Journal of Computational Physics 442 (2021) 110513

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A novel block non-symmetric preconditioner for mixed-hybrid finite-element-based Darcy flow simulations

Stefano Nardean [a], Massimiliano Ferronato [b], Ahmad S. Abushaikha [a,*]

[a] *Division of Sustainable Development, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar*
[b] *Department of Civil, Environmental and Architectural Engineering, University of Padova, Padova, Italy*

ARTICLE INFO

Article history:
Available online 10 June 2021

Keywords:
Flow in porous media
Preconditioning
Block matrices

ABSTRACT

In this work, we propose a novel block preconditioner, labeled Explicit Decoupling Factor Approximation (EDFA), to accelerate the convergence of Krylov subspace solvers used to address the sequence of non-symmetric systems of linear equations originating from flow simulations in porous media. The flow model is discretized blending the Mixed hybrid finite element method for Darcy's equation with the Finite volume scheme for the mass conservation. The EDFA preconditioner is characterized by two features: the exploitation of the system matrix decoupling factors to recast the Schur complement and their inexact fully-parallel computation by means of restriction operators. We introduce two adaptive techniques aimed at building the restriction operators according to the properties of the system at hand. The proposed block preconditioner has been tested through extensive experimentation on both synthetic and real-case applications, pointing out its robustness and computational efficiency.

1. Introduction

Numerical modeling of fluid flow in porous media is a key requirement for a wide number of applications in subsurface hydrology and petroleum engineering. In general, computer simulators are fundamental tools for the proper management and exploitation of aquifer systems, as well as oil and gas fields. The growing demand for higher accuracy of the simulation, assisted by the increasing availability of computational and storage resources, leads to continuous development and refinement of virtual simulators. The degree of approximation of the overall numerical model is defined, first of all, by the underlying mathematical model, but also by the selected discretization scheme. These schemes should handle effectively non-$\mathbb{K}$-orthogonal unstructured grids, as well as highly heterogeneous and anisotropic rock/fluid properties, frequently introduced as full-tensors in the model (see, for instance, Reference [1] about the numerical issues, related to abrupt changes in permeability, in Node control volume finite element discretizations).

The Mixed Hybrid Finite Element (MHFE) method, and, in general, the whole class of Mixed Finite Element (MFE) methods [2], coupled with the Finite Volume (FV) method, has been gaining a growing popularity in recent years. The mass conservation at the element level, the continuity of normal fluxes across internal faces of the discretized domain, the accuracy of the velocity and pressure fields, the possibility of handling either structured or unstructured grids, and the elegant treatment of full tensor fluid properties [3–5] have made the MHFE method attractive for several applications, such as contaminant transport [6–8], energy storage [9], poromechanics [10–13] and, of course, single-phase [14], variably

---

* Corresponding author.
E-mail addresses: snardean@hbku.edu.qa (S. Nardean), massimiliano.ferronato@unipd.it (M. Ferronato), aabushaikha@hbku.edu.qa (A.S. Abushaikha).

https://doi.org/10.1016/j.jcp.2021.110513

saturated [15,16], multi-phase [17–21] and, more recently, compositional [22] flow problems. However, the MHFE method exhibits some critical aspects as well, such as the violation of the Discrete maximum principle [23] and the higher number of unknowns per element, as compared to other schemes like FV or classical Finite element methods. In this regard, much effort was devoted to trying to reduce the overall number of unknowns per cell to one per face [3,5] and even only one per element [24]. The main focus was on triangular cells in two-dimensional (2-D) applications, however, with some limitations in the shape of tetrahedra in three-dimensional (3-D) domains [25].

A MHFE-based simulator was recently developed in [20] to model the two-phase flow in heterogeneous porous media. Fučík et al. [19] introduced multi-component compositional flow in the previous model and also focused on the design of a parallel implementation on both CPU and GPU. A similar approach, but extended to compressible multi-phase flow, was developed in [21] and applied to several real-field applications. Puscas et al. [26] proposed a two-phase flow Multiscale MHFE (MMHFE) simulator, where much care was devoted to the design of a robust parallel implementation, while Devloo et al. [27] introduced the Discrete fracture model in a 2D MMHFE simulator. Abushaikha et al. [22] developed a fully implicit general-purpose MHFE-based simulator for highly heterogeneous reservoirs, which was later adapted to accommodate a Mimetic Finite Difference (MFD) formulation [28,29] for Darcy's equation in [30] and the Discrete fracture model in [31,32]. The modeling approach in [22], considered in this work as well, propounds a novel formulation of the mass balance equation, where the continuity of the fluxes across adjacent grid elements is strongly imposed. This allows for getting rid of possible flux oscillations during the nonlinear iterations, to the benefit of the nonlinear solver robustness and convergence rate. Such an approach, however, results in three challenging numerical properties: (i) a high number of unknowns per element (although the lowest-order Raviart-Thomas ($\mathbb{RT}_0$) space [33] is used), giving rise to large-size systems of equations, (ii) the non-symmetric nature of such systems, and (iii) the inherent block structure of the Jacobian matrix. A key issue to increase the attractiveness of the approach in [22] is the availability of an efficient linear solver to tackle the sequence of inner linear systems with the Jacobian matrix originating during a full-transient simulation. Most of the overall CPU time, in fact, is usually needed for this task. Given the size and sparsity degree of these systems, Krylov subspace methods [34] are normally the method of choice, but their performance needs to be boosted by means of appropriate preconditioning operators.

The main objective of this paper is the efficient solution of the systems of equations stemming from the aforementioned MHFE-FV modeling approach, by designing a specific preconditioning technique that copes with their non-symmetric nature. A popular physics-based preconditioner for reservoir simulations is the Constrained Pressure Residual (CPR) [35–37], which is the standard for commercial and also academic simulators [38–40]. CPR was designed with the aim at exploiting the properties of the different processes that are coupled together. CPR-type algorithms are multi-stage preconditioners (like SIMPLE [41,42]), whose application (to a vector) goes through several steps during which the groups of unknowns are repeatedly updated. CPR has two stages in its original formulation, but other multi-stage variants exist [43,44]. Recently, a two-stage CPR scheme, suitable for non-isothermal multi-phase flow simulations, namely Constrained Pressure-Temperature Residual (CPTR), has been designed by Roy et al. [45]. However, given the ill-conditioning and non-symmetric nature of the systems of equations originating from our modeling approach, CPR-like schemes are often ineffectual.

In this work, we focus on developing an algebraic preconditioner that takes advantage of the block structure of the resulting discrete problem. Preconditioning of large-size block systems of equations is a mature branch of applied mathematics [46–48] and still a very active research field. In recent years, block preconditioning techniques have been developed for different problems, such as the solution of the Navier-Stokes equations [49–53], coupled poromechanics [54–65], flow in fractured porous media [66–68], electromagnetism [69–72], contact mechanics [73–77], as well as single- and multi-phase flow in porous media [45,80,81], including multi-level methods [78,79,82–85] that rely on the block decomposition arising from hierarchical partitioning of the grid into coarse- and fine-scale levels. In particular, the issue of preconditioning in the framework of the MFE and MHFE discretization of flow problems in porous media is not new (see, for instance, [86,87]), but, in its original formulation, the resulting systems had the typical structure of symmetric saddle-point problems [47]. However, in the MHFE-FV formulation we utilize, the property of symmetry is lost. The main feature of our block preconditioner is twofold: (i) the exploitation of the block matrix decoupling factors to recast the Schur complement computation, and (ii) their approximate construction by means of appropriate restriction and prolongation operators. The reference model for the development of our preconditioner is the basic MHFE-FV discretized single-phase flow in porous media. This represents the first stage of a more comprehensive research project, where we plan to extend this preconditioning framework to a MHFE-FV multi-phase reservoir simulator based on the modeling approach described in [22].

The rest of the paper is organized as follows. The model problem is first presented along with the algebraic properties of the system matrix; then the block-structured preconditioning framework is introduced and tested in both synthetic and real-world applications. The experimental stage helped highlight advantages and drawbacks of the proposed preconditioner, which are reported in the discussion section. The conclusions and hints on the ongoing and future work finally close the paper.

2. MHFE-FV model of single-phase flow in porous media

The set of equations governing the single-phase flow in porous media consists of the *mass conservation* and *Darcy's law*. The monolithic solution approach addresses these equations simultaneously by means of a fully implicit coupling.

2.1. Governing equations

Consider the finite porous domain $\Omega \subset \mathbb{R}^3$, its boundary Γ and their union $\overline{\Omega} = \Omega \cup \Gamma$. Γ_p and $\Gamma_{\boldsymbol{v}}$ are partitions of Γ such that $\Gamma_p \cup \Gamma_{\boldsymbol{v}} = \Gamma$ and $\Gamma_p \cap \Gamma_{\boldsymbol{v}} = \varnothing$. Let t and $\mathbb{T} =]0, T[$ indicate the time variable and the simulated open temporal domain, respectively. Denoting with $s : \Omega \times \mathbb{T} \to \mathbb{R}$ the source or sink term, $p : \overline{\Omega} \times [0, T] \to \mathbb{R}$ the fluid pressure, $\boldsymbol{v} : \overline{\Omega} \times [0, T] \to \mathbb{R}^3$ the velocity vector and $c : \overline{\Omega} \to \mathbb{R}^+$ the specific storage coefficient, representative of both the fluid and porous matrix compressibilities, the set of governing PDEs reads:

$$\boldsymbol{v} = -\frac{K}{\gamma}\nabla p \qquad \text{on } \Omega \times \mathbb{T} \qquad \text{(Darcy's law)}, \tag{1a}$$

$$\nabla \cdot \boldsymbol{v} + c\dot{p} = s \qquad \text{on } \Omega \times \mathbb{T} \qquad \text{(mass conservation)}, \tag{1b}$$

where the symbol ∇ indicates the gradient operator, $\nabla \cdot$ the divergence operator and $\dot{()}$ the derivative with respect to time. In equation (1a), γ is the fluid specific weight and K is the conductivity tensor, assumed to be symmetric and positive definite (SPD). The gravitational term is here neglected. The specific storage coefficient c in equation (1b) can be expressed as $c = \alpha + \phi\beta$ where α is the soil compressibility, ϕ the medium porosity and β the fluid volumetric compressibility [88]. The solution to the system of equations (1a) and (1b) is a well-posed problem provided that a set of appropriate initial and boundary conditions is supplied:

$$p|_{t=0} = p_0 \qquad \text{in } \overline{\Omega} \qquad \text{(initial fluid pressure)}, \tag{2a}$$

$$p = \overline{p} \qquad \text{on } \Gamma_p \times \mathbb{T} \qquad \text{(prescribed fluid pressure)}, \tag{2b}$$

$$-\frac{K}{\gamma}\nabla p \cdot \boldsymbol{n} = \overline{v_n} \qquad \text{on } \Gamma_{\boldsymbol{v}} \times \mathbb{T} \qquad \text{(prescribed Darcy's flux)}, \tag{2c}$$

for assigned functions $p_0 : \overline{\Omega} \to \mathbb{R}$, $\overline{p} : \Gamma_p \times \mathbb{T} \to \mathbb{R}$, and $\overline{v_n} : \Gamma_{\boldsymbol{v}} \times \mathbb{T} \to \mathbb{R}$. In equation (2c), $\boldsymbol{n}$ denotes the outer unit normal vector to $\Gamma_{\boldsymbol{v}}$.

2.2. Discretization of the governing equations

The model domain is partitioned into non-overlapping hexahedral elements, which accommodate, as shown in Fig. 1b, two types of pressure unknowns, located on each face barycenter, π, and on the element centroid, p^E. The former acts the part of Lagrange multipliers and expresses the face average pressure, whereas the latter represents the average element value.

Let $\mathcal{E}^h$ and $\mathcal{F}^h$ be the collection of elements and faces of the discretized domain, respectively. In our modeling approach, equation (1a) is discretized by means of the MHFE method, using the $\mathbb{RT}_0$ space to approximate the velocity $\boldsymbol{v}^h$ and the $\mathbb{P}_0$ space for the pressure p^h and Lagrange multipliers π^h:

$$\mathcal{V}^h = \left\{\boldsymbol{v}^h \mid \boldsymbol{v}^h \in H(\text{div}, \mathcal{E}^h),\ -\boldsymbol{v}^h \cdot \boldsymbol{n} = \overline{v_n} \text{ on } \Gamma_{\boldsymbol{v}},\ \boldsymbol{v}^h|_E \in \mathbb{RT}_0(E),\ \forall E \in \mathcal{E}^h\right\}, \tag{3a}$$

$$\mathcal{L}^h = \left\{p^h \mid p^h \in L^2(\mathcal{E}^h),\ p^h|_E \in \mathbb{P}_0(E),\ \forall E \in \mathcal{E}^h\right\}, \tag{3b}$$

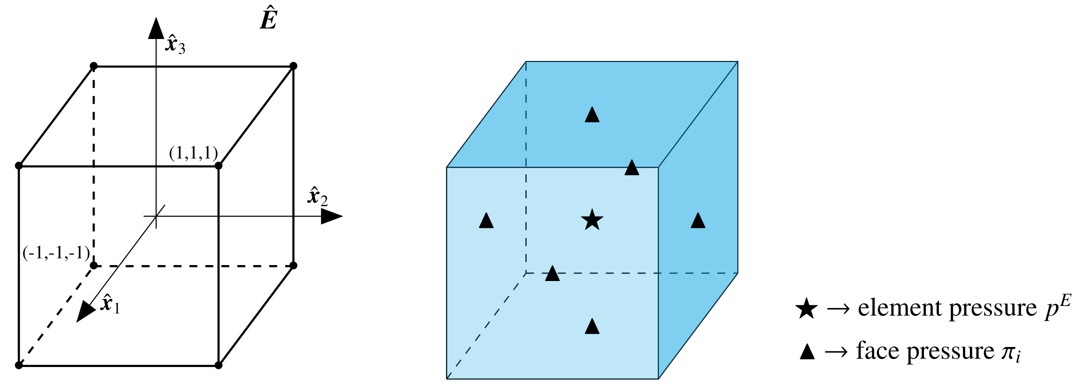
$$\mathcal{M}^h = \left\{\pi^h \mid \pi^h \in L^2(\mathcal{F}^h),\ \pi^h = \overline{p} \text{ on } \Gamma_p,\ \pi^h|_f \in \mathbb{P}_0(f),\ \forall f \in \mathcal{F}^h\right\}, \tag{3c}$$

where $L^2\left(\mathcal{E}^h\right)$ and $L^2\left(\mathcal{F}^h\right)$ denote the spaces of square Lebesgue-integrable functions on $\mathcal{E}^h$ and $\mathcal{F}^h$, respectively, and $H(\text{div}, \mathcal{E}^h) = \left\{\phi \in L^2(\Omega), \nabla \cdot \phi|_E \in L^2(E), \forall E \in \mathcal{E}^h\right\}$ is the Sobolev space of square integrable vector functions with square integrable divergence in E [2,90].

The $\mathcal{V}^h$ trial space for 3-D problems is generated by local piecewise trilinear vector functions, $\boldsymbol{\eta}_i^E(x_1, x_2, x_3)$, defined for each face i of element E [5,89]. Considering the hexahedral reference element, $\hat{E}$, in Fig. 1a, the $\mathbb{RT}_0$ basis functions simply read:

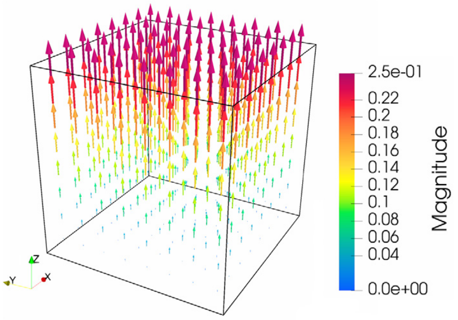
$$\boldsymbol{\eta}^{\hat{E}}_{\pm\hat{x}_a}\left(\hat{x}_1, \hat{x}_2, \hat{x}_3\right) = \frac{1}{8}\left(\hat{x}_a \pm 1\right)\begin{pmatrix}\frac{\partial \hat{x}_1}{\partial \hat{x}_a}\\ \frac{\partial \hat{x}_2}{\partial \hat{x}_a}\\ \frac{\partial \hat{x}_3}{\partial \hat{x}_a}\end{pmatrix}, \quad \text{with } a = 1, 2, 3, \tag{4}$$

where $\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_3$ denote the three coordinates in the reference space. One of these functions is shown in Fig. 1c. Such functions exhibit two basic properties [4]:



(a) The reference hexahedron (modified after [89]).

(b) Location of the units in the hexahedral reference element.



(c) $\mathbb{RT}_0$ basis function of the top face of the hexahedral reference element.

Fig. 1. The hexahedral element in the reference space with distribution of the model unknowns and example of $\mathbb{RT}_0$ basis function. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

1. The flux of $\boldsymbol{\eta}_i^E$ is unitary across face i and null elsewhere:

$$\int_{A_j} \boldsymbol{\eta}_i^E \cdot \boldsymbol{n}_j \, dA = \delta_{ij}, \quad i, j = 1, \ldots, N_f^E, \tag{5}$$

   where $\boldsymbol{n}_j$ denotes the outer normal at face j, A_j the relevant area, δ_{ij} the Kronecker delta and N_f^E the number of faces of element E. From equation (5) it follows that $\boldsymbol{\eta}_i^E$ has a continuous normal component at face i, so the normal fluxes are also continuous.
2. The integral of $\boldsymbol{\eta}_i^E$ divergence is unitary over element E:

$$\int_{\Omega^E} \nabla \cdot \boldsymbol{\eta}_i^E \, d\Omega = 1, \quad i = 1, \ldots, N_f^E, \tag{6}$$

   where Ω^E is the element volume.

Darcy's velocity $\boldsymbol{v}$ is approximated at the element level by a linear combination of the basis functions $\boldsymbol{\eta}_i^E$ [3]:

$$\boldsymbol{v}^{h,E} = \sum_{j=1}^{N_f^E} q_j^E \boldsymbol{\eta}_j^E(x_1, x_2, x_3), \tag{7}$$

where q_j^E represents the flux across face j.

Let $\{\xi^E\}_{E \in \mathcal{E}^h}$ be the set of basis functions for $\mathcal{L}^h$ such that $\xi^E(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in E$ and $\xi^E(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \notin E$. Similarly, the basis for $\mathcal{M}^h$, $\{\zeta_f\}_{f \in \mathcal{F}^h}$, consists of functions for which $\zeta_f(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in f$ and $\zeta_f(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \notin f$. Therefore, the pressure and Lagrange multiplier fields read:

$$p^h = \sum_{E=1}^{N_e} \xi^E p^E \qquad \text{and} \qquad \pi^h = \sum_{f=1}^{N_f} \zeta_f \pi_f, \tag{8}$$

where N_e and N_f are the number of elements and faces in the grid, respectively. The Galerkin weak form of equation (1a) is written element-by-element as:

$$\gamma \int_{\Omega^E} \boldsymbol{\eta}_i^{E,T} K^{E^{-1}} \boldsymbol{v}^E \, d\Omega = - \int_{\Omega^E} \boldsymbol{\eta}_i^{E,T} \nabla p \, d\Omega, \qquad i = 1, \dots, N_f^E. \tag{9}$$

Applying the Green-Gauss lemma to the Right-Hand Side (RHS) of equation (9) and substituting equations (5), (6) and (8) entails:

$$\begin{aligned} - \int_{\Omega^E} \boldsymbol{\eta}_i^{E,T} \nabla p \, d\Omega &= \int_{\Omega^E} \nabla \cdot \boldsymbol{\eta}_i^E \, p \, d\Omega - \sum_{j=1}^{N_f^E} \int_{A_j} \boldsymbol{\eta}_i^E \cdot \boldsymbol{n}_j \, p \, dA \\ &= p^E - \pi_i^E, \qquad i = 1, \dots, N_f^E, \end{aligned} \tag{10}$$

where the superscript on π_i^E indicates that those unknowns belong to element E. Introducing equations (7) and (10) in (9) yields:

$$\gamma \sum_{j=1}^{N_f^E} \int_{\Omega^E} \boldsymbol{\eta}_i^{E,T} K^{E^{-1}} \boldsymbol{\eta}_j^E \, d\Omega \, q_j^E = p^E - \pi_i^E, \qquad i = 1, \dots, N_f^E. \tag{11}$$

Defining the elementary matrices $B^E \in \mathbb{R}^{N_f^E \times N_f^E}$, whose components are [3]:

$$B_{ij}^E = \gamma \int_{\Omega^E} \boldsymbol{\eta}_i^{E,T} K^{E^{-1}} \boldsymbol{\eta}_j^E \, d\Omega, \quad i, j = 1, \dots, N_f^E, \tag{12}$$

the local final expression for equation (11) reads:

$$\boldsymbol{q}^E = B^{E^{-1}} (p^E \mathbf{1} - \boldsymbol{\pi}^E), \tag{13}$$

which allows to link the face fluxes with the local pressure differences, being $\boldsymbol{q}^E$ and $\boldsymbol{\pi}^E$ the vectors gathering the interface fluxes and pressures of element E and $\mathbf{1} \in \mathbb{R}^{N_f^E}$ the vector of unitary components. Being K^E SPD, B^E is so as well. The numerical evaluation of the integrals in equation (12) may be troublesome when performed in the model space with general elements. In this regard, Piola transformation comes into play, allowing to map the element in the model space into the prototype hexahedron in the reference space (Fig. 1a), perform the integrals using functions $\boldsymbol{\eta}_{\pm \hat{x}_a}^{\hat{E}}$ in equation (4) and then map the result back to the physical space (see, for instance, [89,91,92]).

For the discretization of the mass balance equation (1b), we use a FV approximation in space. Choosing the elements of the grid as *control volumes*, we have:

$$\int_{\Omega^E} c\dot{p} \, d\Omega + \int_{\Omega^E} \nabla \cdot \boldsymbol{v} \, d\Omega = \int_{\Omega^E} s \, d\Omega, \quad E = 1, \dots, N_e. \tag{14}$$

Recognizing that the second term on the Left-Hand Side (LHS) is equivalent to the sum of the fluxes across the faces of the element, equation (14) gives:

$$\Omega^E c^E \frac{p_{n+1}^E - p_n^E}{\Delta t_n} + \sum_{i=1}^{N_f^E} q_i^{E,E'} = \Omega^E s^E, \quad E = 1, \dots, N_e, \tag{15}$$

where a first-order backward Finite difference scheme has been introduced for the integration in time. In equation (15), the subscript n indicates the previous time step, $n+1$ the actual one, $\Delta t_n = t_{n+1} - t_n$, c^E and s^E are the mean values of the storage coefficient and source terms in E, and $q_i^{E,E'}$ is the fluid flux exchanged by the adjacent elements E and E' across face i. The expression for the inter-element flux $q_i^{E,E'}$ results from strongly imposing the continuity of local fluxes across face i (see appendix A in [22] for details):

$$q_i^{E,E'} = \frac{B_{ii}^{E'^{-1}} \Lambda^E - B_{ii}^{E^{-1}} \Lambda^{E'}}{B_{ii}^E + B_{ii}^{E'}}, \tag{16}$$

where

$$\Lambda^E = L_{B_i^E} p^E - \sum_{j=1}^{N_f^E} B_{ij}^{E^{-1}} \pi_j^E \qquad \text{with } i \neq j, \qquad L_{B_i^E} = \sum_{j=1}^{N_f^E} B_{ij}^{E^{-1}}.$$

Notice that the main consequences of this formulation are a tightened tying of the local fluxes and the enlargement of the native stencil, since $q_i^{E,E'}$ depends not only on the pressure unknowns of E but also on E''s.

2.3. The MHFE-FV system of equations

The solution to the model problem is achieved by solving the system of equations (15) at each time step, along with the strong enforcement of the fluxes continuity across the faces of the grid:

$$q_i^E + q_i^{E'} = 0, \qquad i = 1, \ldots, N_f, \tag{17}$$

where q_i^E and $q_i^{E'}$ denote the fluxes across face i for elements E and E', respectively. Along the boundary, equation (17) allows also to apply Neumann conditions in a strong form, just by substituting the RHS accordingly and dropping $q_i^{E'}$. Notice that equation (17) uses the fluxes as expressed in (13), unlike equation (15). Finally, it is implicitly assumed that $\pi_i^E = \pi_i^{E'}$ due to continuity reasons.

The resulting system

$$\mathcal{A}\boldsymbol{u} = \boldsymbol{f} \quad \Rightarrow \quad \begin{bmatrix} A_{\pi\pi} & A_{\pi p} \\ A_{p\pi} & A_{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}^{n+1} \\ \boldsymbol{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_\pi \\ \boldsymbol{f}_p \end{bmatrix} \tag{18}$$

exhibits a 2 × 2 block structure and it is solved in a fully implicit framework. In equation (18), $A_{\pi\pi} \in \mathbb{R}^{N_f \times N_f}$, $A_{pp} \in \mathbb{R}^{N_e \times N_e}$ (with $N_f > N_e$), $\boldsymbol{\pi}^{n+1}$ and $\boldsymbol{p}^{n+1}$ gather the face and element pressure unknowns, and $\boldsymbol{f}_\pi$ and $\boldsymbol{f}_p$ are the relevant components of the known term. The Lagrange multipliers and the element pressure unknowns are coupled by means of the rectangular blocks $A_{\pi p}$ and $A_{p\pi}$. As to the properties of $\mathcal{A}$, this matrix has a flipped generalized saddle-point structure, it is sparse, non-symmetric and usually ill-conditioned. In particular, A_{pp} has a symmetric structure, though it is not symmetric, $A_{\pi\pi}$ is a symmetric negative definite matrix, and $A_{\pi p} \neq \pm A_{p\pi}^T$. As mentioned before, in the context of single-phase flow, system (18) is *linear*.

3. The Explicit Decoupling Factor Approximation preconditioner

Solving accurately and efficiently the sequence of linear systems (18) arising from a MHFE-FV unsteady flow simulation is the major purpose of this study. Iterative Krylov subspace solvers are mandatory to address the large-size and sparse systems of equations that stem from real-world 3-D models, especially for the low memory requirements and better scalability as compared to direct solvers [34]. When the system matrix is non-symmetric, the Bi-Conjugate Gradient Stabilized (Bi-CGStab) [93] or the Generalized Minimal Residual (GMRES) [94] methods are usually the selected algorithms. However, improving their performance by supplying an appropriate preconditioning operator $\mathcal{P}^{-1}$ is key to guarantee a fast and smooth convergence.

It is well-known that an *effective* preconditioner is an operator whose application to a vector should resemble as much as possible that of the inverse, $\mathcal{A}^{-1}$, of the system matrix [46,48]. Therefore, a good starting point for the design of our preconditioner is to consider $\mathcal{A}^{-1}$ and take advantage of its block structure, as it is usually done in saddle-point and general block problems [47,77,95,96]. The block $\mathcal{LDU}$ decomposition of the system matrix reads:

$$\mathcal{A} = \begin{bmatrix} I_\pi & \\ A_{p\pi} A_{\pi\pi}^{-1} & I_p \end{bmatrix} \begin{bmatrix} A_{\pi\pi} & \\ & S \end{bmatrix} \begin{bmatrix} I_\pi & A_{\pi\pi}^{-1} A_{\pi p} \\ & I_p \end{bmatrix}, \tag{19}$$

where I_π and I_p are the identity in $\mathbb{R}^{N_f \times N_f}$ and $\mathbb{R}^{N_e \times N_e}$, respectively, and $S = A_{pp} - A_{p\pi} A_{\pi\pi}^{-1} A_{\pi p}$ is the so-called Schur complement. The exact inverse of $\mathcal{A}$ in a factorized form reads:

$$\mathcal{A}^{-1} = \begin{bmatrix} I_\pi & -A_{\pi\pi}^{-1} A_{\pi p} \\ & I_p \end{bmatrix} \begin{bmatrix} A_{\pi\pi}^{-1} & \\ & S^{-1} \end{bmatrix} \begin{bmatrix} I_\pi & \\ -A_{p\pi} A_{\pi\pi}^{-1} & I_p \end{bmatrix}, \tag{20}$$

where the two decoupling factors are defined as:

$$G = -A_{p\pi} A_{\pi\pi}^{-1} \quad \text{and} \quad F = -A_{\pi\pi}^{-1} A_{\pi p}. \tag{21}$$

The decoupling factors F and G are also used to compute the Schur complement as:

$$\begin{aligned} S &= A_{pp} - A_{p\pi} A_{\pi\pi}^{-1} A_{\pi p} \\ &= A_{pp} - A_{p\pi} A_{\pi\pi}^{-1} A_{\pi\pi} A_{\pi\pi}^{-1} A_{\pi p} \\ &= A_{pp} - H, \end{aligned} \tag{22}$$

with $H = G A_{\pi\pi} F$.

Considering equations (21), F and G can be computed explicitly by solving two independent sets of Multiple Right-Hand Side (MRHS) systems:

$$A_{\pi\pi}^T G^T = -A_{p\pi}^T, \tag{23a}$$

$$A_{\pi\pi} F = -A_{\pi p}. \tag{23b}$$

Of course, such an operation cannot be performed exactly because F and G are dense, hence proper approximations have to be introduced. The key feature of the proposed approach, denoted as Explicit Decoupling Factor Approximation (EDFA), is the computation of sparse explicit approximations for F and G, $\widetilde{F}$ and $\widetilde{G}$, respectively, by means of proper *restriction operators*. The approximate decoupling factors $\widetilde{F}$ and $\widetilde{G}$ are used to compute a sparsified Schur complement $\widetilde{S}$:

$$\begin{aligned} \widetilde{S} &= A_{pp} - \widetilde{G} A_{\pi\pi} \widetilde{F} \\ &= A_{pp} - \widetilde{H}. \end{aligned} \tag{24}$$

Recalling equation (20), the final algebraic expression of the EDFA preconditioner reads:

$$\mathcal{P}^{-1} = \begin{bmatrix} I_\pi & -\widetilde{A}_{\pi\pi}^{-1} A_{\pi p} \\ & I_p \end{bmatrix} \begin{bmatrix} \widetilde{A}_{\pi\pi}^{-1} & \\ & \widetilde{S}^{-1} \end{bmatrix} \begin{bmatrix} I_\pi & \\ -A_{p\pi} \widetilde{A}_{\pi\pi}^{-1} & I_p \end{bmatrix}, \tag{25}$$

where $\widetilde{A}_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ are inexact applications of the inverse of the leading block $A_{\pi\pi}$ and the approximate Schur complement, respectively. For instance, local inner preconditioners for $A_{\pi\pi}$ and $\widetilde{S}$ can be used for this task, like incomplete factorizations, approximate inverses or multigrid approaches.

Remark 3.1. The approximate decoupling factors $\widetilde{F}$ and $\widetilde{G}$ are used only for the computation of $\widetilde{S}$ and do not replace the relevant terms in the triangular factors in equation (25). In this sense, the EDFA algorithm can be regarded as a member of the *mixed constraint preconditioners* class [95,97], where a twofold approximation for the inverse of the leading block is inherently introduced. Similarly, it can be also viewed as an example of application in a non-symmetric context of the *multigrid reduction* framework, e.g. [65,80], where face and element pressures play the role of *fine* and *coarse* nodes, respectively, and $\widetilde{F}$ and $\widetilde{G}$ are approximations of the optimal restriction and prolongation operators from the fine to the coarse grid.

The approximation of the decoupling factors F and G is performed by solving the sequence of MRHS systems (23a) and (23b) inexactly in properly restricted subspaces. For the sake of simplicity, we refer to system (23a), but the same developments can be easily extended to (23b). The m-th system reads:

$$-A_{\pi\pi} \boldsymbol{g}^{(m),T} = \boldsymbol{a}_{p\pi}^{(m),T}, \tag{26}$$

where $A_{\pi\pi} = A_{\pi\pi}^T$ for symmetry reasons, $\boldsymbol{g}^{(m),T} = G^T \boldsymbol{e}^{(m)}$, $\boldsymbol{a}_{p\pi}^{(m),T} = A_{p\pi}^T \boldsymbol{e}^{(m)}$, and $\boldsymbol{e}^{(m)}$ is the m-th vector of the canonical basis of $\mathbb{R}^{N_e}$, which plays the role of restriction operator over columns. The minus sign has been introduced at both sides of equation (26) to obtain an SPD problem, since $A_{\pi\pi}$ is negative definite. Let us now consider the set $Q = \{1, \dots, N_f\} \subset \mathbb{N}$ and a sequence of (possibly overlapping) subsets $Q^{(m)} \subseteq Q$, whose size is $|Q^{(m)}| = s^{(m)}$, with $m = 1, \dots, N_e$. The m-th restriction operator over rows,

$$R_r^{(m)} : \mathbb{R}^{N_f} \to \mathbb{R}^{s^{(m)}} \tag{27}$$

is expressed as:

$$R_r^{(m)} = \begin{bmatrix} \boldsymbol{f}_{Q_1^{(m)}}^T \\ \vdots \\ \boldsymbol{f}_{Q_{s^{(m)}}^{(m)}}^T \end{bmatrix}, \tag{28}$$

where $\boldsymbol{f}_\ell$ is the ℓ-th column vector of the canonical basis of $\mathbb{R}^{N_f}$ and $Q_i^{(m)}$ is the i-th member of $Q^{(m)}$. The application of the operator $R_r^{(m)}$ to equation (26) leads to the following system (Fig. 2):

$$-A_{\pi\pi}^{(m)} \widetilde{\boldsymbol{g}}^{(m),T} = R_r^{(m)} \boldsymbol{a}_{p\pi}^{(m),T}, \tag{29}$$

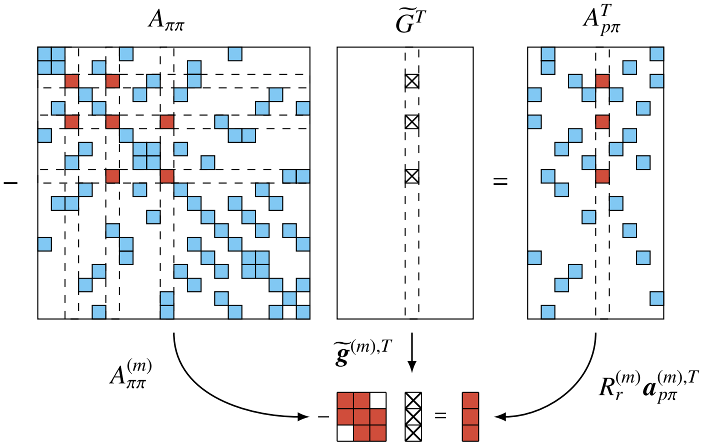


Fig. 2. Schematic representation of the restriction operators' action in the computation of $\widetilde{G}^T$.

where $A_{\pi\pi}^{(m)} = R_r^{(m)} A_{\pi\pi} R_r^{(m),T}$ is a symmetric restriction of $A_{\pi\pi}$ to the entries in the rows and columns with indices in $Q^{(m)}$ and $\widetilde{\boldsymbol{g}}^{(m),T} = R_r^{(m)} \boldsymbol{g}^{(m),T}$ is the restriction of the m-th row of G to the entries in the columns with indices in $Q^{(m)}$. Since $-A_{\pi\pi}^{(m)}$ is a symmetric square submatrix of the SPD matrix $-A_{\pi\pi}$, it is guaranteed to be SPD as well. The sequence of systems (29) can be inexpensively solved by an inner direct solver, provided that the cardinality of $Q^{(m)}$ is small enough.

The restricted vector $\widetilde{\boldsymbol{g}}^{(m)}$ obtained from the solution of system (29) is an approximation of the m-th row of the exact decoupling factor G and inherits an optimal property, as stated by the following result.

Proposition 3.1. *Let $A \in \mathbb{R}^{n\times n}$ be SPD and $R \in \mathbb{R}^{m\times n}$ ($m < n$) be a restriction operator from $\mathbb{R}^n$ to $\mathbb{R}^m$. Then, for any RHS vector $\boldsymbol{b} \in \mathbb{R}^n$, the solution $\boldsymbol{x} \in \mathbb{R}^m$ to the restricted system:*

$$RAR^T\boldsymbol{x} = R\boldsymbol{b} \tag{30}$$

is such that the error $\boldsymbol{e} = A^{-1}\boldsymbol{b} - R^T\boldsymbol{x}$ has minimal energy norm with respect to the A-inner product.

Proof. The energy norm of $\boldsymbol{e}$ with respect to A reads:

$$\|\boldsymbol{e}\|_A = \sqrt{\boldsymbol{e}^T A\boldsymbol{e}}. \tag{31}$$

The contribution under square root in equation (31) is a quadratic function $\Phi(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}^+$:

$$\begin{aligned}\Phi(\boldsymbol{x}) &= \left(A^{-1}\boldsymbol{b} - R^T\boldsymbol{x}\right)^T A \left(A^{-1}\boldsymbol{b} - R^T\boldsymbol{x}\right) \\ &= \boldsymbol{x}^T RAR^T\boldsymbol{x} - 2\boldsymbol{x}^T R\boldsymbol{b} + \boldsymbol{b}^T A^{-1}\boldsymbol{b},\end{aligned} \tag{32}$$

which has a unique minimum in $\mathbb{R}^m$ being A SPD. Hence:

$$\min_{\mathbb{R}^n} \|\boldsymbol{e}\|_A = \sqrt{\Phi(\boldsymbol{t})}, \quad \text{with } \boldsymbol{t} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^m} \Phi(\boldsymbol{x}) \quad \Longleftrightarrow \quad \nabla\Phi(\boldsymbol{x}) = \boldsymbol{0}. \tag{33}$$

Condition (33) applied to equation (32) immediately yields:

$$RAR^T\boldsymbol{x} - R\boldsymbol{b} = \boldsymbol{0}, \tag{34}$$

which completes the proof. □

Remark 3.2. Proposition 3.1 guarantees that the restricted vector $\widetilde{\boldsymbol{g}}^{(m)}$ is the best approximation of $\boldsymbol{g}^{(m)}$ that can be computed for the components selected by the set $Q^{(m)}$, in the sense of the energy norm with respect to the $A_{\pi\pi}$-inner product. Hence, an accurate selection of such components, so as to identify the most important ones for each row, is fundamental for the quality of the approximation $\widetilde{G}$ and, similarly, of $\widetilde{F}$ and $\widetilde{H}$.

Finally, the assemblage of the N_e contributions $\widetilde{\boldsymbol{g}}^{(m)}$ from equation (29), prolonged back to $\mathbb{R}^{N_f}$, gives rise to the approximate factor $\widetilde{G}$. Recalling that restrictions and prolongations are dual operators, $\widetilde{G}$ is easily obtained as:

$$\widetilde{G} = \sum_{m=1}^{N_e} \boldsymbol{e}^{(m)} \widetilde{\boldsymbol{g}}^{(m)} R_r^{(m)}. \tag{35}$$


(a) Cartesian grid

(b) Non-Cartesian grid

Fig. 3. Sketch of the base patterns for Cartesian and non-Cartesian grids. The front and right elements have been removed to improve the readability of the subpanels.

Operating similarly for equation (23b), we obtain:

$$\widetilde{F} = \sum_{m=1}^{N_e} R_r^{(m),T} \widetilde{\boldsymbol{f}}^{(m)} \boldsymbol{e}^{(m),T}, \tag{36}$$

where $\widetilde{\boldsymbol{f}}^{(m)}$ are the solution of the N_e restricted SPD systems:

$$-A_{\pi\pi}^{(m)} \widetilde{\boldsymbol{f}}^{(m)} = R_r^{(m)} \boldsymbol{a}_{\pi p}^{(m)}, \tag{37}$$

with $\boldsymbol{a}_{\pi p}^{(m)} = A_{\pi p} \boldsymbol{e}^{(m)}$. Of course, the restriction operators $R_r^{(m)}$ can be the same as for $\widetilde{G}$ or based on a different sequence of subsets $W^{(m)} \subseteq Q$.

As observed in Remark 3.2, the sequence of subsets $Q^{(m)}$ and $W^{(m)}$, along with their size $s^{(m)}$, affects the density of the approximate decoupling factors and governs the effectiveness of the EDFA preconditioner. In fact, the entries of $Q^{(m)}$ and $W^{(m)}$ are the indices of the non-zero entries computed for the m-th row of $\widetilde{G}$ and column of $\widetilde{F}$, respectively. To be effective, the sets $Q^{(m)}$ and $W^{(m)}$ should roughly identify for each row of $\widetilde{G}$ and column of $\widetilde{F}$ the largest entries of G and F. This key operation is carried out by means of two techniques, referred to as *static* and *dynamic* in the sequel, aimed at selecting the most influential entries expected in G and F.

First of all, for the sake of simplicity, we use a single sequence of sets $Q^{(m)}$ for both decoupling factors. A natural initial guess for $Q^{(m)}$ is the set of indices of the non-zero entries belonging to the columns of $A_{p\pi}^T$, which is denser than $A_{\pi p}$. Such pattern is referred to as $Q_{A_{p\pi}}^{(m)}$. Fig. 2 schematically shows how the restricted systems can be retrieved from the global one using the set $Q_{A_{p\pi}}^{(m)}$. The two strategies for computing $Q^{(m)}$ starting from $Q_{A_{p\pi}}^{(m)}$ are as follows.

1. *Static technique.* The non-zero entries of $Q_{A_{p\pi}}^{(m)}$ can be derived by the discretization. In particular, the non-zeros lying in the m-th row of $A_{p\pi}$ identify the faces of the cells connected with the m-th element, as illustrated in Fig. 3. Notice that the front and right elements have been removed for the sake of readability, being the overall patch symmetric along the three principal directions. The central element (red-filled faces) is the m-th cell, which is connected to six adjacent elements, and the colored faces correspond to the indices of the non-zero entries in the m-th row of $A_{p\pi}$. Note that, depending on whether the grid is Cartesian or non-Cartesian, the patterns are different. This is a direct consequence of the elemental matrices $B^{E,-1}$ structure in equation (12), which derives from the mutual relationships among the basis functions of the $\mathbb{RT}_0$ space for hexahedral elements. For a regular hexahedron, in fact, matrix $B^{E,-1}$ is block-diagonal, while this property is no longer valid for a general-shape hexahedron. Since the solution of system (26) can be physically interpreted as the face pressures induced by the fluid fluxes related to the pressure gradients occurring in neighboring cells, the pattern $Q_{A_{p\pi}}^{(m)}$ can be extended by adding the connection to faces belonging to close cells where the pressure perturbation is expected to propagate. From an algebraic viewpoint, the static technique is based on partitioning the problem domain into overlapping subregions built around each cell and keeping the face connections.
2. *Dynamic technique.* The starting pattern $Q_{(0)}^{(m)} = Q_{A_{p\pi}}^{(m)}$ is progressively enlarged during the computation of $\widetilde{\boldsymbol{g}}^{(m)}$ and $\widetilde{\boldsymbol{f}}^{(m)}$ with the aid of an iterative strategy. After computing $\widetilde{\boldsymbol{g}}_{(0)}^{(m),T}$ from the solution of system (29), with the restriction operator $R_{r,(0)}^{(m)}$ built on $Q_{(0)}^{(m)}$, the residual of the prolonged system
$$\boldsymbol{r}_{(0)}^{(m)} = \boldsymbol{a}_{p\pi}^{(m),T} + A_{\pi\pi} R_{r,(0)}^{(m),T} \widetilde{\boldsymbol{g}}_{(0)}^{(m),T} \tag{38}$$
is obtained and used to expand $Q_{(0)}^{(m)}$ by incorporating the indices of the largest components of $\boldsymbol{r}_{(0)}^{(m)}$, thus obtaining $Q_{(1)}^{(m)}$. The process can be iterated to obtain $Q_{(2)}^{(m)}$, $Q_{(3)}^{(m)}$, etc., until a certain exit criterion is met. The same procedure applies to system (37).

Remark 3.3. Equation (38) is not expensive to compute. In fact, the matrix $A_{\pi\pi} R_{r,0}^{(m),T}$ is the restriction of $A_{\pi\pi}$ to the columns with indices in $Q_{(0)}^{(m)}$. However, such columns are sparse and contain only the connection of a face with the faces of the two sharing cells. Hence, the only non-zero entries of $\boldsymbol{r}_{(0)}^{(m)}$ correspond to the faces' indices of a set of neighboring elements. In practice, the dynamic strategy automatically selects the most significant entries among a subset of potential indices that should resemble the one associated with the static strategy.

Remark 3.4. The use of the prolonged residual to select the most significant entries to be retained is strictly related to the symmetry and positive definiteness of $-A_{\pi\pi}$. In fact, $\boldsymbol{r}_{(0)}^{(m)}$ is the direction of the gradient of the quadratic form associated to $-A_{\pi\pi}$, whose absolute minimum is the exact solution to equation (26). Therefore, the dynamic strategy can be also regarded as an incomplete steepest descent process, where only the largest contributions to the gradient direction are taken into account.

Remark 3.5. The EDFA preconditioner, in both the static and dynamic variants, exhibits the remarkable feature that its computation is embarrassingly parallel. In fact, the row- and column-wise approach, used to tackle the restricted solution to the MRHS systems (23a) and (23b), allows to solve each single linear system independently of the others. All the available processing units can be assigned batches of systems that are approximately solved at the same time, with a full and effective exploitation of the most modern computational architectures.

3.1. Spectral analysis

It is well-known that an eigenspectrum enclosed in the neighborhood of 1 is not a sufficient condition to guarantee a fast convergence of a non-symmetric iterative solver [98]. Nevertheless, computational experience shows that a matrix arising from PDEs discretization with a compact eigenspectrum very rarely produces a bad convergence rate. It is therefore interesting to investigate the influence of the approximations $\widetilde{A}_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ on the eigenspectrum of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$.

Let us introduce the matrices:

$$E_\pi = I_\pi - \widetilde{A}_{\pi\pi}^{-1} A_{\pi\pi}, \tag{39a}$$

$$E_S = I_p - \widetilde{S}^{-1}\widehat{S}, \tag{39b}$$

with $\widehat{S} = A_{pp} - A_{p\pi}\widetilde{A}_{\pi\pi}^{-1}A_{\pi p}$. E_π and E_S can be regarded as a measure of the quality of $\widetilde{A}_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ as approximations of the leading block and the Schur complement of $\mathcal{A}$ inverses, respectively. Recalling equations (18) and (25) and introducing the error matrices (39a) and (39b), the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ reads:

$$\mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} \widetilde{A}_{\pi\pi}^{-1}A_{\pi\pi} - \widetilde{A}_{\pi\pi}^{-1}A_{\pi p}\widetilde{S}^{-1}A_{p\pi}E_\pi & \widetilde{A}_{\pi\pi}^{-1}A_{\pi p}E_S \\ \widetilde{S}^{-1}A_{p\pi}E_\pi & \widetilde{S}^{-1}\widehat{S} \end{bmatrix}, \tag{40}$$

which can be re-written as:

$$\mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} I_\pi & 0 \\ 0 & I_p \end{bmatrix} + \begin{bmatrix} -\widetilde{A}_{\pi\pi}^{-1}A_{\pi p}\widetilde{S}^{-1}A_{p\pi} - I_\pi & \widetilde{A}_{\pi\pi}^{-1}A_{\pi p} \\ \widetilde{S}^{-1}A_{p\pi} & -I_p \end{bmatrix}\begin{bmatrix} E_\pi & 0 \\ 0 & E_S \end{bmatrix} = \mathcal{I} + \mathcal{H}\mathcal{E}. \tag{41}$$

Therefore, the eigenvalues λ of $\mathcal{P}^{-1}\mathcal{A}$ are:

$$\lambda\left(\mathcal{P}^{-1}\mathcal{A}\right) = 1 + \zeta, \tag{42}$$

where ζ denotes the eigenvalues of $\mathcal{H}\mathcal{E}$. As expected, this ensures that $\mathcal{P}^{-1}\mathcal{A}$ converges to the identity as the sub-problem approximations $\widetilde{A}_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ are progressively improved, i.e., E_π and E_S tend to the null matrix.

The leading block $A_{\pi\pi}$ is symmetric negative definite, hence several specific algebraic preconditioners, such as incomplete factorization, approximate inverse or algebraic multigrid, are already available in literature and able to force E_π to approach the null matrix as needed. In the limit of $E_\pi = 0$, we have the result that follows.

Proposition 3.2. *If $E_\pi = 0$, then the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ are either 1 with multiplicity N_f, or equal to those of $\widetilde{S}^{-1}S$ for S given in equation (22).*

Proof. The proof follows immediately from equation (40). □

Hence, the key for the performance of the EDFA preconditioner is the computation of $\widetilde{S}$ and the application of its inverse. Recalling equation (21), the approximate decoupling factors $\widetilde{F}$ and $\widetilde{G}$ can be written as:

$$\widetilde{G} = -A_{p\pi}A_{\pi\pi}^{-1} + R_G, \tag{43a}$$

$$\widetilde{F} = -A_{\pi\pi}^{-1}A_{\pi p} + R_F, \tag{43b}$$

where R_G and R_F collect the sequence of final residuals in equation (38) obtained during the EDFA preconditioner computation. The eigenvalue distribution of $\widetilde{S}^{-1}S$ depends on R_G and R_F according to the following result.

Theorem 3.3. *Let S and $\widetilde{S}$ be the matrices defined in equations (22) and (24), respectively, and $r = \max\{\|R_G\|, \|R_F\|\}$ for any compatible matrix norm. The real and imaginary part of the eigenvalues μ of $\widetilde{S}^{-1}S$ read:*

$$\Re(\mu) = \frac{1+\xi}{(1+\xi)^2+\eta^2}, \qquad \Im(\mu) = \frac{-\eta}{(1+\xi)^2+\eta^2}, \tag{44}$$

for pairs of real numbers (ξ, η) such that $\sqrt{\xi^2+\eta^2} \le \rho(r)$ and:

$$\rho(r) = \|S^{-1}\| \left[\|A_{\pi\pi}\| r^2 + \left(\|A_{\pi p}\| + \|A_{p\pi}\|\right) r \right]. \tag{45}$$

Proof. Using equations (43a) and (43b), the difference matrix $R = \widetilde{S} - S$ reads:

$$\begin{aligned} R &= A_{p\pi} A_{\pi\pi}^{-1} A_{\pi p} - \widetilde{G} A_{\pi\pi} \widetilde{F} \\ &= R_G A_{\pi p} + A_{p\pi} R_F - R_G A_{\pi\pi} R_F. \end{aligned} \tag{46}$$

Since:

$$\begin{aligned} \widetilde{S}^{-1} S &= (S+R)^{-1} S \\ &= \left[S\left(I_p + S^{-1}R\right)\right]^{-1} S \\ &= \left(I_p + S^{-1}R\right)^{-1}, \end{aligned} \tag{47}$$

the eigenvalues $\mu \in \mathbb{C}$ of $\widetilde{S}^{-1}S$ are:

$$\mu = \frac{1}{1+\lambda(S^{-1}R)}, \tag{48}$$

for all eigenvalues $\lambda \in \mathbb{C}$ of $S^{-1}R$. Setting in equation (48) $\lambda = \xi + i\eta$, $\xi, \eta \in \mathbb{R}$, provides relationships (44). The modulus of $\lambda(S^{-1}R)$ is bounded by any matrix norm of $S^{-1}R$ induced by a vector norm, therefore:

$$\sqrt{\xi^2+\eta^2} \le \|S^{-1}\| \left(\|R_G\| \, \|A_{\pi p}\| + \|A_{p\pi}\| \, \|R_F\| + \|R_G\| \, \|A_{\pi\pi}\| \, \|R_F\|\right). \tag{49}$$

The right-hand side of inequality (49) is bounded from above by $\rho(r)$ defined in equation (45), thus completing the proof. □

Remark 3.6. Equations (44) have a singularity for $\xi = -1$ and $\eta = 0$. If eigenvalues of $S^{-1}R$ are located in the neighborhood of this point, there exist eigenvalues of $\widetilde{S}^{-1}S$ with modulus approaching infinity and real part either positive or negative. This occurrence might negatively affect the convergence rate of the iterative solver applied to $\mathcal{P}^{-1}\mathcal{A}$, independently on the quality of the approximation of the leading block inverse. To avoid such a condition, it is therefore important to reduce the residuals in the computation of $\widetilde{G}$ and $\widetilde{F}$ such that $\rho(r) \le \epsilon$, with $\epsilon < 1$. Recalling equation (45), this yields the following upper bound to r:

$$r < \sqrt{\frac{\left(\|A_{\pi p}\| + \|A_{p\pi}\|\right)^2}{4\|A_{\pi\pi}\|^2} + \frac{\epsilon}{\|S^{-1}\|\|A_{\pi\pi}\|}} - \frac{\|A_{\pi p}\| + \|A_{p\pi}\|}{2\|A_{\pi\pi}\|}. \tag{50}$$

The smaller is the right-hand side of inequality (50), the more accurate should be the computation of $\widetilde{G}$ and $\widetilde{F}$ in order to guarantee a fast solver convergence.

Remark 3.7. The norms of the matrices involved in the upper bound (50) depend on the material parameters of the governing PDEs in equations (1a)-(1b) and on the space-time discretization grid size. In particular, we have:

$$\|A_{\pi\pi}\| \simeq \|A_{\pi p}\| \simeq \|A_{p\pi}\| \propto \frac{|K|h}{\gamma}, \qquad \|S\| \propto \frac{ch^3}{\Delta t} + \frac{|K|h}{\gamma}, \tag{51}$$

where h denotes a linear characteristic measure of the elements in the computational grid. Hence, the upper bound of r in equation (50) increases with $c\gamma/|K|$ and, with respect to the discretization parameters, is a function $f(h, \Delta t)$ such that:

$$f \sim O\left(\frac{h}{\sqrt{\Delta t}}\right). \tag{52}$$

Equation (52) shows that the requirement of denser and higher-quality approximations of $\widetilde{G}$ and $\widetilde{F}$ becomes stricter as h is reduced and Δt is increased. In particular, the most severe condition in a transient simulation is expected for $\Delta t \to \infty$, i.e., at steady state.

Remark 3.8. The min/max analysis of the functions in equation (44), within the circle with radius $\rho(r)$ of equation (45), provides a bound for the eigenspectrum size of $\widetilde{S}^{-1}S$ along the real and imaginary axes. Considering $\rho(r) \le \epsilon$, $\epsilon < 1$, we have:

$$\max(\Re(\mu)) = \frac{1}{1-\epsilon}, \qquad \min(\Re(\mu)) = \frac{1}{1+\epsilon}, \qquad \text{and} \qquad \max(|\Im(\mu)|) = \frac{\epsilon}{1-\epsilon^2}. \tag{53}$$

For an appropriate selection of $\widetilde{G}$ and $\widetilde{F}$, i.e., fixing r such that equation (50) is satisfied, using equations (51) in (45) gives:

$$\rho(r) \sim O\left(\frac{\Delta t}{h^2}\right) r^2, \tag{54}$$

hence, the eigenspectrum bounds in equation (53) enlarge with $\Delta t / h^2$.

3.2. Implementation details

The static and dynamic variants of the EDFA preconditioner require a set of user-specified elements to be properly set up.

The static technique needs the sets $Q^{(m)} \subseteq \left\{1, 2, \dots, N_f\right\}$ for $m = 1, \dots, N_e$, which correspond to the indices of faces connected to a certain cell. The level of such a connection, i.e., the neighbors, or the neighbors of the neighbors, and so on, is defined by means of a domain partition into overlapping subregions built around each cell. These subregions are defined on the basis of physical considerations related to the expected directions of fluxes.

The dynamic variant can be regarded as fully algebraic and requires a set of user-specified parameters controlling the enlargement of the initial set $Q^{(m)}_{(0)}$ defined for $m = 1, \dots, N_e$. Assuming $Q^{(m)}_{(0)} = Q^{(m)}_{A_{p\pi}}$, the selected user-defined parameters are:

- n_{add}: maximum number of entries added to $Q^{(m)}_{(k-1)}$ at the k-th step of the dynamic procedure;
- n_{ent}: total maximum number of new entries added to $Q^{(m)}_{(0)}$.

The iterative process continues until n_{ent} has been reached. Alternatively, it is also possible to set a maximum number of steps, $it_{\max}$, instead of n_{ent}.

The computation of $\widetilde{F}$, $\widetilde{G}$ and $\widetilde{S} = A_{pp} - \widetilde{H}$, with either the static or dynamic technique, is followed by a check of the non-zero entries size. *Pre-* and *post-filtration* techniques are implemented with the purpose of further sparsifying the approximate Schur complement by discarding those entries whose absolute value is smaller than a user-defined tolerance, namely τ_{filt}, relative to the Euclidean norm of the corresponding row. Performing pre- and/or post-filtration produces an additional cost in the preconditioner set-up, which might be anyway beneficial at the application stage. With the aim at preventing possible breakdowns in the inexact application of $\widetilde{S}^{-1}$, all the diagonal entries are preserved irrespective of the dropping threshold.

Recalling that A_{pp} is the only block of $\mathcal{A}$ changing during a transient simulation, the preconditioner set-up can be split into two stages. The first one, which can be carried out only once at the beginning of the simulation and then recycled at every system solution, consists of the computation of both the inner preconditioners for the inexact application of $\widetilde{A}^{-1}_{\pi\pi}$ and $\widetilde{H}$, the latter being the most time demanding operation, along with a pre-filtration of $\widetilde{G}$ and $\widetilde{F}$ and/or a post-filtration of $\widetilde{H}$, if needed. The second one, performed at the beginning of each time step, includes the update of $\widetilde{S} = A_{pp} - \widetilde{H}$, the post-filtration, if required, and the computation of the inner preconditioner for the inexact application of $\widetilde{S}^{-1}$. In summary, Algorithms 1 and 2 provide an overview of the sequence of operations needed to compute the first and second stage of the EDFA preconditioner in both its variants. The overall workflow of the simulator in a full transient problem is provided in Algorithm 3.

4. Numerical results

The computational performance of the EDFA preconditioner is investigated in both synthetic and real-world reservoir applications.

First, a simple academic benchmark (Test 0) is discretized by three progressively refined grids and employed to investigate the theoretical behavior of the EDFA preconditioner. In Test 0, $A^{-1}_{\pi\pi}$ and $\widetilde{S}^{-1}$ are applied exactly with the only approximation resting on $\widetilde{S}$. The size of the porous domain is $L = 10$ m, $W = 1$ m and $H = 1$ m (Fig. 4) with a homogeneous and isotropic hydraulic conductivity equal to $1.73 \cdot 10^{-5}\,\frac{\text{m}}{\text{d}}$. The flow scenario simulates the injection and production of water at the two opposite faces of the domain along the main axis.

Algorithm 1 EDFA COMPUTATION: STAGE 1 $\left[\widetilde{H}, \widetilde{A}_{\pi\pi}^{-1}\right]$ = EDFA_first_stage$\left(N_e, Q^{(m)}, n_{\text{ent}}, n_{\text{add}}, \tau_{\text{filt_pre}}, \tau_{\text{filt_post}}, A_{\pi\pi}, A_{\pi p}, A_{p\pi}\right)$.

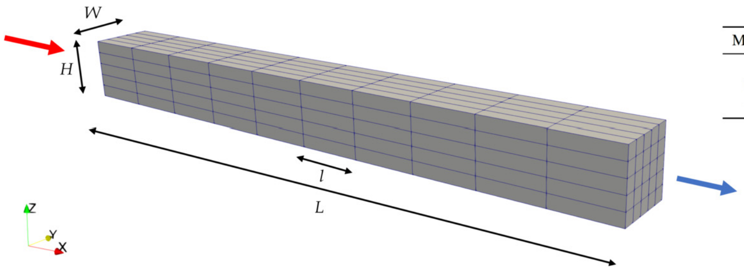
```
if EDFA_static then
    for $m \leftarrow 1, N_e$ do
        $\boldsymbol{a}_{p\pi}^{(m),T} = A_{p\pi}^T \boldsymbol{e}^{(m)}, \quad \boldsymbol{a}_{\pi p}^{(m)} = A_{\pi p} \boldsymbol{e}^{(m)}$
        Build $R_r^{(m)}$ based on $Q^{(m)}$ (Equation (28))
        $A_{\pi\pi}^{(m)} = R_r^{(m)} A_{\pi\pi} R_r^{(m),T}$
        Solve $-A_{\pi\pi}^{(m)} \widetilde{\boldsymbol{g}}^{(m),T} = R_r^{(m)} \boldsymbol{a}_{p\pi}^{(m),T}$
        Solve $-A_{\pi\pi}^{(m)} \widetilde{\boldsymbol{f}}^{(m)} = R_r^{(m)} \boldsymbol{a}_{\pi p}^{(m)}$
        Perform pre-filtration on $\widetilde{\boldsymbol{g}}^{(m),T}$ and $\widetilde{\boldsymbol{f}}^{(m)}$ with tolerance $\tau_{\text{filt\_pre}}$, if required
        $\widetilde{G} \leftarrow \widetilde{G} + \boldsymbol{e}^{(m)} \widetilde{\boldsymbol{g}}^{(m)} R_r^{(m)}, \quad \widetilde{F} \leftarrow \widetilde{F} + R_r^{(m),T} \widetilde{\boldsymbol{f}}^{(m)} \boldsymbol{e}^{(m),T}$
    end for
else if EDFA_dynamic then
    for $m \leftarrow 1, N_e$ do
        $\boldsymbol{a}_{p\pi}^{(m),T} = A_{p\pi}^T \boldsymbol{e}^{(m)}, \quad \boldsymbol{a}_{\pi p}^{(m)} = A_{\pi p} \boldsymbol{e}^{(m)}$
        Build $R_{r,(0)}^{(m)}$ based on $Q_{(0)}^{(m)} = Q_{A_{p\pi}}^{(m)}$ (Equation (28))
        $A_{\pi\pi,(0)}^{(m)} = R_{r,(0)}^{(m)} A_{\pi\pi} R_{r,(0)}^{(m),T}$
        Solve $-A_{\pi\pi,(0)}^{(m)} \widetilde{\boldsymbol{g}}_{(0)}^{(m),T} = R_{r,(0)}^{(m)} \boldsymbol{a}_{p\pi}^{(m),T}$
        Compute $\boldsymbol{r}_{(0)}^{(m)} = \boldsymbol{a}_{p\pi}^{(m),T} + A_{\pi\pi} R_{r,(0)}^{(m),T} \widetilde{\boldsymbol{g}}_{(0)}^{(m),T}$
        $n_{\text{prog}} = 0, \quad k = 0$        ▷ Initializing the new entries and sweeps counters, respectively
        while $n_{\text{prog}} < n_{\text{ent}}$ do
            $k \leftarrow k + 1$
            $n = \min(n_{\text{add}}, n_{\text{ent}} - n_{\text{prog}})$
            Obtain $Q_{(k)}^{(m)}$ by adding to $Q_{(k-1)}^{(m)}$ at most $n$ new indices associated with the largest components of $\left|\boldsymbol{r}_{(k-1)}^{(m)}\right|$
            Update $n_{\text{prog}}$
            Build $R_{r,(k)}^{(m)}$ based on $Q_{(k)}^{(m)}$ (Equation (28))
            $A_{\pi\pi,(k)}^{(m)} = R_{r,(k)}^{(m)} A_{\pi\pi} R_{r,(k)}^{(m),T}$
            Solve $-A_{\pi\pi,(k)}^{(m)} \widetilde{\boldsymbol{g}}_{(k)}^{(m),T} = R_{r,(k)}^{(m)} \boldsymbol{a}_{p\pi}^{(m),T}$
            Compute $\boldsymbol{r}_{(k)}^{(m)} = \boldsymbol{a}_{p\pi}^{(m),T} + A_{\pi\pi} R_{r,(k)}^{(m),T} \widetilde{\boldsymbol{g}}_{(k)}^{(m),T}$
        end while
        Solve $-A_{\pi\pi,(k)}^{(m)} \widetilde{\boldsymbol{f}}^{(m)} = R_{r,(k)}^{(m)} \boldsymbol{a}_{\pi p}^{(m)}$
        Perform pre-filtration on $\widetilde{\boldsymbol{g}}_{(k)}^{(m),T}$ and $\widetilde{\boldsymbol{f}}^{(m)}$ with tolerance $\tau_{\text{filt\_pre}}$, if required
        $\widetilde{G} \leftarrow \widetilde{G} + \boldsymbol{e}^{(m)} \widetilde{\boldsymbol{g}}^{(m)} R_r^{(m)}, \quad \widetilde{F} \leftarrow \widetilde{F} + R_r^{(m),T} \widetilde{\boldsymbol{f}}^{(m)} \boldsymbol{e}^{(m),T}$
    end for
end if
Compute $\widetilde{H} = \widetilde{G} A_{\pi\pi} \widetilde{F}$
Perform post-filtration on $\widetilde{H}$ with tolerance $\tau_{\text{filt\_post}}$, if required
Compute the inner preconditioner for the inexact application of $\widetilde{A}_{\pi\pi}^{-1}$
```

Algorithm 2 EDFA COMPUTATION: STAGE 2 $\left[\widetilde{S}^{-1}\right]$=EDFA_second_stage$\left(A_{pp}, \widetilde{H}, \tau_{\text{filt_post}}\right)$.

```
$\widetilde{S} = A_{pp} - \widetilde{H}$
Perform post-filtration on $\widetilde{S}$ with tolerance $\tau_{\text{filt\_post}}$, if required
Compute the inner preconditioner for the inexact application of $\widetilde{S}^{-1}$
```

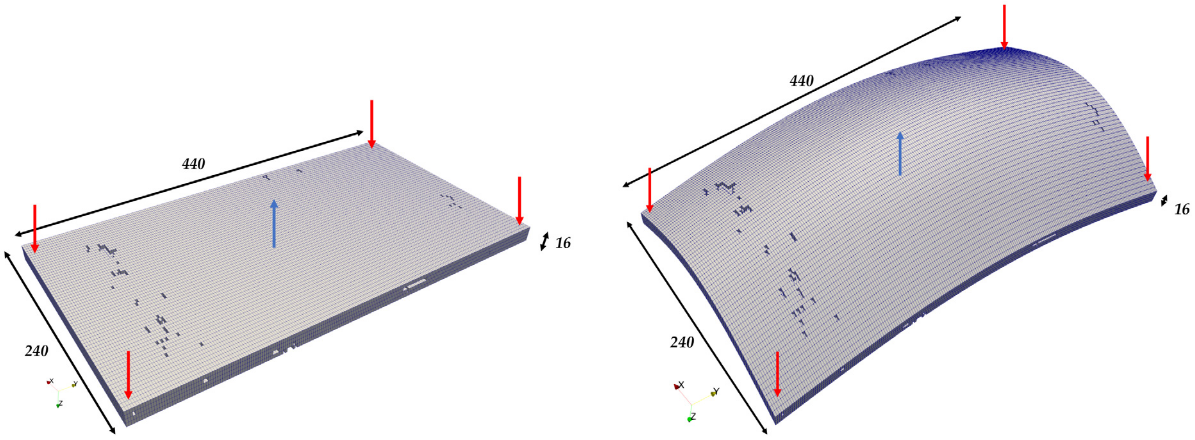
Algorithm 3 PSEUDOCODE OF THE FLOW SIMULATOR.

```
Compute blocks $A_{\pi\pi}$, $A_{\pi p}$, $A_{p\pi}$, and $A_{pp}$
$\left[\widetilde{H}, \widetilde{A}_{\pi\pi}^{-1}\right]$ = EDFA_first_stage$\left(N_e, Q^{(m)}, n_{\text{ent}}, n_{\text{add}}, \tau_{\text{filt\_pre}}, \tau_{\text{filt\_post}}, A_{\pi\pi}, A_{\pi p}, A_{p\pi}\right)$
Initialize $\boldsymbol{p}^0$, $\boldsymbol{\pi}^0$ and the RHS of equation (18), $\boldsymbol{f}$
Initialize the time variable $t$ and the first increment $\Delta t$
while $t < T$ do
    $t = t + \Delta t$
    Update $\boldsymbol{f}$ and $A_{pp}$ with the time-dependent contributes
    $\left[\widetilde{S}^{-1}\right]$ = EDFA_second_stage$\left(A_{pp}, \widetilde{H}, \tau_{\text{filt\_post}}\right)$
    Solve system (18)
    Update $\Delta t$ (see equation (58))
end while
```



Mesh	l	N_e	N_f
a	$L/10$	250	875
b	$L/50$	1250	4275
c	$L/100$	2500	8525

Fig. 4. Test 0: Parallelepiped domain and size of the three meshes employed. The blue and red arrows indicate the position of the producer and injector, respectively.



(a) (b)

Fig. 5. Planar (a) and dome-structured (b) reservoirs, used as domains for Tests 1, 3 and 2, 4, respectively. The domain size is in meters.

Table 1
Set-up of the test cases and number of non-zeros of the resulting matrices, where $N_f = 171,070$ and $N_e = 51,741$. The values of the hydraulic conductivity in brackets are the minimum and maximum of the SPE10 model portion used herein. The distribution of the conductivity values throughout the domain follows that of the SPE10 data set.

Test		1	2	3	4
Reservoir type		Plain	Dome	Plain	Dome
Cond. tensor properties		Homogeneous Isotropic	Homogeneous Isotropic	Heterogeneous Anisotropic	Heterogeneous Isotropic
Cond. tensor type		Diagonal	Diagonal	Diagonal	Full
Horiz. conductivity	$\left[\frac{\text{m}}{\text{d}}\right]$	1.7E-05	1.7E-05	[3.0E-07, 2.0E+00]	[3.0E-07, 2.0E+00]
Vert. conductivity	$\left[\frac{\text{m}}{\text{d}}\right]$	1.7E-05	1.7E-05	[3.9E-11, 6.0E-01]	[3.0E-07, 2.0E+00]
nnz($\mathcal{A}$)		1,711,914	4,069,590	1,711,914	4,069,590
nnz($A_{\pi\pi}$)		481,636	1,723,900	481,636	1,723,900
nnz$(A_{\pi p})$		310,446	310,446	310,446	310,446
nnz$(A_{p\pi})$		589,299	1,704,711	589,299	1,704,711
nnz(A_{pp})		330,533	330,533	330,533	330,533

Then, the preconditioner is tested in four more challenging applications (Test 1 through 4). The relevant grid consists of four layers taken from the SPE10 model [99] and comprises 51,741 elements and 171,070 faces, for a total of 222,811 unknowns. The scenario being tested, depicted in Fig. 5, represents a reservoir with a producer located in the center and four injectors, one at each corner. The wells intercept the full thickness of the reservoir. The initial water pressure is uniform and equal to 140 bar, with the producer and injectors pumping at a constant pressure of 100 and 200 bar, respectively. Different variants of the model domain have been considered. In Test 1 and 3, the grid is Cartesian with a regular hexahedral discretization (Fig. 5a), whereas in Test 2 and 4 the planar structure has been deformed into a dome (Fig. 5b). The resulting grid is, therefore, non-Cartesian. In all tests, porosity spans the interval $[2.6 \cdot 10^{-5}, 0.5]$ with a spatial distribution following that of the SPE10 data set, rock (α) and water (β) compressibilities are $4.67 \cdot 10^{-5}\ \frac{1}{\text{bar}}$ and $4.84 \cdot 10^{-5}\ \frac{1}{\text{bar}}$, respectively, and the water specific weight (γ) is 0.101 $\frac{\text{bar}}{\text{m}}$. A summary of the test cases and their main properties, along with the non-zeros number of matrix $\mathcal{A}$ and its submatrices, is reported in Table 1. Notice that the latter depends only on the grid type.

Test 1, which is characterized by a homogeneous isotropic hydraulic conductivity in the form of a diagonal tensor, is aimed at introducing the operative principles of the proposed preconditioner variants. A sensitivity analysis is carried out on the patterns selected for the static variant and on the two governing user-specified parameters for the dynamic technique. Then, the EDFA preconditioner is employed in a transient simulation to evaluate the effect of time, and, in particular, the size of the time step, Δt, on its performance. Test 2 preserves the same hydraulic properties as Test 1, but highlights the influence of a non-Cartesian mesh in the optimal setting of the preconditioner. Finally, Tests 3 and 4 investigate the efficiency and robustness of the EDFA preconditioner in challenging real-world conditions. Specifically, Test 3 exhibits a highly heterogeneous and anisotropic conductivity distribution, as derived from the properties of the SPE10 model and expressed in the form of a diagonal tensor. On the contrary, the dome reservoir in Test 4 is characterized by a heterogeneous and isotropic conductivity field with a full tensor, obtained by extending the horizontal conductivity values $(K_{x,y})$ to the vertical direction (K_z) and rotating the principal axes of the resulting tensor so as to follow the curvature of the dome reservoir. The sensitivity analysis on the EDFA preconditioner performance is carried out for the system at steady state, then the overall performance is investigated in full-transient simulations.

Full GMRES [94] and Bi-CGStab [93], with the null vector used as initial guess, are elected as Krylov subspace methods to solve the sequence of non-symmetric linear systems (18) for Test 0 and Tests 1-4, respectively. The use of Bi-CGStab in Tests 1-4 is motivated by the faster convergence generally experienced in the numerical tests. The exit criterion for the iteration count relies on the reduction of the 2-norm of the relative residual below a prescribed threshold τ, i.e., $||\mathbf{r}_k||_2/||\mathbf{r}_0||_2 \leq \tau$, where k is the iteration number and $\tau = 10^{-8}$. The computational performance of the preconditioned Bi-CGStab solver is monitored by using the following indicators: (i) the iteration count, n_{it}, (ii) the preconditioner density, μ, defined as

$$\mu = \frac{\text{nnz}\left(\widetilde{A}_{\pi\pi}^{-1}\right) + \text{nnz}\left(A_{\pi p}\right) + \text{nnz}\left(A_{p\pi}\right) + \text{nnz}\left(\widetilde{S}^{-1}\right)}{\text{nnz}\left(A_{\pi\pi}\right) + \text{nnz}\left(A_{\pi p}\right) + \text{nnz}\left(A_{p\pi}\right) + \text{nnz}\left(A_{pp}\right)}, \tag{55}$$

where the function `nnz()` provides the number of non-zeros stored for a sparse matrix, and (iii) the CPU time split into t_{p_0}, t_p and t_s, needed to perform the first and second stage of the EDFA preconditioner set-up (Algorithm 1 and 2, respectively) and to iterate to convergence. We denote by $t_t = t_p + t_s$ the total time associated with the solution of the linear system in a single time step.

For the transient simulations, we consider also the Courant-Friedrichs-Lewy (CFL) number, which is defined as [100,101]:

$$\chi^E = \frac{Q^E \Delta t}{\Omega^E \phi^E}, \tag{56}$$

where Q^E is the water flux through the E-th element during a time step of size Δt. Specifically, two measures are reported depending on the type of analysis:

$$\chi_\infty = \max_E \left(\chi^E\right), \qquad \text{and} \qquad \overline{\chi_\infty} = \frac{\sum_{i=1}^{n_{\text{step}}} \chi_\infty^i}{n_{\text{step}}}, \tag{57}$$

where n_{step} is the number of temporal steps in the simulation.

The size of the time steps is dynamically adjusted during the transient simulations to stabilize the pressure change between two consecutive steps. The underlying criterion relies on the maximum pressure difference at the two previous steps, $\Delta p_{\max} = \max_E \left(p_n^E - p_{n-1}^E\right)$, and a user-defined goal for the pressure change, Δp_T, to define the optimal size of the next one:

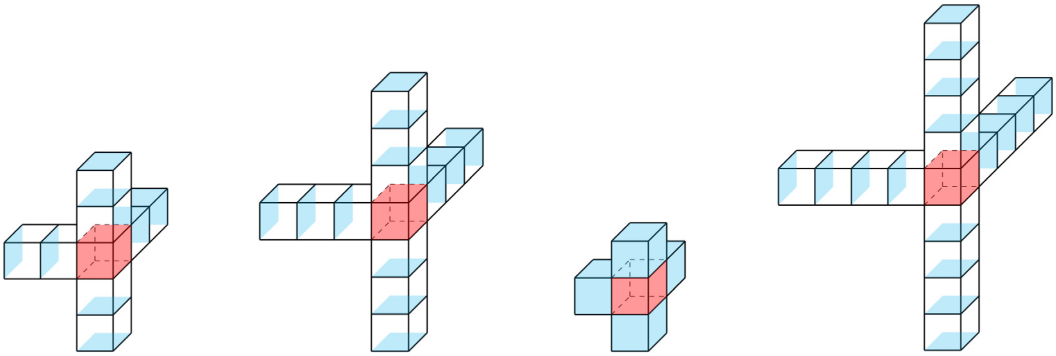
$$\Delta t_{n+1} = \min\left\{\Delta t_n \min\left\{\Delta t_{\text{mult}}, \frac{\Delta p_T}{\Delta p_{\max}}\right\}, \Delta t_{\max}\right\}, \tag{58}$$

where Δt_{mult} is a predefined multiplicative factor and $\Delta t_{\max}$ the maximum time step length. A relaxation factor can be introduced also in equation (58) [22].

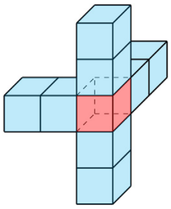
Both the solver and the preconditioner are implemented in Matlab. For the inexact application of $\widetilde{A}_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ we use an incomplete factorization with partial fill-in degree already available in Matlab. Of course, other powerful strategies, also more prone to a fully parallel implementation, can be used and will be considered in future developments of the algorithm. For the computation of $\widetilde{F}$ and $\widetilde{G}$, the `parfor` operator has been exploited. All numerical tests were carried out on an Intel®Core™i7 Quad-Core processor at 2.9 GHz with 16 GB of RAM.

4.1. Test 0: theoretical benchmark

The effectiveness of the EDFA preconditioner mostly relies on the quality of $\widetilde{S}$ (equation (24)) as an approximation of the Schur complement. In order to test the scalability with respect to the grid size, we consider the porous medium depicted in Fig. 4, subjected to a constant water flux along the main axis, x. The domain is discretized with three grids, labeled a, b and c, where the length of the elements along the x axis, l, is progressively reduced (see the table in Fig. 4). In order to assess the effect of $\widetilde{S}$ alone on the EDFA preconditioner quality, the inverses $A_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ are applied exactly in equation (25) via a direct solver. The tests are performed at steady state and the non-symmetric system (18) is solved with the full GMRES.



(a) Pattern A (b) Pattern B (c) Pattern C (d) Pattern D



(e) Pattern E

Fig. 6. Patterns for the static technique. The front and right elements have been removed to improve the readability.

Table 2

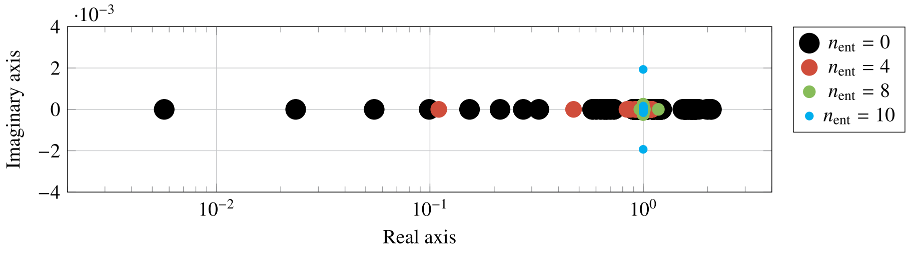
Test 0: Number of iterations to converge with different preconditioner set-up. The system is solved by full GMRES with $A_{\pi\pi}^{-1}$ and $\widetilde{S}^{-1}$ applied exactly through a direct solver.

EDFA static Pattern	EDFA dynamic		n_{it}		
	n_{ent}	n_{add}	Mesh a	Mesh b	Mesh c
-	0	0	24	42	68
-	2	2	17	41	61
-	4	2	10	23	42
-	6	2	6	21	20
-	8	2	4	11	21
-	10	2	3	9	9
-	12	2	2	6	10
-	14	2	1	4	6
A	-	-	13	19	46
B	-	-	8	11	17
D	-	-	3	7	11

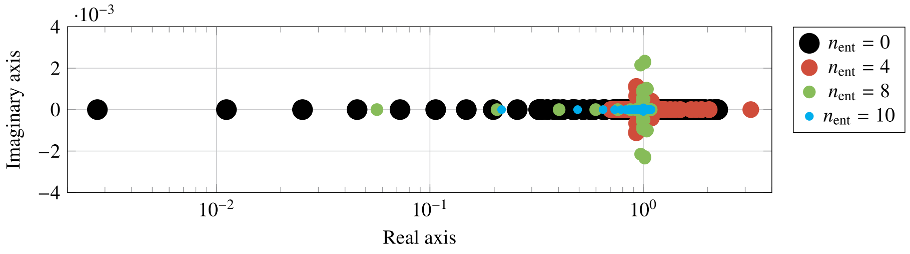
First, we introduce a set of patches of cells associated with the face pattern connection used in the static EDFA variant. The native pattern of the m-th column, previously presented in Fig. 3a for a Cartesian grid, can be statically enlarged by considering the patches A, B and D (Figs. 6a, 6b and 6d), assuming that the flux is mainly oriented along the principal conductivity axes. By distinction, the connections of patterns C and E (Figs. 6c and 6e) assume the presence of significant fluxes through all directions, as it might be for instance expected in the case of a full conductivity tensor. Similarly, a significant permeability anisotropy could suggest privileging one direction with respect to the orthogonal ones.

Table 2 summarizes the iteration count with both the static and dynamic variants for different parameter combinations. For a given preconditioner set-up, the number of iterations increases as the grid is refined, in agreement with Remark 3.7. However, the iteration count can be stabilized by properly setting the stencil of $\widetilde{G}$ and $\widetilde{F}$. This is also consistent with the physical interpretation of the decoupling factor, which provides a link between the nonzero entries of $\widetilde{G}$ and $\widetilde{F}$ and the pressure perturbations induced by the fluid fluxes around the cells. As the grid is refined, more cells are subjected to the spatial pressure changes and the relevant face unknowns need to be included in the column sparsity pattern so as to keep the number of iterations under control. This result gives a helpful indication as to the proper set-up of the EDFA preconditioner.

The eigenspectrum of $\widetilde{S}^{-1}S$ is considered for grids a and b (Fig. 7). In this analysis, $\widetilde{S}$ is computed by means of the dynamic technique. Since the simulations are run at steady state, the grid spacing l is the primary parameter affecting the eigenvalue distribution. Mesh refinement increases the spread of the eigenvalues. Nevertheless, they remain clustered



(a) Mesh a



(b) Mesh b

Fig. 7. Test 0: Eigenspectrum of $\widetilde{S}^{-1}S$ for mesh a and b. $\widetilde{S}$ is built by the dynamic technique with variable n_{ent} and $n_{\text{add}} = 1$.

Table 3
Test 0: Extreme values for the eigenvalues μ of $\widetilde{S}^{-1}S$. Pattern B and the combination $n_{\text{ent}} = 12$, $n_{\text{add}} = 1$ are used for the static and dynamic variants, respectively.

Mesh	EDFA variant	min($\lvert\mu\rvert$)	max($\lvert\mu\rvert$)	min($\Re(\mu)$)	max($\Re(\mu)$)	max($\lvert\Im(\mu)\rvert$)
a	Static	1.64E-01	1.07E+00	1.64E-01	1.07E+00	4.97E-16
b	Static	5.63E-02	1.09E+00	5.63E-02	1.09E+00	0.00E+00
c	Static	1.94E-02	1.11E+00	1.94E-02	1.11E+00	0.00E+00
a	Dynamic	1.00E+00	1.00E+00	1.00E+00	1.00E+00	2.38E-14
b	Dynamic	9.87E-01	2.92E+00	9.87E-01	2.92E+00	1.14E-03
c	Dynamic	9.38E-02	1.06E+00	9.38E-02	1.06E+00	1.57E-05

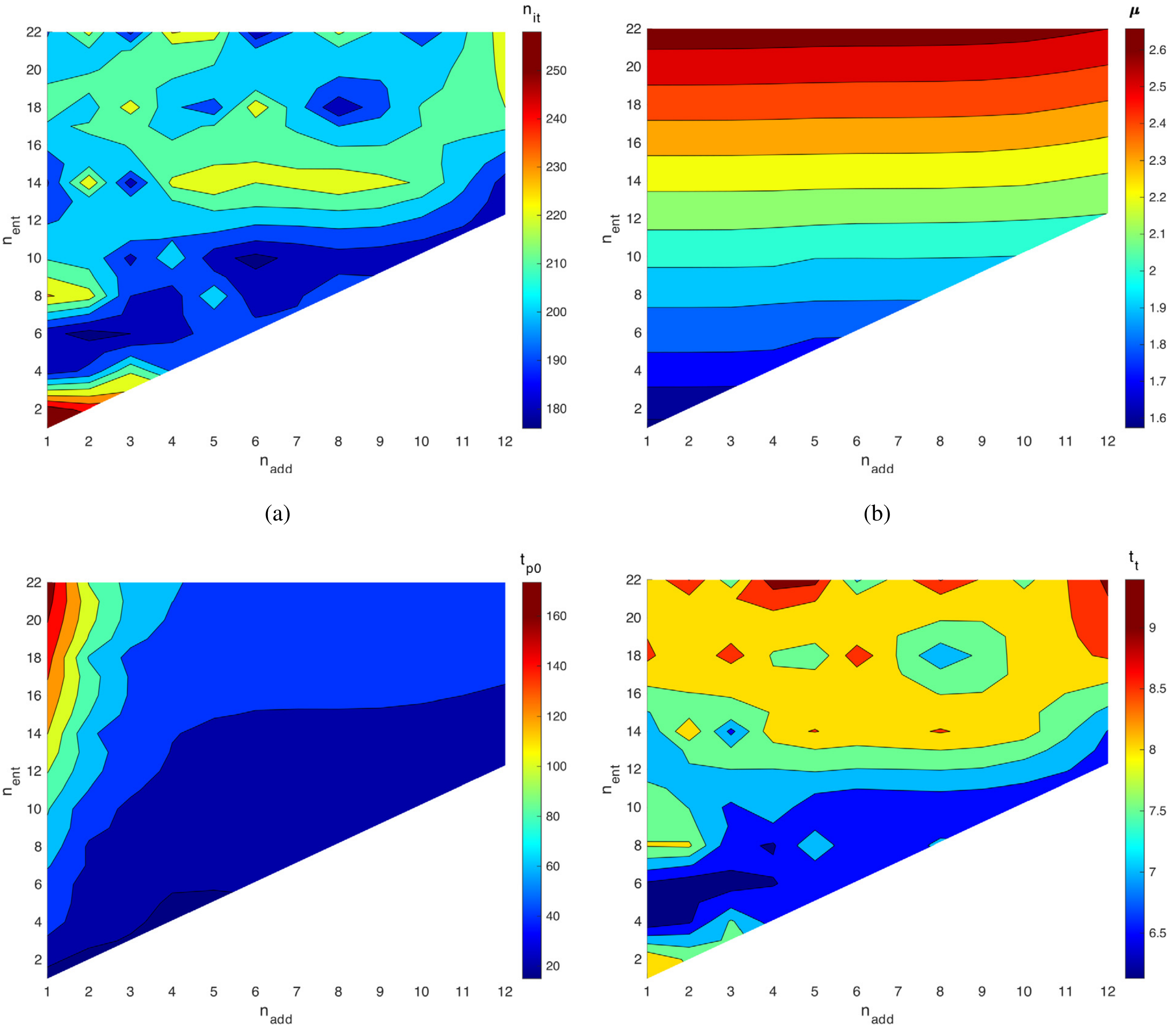
Table 4
Test 1: Numerical performance of the static technique.

#	Pat	Filt	τ_{filt}	n_{it}	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0	Base	-	-	356	2.50	0.03	10.94	10.97	1.518
1	A	-	-	224	3.30	0.03	7.28	7.31	1.739
2	B	-	-	187	3.51	0.05	6.42	6.47	1.953
3	C	-	-	306	3.98	0.02	9.40	9.42	1.518
4	D	-	-	222	4.36	0.06	8.14	8.20	2.160
5	E	-	-	189	4.82	0.03	6.18	6.21	1.739
6	E	Post	1.E-3	224	4.18	3.74	6.75	10.49	1.520
7	E	Pre	1.E-2	225	10.36	0.04	7.49	7.53	1.739

around 1 and their distribution is more and more compact as the quality of $\widetilde{G}$ and $\widetilde{F}$, hence $\widetilde{S}$, is improved, as it is expected from Theorem 3.3. Extreme values of the real and imaginary parts for the $\widetilde{S}^{-1}S$ eigenspectrum are reported in Table 3. As seen in Fig. 7, the imaginary parts remain small, while the real interval increases from mesh a to c.

4.2. Test 1: planar reservoir with homogeneous and isotropic hydraulic conductivity

The main results from the application of the EDFA preconditioner in its static variant to Test 1 (homogeneous and isotropic conductivity with a diagonal tensor) are reported in Table 4. Run 0, denoted as *base*, is taken as benchmark for the following considerations since it refers to the performance of the EDFA preconditioner with the original $A_{p\pi}^T$ non-zero pattern. Expanding such initial pattern, by using the predefined connections A, B, C, D and E (Figs. 6a - 6e) is indeed beneficial, as observed in runs 1 to 5. Considering the reduction in the total CPU time, t_t, per time step as evaluation criterion, the best results are achieved by patterns E, B and A. Specifically, the use of pattern E allows to reduce n_{it} by a



(a) (b)

(c) (d)

Fig. 8. Test 1: Sensitivity analysis on the pair (n_{add}, n_{ent}) in terms of iterations to converge (a), preconditioner density (b), time to compute the first stage of the preconditioner (c) and total time per time step (d).

factor 1.88 and t_t by 1.77, while increasing the preconditioner density by only 1.15. Runs 6 and 7 show the results obtained by applying pre- and post-filtration to pattern E. These techniques introduce a further sparsification of the approximate Schur complement, which is expected to decrease the application cost of the EDFA preconditioner at the price of a slight increase in the iteration count. In this case, such a strategy does not appear to pay off, with the performance substantially getting back to Pattern A at a larger set-up cost.

As to the dynamic technique, Fig. 8 shows the results of a sensitivity analysis carried out on the two user-specified parameters n_{ent} and n_{add}, governing the expansion of the initial pattern $\mathcal{Q}^{(m)}_{(0)} = \mathcal{Q}^{(m)}_{A_{p\pi}}$, versus the number of iterations to converge, n_{it}, the preconditioner density, μ, the time to compute the pre-processing stage of the preconditioner, t_{p_0}, and the total CPU time, t_t, per time step. All the possible settings therein allow to accelerate the convergence compared to the base case of Table 4. The most interesting results are located in the blue to light-blue area in Figs. 8a and 8d, characterized by values of n_{ent} between 4 and 12. Such an interval was also confirmed by the outcome of the static technique, in particular runs 1 and 2, where the number of new entries added per column is 6 and 12, respectively, with a very similar overall performance. Fig. 8b and 8c are self-explanatory; the higher n_{ent} the denser the preconditioner, and the lower n_{add} the higher n_{it} and thus t_{p_0}.

It is interesting to provide the dynamic technique with a physical interpretation, so as to visually locate the position of the faces associated with the entries connected by the dynamically-formed optimal patterns. This analysis, whose outcome

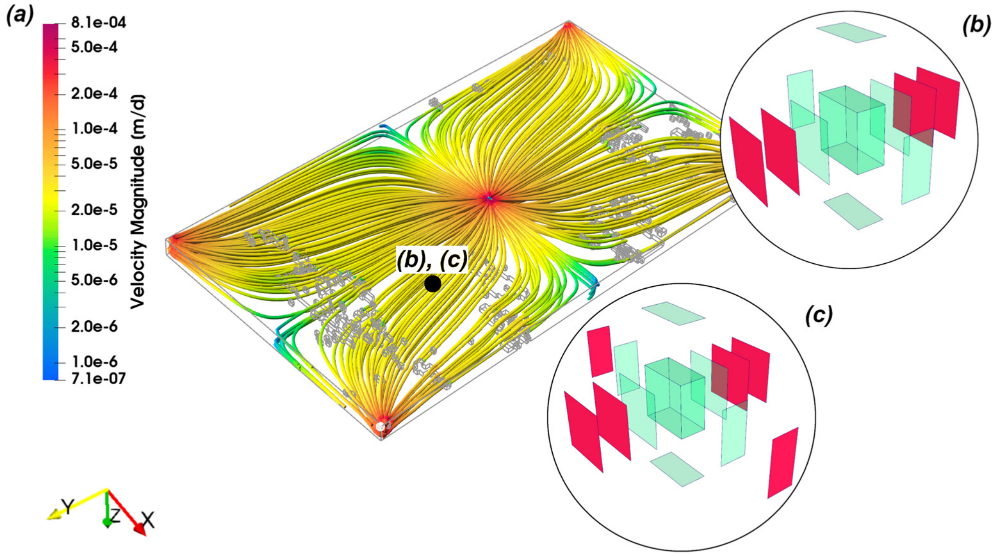


Fig. 9. Test 1: Physical interpretation, based on the flux distribution (a), of two dynamic patterns, obtained with the settings $(1, 4)$ and $(2, 6)$ for the pair $(n_{\text{add}}, n_{\text{ent}})$ (see Fig. 8) in panels (b) and (c), respectively. The red faces represent the extension of the original $A^T_{p\pi}$ pattern in light-green. The resulting patches are uniform throughout the grid.

Table 5
Test 1: Effect of the time step size on the EDFA preconditioner behavior.

Δt [d]	n_{it}	χ_∞	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0.01	3	0.004	35.53	0.04	0.11	0.15	1.749
0.1	6	0.017	-	0.04	0.20	0.24	1.749
1	21	0.099	-	0.04	0.78	0.82	1.749
10	56	5.351	-	0.04	1.87	1.91	1.749
100	127	66.061	-	0.04	4.23	4.27	1.749
1000	199	571.157	-	0.04	6.66	6.70	1.749
Steady	185	-	-	0.04	6.09	6.13	1.749

is offered in Fig. 9, confirms the connection between the static and dynamic strategies, possibly inspiring the selection of better static connections from the dynamic patterns. In particular, we focus on the pairs $(n_{\text{add}}, n_{\text{ent}})$ equal to $(1, 4)$ and $(2, 6)$. Notice that the newly added faces are primarily located following the main flow direction. This is consistent with the physical principle on which the static technique relies. In the scenario of Test 1, in fact, the water flow is essentially horizontal, since the wells penetrate the full thickness of the reservoir, with a principal component along the y axis. These patterns are uniform throughout the grid.

Finally, the effect of the time step size on the EDFA preconditioner in a full transient simulation is assessed in Table 5. The preconditioner is built by means of the dynamic technique, with the setting $(1, 4)$ for the pair $(n_{\text{add}}, n_{\text{ent}})$. The range investigated spans the interval [0.01,1000] days, with the associated χ_∞ parameter up to almost 600. Notice how the number of iterations grows progressively as Δt increases, where the steady state condition can be regarded approximately as an upper limit. This is consistent with Remark 3.7. The reason for this behavior can be also understood from the structure of A_{pp} (equation (A.4)), where the diagonal entries depend on the inverse of Δt. When Δt is small, A_{pp} becomes diagonally dominant and prevails over the other contributions in $\widetilde{S}$.

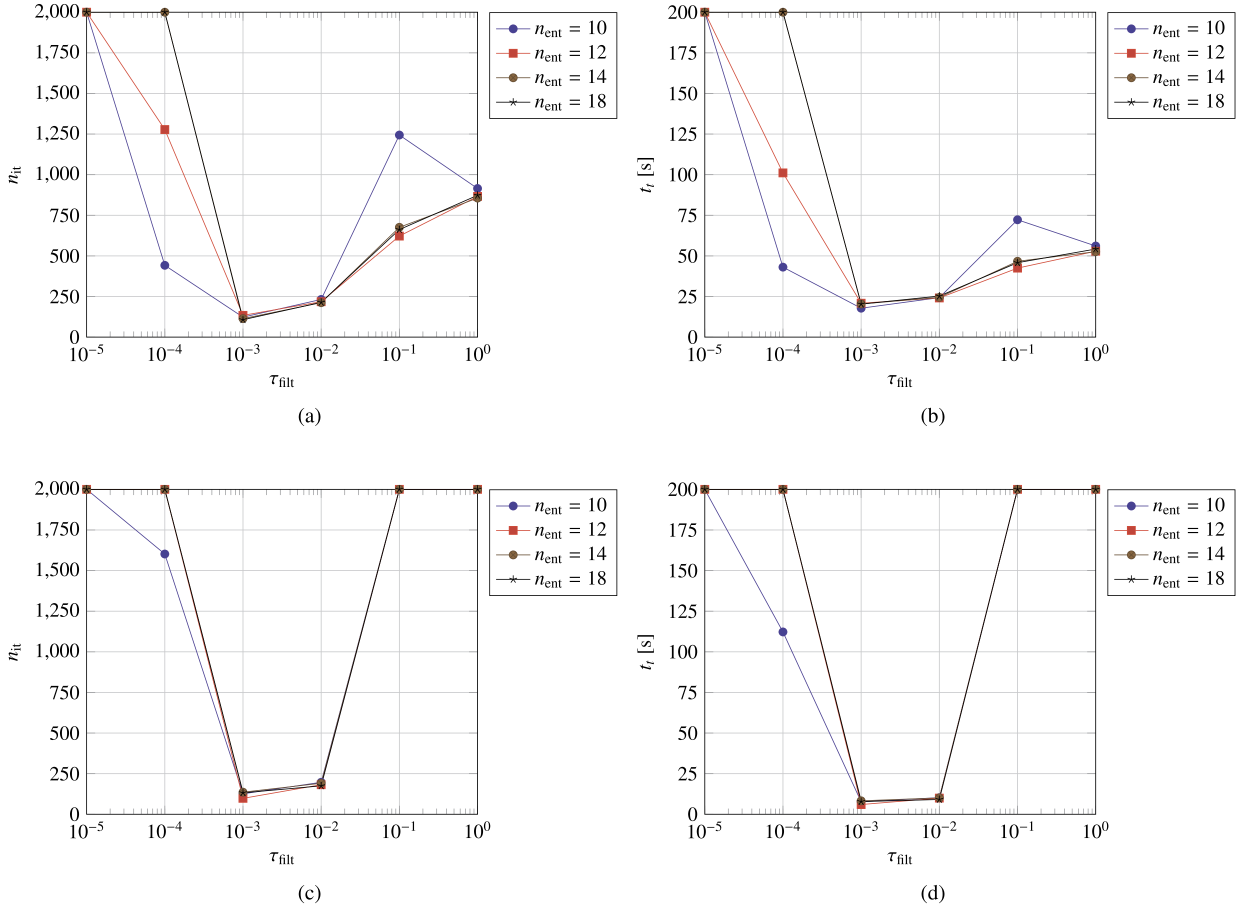
4.3. Test 2: dome reservoir with homogeneous and isotropic hydraulic conductivity

The evolution from a Cartesian to a non-Cartesian grid introduces new challenges for the design of efficient non-zero patterns due to the modification of the native stencils of blocks $A_{\pi\pi}$ and, most of all, $A^T_{p\pi}$, which is accompanied by the increase in the non-zeros number of those two blocks, as shown in Table 1. Specifically, $A^T_{p\pi}$ is 1.9 times denser than with the planar mesh. In fact, the face-to-element connection building the non-zero pattern of the typical column of $A^T_{p\pi}$ moves from Fig. 3a to Fig. 3b.

This modification has a considerable effect on the base case numerical performance (see Table 6), which does not converge after 2,000 iterations. Enlarging the face-to-element connections with patterns A, B and D (runs 1, 2 and 4), however, improves very rapidly the quality of $\widetilde{S}$. By comparing Fig. 3b with 6a, 6b and 6d, the adoption of patterns A, B and D consists of a simultaneous expansion and contraction of $A^T_{p\pi}$'s column non-zero pattern, since at most 24 entries are dis-

Table 6
Test 2: Numerical performance of the static technique. The expression NC means that the solver did not converge within 2,000 iterations.

#	Pat	Filt	τ_{filt}	n_{it}	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0	Base	-	-	NC	5.37	0.29	-	-	2.112
1	A	-	-	896	4.10	0.30	58.91	59.21	2.177
2	B	-	-	118	4.28	0.62	8.64	9.26	2.734
3	C	-	-	NC	6.00	0.29	-	-	2.112
4	D	-	-	136	4.48	1.04	11.24	12.28	3.372
5	E	-	-	NC	7.78	1.36	-	-	3.707
6	B	Post	1.E-7	160	4.03	10.59	10.45	21.04	2.071
7	B	Pre	1.E-3	117	7.51	0.66	8.77	9.43	2.734



(a) (b)

(c) (d)

Fig. 10. Test 2: Sensitivity analysis on the post-filtration tolerance τ_{filt} for different values of n_{ent}. n_{add} is kept constant and equal to 4. Post-filtration is applied to $\widetilde{S}$, (a, b), and $\widetilde{H}$, (c, d), and the results are expressed in terms of number of iterations to converge, n_{it}, (a, c) and total solution time per time step, t_t, (b, d). The maximum number of iterations of Bi-CGStab is set equal to 2,000. The best result, i.e., $n_{\text{it}} = 98$ and t_t=6.00 s, is obtained by setting n_{ent}=12 and performing post-filtration on $\widetilde{H}$ with τ_{filt} =1.E-3 (c, d).

carded and others are added in a variable number. It is a sort of implicit moderate expansion accompanied by a significant filtration. Applying pre- or post-filtration when adopting pattern B, which means in practice sparsifying for the second time the relevant pattern, is not beneficial as proved in runs 6 and 7. Notice that the overall preconditioner density is generally higher than in Test 1 (see Table 4 for a comparison).

As to the dynamic technique, the same strategy, consisting of expanding and contracting the original $A_{p\pi}^T$ pattern, has been followed. Fig. 10 reports the outcomes of a sensitivity analysis on the post-filtration tolerance for different values of n_{ent}, ranging from 10 to 18 with $n_{\text{add}} = 4$. The results are expressed in terms of iteration count to converge (Fig. 10a) and total solution time per time step (Fig. 10b). The best performance is achieved for τ_{filt} =1.E-3, where the number of iterations to converge is similar to that obtained with pattern B in Table 6. An effective way to limit the post-filtration cost consists of applying it to $\widetilde{H}$ as a pre-processing effort. This strategy is successfully investigated in Figs. 10c and 10d. Notice that, while

Table 7
Test 3: Numerical performance of the static technique.

#	Pat	Filt	τ_{filt}	n_{it}	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0	Base	-	-	395	2.67	0.03	11.57	11.60	1.518
1	A	-	-	279	3.28	0.03	8.75	8.78	1.739
2	B	-	-	315	3.40	0.05	10.43	10.48	1.953
3	C	-	-	445	4.69	0.02	13.13	13.15	1.518
4	D	-	-	303	3.80	0.06	10.59	10.65	2.160
5	E	-	-	290	4.81	0.04	9.19	9.23	1.739
6	A	Post (on $\widetilde{H}$)	1.E-7	267	7.08	0.03	8.78	8.81	1.730
7	E	Post (on $\widetilde{H}$)	1.E-8	267	8.12	0.04	8.77	8.81	1.733
8	A	Pre	1.E-3	266	5.81	0.04	8.50	8.54	1.729
9	E	Pre	1.E-4	260	10.36	0.04	7.78	7.81	1.731

Table 8
Test 3: CPU times and memory requirements for Matlab `backslash` operator and Bi-CGStab preconditioned with: (i) optimal ILU(τ) of $\mathcal{A}$, (ii) B_{diag}, and (iii) B_{inv}. Due to memory limitation the test with the direct solver has been carried out on a different platform equipped with an Intel®Xeon™CPU E5-1620 v4 processor at 3.5 GHz with 64 GB of RAM.

Method	n_{it}	t_p [s]	t_s [s]	t_t [s]	Memory peak [GB]	μ
Matlab \	-	-	205.01	205.01	23.68	-
ILU(τ)	113	248.25	4.11	252.36	-	2.730
B_{diag}	NC	0.06	-	-	-	1.276
B_{inv}	316	2.70	9.15	11.85	-	1.405

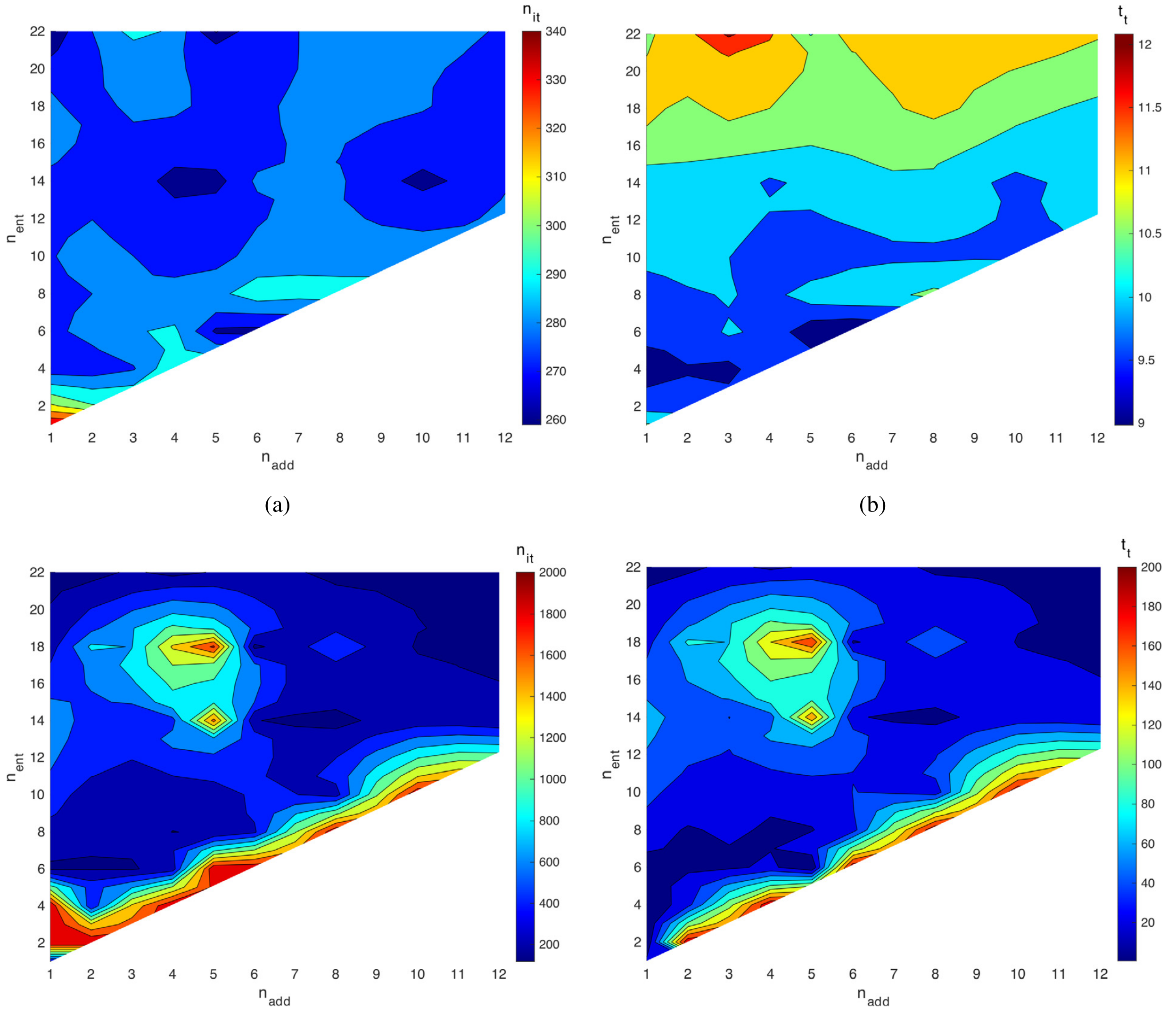
the number of iterations remains approximately the same or is even improved, the CPU time per time step is halved, thus making the dynamically-formed preconditioner competitive to the best statically-derived one (run 2 in Table 6).

4.4. Test 3: planar reservoir with heterogeneous and anisotropic hydraulic conductivity

Introducing a heterogeneous and anisotropic conductivity field in the planar reservoir application worsens the conditioning properties of the associated problem, as shown by the increase in the number of iterations of the base case (Table 7) with respect to Test 1 (Table 4). Like in Test 1, patterns A and E (runs 1, 5) provide improved results that can slightly benefit from further sparsification of the approximate Schur complement. Specifically, pre- and post-filtration are characterized barely by the same performance (runs 6 to 9), with pre-filtration a little more efficient in this application.

A sensitivity analysis on the dynamic parameters n_{ent} and n_{add} is shown in Figs. 11a and 11b in terms of number of iterations to converge, n_{it}, and total solution time per time step, t_t, respectively. All the analyzed combinations succeed in accelerating the convergence with respect to the base case (run 0 in Table 7). The most efficient settings are located in the bottom left portion of graph 11b in the interval $2 \leq n_{\text{ent}} \leq 8$, where t_t is minimized. Specifically, the fastest convergence is achieved with the setting $(6, 6)$ for the pair $(n_{\text{add}}, n_{\text{ent}})$, where $n_{\text{it}} = 267$ and $t_t = 8.98$ s. In the same figure notice how t_t grows with n_{ent}, depending only partially on n_{add}.

To conclude this analysis, we compare the performance obtained by the proposed preconditioner and other available algorithms. Since no specific methods have been developed yet for the non-symmetric block problem considered herein, we start considering general-purpose popular tools, such as Matlab `backslash` operator and Bi-CGStab accelerated by a global preconditioner, such as a threshold-based ILU(τ) of a properly reordered $\mathcal{A}$, to benchmark the performance of the proposed EDFA preconditioner in the steady state case. The relevant outcomes are conveyed in Table 8, where the memory peak reached during the solving phase with Matlab `backslash` replaces the preconditioner density as a measure of the solver memory footprint. The solution time t_t, as well as the memory requirements, are by far higher than those obtained with the EDFA preconditioner (see Table 7 and Figs. 11a and 11b for reference). A more effective strategy may rely on using an iterative approach with the same block preconditioning framework (25), where the approximate Schur complement $\widetilde{S}$ is computed differently from the EDFA approach. A standard default choice, labeled B_{diag}, is using the inverse of the diagonal of $A_{\pi\pi}$, i.e., $\widetilde{S} = A_{pp} - A_{p\pi}\text{diag}(A_{\pi\pi})^{-1}A_{\pi p}$. This option, however, is completely ineffectual since Bi-CGStab does not converge at all. An alternative, and more advanced, option relies on using an explicit approximate inverse, $\overline{A}_{\pi\pi}^{-1}$, such as the one employed in [73,102], to compute $\widetilde{S} = A_{pp} - A_{p\pi}\overline{A}_{\pi\pi}^{-1}A_{\pi p}$. This choice, labeled B_{inv}, actually approaches the EDFA performance (Table 8), though with a limited flexibility, thus supporting the generality and robustness of the preconditioning framework (25).



(a) (b)

(c) (d)

Fig. 11. Test 3 and 4: Sensitivity analysis on the pair $(n_{\text{add}}, n_{\text{ent}})$ for Test 3 (a), (b) and Test 4 (c), (d). The outcome is expressed in terms of iterations to converge (a), (c) and total time per time step (b), (d).

4.5. *Test 4: dome reservoir with full tensor heterogeneous and isotropic hydraulic conductivity*

In this final application, the dome-structured reservoir is characterized by a full-tensor heterogeneous conductivity field, which is obtained by rotating the element local axes x and y, following the curvature of the domain [30]:

$$\hat{K}^E = R^E_{xy} K^E R^{E,T}_{xy}, \quad E = 1, \ldots, N_e, \tag{59}$$

where K^E is the element diagonal conductivity matrix and $R^E_{xy} = R_y(-\theta^E_y) R_x(-\theta^E_x)$ is the overall rotation matrix. θ^E_x and θ^E_y are the average rotations of the domain surface, within element E, around the global Cartesian reference system as per the right-hand rule.

The main results for the static strategy are provided in Table 9. The base case performance (run 0) is not satisfactory and the only pattern yielding an appreciable acceleration is D (runs 1-5). Moreover, in two cases the solver has not converged yet after 2,000 iterations. Filtration here appears to be mandatory and effective, as shown by runs 6 and 7, where a significant post-filtration (τ_{filt} =1.E-2) on factor $\widetilde{H}$ has been applied to patterns D and B. This confirms that the native patterns seem to involve too many connections, which turn out to be not significant to capture the complicated flux nature of this test case. On the other hand, pre-filtration (runs 8 and 9) does not represent a consistent alternative, due to the larger preconditioner density, which leads to an increase in the application costs even for a smaller number of iterations.

Table 9
Test 4: Numerical performance of the static technique.

#	Pat	Filt	τ_{filt}	n_{it}	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0	Base	-	-	667	5.24	0.30	42.30	42.60	2.112
1	A	-	-	NC	4.03	0.32	-	-	2.177
2	B	-	-	521	4.35	0.62	38.50	39.12	2.734
3	C	-	-	667	6.07	0.30	43.10	43.40	2.112
4	D	-	-	386	4.56	1.05	31.76	32.81	3.372
5	E	-	-	NC	7.72	1.37	-	-	3.707
6	D	Post (on $\widetilde{H}$)	1.E-2	231	17.40	0.04	11.79	11.83	1.214
7	B	Post (on $\widetilde{H}$)	1.E-2	233	14.24	0.04	12.35	12.39	1.216
8	D	Pre	1.E-3	217	8.65	1.05	17.96	19.01	3.195
9	B	Pre	1.E-3	424	7.49	0.63	31.68	32.31	2.717

Table 10
Test 4: Pre- and post-filtration on the dynamically-formed preconditioner. Run 0 is obtained with: $n_{\text{ent}} = 6$ and $n_{\text{add}} = 1$.

#	Filt	τ_{filt}	n_{iter}	t_{p_0} [s]	t_p [s]	t_s [s]	t_t [s]	μ
0	-	-	172	69.39	1.13	14.04	15.17	3.072
1	Post	1.E-3	158	70.31	12.47	9.63	22.10	1.395
2	Post (on $\widetilde{H}$)	1.E-3	160	79.21	0.13	9.33	9.45	1.552
3	Pre	1.E-8	149	73.38	1.12	12.09	13.21	3.072

As to the dynamic technique, the sensitivity analysis on n_{ent} and n_{add} in Figs. 11c and 11d reveals that there exists a wide blue area characterized by competitive settings, both in terms of number of iterations and total solving time per step. In this regard, the most attractive portion of the graphs remains the bottom left one. Table 10 aims at assessing the effect of filtration on the dynamically-formed preconditioner obtained with the settings $n_{\text{ent}} = 6$ and $n_{\text{add}} = 1$. The outcomes of post-filtration, applied to both $\widetilde{S}$ and $\widetilde{H}$ in runs 1 and 2 respectively, are here reported to emphasize the greater efficiency of the latter strategy. Although performing post-filtration on $\widetilde{H}$ gives a higher density preconditioner, the filtration is anticipated in the pre-processing stage of the preconditioner set-up, while preserving the quality in the approximation of $\widetilde{S}$. Post-filtration on $\widetilde{H}$ (run 2), in fact, allows to approximately halve the density of the original preconditioner (run 0) to the benefit from a 38% reduction in the total solution time. Pre-filtration, on the other hand, seems not to be effective (run 3). Notice that, in this application, combinations where $n_{\text{ent}} = n_{\text{add}}$, i.e., all the prescribed new entries are subsumed at once, give a bad quality $\widetilde{S}$, hence at least two steps of the dynamic procedure are recommended.

Despite the high density, the unfiltered dynamically-formed preconditioner in Table 10 is by far more competitive than the corresponding static alternatives in Table 9 (runs 0-5). An explanation comes from the analysis of the dynamic pattern representation vs. the position within the dome-grid and the flux distribution, as provided in Fig. 12. The dynamic technique is capable to flexibly catch and exploit the possibly complex physics of the fluxes behind the problem. Due to the dome structure of the domain and the heterogeneity of the hydraulic conductivity, the fluid fluxes are not uniformly distributed in the domain. As an effect, the resulting pattern is not the same throughout the grid and can be hardly guessed, hence a statically designed homogeneous pattern is not well-suited to the specific requirements of this application.

In conclusion, the performance of the EDFA preconditioner is evaluated in a full-transient simulation reproducing the exploitation of a reservoir, initially undisturbed, under the conditions mentioned in Section 4, i.e., four injectors (one at each corner) and a producer (in the center) operating at a constant pressure. The overall computational times are displayed in Fig. 13, where a comparison among a selection of four variants of the EDFA preconditioner is proposed. The performance of the basic setting (V_0, run 0 in Table 9) is used as benchmark to test the advantage provided by the most efficient dynamic (V_1 and V_2, runs 0 and 2 in Table 10) and static strategies (V_3, run 6 in Table 9). The simulated time interval is 130 days with 200 time steps. The other relevant settings are: $\Delta t_{\text{max}} = 5$ d, $\Delta p_T = 5$ bar, $\Delta t_{\text{mult}} = 1.1$. The average CFL number, $\overline{\chi_\infty}$, recorded during the simulation, is 13.54. The best performance is given by variant V_2, which proved to be the most competitive one also in the previous steady state analysis. Nevertheless, the difference with V_1 and V_3 is not as significant as the steady state analysis might have suggested. In any case, all trials, i.e., V_1, V_2 and V_3, clearly outperform V_0, reducing the total time to about one fourth.

5. Discussion

Achieving a fast and cheap solution to the sequence of systems (18), that stem from a full MHFE-FV single-phase flow simulation, is the main objective of this paper. Given the peculiar properties of system (18), which is characterized by a non-symmetric generalized saddle-point structure and usually ill-conditioned blocks, an original preconditioning strategy, denoted as EDFA, has been specifically designed to accelerate the convergence of Krylov subspace solvers. The key features of the proposed approach are twofold: (i) the exploitation of the decoupling factors, G and F, of the system matrix $\mathcal{A}$ block

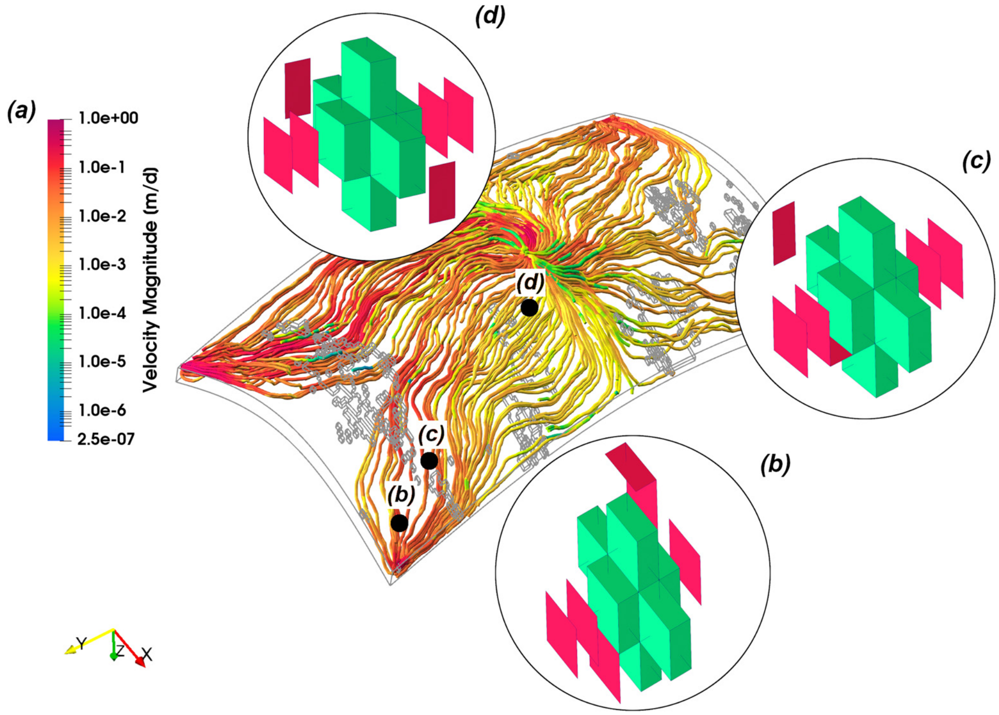
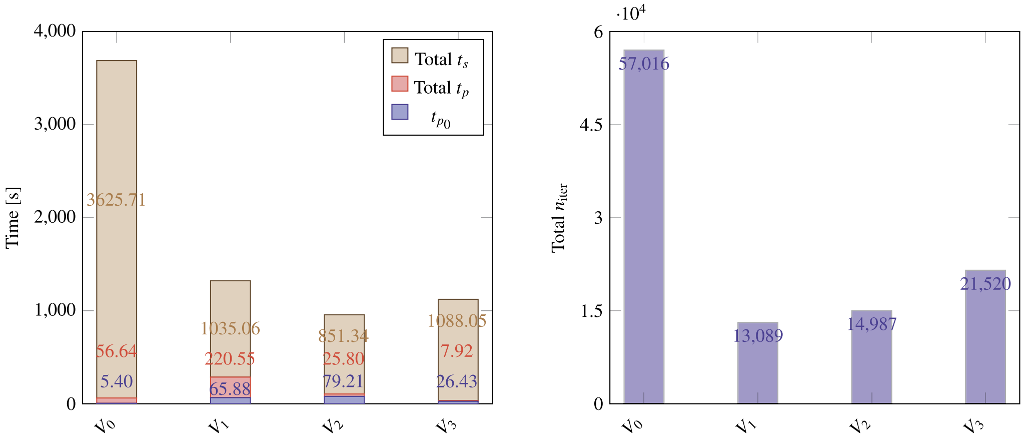


Fig. 12. Test 4: Physical interpretation, based on the flux distribution (a), of the dynamic pattern, obtained with the settings $n_{\text{ent}} = 6$ and $n_{\text{add}} = 1$ at three different locations. The red faces represent the extension of the original $A_{p\pi}^T$ pattern in green.



(a) (b)

Fig. 13. Analyzing the performance of four versions of the EDFA preconditioner in a full-transient simulation, where $\overline{\chi_\infty} = 13.54$. The outcomes are expressed in terms of overall solution time (a), broken down into its basic components, and total number of iterations (b). The preconditioners are obtained with the following settings: $V_0 \rightarrow$ Base pattern (run 0 in Table 9), $V_1 \rightarrow$ Dynamic strategy with $n_{\text{ent}} = 6$, $n_{\text{add}} = 1$ (run 0 in Table 10), $V_2 \rightarrow$ like V_1 with post-filtration on $\widetilde{H}$ and $\tau_{\text{filt}} =$ 1.E-3 (run 2 in Table 10) and $V_3 \rightarrow$ Static strategy with pattern D, post-filtration on $\widetilde{H}$ and $\tau_{\text{filt}} =$1.E-2 (run 6 in Table 9).

$\mathcal{LDU}$ decomposition to recast the Schur complement avoiding the inversion of the leading block (equation (22)); (ii) the inexact computation of these factors by solving two independent sets of MRHS systems (26), by means of a combination of restriction and prolongation operators based on the non-zero pattern of $\widetilde{G}$ and $\widetilde{F}$. Such an approximation turns out to be optimal, with respect to $A_{\pi\pi}$-inner product, for the selected non-zero pattern. The ability to recognize G and F's most representative entries is key to obtain high-quality approximations at a workable sparsity and achieve a fast convergence. This is the motivation behind the introduction of two strategies, namely static and dynamic, aimed at selecting the sparsity pattern of the two decoupling factors by customizing the face-to-element connections contained in the columns of $A_{p\pi}^T$, depending on the properties of the problem at hand.

Table 11
Tests 1-4: Best EDFA performance. The column t_t' reports the total solving time of the base case (run 0).

Test	Variant	Description	Filt	τ_{filt}	n_{iter}	t_t [s]	t_t' [s]	μ
1	Dynamic	$n_{\text{ent}} = 6$, $n_{\text{add}} = 4$	-	-	174	5.83	10.97	1.855
2	Dynamic	$n_{\text{ent}} = 12$, $n_{\text{add}} = 4$	Post (on $\widetilde{H}$)	1.E-3	98	6.00	NC	1.671
3	Static	Pattern E	Pre	1.E-4	260	7.81	11.60	1.731
4	Dynamic	$n_{\text{ent}} = 6$, $n_{\text{add}} = 1$	Post (on $\widetilde{H}$)	1.E-3	160	9.45	42.90	1.552

A physics-based concept underlies the static technique, since the entries of the generic column of the decoupling factors correspond to the face pressure unknowns, which result from the fluid flow arising in a certain small and compact partition of the physical domain. Expanding such a partition to incorporate the most relevant entries is a way for enlarging the native sparsity pattern effectively. The size of the augmented partition and the location of the faces associated with the newly-added entries give rise to a number of possible combinations. As an example, in this paper, we considered five general-purpose prototypes (Figs. 6a-6e). This strategy turns out to be effective when the modeler has a robust idea of the flux distribution, and the resulting pattern can be more or less uniformly extended throughout the domain (compare, for instance, Figs. 9 and 12).

On the other hand, a fully algebraic framework is introduced to define a dynamic variant, where n_{ent} new entries to the initial pattern are progressively added at the locations corresponding to the largest components of the prolonged residual (equation (38)). Compared to the static variant, the dynamic technique is computationally more demanding because, at every step of the pattern construction, a static solution is needed. However, this cost can be easily amortized during a transient simulation and take advantage of an almost ideal parallel degree. Furthermore, this technique is more flexible, since it is capable to implicitly capture the physics behind the problem without the modeler being aware of it and might be used in a black-box fashion as well. Pre- and post-filtration techniques have been introduced with the twofold purpose of controlling the density of $\widetilde{S}$, and improving the quality of its inexact inverse application by removing possibly detrimental near-zero entries.

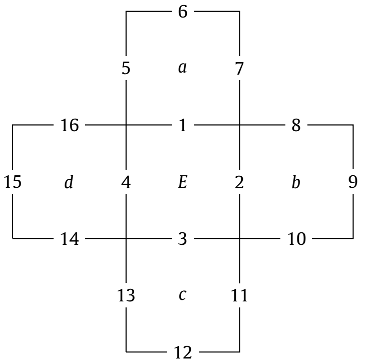
The extensive experimental phase of Section 4 helped understand the EDFA preconditioner potential in different settings, according to the grid type and hydraulic properties, but also suggested some indications about default optimal settings. The EDFA performance with the most efficient set-up is summarized in Table 11 for Tests 1-4 and compared to the base case (run 0) scenario. All the tests revealed that less than 12 new entries are actually needed in addition to the original column patterns, but it is the grid type, i.e., whether Cartesian or non-Cartesian, that mostly influences the optimal EDFA preconditioner set-up. With a Cartesian grid, the suggested static patterns in Fig. 6 seem to be appropriate, and especially E and A, as shown in Tables 4 and 7. By comparing Figs. 6a and 6e, notice that the physical structure of these patterns is similar as the elements involved in their definition are the same. By distinction, optimal results have been obtained for the dynamic variant by setting n_{ent} between 4 and 10, with $n_{\text{add}} \approx n_{\text{ent}}$. Filtration is not strictly necessary, even though some good results were obtained with pre-filtration and τ_{filt} between 1.E-4 and 1.E-3 (Table 7). Conversely, with a non-Cartesian grid, patterns B and D turned out to be the winning choice for the static technique, whereas, for the dynamic one, it is advisable to set $n_{\text{add}} < n_{\text{ent}} < 12$ (Fig. 11d). A significant post-filtration with 1.E-3 $\leq \tau_{\text{filt}} \leq$ 1.E-2 proved to be effective to accelerate, or even to allow for, convergence (see, for instance, Tables 6 and 9). Post-filtration on $\widetilde{H}$, rather than on $\widetilde{S}$, should be preferred.

Finally, Table 5 and Fig. 13 showed that the preconditioner set-up for steady-state conditions plays the role of a worst-case scenario, thus confirming the analytical result in Remark 3.7. In particular, the smaller Δt, the better the accuracy of $\widetilde{S}$ (equation (22)). This observation suggests a different set-up strategy of the preconditioner because building $\widetilde{F}$ and $\widetilde{G}$ with the optimal settings at steady state might be too conservative. The two decoupling factors can be approximated more than once during a full-transient simulation since usually Δt increases as steady state is approached, and cheaper approximations can be effective at the initial stages.

6. Conclusions

In this paper, we introduce a novel preconditioning technique, denoted as EDFA, to solve the sequence of non-symmetric block linear systems arising from the original MHFE-FV discretization of flow problems in porous media developed in [22]. The proposed method is based on the approximation of the decoupling factors of the system matrix by using appropriate restriction operators for the sake of the Schur complement computation. The experimental phase proved its robustness and reliability in different settings, depending on the structure of the grid and the properties of the hydraulic conductivity tensor. The EDFA preconditioning strategy exhibits several attractive features, in particular:

1. Since the process for building the preconditioner is based on the solution of the sets of independent MRHS systems (26), the set-up stage is inherently parallel and can fully exploit the architecture of modern computing platforms;
2. The preconditioner set-up can be split in a two-step procedure, where the first stage is performed only at the beginning of a full-transient simulation and the second stage at each time step;



$S_E = \{a, b, c, d\}$

$F_E = \{1, 2, 3, 4\}$

$P_{E_1} = \{2, 3, 4\}$

$R_{E'} = \{5, 6, \dots, 15, 16\}$

Fig. A.14. Two-dimensional sketch of the element/face connections with an example of the basic sets S_E, F_E, P_{E_i} and $R_{E'}$.

3. The largest computational cost, associated with the approximation of G and F, is concentrated in the first stage, so it can be effectively amortized during a full simulation.

Both the static and dynamic variants proved to be overall efficient. A winner does not stand out clearly, even though it might be better to rely on the dynamic technique when the flux distribution is highly variable throughout the domain and it is hard to define a uniform static pattern prototype. All the same, it is appreciable the relatively easy setting up and cheapness of the static variant and the flexibility of the dynamic.

Research is currently ongoing to develop a fully parallel C++ implementation of the proposed solver and extend the formulation to multi-phase MHFE-FV reservoir models.

CRediT authorship contribution statement

Stefano Nardean: Conceptualization, Software, Writing – original draft. **Massimiliano Ferronato:** Methodology, Supervision, Writing – review & editing. **Ahmad S. Abushaikha:** Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

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

Acknowledgement

This publication was supported by the National Priorities Research Program grant NPRP10-0208-170407 from Qatar National Research Fund. The authors are grateful to two anonymous reviewers whose thoughtful comments helped improve the presentation.

Appendix A. MHFE-FV matrices

In this appendix, the expressions for the four sub-matrices of system (18) are provided based on the definitions (13), (15), (16) and (17). Let us consider the element E of the grid, the set of its faces, F_E, and neighbors, S_E, where E' is an element of S_E. P_{E_i} denotes the set of faces of E without the one, say i, shared with E', i.e., $F_E - F_E \cap F_{E'} = F_E - i$. By extension, $P_{E'_i} = F_{E'} - i$. $R_{E'}$, formally defined as $\cup_{i \in F_E} P_{E'_i}$, is the set of faces of the elements in S_E not shared with E. Fig. A.14 provides a graphical interpretation of the aforementioned sets in a 2-D setting. Let also $g : k \rightarrow p = g(k)$ be the function that converts the global matrix index k into the local one p. The expressions for the $\mathcal{A}$ sub-blocks read:

$$[A_{\pi\pi}]_{ij}: \quad \forall i \in \{1, 2, \dots, N_f\} \quad \begin{array}{ll} \text{if } i = j & -B^{E^{-1}}_{g(i)g(i)} - B^{E'^{-1}}_{g(i)g(i)}, \\ \text{if } j \in P_{E_i} & -B^{E^{-1}}_{g(i)g(j)}, \\ \text{if } j \in P_{E'_i} & -B^{E'^{-1}}_{g(i)g(j)}, \end{array} \tag{A.1}$$

$$[A_{\pi p}]_{ij}: \quad \forall i \in \{1, 2, \dots, N_f\} \quad \text{if } j \in \left\{E, E'\right\} \quad \sum_{k=1}^{N_f^E} B^{j^{-1}}_{g(i)k}, \tag{A.2}$$

$$[A_{p\pi}]_{Ej}: \quad \forall E \in \{1,2,\dots,N_e\} \quad \begin{array}{lll} \text{if } j \in F_E & -\sum_{k\in K_E} B^{E^{-1}}_{g(k)g(j)} \dfrac{B^{E'^{-1}}_{g(k)g(k)}}{B^{E'^{-1}}_{g(k)g(k)} + B^{E^{-1}}_{g(k)g(k)}}, & \text{where } K_E = F_E - j, \\ \text{if } j \in R_{E'} & B^{E'^{-1}}_{g(i)g(j)} \dfrac{B^{E^{-1}}_{g(i)g(i)}}{B^{E'^{-1}}_{g(i)g(i)} + B^{E^{-1}}_{g(i)g(i)}}, & \text{where } i = F_E \cap F_{E'}, \end{array} \tag{A.3}$$

$$[A_{pp}]_{El}: \quad \forall E \in \{1,2,\dots,N_e\} \quad \begin{array}{lll} \text{if } E = l & \sum_{i\in F_E} \dfrac{B^{E'^{-1}}_{g(i)g(i)}}{B^{E'^{-1}}_{g(i)g(i)} + B^{E}_{g(i)g(i)}} \sum_{j\in F_E} B^{E^{-1}}_{g(i)g(j)} + \dfrac{\Omega^E \overline{c}^E}{\Delta t_n}, & \\ \text{if } l \in S_E & -\dfrac{B^{E^{-1}}_{g(i)g(i)}}{B^{l^{-1}}_{g(i)g(i)} + B^{E^{-1}}_{g(i)g(i)}} \sum_{j\in F_l} B^{l^{-1}}_{g(i)g(j)}, & \text{where } i = F_E \cap F_l. \end{array} \tag{A.4}$$

References

[1] A.S. Abd, A.S. Abushaikha, On the performance of the node control volume finite element method for modeling multi-phase fluid flow in heterogeneous porous media, Transp. Porous Media 135 (2) (2020) 409–429, https://doi.org/10.1007/s11242-020-01481-2.

[2] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag New York, New York, NY, 1991.

[3] A. Younes, P. Ackerer, F. Delay, Mixed finite elements for solving 2-D diffusion-type equations, Rev. Geophys. 48 (1) (2010) RG1004, https://doi.org/10.1029/2008RG000277.

[4] R. Mosé, P. Siegel, P. Ackerer, G. Chavent, Application of the mixed hybrid finite element approximation in a groundwater flow model: luxury or necessity?, Water Resour. Res. 30 (11) (1994) 3001–3012, https://doi.org/10.1029/94WR01786.

[5] G. Chavent, J. Roberts, A unified physical presentation of mixed, mixed-hybrid finite elements and standard finite difference approximations for the determination of velocities in waterflow problems, Adv. Water Resour. 14 (6) (1991) 329–348, https://doi.org/10.1016/0309-1708(91)90020-O.

[6] F.A. Radu, M. Bause, A. Prechtel, S. Attinger, A mixed hybrid finite element discretization scheme for reactive transport in porous media, in: K. Kunisch, G. Of, O. Steinbach (Eds.), Numerical Mathematics and Advanced Applications, Springer, Berlin, Heidelberg, 2008, pp. 513–520.

[7] R.N. Yoisell, O.F. Cristiane, F.L. Abimael, M.M. Sandra, A mixed-hybrid finite element method applied to tracer injection processes, Int. J. Model. Simul. Pet. Ind. 6 (1) (2012) 51–59.

[8] F. Brunner, F.A. Radu, P. Knabner, Analysis of an upwind-mixed hybrid finite element method for transport problems, SIAM J. Numer. Anal. 52 (1) (2014) 83–102, https://doi.org/10.1137/130908191.

[9] T. Smejkal, J. Mikyška, R. Fučík, Numerical modelling of adsorption and desorption of water vapor in zeolite 13X using a two-temperature model and mixed-hybrid finite element method numerical solver, Int. J. Heat Mass Transf. 148 (2020) 119050, https://doi.org/10.1016/j.ijheatmasstransfer.2019.119050.

[10] C. Rodrigo, X. Hu, P. Ohm, J. Adler, F. Gaspar, L. Zikatanov, New stabilized discretizations for poroelasticity and the Stokes' equations, Comput. Methods Appl. Mech. Eng. 341 (2018) 467–484, https://doi.org/10.1016/j.cma.2018.07.003.

[11] C. Niu, H. Rui, M. Sun, A coupling of hybrid mixed and continuous Galerkin finite element methods for poroelasticity, Appl. Math. Comput. 347 (2019) 767–784, https://doi.org/10.1016/j.amc.2018.11.021.

[12] C. Niu, H. Rui, X. Hu, A stabilized hybrid mixed finite element method for poroelasticity, Comput. Geosci. 25 (2021), https://doi.org/10.1007/s10596-020-09972-3.

[13] M. Frigo, N. Castelletto, M. Ferronato, J.A. White, Efficient solvers for hybridized three-field mixed finite element coupled poromechanics, Comput. Math. Appl. 91 (2021) 36–52, https://doi.org/10.1016/j.camwa.2020.07.010.

[14] A. Younes, V. Fontaine, Efficiency of mixed hybrid finite element and multipoint flux approximation methods on quadrangular grids and highly anisotropic media, Int. J. Numer. Methods Eng. 76 (3) (2008) 314–336, https://doi.org/10.1002/nme.2327.

[15] M. Bause, P. Knabner, Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods, Adv. Water Resour. 27 (6) (2004) 565–581, https://doi.org/10.1016/j.advwatres.2004.03.005.

[16] B. Belfort, F. Ramasomanana, A. Younes, F. Lehmann, An efficient lumped mixed hybrid finite element formulation for variably saturated groundwater flow, Vadose Zone J. 8 (2) (2009) 352–362, https://doi.org/10.2136/vzj2008.0108.

[17] R. Fučík, J. Mikyška, Mixed-hybrid finite element method for modelling two-phase flow in porous media, J. Math-for-Ind. 3 (2011C-2) (2011) 9–19.

[18] A.S. Abushaikha, M.J. Blunt, O.R. Gosselin, C.C. Pain, M.D. Jackson, Interface control volume finite element method for modelling multi-phase fluid flow in highly heterogeneous and fractured reservoirs, J. Comput. Phys. 298 (2015) 41–61, https://doi.org/10.1016/j.jcp.2015.05.024.

[19] R. Fučík, J. Klinkovský, J. Solovský, T. Oberhuber, J. Mikyška, Multidimensional mixed–hybrid finite element method for compositional two-phase flow in heterogeneous porous media and its parallel implementation on GPU, Comput. Phys. Commun. 238 (2019) 165–180, https://doi.org/10.1016/j.cpc.2018.12.004.

[20] J. Hou, J. Chen, S. Sun, Z. Chen, Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media, J. Comput. Appl. Math. 307 (2016) 262–283, https://doi.org/10.1016/j.cam.2016.01.050.

[21] J. Moortgat, A. Firoozabadi, Mixed-hybrid and vertex-discontinuous-Galerkin finite element modeling of multiphase compositional flow on 3D unstructured grids, J. Comput. Phys. 315 (2016) 476–500, https://doi.org/10.1016/j.jcp.2016.03.054.

[22] A.S. Abushaikha, D.V. Voskov, H.A. Tchelepi, Fully implicit mixed-hybrid finite-element discretization for general purpose subsurface reservoir simulation, J. Comput. Phys. 346 (2017) 514–538, https://doi.org/10.1016/j.jcp.2017.06.034.

[23] H. Hoteit, R. Mosé, B. Philippe, P. Ackerer, J. Erhel, The maximum principle violations of the mixed-hybrid finite-element method applied to diffusion equations, Int. J. Numer. Methods Eng. 55 (12) (2002) 1373–1390, https://doi.org/10.1002/nme.531.

[24] A. Younes, R. Mosé, P. Ackerer, G. Chavent, A new formulation of the mixed finite element method for solving elliptic and parabolic PDE with triangular elements, J. Comput. Phys. 149 (1) (1999) 148–167, https://doi.org/10.1006/jcph.1998.6150.

[25] A. Younes, P. Ackerer, G. Chavent, From mixed finite elements to finite volumes for elliptic PDEs in two and three dimensions, Int. J. Numer. Methods Eng. 59 (3) (2004) 365–388, https://doi.org/10.1002/nme.874.

[26] M.A. Puscas, G. Enchéry, S. Desroziers, Application of the mixed multiscale finite element method to parallel simulations of two-phase flows in porous media, Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles 73 (2018) 38, https://doi.org/10.2516/ogst/2018022.

[27] P. Devloo, W. Teng, C.-S. Zhang, Multiscale hybrid-mixed finite element method for flow simulation in fractured porous media, Comput. Model. Eng. Sci. 119 (1) (2019) 145–163, https://doi.org/10.32604/cmes.2019.04812.

[28] K. Lipnikov, G. Manzini, M. Shashkov, Mimetic finite difference method, J. Comput. Phys. 257 (2014) 1163–1227, https://doi.org/10.1016/j.jcp.2013.07.031.

[29] K. Lipnikov, G. Manzini, J.D. Moulton, M. Shashkov, The mimetic finite difference method for elliptic and parabolic problems with a staggered discretization of diffusion coefficient, J. Comput. Phys. 305 (2016) 111–126, https://doi.org/10.1016/j.jcp.2015.10.031.

[30] A.S. Abushaikha, K.M. Terekhov, A fully implicit mimetic finite difference scheme for general purpose subsurface reservoir simulation with full tensor permeability, J. Comput. Phys. 406 (2020) 109194, https://doi.org/10.1016/j.jcp.2019.109194.

[31] N. Zhang, A.S. Abushaikha, Fully implicit reservoir simulation using mimetic finite difference method in fractured carbonate reservoirs, in: SPE Reservoir Characterisation and Simulation Conference and Exhibition, SPE, Abu Dhabi, UAE, 2019, SPE–196711–MS.

[32] A.S. Abd, N. Zhang, A.S. Abushaikha, Modeling the effects of capillary pressure with the presence of full tensor permeability and discrete fracture models using the mimetic finite difference method, Transp. Porous Media 137 (3) (2021) 739–767, https://doi.org/10.1007/s11242-021-01585-3.

[33] P.A. Raviart, J.M. Thomas, A mixed finite element method for 2-nd order elliptic problems, in: I. Galligani, E. Magenes (Eds.), Mathematical Aspects of Finite Element Methods, in: Lecture Notes in Mathematics, Springer, Berlin, Heidelberg, 1977, pp. 292–315.

[34] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2003.

[35] J. Wallis, Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, San Francisco, California, 1983, pp. 325–334.

[36] J. Wallis, R. Kendall, T. Little, Constrained residual acceleration of conjugate residual methods, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Dallas, Texas, 1985, pp. 415–428.

[37] H. Cao, H.A. Tchelepi, J.R. Wallis, H.E. Yardumian, Parallel scalable unstructured CPR-type linear solver for reservoir simulation, in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers, Dallas, Texas, 2005, SPE–96809–MS.

[38] D.V. Voskov, O. Volkov, Advanced strategies of forward simulation for adjoint-based optimization, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, The Woodlands, Texas, USA, 2013, SPE–163592–MS.

[39] T.T. Garipov, P. Tomin, R. Rin, D.V. Voskov, H.A. Tchelepi, Unified thermo-compositional-mechanical framework for reservoir simulation, Comput. Geosci. 22 (2018) 1039–1057, https://doi.org/10.1007/s10596-018-9737-5.

[40] M. Khait, Delft advanced research Terra simulator: General purpose reservoir simulator with operator-based linearization, Ph.D. thesis, Delft University of Technology, 2019.

[41] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, series in Edition, CRC Press, Boca Raton, Florida, USA, 1980.

[42] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, R. Tuminaro, A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations, J. Comput. Phys. 227 (3) (2008) 1790–1808, https://doi.org/10.1016/j.jcp.2007.09.026.

[43] K. Wang, H. Liu, J. Luo, Z. Chen, Efficient CPR-type preconditioner and its adaptive strategies for large-scale parallel reservoir simulations, J. Comput. Appl. Math. 328 (2018) 443–468, https://doi.org/10.1016/j.cam.2017.07.022.

[44] H. Liu, K. Wang, Z. Chen, A family of constrained pressure residual preconditioners for parallel reservoir simulations, Numer. Linear Algebra Appl. 23 (1) (2016) 120–146, https://doi.org/10.1002/nla.2017.

[45] T. Roy, T.B. Jönsthövel, C. Lemon, A.J. Wathen, A Constrained Pressure-Temperature Residual (CPTR) method for non-isothermal multiphase flow in porous media, SIAM J. Sci. Comput. 42 (4) (2020) B1014–B1040, https://doi.org/10.1137/19M1292023.

[46] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2) (2002) 418–477, https://doi.org/10.1006/jcph.2002.7176.

[47] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137, https://doi.org/10.1017/S0962492904000212.

[48] A.J. Wathen, Preconditioning, Acta Numer. 24 (2015) 329–376, https://doi.org/10.1017/S0962492915000021.

[49] A.-L. Yang, G.-F. Zhang, Y.-J. Wu, General constraint preconditioning iteration method for singular saddle-point problems, J. Comput. Appl. Math. 282 (2015) 157–166, https://doi.org/10.1016/j.cam.2014.12.042.

[50] 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, https://doi.org/10.1137/18M1219370.

[51] N. Bootland, A. Bentley, C. Kees, A. Wathen, Preconditioners for two-phase incompressible Navier–Stokes flow, SIAM J. Sci. Comput. 41 (4) (2019) B843–B869, https://doi.org/10.1137/17M1153674.

[52] Y. Wu, X.-C. Cai, A fully implicit domain decomposition based ALE framework for three-dimensional fluid–structure interaction with application in blood flow computation, J. Comput. Phys. 258 (2014) 524–537, https://doi.org/10.1016/j.jcp.2013.10.046.

[53] J. Liu, W. Yang, M. Dong, A.L. Marsden, The nested block preconditioning technique for the incompressible Navier–Stokes equations with emphasis on hemodynamic simulations, Comput. Methods Appl. Mech. Eng. 367 (2020) 113122, https://doi.org/10.1016/j.cma.2020.113122.

[54] O. Axelsson, R. Blaheta, P. Byczanski, Stable discretization of poroelasticity problems and efficient preconditioners for arising saddle point type matrices, Comput. Vis. Sci. 15 (2012) 191–207, https://doi.org/10.1007/s00791-013-0209-0.

[55] N. Castelletto, J.A. White, M. Ferronato, Scalable algorithms for three-field mixed finite element coupled poromechanics, J. Comput. Phys. 327 (2016) 894–918, https://doi.org/10.1016/j.jcp.2016.09.063.

[56] J.A. White, N. Castelletto, H.A. Tchelepi, Block-partitioned solvers for coupled poromechanics: a unified framework, Comput. Methods Appl. Mech. Eng. 303 (2016) 55–74, https://doi.org/10.1016/j.cma.2016.01.008.

[57] P. Chidyagwai, S. Ladenheim, D.B. Szyld, Constraint preconditioning for the coupled Stokes–Darcy system, SIAM J. Sci. Comput. 38 (2) (2016) A668–A690, https://doi.org/10.1137/15M1032156.

[58] F.J. Gaspar, C. Rodrigo, On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics, Comput. Methods Appl. Mech. Eng. 326 (2017) 526–540, https://doi.org/10.1016/j.cma.2017.08.025.

[59] P. Luo, C. Rodrigo, F.J. Gaspar, C.W. Oosterlee, On an Uzawa smoother in multigrid for poroelasticity equations, Numer. Linear Algebra Appl. 24 (1) (2017) e2074, https://doi.org/10.1002/nla.2074.

[60] M. Bean, K. Lipnikov, S.-Y. Yi, A block-diagonal preconditioner for a four-field mixed finite element method for Biot's equations, Appl. Numer. Math. 122 (2017) 1–13, https://doi.org/10.1016/j.apnum.2017.07.007.

[61] N. Castelletto, S. Klevtsov, H. Hajibeygi, H.A. Tchelepi, Multiscale two-stage solver for Biot's poroelasticity equations in subsurface media, Comput. Geosci. 23 (2019) 207–224, https://doi.org/10.1007/s10596-018-9791-z.

[62] J.A. White, N. Castelletto, S. Klevtsov, Q.M. Bui, D. Osei-Kuffuor, H.A. Tchelepi, A two-stage preconditioner for multiphase poromechanics in reservoir simulation, Comput. Methods Appl. Mech. Eng. 357 (2019) 112575, https://doi.org/10.1016/j.cma.2019.112575.

[63] A. Franceschini, N. Castelletto, M. Ferronato, Approximate inverse-based block preconditioners in poroelasticity, Comput. Geosci. 25 (2021), https://doi.org/10.1007/s10596-020-09981-2.

[64] 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, https://doi.org/10.1137/19M1261250.

[65] Q.M. Bui, D. Osei-Kuffuor, N. Castelletto, J.A. White, A scalable multigrid reduction framework for multiphase poromechanics of heterogeneous media, SIAM J. Sci. Comput. 42 (2) (2020) B379–B396, https://doi.org/10.1137/19M1256117.
[66] A. Budiša, W.M. Boon, X. Hu, Mixed-dimensional auxiliary space preconditioners, SIAM J. Sci. Comput. 42 (5) (2020) A3367–A3396, https://doi.org/10.1137/19M1292618.
[67] A. Budiša, X. Hu, Block preconditioners for mixed-dimensional discretization of flow in fractured porous media, Comput. Geosci. 25 (2021), https://doi.org/10.1007/s10596-020-09984-z.
[68] P.F. Antonietti, J. De Ponti, L. Formaggia, A. Scotti, Preconditioning techniques for the numerical solution of flow in fractured porous media, J. Sci. Comput. 86 (2021) 2, https://doi.org/10.1007/s10915-020-01372-0.
[69] E.C. Cyr, J.N. Shadid, R.S. Tuminaro, R.P. Pawlowski, L. Chacón, A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD, SIAM J. Sci. Comput. 35 (3) (2013) B701–B730, https://doi.org/10.1137/12088879X.
[70] E.G. Phillips, J.N. Shadid, E.C. Cyr, H.C. Elman, R.P. Pawlowski, Block preconditioners for stable mixed nodal and edge finite element representations of incompressible resistive MHD, SIAM J. Sci. Comput. 38 (6) (2016) B1009–B1031, https://doi.org/10.1137/16M1074084.
[71] M. Wathen, C. Greif, D. Schötzau, Preconditioners for mixed finite element discretizations of incompressible MHD equations, SIAM J. Sci. Comput. 39 (6) (2017) A2993–A3013, https://doi.org/10.1137/16M1098991.
[72] M. Wathen, C. Greif, A scalable approximate inverse block preconditioner for an incompressible magnetohydrodynamics model problem, SIAM J. Sci. Comput. 42 (1) (2020) B57–B79, https://doi.org/10.1137/19M1255409.
[73] M. Ferronato, C. Janna, G. Gambolati, Mixed constraint preconditioning in computational contact mechanics, Comput. Methods Appl. Mech. Eng. 197 (45–48) (2008) 3922–3931, https://doi.org/10.1016/j.cma.2008.03.008.
[74] R.R. Settgast, P. Fu, S.D. Walsh, J.A. White, C. Annavarapu, F.J. Ryerson, A fully coupled method for massively parallel simulation of hydraulically driven fractures in 3-dimensions, Int. J. Numer. Anal. Methods Geomech. 41 (5) (2017) 627–653, https://doi.org/10.1002/nag.2557.
[75] T.T. Garipov, M. Karimi-Fard, H.A. Tchelepi, Discrete fracture model for coupled flow and geomechanics, Comput. Geosci. 20 (1) (2016) 149–160, https://doi.org/10.1007/s10596-015-9554-z.
[76] A. Franceschini, N. Castelletto, M. Ferronato, Block preconditioning for fault/fracture mechanics saddle-point problems, Comput. Methods Appl. Mech. Eng. 344 (2019) 376–401, https://doi.org/10.1016/j.cma.2018.09.039.
[77] M. Ferronato, A. Franceschini, C. Janna, N. Castelletto, H.A. Tchelepi, A general preconditioning framework for coupled multiphysics problems with application to contact- and poro-mechanics, J. Comput. Phys. 398 (2019) 108887, https://doi.org/10.1016/j.jcp.2019.108887.
[78] M. Cusini, A.A. Lukyanov, J. Natvig, H. Hajibeygi, Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media, J. Comput. Phys. 299 (2015) 472–486, https://doi.org/10.1016/j.jcp.2015.07.019.
[79] M. Cusini, B. Fryer, C. van Kruijsdijk, H. Hajibeygi, Algebraic dynamic multilevel method for compositional flow in heterogeneous porous media, J. Comput. Phys. 354 (2018) 593–612, https://doi.org/10.1016/j.jcp.2017.10.052.
[80] Q.M. Bui, L. Wang, D. Osei-Kuffuor, Algebraic multigrid preconditioners for two-phase flow in porous media with phase transitions, Adv. Water Resour. 114 (2018) 19–28, https://doi.org/10.1016/j.advwatres.2018.01.027.
[81] M.A. Cremon, N. Castelletto, J.A. White, Multi-stage preconditioners for thermal–compositional–reactive flow in porous media, J. Comput. Phys. 418 (2020) 109607, https://doi.org/10.1016/j.jcp.2020.109607.
[82] M. Cusini, C. van Kruijsdijk, H. Hajibeygi, Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media, J. Comput. Phys. 314 (2016) 60–79, https://doi.org/10.1016/j.jcp.2016.03.007.
[83] O. Axelsson, R. Blaheta, Low-rank improvements of two-level grid preconditioned matrices, J. Comput. Appl. Math. 340 (2018) 432–442, https://doi.org/10.1016/j.cam.2017.09.027.
[84] A. Dziekonski, M. Mrozowski, Block conjugate-gradient method with multilevel preconditioning and GPU acceleration for FEM problems in electromagnetics, IEEE Antennas Wirel. Propag. Lett. 17 (6) (2018) 1039–1042, https://doi.org/10.1109/LAWP.2018.2830124.
[85] L. Delpopolo Carciopolo, M. Cusini, L. Formaggia, H. Hajibeygi, Adaptive multilevel space-time-stepping scheme for transport in heterogeneous porous media (ADM-LTS), J. Comput. Phys.: X 6 (2020) 100052, https://doi.org/10.1016/j.jcpx.2020.100052.
[86] I. Perugia, V. Simoncini, Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations, Numer. Linear Algebra Appl. 7 (7–8) (2000) 585–616, https://doi.org/10.1002/1099-1506(200010/12)7:7/8<585::AID-NLA214>3.0.CO;2-F.
[87] J. Maryska, M. Rozlozník, M. Tuma, Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem, SIAM J. Sci. Comput. 22 (2) (2000) 704–723, https://doi.org/10.1137/S1064827598339608.
[88] G. Gambolati, P. Teatini, Geomechanics of subsurface water withdrawal and injection, Water Resour. Res. 51 (6) (2015) 3922–3955, https://doi.org/10.1002/2014WR016841.
[89] S.F. Matringe, R. Juanes, H.A. Tchelepi, Mixed-finite-element and related-control-volume discretizations for reservoir simulation on three-dimensional unstructured grids, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2007, SPE–106117–MS.
[90] J. Maryška, M. Rozložník, M. Tůma, Mixed-hybrid finite element approximation of the potential fluid flow problem, J. Comput. Appl. Math. 63 (1–3) (1995) 383–392, https://doi.org/10.1016/0377-0427(95)00066-6.
[91] P.S. Huyakorn, G.F. Pinder, Computational Methods in Subsurface Flow, Academic Press, New York, USA, 1983.
[92] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method, Volume 1: The Basis, 5th edition, Butterworth-Heinemann, Oxford, England, 2000.
[93] 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. Comput. 13 (2) (1992) 631–644, https://doi.org/10.1137/0913035.
[94] 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, https://doi.org/10.1137/0907058.
[95] L. Bergamaschi, M. Ferronato, G. Gambolati, Mixed constraint preconditioners for the iterative solution of FE coupled consolidation equations, J. Comput. Phys. 227 (23) (2008) 9885–9897, https://doi.org/10.1016/j.jcp.2008.08.002.
[96] M. Ferronato, L. Bergamaschi, G. Gambolati, Performance and robustness of block constraint preconditioners in finite element coupled consolidation problems, Int. J. Numer. Methods Eng. 81 (3) (2010) 381–402, https://doi.org/10.1002/nme.2702.
[97] M. Ferronato, Preconditioning for sparse linear systems at the dawn of the 21st century: history, current developments, and future perspectives, ISRN Appl. Math. 2012 (2012) 127647, https://doi.org/10.5402/2012/127647.
[98] A. Greenbaum, V. Pták, Z. Strakoš, Any nonincreasing convergence curve is possible for GMRES, SIAM J. Matrix Anal. Appl. 17 (3) (1996) 465–469, https://doi.org/10.1137/S0895479894275030.
[99] M. Christie, M. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Houston, Texas, 2001, pp. 308–317.
[100] H. Cao, Development of techniques for general purpose simulators, Phd thesis, Stanford University, 2002.
[101] K. Coats, IMPES stability: the stable step, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, Houston, Texas, 2001, SPE–69225–MS.
[102] M. Ferronato, C. Janna, G. Pini, Parallel solution to ill-conditioned FE geomechanical problems, Int. J. Numer. Anal. Methods Geomech. 36 (4) (2012) 422–437, https://doi.org/10.1002/nag.1012.