SIAM J. SCI. COMPUT.
Vol. 39, No. 5, pp. S222–S247

MULTIPRECONDITIONED GMRES FOR SHIFTED SYSTEMS*

TANIA BAKHOS†, PETER K. KITANIDIS‡, SCOTT LADENHEIM§,
ARVIND K. SAIBABA¶, AND DANIEL B. SZYLD‖

Abstract. An implementation of GMRES with multiple preconditioners is proposed for solving shifted linear systems with shift-and-invert preconditioners. With this type of preconditioner, the Krylov subspace can be built without requiring the matrix-vector product with the shifted matrix. Furthermore, the multipreconditioned search space is shown to grow only linearly with the number of preconditioners. This allows for a more efficient implementation of the algorithm. The proposed implementation is tested on shifted systems that arise in computational hydrology and the evaluation of different matrix functions. The numerical results indicate the effectiveness of the proposed approach.

Key words. shifted systems, Krylov solvers, GMRES

AMS subject classification. 65F10

DOI. 10.1137/16M1068694

1. Introduction.

We consider the solution of shifted linear systems of the form

$$(A + \sigma_j I)x_j = b, \quad j = 1, \ldots, n_\sigma, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is a nonsingular and possibly nonsymmetric matrix, I is the $n \times n$ identity matrix, and n_σ denotes the number of (possibly complex) shifts σ_j. It is assumed that the systems of (1.1) have unique solutions for $j = 1, \ldots, n_\sigma$; i.e., for each σ_j we assume that $(A+\sigma_j I)$ is invertible. These types of linear systems arise in a wide range of applications, for instance, in quantum chromodynamics [14], hydraulic tomography [40], the evaluation of matrix functions based on the Cauchy integral formula [23], and seismic imaging methods such as the full wave inversion [36, 49]. Computing the solution to these large and sparse shifted systems remains a significant computational challenge in these applications.

The straightforward approach is to compute the LU factorization of each shifted system and solve all the systems individually. For large systems, e.g., those arising in the discretization of three-dimensional (3-D) partial differential equations, solving these systems with direct methods, such as with sparse LU or Cholesky factorizations, is impractical, especially considering that a new factorization must be computed for each shift. A more attractive approach is to use factorizations of a limited number of shifted matrices as preconditioners to solve the full sequence of shifted systems.

*Received by the editors April 1, 2016; accepted for publication (in revised form) January 19, 2017; published electronically October 26, 2017.
http://www.siam.org/journals/sisc/39-5/M106869.html
Funding: This research was supported in part by the U.S. National Science Foundation under grant DMS–1418882 and and by the Department of Energy grant DE–SC0016578.
†BCAM - Basque Center for Applied Mathematics, Mazerrado 14, E48009, Bilbao, Basque Country, Spain (tbakhos@bcamath.org).
‡Huang Engineering Center, Stanford University, Stanford, CA 94305 (peterk@stanford.edu).
§School of Computer Science, University of Manchester, Manchester, UK, M13 9PL (scott.ladenheim@manchester.ac.uk).
¶Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (asaibab@ncsu.edu).
‖Department of Mathematics, Temple University, 1805 N Broad Street, Philadelphia, PA 19122 (szyld@temple.edu).

This is the approach we will follow in this paper. We consider the solution of (1.1) using Krylov subspace methods, specifically, a version of the generalized minimal residual method (GMRES) [39] with multipreconditioning. These methods are well suited for the solution of shifted systems because they are shift invariant; see, e.g., [44]. As a result of this property only a single shift-independent Krylov basis needs to be generated from which all shifted solutions can be computed.

Preconditioning is essential to obtain fast convergence in a Krylov subspace method. The preconditioner transforms the original linear system into an equivalent system with favorable properties so that the iterative method converges faster. For shifted systems, preconditioning can be problematic because it may not preserve the shift-invariant property of Krylov subspaces. There are a few exceptions though, namely, polynomial preconditioners [1, 24], shift-and-invert preconditioners [19, 29, 30, 40], and nested Krylov approaches [6]. While we do not propose to perform an exhaustive literature review, we mention that there are several other strategies to handle sequences of shifted systems. A survey is provided in the review paper [44, section 14.1]. Other approaches include the use of limited memory preconditioners [15, 33], updating preconditioners [8, 9, 18], recycling approaches [25, 46], as well as block and deflated approaches for multiple shifts and multiple right-hand sides [12, 45]. Here, we consider using several shift-and-invert preconditioners of the form

$$P_j^{-1} = (A + \tau_j I)^{-1}, \tag{1.2}$$

where the values $\{\tau_j\}_{j=1}^{n_p}$ correspond to the different shifts with $n_p \ll n_\sigma$. For a generic shift τ, we denote the preconditioner P_τ^{-1}. As shown in the next section, one advantage of using the preconditioner in (1.2) is that the appropriate preconditioned Krylov subspace can be built without performing the matrix-vector product with $(A + \sigma_j I)$.

The reason to consider several shifted preconditioners is that a single preconditioner alone is insufficient to effectively precondition across all shifts, as observed in [19, 40]. Incorporating more than one preconditioner necessitates the use of flexible Krylov subspace methods; see, e.g., [38, 48, 50]. Flexible methods allow for the application of a different preconditioner at each iteration and the preconditioners are cycled in some prespecified order. However, this means that information from only one preconditioner is added to the basis at each iteration. In addition, the order in which the preconditioners are applied can have a significant effect on the convergence of the iterative method. To effectively precondition across the range of shifts we use multipreconditioned GMRES (MPGMRES) [16].

Multipreconditioned Krylov subspace methods offer the capability of using several preconditioners at each iteration during the course of solving linear systems. By using information from all the preconditioners in every iteration a larger and richer search space is built. As a result, the convergence of the multipreconditioned method is expected to be faster than the singly preconditioned one. However, the cost per iteration for such multipreconditioned methods can be substantial. To deal with these large computational costs and, in particular, the exponential growth of the search space, a selective version of the MPGMRES algorithm was proposed in [16], where the growth of the search space is linear per iteration, i.e., the same growth as a block method of block size n_p.

Combining the flexible preconditioning framework developed in [40] with MPGMRES, our major contribution is the development of a new iterative solver to handle shifted systems of the form (1.1) that uses multiple preconditioners together with the

fact that no multiplication with $(A + \sigma_j I)$ is needed. Our proposed method builds a search space using information at each iteration from multiple shift-and-invert preconditioners. By searching for optimal solutions over a richer space, we anticipate the convergence to be rapid and the number of iterations to be low. For this class of preconditioners, the resulting Krylov space has a special form, and we show that the search space grows only linearly. This yields a method with contained computational and storage costs per iteration. Numerical experiments illustrate that the proposed solver is more effective than standard Krylov methods for shifted systems, in terms of both iteration counts and overall execution time.

We show that the proposed approach can also be applied to the solution of more general shifted systems of the form $(K + \sigma M)x = b$ with shift-and-invert preconditioners of the form $(K + \tau M)^{-1}$; see Appendix B and section 4.

The paper is structured as follows. In section 2, we briefly review some basic properties of GMRES for the solution of shifted systems with shift-and-invert preconditioners. We also review the MPGMRES algorithm, and a flexible GMRES (FGMRES) algorithm for shifted systems proposed in [40]. In section 3, we present the proposed MPGMRES implementation for shifted systems with shift-and-invert preconditioners, a new theorem characterizing the linear growth of the MPGMRES search space, and discuss an efficient implementation of the algorithm that exploits this linear growth. In section 4, we present numerical experiments for shifted linear systems arising in hydraulic tomography computations. In section 5, we present another set of numerical experiments for the solution of shifted systems arising from the evaluation of different matrix functions. Concluding remarks are given in section 6.

2. GMRES, MPGMRES, and FGMRES-Sh. In order to introduce the proposed multipreconditioned approach for GMRES applied to shifted linear systems, the original GMRES algorithm [39], the multipreconditioned version MPGMRES [16], GMRES for shifted systems, and flexible GMRES for shifted systems (FGMRES-Sh) [19, 40] are first reviewed.

2.1. GMRES. GMRES is a Krylov subspace iterative method for solving large, sparse, nonsymmetric linear systems of the form

$$Ax = b, \quad A \in \mathbb{R}^{n\times n}, \quad b \in \mathbb{R}^n. \tag{2.1}$$

From an initial vector x_0 and corresponding residual $r_0 = b - Ax_0$, GMRES computes at step m an approximate solution x_m to (2.1) belonging to the affine Krylov subspace $x_0 + \mathcal{K}_m(A, r_0)$, where

$$\mathcal{K}_m(A, r_0) \equiv \text{Span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}.$$

The corresponding residual $r_m = b - Ax_m$ is characterized by the minimal residual condition

$$\|r_m\|_2 = \min_{x\in x_0+\mathcal{K}_m(A,r_0)} \|b - Ax\|_2.$$

The approximate solution is the sum of the vector x_0 and a linear combination of the orthonormal basis vectors of the Krylov subspace, which are generated by the Arnoldi algorithm. The first vector in the Arnoldi algorithm is the normalized initial residual $v_1 = r_0/\beta$, where $\beta = \|r_0\|_2$, and subsequent vectors v_k are formed by

orthogonalizing Av_{k-1} against all previous basis vectors. Collecting the basis vectors into the matrix $V_m = [v_1, \ldots, v_m]$, one can write the Arnoldi relation

$$AV_m = V_{m+1}\bar{H}_m, \tag{2.2}$$

where $\bar{H}_m \in \mathbb{R}^{m+1\times m}$ is an upper Hessenberg matrix whose entries are the orthogonalization coefficients.

Thus, solutions of the form $x = x_0 + V_m y_m$ are sought for some $y_m \in \mathbb{R}^m$. Using the Arnoldi relation, and the fact that V_m is orthonormal, the GMRES minimization problem can be cast as the equivalent, smaller least squares minimization problem

$$\|r_m\|_2 = \min_{y\in\mathbb{R}^m} \|\beta e_1 - \bar{H}_m y\|_2 \tag{2.3}$$

with solution y_m, where $e_1 = [1, 0, \ldots, 0]^T$ is the first standard basis vector in $\mathbb{R}^{m+1}$.

As previously mentioned, a key ingredient to hasten the convergence of a given Krylov subspace method is the use of a preconditioner. This is an invertible matrix P that transforms (2.1) into an equivalent problem, for instance, using a left preconditioner, we have $P^{-1}Ax = P^{-1}b$ or, alternatively, with a right preconditioner the linear system becomes

$$AP^{-1}y = b, \text{ with } Px = y.$$

Note that as a result of right (or left) preconditioning the Arnoldi relation (2.2) changes accordingly. In particular, the right preconditioned Arnoldi relation is given by $AZ_m \equiv AP^{-1}V_m = V_{m+1}\bar{H}_m$. It is well known that the GMRES residual minimization problem for right preconditioned linear systems is equivalent to the original nonpreconditioned minimization problem (2.3). For further details, see, e.g., [44].

2.2. MPGMRES. The multipreconditioned GMRES method allows multiple preconditioners to be used in the solution of a given linear system. The method differs from standard preconditioned Krylov subspace methods in that the approximate solution is found in a larger and richer search space [16]. This multipreconditioned search space grows at each iteration by applying each of the preconditioners to all current search directions. In building this search space some properties of both flexible and block Krylov subspace methods are used. As with GMRES, the MPGMRES algorithm then produces an approximate solution satisfying a minimal residual condition over this larger subspace.

Computing the multi-Krylov basis is similar to computing the Arnoldi basis in GMRES but with a few key differences. Starting from an initial vector x_0 and corresponding initial residual r_0, the first MPGMRES iterate x_1 is found in the affine space

$$x_1 \in x_0 + \text{Span}\{P_1^{-1}r_0, \ldots, P_{n_p}^{-1}r_0\},$$

such that the corresponding residual has minimum 2-norm over all vectors of this form. Equivalently, $x_1 = x_0 + Z^{(1)}y_1$, where $Z^{(1)} = [P_1^{-1}V^{(1)}, \ldots, P_{n_p}^{-1}V^{(1)}] \in \mathbb{R}^{n\times n_p}$, $V^{(1)} = r_0/\beta$, and the vector y_1 minimizes the residual. Note that the corresponding residual belongs to the space

$$r_1 \in r_0 + \text{Span}\{AP_1^{-1}r_0, \ldots, AP_{n_p}^{-1}r_0\}.$$

In other words, at the first iteration, the residual can be expressed as a first degree multivariate (noncommuting) matrix polynomial with arguments AP_j^{-1} applied to the initial residual, i.e.,

$$r_1 = r_0 + \sum_{j=1}^{n_p} \alpha_j^{(1)} AP_j^{-1} r_0 \equiv q_1(AP_1^{-1}, \ldots, AP_{n_p}^{-1}) r_0$$

with the property that $q_1(0, \ldots, 0) = 1$.

The next set of orthonormal basis vectors collected in $V^{(2)}$ are generated by orthogonalizing the columns of $AZ^{(1)}$ against $V^{(1)} = r_0/\beta$ and performing a thin QR factorization. The orthogonalization coefficients generated in this process are stored in matrices $H^{(j,1)}$, $j = 1, 2$. The space of search directions is increased by applying each preconditioner to this new matrix, i.e., $Z^{(2)} = [P_1^{-1}V^{(2)}, \ldots, P_{n_p}^{-1}V^{(2)}] \in \mathbb{R}^{n \times n_p^2}$. The general procedure at step k is to block orthogonalize $AZ^{(k)}$ against the matrices $V^{(1)}, \ldots, V^{(k)}$, and then perform a thin QR factorization to generate $V^{(k+1)}$ with orthonormal columns. By construction, the number of columns in $Z^{(k)}$, called search directions, and $V^{(k)}$, called basis vectors, is n_p^k.

Gathering all of the matrices, $V^{(k)}$, $Z^{(k)}$, and $H^{(j,i)}$ generated in the multipreconditioned Arnoldi procedure into larger matrices yields a decomposition of the form

$$A\mathcal{Z}_m = \mathcal{V}_{m+1}\bar{\mathcal{H}}_m, \tag{2.4}$$

where

$$\mathcal{Z}_m = \begin{bmatrix} Z^{(1)} & \cdots & Z^{(m)} \end{bmatrix}, \qquad \mathcal{V}_{m+1} = \begin{bmatrix} V^{(1)} & \cdots & V^{(m+1)} \end{bmatrix},$$

and

$$\bar{\mathcal{H}}_m = \begin{bmatrix} H^{(1,1)} & H^{(1,2)} & \cdots & H^{(1,m)} \\ H^{(2,1)} & H^{(2,2)} & & H^{(2,m)} \\ & \ddots & & \\ & & H^{(m,m-1)} & H^{(m,m)} \\ & & & H^{(m+1,m)} \end{bmatrix}.$$

Observe that the matrix $\bar{\mathcal{H}}_m$ has upper block-Hessenberg form. Introducing the constant

$$\Sigma_m = \sum_{k=0}^{m} n_p^k = \frac{n_p^{m+1} - 1}{n_p - 1},$$

we have that

$$\mathcal{Z}_m \in \mathbb{R}^{n \times (\Sigma_m - 1)}, \quad \mathcal{V}_{m+1} \in \mathbb{R}^{n \times \Sigma_m}, \quad \bar{\mathcal{H}}_m \in \mathbb{R}^{\Sigma_m \times (\Sigma_m - 1)}.$$

The multipreconditioned search space is spanned by the columns of $\mathcal{Z}_m$ so that approximate solutions have the form $x_m = x_0 + \mathcal{Z}_m y$ for $y \in \mathbb{R}^{\Sigma_m - 1}$. Thus, thanks to (2.4), MPGMRES computes y as the solution of the least squares problem

$$\|r_m\|_2 = \min_{y \in \mathbb{R}^{\Sigma_m - 1}} \|\beta e_1 - \bar{\mathcal{H}}_m y\|_2 \ .$$

This is analogous to the GMRES minimization problem, except that here, a larger least squares problem is solved. Note that both $\mathcal{Z}_m$ and $\mathcal{V}_m$ are stored.

The complete MPGMRES method consists of the multipreconditioned Arnoldi procedure coupled with the above least squares minimization problem. Furthermore, we highlight the expression of the MPGMRES residual as a multivariate polynomial, i.e., we can write $r_m = q_m(AP_1^{-1}, \ldots, AP_{n_p}^{-1})r_0$, where $q_m(X_1, \ldots, X_{n_p}) \in \mathbb{P}_m$. Here $\mathbb{P}_m \equiv \mathbb{P}_m[X_1, \ldots, X_{n_p}]$ is the space of noncommuting polynomials of degree m in n_p variables such that $q_m(0, \ldots, 0) = 1$.

In the above description of MPGMRES we have assumed that the matrix $\mathcal{Z}_m$ is of full rank. However, in creating the multipreconditioned search space, columns of $\mathcal{Z}_m$ may become linearly dependent. Strategies for detecting when such linear dependencies arise and then deflating the search space accordingly have been proposed and are discussed in section 3. Furthermore, though not considered in this paper, it is possible to vary the number of preconditioners n_p at each iteration. For such a situation, a subspace of the full MPGMRES space is built, placing this type of method into the category of selective MPGMRES algorithms. Currently, there is no known optimal strategy for varying the number of preconditioners at each iteration. For full implementation details of the complete MPGMRES method, as well as information on proposed selective variants of the algorithm for maintaining linear growth of the search space, we refer the reader to [16, 17].

2.3. GMRES for shifted systems. The solution of shifted systems using the GMRES method requires minor but important modifications. One key idea is to exploit the shift-invariant property of Krylov subspaces, i.e., $\mathcal{K}_m(A+\sigma I, b) = \mathcal{K}_m(A, b)$, and generate a single approximation space from which all shifted solutions are computed. For shifted systems, the Arnoldi relation is

$$(A + \sigma I)V_m = V_{m+1}\left(\bar{H}_m + \sigma \begin{bmatrix} I_m \\ 0 \end{bmatrix}\right) \equiv V_{m+1}\bar{H}_m(\sigma), \tag{2.5}$$

where I_m is the $m \times m$ identity matrix. The matrices V_m, $\bar{H}_m$ are the same as in (2.2) and are independent of the shift σ.

Using (2.5), the equivalent shift-dependent GMRES minimization problem is

$$\|r_m(\sigma)\|_2 = \min_{y \in \mathbb{R}^m} \|\beta e_1 - \bar{H}_m(\sigma)y\|_2 \ .$$

This smaller least squares problem is solved for each shift. Note that the computationally intensive step of generating the basis vectors V_m is performed only once and the cheaper solution of the projected, smaller minimization problem occurs for each of the shifts. We remark that the GMRES initial vector $x_0 = 0$ for all shifted systems since the initial residual must be shift independent.

2.4. FGMRES for shifted systems. Due to a key observation made in [19, 40] that a single shift-and-invert preconditioner is ineffective for preconditioning over a large range of shift values σ, we consider using several shift-and-invert preconditioners in the solution of shifted systems. Using FMGRES, one can cycle through and apply P_j^{-1} for each value of τ_j. Incorporating information from the different shifts into the approximation space improves convergence compared to GMRES with a single shifted preconditioner.

To build this approximation space the authors in [19] use the fact that for a single, generic preconditioner P_τ^{-1},

$$(A + \sigma I)(A + \tau I)^{-1} = I + (\sigma - \tau)(A + \tau I)^{-1}, \tag{2.6}$$

from which it follows that

$$\mathcal{K}_m((A+\sigma I)P_\tau^{-1}, v) = \mathcal{K}_m(P_\tau^{-1}, v). \tag{2.7}$$

Therefore, they propose building a Krylov subspace based on P_τ^{-1}. Note that the Krylov subspace $\mathcal{K}_m(P_\tau^{-1}, v)$ is independent of σ and therefore each shifted system can be projected onto this approximation space. The flexible approach constructs a basis Z_m of the approximation space, where each column $z_j = P_j^{-1}v_j$ corresponds to a different shift. This gives the following flexible, shifted Arnoldi relation

$$(A+\sigma I)Z_m = (A+\sigma I)[P_1^{-1}v_1, \ldots, P_m^{-1}v_m] = V_{m+1}\left(\bar{H}_m(\sigma I_m - T_m) + \begin{bmatrix} I_m \\ 0 \end{bmatrix}\right),$$

where $T_m = \mathrm{diag}(\tau_1, \ldots, \tau_m)$.

Although this approach is more effective than a single preconditioner, information from only a single shift is incorporated into the search space at each iteration and the order in which the preconditioners are applied can affect the performance of FGMRES. It is worth noting the possibility that not every τ_j is different throughout the m FMGRES iterations; however, the aim here is to use n_p distinct preconditioners. These potential deficiencies motivate the proposed multipreconditioned algorithm, which allows for information from all preconditioners (i.e., all shifts $\tau_1, \ldots, \tau_m$) to be built into the approximation space at every iteration.

3. MPGMRES for shifted systems. In this section, we present a modification of the MPGMRES algorithm to handle shifted systems with multiple shift-and-invert preconditioners, where the relation (2.7) plays a crucial role. The new algorithm is referred to as MPGMRES-Sh. We shall prove that the growth of the search space at each iteration is linear in the number of preconditioners thereby leading to an efficient algorithm. This is in contrast to the original MPGMRES algorithm where the dimension of the search space grows exponentially in the number of preconditioners.

The proposed implementation of MPGMRES for solving shifted systems (1.1) with preconditioners (1.2) adapts the flexible strategy of [40], and using relation (2.7) builds a shift-invariant multipreconditioned search space that is used to solve for each shift. We assume throughout that $(A+\tau_j I)$ is invertible for all $j = 1, \ldots, n_p$ so that the preconditioners $P_j^{-1} = (A+\tau_j I)^{-1}$ are well defined. The MPGMRES algorithm proceeds by applying all n_p preconditioners to the columns of $V^{(k)}$, which results in

$$Z^{(k)} = [(A+\tau_1 I)^{-1}V^{(k)}, \ldots, (A+\tau_{n_p} I)^{-1}V^{(k)}] \in \mathbb{R}^{n\times n_p^k}, \quad V^{(k)} \in \mathbb{R}^{n\times n_p^{k-1}}. \tag{3.1}$$

Rearranging this we obtain the relation

$$AZ^{(k)} + Z^{(k)}T^{(k)} = V^{(k)}E^{(k)}, \quad E^{(k)} = e_{n_p}^T \otimes I_{n_p^{k-1}}, \tag{3.2}$$

where $e_{n_p}^T = \underbrace{[1, \ldots, 1]}_{n_p \text{ times}}^T$ and

$$T^{(k)} = \mathrm{diag}\left\{\underbrace{\tau_1, \ldots, \tau_1}_{n_p^{k-1}\text{times}}, \ldots, \underbrace{\tau_{n_p}, \ldots, \tau_{n_p}}_{n_p^{k-1}\text{times}}\right\} \in \mathbb{R}^{n_p^k \times n_p^k}. \tag{3.3}$$

Concatenating the matrices $Z^{(k)}$ and $V^{(k)}$ for $k = 1, \ldots, m$, into $\mathcal{Z}_m$ and $\mathcal{V}_m$, we rewrite the m equations in (3.2) into a matrix relation

$$A\mathcal{Z}_m + \mathcal{Z}_m T_m = \mathcal{V}_m E_m, \tag{3.4}$$

where $T_m = \text{block diag}\{T^{(1)}, \ldots, T^{(m)}\}$ and $E_m = \text{block diag}\left\{E^{(1)}, \ldots, E^{(m)}\right\}$.

To generate the next set of vectors $V^{(k+1)}$ we use a block Arnoldi relationship of the form

$$V^{(k+1)}H^{(k+1,k)} = Z^{(k)} - \sum_{j=1}^{k} V^{(j)}H^{(j,k)}$$

in combination with (3.1), so that at the end of m iterations, the flexible multipreconditioned Arnoldi relationship holds, i.e., $\mathcal{Z}_m = \mathcal{V}_{m+1}\bar{\mathcal{H}}_m$. In summary, we have the system of equations

$$\mathcal{Z}_m = \mathcal{V}_{m+1}\bar{\mathcal{H}}_m, \tag{3.5}$$

$$\mathcal{Z}_m T_m = \mathcal{V}_{m+1}\bar{\mathcal{H}}_m T_m, \tag{3.6}$$

$$A\mathcal{Z}_m + \mathcal{Z}_m T_m = \mathcal{V}_m E_m. \tag{3.7}$$

Remark 3.1. From (2.6), it follows that $\text{Span}\{\mathcal{Z}_m\} = \text{Span}\{(A+\sigma I)\mathcal{Z}_m\}$; see also (2.7). As a consequence, in step 4 of Algorithm 1 we do not need to explicitly compute matrix-vector products with A.

Combining relations (3.5)–(3.7), we obtain the following flexible multipreconditioned Arnoldi relationship for each shift σ:

$$(A+\sigma I)\mathcal{Z}_m = \mathcal{V}_{m+1}\left(\begin{bmatrix} E_m \\ 0 \end{bmatrix} + \bar{\mathcal{H}}_m(\sigma I_{\Sigma_m} - T_m)\right) \equiv \mathcal{V}_{m+1}\bar{\mathcal{H}}_m(\sigma; T_m). \tag{3.8}$$

Finally, searching for approximate solutions of the form $x_m = \mathcal{Z}_m y_m$ (recall that $x_0 = 0$) and using the minimum residual condition we have the following minimization problem for each shift:

$$\begin{aligned} \|r_m(\sigma)\|_2 &= \min_{x \in \text{Span}\{\mathcal{Z}_m\}} \|b - (A+\sigma I)x_m\|_2 = \min_{y \in \mathbb{R}^{\Sigma_{m-1}}} \|b - (A+\sigma I)\mathcal{Z}_m y\|_2 \\ &= \min_{y \in \mathbb{R}^{\Sigma_{m-1}}} \|\mathcal{V}_{m+1}(\beta e_1 - \bar{\mathcal{H}}_m(\sigma; T_m)y)\|_2 = \min_{y \in \mathbb{R}^{\Sigma_{m-1}}} \|\beta e_1 - \bar{\mathcal{H}}_m(\sigma; T_m)y\|_2. \end{aligned} \tag{3.9}$$

The application of the multipreconditioned Arnoldi method using each preconditioner P_j^{-1} in conjunction with the solution of the above minimization problem is the complete MPGMRES-Sh method; see Algorithm 1. It is worth remarking that as a result of (3.5) the storage of $\mathcal{Z}_m$ in line 13 of Algorithm 1 can be avoided. However, this is only possible for linear systems and preconditioners of the form (1.1), (1.2), respectively. Since we aim to solve general systems of the form $K + \sigma M$ (cf. Appendix B) the more general form of the algorithm is presented.

3.1. Growth of the search space. It can be readily seen in Algorithm 1 that the number of columns of $\mathcal{Z}_m$ and $\mathcal{V}_m$ grows exponentially. However, as we shall prove below, with the use of shift-and-invert preconditioners the dimension of this space grows only linearly.

Noting that $r_m(\sigma) \in \text{Span}\{\mathcal{V}_{m+1}\}$, the residuals produced by the MPGMRES-Sh method are of the form

$$r_m(\sigma) \in q_m(P_1^{-1}, \ldots, P_{n_p}^{-1})b \tag{3.10}$$

for a polynomial $q_m \in \mathbb{P}_m$. Recall that $\mathbb{P}_m$ is the space of multivariate polynomials of degree m in n_p (noncommuting) variables such that $q_m(0, \ldots, 0) = 1$. Note that this polynomial is independent of the shifts σ_j.

Algorithm 1. Complete MPGMRES-Sh.

Require: Right-hand side b, preconditioners $\{P_j = A + \tau_j I\}_{j=1}^{n_p}$, shifts $\{\sigma_j\}_{j=1}^{n_\sigma}$, and number of iterations m.

Compute $\beta = \|b\|_2$ and $V^{(1)} = b/\beta$.
for $k = 1, \ldots, m$ **do**
  $Z^{(k)} = [P_1^{-1}V^{(k)}, \ldots, P_{n_p}^{-1}V^{(k)}]$
  $W = Z^{(k)}$
  for j = 1,. . . ,k **do**
    $H^{(j,k)} = (V^{(j)})^T W$
    $W = W - V^{(j)}H^{(j,k)}$
  end for
  $W = V^{(k+1)}H^{(k+1,k)}$ {thin QR factorization}
end for
for $j = 1, \ldots, n_\sigma$ **do**
  Compute $y_m(\sigma_j) = \arg\min_y \|\beta e_1 - \bar{\mathcal{H}}_m(\sigma_j; T_m)y\|$ for each shift.
  $x_m(\sigma_j) = \mathcal{Z}_m y_m(\sigma_j)$, where $\mathcal{Z}_m = [Z^{(1)}, \cdots, Z^{(m)}]$
end for
return The approximate solution $x_m(\sigma_j)$ for $j = 1, \ldots, n_\sigma$.

To illustrate this more clearly, for the case of $n_p = 2$ preconditioners, the first two residuals are of the form

$$\begin{aligned} r_1(\sigma) &\in b + \alpha_1^{(1)}P_1^{-1}b + \alpha_2^{(1)}P_2^{-1}b, \\ r_2(\sigma) &\in b + \alpha_1^{(2)}P_1^{-1}b + \alpha_2^{(2)}P_2^{-1}b + \alpha_3^{(2)}(P_1^{-1})^2 b + \alpha_4^{(2)}P_1^{-1}P_2^{-1}b \\ &\quad + \alpha_5^{(2)}P_2^{-1}P_1^{-1}b + \alpha_6^{(2)}(P_2^{-1})^2 b. \end{aligned}$$

The crucial point we subsequently prove is that the cross-product terms of the form $P_i^{-1}P_j^{-1}v \in \text{Span}\{P_i^{-1}v, P_j^{-1}v\}$ for $i \neq j$. In particular, only the terms that are purely powers of the form $P_i^{-k}v$ need to be computed. This allows $\text{Span}\{\mathcal{Z}_m\}$ to be expressed as the sum of Krylov subspaces generated by individual shift-and-invert preconditioners P_i^{-1}; cf. (2.7). Thus, the search space has only linear growth in the number of preconditioners.

For ease of demonstration, we first prove this for $n_p = 2$ preconditioners. To prove the theorem we need the following two lemmas.

LEMMA 3.2. *Let P_1^{-1}, P_2^{-1} be shift-and-invert preconditioners as defined in* (1.2) *with $\tau_1 \neq \tau_2$. Then $P_1^{-1}P_2^{-1}v \in \text{Span}\{P_1^{-1}v, P_2^{-1}v\}$.*

Proof. We show there exist constants γ_1, γ_2 such that

$$P_1^{-1}P_2^{-1}v = \gamma_1 P_1^{-1}v + \gamma_2 P_2^{-1}v = P_2^{-1}P_1^{-1}v. \tag{3.11}$$

Setting $v = P_2 w$ and left multiplying by P_1 gives the following equivalent formulation

$$\begin{aligned} w &= \gamma_1 P_2 w + \gamma_2 P_1 w = \gamma_1(A + \tau_2 I)w + \gamma_2(A + \tau_1 I)w \\ &= (\gamma_1 + \gamma_2)Aw + (\gamma_1\tau_2 + \gamma_2\tau_1)w. \end{aligned}$$

By equating coefficients we obtain that (3.11) holds for

$$\gamma_1 = \frac{1}{\tau_2 - \tau_1} = -\gamma_2. \qquad \square$$

Remark 3.3. Note that $P_1^{-1}P_2^{-1} = P_2^{-1}P_1^{-1}$, i.e., the preconditioners commute even though we have not assumed that A is diagonalizable. We make use of this observation repeatedly.

LEMMA 3.4. *Let P_1^{-1}, P_2^{-1} be shift-and-invert preconditioners as defined in* (1.2) *with $\tau_1 \neq \tau_2$, then*

$$P_1^{-m}P_2^{-n}v \in \mathcal{K}_m(P_1^{-1}, P_1^{-1}v) + \mathcal{K}_n(P_2^{-1}, P_2^{-1}v).$$

Proof. We proceed by induction. The base case when $m = n = 1$ is true by Lemma 3.2. Assume that the induction hypothesis is true for $1 \leq j \leq m$, $1 \leq k \leq n$, that is, we assume there exist coefficients such that

$$P_1^{-m}P_2^{-n}v = \sum_{j=1}^{m} \alpha_j P_1^{-j}v + \sum_{j=1}^{n} \alpha'_j P_2^{-j}v.$$

Now consider

$$\begin{aligned}
P_1^{-(m+1)}P_2^{-(n+1)}v &= (P_1^{-1}P_2^{-1})(P_1^{-m}P_2^{-n}v) \\
&= \left(\gamma_1 P_1^{-1} + \gamma_2 P_2^{-1}\right)\left(\sum_{j=1}^{m} \alpha_j P_1^{-j}v + \sum_{j=1}^{n} \alpha'_j P_2^{-j}v\right) \\
&= \sum_{j=2}^{m+1} \gamma_1\alpha_{j-1} P_1^{-j}v + \sum_{j=2}^{n+1} \alpha'_{j-1}\gamma_2 P_2^{-j}v \\
&\quad + \sum_{j=1}^{m} \alpha_j\gamma_2 P_2^{-1}P_1^{-j}v + \sum_{j=1}^{n} \gamma_1\alpha'_j P_1^{-1}P_2^{-j}v \\
&= \sum_{j=1}^{m+1} \tilde{\alpha}_j P_1^{-j}v + \sum_{j=1}^{n+1} \tilde{\alpha}'_j P_2^{-j}v.
\end{aligned}$$

The first equality follows from the commutativity of P_1^{-1} and P_2^{-1}. The second equality is the induction hypothesis. The third equality is just a result of the distributive property. The final equality results from applications of Lemma 3.2 and then expanding and gathering like terms. It can be readily verified that every term in this expression for $P_1^{-(m+1)}P_2^{-(n+1)}v$ belongs to $\mathcal{K}_{m+1}(P_1^{-1}, P_1^{-1}v) + \mathcal{K}_{n+1}(P_2^{-1}, P_2^{-1}v)$. ∎

THEOREM 3.5. *Let P_1^{-1}, P_2^{-1} be shift-and-invert preconditioners as defined in* (1.2) *with $\tau_1 \neq \tau_2$. At the mth step of the MPGMRES-Sh algorithm we have*

$$r_m(\sigma) = q_m^{(1)}(P_1^{-1})b + q_m^{(2)}(P_2^{-1})b, \tag{3.12}$$

where $q_m^{(1)}, q_m^{(2)} \in \mathbb{P}_m[X]$ are such that $q_m^{(1)}(0) + q_m^{(2)}(0) = 1$. That is, the residual is expressed as the sum of two (single-variate) polynomials of degree m on the appropriate preconditioner, applied to b.

Proof. Recall that after m iterations of the MPGMRES-Sh algorithm the residual can be expressed as the multivariate polynomial (cf. (3.10))

$$r_m(\sigma) = q_m(P_1^{-1}, P_2^{-1})b, \tag{3.13}$$

with $q_m \in \mathbb{P}_m[X_1, X_2]$ and $q_m(0,0) = 1$. Thus, we need only show that the multivariate polynomial q_m can be expressed as the sum of two polynomials in P_1^{-1} and P_2^{-1}. By Lemma 3.4, any cross-product term involving $P_1^{-j}P_2^{-\ell}$ can be expressed as a linear combination of powers of only P_1^{-1} or P_2^{-1}. Therefore, gathering like terms we can express (3.13) as

$$r_m(\sigma) = \sum_{j=1}^{m} \alpha_j P_1^{-j} b + \sum_{\ell=1}^{m} \alpha'_\ell P_2^{-\ell} b = q_m^{(1)}(P_1^{-1})b + q_m^{(2)}(P_2^{-1})b,$$

where $q_m^{(i)} \in \mathbb{P}_m[X]$ and $q_m^{(1)}(0) + q_m^{(2)}(0) = 1$. ∎

For the general case of $n_p > 2$, we have the same result, as stated below. Its proof is given in Appendix A.

THEOREM 3.6. *Let $\{P_j^{-1}\}_{j=1}^{n_p}$ be shift-and-invert preconditioners as defined in* (1.2)*, where $\tau_j \neq \tau_i$ for $j \neq i$. At the mth step of the MPGMRES-Sh algorithm we have*

$$r_m(\sigma) = \sum_{j=1}^{n_p} q_m^{(j)}(P_j^{-1})b,$$

where $q_m^{(j)} \in \mathbb{P}_m[X]$ and $\sum_{j=1}^{n_p} q_m^{(j)}(0) = 1$. In other words, the residual is a sum of n_p single-variate polynomials of degree m on the appropriate preconditioner, applied to b.

Remark 3.7. The residual can be equivalently expressed in terms of the Krylov subspaces generated by the preconditioners P_j^{-1} and right-hand side b:

$$r_m(\sigma) \in b + \mathcal{K}_m(P_1^{-1}, P_1^{-1}b) + \cdots + \mathcal{K}_m(P_{n_p}^{-1}, P_{n_p}^{-1}b).$$

Thus, at each iteration, the cross-product terms do not add to the search space and the dimension of the search space grows linearly in the number of preconditioners.

3.2. Implementation details. As a result of Theorem 3.6, the MPGMRES-Sh search space can be built more efficiently than the original complete MPGMRES method. Although both methods compute the same search space, the complete MPGMRES method as described in Algorithm 1 builds a search space $\mathcal{Z}_m$ with $\frac{n_p^m - n_p}{n_p - 1}$ columns. However, by Theorem 3.6, $\dim(\mathrm{Span}\{\mathcal{Z}_m\}) = mn_p$, i.e., the search space has only linear growth in the number of iterations and number of preconditioners. Thus, an appropriate implementation of complete MPGMRES-Sh applied to our case would include deflating the linearly dependent search space; for example, using a strong rank-revealing QR factorization as was done in [16]. This process would entail unnecessary computations, namely, in the number of applications of the preconditioners, as well as in the orthonormalization and deflation.

These additional costs can be avoided by directly building the linearly growing search space. Implementing this strategy only requires appending to the matrix of search directions $\mathcal{Z}_{k-1}$, n_p orthogonal columns spanning $\mathcal{R}(Z^{(k)})$, the range of $Z^{(k)}$. Thus, only n_p independent search directions are needed. It follows from Theorem 3.6 that any set of n_p independent vectors in $\mathcal{R}(\mathcal{Z}_k) \setminus \mathcal{R}(\mathcal{Z}_{k-1})$ suffices. We generate these vectors by applying each of the n_p preconditioners to the last column of $V^{(k)}$.

In other words, we replace step 3 in Algorithm 1 with

$$3': \qquad Z^{(k)} = [P_1^{-1}\hat{v}_k, \ldots, P_{n_p}^{-1}\hat{v}_k], \quad \text{where} \quad \hat{v}_k = V^{(k)}e_{n_p}, \tag{3.14}$$

and where $e_{n_p} \in \mathbb{R}^{n_p \times 1}$ is the last column of the $n_p \times n_p$ identity matrix I_{n_p}. Note that this implies that in line 12 the approximate solutions have the form $x_m(\sigma) = \mathcal{Z}_m y_m(\sigma)$ and the corresponding residual minimization problem is

$$\|r_m(\sigma)\|_2 = \min_{y \in \mathbb{R}^{mn_p}} \|\beta e_1 - \bar{\mathcal{H}}_m(\sigma; T_m) y\|_2. \tag{3.15}$$

Remark 3.8. Algorithm 1 with step 3′ as above is essentially what is called *selective MPGMRES* in [16]. The major difference here is that this "selective" version captures the whole original search space (by Theorem 3.6), while in [16] only a subspace of the whole search space is chosen.

This version of the algorithm is precisely the one we use in our numerical experiments. From (3.14) it can be seen that we need n_p solves with a preconditioner per iteration, and that at the kth iteration, one needs to perform $\left(k - \frac{1}{2}\right) n_p^2 + \frac{3}{2} n_p$ inner products for the orthogonalization. This is in contrast to FGMRES-Sh where only one solve with a preconditioner per iteration is needed and only $k + 1$ inner products need to be performed at the kth iteration. Nevertheless, as we shall see in sections 4 and 5, MPGMRES-Sh achieves convergence in fewer iterations and lower computational times. The storage and computational cost of the MPGMRES-Sh algorithm is comparable to that of a block Krylov method. The preconditioner solves can also be parallelized across multiple cores, which further reduces the computational cost.

We now discuss the issue of breakdowns, i.e., the case when the last subdiagonal entry of the matrix $H^{(k,k)}$ is zero, or nearly zero. This may occur when the algorithm has converged to an exact solution, the so-called "lucky breakdown," or the vectors in the search space maybe collinear (for example, when the preconditioner shifts are close to each other) resulting in a breakdown before the algorithm has converged to the exact solution. Following the strategy in [16], we employ a rank-revealing factorization to detect the numerical rank of the matrix, e.g., the LAPACK routine xGEQP3, or strong rank-revealing QR factorization [20]. The details on detecting and handling breakdowns is available in [16]. The complete algorithm is provided in Algorithm 2 in Appendix B.

Finally, the proposed MPGMRES-Sh method, namely, Algorithm 1 with step 3′ as in (3.14), can be easily adapted for solving more general shifted systems of the form

$$(K + \sigma_j M) x(\sigma) = b, \quad j = 1, \ldots, n_\sigma, \tag{3.16}$$

with shift-and-invert preconditioners of the form $P_j = (K + \tau_j M)^{-1}$, $j = 1, \ldots, n_p$. The details can be found in Appendix B.

3.3. Stopping criteria. We discuss here the stopping criteria used to test the convergence of the solver. As was suggested by Paige and Saunders [35], we consider the stopping criterion based on a backward error analysis

$$\|r_m(\sigma)\| \leq \text{btol} \cdot \|b\| + \text{atol} \cdot \|A + \sigma I\| \|x_m(\sigma)\|. \tag{3.17}$$

We found this criterion gave better performance on highly ill-conditioned matrices as compared to the standard stopping criterion obtained by setting atol = 0. We can evaluate $\|x_m(\sigma)\|$ using the relation $\|x_m(\sigma)\| = \|\mathcal{Z}_m y_m(\sigma)\|$. While we do not have access to $\|A + \sigma I\|$, we propose estimating it as follows:

$$\|A+\sigma I\| \approx \max_{k=1,\ldots,mn_p} |\lambda_k|, \qquad \mathcal{H}_m(\sigma;T_m)z_k = \lambda_k \mathcal{H}_m z_k. \tag{3.18}$$

The reasoning behind this approximation is that the solutions of the generalized eigenvalue problem $\mathcal{H}_m(\sigma;T_m)z_k = \lambda_k \mathcal{H}_m z_k$ are harmonic Ritz eigenpairs, i.e., they satisfy

$$(A+\sigma I)u - \theta u \perp \text{Span}\{\mathcal{V}_m\}, \qquad u \in \text{Span}\{\mathcal{V}_{m+1}\bar{\mathcal{H}}_m\},$$

and, therefore, can be considered to be approximate eigenvalues of $A+\sigma I$. The proof is a straightforward extension of [40, Proposition 1]. Numerical experiments confirm that MPGMRES-Sh with the above stopping criterion does indeed converge to an acceptable solution and is especially beneficial for highly ill-conditioned matrices A. The use of backward error for stopping criteria was developed in [34, 37] and also discussed in [3]. A review of this analysis is also provided in [22, Chapter 7] and [47, III section 2.3]. For PDE-based problems, more sophisticated stopping criteria may be developed following the works of [2, 4, 31, 32] but have not been considered here.

Another modification to the stopping criterion needs to be made to account for inexact preconditioner solves. Following the theory developed in [43], a simple modification was proposed in [40]. Computing the true residual would require an extra application of a matrix-vector product with A. Typically, this additional cost is avoided using the Arnoldi relationship and the orthogonality of V_m to compute the residual. However, when inexact preconditioners are used, the computed search space is a perturbation of the desired search space and the residual can only be computed approximately. Assuming that the application of the inexact preconditioners has a relative accuracy ε, based on [40], we use the modified stoppingcriterion

$$\|r_m(\sigma)\| \leq \text{btol} \cdot \|b\| + \text{atol} \cdot \|A+\sigma I\|\|x_m(\sigma)\| + \varepsilon \cdot \|y_m(\sigma)\|_1, \tag{3.19}$$

where $y_m(\sigma)$ is the solution of (3.9), i.e., step 12 of Algorithm 1 with step $3'$ as in (3.14) (or Algorithm 2 with $K = A$ and $M = I$).

4. Application to hydrology.

4.1. Background and motivation. Imaging the subsurface of the earth is an important challenge in many hydrological applications such as groundwater remediation and the location of natural resources. Oscillatory hydraulic tomography is a method of imaging that uses periodic pumping tests to estimate important aquifer parameters, such as specific storage and conductivity. Periodic pumping signals are imposed at pumping wells and the transmitted effects are measured at several observation locations. The collected data are then used to yield a reconstruction of the hydrogeological parameters of interest. The inverse problem can be tackled using the geostatistical approach; for details of this application, see [40]. A major challenge in solving these inverse problems using the geostatistical approach is the cost of constructing the Jacobian, which represents the sensitivity of the measurements to the unknown parameters. In [40], it is shown that constructing the Jacobian requires repeated solution of a sequence of shifted systems. An efficient solver for the forward problem can drastically reduce the overall computational time required to solve the resulting inverse problem. In this section, we demonstrate the performance of MPGMRES-Sh for solving the forward problem with a periodic pumping source.

The equations governing groundwater flow through an aquifer for a given domain Ω with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \Omega_W$ are given by

$$
\begin{aligned}
(4.1) \qquad S_s(x)\frac{\partial \phi(x,t)}{\partial t} - \nabla \cdot (K_\kappa(x)\nabla\phi(x,t)) &= q(x,t), && x \in \Omega,\\
\phi(x,t) &= 0, && x \in \partial\Omega_D,\\
\nabla\phi(x,t)\cdot n &= 0, && x \in \partial\Omega_N,\\
K_\kappa(x)\left(\nabla\phi(x,t)\cdot n\right) &= S_y\frac{\partial\phi(x,t)}{\partial t}, && x \in \partial\Omega_W,
\end{aligned}
$$

where $S_s(x)$ (units of L^{-1}) represents the specific storage, S_y (dimensionless) represents the specific yield and $K_\kappa(x)$ (units of L/T) represents the hydraulic conductivity. The boundaries $\partial\Omega_D, \partial\Omega_N$, and $\partial\Omega_W$ represent Dirichlet, Neumann, and linearized water table boundaries, respectively. In the case of one source oscillating at a fixed frequency ω (units of radians/T), $q(x,t)$ is given by

$$q(x,t) = Q_0\delta(x - x_s)\cos(\omega t).$$

We assume the source to be a point source oscillating at a known frequency ω and peak amplitude Q_0 at the source location x_s. Since the solution is linear in time, we assume the solution (after some initial time has passed) can be represented as

$$\phi(x,t) = \Re(\Phi(x)\exp(i\omega t)),$$

where $\Re(\cdot)$ is the real part and $\Phi(x)$, known as the phasor, is a function of space only that contains information about the phase and amplitude of the signal. Assuming this form of the solution, (4.1) in the so-called phasor domain is

$$
\begin{aligned}
(4.2) \qquad -\nabla\cdot(K_\kappa(x)\nabla\Phi(x)) + i\omega S_s(x)\Phi(x) &= Q_0\delta(x - x_s), && x \in \Omega,\\
\Phi(x) &= 0, && x \in \partial\Omega_D,\\
\nabla\Phi(x)\cdot n &= 0, && x \in \partial\Omega_N,\\
K_\kappa(x)\left(\nabla\Phi(x)\cdot n\right) &= i\omega S_y\Phi(x), && x \in \partial\Omega_W.
\end{aligned}
$$

The differential equation (4.2) and boundary conditions are discretized using standard finite elements implemented through the FEniCS software package [26, 27, 28]. Solving the discretized equations for several frequencies results in solving systems of shifted equations of the form

$$(4.3) \qquad (K + \sigma_j M)\, x_j = b \qquad j = 1, \ldots, n_\sigma,$$

where, K and M are the stiffness and mass matrices, respectively. The relevant MPGMRES-Sh algorithm for this system of equations is the one described in Appendix B, and summarized in Algorithm 2.

4.2. Numerical examples. Here, we consider two synthetic aquifers in our test problems. In both examples, the equations are discretized using standard linear finite elements implemented using FEniCS and Python as the user interface. The first example is a two-dimensional depth-averaged aquifer in a rectangular domain discretized using a regular grid. The second test problem is a challenging 3-D synthetic example chosen to reflect the properties observed at the Boise Hydrogeological Research Site [10]; see Figure 1. Although the domain is regular, modeling the pumping well

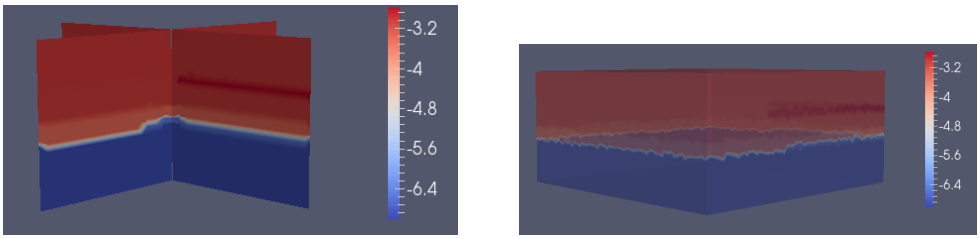

FIG. 1. *Two views of the log conductivity field of the synthetic example. The pumping well is located at the center of the domain.*

TABLE 1
Parameters chosen for Test Problem 1

Definition	Parameters	Values
Aquifer length	L (m)	500
Specific storage	$\log S_s$ (m^{-1})	-11.52
Mean conductivity	$\mu(\log K_\kappa)$ (m/s)	-11.52
Variance of conductivity	$\sigma^2(\log K_\kappa)$	2.79
Frequency range	ω (s^{-1})	$[\frac{2\pi}{600}, \frac{2\pi}{3}]$

accurately requires the use of an unstructured grid. The numerical experiments were all performed on an HP workstation with 16 core Intel Xeon E5-2687W (3.1 GHZ) processor running Ubuntu 14.04. The machine has 128 GB RAM and 1 TB hard disk space.

4.2.1. Test problem 1. We consider a two-dimensional aquifer in a rectangular domain with Dirichlet boundary conditions on all boundaries. The log conductivity is chosen to be a random field generated from the Gaussian process $\mathcal{N}(\mu(x), \kappa(x, y))$, where $\kappa(\cdot, \cdot)$ is an exponential kernel taking the form

$$\kappa(x, y) = 4 \exp\left(-\frac{2\|x - y\|_2}{L}\right) \tag{4.4}$$

with $\mu(x)$ the mean of the log-conductivity chosen to be constant and L the length of the aquifer domain. Other parameters for the model are summarized in Table 1; see Figure 2 for the particular realization of the conductivity field. As mentioned earlier, the bottleneck for the large-scale reconstruction of parameters is the repeated solution of the shifted system of (4.3). To choose realistic parameters following [40], 200 frequencies evenly spaced in the range $\omega \in [\frac{2\pi}{600}, \frac{2\pi}{3}]$ were considered, which results in 200 shifted systems of dimension 22801. Further details of the discretization and the problem setup are provided in [40, section 6]. Note that unlike the Helmholtz equation, the fundamental solutions are not wave-like but exhibit decay as the distance increases, and a finer grid is required for the smaller frequencies. For the frequency range we considered, the grid was chosen to be fine enough so that at the smallest frequency, the discretization errors do not dominate the overall error. The same grid is used for the other frequencies.

We compare the MPGMRES-Sh algorithm proposed here with several other solvers. We consider the "direct" approach, which corresponds to solving the system for each frequency using a direct solver. Since this requires factorizing a matrix for each frequency, this is expensive. "GMRES-Sh" refers to using preconditioned

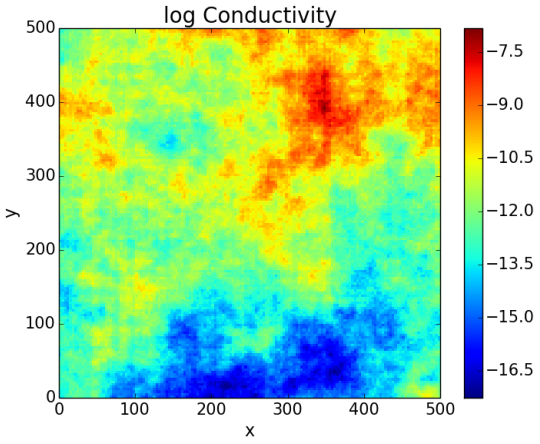

FIG. 2. *A realization of the log conductivity field drawn from* $\mathcal{N}(\mu(x), \kappa(x, y))$ *used in Test Problem* 1. *The parameter* μ *is specified in Table* 1 *and* $\kappa(x, y)$ *is given by* (4.4). *The dimension of the system is* 22801.

GMRES to solve the shifted system of equations using a single shift-and-invert preconditioner, with the frequency chosen to be the midpoint of the range of frequencies. While this is not the optimal choice of preconditioner, it is a representative example for illustrating that one preconditioner does not effectively precondition all systems. We found that the choice of seed did not significantly alter the conclusions of the paper and, in general, finding the seed that gives the best performance is an open problem, and outside the scope of this paper. We also provide a comparison with, "FGMRES-Sh." Following [40], we choose preconditioners of the form $K + \tau_k M$ for $k = 1, ..., m$, where m is the maximum dimension of the Arnoldi iteration. Let $\bar{\tau} = \{\bar{\tau}_1, \ldots, \bar{\tau}_{n_p}\}$ be the list of values that τ_k can take with n_p denoting the number of distinct preconditioners used. In [40], it is shown that systems with frequencies nearer to the origin converge slower, so we choose the values of $\bar{\tau}$ to be evenly spaced in a log scale in the frequency range $\omega \in [\frac{2\pi}{600}, \frac{2\pi}{3}]$. For FGMRES-Sh with n_p preconditioners we assign the first m/n_p values of τ_k as $\bar{\tau}_1$, the next m/n_p values of τ_k to $\bar{\tau}_2$, and so on. If the algorithm has not converged in m iterations, we cycle over the same set of preconditioners. "MPGMRES-Sh" uses the same preconditioners as FGMRES-Sh but builds a different search space. We set the size of the bases per preconditioner to be 5, i.e., $m/n_p = 5$. A relative residual of less than 10^{-10} was chosen as the stopping criterion for all the iterative solvers, i.e., atol was set to 0. Furthermore, all the preconditioner solves were performed using direct solvers and can be treated as exact in the absence of round-off errors.

We report the results of the various aforementioned solvers in Figure 3. In the top two plots, we compare MPGMRES-Sh with FGMRES-Sh and we see that the iteration count to reach the desired tolerance is less with MPGMRES-Sh than with the use of FGMRES-Sh. The vertical lines on the plot denote the preconditioners chosen in the case when $n_p = 5$. In the bottom plot, we compare the solution times (in seconds) of the solvers to solve the 200 shifted systems. We observe that the direct method is computationally prohibitive because the growth is linear with the number of shifted systems. GMRES-Sh uses the shift-invariance property but the convergence with each frequency is different and the number of iterations for some frequencies can be

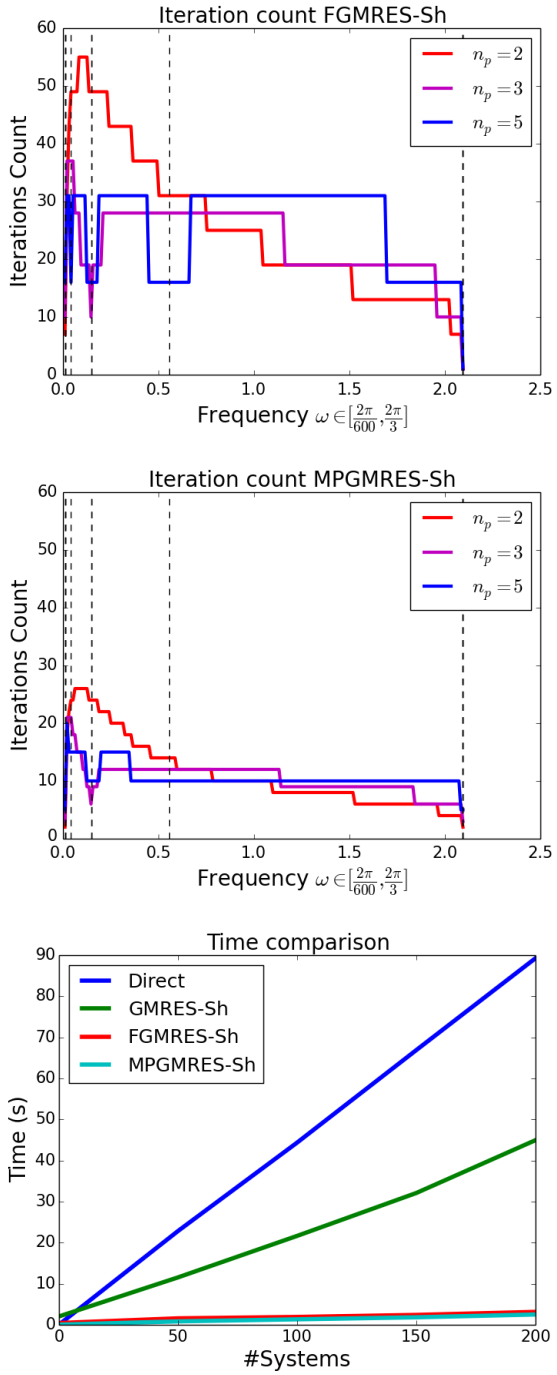

FIG. 3. *Iteration counts comparing FGMRES-Sh (top) to MPGMRES-Sh (center) across the frequency range—the number of preconditioners is varied from* 2 *to* 5. *The system size is* 22801. *(bottom) Time comparison (in seconds) between different solvers—direct, GMRES-Sh, FGMRES-Sh, and MPGMRES-Sh (Algorithm* 2*).*

quite large. Both FMRES-Sh and MPGMRES-Sh appear to behave independently of the number of shifted systems to be solved and MPGMRES-Sh converges marginally faster.

4.2.2. Test problem 2. We now compare the solution times of FMGRES-Sh and MPGMRES-Sh for a realistic 3-D problem on an unconfined aquifer. The aquifer is of size $60 \times 60 \times 27$ m^3 and the pumping well is located in the center with the pumping occurring over a 1 m interval starting 2 m below the water table. For the free surface boundary (top), we use the linearized water table boundary $\partial\Omega_W$; for the other boundaries we use a no-flow boundary for the bottom surface $\partial\Omega_N$ and Dirichlet boundary conditions for the side walls $\partial\Omega_D$. The order of the sparse matrices K and M is 132089 with approximately 1.9 million nonzero entries each (both matrices have the same sparsity pattern). The number of shifts σ_j chosen is 100 with the periods evenly spaced in range from 10 seconds to 15 minutes, resulting in $\omega \in [\frac{2\pi}{900}, \frac{2\pi}{10}]$. Since the direct and GMRES-Sh approaches are infeasible on problems of this size, we do not provide comparisons with them. We focus on studying the iteration counts and time taken by FGMRES-Sh and MPGMRES-Sh. The choice of preconditioners and the construction of bases for FGMRES-Sh and MPGMRES-Sh are the same as that for the previous problem. The number of preconditioners n_p is varied in the range 2–5. However, one crucial difference is that the preconditioner solves are now done using an iterative method, specifically, preconditioned CG with an algebraic multigrid (AMG) solver as a preconditioner available through the PyAMG package [7]. Following [40], the stopping criterion used for the preconditioner solve required the relative residual to be less than 10^{-12}. The number of iterations and the CPU time taken by FGMRES-Sh and MPGMRES-Sh have been displayed in Table 2; as can be seen, MPGMRES-Sh has the edge over FGMRES-Sh both in iteration counts and CPU times. In Figure 4 we report on the number of iterations for the different shifted systems.

In the next numerical experiment, we compare the performance of MPGMRES-Sh with AMG preconditioned BiCGSTAB applied to each system individually, similarly to Figure 3. In total, we solved 100 shifted systems with a stopping criterion based on the relative residual less than 10^{-8}. Each AMG preconditioner was constructed specifically for that specific system, resulting in 100 different preconditioners. The total CPU time for solving these systems was 170.73. Note that this does not include the setup cost for constructing the preconditioners. From Table 2 it is clear that the run time of MPGMRES-Sh is much better compared to solving the systems individually using a different iterative solver.

5. Matrix functions. Matrix function evaluations are relevant in many applications; for example, the evaluation of $\exp(-tA)b$ is important in the time integration of large-scale dynamical systems [5]. In the field of statistics and uncertainty quantification, several computations involving a symmetric positive definite covariance matrix A can be expressed in terms of matrix functions. For example, evaluation

TABLE 2
Comparison of MPGMRES-Sh with FGMRES-Sh on Test Problem 2.

n_p	FGMRES-Sh		MPGMRES-Sh	
	Matvecs	CPU time [s]	Matvecs	CPU time [s]
2	58	160.3	36	87.0
3	52	130.3	24	52.1
5	44	104.2	20	36.7

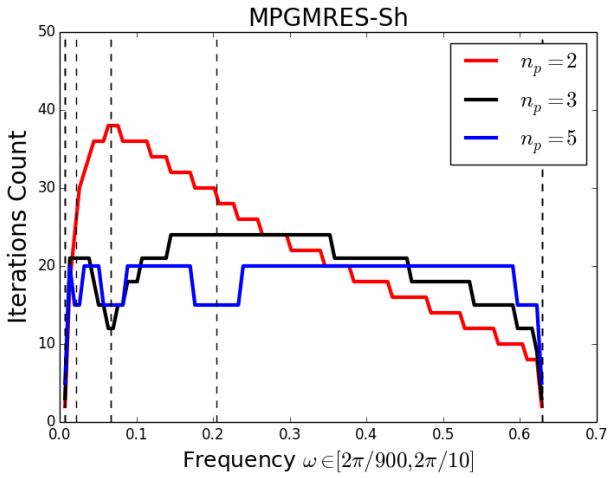

FIG. 4. *Variation of iteration count with increasing number of preconditioners in MPGMRES-Sh (Algorithm 2) for Test Problem 2. The preconditioners are chosen on a log scale and their locations are highlighted in dashed lines.*

of $A^{1/2}\xi$ can be used to sample from $\mathcal{N}(0, A)$, where $\xi \sim \mathcal{N}(0, I)$ as was implemented in [41], and an unbiased estimator to $\log(\det(A))$ can be constructed as

$$\log(\det(A)) = \operatorname{trace}(\log(A)) \approx \frac{1}{n_s} \sum_{k=1}^{n_s} \zeta_k^T \log(A) \zeta_k,$$

where ζ_k is drawn independently and identically distributed from a Rademacher or Gaussian distribution [42]. The evaluation of the matrix function can be carried out by representing the function as a contour integral

$$f(A) = \frac{1}{2\pi i} \int_\Gamma f(z)(zI - A)^{-1} dz,$$

where Γ is a closed contour lying in the region of analyticity of f. The matrix A is assumed to be positive definite and we consider functions f that are analytic except for singularities or a branch cut on $(-\infty, 0]$. We consider the approach in [21] that uses a conformal map combined with the trapezoidal rule to achieve an exponentially convergent quadrature rule as the number of quadrature nodes $n_\sigma \to \infty$. The evaluation of the matrix function $f(A)b$ can then be approximated by the sum

$$f(A)b \quad \approx \quad f_{n_\sigma}(A)b \quad = \quad \sum_{j=1}^{n_\sigma} w_j (z_j I - A)^{-1} b, \tag{5.1}$$

where w_k and z_k are quadrature weights and nodes. The convergence of the approximation as $n_\sigma \to \infty$ is given by an expression of the type

$$\|f(A) - f_{n_\sigma}(A)\| \; = \; \mathcal{O}\left(e^{-C_1 \pi^2 n_\sigma / (\log(M/m) + C_2)}\right),$$

TABLE 3

Comparison of the number of iterations by the FGMRES-Sh and MPGMRES-Sh solvers for the computation of $\exp(-A)b$, $\log(A)b$, *and* $A^{1/2}b$ *evaluated using* (5.1). *CPU times are reported in seconds.*

Matrix name	Condition number	FGMRES-Sh		MPGMRES-Sh	
		Iter. count	CPU Time	Iter. count	CPU Time
$\exp(-A)b$					
plbuckle	1.28×10^6	433	6.11	114	1.66
nasa1824	1.89×10^6	529	9.20	120	2.82
1138_bus	8.57×10^6	505	8.42	126	2.58
$\log(A)b$					
plbuckle	1.28×10^6	217	1.07	78	0.48
nasa1824	1.89×10^6	217	1.65	81	0.85
1138_bus	8.57×10^6	193	0.76	75	0.44
$A^{1/2}b$					
plbuckle	1.28×10^6	64	0.43	60	0.19
nasa1824	1.89×10^6	82	0.67	75	0.36
1138_bus	8.57×10^6	82	0.38	69	0.23

where C_1 and C_2 are two constants depending on the particular method used and m, M are the smallest and largest eigenvalues of A, respectively [21]. Given a tolerance ϵ, we choose n_σ according to the formula

$$n_\sigma = \left\lceil \frac{(\log(M/m) + C_2)}{C_1 \pi^2 \epsilon} \right\rceil.$$

In our application, we choose $\epsilon = 10^{-6}$, so that the expression (5.1) requires the solution of a shifted system of equations $(z_j I - A)x_j = b$ for $j = 1, \ldots, n_\sigma$ and can be computed efficiently using the MPGMRES-Sh algorithm. Note that n_σ depends on the function $f(\cdot)$ and the condition number of the matrix A.

For these experiments we used the stopping criterion in section 3.3 and set atol = btol = 10^{-10}. Three preconditioners were used with shifts at z_1, $z_{n_\sigma/2}$, and z_{n_σ}. For FGMRES-Sh, we use the same strategy to cycle through the preconditioners. Furthermore, m/n_p was set to be 5.

We focus on evaluating the following three important matrix functions $\exp(-A)$, $\log(A)$ and $A^{1/2}$, and in each case we evaluate $f(A)b$ for a randomly generated vector of appropriate dimension. For the evaluation of $\exp(-A)b$, we use method 1, whereas for $\log(A)b$ we use method 2, and finally for $A^{1/2}$ we use method 3 as described in [21]. We take several matrices from the UF sparse matrix collection [13], which have previously been studied in the context of computing matrix functions in [11]. The number of iterations taken by FGMRES-Sh and MPGMRES-Sh is provided in Table 3. As can be seen, MPGMRES-Sh takes fewer preconditioner solves (recall that the matrix-vector products with the shifted matrices are not necessary) to converge to the desired tolerance. The CPU times reported are averaged over 5 independent runs to get accurate timing results. In these examples, as in the hydrology examples, MPGMRES-Sh is more effective both in terms of iteration count and overall run time.

6. Conclusions. In this paper, we derived a new algorithm for efficiently solving shifted systems of equations as described by (1.1) and (3.16). The newly proposed algorithm combines the flexible Krylov framework for shifted systems, developed in [40], with multipreconditioned GMRES, developed in [16], and allows one to build a search space using multiple shift-and-invert preconditioners. We showed that the search

space produced by our algorithm grows linearly with the number of preconditioners and number of iterations. The resulting algorithm converges in fewer iterations than related solvers GMRES-Sh and FGMRES-Sh, and although the cost per iteration of MPGMRES-Sh is higher, in our experience, the overall execution time is lower. The numerical examples drawn from applications in hydrology and matrix function evaluations demonstrate the superior performance in terms of iteration counts and overall run time of MPGMRES-Sh as compared to other standard preconditioned solvers that use a single preconditioner at each iteration.

Appendix A. Case of $n_p > 2$ preconditioners. In this appendix we prove Theorem 3.6. We first prove versions of Lemmas 3.2 and 3.4 for $n_p \geq 2$ preconditioners.

LEMMA A.1. *Let $\{P_j^{-1}\}_{j=1}^{n_p}$ be shift-and-invert preconditioners as defined in* (1.2)*, where $\tau_j \neq \tau_i$ for $j \neq i$. Then for any vector v,*

$$\left(\prod_{j=1}^{n_p} P_j^{-1}\right) v \in \text{Span}\{P_1^{-1}v, \ldots, P_{n_p}^{-1}v\}.$$

Proof. We use induction on n_p. We show that there exist unique constants $\{\gamma_j\}_{j=1}^{n_p}$ so that

$$\left(\prod_{j=1}^{n_p} P_j^{-1}\right) = \sum_{j=1}^{n_p} \gamma_j P_j^{-1} \tag{A.1}$$

holds for any integer n_p. The proof for $n_p = 1$ is straightforward. Assume (A.1) holds for $n_p > 1$. For $n_p + 1$

$$\left(\prod_{j=1}^{n_p+1} P_j^{-1}\right) = \left(\prod_{j=1}^{n_p} P_j^{-1}\right) P_{n_p+1}^{-1} = \left(\sum_{j=1}^{n_p} \gamma_j P_j^{-1}\right) P_{n_p+1}^{-1}.$$

The last equality follows because of the induction hypothesis. Next, assuming that $\tau_{n_p+1} \neq \tau_j$ for $j = 1, \ldots, n_p$, we use Lemma 3.4 to simplify the expression

$$\begin{aligned}\sum_{j=1}^{n_p} \gamma_j P_j^{-1} P_{n_p+1}^{-1} &= \sum_{j=1}^{n_p} \gamma_j \left(\gamma_1^{(j)} P_j^{-1} + \gamma_2^{(j)} P_{n_p+1}^{-1}\right) \\ &= \sum_{j=1}^{n_p} \gamma_j \gamma_1^{(j)} P_j^{-1} + \left(\sum_{j=1}^{n_p} \gamma_j \gamma_2^{(j)}\right) P_{n_p+1}^{-1} \equiv \sum_{j=1}^{n_p+1} \gamma_j' P_j^{-1}. \qquad \square\end{aligned}$$

LEMMA A.2. *Let $\{P_j^{-1}\}_{j=1}^{n_p}$ be shift-and-invert preconditioners as defined in* (1.2)*, where $\tau_j \neq \tau_i$ for $j \neq i$. Then, for all vectors v and all integers m,*

$$\left(\prod_{j=1}^{n_p} P_j^{-m}\right) v \in \sum_{j=1}^{n_p} \mathcal{K}_m(P_j^{-1}, P_j^{-1}v).$$

Proof. We proceed by induction on m. The case when $m = 1$ is true by Lemma A.1. Assume that the Lemma holds for $m > 1$, i.e., there exist coefficients such that

$$\left(\prod_{k=1}^{n_p} P_k^{-m}\right) v = \sum_{j=1}^{m} \alpha_1^{(j)} P_1^{-j} v + \cdots + \sum_{j=1}^{m} \alpha_{n_p}^{(j)} P_{n_p}^{-j} v.$$

Next, consider

$$\begin{aligned}\left(\prod_{k=1}^{n_p} P_k^{-(m+1)}\right) v &= \left(\prod_{k=1}^{n_p} P_k^{-1}\right)\left(\prod_{k=1}^{n_p} P_k^{-m}\right) v \\ &= \left(\prod_{k=1}^{n_p} P_k^{-1}\right)\left(\sum_{j=1}^{m} \alpha_1^{(j)} P_1^{-j} v + \cdots + \sum_{j=1}^{m} \alpha_{n_p}^{(j)} P_{n_p}^{-j} v\right) \\ &= \sum_{j=1}^{m} \alpha_1^{(j)} \left(\prod_{k=1}^{n_p} P_k^{-1}\right) P_1^{-j} v + \cdots + \sum_{j=1}^{m} \alpha_{n_p}^{(j)} \left(\prod_{k=1}^{n_p} P_k^{-1}\right) P_{n_p}^{-j} v.\end{aligned}$$

The first equality is a result of the commutativity of the shift-and-invert preconditioners. The second equality follows from the application of the induction hypothesis. Next, consider the last expression, which we rewrite as

$$\begin{aligned}\sum_{i=1}^{n_p}\sum_{j=1}^{m} \alpha_i^{(j)} P_i^{-j} \left(\prod_{k=1}^{n_p} P_k^{-1}\right) v &= \sum_{i=1}^{n_p}\sum_{j=1}^{m} \alpha_i^{(j)} P_i^{-j} \left(\sum_{k=1}^{n_p} \gamma_k P_k^{-1} v\right) \\ &= \sum_{i=1}^{n_p}\sum_{j=1}^{m}\sum_{k=1}^{n_p} \alpha_i^{(j)} \gamma_k P_i^{-j} P_k^{-1} v,\end{aligned}$$

where the first equality follows from an application of Lemma A.1. By repeated application of Lemma 3.4, the lemma follows. ∎

Proof of Theorem 3.6. Recall that by (3.10) the residual can be expressed as a multivariate polynomial in the preconditioners. Due to Lemma A.2, any cross-product term can be reduced to a linear combination of powers of P_j^{-1} only. Gathering like terms together, the residual can be expressed as a sum of single-variate polynomials in each preconditioner. ∎

Appendix B. Algorithm for more general shifted systems. In this appendix, we show that Algorithm 1 can also be applied to the solution of more general shifted systems of the form $(K + \sigma M)x = b$ with shift-and-invert preconditioners of the form $(K + \tau M)^{-1}$. Using the identity

$$(K + \sigma M)P_\tau^{-1} = (K + \sigma M)(K + \tau M)^{-1} = I + (\sigma - \tau)MP_\tau^{-1}, \tag{B.1}$$

the same multipreconditioned approach described for the case $M = I$ can be applied. The difference for this more general case is that the multipreconditioned search space is now based on MP_τ^{-1}.

In this case, analogously to (3.1), we have

$$Z^{(k)} = [(K + \tau_1 M)^{-1} V^{(k)}, \ldots, (K + \tau_t M)^{-1} V^{(k)}] \in \mathbb{R}^{n \times n_p^k}, \quad V^{(k)} \in \mathbb{R}^{n \times n_p^{k-1}},$$

Algorithm 2. Selective MPGMRES-Sh for (3.16).

Require: Matrix M, right-hand side b, preconditioners $\{P_j = K + \tau_j M\}_{j=1}^{n_p}$, shifts $\{\sigma_j\}_{j=1}^{n_\sigma}$, and number of iterations m.

Compute $\beta = \|b\|_2$ and $V^{(1)} = b/\beta$.
for $k = 1, \ldots, m$ **do**
  $\hat{v}_k = V^{(k)} e_{n_p}$, and $Z^{(k)} = [P_1^{-1}\hat{v}_k, \ldots, P_{n_p}^{-1}\hat{v}_k]$.
  $W = MZ^{(k)}$
  for j = 1,...,k **do**
    $H^{(j,k)} = (V^{(j)})^T W$
    $W = W - V^{(j)} H^{(j,k)}$
  end for
  $WP = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R & \hat{R} \\ 0 & 0 \end{bmatrix}$ {Thin rrQR factorization}
  Set $V^{(k+1)} = Q_1$, $H^{(k+1,k)} = R$. { Note: $R \in \mathbb{C}^{r \times r}$}
  $Z^{(k)} \leftarrow Z^{(k)} P$.
  for j = 1,...,k **do**
    $H^{(j,k)} \leftarrow H^{(j,k)} P(:, 1:r)$ {Using MATLAB notation.}
  end for
end for
for $j = 1, \ldots, n_\sigma$ **do**
  Compute $y_m(\sigma_j) = \arg\min_y \|\beta e_1 - \bar{\mathcal{H}}_m(\sigma_j; T_m) y\|$ for each shift.
  $x_m(\sigma_j) = \mathcal{Z}_m y_m(\sigma_j)$, where $\mathcal{Z}_m = [Z^{(1)}, \ldots, Z^{(m)}]$
end for
return The approximate solution $x_m(\sigma_j)$ for $j = 1, \ldots, n_\sigma$.

which is equivalently expressed as

$$KZ^{(k)} + MZ^{(k)}T^{(k)} = V^{(k)}E^{(k)}. \tag{B.2}$$

Using (B.2) and concatenating $Z^{(k)}$, $V^{(k)}$, $T^{(k)}$, and $E^{(k)}$ into matrices we obtain the matrix relation

$$K\mathcal{Z}_m + M\mathcal{Z}_m T_m = \mathcal{V}_m E_m, \tag{B.3}$$

where T_m and E_m are defined as in (3.4).

Note that by (B.1), the multipreconditioned Arnoldi relationship holds, i.e., $M\mathcal{Z}_m = \mathcal{V}_{m+1}\bar{\mathcal{H}}_m$, which, in conjunction with (B.3), gives the general version of (3.8):

$$(K + \sigma M)\mathcal{Z}_m = \mathcal{V}_{m+1}\left(\begin{bmatrix} E_m \\ 0 \end{bmatrix} + \bar{\mathcal{H}}_m(\sigma I - T_m)\right) \equiv \mathcal{V}_{m+1}\bar{\mathcal{H}}_m(\sigma; T_m).$$

As before, the approximate solutions have the form $x_m(\sigma) = \mathcal{Z}_m y_m(\sigma)$ and the corresponding residual minimization problem is (3.15). It follows from (B.1) that only a basis for the space $\mathrm{Span}\{M\mathcal{Z}_m\}$ needs to be computed. This is due to the shift-invariant property

$$\mathrm{Span}\{M\mathcal{Z}_m\} = \mathrm{Span}\{(K + \sigma M)\mathcal{Z}_m\}.$$

Thus, all that is needed is to use preconditioners of the form $P_j^{-1} = (K + \tau_j M)^{-1}$ in the input to Algorithm 1 and replace step 4 in Algorithm 1 with

$$4': \quad W = MZ^{(k)}. \tag{B.4}$$

We remark that the analysis and results of Theorem 3.6 remain valid for these more general shifted systems. To see this, consider the transformation of (3.16) to $(KM^{-1} + \sigma I)x_M(\sigma) = b$ by the change of variables $x_M(\sigma) = Mx(\sigma)$. Note that this transformation is only valid when M is invertible. As a result, we have a linear system of the form (1.1) and all the previous results are applicable here. The analysis can be extended to the case when M is not invertible; however, this is outside the scope of this paper and is not considered here. For completeness, the resulting algorithm is summarized in Algorithm 2.

For completeness, we provide the selective version of the algorithm for solving (3.16). The special case for (1.1) can be obtained by setting $K = A$ and $M = I$. It is worth mentioning that the rank-revealing QR factorization is usually more expensive compared to the standard QR factorization. In our numerical experiments, we did not observe breakdown and, therefore, we did not include it in the timing of the numerical experiments.

Acknowledgments. The authors are greatly indebted to Michael Saunders for a careful and thorough reading of the paper. We would also like to thank Warren Barrash for his help with the hydrology examples.

REFERENCES

[1] M. I. Ahmad, D. B. Szyld, and M. B. van Gijzen, *Preconditioned multishift BiCG for $\mathcal{H}_2$-optimal model reduction*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 401–424.

[2] M. Arioli, *A stopping criterion for the conjugate gradient algorithm in a finite element method framework*, Numer. Math., 97 (2004), pp. 1–24.

[3] M. Arioli, I. Duff, and D. Ruiz, *Stopping criteria for iterative solvers*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 138–144.

[4] M. Arioli, D. Loghin, and A. J. Wathen, *Stopping criteria for iterations in finite element methods*, Numer. Math., 99 (2005), pp. 381–410.

[5] T. Bakhos, A. K. Saibaba, and P. K. Kitanidis, *A fast algorithm for parabolic PDE-based inverse problems based on Laplace transforms and flexible Krylov solvers*, J. Comput. Phys., 299 (2015), pp. 940–954.

[6] M. Baumann and M. B. van Gijzen, *Nested Krylov methods for shifted linear systems*, SIAM J. Sci. Comput., 37 (2015), pp. S90–S112.

[7] W. N. Bell, L. N. Olson, and J. B. Schroder, *PyAMG: Algebraic Multigrid Solvers in Python v*3.0, Release 3.2, 2015, https://github.com/pyamg/pyamg.

[8] S. Bellavia, V. De Simone, D. Di Serafino, and B. Morini, *Efficient preconditioner updates for shifted linear systems*, SIAM J. Sci. Comput., 33 (2011), pp. 1785–1809.

[9] M. Benzi and D. Bertaccini, *Approximate inverse preconditioning for shifted linear systems*, BIT, 43 (2003), pp. 231–244.

[10] M. Cardiff, W. Barrash, and P. K. Kitanidis, *Hydraulic conductivity imaging from 3-D transient hydraulic tomography at several pumping/observation densities*, Water Res. Res., 49 (2013), pp. 7311–7326.

[11] J. Chen, M. Anitescu, and Y. Saad, *Computing f(A)b via least squares polynomial approximations*, SIAM J. Sci. Comput., 33 (2011), pp. 195–222.

[12] D. Darnell, R. B. Morgan, and W. Wilcox, *Deflated GMRES for systems with multiple shifts and multiple right-hand sides*, Linear Algebra Appl., 429 (2008), pp. 2415–2434.

[13] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), 1.

[14] A. Frommer, B. Nöckel, S. Güsken, T. Lippert, and K. Schilling, *Many masses on one stroke: Economic computation of quark propagators*, Internat. J. Modern Phys. C, 6 (1995), pp. 627–638.

[15] S. Gratton, A. Sartenaer, and J. Tshimanga, *On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides*, SIAM J. Optim., 21(2011), pp. 912–935.

[16] C. Greif, T. Rees, and D. B. Szyld, *GMRES with multiple preconditioners*, SeMA J., 74 (2017), pp. 213–231.

[17] C. Greif, T. Rees, and D. B. Szyld, *Additive Schwarz with variable weights*, in Domain Decomposition Methods in Science and Engineering XXI, J. Erhel, M. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., Lect. Notes Comput. Sci. Eng. 98, Springer, Berlin, 2014, pages 661–668.

[18] A. Grim-McNally, E. de Sturler, and S. Gugercin, *Preconditioning Parametrized Linear Systems*, preprint, arXiv:1601.05883, 2016.

[19] G.-D. Gu, X.-L. Zhou, and L. Lin, *A flexible preconditoned Arnoldi method for shifted linear systems*, J. Comput. Math., 25 (2007), pp. 522–530.

[20] M. Gu and S. C. Eisenstat *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

[21] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing A^α, $\log(A)$, and related matrix functions by contour integrals*, SIAM J. Numer. Anal., 46 (2008), pp. 2505–2523.

[22] N. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[23] N. J. Higham and A. H. Al-Mohy, *Computing matrix functions*, Acta Numer., 19 (2010), pp. 159–208.

[24] B. Jegerlehner, *Krylov Space Solvers for Shifted Linear Systems*, preprint, arXiv:hep-lat/9612014, 1996.

[25] M. E. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.

[26] A. Logg, K.-A. Mardal, and G. N. Wells, *Automated Solution of Differential Equations by the Finite Element Method*, Springer, Heidelberg, 2012.

[27] A. Logg and G. N. Wells, *DOLFIN: Automated finite element computing*, ACM Trans. Math. Software, 37 (2010), 20.

[28] A. Logg, G. N. Wells, and J. Hake, *DOLFIN: A C++/Python finite element library*, in Automated Solution of Differential Equations, A. Logg, K.-A. Mardal, and G. N. Wells, eds., Lect. Notes Comput. Sci. Eng. 84, Springer, Berlin, 2012, pp. 173–225.

[29] K. Meerbergen, *The solution of parametrized symmetric linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1038–1059.

[30] K. Meerbergen and Z. Bai, *The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1642–1662.

[31] G. Meurant, *The computation of bounds for the norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 16 (1997), pp. 77–87.

[32] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.

[33] J. L. Morales and J. Nocedal, *Automatic preconditioning by limited memory quasi–Newton updating*, SIAM J. Optim., 10 (2000), pp. 1079–1096.

[34] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Nume. Math., 6 (1964), pp. 405–409.

[35] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), 43–71.

[36] R.-É. Plessix and C. A. Pérez Solano, *Modified surface boundary conditions for elastic waveform inversion of low-frequency wide-angle active land seismic data*, Geophys. J. Int., 201 (2015), pp. 1324–1334.

[37] J.-L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. ACM, 14 (1967), pp. 543–548.

[38] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[39] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[40] A. K. Saibaba, T. Bakhos, and P. K. Kitanidis, *A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography*, SIAM J. Sci. Comput., 35 (2013), pp. A3001–A3023.

[41] A. K. Saibaba and P. K. Kitanidis, *Efficient methods for large-scale linear inversion using a geostatistical approach*, Water Res. Res., 48 (2012), W05522.

[42] A. K. Saibaba and P. K. Kitanidis, *Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems*, Adv. Water Res., 82 (2015), pp. 124–138.

[43] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.

[44] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[45] K. M. Soodhalter, *Block Krylov subspace recycling for shifted systems with unrelated right-hand sides*, SIAM J. Sci. Comput., 38 (2016), pp. A302–A324.
[46] K. M. Soodhalter, D. B. Szyld, and F. Xue, *Krylov subspace recycling for sequences of shifted linear systems*, Appl. Numer. Math., 81 (2014), 105–118.
[47] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[48] D. B. Szyld and J. A. Vogel, *FQMR: A flexible quasi-minimal residual method with inexact preconditioning*, SIAM J. Sci. Comput., 23 (2001), pp. 363–380.
[49] J. Virieux and S. Operto, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1–WCC26.
[50] J. A. Vogel, *Flexible BiCG and flexible Bi-CGSTAB for nonsymmetric linear systems*, Appl. Math. Comput., 188 (2007), pp. 226–233.