Numerical investigation on a block preconditioning strategy to improve the computational efficiency of DFN models

Laura Gazzola*, Massimiliano Ferronato*, Stefano Berrone†, Sandra Pieraccini† and Stefano Scialò†

* University of Padova
Padova, Italy
e-mail: laura.gazzola.1@phd.unipd.it
e-mail: massimiliano.ferronato@unipd.it

† Politecnico di Torino
Torino, Italy
e-mail: stefano.berrone@polito.it
e-mail: sandra.pieraccini@polito.it
e-mail: stefano.scialo@polito.it

Key words: Discrete Fracture Network, Preconditioning

Abstract: *The simulation of underground flow across intricate fracture networks can be addressed by means of discrete fracture network models. The combination of such models with an optimization formulation allows for the use of nonconforming and independent meshes for each fracture. The arising algebraic problem produces a symmetric saddle-point matrix with a rank-deficient leading block. In our work, we investigate the properties of the system to design a block preconditioning strategy to accelerate the iterative solution of the linearized algebraic problem. The matrix is first permuted and then projected in the symmetric positive-definite Schur-complement space. The proposed strategy is tested in applications of increasing size, in order to investigate its capabilities.*

1 INTRODUCTION

The simulation of the flow in highly fractured systems can be particularly demanding from a computational standpoint, because of the size and complexity of the domain and the uncertainty characterizing the rock properties and the fracture geometry.

In this context, discrete fracture network (DFN) models can be used, and are preferred particularly when the presence of fractures has a dominant impact on the fluid flow dynamics. DFN models represent only the fractures as intersecting planar polygons, neglecting the surrounding underground rock formation. Differently from homogenization-based techniques, DFN models provide an explicit representation of the fractures and their properties in a 3D structure, prescribing continuity constraints for the fluid flow along the linear intersections. The number of the fractures and their different size, that can change of orders of magnitude, entail a complex and multi-scale geometry, which is not trivial to address. The problem has been effectively reformulated as a PDE-constrained optimization problem in [1, 2]. The formulation relies on the use of non-conforming discretizations of the single fractures and on the minimization of a functional to couple intersecting planes. Thus, no match between the meshes of the fractures and the traces are required, simplifying the mesh generation process. Moreover, the problem on the entire DFN can be decoupled in several local problems on the fractures with a moderate exchange of data among fractures, being suitable for a massive parallel implementation [2].

The linearized algebraic problem that derives from such a formulation produces a large size symmetric saddle-point matrix with a rank-deficient leading block. In this work, we focus on accelerating the iterative solution of the linear system by introducing effective block preconditioning techniques. In particular, an appropriate permutation of the global matrix is first

https://doi.org/10.4995/YIC2021.2021.12234

performed, in order to avoid a singular leading block. Though the permuted matrix is no longer symmetric, this approach should be better suited for the solution with Krylov subspace methods. Then, the matrix is projected in the symmetric positive-definite Schur complement space of the fluxes along the intersection traces. The properties and the structure of matrix blocks are properly exploited in order to guarantee an efficient parallel implementation. The matrix properties are tested in applications of increasing size to verify pros and cons of the approach.

The manuscript is organized as follows. In section 2 the mathematical problem and the related discrete algebraic form are introduced. In section 3 the preconditioner framework is described. In section 4 numerical results for four problems of increasing size and complexity are analyzed and discussed.

2 PROBLEM STATEMENT

We consider a connected three-dimensional fracture network made by a system of intersected polygonal fractures surrounded by an impervious matrix. The flow occurs only along the fractures and their intersections, called traces. The flow along the fractures is modeled by means of Darcy's law with appropriate boundary conditions. Coupling conditions are imposed on the traces, in order to guarantee the continuity of the solution and the balance of the fluxes. The whole problem can be reformulated as PDE-constrained optimization problem [1]. Introducing an independent mesh on each fracture and trace, the Darcy equation, as well as the optimization problem, can be discretized following the standard finite element method. The result is the following algebraic problem [2]:

$$G^h\mathbf{h} - \alpha B\mathbf{u} + A^T\mathbf{p} = \mathbf{0}, \qquad \text{(energy minimization)} \tag{1a}$$

$$-\alpha B^T\mathbf{h} + G^u\mathbf{u} - C^T\mathbf{p} = \mathbf{0}, \qquad \text{(energy minimization)} \tag{1b}$$

$$A\mathbf{h} - C\mathbf{u} = \mathbf{q}, \qquad \text{(mass balance)} \tag{1c}$$

where $\mathbf{h} \in \mathbb{R}^{n^h}$ is the hydraulic head on the fractures, $\mathbf{u} \in \mathbb{R}^{n^u}$ is the flux on the traces, $\mathbf{p} \in \mathbb{R}^{n^p}$ are Lagrange multipliers and $\mathbf{q} \in \mathbb{R}^{n^p}$ derives from the boundary conditions and the forcing terms. Usually, $n^p = n^h$, while according to the problem n^u can be either larger or smaller than n^h. The coefficient $\alpha \in \mathbb{R}$ is a user-specified positive parameter, usually on the order of 1. The matrices $G^h \in \mathbb{R}^{n^h \times n^h}$, $A \in \mathbb{R}^{n^h \times n^h}$ and $C \in \mathbb{R}^{n^h \times n^u}$ are fracture-local, whereas $B \in \mathbb{R}^{n^h \times n^u}$ and $G^u \in \mathbb{R}^{n^u \times n^u}$ operate on degrees of freedom related to different fractures. Their properties can be summarized as follows:

- G^h and G^u are symmetric positive semi-definite (SPSD), usually rank-deficient;
- B and C are rectangular coupling blocks, whose entries are given by inner products between the basis functions of the main unknowns along the fracture traces;
- A is symmetric positive definite (SPD) with a block diagonal structure. Each diagonal block arises from the discretization of a $\nabla \cdot (\kappa\nabla)$ operator over a fracture, where κ is a proper diffusion tensor, hence inherits the usual structure of a 2-D discrete Laplacian. Block size depends on each fracture dimension and can be significantly different one from the other.

Equations (1) can be written in a compact form as:

$$\begin{bmatrix} G^h & -\alpha B & A^T \\ -\alpha B^T & G^u & -C^T \\ A & -C & 0 \end{bmatrix} \begin{pmatrix} \mathbf{h} \\ \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{q} \end{pmatrix} \quad \Rightarrow \quad \mathbf{K}\mathbf{x} = \mathbf{f} \tag{2}$$

where $\mathbf{K}$ is a symmetric saddle-point matrix with a rank-deficient leading block. Solution to such problems arise in several applications and is the object of a significant number of works. For a review on methods and ideas, see for instance [3]. With an SPD leading block, as it often arises in Navier-Stokes equations, mixed finite element formulations of flow in porous media, poroelasticity, etc., an optimal preconditioner exists based on the approximation of the Schur complement matrix [4]. However, if the leading block is singular the problem is generally more difficult and the only available result is for the case of maximal rank deficiency [5].

3 PRECONDITIONER FRAMEWORK

Matrix $\mathbf{K}$ in equation (2) is a classical example of the discretization of a coupled multi-physics problem. A general preconditioning framework for such problems can be developed following the results in [6], where the different unknown fields are approximately decoupled to obtain a block diagonal problem.

Theorem 1 of [6] holds true if the leading blocks of $\mathbf{K}$ are non singular. In order to satisfy this hypothesis, a proper row and column block permutation, $\mathbf{P}_r$ and $\mathbf{P}_c$, can be applied:

$$\widetilde{\mathbf{K}} = \mathbf{P}_r \mathbf{K} \mathbf{P}_c, \qquad \widetilde{\mathbf{x}} = \mathbf{P}_c^T \mathbf{x}, \qquad \widetilde{\mathbf{f}} = \mathbf{P}_r \mathbf{f}, \tag{3}$$

such that a decoupling operator can be computed for the equivalent system $\widetilde{\mathbf{K}}\widetilde{\mathbf{x}} = \widetilde{\mathbf{f}}$. A possible choice is:

$$\widetilde{\mathbf{K}} = \begin{bmatrix} A & 0 & -C \\ G^h & A^T & -\alpha B \\ -\alpha B^T & -C^T & G^u \end{bmatrix}, \qquad \widetilde{\mathbf{x}} = \begin{pmatrix} \mathbf{h} \\ \mathbf{p} \\ \mathbf{u} \end{pmatrix}, \qquad \widetilde{\mathbf{f}} = \begin{pmatrix} \mathbf{q} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \tag{4}$$

Let us define the decoupling operator factors $\mathbf{G}, \mathbf{F} \in \mathbb{R}^{N \times N}$ of $\widetilde{\mathbf{K}}$, being $N = 2n^h + n^u$, as:

$$\mathbf{G} = \begin{bmatrix} I & 0 & 0 \\ G_{21} & I & 0 \\ G_{31} & G_{32} & I \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} I & F_{12} & F_{13} \\ 0 & I & F_{23} \\ 0 & 0 & I \end{bmatrix}, \tag{5}$$

with $G_{21}, F_{12} \in \mathbb{R}^{n^h \times n^h}$ and $G_{31}, G_{32}, F_{13}^T, F_{23}^T \in \mathbb{R}^{n^u \times n^h}$, and such that $\mathbf{G}\widetilde{\mathbf{K}}\mathbf{F} = \mathbf{S}$, with $\mathbf{S}$ a block diagonal matrix. Then, the off-diagonal blocks of $\mathbf{F}$ satisfy the relationships:

$$\begin{cases} AF_{12} = 0 \\ \begin{bmatrix} A & 0 \\ G^h & A^T \end{bmatrix} \begin{bmatrix} F_{13} \\ F_{23} \end{bmatrix} = \begin{bmatrix} C \\ \alpha B \end{bmatrix} \end{cases}. \tag{6}$$

from which we obtain:

$$F_{12} = 0, \qquad F_{13} = A^{-1}C, \qquad F_{23} = A^{-T}\left(\alpha B - G^h A^{-1} C\right). \tag{7}$$

Similarly, the off-diagonal blocks of $\mathbf{G}$ read:

$$\begin{cases} G_{21}A = -G^h \\ \begin{bmatrix} G_{31} & G_{32} \end{bmatrix} \begin{bmatrix} A & 0 \\ G^h & A^T \end{bmatrix} = \begin{bmatrix} \alpha B^T & C^T \end{bmatrix} \end{cases}, \tag{8}$$

which provides:

$$G_{21} = -G^h A^{-1}, \qquad G_{32} = C^T A^{-T}, \qquad G_{31} = \left(\alpha B^T - C^T A^{-T} G^h\right) A^{-1}. \tag{9}$$

It is easy to observe that $G_{32} = F_{13}^T$ and $G_{31} = F_{23}^T$, hence only three off-diagonal blocks, namely F_{13}, F_{23}, and G_{21}, are needed. Recalling that $\mathbf{GKF} = \mathbf{S}$, i.e.:

$$\begin{bmatrix} I & 0 & 0 \\ G_{21} & I & 0 \\ F_{23}^T & F_{13}^T & I \end{bmatrix} \begin{bmatrix} A & 0 & -C \\ G^h & A^T & -\alpha B \\ -\alpha B^T & -C^T & G^u \end{bmatrix} \begin{bmatrix} I & 0 & F_{13} \\ 0 & I & F_{23} \\ 0 & 0 & I \end{bmatrix} = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} \tag{10}$$

we have:

$$S_1 = A, \qquad S_2 = A^T, \tag{11}$$

and

$$\begin{aligned} S_3 &= \left(F_{23}^T A + F_{13}^T G^h - \alpha B^T\right) F_{13} + \left(F_{13}^T A^T - C^T\right) F_{23} + G^u - F_{23}^T C - \alpha F_{13}^T B \\ &= G^u - F_{23}^T C - \alpha F_{13}^T B. \end{aligned} \tag{12}$$

Remark 1 *Using the definitions of F_{13} and F_{23}, it is easy to observe that the matrix S_3 of equation (12) is actually the Schur complement of $\widetilde{\mathbf{K}}$ computed with respect to the third block row:*

$$S_3 = G^u - \begin{bmatrix} \alpha B^T & C^T \end{bmatrix} \begin{bmatrix} A & 0 \\ G^h & A^T \end{bmatrix}^{-1} \begin{bmatrix} C \\ \alpha B \end{bmatrix}. \tag{13}$$

Similarly, S_1 and S_2 can be also regarded as the Schur complements computed with respect to the first and second block row of $\widetilde{\mathbf{K}}$, respectively.

Introducing the matrix $E = B - C$, the definition of the Schur complement (12) can be rewritten also as a function of F_{13} only:

$$S_3 = G^u + F_{13}^T \left(G^h - 2\alpha A\right) F_{13} - \alpha \left(E^T F_{13} + F_{13}^T E\right) \tag{14}$$

From equation (10) it follows immediately:

$$\widetilde{\mathbf{K}}^{-1} = \mathbf{F}\mathbf{S}^{-1}\mathbf{G}, \tag{15}$$

that is, the expression of the exact inverse of the block matrix $\widetilde{\mathbf{K}}$. Of course, equation (15) cannot be computed explicitly in large-size applications, because both the decoupling off-diagonal blocks in $\mathbf{F}, \mathbf{G}$ and the diagonal blocks in $\mathbf{S}^{-1}$ are dense. However, we can use the factorization (15) to build an inexact application of $\widetilde{\mathbf{K}}^{-1}$ that can be used as a *preconditioner* in a Krylov subspace method.

Since our aim is to compute the product of $\widetilde{\mathbf{K}}^{-1}$ by a vector $\mathbf{r} \in \mathbb{R}^N$, we do not necessarily need to form an explicit expression of $\mathbf{F}$ and $\mathbf{G}$, but just to define an algorithm to compute their products by portions of size n^h and n^u of a vector lying in $\mathbb{R}^N$. This can be done exactly and efficiently in a parallel computational environment by recalling the properties of matrix A (see section 2). Similarly, also S_1^{-1} and S_2^{-1} (equation (11)) can be exactly applied to a vector. Hence, the block preconditioner $\mathbf{M}^{-1}$ for $\widetilde{\mathbf{K}}$ can be defined as:

$$\mathbf{M}^{-1} = \mathbf{F}\hat{\mathbf{S}}^{-1}\mathbf{G}, \tag{16}$$

where $\hat{\mathbf{S}}^{-1}$ reads:

$$\hat{\mathbf{S}}^{-1} = \begin{bmatrix} A^{-1} & 0 & 0 \\ 0 & A^{-T} & 0 \\ 0 & 0 & \hat{S}_3^{-1} \end{bmatrix}, \tag{17}$$

$\hat{S}_3^{-1}$ being some approximation, either implicit or explicit, of S_3.

For the eigenspectrum of the preconditioned matrix $\mathbf{M}^{-1}\widetilde{\mathbf{K}}$, the following result holds true.

Lemma 1 *Let* $\widetilde{\mathbf{K}}, \mathbf{M}^{-1} \in \mathbb{R}^{N\times N}$ *be the matrices defined in* (3) *and* (16)*, respectively. Then, the eigenvalues* λ *of* $\mathbf{M}^{-1}\widetilde{\mathbf{K}}$ *are either 1, with multiplicity* $2n^h$*, or equal to those of the matrix* $\hat{S}_3^{-1}S_3$*.*

Proof 1 *By using equation* (16)*, the matrix* $\mathbf{M}^{-1}\widetilde{\mathbf{K}}$ *reads:*

$$\mathbf{M}^{-1}\widetilde{\mathbf{K}} = \mathbf{F}\hat{\mathbf{S}}^{-1}\mathbf{G}\widetilde{\mathbf{K}}, \tag{18}$$

which is similar to $\hat{\mathbf{S}}^{-1}\mathbf{G}\widetilde{\mathbf{K}}\mathbf{F}$*. Recalling* (10)*, we have:*

$$\begin{aligned} \hat{\mathbf{S}}^{-1}\mathbf{G}\widetilde{\mathbf{K}}\mathbf{F} &= \hat{\mathbf{S}}^{-1}\mathbf{S} \\ &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \hat{S}_3^{-1}S_3 \end{bmatrix}, \end{aligned} \tag{19}$$

which completes the proof.

The key for the effectiveness of $\mathbf{M}^{-1}$ as a preconditioner of $\widetilde{\mathbf{K}}$ is therefore the selection of $\hat{S}_3^{-1}$. In the next paragraph, we analyze the results from different choices for $\hat{S}_3^{-1}$.

4 NUMERICAL RESULTS

Since the effectiveness of $\mathbf{M}^{-1}$ depends on $\hat{S}_3^{-1}$ only, we reduce the system (4) on the flux space:

$$S_3\mathbf{u} = \mathbf{b} \quad \text{with} \quad \mathbf{b} = \left(\alpha B^T - F_{13}^T G^h\right) A^{-1}\mathbf{q} \tag{20}$$

Since S_3 is SPD, system (20) is solved by a preconditioned CG method, setting the maximum number of iterations to 1500 and the exit tolerance on the relative residual to 10^{-6}. Four problems of increasing size have been analyzed (Table 1). Figure 1 shows the mesh domain for the case $\mathrm{P_C}$.

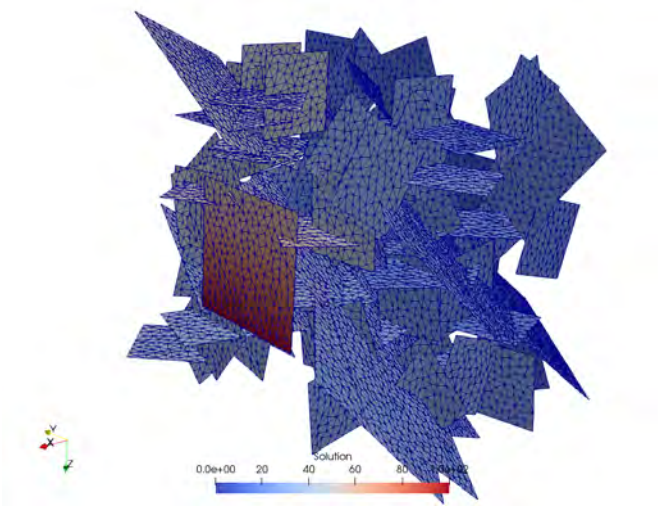

Figure 1: 3D mesh domain for the case $\mathrm{P_C}$.

Table 1: Problem size.

	$\mathrm{P_A}$	$\mathrm{P_B}$	$\mathrm{P_C}$	$\mathrm{P_D}$
n^h	787	13732	39288	93768
n^u	206	5085	8219	18276
N	1780	32549	86795	205812

The non-zero pattern of the matrices of the smallest problem is shown in figure 2. Matrices A, C and G^h are block diagonal. Being each block related to a fracture, these matrices are fracture-local. Instead, matrices B and G^u connect degrees of freedom related to different fractures. In particular, matrix B is made by the same diagonal blocks as C with additional extra-diagonal

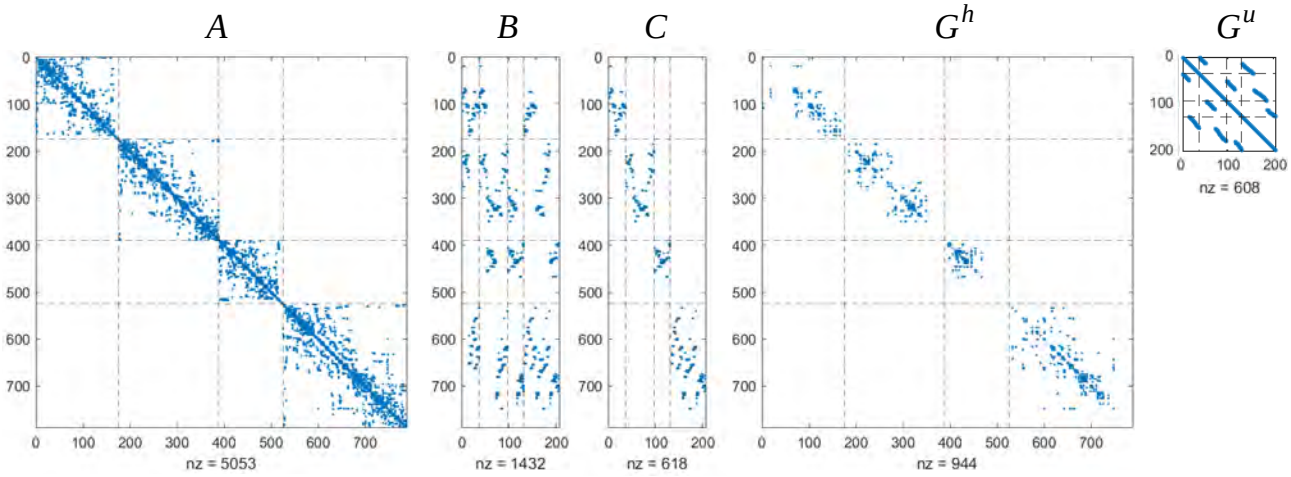

Figure 2: Structure and number of non-zeros of the matrices for case $\mathrm{P_A}$.

terms corresponding to intersections between fractures. Thus, the matrix $E = B - C$ is zero on the diagonal blocks and contains the same terms as B outside. Matrix F_{13}, being defined as $A^{-1}C$, is also block diagonal, with the same size and structure as C.

On the basis of these considerations, the Schur complement can be written as (see equation (14)):

$$S_3 = S_D - S_E \tag{21}$$

where $S_D = G^u + F_{13}^T \left(G^h - 2\alpha A\right) F_{13}$ and $S_E = \alpha \left(E^T F_{13} + F_{13}^T E\right)$. Matrix S_D contains the diagonal blocks of S_3 and S_E the off-diagonal part. Therefore, S_D is SPD, whereas S_E is indefinite.

A key property for $\hat{S}_3$ is being SPD. It is therefore natural to consider $\hat{S}_3 = S_D$, that is the block diagonal and positive definite part. The results in terms of number of iterations (iter), ratio between the non-zeros of the approximate Schur complement and the exact one (μ) and the conditioning number (ξ) are reported in Table 2. Despite the preconditioning, the number of iterations required to solve the system is still high and the conditioning number of the preconditioned matrix is not very different from the original.

Table 2: Results considering the approximation $\hat{S}_3 = S_D$. The $*$ indicates that the problem does not converge, with the residual stagnating around 10^{-5}.

Case	iter	μ	$\xi\left(\hat{S}_3^{-1} S_3\right)$	$\xi\left(S_3\right)$
$\mathrm{P_A}$	125	0.3921	3.10e+04	1.67e+04
$\mathrm{P_B}$	300	0.3958	2.08e+06	4.90e+05
$\mathrm{P_C}$	$*$	0.3619	1.40e+08	1.72e+09
$\mathrm{P_D}$	957	0.3594	7.39e+06	1.15e+09

Approximating S_3 with its diagonal blocks appears to be not enough for an efficient solution of the system. Thus, in the following also the off-diagonal part is taken into account. Aiming at understanding the importance of the single blocks of S_3 as a preconditioner, we filter the two contributions S_D and S_E separately, naming $\hat{S}_D$ and $\hat{S}_E$ their approximation. First, only the extra-diagonal part of S_3 is approximated:

$$\hat{S}_3 = S_D - \hat{S}_E \tag{22}$$

where $\hat{S}_E$ is obtained by filtering each column j of the product $E^T F_{13}$ neglecting the components

such that:

$$\left|\left(E^T F_{13}\right)_{ij}\right| < \tau \left\|\left(E^T F_{13}\right)_j\right\|_2 \tag{23}$$

Results for different values of τ are reported in Table 3.

Table 3: Results computing S_3 with the sparsified S_E. The * indicates the case when $\hat{S}_3$ becomes indefinite.

τ	case P_A			case P_B		
	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$
5×10^{-2}	*	0.8306	5.47e+03	*	0.4747	6.41e+08
10^{-2}	8	0.9398	1.79e+02	26	0.6171	5.07e+06

	case P_C			case P_D		
10^{-2}	*	0.9577	3.26e+09	*	0.8056	4.68e+07
10^{-3}	7	0.9950	2.28e+04	*	0.9742	3.44e+10

Finally, we consider the preconditioner $\hat{S}_3$:

$$\hat{S}_3 = \hat{S}_D - S_E \tag{24}$$

where the extra-diagonal blocks are computed exactly, while the diagonal ones are approximated neglecting the components s_{ij} of the product $F_{13}^T\left(G^h - 2\alpha A\right)F_{13}$ such that:

$$|s_{ij}| < \tau\sqrt{|s_{ii}\ s_{jj}|} \tag{25}$$

Results for the four matrices are reported in Table 4.

Table 4: Results computing S_3 after the sparsification of S_D. The * indicates the case when $\hat{S}_3$ becomes indefinite.

τ	case P_A			case P_B		
	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$
5×10^{-1}	*	0.6604	1.47e+04	*	0.6114	3.51e+08
10^{-1}	10	0.9217	1.99e+03	*	0.6351	4.62e+07
10^{-2}	3	0.9910	2.99e+00	*	0.8902	1.84e+05
10^{-3}	2	0.9987	1.06e+00	*	0.9925	2.33e+04

	case P_C			case P_D		
10^{-2}	*	0.9928	2.34e+04	*	0.9851	4.39e+05
10^{-3}	2	0.9993	2.47e+01	2	0.9987	6.88e+01

In both cases, i.e. when approximating only S_E or S_D, the level of fill-in of $\hat{S}_3$ required for the convergence is near to the one of the exact Schur complement ($\mu \simeq 1$). This is because $\hat{S}_3$ can easily become indefinite after the filtering. As an example, in figure 3 the ten smallest eigenvalues of the exact and the approximated (with τ equal to 5×10^{-1}) Schur complement for the case P_A are shown. While S_3 is positive definite, the eigenvalues of $\hat{S}_3$ are both positive and negative.

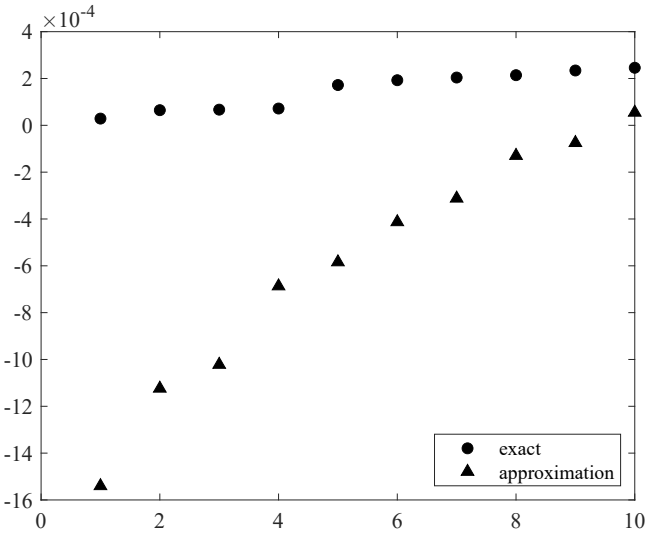

Figure 3: Case $\mathrm{P_A}$: ten smallest eigenvalues of S_3 and $\hat{S}_3$ computed through equation (24) with $\tau = 5 \times 10^{-1}$.

In the last test, the preconditioner is computed approximating both S_D and S_E:

$$\hat{S}_3 = \hat{S}_D - \hat{S}_E \tag{26}$$

To this aim, a sparsified F_{13} is computed by filtering the smallest components. Since F_{13} is block diagonal, it can be efficiently computed in a parallel computational environment exploiting a Cholesky factorization of the blocks of A. A relative drop tolerance is used, removing the components such that:

$$|F_{13,ij}| < \tau \left\|F_{13,j}\right\|_2 \tag{27}$$

Results are reported in Table 5. The iterations count can decrease significantly with respect to Table 2, with densities that are even smaller than those obtained keeping S_D only. However, in difficult problems, such as $\mathrm{P_C}$, quite a high fill-in can be required and the performance can be very sensitive to the τ selection.

Table 5: Results computing S_3 with the approximation of F_{13}.

τ	case $\mathrm{P_A}$			case $\mathrm{P_B}$		
	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$	iter	μ	$\xi\left(\hat{S}_3^{-1}S_3\right)$
10^{-1}	28	0.2697	3.75e+04	136	0.1226	1.88e+07
5×10^{-2}	19	0.8595	5.85e+04	57	0.2508	1.45e+07
10^{-2}	1	1.0000	1	17	0.6072	1.48e+06

	case $\mathrm{P_C}$			case $\mathrm{P_D}$		
5×10^{-2}	1483	0.5400	1.08e+11	445	0.3196	1.06e+09
2.5×10^{-2}	8	0.9952	1.64e+06	128	0.5663	8.89e+07
10^{-2}	4	0.9990	3.38e+04	41	0.8100	1.57e+07
10^{-3}	1	1.0000	1	5	0.9912	1.10e+05

Considering as a preconditioner an approximation $\hat{S}_3$ obtained by filtering S_3 or its components can be efficient (as results in Table 5 demonstrate), but also quite fragile because of the possible indefiniteness of the approximation (see Table 3 and Table 4).

5 CONCLUSIONS

A symmetric saddle-point matrix with a rank-deficient leading block arises from the combination of DFN models with an appropriate optimization formulation. Here, we focused on accelerating the iterative solution of this system with a block preconditioning technique. First, an appropriate permutation of the matrix is performed and then a projection on the Schur complement space of the flux is performed. The Schur complement proves to be the key for an effective preconditioner, therefore we investigate different approaches to approximate it. Both the diagonal and off-diagonal blocks of the Schur complement are fundamental for an efficient solution of the system. Independent filterings of such components reveal the fragility of the approximated Schur complement, that can easily become indefinite. When the filter step regards the matrix F_{13}, before the computation of the Schur complement, results are more promising. This suggests to investigate different other filtering approaches for F_{13}, aiming at finding a more robust and less τ dependent solution. Alternatively, a polynomial acceleration in a matrix-free implementation can help improving the performance when working in a parallel environment. Comparing the approaches we investigated, we noted that the conditioning number does not vary according to the number of iterations, as one can expect. This can be related to the distribution of the eigenvalues, that means that the eigenspectrum is mainly grouped, but there are a few outliers. In order to fix this problem, a deflation approach can be used to remove the eigenvectors related to the extreme eigenvalues. This technique requires the a priori knowledge of these eigenvalues, that is quite computational expensive, but reasonable in an iterative framework.

REFERENCES

[1] Berrone, S. and Pieraccini, S. and Scialò, S. A PDE-constrained optimization formulation for Discrete Fracture Network flows. *SIAM J. Sci. Comput.*, Vol. **35**, pp. B487–B510, (2013).

[2] Berrone, S. and Scialò, S. and Vicini, F. Parallel meshing, discretization and computation of flow in massive Discrete Fracture Networks. *SIAM J. Sci. Comput.*, Vol. **41**, pp. C317-C338, (2019).

[3] Benzi, M. and Golub, G.H. and Liesen, J. Numerical solution of saddle point problems. *Acta Numerica*, Vol. **14**, pp. 1–137, (2005).

[4] Elman, H.C. and Silvester, D.J. and Wathen, A.J. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.

[5] Estrin, R. and Greif, C. On nonsingular saddle-point systems with a maximally rank-deficient leading block. *SIAM Journal on Matrix Analysis and Applications*, Vol. **36**, pp. 367–384, (2015).

[6] Ferronato, M. and Franceschini, A. and Janna, C. and Castelletto, N. and Tchelepi, H.A. A general preconditioning framework for coupled multi-physics problems. *J. Comput. Phys.*, Vol. **398**, (2019).