PHYSICS-BASED PRECONDITIONERS FOR POROUS MEDIA FLOW APPLICATIONS

BURAK AKSOYLU, HECTOR KLIE, AND MARY F. WHEELER

Abstract. Eigenvalues of smallest magnitude have known to be a major bottleneck for iterative solvers. Such eigenvalues become a more dramatic bottleneck when the underlying physical properties have severe contrasts. These contrasts are commonly found subsurface geological properties such as permeability and porosity. We intend to construct a method as algebraic as possible. In particular, we propose an algebraic way of using the underlying permeability field to mark certain degrees of freedom as high permeable when they exceed a certain threshold. This marking process will define a permutation matrix which allows us to collect the degrees of freedom that causes the smallest eigenvalues in a subblock. We claim the responsibility of ill-conditioning to this subblock of the system matrix. The remaining of the matrix will then be well-conditioned if certain heuristics about the permeability field are satisfied. In our two-stage preconditioning approach, the first stage comprises the process of collecting small eigenvalues and solving them separately; the second stage deals with the remaining of the matrix possibly with a deflation strategy if needed. Numerical examples are shown for one- and two-phase flow scenarios in reservoir simulation applications. We demonstrate that our two-stage preconditioners are more effective and robust compared to deflation methods. Due to their algebraic nature, they support flexible and realistic reservoir topology.

Keywords. Preconditioning, two-stage, Schur complement, porous media flow, Krylov subspace, GMRES, iterative solver, reservoir simulation, deflation preconditioners.

1. Introduction

The main objective of the present work is to introduce a novel physics-based preconditioning strategy for solving problems with high physical contrasts in porous media applications. These stringent situations commonly arise, for example, in multilayered geological formations composed of different type of rocks (e.g., sandtones, shales, carbonates, limestones). In a multiscale and broader geometrical sense, these formations are also composed of different minerals and are subject to various types of deformations. In terms of porous media flow applications, these rock variations are characterized by porosity and permeability values. Although both properties affect different components of flow equations, we focus the attention on the effects that the permeability field has in the resulting algebraic system. This is motivated by the fact that the permeability is generally the main driving factor for the conductance or allowance of fluid to flow through the porous media. In fact, Darcy's Law states that permeability values determine the magnitude of fluid velocities given a change (gradient) of subsurface fluid pressures [21, 37]. There is, in general, a direct proportionality between permeability and porosity: low permeability zones translates into low porosity zones causing the fluid to move slowly or even being absent in those regions, and conversely, high-permeability zones coincides with high-porosity zones, favoring fast fluid flow and the occurrence of regions invaded by the fluid.

Our algebraic approach is strongly motivated by the permeability distribution and its local contrasts. In particular, we are assuming that the porous media consist of highly permeable interconnected regions allowing for a strong global flow conductivity (e.g., channelized media). Figure 1 illustrates this type of permeability distribution settings that we are interested in handling efficiently from the iterative solution standpoint. Note that most of the relevant flow physics is governed by the high permeable network structure.

Therefore, we exploit the fact that coefficients associated with distinguishable quasi-homogeneous permeability regions should be grouped in the same block. Thus, we separate the matrix coefficients into high and low permeability blocks given thus giving rise to a 2×2 block linear system. The solution of this linear system is carried out by a Krylov iterative solver using a constrained solution to the high-permeability block as a one preconditioning stage. Error components that still remain high after this stage can be additionally smoothed out by the action of a global preconditioning stage. This second stage is generally cheap and is designed to capture solution components associated with the original coupling and possible roughness within the low-permeability block. The rationale behind this two-stage preconditioning is that high permeability block captures major pressure and fluid velocity changes which governs the solution of the overall coupled system.

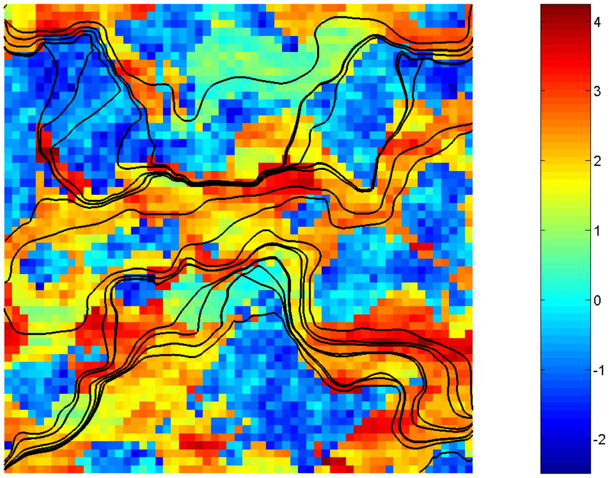

Figure 1. A channelized reservoir described in terms of the log permeability field. Streamlines indicate preferential flow paths.

Due to the target application considered here and the use of the deflation as a second preconditioning stage, our work has relevant connections with the pioneering research that Vuik and several of his coauthors have developed on deflation methods for layered problems with extreme permeability contrasts; see e.g., [22, 43, 56, 57, 58]. However, the present work adds a new dimension to their work in the following aspects: (1) the present framework may accommodate domain-based deflation strategies as proposed by Vuik and coauthors as well as other preconditioning strategies such as coarse grid correction, domain decomposition and multigrid; (2) geometry of high- and low-permeability regions may vary arbitrarily in connected region; (3) comparisons are made with Krylov-based deflation approaches in the setting of porous media for single and two-phase flow; and, (4) the proposed preconditioner is inspired on previous two-stage preconditioning work successfully applied for solving coupled linear systems involving different physical variables arising in fully implicit formulations [11, 18, 31, 52].

The structure of the paper is as follows. Section 2 outlines the physical motivation supporting our approach. This includes an assessment of the type of domains and the formalization of matrix reorderings we are interested in constructing. Section 3 focuses in describing the two-stage preconditioning strategy designed to reflect the physics behind the linear systems. We show connections of these preconditioners with Schur complement formulations and possible extensions to the proposed methodology. Section 4 presents and relates Krylov-based and domain-based deflation approaches; including discussion on how to approximate eigenvectors and construct deflation operators for both approaches. This description has a two-fold blend: propose efficient ways to compute the second stage of two-stage preconditioners and revisit and compare some of the Krylov-based deflation operators that have been proposed in the literature. Section 5 numerically compares the proposed physics-based preconditioning strategy against some well established Krylov-based deflation methods. We end the paper with conclusions and directions of future work.

2. Formulation of the Problem: Exploiting the Physics

The complexity of fluid flow in a heterogeneous reservoir is one of the major reasons why reservoir simulators are needed in reservoir engineering studies. The present section provides a brief discussion on the porous media flow equations, the physical domain under study and its implications in forming the linear system.

2.1. **The Porous Media Flow Model.**

The basic equations for multiphase flow in porous media consist of mass conservation equations for different phases (i.e., oil, gas and water in oil reservoir simulation) plus a set of constitutive (closure) relations that relate pressure and saturations of these phases; a detailed description of the model can be found in [37]. The mass conservation of each phase i is given by

$$\frac{\partial(\phi\rho_i S_i)}{\partial t} + \nabla\cdot(\rho_i \mathbf{u}_i) = q_i, \tag{2.1}$$

where ρ_i is the density, ϕ is the porosity, S_i is the saturation, t is time, q_i is the well term defined by production/injection rates at reservoir conditions, and $\mathbf{u}_i$ is the phase Darcy velocity which is expressed as

$$\mathbf{u}_i = -\frac{\mathbf{K}k_{ri}}{\mu_i}\left(\nabla P_i - \rho_i g \nabla Z\right),$$

where $\mathbf{K}$ is the absolute permeability tensor, k_{ri} is the relative permeability, μ_i is the viscosity, P_i is the pressure, g is the gravity and Z is the depth. The primary unknowns of this model are phase saturations and phase pressures. In the event of incompressibility (i.e., constant density), the temporal term vanishes. Additionally, omitting gravity effects we then obtain the following simplified elliptic equation:

$$\nabla\cdot\mathbf{u} = -\nabla\cdot\left(\frac{\mathbf{K}\left(x\right)}{\mu}\nabla P\right) = \frac{q}{\phi}. \tag{2.2}$$

We have intentionally dropped the subindex i and the relative permeability term k_r to indicate that this is a single-phase flow model. We are also emphasizing the fact that $\mathbf{K}\left(x\right)$ depends on space and can be highly variant with possible discontinuities, but bounded below by a positive constant (i.e., coercivity condition). To introduce further notation, we assume that equation (2.2) is defined on a domain Ω with non-flux boundary conditions, that is,

$$\mathbf{u}\cdot\vec{n} = 0 \quad \text{in} \quad \partial\Omega. \tag{2.3}$$

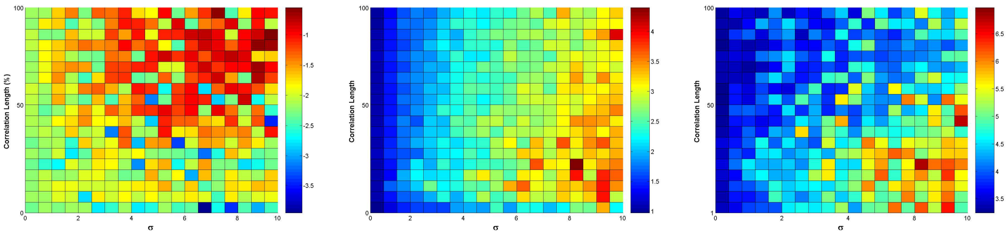

FIGURE 2. Min eigenvalue, maximum eigenvalue and condition number as a function of variance and correlation length of exponentially distributed random permeability fields.

Robust simulators need to be able to handle high variations of permeability and porosity (i.e., high heterogeneity contrasts). Many of these variations can be encountered in different geological situations that include: (1) vertical stratification with resultant channeling of fluids in high-permeability zones; (2) low-permeability material (e.g., shales) that may influence vertical fluid displacement or areal sweep efficiency; (3) discontinuous permeable zones that may affect vertical sweep efficiency; (4) natural fractures or cracks that may reduce recovery from rock matrix; and, (5) areal permeability variations that prevent flow of injected material. In any of these situations, solution of linear systems is a challenging task. In fact, linear system solutions approximately consumes 75%-90% of the overall simulation time regardless of the discretization method.

Figure 2 shows the effect of heterogeneities in the value of extreme eigenvalues and in the resulting condition number of the linear system associated to equation (2.2). In this example, we consider the log of the permeability $W\left(\mathbf{x}\right) = \ln K\left(\mathbf{x}\right)$ following a second-order stationary distribution with a separable exponential covariance function of the form

$$C_W\left(\mathbf{x_1},\mathbf{x_2}\right) = C_W\left(x_{1,1},x_{1,2};x_{2,1},x_{2,2}\right) = \sigma_W^2 \exp\left[\frac{-1}{\eta}\left\{|x_{1,1}-x_{2,1}| + |x_{1,2}-x_{2,2}|\right\}\right], \tag{2.4}$$

where σ_W^2 is the spatial variability of the log permeability field and η is the correlation length. Several realizations were generated for different combinations of σ_W^2 and η. Note that for a constant problem size and gridsize, the smaller the correlation length or the greater the spatial variability, the larger the matrix condition number. Similar trend can be obtained in other stationary and non-stationary distributions such as those arising in layered or channelized formations where the greater the coefficient jump (spatial variability) the higher the associated linear system condition number.

Based on the above observations, the absolute permeability $\mathbf{K}$ plays a decisive role in the distribution of saturation, pressure and fluxes on the simulation domain Ω. To simplify the proposed procedure, we assume that $\mathbf{K}$ is a positive scalar function which depends on position and describes highly permeable regions such as channels either branching or merging across the domain Ω. Additionally, we assume that $\mathbf{K}$ is isotropic, that is, permeability does not vary with direction: $(\mathbf{K})_{ij} = K\delta_{ij}$ for $i, j = 1, 2, 3$.

2.2. **Ordering Unknowns According to Permeability Contrasts.** Since $\mathbf{K}$ defines highly conductive paths, it is convenient to partition Ω in two distinguishable regions. Let Ω_h denote the high permeable region and Ω_l denote the low permeable region. We assume that $\Omega_h \cup \Omega_l = \Omega$ and $\Omega_h \cap \Omega_l = \emptyset$.

The following rationale defines the development of our forthcoming ideas: high-permeability zones should yield faster changes in the solution and conversely, low-permeability zones should yield almost constant solutions. Obviously, before putting this in practice we need to formalize what we mean by high- and low-permeability zones in a layered system. This can be done by defining an average or threshold permeability value $\langle K \rangle$.

Clearly, the computation of $\langle K \rangle$ will depend on distribution of permeability scales in Ω. This procedure is a challenging task by itself and has been subject of intensive research by many authors; see e.g., [46, 51]. A practical choice for $\langle K \rangle$ is the geometric mean, that is,

$$\langle K \rangle = \exp\left\{ \frac{\int_\Omega \ln \mathbf{K}(x)\, d\mathbf{x}}{\int_\Omega d\mathbf{x}} \right\}. \tag{2.5}$$

This definition of $\langle K \rangle$ is generally employed to find an effective value for a log-normal permeability distribution. Nevertheless, we should remark that geometric averaging is generally inaccurate to describe channelized media but sufficient for the type of cases we will analyze in this work. A more accurate computation of effective media in channelized media is usually performed via percolative methods [48].

In this way, all grid elements with a permeability value larger than $\langle K \rangle$ are numbered first and those with a value lower are numbered after. This gives rise to a 2×2 block system (for either pressures or concentrations) of the following form:

$$A^{\text{orig}} = \begin{bmatrix} A_h^{\text{orig}} & A_{hl}^{\text{orig}} \\ A_{lh}^{\text{orig}} & A_l^{\text{orig}} \end{bmatrix}. \tag{2.6}$$

A_h^{orig} and A_l^{orig} denote the blocks corresponding to high and low permeable regions, respectively, A_{hl}^{orig} and A_{lh}^{orig} denote couplings. We have emphasize on the use of the superscript orig to distinguish this system from the scaled one (to be described in the next section). Correspondingly, the Schur complement of A^{orig} is given by $A_S^{\text{orig}} = A_l^{\text{orig}} - A_{lh}^{\text{orig}} A_h^{\text{orig}^{-1}} A_{hl}^{\text{orig}}$.

2.3. **Properties and Diagonal Scaling.** Given the block form (2.6), we expect to have concentrated the most relevant solution of the system in block A_h^{orig}. Some additional comments are in order:

- Given model equation (2.2), the coefficient blocks A_h^{orig} and A_l^{orig} are both symmetric positive definite (SPD) and diagonally dominant. This fact results from considering coercivity/boundedness assumptions and standard discretization procedures. We should stress, however, that in fully-implicit formulations of multiphase flow these properties do not hold and in most cases, are replaced by nonsymmetry and M-matrix property assumptions (see e.g., [4, 17, 31]).
- The dimension of A_h^{orig} and A_l^{orig} may differ significantly depending on the layered reservoir situation. Important computational savings may be obtained when $|\Omega_l| \gg |\Omega_h|$.

- Each block $A_h^{\rm orig}$ and $A_l^{\rm orig}$ may consist of several main diagonal blocks associated with disconnected high-permeability regions (e.g., layering system where low- and high-permeability layers alternate in the formation). We should later stress that disconnected regions may not be necessarily included in the solution process.
- The number of layer blocks in each block $A_h^{\rm orig}$ and $A_l^{\rm orig}$ may vary in size and magnitude of its entries due to the presence of low-scale heterogeneities within each distinguishable permeability zone.
- The blocks $A_{hl}^{\rm orig}$ and $A_{lh}^{\rm orig}$ tend to have smaller entries as the permeability contrast becomes higher.

Since $A^{\rm orig}$ is SPD and diagonally dominant there are several implications in the properties associated with the blocks. First of all, note that these properties are invariant upon row and column permutations. Also, diagonal scaling creates a clustering effect and the spectral radius of the diagonally scaled matrix becomes $0 < \rho(A) < 2$ and difficulties in the solution process may arise when eigenvalues are either lying too close to the origin or 2.

We invoke some important results associated with the 2×2 block partitioning given by (2.6) (see [3]):

Theorem 2.1. [On symmetry and positive definiteness]. *If $A^{\rm orig}$ is SPD then each of the following statements follows:*

(1) $A_h^{\rm orig}$ *and* $A_l^{\rm orig}$ *are SPD.*
(2) $A_S^{\rm orig}$ *is SPD .*
(3) $\kappa_2(A_S^{\rm orig}) \le \kappa_2(A^{\rm orig})$.
(4) $\|A_{lh}^{\rm orig} {A_h^{\rm orig}}^{-1}\|_2^2 \le \kappa_2(A^{\rm orig})$.

On the other hand, diagonal scaling brings up other important implications:

Theorem 2.2. [On diagonal scaling]. *Let $D_l^{\rm orig} = {\rm diag}(A_l^{\rm orig})$ and $D_h^{\rm orig} = {\rm diag}(A_h^{\rm orig})$, then the following statements hold:*

(1) $S = {D_l^{\rm orig}}^{-1} S^{\rm orig}$.
(2) $\kappa_2(A_S) \le \kappa_2(D_l^{\rm orig})\kappa_2(A_S^{\rm orig})$.
(3) A_S *is similar to the SPD matrix* $T := {D_l^{\rm orig}}^{-1/2} A_S^{\rm orig} {D_l^{\rm orig}}^{-1/2}$*. Therefore, eigenvalues of A_S are all positive.*

Based on the just aforementioned facts, we construct a scaling or Jacobi preconditioner operator D given by

$$D^{\rm orig} = \begin{bmatrix} {\rm diag}(A_h^{\rm orig}) & 0 \\ 0 & {\rm diag}(A_l^{\rm orig}) \end{bmatrix} =: \begin{bmatrix} D_h^{\rm orig} & 0 \\ 0 & D_l^{\rm orig} \end{bmatrix} \tag{2.7}$$

and compute the scaled system

$$A = {D^{\rm orig}}^{-1} A^{\rm orig} = \begin{bmatrix} {D_h^{\rm orig}}^{-1} A_h^{\rm orig} & {D_h^{\rm orig}}^{-1} A_{hl}^{\rm orig} \\ {D_l^{\rm orig}}^{-1} A_{lh}^{\rm orig} & {D_l^{\rm orig}}^{-1} A_l^{\rm orig} \end{bmatrix} =: \begin{bmatrix} A_h & A_{hl} \\ A_{lh} & A_l \end{bmatrix}. \tag{2.8}$$

We observe that diagonal scaling improves the clustering of eigenvalues, therefore, diagonal scaling is always the initial default preconditioner. In various deflation preconditioner frameworks [57, 59], the initial default preconditioner is chosen to be ILU(0). The simple diagonal scaling also conforms to our intention of keeping the preconditioner as simple as possible. We denote the original system matrix by $A^{\rm orig}$ and the diagonally scaled matrix by $A := {D^{\rm orig}}^{-1} A^{\rm orig}$, where $D^{\rm orig} = {\rm diag}(A^{\rm orig})$.

3. Physics-Based Preconditioning

3.1. **Two-Stage Preconditioning and the Schur Complement.** Two stage-preconditioning refers to the concept of effectively combining the action of two preconditioners into a single one [16]. For instance, given the right-preconditioned matrix $A_1 = AM_1$, we can apply a second preconditioner (either from the right or left) to A_1. The idea can be easily generalized to multi-stage preconditioning by extending the recurrence several times. The motivation for combining multiple preconditioning stages obeys to the need of taking advantage of particular features of the problem within a single preconditioning stage. In a certain way, multigrid, and domain decomposition methods also fit into this framework. In the arena of porous media flow applications, two-stage preconditioning has made a name by its own for the solution of fully coupled systems; see e.g. [18, 31, 53]. Moreover, this preconditioning technology is currently being considered an important initiative for the development of new generation of reservoir simulators for multiphase and compositional flow in porous media [11, 23, 38, 55].

The two-stage preconditioner framework that we are interested in developing to be able to exploit the physics of the problem is the following:

Algorithm 3.1. (Physics-based two-stage preconditioner)

(1) Solve high permeability system: $A_h y_h = r_h$, where $A_h := R^t A R,\ r_h := R^t r$.
(2) Obtain expanded solution: $y = R y_h$.
(3) Compute new residual: $\hat{r} = r - Ay$.
(4) Correct the residual: $\hat{v} = \hat{r} + y$.
(5) (If needed) apply a stage two preconditioner M_d: $v = M_d^{-1}\hat{v}$.

The action of the whole preconditioner can be compactly written as

$$v = M_{\text{left}}^{-1} r = M_d^{-1}\left[I - (A - I)\, R \left(R^t A R\right)^{-1} R^t\right] r. \tag{3.1}$$

M_d^{-1} is an appropriate preconditioner of some desired kind such as a deflation preconditioner. In any case, the preconditioner M_d is used to solve those frequencies associated with the coupling to the low-permeability block A_l. This step takes care of the small eigenvalues generated by the permeability contrast existing at the interface between layers. We note that M is an exact left inverse of A on the subspace spanned by the columns of R. That is, $\left(M_{\text{left}}^{-1} A\right) R = R$.

The inclusion operator under consideration is given as: $R = \begin{bmatrix} I_h \\ 0 \end{bmatrix}$. Then,

$$R \left(R^t A R\right)^{-1} R^t = \begin{bmatrix} A_h^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

If we do not use a stage two preconditioner, $M_d = I$, we get

$$M_{\text{left}}^{-1} = \begin{bmatrix} A_h^{-1} & 0 \\ -A_{lh} A_h^{-1} & I_l \end{bmatrix}. \tag{3.2}$$

Then, M_{left}^{-1} becomes the exact inverse of $\begin{bmatrix} A_h & 0 \\ A_{lh} & I_l \end{bmatrix}$.

When we further decompose (3.2) as

$$M_{\text{left}}^{-1} = \begin{bmatrix} A_h^{-1} & 0 \\ 0 & I_l \end{bmatrix} \begin{bmatrix} I_h & 0 \\ -A_{lh} A_h^{-1} & I_l \end{bmatrix}, \tag{3.3}$$

we can connect M_{left}^{-1} to a factorization of A which contains a Schur complement:

$$A = \begin{bmatrix} I_h & 0 \\ A_{lh} A_h^{-1} & I_l \end{bmatrix} \begin{bmatrix} A_h & A_{hl} \\ 0 & A_S \end{bmatrix}, \tag{3.4}$$

where A_S denotes the Schur complement of A_h in A;

$$A_S = A_l - A_{lh} A_h^{-1} A_{hl}. \tag{3.5}$$

Now, combining (3.3) and (3.4) we get:

$$M_{\text{left}}^{-1} A = \begin{bmatrix} I_h & A_h^{-1} A_{hl} \\ 0 & A_S \end{bmatrix}, \tag{3.6}$$

which indicates that

$$\sigma(M_{\text{left}}^{-1}A) = \sigma(A_S) \cup \{1\}. \tag{3.7}$$

If we supplement M_{left}^{-1} in (3.2) by including the inversion of A_S, then a perfect preconditioner can be obtained which will be defined as $M_{\text{left,opt}}^{-1}$:

$$\begin{aligned} M_{\text{left,opt}}^{-1} &= \begin{bmatrix} A_h^{-1} & 0 \\ 0 & A_S^{-1} \end{bmatrix} \begin{bmatrix} I_h & 0 \\ -A_{lh}A_h^{-1} & I_l \end{bmatrix}, & (3.8) \\ M_{\text{left,opt}}^{-1}A &= \begin{bmatrix} I_h & A_h^{-1}A_{hl} \\ 0 & I_l \end{bmatrix}, & (3.9) \end{aligned}$$

with $\sigma(M_{\text{left,opt}}^{-1}A) = \{1\}$. However this comes with the heavy price of inverting the Schur complement A_S. First of all, A_S loses sparsity due to the inversion of A_h. Secondly, when the Schur complement A_S ill-conditioned, inverting it can be quite troublesome.

We conclude that the dependence of the Schur complement is the inherit property of our two-stage preconditioner. The availability of an effective preconditioner of A_S in our two-stage preconditioning would be an ideal scenario. In general, we do not have control on either the conditioning or the sparsity of A_S. Hence, we proposed the heuristic strategy outlined in §2. which utilizes the underlying physical properties and forms a new permutation of the original system matrix such that the ill-conditioning (namely, eigenvalues of smallest magnitude) is collected in a small A_h block so that A_S is large but well-conditioned so that replacing A_S by I_l becomes a viable option as we see when (3.6) and (3.9) are compared.

In a time-dependent and Newton-Krylov setting as a fully-implicit formulation suggests, the computation of some components of M or M_d might be frozen for a sequence of nonlinear iterations and time steps. Also, since A_h may still have a considerable size the computation of step (1) of the above algorithm may rely on a iterative method, giving then rise to a nested iterative procedure. It is worth to add, however, that the solution associated to system A_h may be suitable for a family of efficient porous media solvers on homogeneous models, such as line correction, ILU, supercoarsening multigrid, domain decomposition and algebraic multigrid (see e.g., [5, 6, 29, 30, 32, 60, 62]).

Another important aspect is that the formulation of (3.1) may be applied in a nested fashion. That is, we can repeat the above procedure for the solution of the A_h. This may be useful if the high permeability block may still have strong permeability contrasts within. This implies to solve a hierarchy of subproblems for which we calculate $\langle K \rangle_i$ at each level i and apply the deflation M_d step as a smoothing step. The procedure is repeated until the underlying subproblem either resembles a purely homogeneous case or is sufficiently small to enable the use of a direct or fast iterative solver.

The preconditioner defined above belongs to an extensive family of two-stage preconditioners [11, 18, 31, 52] for solving fully-coupled systems where each of the variables involved follow different physical behavior (e.g., pressures and saturations sharing the same discretization block). We have been inspired by this idea to define the above preconditioner strategy to simultaneously accommodate full high-permeability solutions with deflated low-permeability solutions. To our knowledge, this is the first time that a two-stage preconditioner approach has been specifically used to deal with high permeability contrasts.

3.2. **Left versus Right Preconditioner.** Let us consider the below decompositions of A:

$$\begin{aligned} A &= \left\{ \begin{bmatrix} I_h & 0 \\ A_{lh}A_h^{-1} & I_l \end{bmatrix} \begin{bmatrix} A_h & 0 \\ 0 & I_l \end{bmatrix} \right\} \begin{bmatrix} I_h & 0 \\ 0 & A_S \end{bmatrix} \begin{bmatrix} I_h & A_h^{-1}A_{hl} \\ 0 & I_l \end{bmatrix} \\ &= M_{\text{left}} \begin{bmatrix} I_h & A_h^{-1}A_{hl} \\ 0 & A_S \end{bmatrix}, \end{aligned} \tag{3.10}$$

$$\begin{aligned} A &= \begin{bmatrix} I_h & 0 \\ A_{lh}A_h^{-1} & I_l \end{bmatrix} \begin{bmatrix} I_h & 0 \\ 0 & A_S \end{bmatrix} \left\{ \begin{bmatrix} A_h & 0 \\ 0 & I_l \end{bmatrix} \begin{bmatrix} I_h & A_h^{-1}A_{hl} \\ 0 & I_l \end{bmatrix} \right\} \\ &= \begin{bmatrix} I_h & 0 \\ A_{lh}A_h^{-1} & A_S \end{bmatrix} M_{\text{right}}. \end{aligned} \tag{3.11}$$

We can define a left and right preconditioner from (3.10) and (3.11), respectively:

$$M_{\text{left}}^{-1} = \begin{bmatrix} A_h^{-1} & 0 \\ -A_{lh}A_h^{-1} & I_l \end{bmatrix}, \tag{3.12}$$

$$M_{\text{right}}^{-1} = \begin{bmatrix} A_h^{-1} & -A_h^{-1}A_{hl} \\ 0 & I_l \end{bmatrix}. \tag{3.13}$$

Then,

$$M_{\text{left}}^{-1}A = \begin{bmatrix} I_h & A_h^{-1}A_{hl} \\ 0 & A_S \end{bmatrix}, \tag{3.14}$$

$$AM_{\text{right}}^{-1} = \begin{bmatrix} I_h & 0 \\ A_{lh}A_h^{-1} & A_S \end{bmatrix}. \tag{3.15}$$

We conclude that the spectra of the preconditioned systems (3.14) and (3.15) are the same:

$$\sigma(M_{\text{left}}^{-1}A) = \sigma(AM_{\text{right}}^{-1}) = \sigma(A_S) \cup \{1\}. \tag{3.16}$$

The main difference between (3.12) and (3.13) is their action on $\begin{bmatrix} x_h \\ x_l \end{bmatrix}$. Equation (3.12) requires only $A_h^{-1}x_h$, whereas (3.13) requires $A_h^{-1}x_h$ and $A_h^{-1}(A_{hl}x_l)$. So, using (3.13) is more costly.

If a stage two preconditioner is introduced, then the preconditioned systems in (3.14) and (3.15) will take the forms

$$M_d^{-1}\left(M_{\text{left}}^{-1}A\right), \tag{3.17}$$

$$\left(AM_{\text{right}}^{-1}\right)M_d^{-1}. \tag{3.18}$$

In order to have a spectrally equivalent preconditioned system as in (3.16), the left preconditioner should be in the immediate left position to A. Similarly, the right preconditioner should be in the immediate right position to A. In (3.17), M_{left}^{-1} and M_d^{-1} become the stage one and stage two preconditioners, respectively. The order of preconditioners is changed in the right preconditioned case. In (3.18), M_d^{-1} and M_{right}^{-1} become the stage one and stage two preconditioners, respectively. We note that the order of stage one and stage two preconditioners is the main difference between left and right preconditioned cases.

3.3. **Spectral Analysis and Matrix Conditioning.** Spectral plots related to A^{orig} and its related subblocks A_h^{orig}, A_l^{orig}, and A_S^{orig} reveal the corresponding condition numbers. Since A^{orig} is symmetric positive definite (SPD), all the related subblocks become SPD. Hence, the condition numbers are simply the ratio of the maximum eigenvalue over the minimum one.

When we introduce diagonal scaling, $A := D^{\text{orig}^{-1}}A^{\text{orig}}$, A is not symmetric, hence, $\kappa(A)$ is not necessarily equal to $\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$. A natural question arises: How are $\kappa(A)$ and $\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ related?

Since A^{orig} is SPD, then $D^{\text{orig}^{1/2}}$ is defined as a real matrix. In order to make this connection, let us introduce $A^{\text{sym}} := D^{\text{orig}^{-1/2}}A^{\text{orig}}D^{\text{orig}^{-1/2}}$. Note that A and A^{sym} are similar through

$$A = D^{\text{orig}^{-1/2}}A^{\text{sym}}D^{\text{orig}^{1/2}}. \tag{3.19}$$

Then, they share the same spectrum:

$$\sigma(A) = \sigma(A^{\text{sym}}). \tag{3.20}$$

Using (3.20) and the fact that A^{sym} is SPD, we get the following:

$$\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \frac{\lambda_{\max}(A^{\text{sym}})}{\lambda_{\min}(A^{\text{sym}})} = \kappa(A^{\text{sym}}). \tag{3.21}$$

Since A and A^{sym} are similar, using (3.19) and the fact that $\kappa^2({D^{\text{orig}}}^{1/2}) = \kappa(D^{\text{orig}})$, we get an upper and a lower bound for $\kappa(A)$

$$\frac{1}{\kappa(D^{\text{orig}})}\ \kappa(A^{\text{sym}}) \leq \kappa(A) \leq \kappa(D^{\text{orig}})\ \kappa(A^{\text{sym}}). \tag{3.22}$$

Equation (3.21) together with (3.22) reveal the connection we are after:

$$\frac{1}{\kappa(D^{\text{orig}})}\ \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \leq \kappa(A) \leq \kappa(D^{\text{orig}})\ \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}. \tag{3.23}$$

$\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ gives a good estimate of $\kappa(A)$ when $\kappa(D^{\text{orig}}) \approx 1$. Since our permeability field is highly heterogeneous, this will hardly happen. Therefore, $\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ does not reflect $\kappa(A)$. Furthermore, subblocks of A are related to that of A^{orig} through the following:

$$\begin{array}{rclcrcl} A_h &=& {D_h^{\text{orig}}}^{-1} A_h^{\text{orig}}, & & A_h &\sim& {D_h^{\text{orig}}}^{-1/2} A_h^{\text{orig}} {D_h^{\text{orig}}}^{1/2} \\ A_l &=& {D_l^{\text{orig}}}^{-1} A_l^{\text{orig}}, & & A_l &\sim& {D_l^{\text{orig}}}^{-1/2} A_l^{\text{orig}} {D_l^{\text{orig}}}^{1/2} \\ A_S &=& {D_l^{\text{orig}}}^{-1} A_S^{\text{orig}}, & & A_S &\sim& {D_l^{\text{orig}}}^{-1/2} A_S^{\text{orig}} {D_l^{\text{orig}}}^{1/2}. \end{array}$$

First of all, each subblock is similar to a SPD matrix, implying that the eigenvalues are real positive. In particular, positive definiteness of A_S comes from the fact that the Schur complement A_S^{orig} of the SPD matrix A^{orig} is also SPD.

For each of the above subblocks, if we employ similar arguments that led to (3.23), we observe that the ratio of maximum over minimal eigenvalues give us good estimate when $\kappa(D_h^{\text{orig}})$ and $\kappa(D_l^{\text{orig}})$ are both close to 1—which is more likely to happen compared to $\kappa(D^{\text{orig}}) \approx 1$ because permeability values are more homogeneous within high and low permeable regions. In short, the spectral values of A and its subblocks do not reveal much in terms of the condition number. That is why we directly computed condition number values as given in Table 5. However, it is not just the condition number that plays the essential role for the convergence of solvers. The distribution of the eigenvalues seems to be crucial as well. Especially, smallest eigenvalues in spectral plots will help us reveal certain aspects of the solver performance. Due to (3.7), we would like to emphasize that the spectrum of A_S directly dictates the spectrum of the preconditioned system. The spectral plot of A_S can help to justify to employ a stage two preconditioner. Namely, if one observes an outlier smallest eigenvalue in spectral plot of A_S, this is the exactly the same eigenvalue which could not be resolved by the stage one preconditioner. Then we can consider to use a stage two preconditioner to address the complications that the outlier smallest eigenvalues can cause.

4. Deflation Methods

4.1. **Fundamentals.** In the recent years, deflation methods have been increasingly receiving particular attention as a way for improving the convergence of linear iterative solvers. Deflation operators provide means to remove the negative effect that extreme (usually small) eigenvalues have in the convergence of Krylov iterative methods for solving symmetric and non-symmetric systems [10, 14, 19, 24, 40, 41, 42, 49, 50]. In most of these research efforts, deflation methods have been developed as a mechanism for systematically expanding and refreshing the underlying Krylov subspace or for conceiving more effective preconditioning techniques. Nevertheless, the use of deflation is not strictly confined to the setting of Krylov subspace iterations (e.g., [8, 22, 58, 27]) or even to the solution of linear system of equations (e.g., [7, 33, 45]).

In addition, a given preconditioner can be complemented with a deflation procedure. In fact, deflation can be used to take care of the "roughness" that was left out upon, for instance, decoupling the system. Thus, deflation allows for knocking down components of the solutions associated with both small and large eigenvalues generated by the permeability contrast. The two well-known strategies for deflation are the Krylov based and the domain based methods. The former utilizes eigenspace information (in the form of Ritz and harmonic Ritz vectors) from the underlying Krylov iterative solver to construct the deflating subspace; see e.g. [24, 41]. Domain based approaches exploit coefficient contrasts to identify subdomain blocks in which the solution demonstrates a certain behavior. Using the two information together geometric or algebraic operators are constructed to approximate eigenvectors; see [22, 56]. This makes the deflation dependent highly on the domain and the behavior of the solution. The main advantage of the domain based deflation methods is that they do not rely on the computation of approximate eigenvalues and eigenvectors. However, we would like to accommodate domains with arbitrarily heterogeneous permeability fields. Therefore, we only consider Krylov based deflation methods in this article.

A typical deflation operator is designed to process the extremal eigenvalues in such a way that the resulting operator will a have better condition number in general. This goal can be accomplished in many ways. For instance, mapping the smallest eigenvalues to 0 or shifting them to 1 or $\lambda_{\max}(A)$. Let $U \in \mathbb{C}^{n\times r}$ be the exact invariant subspace corresponding to r smallest eigenvalues. One type of deflation operator that shifts the r smallest eigenvalues to $|\lambda_{\max}(A)|$ and leaves the rest of the spectrum unchanged is given by [9]:

$$C^{-1} = |\lambda_{\max}(A)|\; U(U^T AU)^{-1}U^T + (I - UU^T), \tag{4.1}$$

where $|\lambda_{\max}(A)|$ is the magnitude of the largest eigenvalue. We utilize the operator in (4.1) as the stage two preconditioner. If U is a near invariant subspace, then depending on the approximation quality of U, the preconditioned matrix AC^{-1} will have eigenvalues close to the set $\{\lambda_{r+1}, \ldots, \lambda_{\max}(A), |\lambda_{\max}(A)|, \ldots, |\lambda_{\max}(A)|\}$.

A preconditioning technique which aims at utilizing the spectral information when restarting would be ideal. The idea is to compute a near invariant subspace corresponding to the smallest eigenvalues. Indeed, the rate of convergence is mostly governed by these smallest eigenvalues [9, 20]. The full-GMRES version behaves as if the smallest eigenvalues were removed after some iterations. But this is no longer true in the restarted case. Therefore, we remove them with the help of a deflation preconditioner. After each cycle of GMRES(m), the preconditioner is updated by pulling out new eigenvalues. At each restart, new approximate eigenvectors are estimated in order to increase the quality of the invariant subspace.

Deflation methods can be classified as *static* (domain-based) or *dynamic* (Krylov-based). In static deflation, the deflation operator is determined before the iteration process starts and remains fixed throughout. In the dynamic version, the deflation operator is regularly updated as the fresh Krylov subspace information is computed. Each class of deflation is appropriate for different scenarios. Static deflation would be effective if *a priori* physical properties are known, thereby intrinsic physics would be retained through the iterative process. Usually, the domain based methods are static deflation methods by their nature. When the domain and solution dependence are removed from the deflation method, we are rely on the information encoded in the near invariant subspace. But, they are algebraic representations of the underlying physics and the two do not exactly correspond. Adapting or updating available physical information can be needed and justifiably dynamic deflation methods seem to be a right choice in such scenarios. Hence, *dynamic deflation* methods are attractive because they have the capability to exploit available useful information in the Hessenberg matrix.

In spite of all the aforementioned advances, the idea of deflating unwanted eigenvectors from the solution is not new. During the end of the 60's, all 70's and beginning of the 80's, deflation was primarily employed for constructing meaningful solutions for (almost) singular linear systems [12, 13, 34, 47, 54]. A novel view of the approach was provided by Nicolaides [44] who fundamentally propose to split the conjugate gradient solution as the sum of a deflated subspace conjugate solution plus a particular solution into the complementary subspace. Nicolaides realized that the new procedure was amenable to use in conjunction with other preconditioners techniques. The idea was further explored by Mansfield in the setting of domain decomposition [35, 36].

4.2. **Deflation Methods under Consideration.**

We consider three well-known deflation methods: harmonic [20], augmented [39], and Burrage-Erhel [9]. In the numerical experiments, we will denote these methods as `harmo, aug, BE`, respectively. All the methods in this article are implemented as preconditioners to `gmres(m)`.

All the deflation methods under consideration utilize a near invariant subspace extracted from the Hessenberg matrix produced by the underlying `gmres(m)` method. Chapman and Saad [15] report that harmonic projection produces approximate eigenvectors that are more accurate than the ones produces by an oblique projection. We prefer to compute the near invariant subspace corresponding to smallest eigenvalues by *harmonic Ritz projection* due to its favorable approximation properties of the eigenvectors corresponding to smallest eigenvalues.

In Algorithms 4.1 and 4.2, we give the description of the harmonic and augmented deflation methods similar to the ones given in [9]. The Burrage-Erhel deflation was introduced in [20] and it is an improved version of the harmonic deflation method. Algorithmically, it is identical to harmonic deflation except that the harmonic Ritz projection is used in the following improved way. A near invariant subspace U_{old} is computed by using harmonic projection of the eigenresidual onto the Krylov subspace just like it is done in the harmonic deflation method. At the end of the cycle, U_{old} is retained and a fresh near invariant subspaces, U_{fresh}, is computed, then orthogonalized against U_{old}. They are appended to form a bigger subspace $[U_{\text{old}}, U_{\text{fresh}}]$. At the end, an other harmonic projection of the eigenresidual is performed onto $[U_{\text{old}}, U_{\text{fresh}}]$ to form the new invariant subspace U_{new}. This way of updating U is slightly more costly but more dynamic and seems to give better convergence rates.

Algorithm 4.1. harmo(m, l) (Deflation by eigenvalue shift) :

```
: convergence := false;
: choose $x_0$;
: $C := I_N$;
: $U := [\ ]$;
: until convergence do
    : $r_0 = b - Ax_0$;
    : apply Arnoldi process to $AC^{-1}$ to compute $V_m$;
    : $y_m = \text{argmin}_{y \in \mathbb{R}^m} \|\beta e_1 - \bar{H}_m y\|$;
    : $x_m := x_0 + C^{-1} V_m y_m$;
    : if $\|b - Ax_m\| <$ tolerance; convergence := true;
    : else
        : $x_0 = x_m$;
        : compute an invariant subspace $U$ of $A$ of size $l$ by harmonic projection;
        : compute $C^{-1} = |\lambda_{\max}(A)|\ U(U^T AU)^{-1} U^T + (I - UU^T)$;
    : endif;
: enddo;
```

Algorithm 4.2. aug(m, l) (Deflation by augmenting) :

```
: convergence := false;
: choose $x_0$;
: $U := [\ ]$;
: until convergence do
    : $r_0 = b - Ax_0$;
    : apply Arnoldi process to $A$ to compute $V_m$;
    : $W = [V_m, U]$;
    : compute $AU$;
    : orthogonalize $AW$ to get $V$;
    : $y_m = \text{argmin}_{y \in \mathbb{R}^{m+l}} \|\beta e_1 - \bar{H} y\|$;
    : $x_m := x_0 + W y_m$;
    : if $\|b - Ax_m\| <$ tolerance; convergence := true;
    : else
        : $x_0 = x_m$;
        : compute an invariant subspace $U$ of $A$ of size $l$ by harmonic projection;
    : endif;
: enddo;
```

In the original construction harmonic and Burrage-Erhel methods are designed to be right preconditioners. In the right preconditioned case, the system becomes $AC^{-1}u = b$, where $x = C^{-1}u$. The residual expression is the following:

$$r_i = b - AC^{-1}u_i = b - Ay_i.$$

Assuming a restart at every m-th iteration, at iteration i the GMRES algorithm finds $\|\bar{r}_i\| = \|b - Ax_i\|$:

$$\|\bar{r}_i\| = \min_{y_i \in \kappa_{\mathcal{J}(i)}(A, r_0)} \|r_i\|,$$

where

$$\mathcal{J}(i) = \begin{cases} i, & i \neq 0 \bmod m \\ m, & i = 0 \bmod m \end{cases}$$

and in the next iteration $i+1$, the vector $A^{\mathcal{J}(i+1)-1}r_0$ is added to Krylov subspace. Within each cycle, the Krylov subspace over which we search the minimum gets bigger, this implies the well-known nonincreasing residual:

$$\|\bar{r}_{i+1}\| \leq \|\bar{r}_i\|, \text{ for } i \text{ that stays in the same cycle.} \tag{4.2}$$

After a restart, for instance at iteration $m+1$, computations take place in then new cycle as we look for

$$\|\bar{r}_{m+1}\| = \min_{y_{m+1} \in \kappa_1(A, r_0')} \|r_{m+1}\|, \tag{4.3}$$

where $r_0' = \bar{r}_m$ due to assigning $y_{m+1} := x_m$ before the start of the new cycle. (4.3) implies that $\|\bar{r}_{m+1}\| \leq \|b - Ay_{m+1}\|$ for any $y_{m+1} \in \kappa_1(A, r_0')$, in particular for x_m. Therefore, in exact arithmetic, GMRES(m) generates a nonincreasing residual for the iteration between two cycles as well:

$$\|\bar{r}_{m+1}\| \leq \|b - Ax_m\| = \|\bar{r}_m\|. \tag{4.4}$$

In dynamic deflation, the preconditioner C^{-1} is updated at every cycle. But the residual expression does not contain C^{-1} in the right preconditioned case. In the left preconditioner case, the system is $C^{-1}Ax = C^{-1}b$. Then, $r_i = C^{-1}b - C^{-1}Ay_i = b - Ay_i$. In the deflation methods used, C^{-1} depends on the near invariant subspace. Since, the near invariant subspace is updated at the end of each cycle, the preconditioner does not change within the cycle. At the restart, C^{-1} is updated, hence (4.4) does not hold. So, in exact arithmetic, one can expect increase at the restart but (4.2) will hold within each cycle. This explains the oscillations we observe for the left preconditioned case.

In conclusion, we observe in the numerical experiments that the performance of left and right preconditioners are similar. However, the left preconditioner is less costly as explained in §3.2, but can create increasing residual at the GMRES(m) restarts due to the updated near invariant subspace.

5. Numerical Experiments

The goal of the numerical experiments is to establish that our physics-based two-stage preconditioner is an effective alternative for porous media flow applications in which our permeability-based heuristic holds. Since the preconditioner is a two-stage preconditioner it has two components. The main component (in the left preconditioned case) is the preconditioner in (3.2) or in (3.12). If needed, a stage two preconditioner can be employed as in (3.17) to complement the stage one preconditioner. The stage two preconditioner chosen is one of the mentioned deflation methods. Our two-stage preconditioner has also a version based on right preconditioning. The one stage right preconditioner is given in (3.13) and its two stage counterpart is given (3.18). The right preconditioner will give rise to nonincreasing residual in exact arithmetic whereas the residual in the left preconditioner can have oscillations because the residual at the restarts can increase due to the updated near invariant subspace as explained in §4.2.

The methods employed in this paper have been implemented in MATLAB 7.1. We define the preconditioner effectiveness as the rate of decay of the relative residual with respect to matrix-vector multiplications (MVPs). For each test problem, we report the residual and error convergence history and the iteration counts for the left and the right preconditioned cases. In a given iteration plot, we report only the methods that converged. If there is no corresponding iteration count, this means that the method did not meet the convergence criterion. The stopping criterion is chosen to be $\|\frac{r_m}{r_0}\|_2 < 1.0e-10$.

Table 1 summarizes the characteristics associated with each of the 7 cases. As we can see, the cases represent different permeability distribution geometries and contrasts. Consequently, the condition number of the associated matrix varies and it is relatively high despite the modest size of the cases considered. We have listed the condition numbers associated with the most relevant blocks without and with diagonal scaling. We can clearly observe the crucial role that scaling and the proposed physics-based two-stage preconditioning play in decreasing the condition number of the original matrix. In the results shown in the table, $M_d = I$ (i.e., the two-stage preconditioner only includes high-permeability block solutions).

The system matrices for test problems 2–7 are the discretization matrices of one phase flow and are generated by the reservoir simulator written in MATLAB by Aarnes, Gimse, and Lie [2]. In test problems 2–5, the flow is directed from one corner to the opposite diagonal corner through a highly permeable layer-like channel (see Figures 6, 8, 10, 12.) Test problem 1 is generated by using IPARS [61] in which a two phase flow in 3D takes place in a depleted reservoir with a production well located at the center. In each numerical experiment, the matrix corresponds to the pressure block in a pressure-saturation coupled system of a fully implicit discretization of the underlying PDE systems. All of the system matrices, A^{orig}, are symmetric positive definite, diagonally dominant, and highly ill-conditioned.

5.1. **Spectral Analysis and Observations.** In this section, we outline some of the connections between the permeability field and the spectrum. There are only two permeability values (high and low value) in test problems 1, 2, 3, 6, and 7, whereas in test problems 4 and 5 we allow variation in the permeability values within the high and low permeable regions. There is several orders of magnitude difference between these eigenvalues similar to the permeability values. The entries of A^{orig} corresponding to A_h^{orig} are larger than that of A_l^{orig} due to high values of permeability in the high permeable region.

The severe contrast in permeability values creates two main clusters of eigenvalues of A^{orig}: large and small. Our immediate observation is that the large and small eigenvalues of A^{orig} correspond to eigenvalues of A_h^{orig} and A_l^{orig}, respectively.

When the permeability contrasts can be identified as layer-like channels, the cluster of large and small eigenvalue is well separated. The separation becomes more apparent when the permeability values are homogeneous within the high and low permeable regions (see Figures 4, 6, 8, 14) and the separation diminishes with less homogeneous permeability values (see Figure 12) and almost disappears (see Figure 10).

When the separation of high and low permeable regions are layer-like as in the case of a stratified reservoir, there are one or more eigenvalues between the the large and small clusters (see Figures 4, 6, 8, 12). We claim that these eigenvalues are associated to the permeability contrast and are captured by the eigenvalues of A_h^{orig}. $\sigma(A_h^{\text{orig}})$ and $\sigma(A_l^{\text{orig}})$ share common eigenvalues in the small valued cluster (see Figures 6, 8, 12). Comparing Figures 6, 10, and 12, we observe that the number of shared eigenvalues increase as the the variation in the permeability values increases.

We emphasize that the above observations are valid when the permeability contrast can be identified as layer-like channels where the high permeable regions are sandwiched by low permeable regions. One can view these scenarios as generalizations of a stratified reservoir with alternating permeability values as in Figure 4. When, this assumption is not in place, our heuristic for the separation of eigenvalue clusters is not necessarily valid as seen in Figure 16.

5.2. **Effects of Diagonal Scaling and Collecting Small Eigenvalues into a Subblock.** When diagonal scaling is introduced, eigenvalues of A_h capture most of the main features of the spectrum of A, especially the smallest eigenvalues of A; see Figures 6, 8, etc. We observe that diagonal scaling greatly helps collecting smallest eigenvalues of A in A_h. Therefore, when we introduce diagonal scaling, the main motivation is to find smallest eigenvalues of A that are responsible for permeability contrasts. Once these small eigenvalues are found, the preconditioner will target them to eliminate their complications for the solver. The existing works by Vuik et.al. [57, 59] and Graham and Hagger [26] explain the effects of diagonal scaling in the case of diffusion equation with highly contrasting diffusion coefficients. In particular, the number of high permeable regions in the reservoir that are sandwiched by low permeable region gives the exact the number of smallest eigenvalues of A.

Next, we report some of our observations and state same heuristics on the behavior of smallest eigenvalues in the spectrum. The number of well-identifiable high permeability regions sandwiched by low permeability regions corresponds to the number of smallest eigenvalues of A. For instance, one smallest eigenvalue in test problems 2 and 5 (see Figures 6 and 12) and several smallest eigenvalues in 1, 3, 4, 6, and 7 (see Figures 4, 8, 10, 14, and 16). In all these cases, smallest eigenvalues of A are captured as the smallest eigenvalues of A_h subblock. For instance, they are very well captured in Figures 4, 6, 12, well in Figure 8, somewhat well in Figures 10, 14, 16.

We observe that A_h and A_S behave as if they are spectral complements in the following sense. If smallest eigenvalues of A are well captured by A_h, then the Schur complement is free from smallest eigenvalues (as in Figures 4, 6, 8, 12, 16), and by (3.16), so is the preconditioned system (see for example Figure 12). If not (see Figures 10, 14), we employ a deflation method as stage two preconditioner on top of M_{left}^{-1}. In Figures 10 and 14, we illustrate that several smallest eigenvalues of A of varying magnitude are not captured by A_h, they show up as smallest eigenvalues of A_S in varying magnitude. This is exactly where we employ a stage two preconditioner and we show that this strategy is effective as in test problem 6 and Figure 15.

TABLE 1. The numbers of degrees of freedom and condition numbers.

Test Problem	1	2	3	4	5	6	7
N	1100	1600	1600	1600	1600	1600	1600
$N_h; N_l$	300;800	340;1260	421;1179	457;1143	462;1138	562;1038	747;853
K_{min}	1.00e-03	1.00e-04	1.00e-04	1.00e-07	1.00e-02	3.16e-02	1.00e-01
K_{max}	2.00e+06	1.00e+04	1.e+04	3.98e+04	1.00e+05	1.00e+05	1.00e+03
$\langle K \rangle$	3.40e-01	2.20e-03	5.01e-04	2.29e-02	1.66e-01	6.03e+00	7.41e+00
cond(A)	1.92e+04	1.65e+05	2.54e+11	1.76e+13	1.33e+10	5.89e+10	1.72e+06
cond(A_h)	7.12e+03	2.92e+04	2.08e+09	4.52e+10	4.58e+08	8.43e+06	8.07e+04
cond(A_l)	2.89e+00	1.64e+03	1.05e+03	1.54e+05	1.91e+04	8.08e+00	3.27e+02
cond(A_S)	2.90e+00	1.77e+03	2.09e+03	3.17e+06	3.62e+04	1.34e+04	3.46e+02
cond($M_{\text{left}}^{-1}A$)	2.92e+00	2.00e+03	1.23e+04	4.69e+07	3.18e+05	3.23e+04	9.45e+02
cond(A^{orig})	1.92e+05	1.65e+11	2.34e+11	8.29e+12	1.57e+10	3.20e+10	3.36e+06
cond(A_h^{orig})	6.57e+03	4.35e+08	4.04e+09	1.61e+12	2.62e+08	7.37e+06	4.84e+04
cond(A_l^{orig})	4.03e+00	1.52e+03	9.81e+02	3.22e+05	2.08e+04	1.07e+01	3.09e+02
cond(A_S^{orig})	4.09e+00	1.64e+03	1.93e+03	1.89e+06	3.56e+04	1.09e+04	5.07e+02
cond($M_{\text{left}}^{\text{orig}^{-1}} A^{\text{orig}}$)	5.92e+03	2.07e+06	2.67e+07	1.60e+09	4.58e+05	6.90e+04	2.19e+03

For a fair comparison between the methods, we need to explain clearly what the Krylov subspace size means for each preconditioner. When a deflation method is employed, the Krylov subspace employed has two pieces. The main piece has $m - l$ vectors and the restarted GMRES uses these vectors. The other piece has l vectors and we extract a near invariant subspace of size l corresponding to the smallest eigenvalues. For instance, $m + l = 50$ means, `harmo, aug, BE` preconditioners are operating on a Krylov subspace of size 40 and the near invariant subspace is extracted by using a collection of $l = 10$ vectors in the harmonic Ritz projection. For all the experiments, the ratio of the Krylov subspace size over the near invariant subspace size is maintained as $m : l = 4 : 1$. Since, deflation preconditioners `2Stage+d(40,10)`, `harmo(40,10)`, `aug(40,10)` and `BE(40,10)` use 50 vectors, we compare them against `gmres(50), 2Stage(50)`. In the convergence plots, we always report the case $m+l = 40+10$. The iteration count plots contain the following m, l combinations: $8, 2$; $12, 3$; $16, 4$; $20, 5$; $24, 6$; $28, 7$; $32, 8$; $36, 9$; $40, 10$; $44, 11$; $48, 12$.

The deflation methods we compare are given in §4.2. These methods are `harmo(m,l)`, `aug(m,l)`, and `BE(m,l)`. The comparisons are made against `gmres(m+l)`, `2Stage(m+l)`, and `2Stage+d(m,l)` where `2Stage(m+l)` means the two-stage preconditioner without a stage two preconditioner and `2Stage+d(m,l)` means that the Algorithm 4.1 is used as a stage two preconditioner.

There are two important features with 2Stage preconditioner. The first one is the number of degrees of freedom to be marked as high permeable should be small compared to the total number of unknowns. In this work, we are using the geometric averaging as a way to separate the high permeability coefficients from the small ones. In general, we have to be careful to decide the threshold (averaging) procedure that suits best the separation between these regions. Additionally, high permeability regions may be disconnected and create difficulties in the solution of corresponding block A_h. A finer procedure based on the conductivity should basically place these coefficients out of block A_h since they do not contribute in the physical behavior of flow.

The other main feature is to be able to solve a system in the subblock A_h. We expect to collect smallest eigenvalues in A_h, thereby, this subblock contains the difficult part of the underlying problem. Hence, A_h is ill-conditioned but small in size. In a recursive manner, further ordering can be applied to DOF in A_h if there is extra variation in the high permeability values. This makes the size of A_h even smaller, hence, its system solve easier. One can utilize various solvers considering the small size of A_h such as direct methods or iterative methods like the deflation methods and AMG. For convenience, the system solve of A_h is done by using the backslash solver in MATLAB.

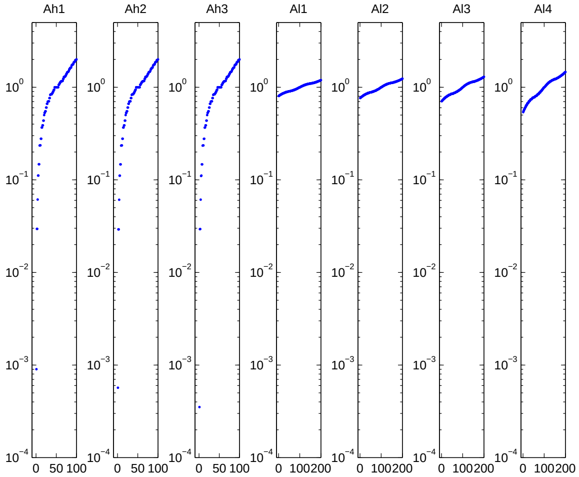

FIGURE 3. Test problem 1: Spectra corresponding to 7 layers.

5.3. **Preconditioning Results on Different Test Problems.** We now proceed to describe in detail the results obtained for each test problem.

5.3.1. *Test problem 1.* This problem represents a two-phase flow in the so-called *3D pancake topology* and the data is generated by IPARS [61]. The reservoir consists of 7 layers with contrasting low and high permeability values of $1.0e-3$ and $2.0e+6$, respectively. There are 3 layers of thin high permeable sandwiched by 4 layers of thick low permeable layers (see Figure 4). Thin and thick layers consist of 100 and 200 degrees of freedom, respectively. This means that $N_h = 300$ and $N_l = 800$ (see Figure 3). The permeability field can be determined directly by the underlying geometry without employing our marking strategy.

First of all, we observe that $\sigma(A_h^{\text{orig}})$ and $\sigma(A_l^{\text{orig}})$ are disjoint. This can be an indication that the coupling between A_h^{orig} and A_l^{orig} is weak. In other words, they exhibit a behavior what we call as *spectral complementarity*. When there is no coupling between A_h^{orig} and A_l^{orig}, then they become literally *spectral complements*:

$$\sigma(A_h^{\text{orig}}) = \sigma(A^{\text{orig}}) \setminus \sigma(A_l^{\text{orig}}). \quad (5.1)$$

When the coupling is weak, we can observe a similar behavior. However, (5.1) does not hold in the exact sense but approximately. By abusing the concept in (5.1), we load more meaning to *spectral complementarity* when we compare A_h and A_S. If smallest eigenvalues of A are well captured by A_h, then the Schur complement A_S is free from smallest eigenvalues. In this case, we also identify A_h and A_S as *spectral complements*.

For the three high permeable layers, we see exactly 3 smallest eigenvalues of A. This is exactly the same behavior reported [26, 57, 59] for the diffusion equation. The 3 smallest eigenvalues of A fully captured by A_h, hence, we report a perfect example of spectral complementarity between A_h and A_S. Due to this fact, `2Stage(50)` preconditioner enjoys a constant number of 19 iterations independent of the size of the Krylov subspace m. Since A_S is free from small eigenvalues, there is no need for a stage two preconditioner. This is justified by observing that there is no convergence improvement when stage two preconditioner is introduced.

All the deflation methods converge exhibiting roughly the same convergence behavior and outperform the `gmres(m)` method. However, `aug(m,l)` shows sign of instability. Our preconditioners outperform the deflation methods by observing that the decay of the residual and error of the `2Stage(50)` and `2Stage(40,10)` preconditioners are the fastest; see Figure 5. The left and right preconditioned versions exhibit similar behavior.

5.3.2. *Test problem 2.* The permeability field is designed to form a high permeability channel crossing the reservoir diagonally. Since this channel is sandwiched by low permeable regions, we expect only 1 smallest eigenvalue in A; see Figure 6. Indeed, we see an eigenvalue of A of magnitude $O(e-04)$ which is fully captured by A_h, thereby, $\lambda_{\min}(A_S) = O(e-03)$ and $\kappa(A_S) = O(e+03)$.

Deflation methods almost always converged with the exception of `aug(m,l)` and `BE(m,l)` is the most effective. Our preconditioners outperform deflation methods and employing the stage two preconditioner accelerates the convergence; see Figure 7. `2Stage+d(40,10)` enjoys the fastest convergence among the methods with $(m,l) = (40,10)$. The left and right preconditioned versions exhibit similar behavior.

5.3.3. *Test problem 3.* The permeability field is designed to form two high permeability channels; see Figure 8. The channels contain disconnected pieces, hence, we expect several smallest eigenvalues of A. Indeed, there is a cluster of smallest eigenvalues of A which ranges between $O(e-10)$ to $O(e-07)$. This problem is a difficult one considering the mentioned cluster and $\kappa(A) = O(e+11)$. A_h captures smallest eigenvalues of $O(e-08)$ and misses the smaller ones. This yields several small eigenvalues of $O(e-03)$ in A_S.

Considering the difficulty of the problem, `harmo(m,l)` ran into numerical problems during the computation of the harmonic Ritz projection. The other methods did not have such troubles. However, none of the deflation methods converged upto the desired accuracy both in the left and right preconditioned versions; see Figure 9. Furthermore, we observe a remarkable error reduction for `2Stage(40,10), 2Stage(50)` preconditioners, where as for the rest it did not decrease at all. This is a very good indication that our preconditioners are robust. Our left preconditioner performs better than the right one due to stall in both the error and residual before reaching the desired accuracy. In addition, the stall comes much later when stage two preconditioner is employed.

5.3.4. *Test problem 4.* The permeability field is designed to test if our preconditioners would work when there is a large variation in the permeability values. So both high and low permeability regions contain highly heterogeneous values. Given that $\kappa(A) = O(e+13)$ and that there is a cluster of smallest eigenvalues of A which ranges between $O(e-10)$ to $O(e-05)$, the system matrix is the most difficult among the test problems. The main feature that causes these difficulties is the highly heterogeneous permeability field. Employing M_{left}^{-1} brings $\kappa(A)$ down to $\kappa(M_{\text{left}}^{-1}A) = O(e+07)$. However, A_h cannot capture the smallest eigenvalues of A, thereby, an eigenvalue of $O(e-05)$ is formed in A_S and that seems to be the main reason for the convergence failure for `2Stage(50)`. Employing the stage two preconditioner does not seem to eliminate the complication. We observe a reduction in the norm of the error of `2Stage+d(40,10)` whereas all the other methods fail. Left preconditioned `2Stage+d(40,10)` seems to reduce the error more than the right version. In summary, none of the methods can solve this hard problem.

5.3.5. *Test problem 5.* This test problem is designed in such a way that the heterogeneity in the permeability field of test problem 4 is decreased. This decreases the condition number, $\kappa(A) = O(e+10)$, and only one eigenvalue of $O(e-09)$ appears in $\sigma(A)$. One can still say that test problem 5 is among the difficult ones.

A_h is very successful in capturing the smallest eigenvalues of A, thereby, $\lambda_{\min}(A_S) = O(e-03)$ and $\kappa(A_S) = O(e+04)$. For the left preconditioned case, `2Stage(50)` converges and `2Stage+d(40,10)` converges faster. `2Stage+d(m,l)` converges for a bigger set of (m,l) which makes it more robust with respect to the sizes of the Krylov subspace and the near invariant subspace. Among the deflation methods, `BE(m,l)` seems to be the only converging one. For the left preconditioned case, the error reduction for `2Stage+d(40,10)` is better than that of `BE(40,10)`. We notice that left preconditioning is more effective and results in more convergence when stage two preconditioner is employed.

5.3.6. *Test problem 6.* This test problem contains a permeability field whose topology resembles a checker board. One can still identify layer-like channels between the opposite corners. This test problem is designed to test if our heuristics about the smallest eigenvalues is valid in this extreme case. Namely, when there is layer-like channels, we can identify contrasting layers, hence, smallest eigenvalues arise due to these contrasts.

Test problem 6 is a difficult one with $\kappa(A) = O(e+10)$ and A has a cluster of smallest eigenvalues which ranges between $O(e-09)$ to $O(e-06)$. Among the smallest eigenvalues , there is only one outlier with magnitude $O(e-09)$ and A_h cannot capture that. This is reflected to A_S as an outlier eigenvalue where $\lambda_{\min}(A_S) = O(e-04)$. However, A_h captures smallest eigenvalues of A of $O(e-06)$ which gives rise to a favorable condition number; $\kappa(M_{\text{left}}^{-1}A) = O(e+04)$. We observe a very effective `2Stage+d(m,l)` for both left and right preconditioned cases whereas all the deflation methods fail. Furthermore, in the left preconditioned case, `2Stage(50)` converges but slower than `2Stage+d(40,10)`. `2Stage(m+l)` does not converge for most of (m,l) combinations, employing a stage two preconditioner brings robustness and `2Stage+d(m,l)` converges for almost all (m,l) combinations in both left and right preconditioned cases. This is an effective use of the stage two feature of our preconditioner which addresses the smallest eigenvalues of A_S. In the left preconditioned case, we notice a more improved convergence when the stage two preconditioner is employed compared to right preconditioned case. However, right preconditioner gives a constant number of 36 iterations for a large combination of (m,l) values where as left preconditioner iteration counts are in the vicinity of 400.

5.3.7. *Test problem 7.* We use a SPE10 data set to create this test problem with a mildly difficult condition number; $\kappa(A) = O(e+06)$. Our heuristics about the eigenvalues are not necessarily valid due to the absence of layer-like channels. Nevertheless, A_h captures most of the smallest eigenvalues of A, thereby, $\kappa(M_{\text{left}}^{-1}A) = O(e+03)$. There is an outlier smallest eigenvalue of A_S of $O(e-02)$ which is not small enough to cause convergence difficulties for `2Stage(m+l)` and `2Stage+d(m,l)`. In conclusion, `2Stage(m+l)` and `2Stage+d(m,l)` converge for almost every (m,l) combination for both left and right preconditioned cases, whereas deflation methods fail with the exception that `BE(m,l)` show a sign slow convergence. The left and right preconditioned versions exhibit similar behavior.

6. Conclusions

In this article, we present two-stage physics-based preconditioners that are designed to address severe contrasts in the underlying physical quantities such as permeability. The contrasts give rise to extremely small eigenvalues and they seem to be the main bottleneck for iterative solvers. The application of interest is single- or multi-phase flow in porous media where jumps in the PDE coefficients come from the contrasts in the permeability field.

The main objective of the present work is to introduce a novel physics-based preconditioning strategy for solving problems with high physical contrasts in porous media applications. These stringent situations commonly arise, for example, in multilayered geological formations composed of different type of rocks. We assume that the porous media consist of highly permeable interconnected regions allowing for a strong global flow conductivity (e.g., channelized media). Therefore, the main assumption behind this article is that permeability is the fundamental physical quantity that basically defines and governs flow trends in porous media. Hence, permeability should dictate the way we solve linear equations. Figure 1 illustrates this type of permeability distribution settings that we are interested in handling efficiently from the iterative solution standpoint. We propose a two-stage physics based preconditioner with an optional stage two deflation preconditioner as outlined in Algorithm 3.1.

The matrices under investigation correspond to the pressure block in a pressure-saturation coupled system of a fully implicit discretization of the underlying PDE system and they are symmetric positive definite, diagonally dominant, and highly ill-conditioned.

Since, historically deflation methods are designed to address extremal eigenvalues, we compare our preconditioner to these methods as well as use them as an optional stage two preconditioner. We compare our preconditioner to three well-known deflation methods: harmonic [20], augmented [39], and Burrage-Erhel [9]. These are *dynamic* deflation methods where the near invariant subspace U is extracted by a harmonic Ritz projection from the Hessenberg matrix. This is the standard way of computing the near invariant subspaces in all of our numerical experiments. We report that our preconditioners–even without stage two preconditioner–outperform all of the three methods. The best competitor and the most unstable methods are `Burrage-Erhel(m,l)` and `augment(m,l)`, respectively. Furthermore, our preconditioners are robust with respect to the Krylov subspace size. We report convergence for far more combinations of (m, l) compared to deflation methods.

The selection of DOF in A_h is purely algebraic, hence, we emphasize that our preconditioners are algebraic in nature. The biggest advantage of our preconditioners is the fact that they can handle *flexible* and *realistic* reservoir topology. Since, a typical dynamic deflation method is also algebraic in nature and our two-stage preconditioners allow the use of a stage two preconditioner, a dynamic deflation method becomes a suitable choice for a stage two preconditioner in our framework. Performance of left and right preconditioners are similar. Left preconditioner is less costly but can create increasing residual at the GMRES(m) restarts due to the updated near invariant subspace.

Despite the effectiveness of the proposed physics-based preconditioners there are many research issues that remain open. We list some of the one that we consider promising to address in the near future:

- In some stringent situations, deflation may not be sufficiently robust as a second stage preconditioner in the advent of either extreme ill-posedness (due to inner significant heterogeneities in low- or high-permeability regions) or to the size of the block A_l. Thus, we need to explore variations of the method such as multi-stage preconditioning and the use of solvers such as algebraic multigrid (AMG), algebraic multilevel iterations (AMLI) and sparse approximate inverse (SPAI) methods to hopefully speedup the solution of both the A_h and A_l blocks.
- Our method is based on static permeability information and does not account for a better assessment of the true media connectivity. We believe that streamlines or percolative methods should be very useful in defining improved physics-based preconditioning strategies since they can detect preferential flow paths in a more reliable way.
- Connections of nested versions of the present methodology (i.e., multi-stage preconditioning) with AMG methods seems to be in order to design improve solution heuristics according to the connectivity strength between matrix coefficients and the underlying physical domain.
- Last but not least, many researchers are devoted to the development of accurate and efficient multiscale methods. We believe that our method can be seen as high-level approach where these methods can be further extended to tackled fine scale solutions such as in [1, 25, 28].

Acknowledgements

The authors want to thank Ivan Graham, Robert Scheichl, and Jan Van Lent from the University of Bath at UK, for the interesting discussions on the addressed topic during their visit to CSM.

References

[1] J. E. Aarnes, *On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation*, Multiscale Model. and Simul., 2 (2004), pp. 421–439.

[2] J. E. Aarnes, T. Gimse, and K.-A. Lie, *An introduction to the numerics of flow in porous media using Matlab*, in Geometrical Modeling, Numerical Simulation, and Optimization: Industrial Mathematics at SINTEF, Springer Verlag, 2007.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[4] K. Aziz and A. Sethari, *Petroleum Reservoir Simulation*, Applied Science Publisher, 1983.

[5] P. Bastian and R. Helmig, *Efficient fully-coupled solution techniques for two-phase flow in porous media: Parallel multigrid solution and large scale computations*, Advances in Water Resources, 23 (1999), pp. 199–216.

[6] R. Bhogeswara and J. E. Killough, *Domain decomposition and multigrid solvers for flow simulation in porous media on distributed memory parallel processors*, Journal of Scientific Computing, 7 (1992), pp. 127–162.

[7] L. Blank, *Preconditioning via a Schur complement method: An application in state estimation*, SIAM J. Sci. Comput., 25 (2003), pp. 942–960.

[8] K. Burrage and J. Erhel, *On the performance of various adaptive preconditioned GMRES strategies*, Numer. Linear Alg. Appl., 5 (1998), pp. 101–121.

[9] ———, *On the performance of various adaptive preconditioned GMRES strategies*, Numer. Linear Alg. Appl., 5 (1998), pp. 101–121.

[10] C. L. Calvez and B. Molina, *Implicit restarted and deflated GMRES*, Numer. Alg., 21 (1999), pp. 261–285.

[11] H. Cao, H. Tchelepi, J. Wallis, and H. Yardumian, *Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation* , in SPE Annual Technical Conference and Exhibition, S. of Petroleum Engineers, ed., Dallas, Texas, 9-12 October 2005.

[12] T. Chan, *Deflated decomposition of solutions of nearly singular systems*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 121–134.

[13] ———, *Deflation techniques and block-elimination algorithms for solving bordered singular systems*, SIAM J. Numer. Anal., 21 (1984), pp. 738–754.

[14] A. Chapman and Y. Saad, *Deflated and Augmented Krylov Subspace Techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[15] ———, *Deflated and Augmented Krylov Subspace Techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[16] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, 2005.

[17] Z. Chen, G. Huan, and Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, SIAM, 2006.

[18] C. Dawson, H. Klíe, M. Wheeler, and C. Woodward, *A Parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver*, Comp. Geosciences, 1 (1997), pp. 215–249.

[19] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. of Comput. and Appl. Math., 69 (1996), pp. 303–318.

[20] ———, *Restarted GMRES preconditioned by deflation*, J. of Comput. and Appl. Math., 69 (1996), pp. 303–318.

[21] J. R. Fanchi, *Principles of Applied Reservoir Simulation*, Gulf Publishing, 2000.

[22] J. Frank and C. Vuik, *On the construction of deflation-based preconditioners*, SIAM J. SCi. Comput., 23 (2001), pp. 442–462.

[23] L. Fung and A. Dogru, *Parallel Unstructured Solver Methods for Complex Giant Reservoir Simulation*, in SPE Reservoir Simulation Symposium, S. of Petroleum Engineers, ed., Houston, Texas, Feb. 26-28 2007.

[24] S. Goossen and D. Roose, *Ritz and harmonic Ritz values and the convergence of FOM and GMRES*, Numer. Linear Alg. Appl., 6 (1999), pp. 281–293.

[25] I. Graham, P. Lechner, and R. Scheichl, *Domain decomposition for multiscale PDEs*, Tech. Rep. Preprint 11/06, Bath Institute For Complex Systems, University of Bath, UK, 2006. available at http://www.bath.ac.uk/math-sci/BICS.

[26] I. G. Graham and M. J. Hagger, *Unstructured additive Schwarz-conjugate gradient method for elliptic problems with highly discontinuous coefficients*, SIAM J. Sci. Comp., 20 (1999), pp. 2041–2066.

[27] W. H, J. Fish, R. Tuminaro, and J. Shadid, *The Generalized global basis (GBB) method*, Int. J. Numer. Meth. Engng., 61 (2004), pp. 1243–1269.

[28] T. Y. Hou and X. H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comput. Phys., 134 (1997), pp. 169–189.

[29] J. Jones and C. S. Woodward, *Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems*, Advances in Water Resources, 24 (2001), pp. 733–774.

[30] J. Killough and M. Wheeler, *Parallel iterative linear equation solvers: An investigation of domain decomposition solvers for reservoir simulation*, in Ninth SPE Symposium on Reservoir Simulation, SPE paper no. 16021, San Antonio, Texas, 1987.

[31] H. Klie, *Krylov-Secant methods for solving Large Scale Systems of Coupled Nonlinear Parabolic Equations*, PhD thesis, Dept. of Computational and Applied Mathematics , Rice University, Houston, TX, 1996.

[32] S. Lacroix, Y. Vassileski, J. Wheeler, and M. Wheeler, *Iterative solution methods for modeling multiphase flow in porous media fully implicitly*, SIAM J. Sci. Comput, 25 (2003), pp. 905–926.

[33] K. Lust, D. Roose, A. Spence, and A. R. Champneys, *An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions*, SIAM J. Sci. Comput., 19 (1998), pp. 1188–1209.

[34] M. Lynn and W. Timlake, *The use of multiple deflations in the numerical solution of singular systems of equations, with applications to potential theory*, SIAM J. Numer. Anal., 5 (1968), pp. 303–322.

[35] L. Mansfield, *On the conjugate gradient solution of the Schur complement system obtained from domain decomposition*, SIAM J. Numer. Anal., 27 (1990), pp. 1612–1620.

[36] ———, *Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 1314–1323.

[37] C. Mattax and R. Dalton, *Reservoir Simulation*, vol. 13, SPE–Monograph Series, Richardson, TX, 1990.

[38] J. Monteagudo and A. Firoozabadi, *Numerical Simulation of Water Injection in Disconnected and Connected Fractured Media using Jacobian-Free Fully Implicit Control Volume Method*, in 14th Symposium on Improved Oil Recovery, SPE paper No. 89449, Tulsa, OK, 2004.

[39] R. B. Morgan, *A Restarted GMRES Method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[40] R. B. Morgan, *Implicit restarted GMRES and Arnoldi Methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.

[41] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Statist. Comput., 24 (2002), pp. 20–27.

[42] R. B. Morgan, *Restarted block-GMRES with deflation of eigenvalues*, Appl. Numer. Math., 28 (2004), pp. 1–15.

[43] R. Nabben and C. Vuik, *A comparison of deflation and coarse grid correction applied to porous media flow*, SIAM J. Numer. Anal., 42 (2004), pp. 1631–1647.

[44] R. Nicolaides, *Deflation of conjugate gradients with applications to boudary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.

[45] T. Noorden, S. Verduyn, and A. Bliek, *A Broyden rank p+1 update continuation method with subspace iteration*, SIAM J. Sci. Comput., 25 (2004), pp. 1921–1940.

[46] P. Renard, G. L. Loc'h, G. de Marsily, and R. Mackay, *A fast algorithm for the estimation of the equivalent hydraulic conductivity of heterogeneous media*, Water Resources Research, 36 (2000), pp. 3567–3580.

[47] W. Rheinboldt, *Numerical methods for a class of finite dimensional bifurcation problems*, SIAM J. Numer. Anal., 15 (1977), pp. 1–11.

[48] M. Ronayne and S. Gorelick, *Effective permeability of porous media containing branching channel networks*, Physical Review E, 73 (2006), pp. 1–10.

[49] Y. Saad, *Analysis of Augmented Krylov Subspace Methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.

[50] Y. Saad, M. Yeung, J. Erhel, and F. Guyomar'h, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[51] X. Sanchez-Vila, A. Guadagnini, and J. Carrera, *Representative Hydraulic Conductivities in Saturated Groundwater Flow*, Reviews of Geophysics, 44 (2006), pp. 1–46.

[52] R. Scheichl, R. Masson, and J. Wendebourg, *Decoupling and block preconditioning for sedimentary basin formulations*, Computational Geosciences, 7 (2003), pp. 295–318.

[53] ———, *Decoupling and block preconditioning for sedimentary basin formulations*, Computational Geosciences, 7 (2003), pp. 295–318.

[54] G. Stewart, *On the implicit deflation of nearly singular systems of linear equations*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 136–140.

[55] K. Stuben, T. Kees, H. Klie, and M. Wheeler, *Algebraic Multigrid Methods (AMG) for the Efficient Solution of Fully Implicit Formulations in Reservoir Simulation*, in SPE Reservoir Symulation Symposium, Houston, Texas, Feb. 26-28 2007, SPE paper No. 105832.

[56] F. Vermolen, C. Vuik, and A. Segal, *Deflation in preconditioned conjugate gradient methods for finite element problems*, in Conjugate Gradient and Finite Element Methods, Springer-Verlag, ed., Berlin, 2004, SCS, pp. 103–129.

[57] C. Vuik, A.Segal, and J. Meijerink, *An efficient preconditioned CG method for the solution of a classof layered problems with extreme contrasts of coefficients*, J. of Comp. Phys., 152 (1999), pp. 385–403.

[58] C. Vuik, A.Segal, J. Meijerink, and G.T.Wijma, *The construction of projection vectors for a ICCG method applied to problems with extreme contrasts in the coefficients*, J. of Comp. Phys., 172 (2001), pp. 426–450.

[59] ———, *The construction of projection vectors for a ICCG method applied to problems with extreme contrasts in the coefficients*, J. of Comp. Phys., 172 (2001), pp. 426–450.

[60] J. Wallis, R. Kendall, and T. Little, *Constrained residual acceleration of conjugate residual methods*, in SPE Reservoir Simulation Symposium, SPE paper no. 13536, Dallas, Texas, 1985.

[61] P. Wang, I. Yotov, M. F. Wheeler, T. Arbogast, C. N. Dawson, M. Parashar, and K. Sepehrnoori, *A new generation EOS compositional reservoir simulator. Part I: Formulation and discretization*, in Fourteenth SPE Symposium on Reservoir Simulation, Dallas, Texas, Society of Petroleum Engineers, June 1997, pp. 55–64.

[62] J. Watts, *A method of improving line succesive overrelaxation in anisotropic problems–a theoretical analysis*, Soc. of Pet. Eng. J., (1973), pp. 105–118.

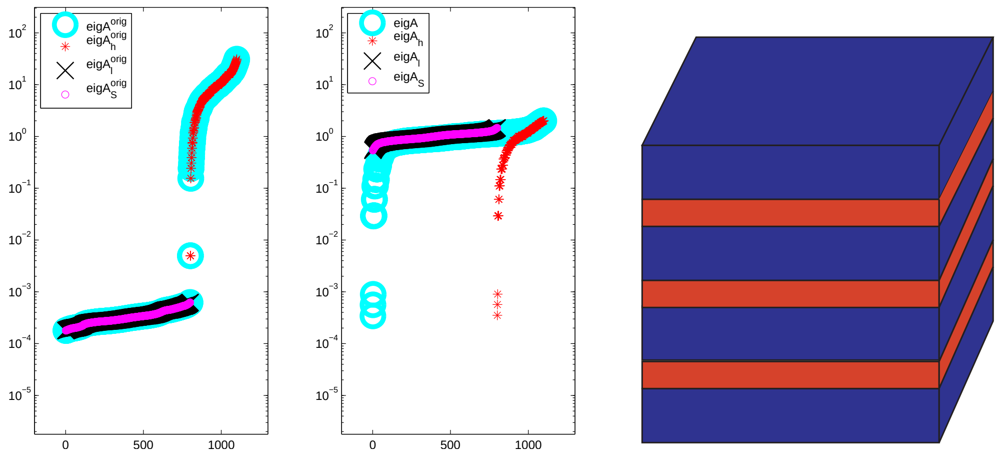

FIGURE 4. Test problem 1: (Left) Spectra. (Right) 3D two-phase flow generated in IPARS: Log permeability field. High and low permeability values are 2.0e+6 and 1.0e-3, respectively.

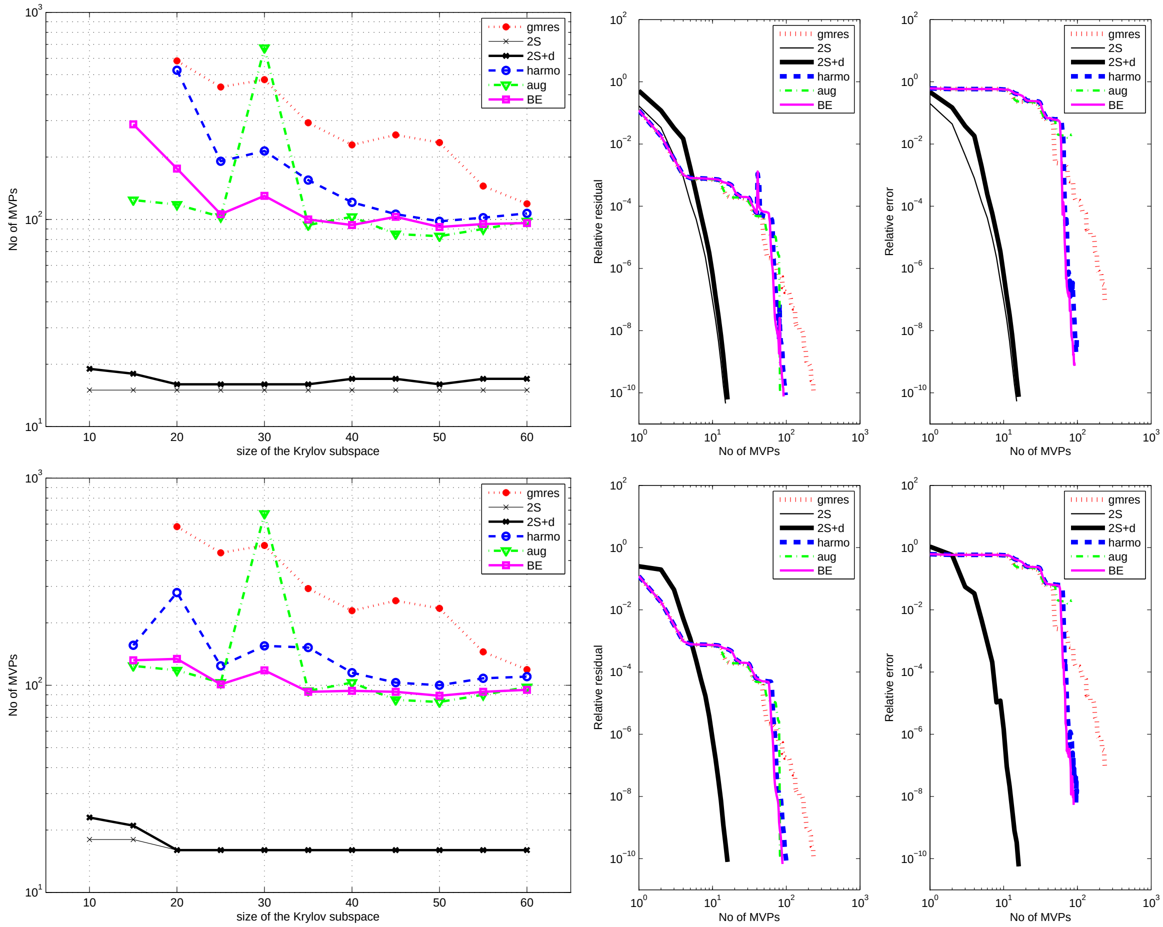

FIGURE 5. Test problem 1: Iteration counts and convergence history for $m, l = 40, 10$ and $m+l = 40+10$. Top (left preconditioned), bottom (right preconditioned).

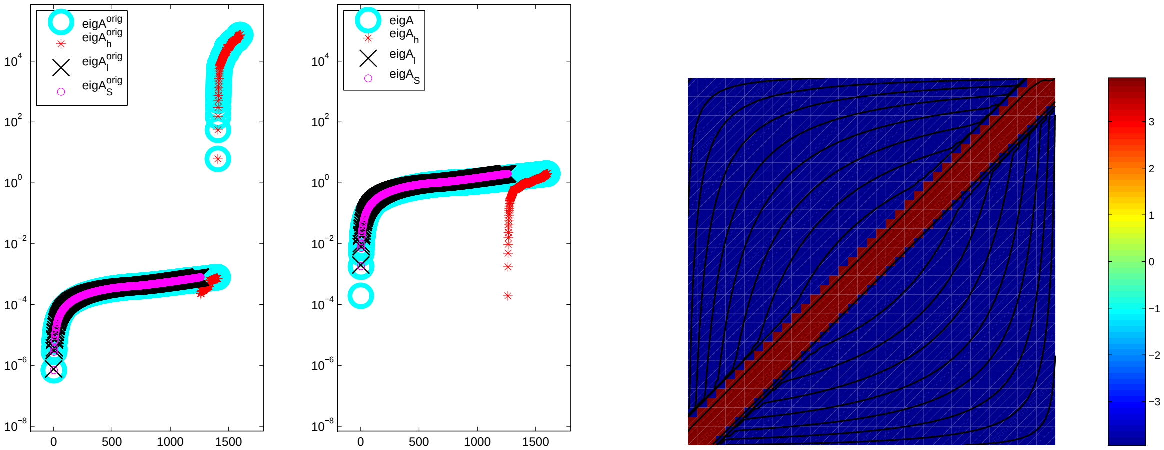

FIGURE 6. Test problem 2: (Left) Spectra of various matrices. (Right) Log permeability field.

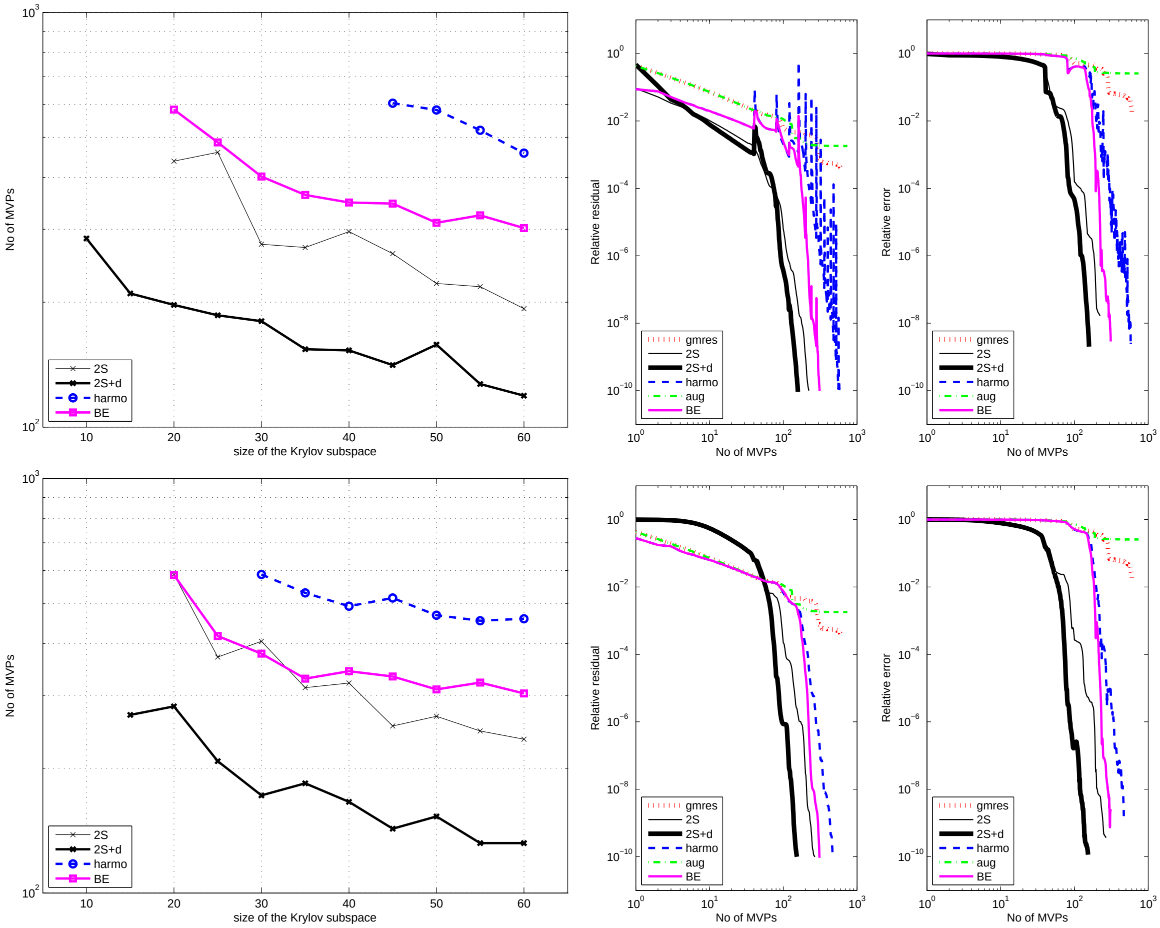

FIGURE 7. Test problem 2: Iteration counts and convergence history for $m, l = 40, 10$ and $m+l = 40+10$. Top (left preconditioned), bottom (right preconditioned).

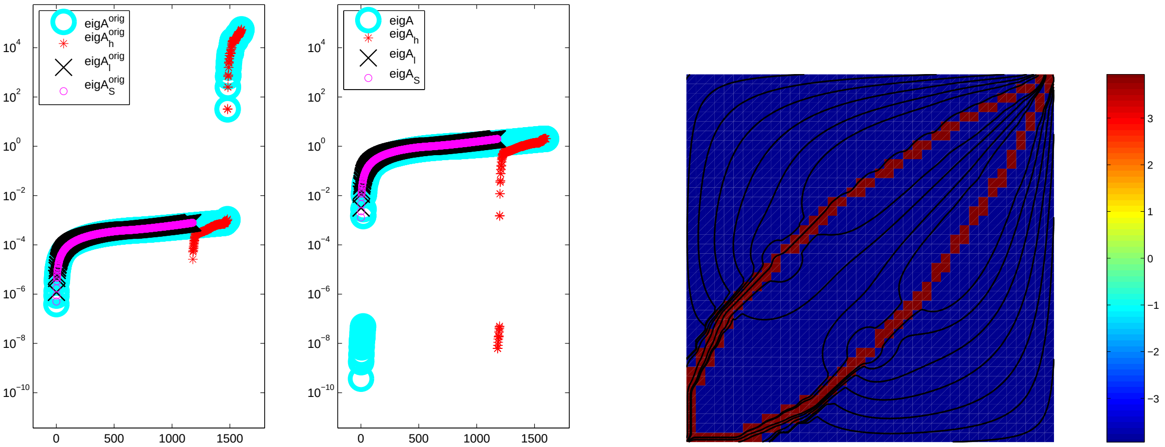

FIGURE 8. Test problem 3: (Left) Spectra of various matrices. (Right) Log permeability field.

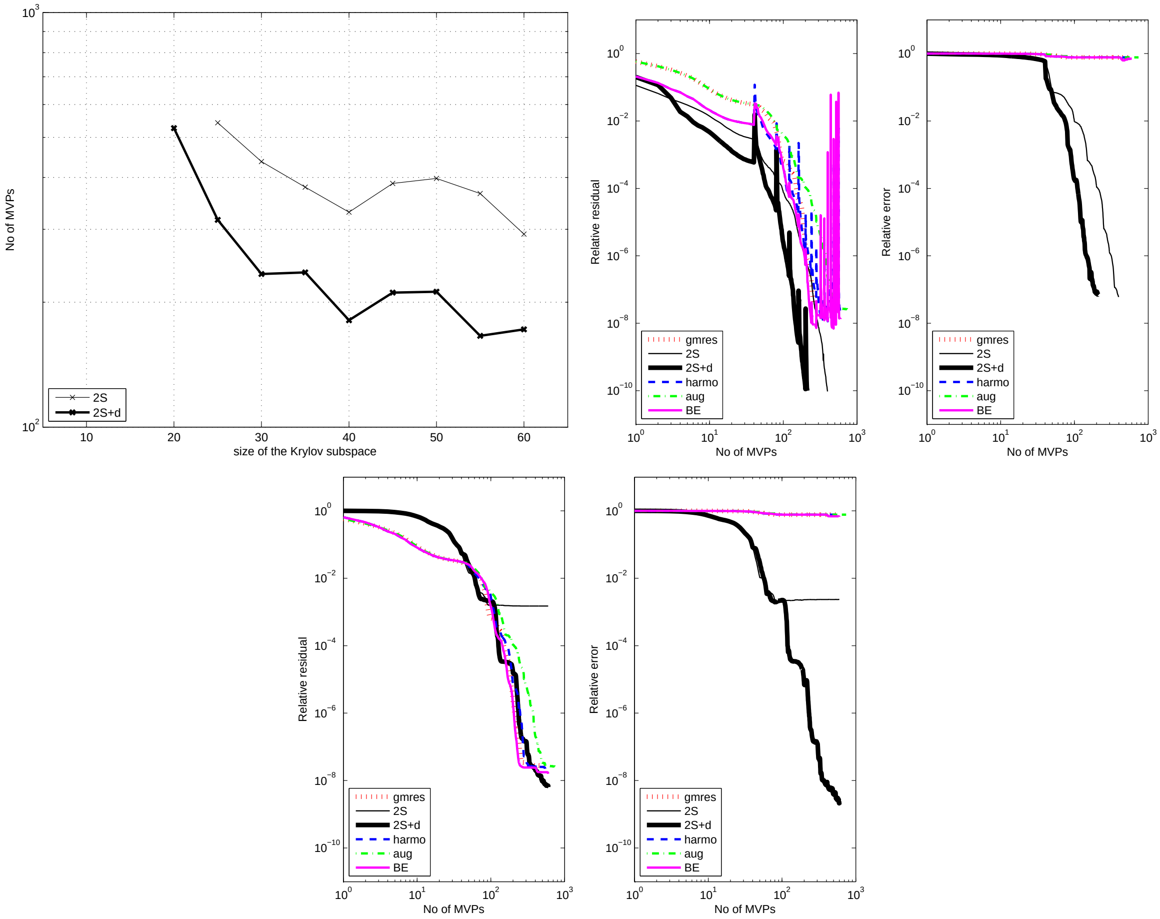

FIGURE 9. Test problem 3: Iteration counts and convergence history for $m, l = 40, 10$ and $m+l = 40{+}10$. Top (left preconditioned); note that only `2Stage(50), 2Stage+d(40,10)` converge. Bottom (right preconditioned); note that `2Stage+d(40,10)` has a dramatically decreasing error.

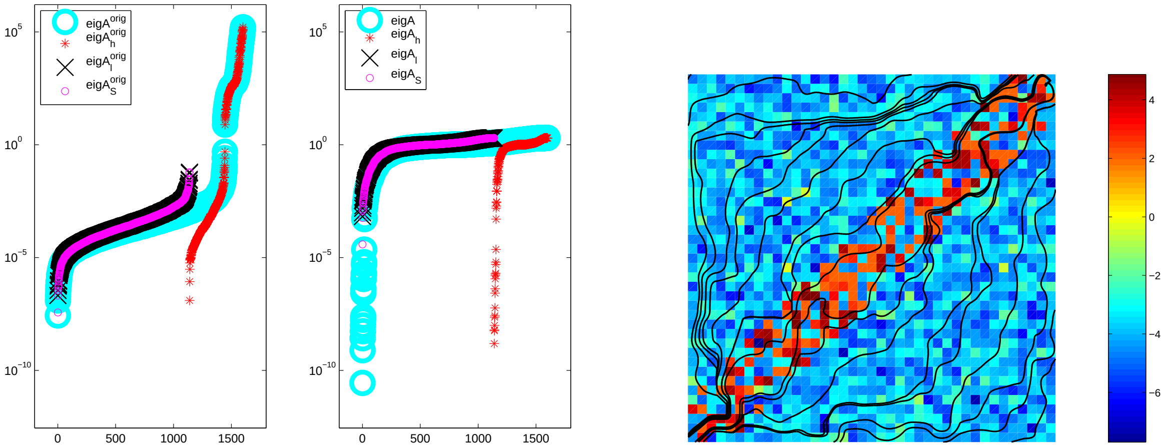

FIGURE 10. Test problem 4: (Left) Spectra of A^{orig}, A_h^{orig} and A_l^{orig}, A_S^{orig}; A, A_h and A_l A_S. (Right) Log permeability field.

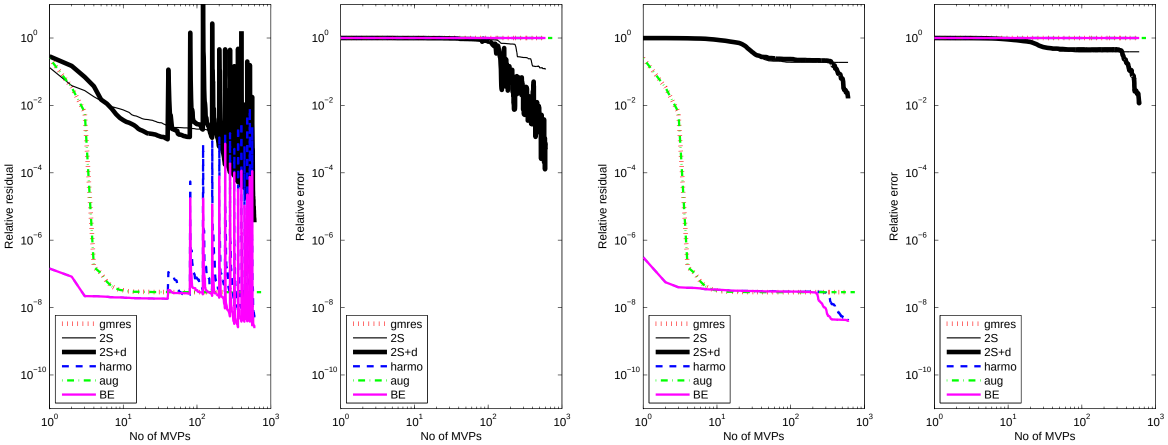

FIGURE 11. Test problem 4: Convergence history for $m, l = 40, 10$ and $m + l = 40 + 10$. Left (left preconditioned); note that none of the methods converge. Right (right preconditioned); note that none of the methods converge.

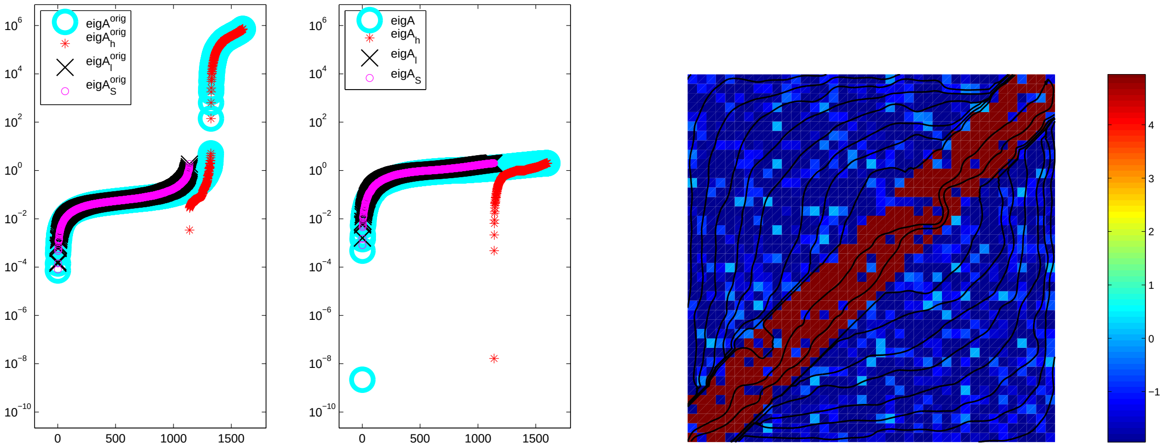

FIGURE 12. Test problem 5: (Left) Spectra of various matrices. (Right) Log permeability field.

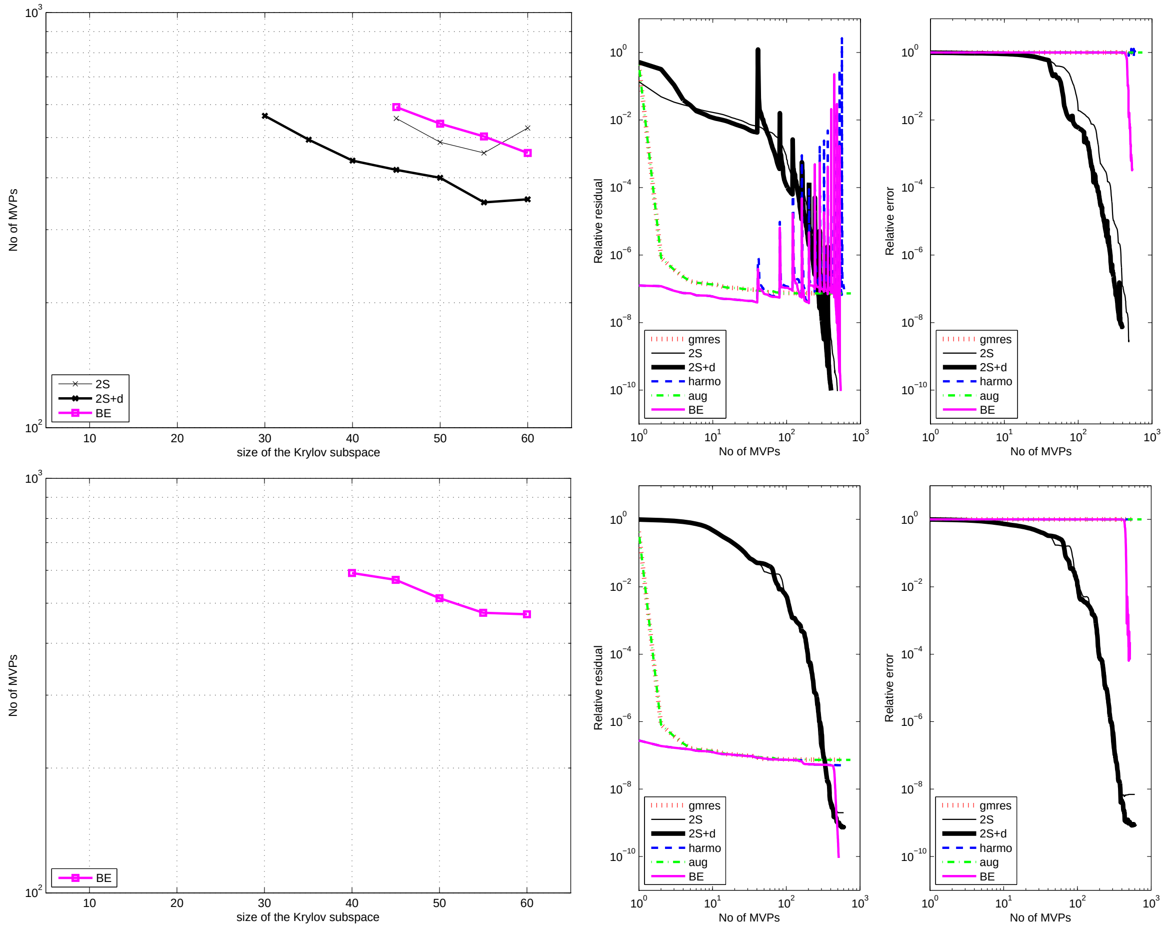

FIGURE 13. Test problem 5: Top (left preconditioned); note that only `2Stage(50)`, `2Stage+d(40,10)`, `BE(40,10)` converge. Bottom (right preconditioned); note that error reduction for `2Stage(50)`, `2Stage+d(40,10)` is much better than `BE(40,10)`.

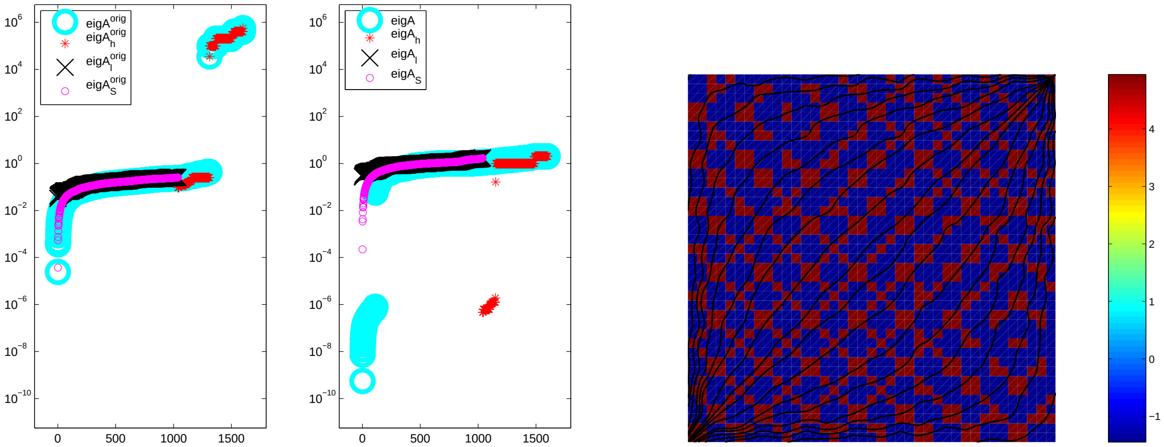

FIGURE 14. Test problem 6: (Left) Spectra of various matrices. (Right) Log permeability field.

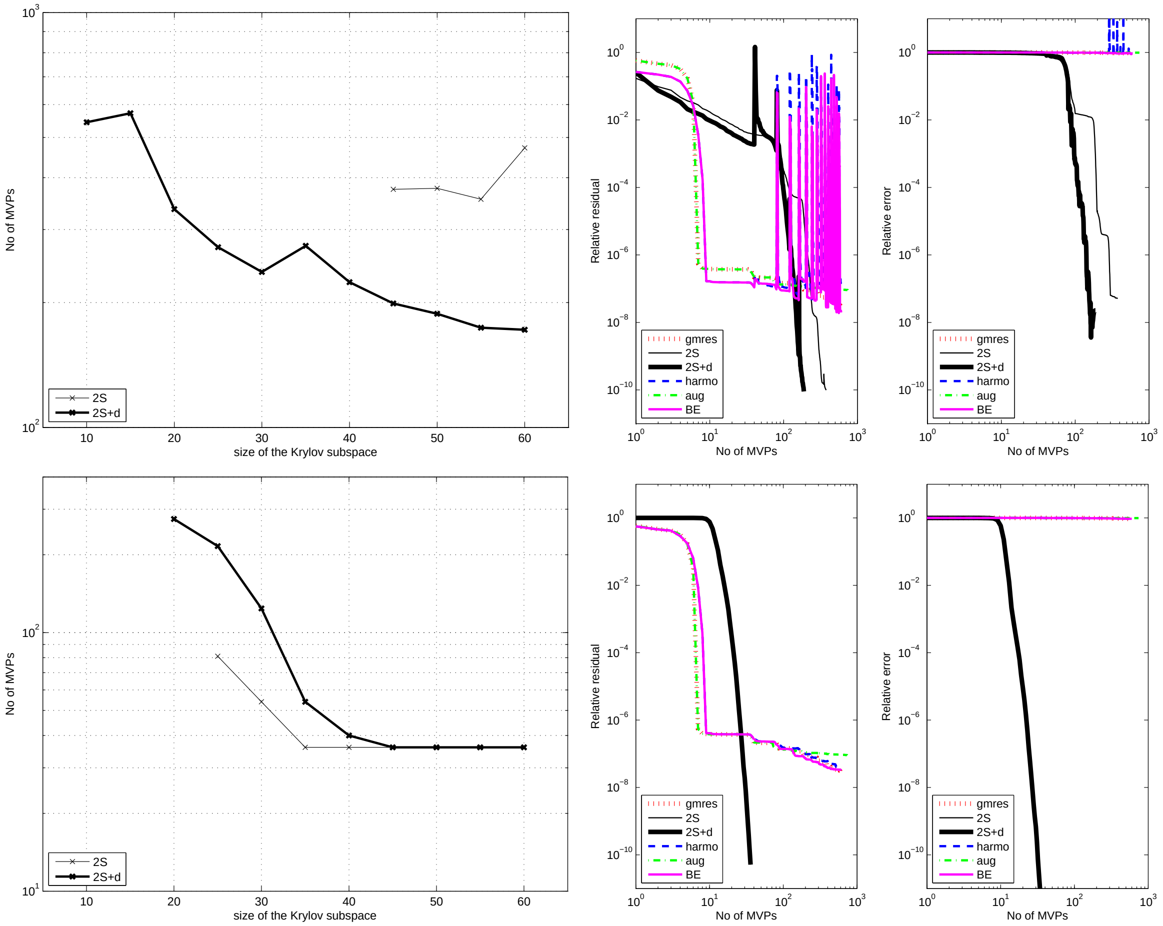

FIGURE 15. Test problem 6: Convergence history for $m, l = 40, 10$ and $m + l = 40 + 10$. Top (left preconditioned); note that only `2Stage(50)`, `2Stage+d(40,10)` converge. Bottom (right preconditioned); note that only `2Stage(50)`, `2Stage+d(40,10)` converge.

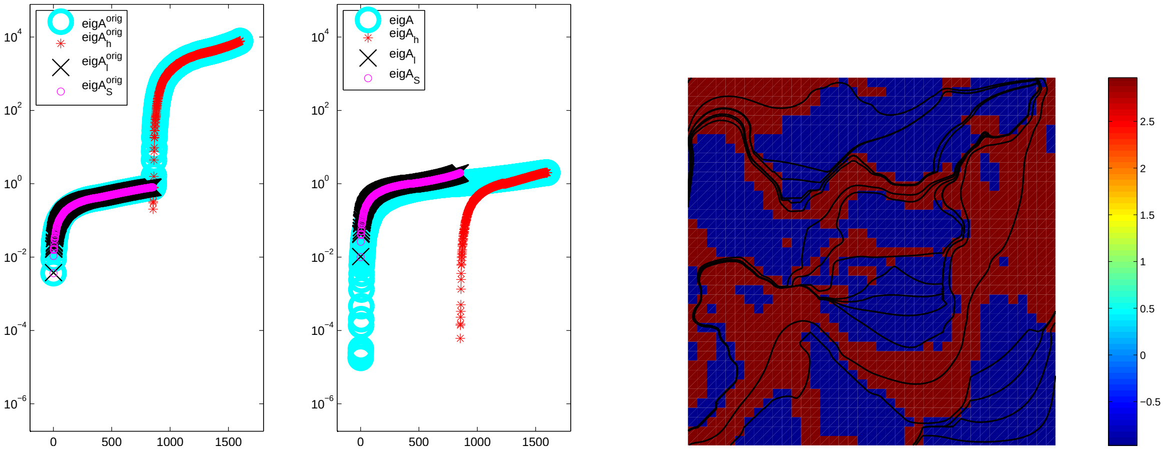

FIGURE 16. Test problem 7: (Left) Spectra of various matrices. (Right) Log permeability field.

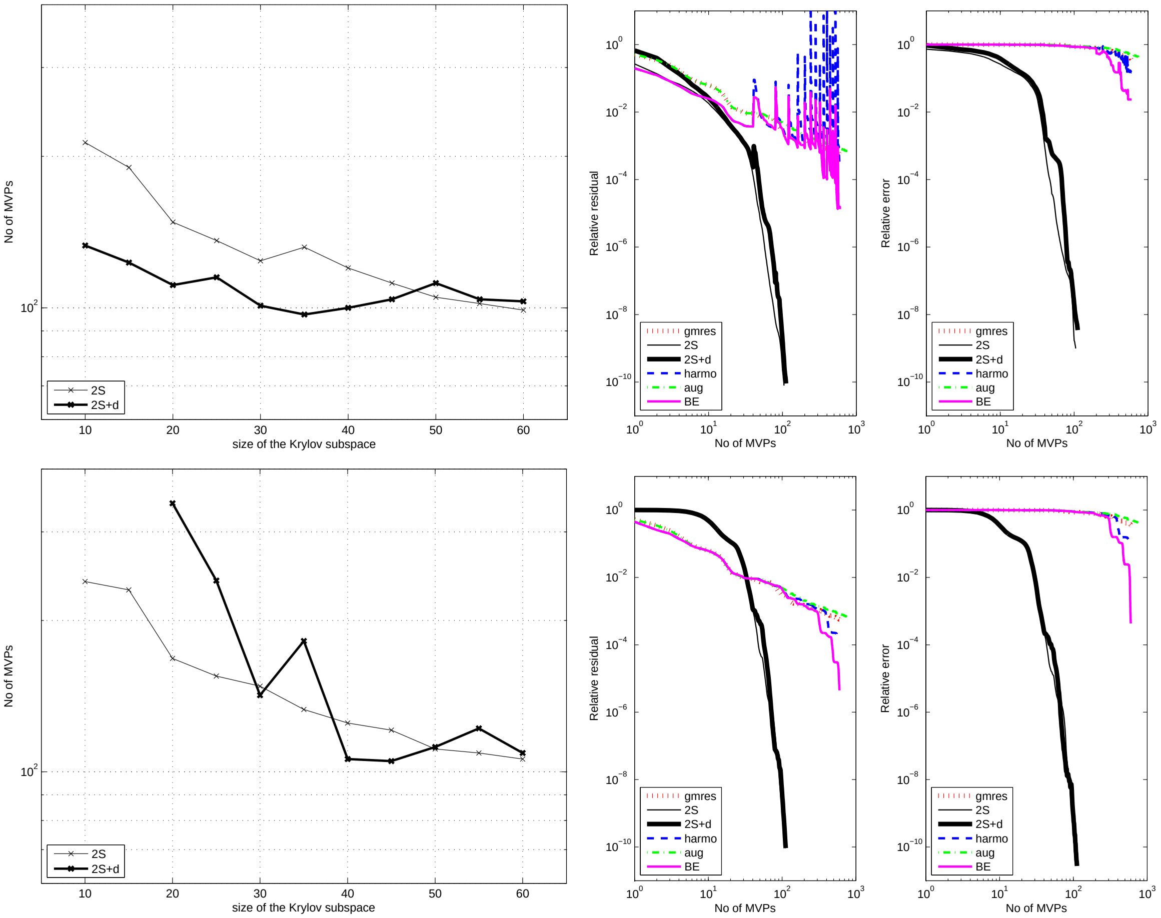

FIGURE 17. Test problem 7: Convergence history for $m, l = 40, 10$ and $m + l = 40 + 10$. Top (left preconditioned); note that only `2Stage(50)`, `2Stage+d(40,10)` converge. Bottom (right preconditioned); note that only `2Stage(50)`, `2Stage+d(40,10)` converge.

Department of Mathematics and Center for Computation and Technology (CCT), Louisiana State University, burak@cct.lsu.edu.

Center for Subsurface Modeling, Institute for Computational Science and Engineering, The University of Texas at Austin, klie@ices.utexas.edu.

Center for Subsurface Modeling, Institute for Computational Science and Engineering, The University of Texas at Austin, mfw@ices.utexas.edu.