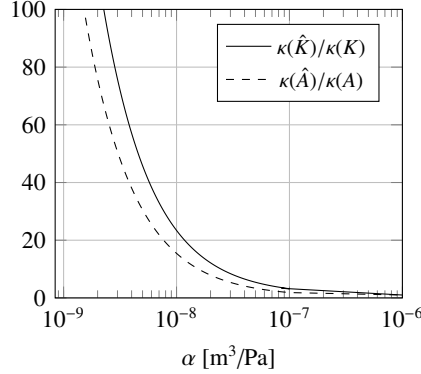


Figure 3: Test 1, Mandel’s problem ($a/h = 10$): variation of the condition number κ of \hat{K} and \hat{A} with respect to that of K and A , respectively, vs α .

The preconditioner variant \mathbf{M}_1^{-1} uses nested direct solvers, hence it has mainly a theoretical value. It is useful to isolate the impact of the proposed schemes (Algorithms 3 and 4) on the convergence behavior by varying the discretization steps in both space and time. We recall on passing that $\gamma = \theta\Delta t$, with $\theta \in [0.5, 1]$, and that α is proportional to $\sqrt{\gamma}$, α_K is constant with γ , and α_A depends linearly on γ (see Section 2). Hence:

$$\lim_{\Delta t \rightarrow 0} \frac{\alpha}{\alpha_K} = 0, \quad \lim_{\Delta t \rightarrow +\infty} \frac{\alpha}{\alpha_A} = 0, \quad (62)$$

i.e., the conditions $\alpha < \alpha_K$ and $\alpha < \alpha_A$ are encountered for small and large time-step sizes, respectively, and are not likely to be satisfied simultaneously.

Table 4 provides the outer iteration count obtained in Mandel’s problem with ERPF1 by varying the mesh-size, the time-step and the number of inner iterations n_{in} . As expected, the number of outer iterations decreases when growing the number of inner iterations. Since the computational cost for ERPF1 application increases with n_{in} , setting $n_{in} = 2$ or $n_{in} = 3$ appears to be already a good trade-off between solver acceleration and computational cost per iteration.

The ERPF1 effectiveness is actually dependent on the value of α/α_K or α/α_A . Lemma 3.2 shows that the convergence rate of the inner stationary method used with \hat{K} or \hat{A} tends to 0 with the ratio α/α_K or α/α_A . This is confirmed by Table 4, which shows a performance deterioration for both α/α_K and α/α_A approaching 0. Such a deterioration, however, appears to be much more pronounced with α/α_A , i.e., for large values of the time-step Δt (see Remark 3.1). Table 4 shows also a mild dependency on the mesh-size h , with the outer iteration count increasing moderately only for large Δt values.

A similar analysis is carried out for ERPF2 with the \mathbf{M}_1^{-1} variant. Table 5 provides the results in terms of outer iteration count by varying Δt and h . This approach turns out to be stable with respect to variations of both Δt and h , also for extreme values of the time-step size. A comparison with Table 4 reveals also that ERPF2 appears to be more effective than ERPF1, as far as the number of outer iterations is concerned, especially when $\alpha < \alpha_A$, i.e., large time steps. For this reason, we can conclude that ERPF2 should be usually preferred as a full poromechanical simulation proceeds towards steady-state conditions.

Of course, the results of Table 4 and 5 can change when the nested direct solvers are replaced by inner preconditioners, such as with the variants \mathbf{M}_n^{-1} and \mathbf{M}_m^{-1} . For example, Figure 4 and 5 compare the convergence profile for $a/h = 10$ and $a/h = 20$, respectively, and very large values of the time step ($\Delta t = 10^2 t_c$ and $\Delta t = 10^6 t_c$) obtained with the original RPF (\mathbf{M}_1^{-1} , \mathbf{M}_n^{-1} , and \mathbf{M}_m^{-1} variants) and with ERPF2 (\mathbf{M}_n^{-1} and \mathbf{M}_m^{-1} variants). All incomplete Cholesky factorizations are computed in this case with zero fill-in, while algebraic multigrid is applied keeping the default parameters. It can be observed that, even in this academic example, the original RPF preconditioner with an inexact solve for \hat{K} or \hat{A} might not converge with both \mathbf{M}_n^{-1} and \mathbf{M}_m^{-1} variants. Hence, this outcome appears to be an issue related to the nature of \hat{K} and \hat{A} , rather than the choice of the inner preconditioner. By distinction, the ERPF2 performance is just moderately affected by the use of inexact solves, exhibiting a very stable convergence profile with respect to a Δt variation. proposed algorithms are able to enhance not only the RPF performance, but also its robustness.

Table 4: Test 1, Mandel's problem: number of outer iteration with ERPF1 and \mathbf{M}_1^{-1} by varying the mesh-size h , the time-step Δt and the inner iterations n_{in} . The time-step size is relative to the characteristic consolidation time $t_c = 900$ s [12, 47].

a/h	$\Delta t/t_c$	α/α_K	α/α_A	# of outer iterations				
				$n_{in} = 1$	$n_{in} = 2$	$n_{in} = 3$	$n_{in} = 4$	$n_{in} = 5$
10	$5.6 \cdot 10^{-7}$	0.10	> 1	12	9	7	5	3
	$3.4 \cdot 10^{-6}$	0.25	> 1	10	7	4	3	3
	$1.7 \cdot 10^{-4}$	0.50	> 1	7	4	4	3	3
	$1.0 \cdot 10^{-3}$	> 1	> 1	8	–	–	–	–
	$5.0 \cdot 10^{-1}$	> 1	0.50	14	10	8	8	8
	$1.7 \cdot 10^{+0}$	> 1	0.25	22	12	10	9	8
	$5.6 \cdot 10^{+0}$	> 1	0.10	24	15	11	11	10
20	$5.6 \cdot 10^{-7}$	0.10	> 1	12	9	6	4	3
	$3.4 \cdot 10^{-6}$	0.25	> 1	9	7	4	3	2
	$1.7 \cdot 10^{-4}$	0.50	> 1	7	4	3	3	3
	$1.0 \cdot 10^{-3}$	> 1	> 1	7	–	–	–	–
	$5.0 \cdot 10^{-1}$	> 1	0.50	17	12	12	11	11
	$1.7 \cdot 10^{+0}$	> 1	0.25	29	16	13	13	12
	$5.6 \cdot 10^{+0}$	> 1	0.10	38	22	16	14	13

Table 5: Test 1, Mandel's problem: number of outer iterations with ERPF2 and \mathbf{M}_1^{-1} by varying the mesh-size h and the time-step Δt .

a/h	$\Delta t/t_c$	α/α_K	# of iter.	$\Delta t/t_c$	α/α_A	# of iter.
10	10^{-6}	$1.3 \cdot 10^{-1}$	5	10^2	$3.5 \cdot 10^{-2}$	8
	10^{-7}	$4.3 \cdot 10^{-2}$	6	10^3	$1.1 \cdot 10^{-2}$	8
	10^{-8}	$1.4 \cdot 10^{-2}$	6	10^4	$3.5 \cdot 10^{-3}$	7
20	10^{-6}	$3.0 \cdot 10^{-1}$	6	10^2	$1.9 \cdot 10^{-2}$	8
	10^{-7}	$9.5 \cdot 10^{-2}$	6	10^3	$6.0 \cdot 10^{-3}$	7
	10^{-8}	$3.0 \cdot 10^{-2}$	6	10^4	$1.9 \cdot 10^{-3}$	7
40	10^{-6}	$1.9 \cdot 10^{-1}$	5	10^2	$9.5 \cdot 10^{-3}$	9
	10^{-7}	$6.0 \cdot 10^{-2}$	6	10^3	$3.0 \cdot 10^{-3}$	7
	10^{-8}	$1.8 \cdot 10^{-2}$	6	10^4	$9.5 \cdot 10^{-4}$	7
80	10^{-6}	$7.9 \cdot 10^{-1}$	6	10^2	$6.2 \cdot 10^{-3}$	11
	10^{-7}	$2.5 \cdot 10^{-1}$	6	10^3	$1.9 \cdot 10^{-3}$	8
	10^{-8}	$8.0 \cdot 10^{-2}$	6	10^4	$6.2 \cdot 10^{-4}$	7

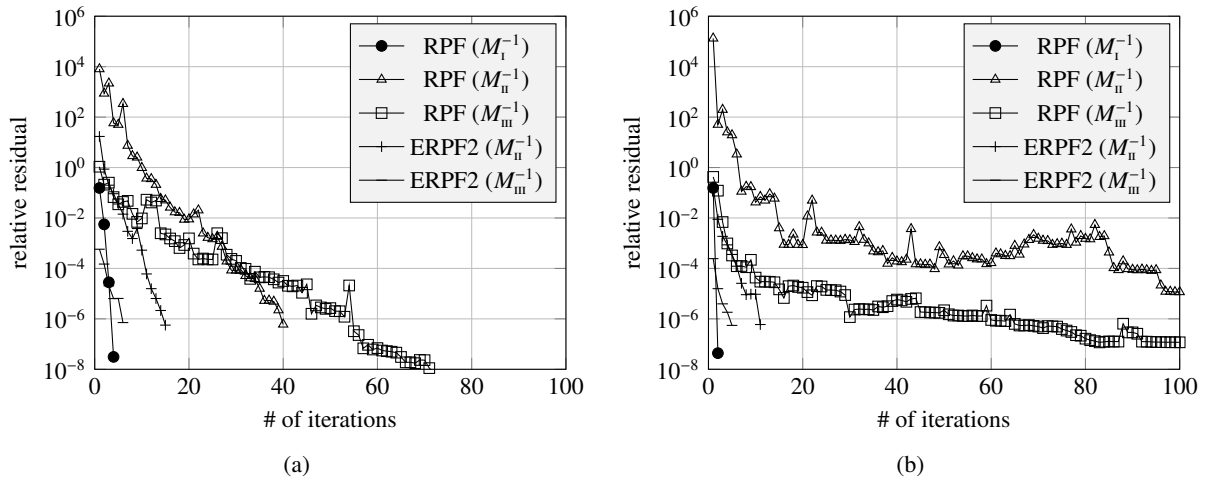


Figure 4: Test 1, Mandel's problem: Convergence profiles for $\Delta t = 10^2 t_c$ (a) and $\Delta t = 10^6 t_c$ (b) with $a/h = 10$.

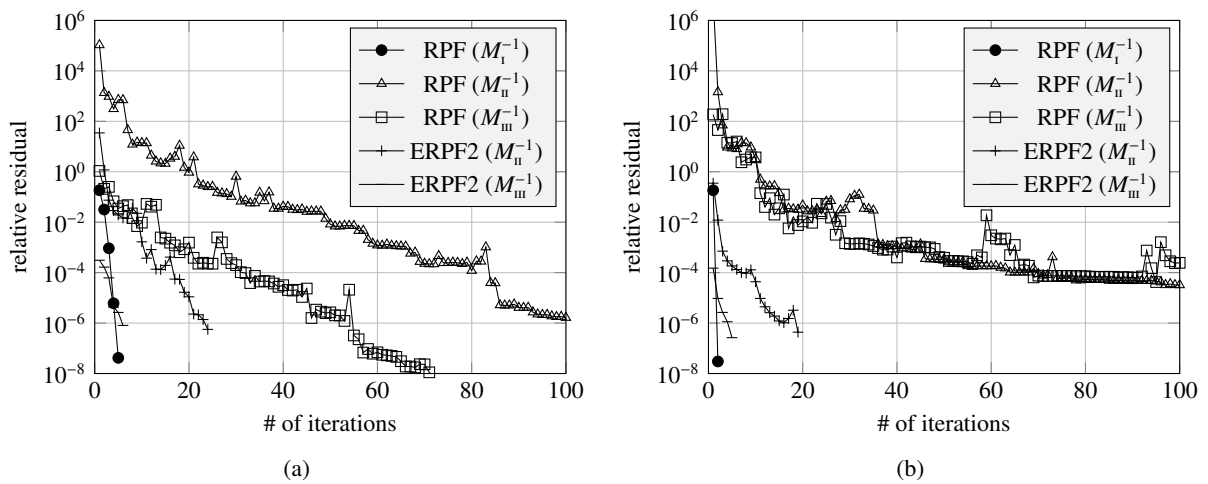


Figure 5: Test 1, Mandel's problem: the same as Figure 4 for $a/h = 20$.

Table 6: Test 1, Mandel’s problem: number of outer iterations with ERPF2 and $\mathbf{M}_{\text{in}}^{-1}$ by varying the mesh-size h and the time-step Δt .

a/h	$\Delta t/t_c$	α/α_K	# of iter.	$\Delta t/t_c$	α/α_A	# of iter.
10	10^{-6}	$1.3 \cdot 10^{-1}$	14	10^2	$3.5 \cdot 10^{-2}$	12
	10^{-7}	$4.3 \cdot 10^{-2}$	14	10^3	$1.1 \cdot 10^{-2}$	14
	10^{-8}	$1.4 \cdot 10^{-2}$	14	10^4	$3.5 \cdot 10^{-3}$	14
20	10^{-6}	$3.0 \cdot 10^{-1}$	19	10^2	$1.9 \cdot 10^{-2}$	13
	10^{-7}	$9.5 \cdot 10^{-2}$	19	10^3	$6.0 \cdot 10^{-3}$	12
	10^{-8}	$3.0 \cdot 10^{-2}$	19	10^4	$1.9 \cdot 10^{-3}$	14
40	10^{-6}	$1.9 \cdot 10^{-1}$	22	10^2	$9.5 \cdot 10^{-3}$	11
	10^{-7}	$6.0 \cdot 10^{-2}$	19	10^3	$3.0 \cdot 10^{-3}$	12
	10^{-8}	$1.8 \cdot 10^{-2}$	17	10^4	$9.5 \cdot 10^{-4}$	11
80	10^{-6}	$7.9 \cdot 10^{-1}$	12	10^2	$6.2 \cdot 10^{-3}$	16
	10^{-7}	$2.5 \cdot 10^{-1}$	19	10^3	$1.9 \cdot 10^{-3}$	14
	10^{-8}	$8.0 \cdot 10^{-2}$	19	10^4	$6.2 \cdot 10^{-4}$	11

The use of incomplete Cholesky factorizations with a partial fill-in as inner preconditioners of SPD blocks is efficient in sequential computations, but can create concerns in parallel environments. Moreover, as it is well-known, it can prevent from obtaining an optimal weak scalability with respect to the mesh size h . For instance, this can be observed in Figure 4 and 5, ERPF2 profiles, where the number of iterations is not constant with h by using incomplete Cholesky factorizations. The scalability with h can be restored by using scalable inner preconditioners, such as algebraic multigrid methods. The same analysis as Table 5 is here carried out by using the $\mathbf{M}_{\text{in}}^{-1}$ variant. Table 6 provides the outer iteration counts by varying Δt and h . As expected, the scalability with h is much improved, since the number of iterations is quite stable between the progressive refinements.

4.2. Test2: Real-world applications

The ERPF performance is finally tested in two large-size challenging cases, with unstructured grids and severe material anisotropy and heterogeneity. In particular, the solver performance is investigated in presence of strong variability of both mechanical, i.e. compressibility, and hydraulic, i.e. permeability and porosity, material parameters, with severe jumps occurring between adjacent cells. This condition poses particular challenges as to the algorithmic robustness of the proposed methods. We have considered two real-world applications:

- Test 2a: Chaobai. This model is used to predict land subsidence due to a shallow multi-aquifer system over-exploitation in the Chaobai River alluvial fan, North of Beijing, China [64]. The strong heterogeneity of the alluvial fan, which covers an overall areal extent of more than 1,100 km², is accounted for by means of a statistical inverse framework in a multi-zone transition probability approach [65, 66]. Figure 6a shows a reconstruction of the lithofacies distribution. Details on the model implementation are provided in [64];
- Test 2b: SPE10. This is a typical petroleum reservoir engineering application reproducing a well-driven flow in a deforming porous medium. The model setup is based on the 10th SPE Comparative Solution Project [67], a well-known severe benchmark in reservoir applications, assuming a poroelastic mechanical behavior with incompressible fluid and solid constituents. The porous medium is populated with homogeneous elastic properties, namely, Young’s modulus $E = 8.3 \cdot 10^3$ MPa, Poisson’s ratio $\nu = 0.3$, and Biot’s coefficient $b = 1.0$. The model is limited to the top 35 layers, which are representative of a shallow-marine Tarbert formation characterized by extreme permeability variations (Figure 6b), covering an areal extent of 365.76×650.56 m² and for 21.34 m in the vertical direction. One injector and one production well, located at opposite corners of

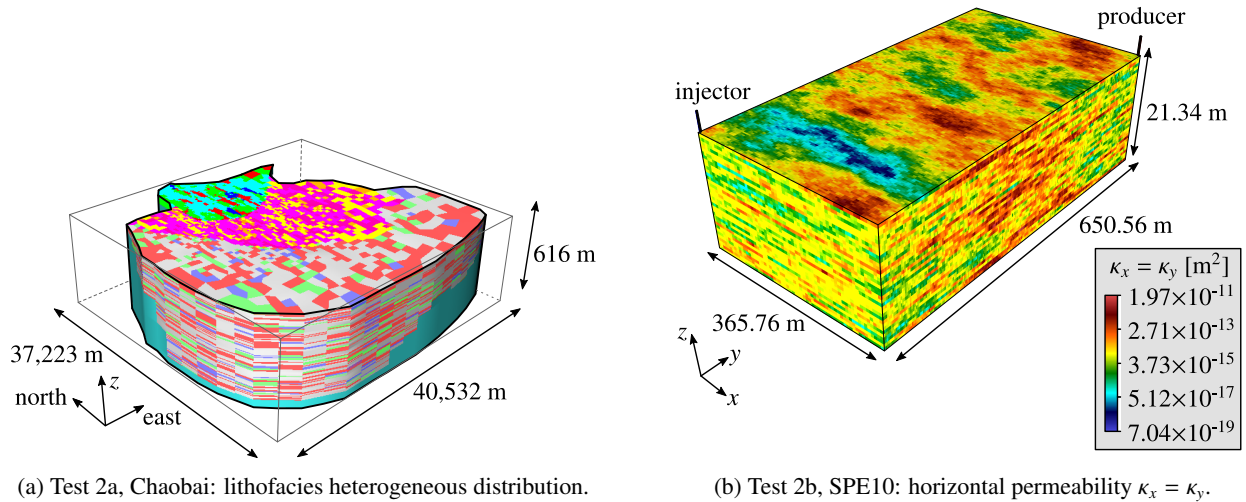


Figure 6: Test 2, Real-world applications: heterogeneous property distribution in the porous domain for the Chaobai (a) and SPE10 (b) test cases.

Table 7: Test 2, real-world applications: size and number of non-zeros of the test matrices.

Test	n_u	n_q	n_p	# non-zeros
2a: Chaobai	2,152,683	2,132,612	707,600	143,359,342
2b: SPE10	1,455,948	1,409,000	462,000	94,317,731

the domain, penetrate vertically the entire reservoir and drive the porous fluid flow. The reader can refer to [68] for additional details.

365 The size and the number of non-zeros of the matrices arising from Test 2a and 2b are provided in Table 7.

The size of the real-world problems addressed in Test 2 prevents from the use of the \mathbf{M}_r^{-1} variant with nested direct solvers. Therefore, we compare the performance of the original RPF with ERPF1 and ERPF2 in the \mathbf{M}_i^{-1} variant, which makes use of incomplete Cholesky factorizations with partial fill-in as inner preconditioners. We employ a limited-memory implementation [61], where the user can specify the row-wise number of entries ρ to be retained in addition to the row-wise number of non-zeros of the original matrix. The parameters of the inner preconditioners are set up to minimize the total CPU time T_t with the RPF preconditioner and intermediate values of the time-step size Δt ($10^{-4} < \Delta t < 10^0$ day for Test 2a and $10^{-6} < \Delta t < 10^0$ day for Test 2b).

375 Table 8 provides the results obtained for Test 2a in terms of iteration count, solver CPU time T_s and total CPU time T_t . For small Δt values, i.e., when $\alpha < \alpha_K$, ERPF1 outperforms ERPF2, providing smaller iteration counts and total CPU times. In this test case, ERPF1 turns out to be comparable with the original RPF, which generally proves, however, less robust. As Δt increases, the iteration count with RPF quickly grows and soon becomes unacceptable. With ERPF1, the iterations to converge increase as well, though at a lesser extent, while they remain stable when using ERPF2. In other words, inspection of Table 8 reveals that, in a full transient problem where the time-step size typically increases as the simulation proceeds towards the steady state, the most efficient strategy consists of switching from ERPF1 when $\alpha < \alpha_K$ to ERPF2 when $\alpha < \alpha_A$, preserving the original RPF for the intermediate steps.

380 Table 9 provides the same results as Table 8 for Test 2b. In this case, the condition $\alpha < \alpha_K$ is never met for time-step sizes with a relevant physical meaning. Therefore, we report only the performance obtained by the original RPF and the ERPF2 methods, being ERPF1 more convenient for small Δt only. As in Test 2a, ERPF2 always outperforms the original RPF preconditioner, proving much more robust and practically time-step independent.

Table 8: Test 2a, Chaobai: iteration count and CPU times for a variable time-step size by using the \mathbf{M}_u^{-1} variant as a preconditioner (Table 1). The fill-in parameter for: \hat{K}_{ic}^{-1} , $\hat{K}_{t,ic}^{-1}$, and K_{ic}^{-1} is $\rho_K = 60$; \hat{A}_{ic}^{-1} and $\hat{A}_{t,ic}^{-1}$ is $\rho_A = 50$; $\tilde{S}_{A,ic}^{-1}$ is $\rho_S = 40$. The number of inner iteration for ERPF1 is 2.

Δt [day]	α/α_K	α/α_A	RPF			ERPF1			ERPF2		
			n_{it}	T_s [s]	T_t [s]	n_{it}	T_s [s]	T_t [s]	n_{it}	T_s [s]	T_t [s]
10^{-6}	$1.0 \cdot 10^{-2}$	> 1	49	124.9	203.9	25	102.2	181.4	58	216.0	286.6
10^{-5}	$2.8 \cdot 10^{-2}$	> 1	45	114.2	192.4	32	133.8	212.9	54	201.4	271.8
10^{-4}	$9.1 \cdot 10^{-2}$	> 1	45	114.7	193.1	29	118.2	197.2	46	170.1	240.1
10^0	> 1	$2.8 \cdot 10^{-1}$	247	629.7	709.1	179	582.9	661.4	134	274.4	334.0
10^1	> 1	$9.4 \cdot 10^{-2}$	> 500	-	-	273	884.0	962.4	127	260.3	321.2
10^2	> 1	$2.8 \cdot 10^{-2}$	> 500	-	-	386	1247.5	1326.1	135	277.4	337.2

Table 9: Test 2b, SPE10: the same as Table 8 for RPF and ERPF2. The fill-in parameter for: \hat{K}_{ic}^{-1} is $\rho_K = 30$; \hat{A}_{ic}^{-1} is $\rho_A = 40$; $\tilde{S}_{A,ic}^{-1}$ is $\rho_S = 40$.

Δt [day]	α/α_K	α/α_A	RPF			ERPF2		
			n_{it}	T_s [s]	T_t [s]	n_{it}	T_s [s]	T_t [s]
10^0	> 1	$9.2 \cdot 10^{-4}$	348	467.7	503.3	84	79.8	102.2
10^1	> 1	$2.9 \cdot 10^{-4}$	> 1000	-	-	87	82.7	104.5
10^2	> 1	$9.2 \cdot 10^{-5}$	> 1000	-	-	47	44.8	67.3

5. Conclusions

The Relaxed Physical Factorization introduced in [47] has proved an efficient preconditioning strategy for the three-field formulation of coupled poromechanics with respect to domain heterogeneity, anisotropy and grid distortion. However, a performance degradation was observed for the values of time-step size that are typically required at the beginning of a full transient simulation and toward steady-state conditions. In fact, at the beginning of the poromechanical process very small time steps are necessary to capture accurately the pressure and deformation evolution in almost undrained conditions, while larger and larger steps are convenient when proceeding towards the steady state. The origin of such a drawback stems from the need of inverting in an inexact way inner blocks in the form $\hat{C} = C + \beta FF^T$ with $\beta \gg 1$, i.e., where the rank-deficient term FF^T prevails on the regular matrix C .

Two fully algebraic methods are presented to address the RPF issues:

1. a natural splitting of \hat{C} is introduced to define a particular stationary scheme. The scheme is unconditionally convergent and does not require the inversion of \hat{C} . Hence, the inner solve with \hat{C} is replaced by a few iterations of such a scheme;
2. an appropriate projection operator is developed in order to annihilate the components of the solution vector of $\hat{C}\mathbf{w} = \mathbf{b}$ lying in the near-null space of \hat{C} . The projected system is solved in a stable way, while the remaining part of the solution vector can be computed requiring the inversion of the regular SPD matrix C only, instead of \hat{C} .

The proposed methods can be generalized to applications where a similar algebraic issue arises, such as in the use of an augmented Lagrangian approach for Navier-Stokes or incompressible elasticity. In the context of coupled poromechanics, they give rise to an Enhanced RPF preconditioner, ERPF1 and ERPF2, respectively, which has been tested in both theoretical benchmarks and challenging real-world large-size applications. The following results are worth summarizing.

- Both methods are effective in improving the performance and robustness of the original RPF algorithm in the most critical situations, i.e., $\Delta t \rightarrow 0$ and $\Delta t \rightarrow \infty$, stabilizing the iteration counts to convergence independently of the time step size.
- 410 • ERPF1 is usually more efficient for small time step values, while ERPF2 outperforms, also by a large amount, RPF and ERPF1 for large time step values. Therefore, the most convenient approach in a full transient poromechanical application appears to be switching from ERPF1 when $\alpha < \alpha_K$ at the simulation beginning, to RPF when $\alpha > \alpha_K$ and $\alpha > \alpha_A$ with intermediate steps, to ERPF2 when $\alpha < \alpha_A$ approaching steady state conditions.
- 415 • The use of nested direct solvers ensures a scalable behavior of ERPF2 with respect to both the mesh and time spacings. Such a property is generally lost by using inexact inner solves, for instance by incomplete factorizations, but can be restored with scalable inner preconditioners, such as algebraic multigrid methods.

Acknowledgments

Partial funding was provided by TotalEnergies S.A. through the FC-MAELSTROM Project. Portions of this work were performed within the 2020 INdAM-GNCS project ‘‘Optimization and advanced linear algebra for PDE-governed problems’’. 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. The authors are grateful to two anonymous reviewers, whose comments significantly helped improve the presentation.

Appendix A. Alternative ERPF2 application

With the ERPF2 approach, the inner solution to $\hat{C}\mathbf{w} = \mathbf{b}$ is obtained through equation (54) whenever $\beta > \beta_\ell$. With specific reference to Algorithm 2, this can happen twice at each preconditioner application either:

1. for $\alpha < \alpha_K$, having set $C = K$, $F = Q$, and $\beta = \alpha^{-1}$, or
2. for $\alpha < \alpha_A$, having set $C = A$, $F = B$, and $\beta = \gamma/\alpha$.

For the sake of brevity, let us consider the case no. 2, which typically occurs more often in practical applications, e.g., for large values of the time-step size. Similar considerations hold for the case no. 1.

430 Consider Algorithm 2. Recalling that $\mathbf{z}_q = \mathbf{r}_q + \alpha^{-1}B\mathbf{y}_p$ (row 7), the application of \hat{A}^{-1} on \mathbf{z}_q to get \mathbf{t}_q using equation (54) yields:

$$\begin{aligned}
\mathbf{t}_q &= \hat{A}^{-1}\mathbf{z}_q \\
&= A^{-1}\mathbf{r}_q + \alpha^{-1}A^{-1}B\mathbf{y}_p - \beta A^{-1}B\tilde{S}_A^{-1}B^T A^{-1}\mathbf{r}_q - \beta\alpha^{-1}A^{-1}B\tilde{S}_A^{-1}S_A\mathbf{y}_p \\
&= A^{-1}\mathbf{r}_q + \alpha^{-1}A^{-1}B\tilde{S}_A^{-1}(\tilde{S}_A - \beta S_A)\mathbf{y}_p - \beta A^{-1}B\tilde{S}_A^{-1}B^T A^{-1}\mathbf{r}_q \\
&= A^{-1}\mathbf{r}_q + \alpha^{-1}A^{-1}B\tilde{S}_A^{-1}\mathbf{y}_p - \beta A^{-1}B\tilde{S}_A^{-1}B^T A^{-1}\mathbf{r}_q \\
&= A^{-1}\mathbf{r}_q + A^{-1}B\tilde{S}_A^{-1}(\alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q) \\
&= A^{-1}\left[\mathbf{r}_q + B\tilde{S}_A^{-1}(\alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q)\right].
\end{aligned} \tag{A.1}$$

Recalling that $\beta = \gamma/\alpha$, lines 9 and 10 of Algorithm 2 can be rearranged as follows:

$$\begin{aligned}
\mathbf{t}_p &= \alpha^{-1}\mathbf{y}_p - \beta B^T \mathbf{t}_q \\
&= \alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q - \beta B^T A^{-1}B\tilde{S}_A^{-1}(\alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q) \\
&= (I_p - \beta S_A \tilde{S}_A^{-1})(\alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q) \\
&= \tilde{S}_A^{-1}(\alpha^{-1}\mathbf{y}_p - \beta B^T A^{-1}\mathbf{r}_q).
\end{aligned} \tag{A.2}$$

Algorithm 5 Alternative ERPF2 application: $[\mathbf{t}_u^T, \mathbf{t}_q^T, \mathbf{t}_p^T] = \text{ALTERPF2_APPLY}(\mathbf{r}_u, \mathbf{r}_q, \mathbf{r}_p, \mathbf{A}, \mathbf{M})$

```

1: if  $\alpha > \alpha_K$  then
2:    $\mathbf{x}_u = \alpha^{-1} Q \mathbf{r}_p$ 
3:    $\mathbf{x}_u \leftarrow \mathbf{r}_u + \mathbf{x}_u$ 
4:   Apply  $M_K^{-1}$  to  $\mathbf{x}_u$  to get  $\mathbf{t}_u$ 
5:    $\mathbf{y}_p = Q^T \mathbf{t}_u$ 
6:    $\mathbf{y}_p \leftarrow \mathbf{r}_p - \mathbf{y}_p$ 
7: else
8:   Apply  $M_K^{-1}$  to  $\mathbf{r}_u$  to get  $\mathbf{x}_u$ 
9:    $\mathbf{x}_p = Q^T \mathbf{x}_u$ 
10:   $\mathbf{x}_p \leftarrow \mathbf{r}_p - \mathbf{x}_p$ 
11:  Apply  $M_{\tilde{S}_K}^{-1}$  to  $\mathbf{x}_p$  to get  $\mathbf{y}_p$ 
12:   $\mathbf{x}_u = Q \mathbf{y}_p$ 
13:   $\mathbf{x}_u \leftarrow \alpha \mathbf{r}_u + \mathbf{x}_u$ 
14:  Apply  $M_K^{-1}$  to  $\mathbf{x}_u$  to get  $\mathbf{t}_u$ 
15: end if
16: if  $\alpha > \alpha_A$  then
17:    $\mathbf{z}_q = B \mathbf{y}_p$ 
18:    $\mathbf{z}_q \leftarrow \mathbf{r}_q + \alpha^{-1} \mathbf{z}_q$ 
19:   Apply  $M_{\hat{A}}^{-1}$  to  $\mathbf{z}_q$  to get  $\mathbf{t}_q$ 
20:    $\mathbf{t}_p = B^T \mathbf{t}_q$ 
21:    $\mathbf{t}_p \leftarrow \alpha^{-1} (\mathbf{y}_p - \gamma \mathbf{t}_p)$ 
22: else
23:   Apply  $M_A^{-1}$  to  $\mathbf{r}_q$  to get  $\mathbf{z}_q$ 
24:    $\mathbf{z}_p = B^T \mathbf{z}_q$ 
25:    $\mathbf{y}_p \leftarrow \alpha^{-1} (\mathbf{y}_p - \gamma \mathbf{z}_p)$ 
26:   Apply  $M_{\tilde{S}_A}^{-1}$  to  $\mathbf{y}_p$  to get  $\mathbf{t}_p$ 
27:    $\mathbf{z}_q = B \mathbf{t}_p$ 
28:    $\mathbf{z}_q \leftarrow \mathbf{r}_q + \mathbf{z}_q$ 
29:   Apply  $M_A^{-1}$  to  $\mathbf{z}_q$  to get  $\mathbf{t}_q$ 
30: end if

```

The vectors \mathbf{t}_p and \mathbf{t}_q are therefore computed by:

$$\mathbf{t}_p = \tilde{S}_A^{-1} (\alpha^{-1} \mathbf{y}_p - \beta B^T A^{-1} \mathbf{r}_q), \quad (\text{A.3})$$

$$\mathbf{t}_q = A^{-1} (\mathbf{r}_q + B \mathbf{t}_p). \quad (\text{A.4})$$

Equations (A.3) and (A.4) shows that Method 2 used for the inner solve with \hat{A} is equivalent to apply the inverse of the following block factorization for \mathbf{M}_2 :

435

$$\begin{bmatrix} I_u & 0 & 0 \\ 0 & A & -B \\ 0 & \gamma B^T & \alpha I_p \end{bmatrix} = \begin{bmatrix} I_u & 0 & 0 \\ 0 & A & 0 \\ 0 & \gamma B^T & \tilde{S}_A \end{bmatrix} \begin{bmatrix} I_u & 0 & 0 \\ 0 & I_q & -A^{-1} B \\ 0 & 0 & \alpha I_p \end{bmatrix}. \quad (\text{A.5})$$

In fact, solving the block system $\mathbf{M}_2 \mathbf{t} = \mathbf{y}$ with the aid of (A.5) yields:

$$\begin{pmatrix} \mathbf{t}_u \\ \mathbf{t}_q \\ \mathbf{t}_p \end{pmatrix} = \begin{bmatrix} I_u & 0 & 0 \\ 0 & I_q & \alpha^{-1} A^{-1} B \\ 0 & 0 & \alpha^{-1} I_p \end{bmatrix} \begin{bmatrix} I_u & 0 & 0 \\ 0 & A^{-1} & 0 \\ 0 & -\gamma \tilde{S}_A^{-1} B^T A^{-1} & \tilde{S}_A^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{y}_u \\ \mathbf{y}_q \\ \mathbf{y}_p \end{pmatrix}, \quad (\text{A.6})$$

which provides equations (A.3)-(A.4) for $\mathbf{y}_q = \mathbf{r}_q$.

With similar arguments, after some algebra it can be easily proved that Method 2 for the inner solve with \hat{K} is equivalent to apply the inverse of the following factorization of \mathbf{M}_1 :

$$\begin{bmatrix} K & 0 & -Q \\ 0 & \alpha I_q & 0 \\ Q^T & 0 & \alpha I_p \end{bmatrix} = \begin{bmatrix} K & 0 & 0 \\ 0 & \alpha I_q & 0 \\ Q^T & 0 & \tilde{S}_K \end{bmatrix} \begin{bmatrix} I_u & 0 & -K^{-1}Q \\ 0 & I_q & 0 \\ 0 & 0 & \alpha I_p \end{bmatrix}. \quad (\text{A.7})$$

440 The solution to the system $\alpha \mathbf{M}_1 \mathbf{y} = \mathbf{r}$ using (A.7) reads:

$$\begin{pmatrix} \mathbf{y}_u \\ \mathbf{y}_q \\ \mathbf{y}_p \end{pmatrix} = \alpha \begin{bmatrix} I_u & 0 & \alpha^{-1} K^{-1} Q \\ 0 & I_q & 0 \\ 0 & 0 & \alpha^{-1} I_p \end{bmatrix} \begin{bmatrix} K^{-1} & 0 & 0 \\ 0 & \alpha^{-1} I_q & 0 \\ -\tilde{S}_K^{-1} Q^T K^{-1} & 0 & \tilde{S}_K^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_q \\ \mathbf{r}_p \end{pmatrix}, \quad (\text{A.8})$$

which provides:

$$\mathbf{y}_p = \tilde{S}_K^{-1} (\mathbf{r}_p - Q^T K^{-1} \mathbf{r}_u), \quad (\text{A.9})$$

$$\mathbf{y}_u = K^{-1} (\alpha \mathbf{r}_u + Q \mathbf{y}_p), \quad (\text{A.10})$$

with $\mathbf{y}_q = \mathbf{r}_q$.

As already noticed in Section (3.2), the inner solves with K , A , \tilde{S}_K , and \tilde{S}_A can be conveniently replaced by inexact solves with the aid of inner preconditioners, say M_K^{-1} , M_A^{-1} , $M_{\tilde{S}_K}^{-1}$, and $M_{\tilde{S}_A}^{-1}$, respectively. The equivalent
445 ERPF2 application obtained by using the factorizations above is finally summarized in Algorithm 5.

References

- [1] M. A. Biot, General theory of three-dimensional consolidation, *J. Appl. Phys.* 12 (1941) 155–164. doi:10.1063/1.1712886.
- [2] O. Coussy, *Poromechanics*, Wiley, Chichester, UK, 2004. doi:10.1002/0470092718.
- [3] A. J. H. Frijns, A four-component mixture theory applied to cartilaginous tissues: numerical modelling and experiments, PhD thesis, Technische Universiteit Eindhoven, The Netherlands (2000). doi:10.6100/IR537990.
- 450 [4] P. J. Phillips, M. F. Wheeler, A coupling of mixed and continuous Galerkin finite element methods for poroelasticity I: the continuous in time case, *Comput. Geosci.* 11 (2) (2007a) 131–144. doi:10.1007/s10596-007-9045-y.
- [5] P. J. Phillips, M. F. Wheeler, A coupling of mixed and continuous Galerkin finite element methods for poroelasticity II: the discrete-in-time case, *Comput. Geosci.* 11 (2) (2007b) 145–158. doi:10.1007/s10596-007-9044-z.
- 455 [6] M. Bause, F. A. Radu, U. Köcher, Space–time finite element approximation of the Biot poroelasticity system with iterative coupling, *Comput. Meth. Appl. Mech. Eng.* 320 (2017) 745–768. doi:10.1016/j.cma.2017.03.017.
- [7] X. Hu, C. Rodrigo, F. J. Gaspar, L. T. Zikatanov, A nonconforming finite element method for the Biot’s consolidation model in poroelasticity, *J. Comput. Appl. Math.* 310 (2017) 143–154. doi:10.1016/j.cam.2016.06.003.
- [8] J. W. Both, K. Kumar, J. M. Nordbotten, F. A. Radu, Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media, *Comput. Math. Appl.* 77 (6) (2019) 1479–1502. doi:10.1016/j.camwa.2018.07.033.
- 460 [9] V. Girault, K. Kumar, M. F. Wheeler, Convergence of iterative coupling of geomechanics with flow in a fractured poroelastic medium, *Comput. Geosci.* 20 (5) (2016) 997–1011. doi:10.1007/s10596-016-9573-4.
- [10] Girault, Vivette, Wheeler, Mary F., Almani, Tameem, Dana, Saunik, A priori error estimates for a discretized poro-elastic-elastic system solved by a fixed-stress algorithm, *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles* 74 (2019) 24. doi:10.2516/ogst/2018071.
- 465 [11] M. Ferronato, N. Castelletto, G. Gambolati, A fully coupled 3-D mixed finite element model of Biot consolidation, *J. Comput. Phys.* 229 (12) (2010) 4813–4830. doi:10.1016/j.jcp.2010.03.018.
- [12] N. Castelletto, J. A. White, M. Ferronato, Scalable algorithms for three-field mixed finite element coupled poromechanics, *J. Comput. Phys.* 327 (2016) 894–918. doi:10.1016/j.jcp.2016.09.063.
- [13] R. W. Lewis, B. A. Schrefler, *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*, 2nd Edition, Wiley, Chichester, UK, 1998.
- 470 [14] N. Castelletto, M. Ferronato, G. Gambolati, Thermo-hydro-mechanical modeling of fluid geological storage by Godunov-mixed methods, *Int. J. Numer. Meth. Eng.* 90 (8) (2012) 988–1009. doi:10.1002/nme.3352.
- [15] B. Jha, R. Juanes, Coupled multiphase flow and poromechanics: A computational model of pore pressure effects on fault slip and earthquake triggering, *Water Resour. Res.* 5 (2014) 3776–3808. doi:10.1002/2013WR015175.
- 475 [16] J. Choo, S. Lee, Enriched galerkin finite elements for coupled poromechanics with local mass conservation, *Computer Methods in Applied Mechanics and Engineering* 341 (2018) 311 – 332. doi:https://doi.org/10.1016/j.cma.2018.06.022.
- [17] J. Choo, Large deformation poromechanics with local mass conservation: An enriched galerkin finite element framework, *International Journal for Numerical Methods in Engineering* 116 (1) (2018) 66–90. doi:10.1002/nme.5915.
- [18] K. Lipnikov, *Numerical Methods for the Biot Model in Poroelasticity*, PhD thesis, University of Houston (2002).

- 480 [19] C. Rodrigo, X. Hu, P. Ohm, J. H. Adler, F. J. Gaspar, L. T. Zikatanov, New stabilized discretizations for poroelasticity and the Stokes' equations, *Comput. Meth. Appl. Mech. Eng.* 341 (2018) 467–484. doi:10.1016/j.cma.2018.07.003.
- [20] C. Niu, H. Rui, X. Hu, A stabilized hybrid mixed finite element method for poroelasticity, *Comput. Geosci.* 25 (2021) 757–774. doi:10.1007/s10596-020-09972-3.
- [21] H. Honorio, C. Maliska, M. Ferronato, C. Janna, A stabilized element-based finite volume method for poroelastic problems, *J. Comput. Phys.* 364 (2018) 49–72. doi:10.1016/j.jcp.2018.03.010.
- 485 [22] J. T. Camargo, J. A. White, R. I. Borja, A macroelement stabilization for multiphase poromechanics, *Comput. Geosci.* 25 (2021) 775–792. doi:10.1007/s10596-020-09964-3.
- [23] M. Frigo, N. Castelletto, M. Ferronato, J. A. White, Efficient solvers for hybridized three-field mixed finite element coupled poromechanics, *Computers & Mathematics with Applications* 91 (2021) 36–52. doi:https://doi.org/10.1016/j.camwa.2020.07.010.
- 490 [24] H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.* 13 (2) (1992) 631–644. doi:10.1137/0913035.
- [25] Y. Saad, M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 856–869. doi:10.1137/0907058.
- [26] B. Jha, R. Juanes, A locally conservative finite element framework for the simulation of coupled flow and reservoir geomechanics, *Acta Geotech.* 2 (3) (2007) 139–153. doi:10.1007/s11440-007-0033-0.
- 495 [27] T. Almani, K. Kumar, A. Dogru, G. Singh, M. Wheeler, Convergence analysis of multirate fixed-stress split iterative schemes for coupling flow with geomechanics, *Comput. Methods Appl. Mech. Eng.* 311 (2016) 180–207. doi:10.1016/j.cma.2016.07.036.
- [28] J. W. Both, M. Borregales, J. M. Nordbotten, K. Kumar, F. A. Radu, Robust fixed stress splitting for Biot's equations in heterogeneous media, *Appl. Math. Lett.* 68 (2017) 101–108. doi:10.1016/j.aml.2016.12.019.
- 500 [29] M. Borregales, F. A. Radu, K. Kumar, J. M. Nordbotten, Robust iterative schemes for non-linear poromechanics, *Comput. Geosci.* 22 (4) (2018) 1021–1038. doi:10.1007/s10596-018-9736-6.
- [30] S. Dana, B. Ganis, M. F. Wheeler, A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs, *J. Comput. Phys.* 352 (2018) 1–22. doi:10.1016/j.jcp.2017.09.049.
- [31] S. Dana, M. F. Wheeler, Convergence analysis of two-grid fixed stress split iterative scheme for coupled flow and deformation in heterogeneous poroelastic media, *Comput. Methods Appl. Mech. Eng.* 341 (2018) 788–806. doi:10.1016/j.cma.2018.07.018.
- 505 [32] Q. Hong, J. Kraus, M. Lymbery, M. F. Wheeler, Parameter-robust convergence analysis of fixed-stress split iterative method for multiple-permeability poroelasticity systems, *Multiscale Model. Sim.* 18 (2) (2020) 916–941. doi:10.1137/19M1253988.
- [33] Y. Kuznetsov, K. Lipnikov, S. Lyons, S. Maliassov, Mathematical modeling and numerical algorithms for poroelastic problems, in: Chen, Z. and Glowinski, R. and Li, K. (Ed.), *Current Trends in Scientific Computing*, Vol. 329 of *Contemporary Mathematics*, Amer. Mathematical Soc., P.O. Box 6248, Providence, RI 02940 USA, 2003, pp. 191–202. doi:10.1090/conm/329.
- 510 [34] J. A. White, R. Borja, Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics, *Comput. Geosci.* 15 (4) (2011) 647–659. doi:10.1007/s10596-011-9233-7.
- [35] J. B. Haga, H. Osnes, H. P. Langtangen, A parallel block preconditioner for large-scale poroelasticity with highly heterogeneous material parameters, *Comput. Geosci.* 16 (3) (2012) 723–734. doi:10.1007/s10596-012-9284-4.
- 515 [36] J. B. Haga, H. Osnes, H. P. Langtangen, On the causes of pressure oscillations in low-permeable and low-compressible porous media, *Int. J. Numer. Anal. Methods Geomech.* 36 (12) (2012) 1507–1522. doi:10.1002/nag.1062.
- [37] O. Axelsson, R. Blaheta, P. Byczanski, Stable discretization of poroelasticity problems and efficient preconditioners for arising saddle point type matrices, *Comput. Vis. Sci.* 15 (4) (2012) 191–207. doi:10.1007/s00791-013-0209-0.
- [38] E. Turan, P. Arbenz, Large scale micro finite element analysis of 3D bone poroelasticity, *Parallel Comput.* 40 (7) (2014) 239–250. doi:10.1016/j.parco.2013.09.002.
- 520 [39] F. J. Gaspar, C. Rodrigo, On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics, *Comput. Meth. Appl. Mech. Eng.* 326 (2017) 526–540. doi:10.1016/j.cma.2017.08.025.
- [40] P. Luo, C. Rodrigo, F. J. Gaspar, C. W. Oosterlee, On an Uzawa smoother in multigrid for poroelasticity equations, *Numer. Linear Algebr. Appl.* 24 (1) (2017) e2074. doi:10.1002/nla.2074}.
- 525 [41] J. J. Lee, K.-A. Mardal, R. Winther, Parameter-Robust Discretization and Preconditioning of Biot's Consolidation Model, *SIAM J. Sci. Comput.* 39 (1) (2010) A1–A24. doi:10.1137/15M1029473.
- [42] F. J. Adler, J. H. Gaspar, X. Hu, C. Rodrigo, L. T. Zikatanov, Robust Block Preconditioners for Biot's Model, in: P. E. Bjøstad, S. Brenner, L. Halpern, R. Kornhuber, H. H. Kim, T. Rahman, O. B. Widlund (Eds.), *Domain Decomposition Methods in Science and Engineering XXIV*, Springer International Publishing, 2018, pp. 3–16. doi:10.1007/978-3-319-93873-8.
- 530 [43] Q. Hong, J. Kraus, Parameter-robust stability of classical three-field formulation of Biot's consolidation model, *Electron. Trans. Numer. Anal.* 48 (2018) 202–226. doi:10.1553/etna_vol48s202.
- [44] M. Ferronato, A. Franceschini, C. Janna, N. Castelletto, H. Tchelepi, A general preconditioning framework for coupled multi-physics problems, *J. Comput. Phys.* 398 (2019) 108887. doi:https://doi.org/10.1016/j.jcp.2019.108887.
- [45] J. H. Adler, F. J. Gaspar, X. Hu, P. Ohm, C. Rodrigo, L. T. Zikatanov, Robust preconditioners for a new stabilized discretization of the poroelastic equations, *SIAM J. Sci. Comput.* 42 (3) (2020) B761–B791. doi:10.1137/19M1261250.
- 535 [46] Q. M. Bui, D. Osei-Kuffuor, N. Castelletto, J. A. White, A scalable multigrid reduction framework for multiphase poromechanics of heterogeneous media, *SIAM Journal on Scientific Computing* 42 (2) (2020) B379–B396. doi:10.1137/19M1256117.
- [47] M. Frigo, N. Castelletto, M. Ferronato, A Relaxed Physical Factorization Preconditioner for mixed finite element coupled poromechanics, *SIAM J. Sci. Comput.* 41 (2019) B694–B720. doi:10.1137/18M120645X.
- 540 [48] M. Benzi, N. M., Q. Niu, Z. Wang, A Relaxed Dimensional Factorization preconditioner for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 230 (16) (2011) 6185–6202. doi:10.1016/j.jcp.2011.04.001.
- [49] M. Benzi, S. Deparis, G. Grandperrin, A. Quarteroni, Parameter estimates for the Relaxed Dimensional Factorization preconditioner and application to hemodynamics, *Comput. Meth. Appl. Mech. Eng.* 300 (Supplement C) (2016) 129–145. doi:10.1016/j.cma.2015.11.016.
- [50] U. Brink, Stein, On some mixed finite element methods for incompressible and nearly incompressible finite elasticity, *Computational Me-*

- chanics 19 (1996) 1432–0924. [doi:10.1007/BF02824849](https://doi.org/10.1007/BF02824849).
- [51] M. Olshanskii, M. Benzi, An augmented Lagrangian approach to linearized problems in hydrodynamic stability, *SIAM J. Sci. Comp.* 30 (2007) 1459–1473. [doi:10.1137/070691851](https://doi.org/10.1137/070691851).
- [52] M. Benzi, M. Olshanskii, Field-of-values convergence analysis of Augmented Lagrangian preconditioners for the linearize Navier-Stokes problem, *SIAM J. Numer. Anal.* 49 (2011) 770–788. [doi:10.1137/100806485](https://doi.org/10.1137/100806485).
- [53] M. Benzi, M. Olshanskii, Z. Wang, Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations, *Int. J. Numer. Meth. Fluids* 66 (2011) 486–508. [doi:10.1002/flid.2267](https://doi.org/10.1002/flid.2267).
- [54] P. E. Farrell, L. Mitchell, F. Wechsung, An augmented Lagrangian preconditioner for the 3d stationary incompressible Navier-Stokes equations at high Reynolds number, *SIAM J. Sci. Comput.* 41 (5) (2019) A3073–A3096. [doi:10.1137/18M1219370](https://doi.org/10.1137/18M1219370).
- [55] N. Castelletto, J. White, H. Tchelepi, Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics, *International Journal for Numerical and Analytical Methods in Geomechanics* 39 (2015) 1593–1618. [doi:10.1002/nag.2400](https://doi.org/10.1002/nag.2400).
- [56] J. A. White, N. Castelletto, H. A. Tchelepi, Block-partitioned solvers for coupled poromechanics: A unified framework, *Comput. Meth. Appl. Mech. Eng.* 303 (2016) 55–74. [doi:10.1016/j.cma.2016.01.008](https://doi.org/10.1016/j.cma.2016.01.008).
- [57] L. Bergamaschi, S. Mantica, G. Manzini, A mixed finite element-finite volume formulation of the black oil model, *SIAM J. Sci. Comput.* 20 (1998) 970–997. [doi:10.1137/S1064827595289303](https://doi.org/10.1137/S1064827595289303).
- [58] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, 2nd Edition, Springer, Cham, Switzerland, 2016. [doi:10.1007/978-3-319-28483-5](https://doi.org/10.1007/978-3-319-28483-5).
- [59] J. Mandel, Consolidation des sols (Étude mathématique), *Geotechnique* 3 (7) (1953) 287–299. [doi:10.1680/geot.1953.3.7.287](https://doi.org/10.1680/geot.1953.3.7.287).
- [60] Y. Saad, ILUT: A dual threshold incomplete ILU factorization, *Numer. Lin. Alg. Appl.* 1 (1994) 387–402. [doi:10.1002/nla.1680010405](https://doi.org/10.1002/nla.1680010405).
- [61] C. Lin, J. Moré, Incomplete Cholesky factorizations with limited memory, *SIAM J. Sci. Comput.* 21 (1) (1999) 24–45. [doi:10.1137/S1064827597327334](https://doi.org/10.1137/S1064827597327334).
- [62] J. Q. Ruge, K. Stüben, Algebraic Multigrid, Vol. 3 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1987, Ch. 4, pp. 73–130. [doi:10.1137/1.9781611971057.ch4](https://doi.org/10.1137/1.9781611971057.ch4).
- [63] J. Boyle, M. D. Mihajlovic, J. A. Scott, HSL_MI20: An efficient AMG preconditioner for finite element problems in 3D, *Int. J. Numer. Meth. Eng.* 82 (2010) 64–98. [doi:10.1002/nme.2758](https://doi.org/10.1002/nme.2758).
- [64] M. Ferronato, L. Gazzola, N. Castelletto, P. Teatini, L. Zhu, A coupled Mixed Finite Element Biot model for land subsidence prediction in the Beijing area, in: M. Vandamme, P. Dangla, J. Pereira, S. Ghabezloo (Eds.), *Poromechanics VI*, American Society of Civil Engineers, 2017, pp. 182–189. [doi:10.1061/9780784480779.022](https://doi.org/10.1061/9780784480779.022).
- [65] L. Zhu, Z. Dai, H. Gong, C. Gable, P. Teatini, Statistic inversion of multi-zone transition probability models for aquifer characterization in alluvial fans, *Stoch. Environ. Res. Risk Assess.* 30 (2016) 1005–1016. [doi:10.1007/s00477-015-1089-2](https://doi.org/10.1007/s00477-015-1089-2).
- [66] L. Zhu, A. Franceschini, H. Gong, M. Ferronato, Z. Dai, Y. Ke, Y. Pan, X. Li, R. Wang, P. Teatini, The 3-d facies and geomechanical modeling of land subsidence in the chaobai plain, beijing, *Water Resour. Res.* 56 (3) (2020) e2019WR027026. [doi:10.1029/2019WR027026](https://doi.org/10.1029/2019WR027026).
- [67] M. Christie, M. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques, *SPE Reserv. Eval. Eng.* 4 (4) (2001) 308–316. [doi:10.2118/72469-PA](https://doi.org/10.2118/72469-PA).
- [68] Z. Chen, G. Huan, Y. Ma, *Computational Methods for Multiphase Flows in Porous*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006. [doi:10.1137/1.9780898718942](https://doi.org/10.1137/1.9780898718942).