BLOCK KRYLOV SUBSPACE METHODS FOR SHIFTED SYSTEMS 
WITH DIFFERENT RIGHT-HAND SIDES* 
KIRK M. SOODHALTER†

Abstract. Many Krylov subspace methods for shifted (non-Hermitian) linear systems take advantage of the invariance of the Krylov subspace under a scalar shift of the coefficient matrix. However, exploiting this fact introduces restrictions; e.g., initial residuals must be collinear and this collinearity must be maintained at restart. Two practical consequences of this are that we cannot simultaneously solve (in general) shifted systems with unrelated right-hand sides using the shift invariance, and all shifted residuals cannot be simultaneously minimized over a Krylov subspace such that collinearity is maintained. Therefore, we seek an alternative path which exploits the relationship between coefficient matrices without the need for collinearity.

We present two methods which circumvent this problem. Block Krylov subspaces are shift invariant just as their single-vector counterparts. Thus by collecting all initial residuals into one block vector, we can generate the block Krylov subspace with respect to the unshifted coefficient matrix and the block residual. Due to the shift invariance of the block subspace, we can define block FOM- and GMRES-type projection methods to simultaneously solve all shifted systems. These are not block versions of the shifted FOM method of Simoncini [BIT ’03] or the shifted GMRES method of Frommer and Glässner [SISC ’98]. These methods are compatible with restarting and unrelated right-hand sides. Furthermore, we realize the benefits of block sparse matrix operations which arise in the context of high-performance computing applications.

In this paper, we show that the block Krylov subspace built from an appropriate block starting vector is compatible with solving individual shifted systems and use this to derive our block FOM and GMRES methods for shifted systems. Numerical experiments demonstrate the effectiveness of the methods.

Key words. Krylov subspace methods, shifted systems, block Krylov methods, high-performance computing

1. Introduction. For a given coefficient matrix \( A \in \mathbb{C}^{n \times n} \), a problem which often arises in applied mathematics is to solve multiple linear systems in which the coefficient matrix of each system differs from \( A \) by a scalar multiple of the identity, i.e., we must solve

\[
(A + \sigma_i I)x(\sigma_i) = b(\sigma_i) \quad \text{with} \quad i = 1, 2, \ldots, L.
\]  

(1.1)

When the coefficient matrix is large and sparse, matrix-free iterative methods such as Krylov subspace methods are of interest. When we are solving multiple shifted linear systems, such as those in (1.1), Krylov subspace methods are particularly attractive because, under certain assumptions, the Krylov subspace \( K_j(A, u) \), c.f., (2.2), is invariant under scalar shift of the coefficient matrix. Specifically, we have that

\[
K_j(A + \sigma_{i_1}, u) = K_j(A + \sigma_{i_2}, \bar{u})
\]  

(1.2)

as long as \( \bar{u} = \beta u \) where \( \beta \in \mathbb{C} \setminus \{0\} \). This shift invariance has led to numerous methods for solving the systems in (1.1) over a single Krylov subspace; see, e.g., [2, 5, 8, 13, 21, 25, 33, 34]. However, the collinearity requirement means, in general, we cannot use the invariance (1.2) when the right-hand sides \( \{b(\sigma_i)\}_{i=1}^L \) are unrelated; and in any case, for GMRES type methods, we cannot simultaneously minimize all residuals while maintaining collinearity [14].
Therefore, we propose an alternative. In this paper, we show that we can still exploit the shift invariance of the Krylov subspace but avoid the collinearity restriction by exploiting the invariance of a block Krylov subspace. By collecting initial residuals for all systems in \((1.1)\) as columns of \(R^0_\sigma \in \mathbb{C}^{n \times L}\) and building a block Krylov subspace, we can construct approximations for each shifted system over one block Krylov subspace according to a Petrov-Galerkin condition on each residual (e.g., GMRES \[32\] or FOM \[30\]). By building the block Krylov subspace from all the residuals we avoid the above discussed problems arising from a lack of collinearity. Building upon block Krylov subspace technology allows us to use existing, well-tested implementation strategies with minor modifications. Furthermore, building methods from block Krylov subspace techniques allows us to realize the benefits in communication efficiency which have been observed for block Krylov methods with their sparse block operations; see, e.g., \[19, 20, 28\].

For clarity, it should be noted, the methods presented in this paper are NOT extensions of the shifted GMRES method \[14\] or the shifted FOM \[34\] methods to block Krylov subspace to solve problems with multiple right-hand sides.

The rest of this paper is organized as follows. In the next section, we review methods for shifted linear systems based upon the invariance \((1.2)\). We further discuss the restrictions on methods based on \((1.2)\). In Section 3 we review block Krylov subspace methods and show how through the block invariance, we get a shifted Arnoldi relation for each individual shifted system in \((1.1)\). In Section 4 we use the block shift invariance to derive GMRES- and FOM-like methods which allow simultaneous projection of all residuals according to a Petrov-Galerkin condition over the block Krylov subspace. Further algorithmic details will also be discussed. In Section 5 we will discuss stagnation and convergence of these new methods as well as other issues of performance of the algorithm. Numerical results demonstrating proof of concept and effectiveness of these methods will be shown in Section 6.

2. Preliminaries. We begin with a brief review of Krylov subspace methods as well as techniques for solving shifted linear system. Recall that in many Krylov subspace iterative methods for solving the unshifted system

\[
Ax = b
\]  

with \(A \in \mathbb{C}^{n \times n}\), we generate an orthonormal basis for

\[
K_j(A, u) = \text{span}\{u, Au, \ldots, A^{j-1}u\}
\]  

with the Arnoldi process, where \(u\) is some starting vector. Let \(V_j \in \mathbb{C}^{n \times j}\) be the matrix with orthonormal columns generated by the Arnoldi process spanning \(K_j(A, u)\). Then we have the Arnoldi relation

\[
AV_j = V_{j+1}\overline{H}_j
\]  

where \(\overline{H}_j \in \mathbb{C}^{(j+1) \times j}\) is upper Hessenberg; see, e.g., \[31\] Section 6.3 and \[37\]. Let \(x_0\) be an initial approximation to the solution of \((2.1)\) and \(r_0 = b - Ax_0\) be the initial residual. At iteration \(j\), we choose \(x_j = x_0 + t_j\), with \(t_j \in K_j(A, r_0)\). In GMRES \[32\], \(t_j\) satisfies

\[
b - A(x_0 + t_j) \perp AK_j(A, r_0)
\]

which is equivalent to

\[
t_j = \arg\min_{t \in K_j(A, r_0)} \|b - A(x_0 + t)\|
\]
which is itself equivalent to solving the $(j + 1) \times j$ minimization problem

$$y_j = \arg\min_{y \in C^j} \left\| \Pi_j y - r_0 \right\| e_1^{(j+1)} ,$$

(2.4)

where $e_1^{(j)}$ denotes the $J$th Cartesian basis vector in $\mathbb{C}^i$. We then set $x_j = x_0 + V_j y_j$.

Recall that in restarted GMRES, often called GMRES($m$), we run an $m$-step cycle of the GMRES method and compute an approximation $x_m$. We halt the process, discard $V_m$, and restart with the new residual. This process is repeated until we achieve convergence.

A similar derivation leads to the related Full Orthogonalization Method (FOM) [30]. Here we enforce the condition that

$$b - A(x_0 + t_j) \perp K_j(A, r_0)$$

which is equivalent to solving the $j \times j$ linear system

$$H_j y_j = \beta e_1^{(j)}$$

(2.5)

where $H_j \in \mathbb{C}^{j \times j}$ is simply the matrix obtained by deleting the last row of $\Pi_j$. The iterates produced by the GMRES and FOM algorithms are closely related; see, e.g., [31, Section 6.5.7] as well as [7]. As with GMRES, a restarted version of the FOM method has been proposed called FOM($m$).

Many methods for the simultaneous solution of shifted systems take advantage of the shift invariance (1.2); see, e.g., [8, 13, 14, 15, 21, 22, 34]. However, in a non-symmetric method with restarting, collinearity must be maintained at restart. In [40], this was shown to be a troublesome restriction when attempting to extend such techniques to augmented Krylov methods. Furthermore, note that the shift-invariance (1.2) no longer holds if general preconditioning is used. However, specific polynomial preconditioners can be constructed (see, e.g., [2, 5, 20, 42]) for which shift invariance can be maintained. In this paper, though, we treat only the unpreconditioned problem, as in, e.g., [8, 14, 34].

It should also be noted that methods have been proposed which do not rely on the shift invariance property of Krylov subspace methods. Kressner and Tobler treated the more general situation of parameter dependent linear systems where dependence on the parameter of the matrix and right-hand sides are sufficiently smooth [23]. In [38], the relationship between the shifted coefficient matrices is exploited without using the shift invariance by solving one system and projecting the other residuals in a Lanczos-Galerkin procedure.

We end this section by briefly reviewing the restarted GMRES method for shifted systems of Frommer and Glässner [14] and the restarted FOM method for shifted systems of Simoncini [34], both developed to solve (1.1).

Frommer and Glässner [14] proposed a restarted GMRES method to solve (1.1) in the case that the initial residuals are collinear. Within a cycle, the residual for one system from (1.1) is minimized. We call this the base residual. Approximations for all other systems are chosen such that their residuals are collinear with the base residual. This procedure reduces to solving $L - 1$ small $(m + 1) \times (m + 1)$ linear systems at the end of each cycle. Since all residuals are then collinear at the end of the cycle, the shift invariance of the Krylov subspace holds at the beginning of the next cycle. It is not guaranteed for all matrices and all shifts that collinear residuals can be computed; however, conditions are derived for when such residuals can be constructed.
Specifically, for a positive-real matrix $A$ (field of values being contained in the right half-plane), restarted GMRES for shifted linear systems computes solutions at every iteration for all real shifts $\sigma_i > 0$.

Simoncini proposed an algorithm for simultaneously solving the systems in (1.1) based on FOM($m$) [34]. Due to the properties of the residual produced by the FOM algorithm, the method is conceptually simpler to describe. For each cycle, the common shift-invariant Krylov subspace is generated. For each shifted system, the approximation is computed according to the Petrov-Galerkin condition which defines FOM. Residuals produced by the FOM Petrov-Galerkin condition at step $m$ are always collinear with the $(m + 1)$st Arnoldi vector $v_{m+1}$. Therefore, FOM for shifted systems produces collinear residuals by default, and the Krylov subspace remains invariant after restart. Thus, as long as the initial residuals for all shifted systems in (1.1) are collinear, the shifted FOM algorithm is applicable without modification. However, if the right-hand sides are in general unrelated, we cannot use this method to simultaneously solve all linear systems in (1.1).

3. Shift Invariance of Block Krylov Subspace. Krylov subspace methods have been extended to solve linear systems with multiple right-hand sides

$$AXB = B \quad \text{where} \quad B \in \mathbb{R}^{n \times L};$$

(3.1)

see, e.g., [16, 17, 13, 35, 36, 38, 41]. Let $X_0$ be the initial approximation to the block solution and $R_0 = B - AX_0$ be the initial block residual. The block Krylov subspace $\mathbb{K}_j(A, R_0)$ is a generalization of the definition of a Krylov subspace, i.e.,

$$\mathbb{K}_j(A, R_0) = \text{span} \{R_0, AR_0, A^2R_0, \ldots A^{j-1}R_0\}$$

where the span of a sequence of block vectors is simply the span of the columns of all the blocks combined. It is straightforward to show that this definition is equivalent to

$$\mathbb{K}_j(A, R_0) = \mathbb{K}_j(A, r_0^{(1)}) + \mathbb{K}_j(A, r_0^{(2)}) + \cdots + \mathbb{K}_j(A, r_0^{(L)}).$$

(3.2)

Except for in Section 5.4, we assume throughout this paper that $\dim \mathbb{K}_j(A, R_0) = jL$. Following from the description in [31, Section 6.12], we represent $\mathbb{K}_j(A, R_0)$ in terms of the block Arnoldi basis $\{V_1, V_2, \ldots, V_j\}$ where $V_i \in \mathbb{C}^{n \times L}$ has orthonormal columns and each column of $V_i$ is orthogonal to all columns of $V_j$ for all $j \neq i$. We obtain $V_1$ via the reduced QR-factorization $R_0 = V_1S_0$ where $S_0 \in \mathbb{C}^{L \times L}$ is upper triangular. Let

$$W_j = [V_1 \quad V_2 \cdots V_j] \in \mathbb{C}^{n \times mL}.$$  

Let $\Pi_j = (H_{ij}) \in \mathbb{C}^{(j+1)L \times jL}$ be the block upper Hessenberg matrix generated by the block Arnoldi method where $H_{ij} \in \mathbb{C}^{L \times L}$. This yields the block Arnoldi relation

$$AW_j = W_{j+1} \Pi_j.$$  

(3.3)

A straightforward generalization of GMRES for block Krylov subspaces (called block GMRES) has been described and a block FOM method can be similarly derived, see, e.g., [31, Chapter 6].

A great deal of work on the theory and implementation of block Krylov subspace methods has been published; see, e.g., [16, 13, 17, 36, 35, 38, 41]. The shift
invariance properties of Krylov subspaces extend to the block setting. The following straightforward proposition directly follows from their construction.

**Proposition 3.1.** The block Krylov subspace is invariant under scalar shifts of the coefficient matrix, i.e.,

\[ \mathbb{K}_j(A, R_0) = \mathbb{K}_j(A + \sigma I, R_0) \]  \hspace{1cm} (3.4)

with \( \sigma \in \mathbb{C} \setminus \{0\} \) and satisfies the shifted block Arnoldi relation

\[ (A + \sigma I)W_j = W_{j+1}\overline{H}_j(\sigma) \]  \hspace{1cm} (3.5)

where

\[ \overline{H}_j(\sigma) = \overline{H}_j + \sigma \begin{bmatrix} I_{jL \times jL} \\ 0_{L \times jL} \end{bmatrix} . \]

**Proof.** That the block Krylov subspace is invariant under a scalar shift can be seen by observing that the single-vector Krylov subspaces in the sum on the right-hand side of (3.2) are themselves shift invariant. Using the block Arnoldi relation (3.3), it is straightforward to derive a shifted Arnoldi relation in the context of the block Krylov subspace setting. We have the following equalities

\[ (A + \sigma I)W_j = W_{j+1}\overline{H}_j(\sigma) . \]

The block shift invariance has previously been exploited in [8]. We can use this invariance to represent noncollinear residuals in one Krylov subspace.

**Proposition 3.2.** For the linear systems (1.1) with respective initial approximations \( x_{0(\sigma)} \) and residuals \( r_{0(\sigma)} = b(\sigma) - (A + \sigma I)x_{0(\sigma)} \), let \( R_0^\sigma \in \mathbb{C}^{n \times L} \) be the matrix with the initial residuals as columns, where \( \sigma = \{\sigma_1, \ldots, \sigma_L\} \) denotes the collection of shifts. If we build the block Krylov subspace \( \mathbb{K}_j(A, R_0^\sigma) \), generating the associated \( W_{j+1} \) and \( \overline{H}_j \), then we have for each initial residual the two equalities

\[ r_{0(\sigma)}(i) = W_jE_1^{(j)}S_0e_i^{(L)} , \]  \hspace{1cm} (3.6)

and

\[ r_{0(\sigma)}(i) = W_{j+1}E_1^{(j+1)}S_0e_i^{(L)} , \]  \hspace{1cm} (3.7)

where \( E_1^{(j)} \in \mathbb{C}^{L \times L} \) has an \( L \times L \) identity matrix in the first \( L \) rows and zeros below.

**Proof.** This follows directly from building \( \mathbb{K}_j(A, R_0^\sigma) \) since we can write

\[ r_{0(\sigma)}(i) = R_0^\sigma e_i^{(L)} = V_1S_0 e_i^{(L)} = W_jE_1^{(j)}S_0e_i^{(L)} = W_{j+1}E_1^{(j+1)}S_0e_i^{(L)} . \]

These basic identities allow us to derive GMRES- and FOM-type algorithms with respect to \( \mathbb{K}_j(A, R_0^\sigma) \) in the next section, and these methods will be able to simultaneously solve the systems in (1.1) for unrelated right-hand sides/initial residuals.
4. Shifted Block GMRES and FOM. Using the relations (3.5) and, respectively, (3.7) and (3.9) we can derive GMRES- and FOM-type methods over the block Krylov subspace $K_j(A, R_0^\sigma)$ similar to GMRES [32] and FOM [30], respectively.

To derive our GMRES algorithm, we begin by restating the problem according to the minimum residual Petrov-Galerkin condition for the $i$th shifted system at iteration $j$, i.e., compute $t_{j(\sigma_i)} \in K_j(A, R_0^\sigma)$ such that

$$b_{(\sigma_i)} - (A + \sigma_i I)(x_0(\sigma_i) + t_{j(\sigma_i)}) \perp (A + \sigma_i I)K_j(A, R_0^\sigma). \tag{4.1}$$

**Theorem 4.1.** Satisfying the minimum residual condition (4.1) is equivalent to solving the small GMRES-like least squares problem

$$y_{j(\sigma_i)} = \arg\min_{y \in \mathbb{C}^L} \| E_i^{(j+1)} S_0 e_i^{(L)} - \overline{\Pi}_{j(\sigma_i)} y \|. \tag{4.2}$$

**Proof.** Using (3.7) and (3.3), we can derive this assertion,

$$[(A + \sigma_i I)W_j]^* [b_{(\sigma_i)} - (A + \sigma_i I)(x_0(\sigma_i) + t_{j(\sigma_i)})] = 0$$

$$[W_{j+1} \overline{\Pi}_{j(\sigma_i)}]^* [r_{0(\sigma_i)} - (A + \sigma_i I)t_{j(\sigma_i)}] = 0$$

$$\overline{\Pi}_{j(\sigma_i)}^* W_{j+1}^* \left[ W_{j+1} E_i^{(j+1)} S_0 e_i^{(L)} - (A + \sigma_i I)W_j y_{j(\sigma_i)} \right] = 0$$

$$\overline{\Pi}_{j(\sigma_i)}^* E_i^{(j+1)} S_0 e_i^{(L)} - \overline{\Pi}_{j(\sigma_i)} y_{j(\sigma_i)} = 0.$$ 

Finally this leads to the equation

$$\overline{\Pi}_{j(\sigma_i)}^* E_i^{(j+1)} S_0 e_i^{(L)} = \overline{\Pi}_{j(\sigma_i)}^* \overline{\Pi}_{j(\sigma_i)} y_{j(\sigma_i)}. \tag{4.3}$$

As it can be appreciated, (4.3) is the normal equations formulation of the $(j+1)L \times jL$ least squares problem (4.2).

This least squares problem can be solved using already well-described techniques for band upper Hessenberg matrices arising in the block Arnoldi algorithm; see, e.g., [16] [17] [18]. We must solve $L$ such least squares problems. Unlike in block GMRES, we cannot simultaneously factorize then all, but we can nonetheless efficiently solve each problem at low-cost using Householder reflections. As in the block GMRES case, a progressively updated least squares residual norm is available at each iteration, and the actual correction is only constructed at the end of a cycle or upon convergence.

It should be noted that this proposed algorithm differs from that of Frommer and Glässner [14]. Here we minimize each residual, and there is no collinearity requirement whereas in the algorithm from [14], only one residual is minimized and the others are forced to be collinear. Therefore, if we begin with collinear initial residuals, we can choose which algorithm to use. However, to apply our proposed algorithm, we must carefully handle the collinearity, see Section 4.1.1 below. Algorithm [1] describes the method applied to all shifted systems including the technique proposed in Section 4.1.1 to handle the case in which we begin with collinear residuals.

In a similar fashion, we can derive a shifted FOM-type method by imposing the FOM Petrov-Galerkin condition. For the $i$th shifted system, this means we compute $t_{j(\sigma_i)} \in K_j(A, R_0^\sigma)$ such that

$$b_{(\sigma_i)} - (A + \sigma_i I)(x_0(\sigma_i) + t_{j(\sigma_i)}) \perp K_j(A, R_0^\sigma). \tag{4.4}$$
Algorithm 4.1: Shifted Block GMRES - Outline

\[
\text{Input: } A \in \mathbb{C}^{n \times n}, \{\sigma_i\}_{i=1}^L \subset \mathbb{C}, b_{(\sigma_i)}, \ldots, b_{(\sigma_L)} \in \mathbb{C}^n, x_{(\sigma_i)}, \ldots, x_{(\sigma_L)} \in \mathbb{C}^n, \\
\varepsilon \geq 0 \text{ the convergence tolerance, } m > 0 \text{ a cycle length.}
\]

\[
\text{Output: } x_{(\sigma_i)}, \ldots, x_{(\sigma_L)} \in \mathbb{C}^{n \times P} \text{ such that } \|b_{(\sigma_i)} - (A + \sigma_i I)x_{(\sigma_i)}\| \leq \varepsilon \text{ for all } 1 \leq i \leq L
\]

1. for \( i = 1, 2, \ldots, L \) do
2. \( r_{(\sigma_i)} = b_{(\sigma_i)} - (A + \sigma_i I)x_{(\sigma_i)} \)
3. if Initial residuals are collinear with \( r_{(\sigma_i)} = \beta_i r_{(\sigma_i)} \) then
4. Build \( K_m(A, r_{(\sigma_i)}) \) using the Arnoldi method, generating
5. \( V_{m+1} \in \mathbb{C}^{n \times (m+1)} \) and \( \overline{H}_m \in \mathbb{C}^{(m+1) \times m} \).
6. for \( i = 1, 2, \ldots, L \) do
7. \( y \leftarrow \arg \min_{y \in \mathbb{R}^m} \|\beta_i e_i - \overline{H}_m(\sigma_i)y\| \)
8. \( x_{(\sigma_i)} \leftarrow x_{(\sigma_i)} + V_{m+1}y \)
9. \( r_{(\sigma_i)} \leftarrow r_{(\sigma_i)} - V_{m+1}\overline{H}_m(\sigma_i)y \)

else while max \( 1 \leq i \leq L \{\|r_{(\sigma_i)}\|\} > \varepsilon \) do
10. \( R^\sigma \leftarrow [r_{(\sigma_1)} \ r_{(\sigma_2)} \ \ldots \ r_{(\sigma_L)}] \in \mathbb{C}^{n \times L} \)
11. Build \( K_m(A, R^\sigma) \) using the block Arnoldi method, generating
12. \( W_{m+1} \in \mathbb{C}^{n \times (m+1)L} \) and \( \overline{H}_m \in \mathbb{C}^{(m+1)L \times mL} \).
13. for \( i = 1, 2, \ldots, L \) do
14. \( \tilde{y} \leftarrow \arg \min_{y \in \mathbb{R}^{mL}} \|E_{1}^{(m+1)}S_0 e_i - \overline{H}_m(\sigma_i)y\| \)
15. \( x_{(\sigma_i)} \leftarrow x_{(\sigma_i)} + W_{m+1}y \)
16. \( r_{(\sigma_i)} \leftarrow r_{(\sigma_i)} - W_{m+1}\overline{H}_m(\sigma_i)y \)

The derivation then follows a similar course as for GMRES-type method arising from Theorem 4.1.

**Theorem 4.2.** Satisfying the Petrov-Galerkin condition \( 4.3 \) is equivalent to solving the small \( jL \times jL \) FOM-like linear system

\[ H_{j(\sigma_i)}Y_{j(\sigma_i)} = E_{j}^{(j)}S_0 e_i^{(L)}. \] \hspace{1cm} (4.5)

**Proof.** As in Theorem 4.1, we can derive this assertion using \( 3.0 \) and \( 3.9 \),

\[
W_j^* [b_{(\sigma_i)} - (A + \sigma_i I)(x_{0(\sigma_i)} + t_{j(\sigma_i)})] = 0
\]

\[
W_j^* [r_{0(\sigma_i)} - (A + \sigma_i I)W_jY_{j(\sigma_i)}] = 0
\]

\[
W_j^* \left[ W_j E_{j}^{(j)}S_0 e_i^{(L)} - W_{j+1}\overline{H}_j(\sigma_i)Y_{j(\sigma_i)} \right] = 0
\]

\[
E_{j}^{(j)}S_0 e_i^{(L)} - H_{j(\sigma_i)}Y_{j(\sigma_i)} = 0.
\] \hspace{1cm} (4.6)

This yields \( 4.5 \).

We must solve \( L \) linear systems. Each system can be solved progressively, and the residual norms are available at each iteration without explicit computation of the each correction.

In the case that the initial residuals are collinear, the above-derived algorithm is algebraically equivalent to Simoncini’s restarted FOM for shifted systems \( 34 \), though
our algorithm would likely break down in this case due to dependent block Arnoldi vectors being generated. In the case that the right-hand sides are unrelated (or initial residuals not collinear), one can consider our algorithm as a generalization of the shifted FOM algorithm [34]. However, we reiterate that this is not simply an extension of the shifted FOM algorithm to block Krylov subspaces.

In the case of collinear residuals, the single-vector restarted FOM for shifted systems might be preferred over what we have described; however, we can again modify our algorithm to take advantage of block Krylov subspace technology; see Section 4.1. We present the method in greater detail, including the case when the right-hand sides are collinear, as Algorithm 2.

4.1. When residuals are collinear. In the case of collinear residuals, if we want to use either of our proposed block methods, an initial special iteration cycle is necessary to produce non-collinear residuals compatible with our block methods. This can be accomplished in two ways: through a cycle of GMRES for each shifted system over the common single-vector Krylov subspace they share or a cycle of block FOM in which the block is constructed with the collinear residual vector as the first column and random vectors for the other columns. At the end of each cycle, either technique produces residuals which are not collinear, and we can apply either of our proposed methods. We briefly describe both ideas; however we prefer to apply a cycle of GMRES to each shifted system over the use of the block FOM technique with random vectors, as this idea can behave differently for different sets of random vectors, with variable final outcomes; see Experiment 6.8 for details.

4.1.1. One cycle of single-vector GMRES. In the first cycle, we can generate a single-vector Krylov subspace (due to residual collinearity) and minimize all residuals over this subspace. As long as we avoid stagnation for all right-hand sides, the residuals will not be collinear at the end of the cycle. In subsequent cycles, we then generate the block Krylov subspace, as described above.

4.1.2. Block FOM with random block vectors. We can also use a FOM iteration to obtain noncollinear residuals, but the iteration cannot be over the single-vector Krylov subspace. Shifted FOM naturally produces collinear residuals; thus, we must do something else. Suppose that for each shifted system, we have that the initial residual satisfies \( r_{0}(\sigma) = \beta_{i}v_{1} \). Since all initial residuals are collinear, we can build the block Krylov subspace

\[
\mathcal{K}_{m}(A, R_{0}^\sigma) \quad \text{where} \quad R_{0}^\sigma = [v_{1} \ \bar{v}_{2} \ \cdots \ \bar{v}_{L}]
\]

from one normalized residual \( v_{1} \) (since they are all the same except for scaling) and some randomly generated vectors \( \{\bar{v}_{i}\}_{i=2}^{L} \), similar to procedures for increasing the block size described in [4, 28, 38]. This allows us to still apply the FOM Petrov-Galerkin condition with respect to a block Krylov subspace. Since the collinear residual is the first column of the block, at the end of the cycle, for the \( i \)th shifted system, we now solve a problem of the form

\[
y_{m}(\sigma_{i}) = \beta_{i}H_{m}(\sigma_{i})^{-1}E_{4}^{(m)}S_{0}e_{1} \quad \text{where} \quad H_{m}(\sigma_{i}) = H_{m} + \sigma_{i}I
\]

and updating

\[
x_{m}(\sigma_{i}) = x_{0}(\sigma_{i}) + W_{m}y_{m}(\sigma_{i}).
\]

After the first cycle, the residuals are no longer collinear, and we proceed as before.
Algorithm 4.2: Shifted Block FOM - Outline

Input: \( A \in \mathbb{C}^{n \times n}, \{ \sigma_i \}_{i=1}^L \subset \mathbb{C}, b_{(\sigma_i)}, \ldots, b_{(\sigma_i)} \in \mathbb{C}^n, x_{(\sigma_i)}, \ldots, x_{(\sigma_i)} \in \mathbb{C}^n \),
\( \varepsilon > 0 \) the convergence tolerance, \( m > 0 \) a cycle length, \( m_{\text{init}} > 0 \) an initial cycle length

Output: \( x_{(\sigma_i)}, \ldots, x_{(\sigma_i)} \in \mathbb{C}^{n \times p} \) such that \( \|b_{(\sigma_i)} - (A + \sigma_i I)x_{(\sigma_i)}\| \leq \varepsilon \) for all \( 1 \leq i \leq L \)

1. for \( i = 1, 2, \ldots, L \) do
   2. \( r_{(\sigma_i)} = b_{(\sigma_i)} - (A + \sigma_i I)x_{(\sigma_i)} \)
   3. if \( \text{Initial residuals are collinear with } \mathbf{r}_{(\sigma_i)} = \beta_i \mathbf{r}_{(\sigma_i)} \) then
   4. Build \( K_m(A, r_{(\sigma_i)}) \) using the Arnoldi method, generating
   \[ V_{m+1} \in \mathbb{C}^{n \times (m+1)} \text{ and } H_m \in \mathbb{C}^{(m+1) \times m}. \]
   5. for \( i = 1, 2, \ldots, L \) do
   6. \( \bar{y} \leftarrow \arg\min_{y \in \mathbb{R}^m} \| \beta_i e_1 - H_m(\sigma_i) \bar{y} \| \)
   7. \( x_{(\sigma_i)} \leftarrow x_{(\sigma_i)} + V_m \bar{y} \)
   8. \( r_{(\sigma_i)} \leftarrow r_{(\sigma_i)} - V_{m+1} H_m(\sigma_i) \bar{y} \)
   9. else
   10. while \( \max_{1 \leq i \leq L} \{ \| r_{(\sigma_i)} \| \} > \varepsilon \) do
   11. \( R^\sigma \leftarrow [r_{(\sigma_1)} \; r_{(\sigma_2)} \; \cdots \; r_{(\sigma_L)}] \in \mathbb{C}^{n \times L} \)
   12. Build \( K_m(A, R^\sigma) \) using the block Arnoldi method, generating
   \[ W_{m+1} \in \mathbb{C}^{n \times (m+1)L} \text{ and } H_m \in \mathbb{C}^{(m+1)L \times mL}. \]
   13. for \( i = 1, 2, \ldots, L \) do
   14. \( \bar{y} \leftarrow H_m(\sigma_i)^{-1}E^p S_0 e_i \)
   15. \( x_{(\sigma_i)} \leftarrow x_{(\sigma_i)} + W_m \bar{y} \)
   16. \( r_{(\sigma_i)} \leftarrow r_{(\sigma_i)} - W_{m+1} H_m(\sigma_i) \bar{y} \)

5. Performance of the algorithms. In this section, we discuss four performance-related topics: stagnation and the relationship between the shifted block GMRES (Algorithm 1) and block FOM (Algorithm 2) methods, residual norms of Algorithm 1 compared to single-vector GMRES, growth of the block size due to the number of shifts, and the occurrence of linear dependence in the block Arnoldi vectors.

5.1. Stagnation and the relationship of block GMRES and block FOM.

The two algorithms we have proposed are GMRES and FOM type methods which can be defined over the same subspace. The natural question arises: during a cycle, can we relate the approximations produced by Algorithm 1 and Algorithm 2 in the same way that single-vector GMRES and FOM are related; see, e.g., [31, Section 6.5.5]? To answer this question, we must actually explore the relationships of the standard block GMRES and block FOM methods used to solve \( (A + \sigma_i I) \bar{x} = R^\sigma_0 \) (5.1) with initial approximation \( \bar{x}_0 = 0 \) and taking \( x_j(:, i) = \bar{x}_j(:, i) \). Thus we see that
understanding the behavior block GMRES and its relationship to block FOM is essential for understanding the relationship between Algorithms 1 and Algorithm 2.

In a companion work to this paper [39], the occurrence of stagnation during an iteration of block GMRES (for some or all columns of the approximation) and its relation to block FOM is characterized. We list here some results from [39], relevant to our discussion of the behavior of Algorithms 1 and 2. We state the results with respect to the dummy problem (5.1). Also, we define $\tilde{X}_j^{(F)} = X_j + W_j \hat{Y}_j$ to be the $j$th generalized block FOM approximation, where $\hat{Y}_j = H_j^1 E_1^{(j)} S_0$ and $H_j^1$ is the Moore-Penrose pseudoinverse. If $H_j$ is nonsingular, this yields the block FOM approximation. However, it still produces an approximation in the case that $H_j$ is singular.

**Theorem 5.1.** Let $K_{j+1}(A + \sigma_j I, R_0^2)$ be generated after $j$ iterations of the block Arnoldi algorithm with $W_{j+1}$ and $\Pi_j \in \mathbb{C}^{(j+1)L \times L}$ being defined as earlier for a block Krylov subspace. The matrix $H_j$ is singular with rank $H_j = (j-1)L$ if and only if block GMRES totally stagnates at iteration $j$, i.e., $X_j^{(G)} = X_{j-1}^{(G)}$. Furthermore, if rank $H_j = (j-1)L$ then $\hat{X}_j^{(F)} = X_j^{(G)}$.

In the case that $H_j$ is singular but we do not have total stagnation at iteration $j$, we can still describe the stagnation of individual columns of $X_j^{(G)}$. Here, let $I$ denote an indexing set for some subset of column indices of $X_j^{(G)}$, and $E_i^{(L)} \in \mathbb{R}^{L \times |I|}$ be such $X_j^{(G)}E_i^{(L)} \in \mathbb{C}^{L \times |I|}$ has only as columns those from $X_j^{(G)}$ corresponding to the indices of $I$.

**Theorem 5.2.** Let the same assumptions as Theorem 5.1 hold. Suppose that at iteration $j$ block GMRES stagnates only for columns corresponding to the indices in $I$, i.e.,

$$X_j^{(G)}E_i^{(L)} = X_{j-1}^{(G)}E_i^{(L)}.$$

Then the associate columns of the $j$th generalized FOM approximation can be written

$$\hat{X}_j^{(F)} = X_{j-1}^{(G)} + (X_j^{(G)} - X_{j-1}^{(G)}) U + T$$

where $U \in \mathbb{C}^{L \times L}$ has zeros at rows with indices corresponding to $I$ and each column of $T$ is in $K_j(A, R_0)$.

Thus, for any column of block GMRES that stagnates at iteration $j$, we have that the associated column of the $j$th generalized block FOM approximation can be written as the sum of the associated column of the block GMRES approximation and a linear combination of the differences between the $j$th and $(j-1)$st block GMRES approximations for the non-stagnated columns plus one additional vector from $K_j(A, R_0)$. Note that we have omitted the here that $U$ and $T$ are related such that we can choose $T$ to be arbitrarily small (but nonzero) by adjusting $U$. See [39] Theorem 3.9] for more details.

Since for each shifted system, we can interpret Algorithms 1 and 2 as block GMRES and FOM applied to the dummy problem (5.1), we have the following theorem describing the circumstances under which Algorithm 1 stagnates and how this relates to Algorithm 2.

**Theorem 5.3.** Let $x_j^{(G)}(\sigma_i)$ and $\hat{x}_j^{(F)}(\sigma_i)$ be the $j$th approximations for shifted system $i$ produced by Algorithms 1 and 2, respectively. Furthermore, let $\hat{X}_j^{(G)}$ and $\hat{X}_j^{(F)}$ be the block GMRES and generalized FOM approximations produced at iteration...
for $\sigma_i$. Then if Algorithm [1] stagnates with $\mathbf{x}_{j}^{(G)}(\sigma_i) = \mathbf{x}_{j-1}^{(G)}(\sigma_i)$, one of the following two statements must hold,

1. Block GMRES applied to (5.1) totally stagnates at iteration $j$ with $\mathbf{X}_j^{(G)} = \mathbf{X}_{j-1}^{(G)}$ and we have $\mathbf{x}_{j}^{(F)}(\sigma_i) = \mathbf{x}_{j-1}^{(G)}(\sigma_i)$.

2. Block GMRES applied to (5.1) stagnates for some columns of $\mathbf{X}_j^{(G)}$, indexed by the set $I \subseteq \{1, 2, \ldots, L\}$ with $i \in I$, and we have

$$\mathbf{x}_{j}^{(F)}(\sigma_i) = \mathbf{x}_{j}^{(G)}(\sigma_i) + (\mathbf{X}_j^{(G)} - \mathbf{X}_{j-1}^{(G)}) \mathbf{u}(\sigma_i) + \mathbf{t}(\sigma_i),$$

where $\mathbf{u}(\sigma_i) \in \mathbb{C}^L$ has zeros at entries corresponding to elements of $I$, and $\mathbf{t}(\sigma_i) \in \mathbb{K}_j(\mathbf{A}, \mathbf{R}_0^{\sigma_i})$.

**Proof.** This theorem is simply a consequence of Theorems 5.1 and 5.2 along with the fact that for each shifted system, Algorithms [1] and [2] produce the approximations for each shifted system as if we applied block GMRES or block FOM to (5.1).

Observe that from [39, Theorem 3.9] we again know that $\mathbf{u}(\sigma_i)$ and $\mathbf{t}(\sigma_i)$ are related and that $\mathbf{t}(\sigma_i)$ can arbitrarily small.

### 5.2. Comparison of block GMRES to single-vector GMRES.

The performance of block methods and comparisons to single-vector counterparts have been well described by many different authors; see, e.g., [18, 28, 36, 38, 41]. In those cases, the analysis assumed one coefficient matrix and multiple right-hand sides. However, much of the convergence analysis does not specifically concern the fact that the block Krylov subspace arises from multiple right-hand sides. The residual bounds in, e.g., [36], which compare a single residual minimized over a block Krylov subspace to the same residual minimized over a single-vector Krylov subspace, are simply derived from the fact that the block GMRES minimization is performed over a larger space. Thus the following theorem falls directly from similar analysis done by previous researchers.

**Theorem 5.4.** Let $\mathbf{A}$, $\mathbf{B}$, $\sigma$, $\mathbf{X}_0$ and $\mathbf{R}_0^{\sigma}$ be defined as above. Let $j$ be such that $\dim \mathbb{K}_j(\mathbf{A}, \mathbf{R}_0^{\sigma}) = jL$. Let $\mathbf{X}_j$ be the block approximation which results from the application of $j$ iterations of the shifted block GMRES algorithm (Algorithm [1]). Furthermore, let $\tilde{\mathbf{X}}_j$ be the block approximation which results from applying the single-vector GMRES algorithm to each shifted system individually. Then we have the following residual norm bound for each $i = 1, 2, \ldots, L$,

$$\|\mathbf{B}(\cdot, i) - (\mathbf{A} + \sigma \mathbf{I})\mathbf{X}_m(\cdot, i)\| \leq \|\mathbf{B}(\cdot, i) - (\mathbf{A} + \sigma \mathbf{I})\tilde{\mathbf{X}}_m(\cdot, i)\|$$

**Proof.** This follows directly from the containment

$$\mathbb{K}_m(\mathbf{A} + \sigma \mathbf{I}, \mathbf{R}_m^{\sigma}(\cdot, i)) \subseteq \mathbb{K}_m(\mathbf{A}, \mathbf{R}_0^{\sigma})$$

which itself follows from the invariance (3.3) and the sum identity for block Krylov subspaces (3.2). This containment implies that the single-vector GMRES applied to the $i$th shifted system minimizes over a subspace contained in the subspace over which shifted block GMRES minimizes. Thus the minimum residual of shifted block GMRES is at least as small as that of single vector GMRES.

### 5.3. Block size growth with the number of shifts.

This method allows us to solve shifted linear systems simultaneously without a collinearity requirement but at a cost. The block size is not fixed; it is dependent on the number of shifts. As
we have stated earlier, the use of a block iteration in this context brings with it the
benefits associated to high-performance computing concerns. However, as the number
of shifts increases, the block size also increases, and eventually the block size will be
large enough that the benefits in data-movement efficiency will no longer outweigh the
costs of the larger block size. Thus, we must consider what modifications can be made
to accommodate this situation. The simplest would be to choose an optimal block
size $P$ and solve the shifted systems $P$ shifts at a time. However, an improvement on
this strategy would be to solve $P$ systems at a time and minimize the residuals of the
remaining systems according to the strategy advocated in [38].

5.4. Linear dependence of block Arnoldi vectors. As with any iteration
built upon a block Krylov subspace, we must address the possibility that during the
iteration, a dependent Arnoldi vector may be produced. As was shown in [18], the
notion of the grade of a Krylov subspace extends to the block setting. However,
unlike the single-vector case, the occurrence of a dependent Arnoldi vector does not
indicate that the method has achieved the grade of the block Krylov subspace (which
would imply convergence). Many different strategies has been suggested for gracefully
handling a dependent Arnoldi vector, see, e.g., [3, 6, 11, 12, 27].

However, we advocate replacing the dependent Arnoldi vector with a randomly
generated vector, as in [24, 28, 29, 38]. This serves the purpose of maintaining the
block size in order to continue to realize the data movement efficiencies associated to
block methods. However, unlike [39], there is no need in the nonsymmetric case to
generate these random vectors in advance.

6. Numerical Experiments. We present now two sets of experiments demon-
strating the performance of Algorithms 1 and 2 and compare their performance to the
algorithms Simoncini [34] and Frommer and Gläsner [14]. In the first experiment, we
demonstrate stagnation behavior of the block Algorithms to supplement the analysis
in Section 5.1. The rest of the experiments are performance tests for methods applied
to some sets of shifted systems. These experiments were performed on a Macbook Air
with an Intel Core 2 Duo 1.86 GHz processor, 2 GB of 1067 MHz DDR3 main memory
running OS-X 10.9.4. These experiments were performed with Matlab R2013a 64-bit
edition. In our Matlab implementation of FOM type methods, the FOM approxima-
tion is computed using the Moore-Penrose pseudoinverse $H_j$ only when the reciprocal
condition number of $H_j$ as computed by $\text{rcond}()$ fell below $10^{-10}$. We stress that
computing the generalized block FOM approximation using the pseudoinverse of $H_j$
is done here only to demonstrate theoretical results and for stability purposes. This
should not be seen as an advocation for the use of the pseudoinverse in a production
code.

In all comparison experiments, we judge algorithms by iteration counts rather
than timings. The matrices used in the experiments are relatively small. Thus the
expense of a matrix-vector product will still be dominated by FLOPS rather than
data movement. In data movement costs, it has been shown [28] that block matrix
vector products are only slightly more expensive than single matrix-vector products
(for moderately sized blocks). Thus for problems with millions or even billions of
unknowns, the benefits of using a block method would outweigh the costs. In these
experiments, we will not realize those benefits. Furthermore, all methods were imple-
mented in Matlab, and thus the overhead costs of Matlab itself render it difficult to
obtain accurate timings for these experiments. Thus, we compare iteration counts in
the following experiments. To be fair, though, for Experiments 6.6 and 6.7 for our
block methods, we also show the number of iterations multiplied by a block matvec versus single matvec cost multiplier. This factor of 3.3 was determined by performing single versus block matvec timings with matrices of the same size and sparsity pattern as those used in Experiments 6.6 and 6.7 using compiled Trilinos libraries [1].

We performed experiments with two different sets of matrices. Experiments 6.1 and 6.2 concern the stagnation of shifted block GMRES (Algorithm 1). We constructed a toy examples used, e.g., in [7] to demonstrate perfect stagnation. Let $A_{st} \in \mathbb{R}^{30 \times 30}$ be defined as the matrix which acts upon the Euclidean basis as follows,

$$A_{st} e_i = \begin{cases} e_1 & \text{if } i = 30 \\ e_{i+1} & \text{otherwise} \end{cases}.$$  

(6.1)

From this matrix and appropriately chosen right-hand sides, we can generate problems for which Algorithm 1 is guaranteed to stagnate.

The second set of matrices, used in Experiments 6.3 – 6.8 are from two sets of Lattice QCD matrices (Group 1 and Group 2) which are of sizes $3072 \times 3072$ and $49152 \times 49152$, respectively, available at [9]. With each such matrix $D$ is provided a number called $\kappa_c$ such that $I - \tilde{\kappa}D$ is positive-real for all $0 \leq \tilde{\kappa} \leq \kappa_c$. We can equivalently state that the matrix $-D + \kappa I$ is positive-real for all $1/\kappa_c \leq \kappa < \infty$. For each $D$, we generate a base matrix $A = -D + (10^{-3} + \kappa_c)I$ and choose only positive shifts to create our shifted family of linear systems. Coefficient matrices with smaller shifts yield more poorly conditioned systems requiring more iterations to solve.

![Relative two-norm residual curves for stagnating shifted block GMRES and shifted block FOM for the $30 \times 30$ shift matrix. The black solid and dashed curves correspond respectively to the block shifted FOM and GMRES residuals for the base system. Similarly, the gray solid and dashed curves, respectively, correspond to the shifted system.](image.png)

6.1. Stagnation of block-shifted GMRES and FOM. Using $A_{st}$, we can construct a problem for which Algorithm 1 will stagnate; a family of two linear systems of the form (1.1) with one shift $\sigma = 2$. The right-hand side for the base system is of the form $b = e_1$ and the right-hand side for the shifted system is $b(\sigma) = e_{15} + 2e_{16}$. From the construction of $A_{st}$, we know $(A_{st} + I)^{-1}b(\sigma) = e_{15}$ and that absent any replacement of dependent basis vectors, there will be 15 iterations of total stagnation for the base system. We see that we also have as many iterations of near stagnation for the shifted system.
6.2. Stagnation of block-shifted GMRES and FOM with dependence of an Arnoldi vector. By adding a third shifted system, now with $\sigma_1 = 2$ and $\sigma_2 = 3$, with the third right-hand side constructed so that a dependent Arnoldi vector will be produced at iteration 5, we still retain some exact stagnation behavior of the unshifted system, but after the dependent vector is replaced with a random one, we see instead near-stagnation of the approximation for the base system. As in Experiment 6.1, we construct the $b = e_1$ and $b(\sigma_1) = e_{15} + 2e_{16}$. Now we set $b(\sigma_2) = e_{20} + 2e_{21}$.

6.3. Block sGMRES vs. sFOM. In Figure 6.3, we compare the convergence histories measured with the Frobenius norm of the relative residual of block sGMRES versus block sFOM. The matrix used is the first from Group 1, and the shifts were

\{0.0001, 0.0002, 0.0003, 0.0004, 0.001, 0.002, 0.003, 0.004, 0.01, 0.02, 0.03, 0.04\}.
The right-hand side for each shifted system is the same randomly generated vector, and since our initial approximation for each shifted system is the zero vector, we begin with collinear residuals. Here, we observe that for this particular family of shifted systems, the performances are quite similar.

![Graph 6.4](block_sGMRESvsFrommerGlässnersGMRES.jpg)

**Fig. 6.4.** Convergence comparison between shifted block GMRES method and the method of Frommer and Glässner for restart parameter $m = 40$ and twelve shifts for the fourth matrix from Group 1. We began with collinear residuals (same right hand side and all zero initial approximations)

6.4. **Block sGMRES vs. Frommer and Glässner’s GMRES.** In Figure 6.4, we compare convergence histories of block sGMRES with the shifted GMRES method of Frommer and Glässner for the same matrix, shifts, right-hand sides, and initial approximations as before. Comparing iteration-for-iteration, we see that our new method converges in fewer iterations than the shifted GMRES algorithm.

![Graph 6.5](block_sFOMvsSimonicinisFOM.jpg)

**Fig. 6.5.** Convergence comparison between shifted block FOM method and the method of Simoncini for restart parameter $m = 40$ and twelve shifts for the fourth matrix from Group 1. We began with collinear residuals (same right hand side and all zero initial approximations)

6.5. **Block sFOM vs. Simoncini’s sFOM.** In Figure 6.5, for the same matrix, shifts, right-hand sides, and initial approximations as before, we compare our block sFOM method with Simoncini’s shifted restarted FOM method. Again, we see that our method converges in fewer iterations.
Table 6.1
Comparison of Algorithms 1 and 2 with the shifted GMRES method of Frommer and Glässner and the shifted FOM method of Simoncini for four matrices from Group 2. For three of the matrices, the proposed methods yield improved matvec counts, but greater costs when the block matrix-vector product cost multiplier is used. This ratio will be different for different matrices and in different computing environments. The other matrices not shown yielded similar results.

| Method              | Matrix 1 | Matrix 2 | Matrix 3 | Matrix 4 |
|---------------------|----------|----------|----------|----------|
| Sh. Bl. GMRES (mv ×3.3) | 602 (2408) | 1759 (5805) | 634 (2092) | 555 (1832) |
| sGMRES              | 938      | 1090     | 665      | 597      |
| Sh. Bl. FOM (mv ×3.3) | 982 (3241) | 1195 (3944) | 742 (2449) | 733 (2419) |
| sFOM                | 1177     | 1242     | 756      | 882      |

6.6. Mat-Vec counts comparing four methods. In Table 6.1 we compared the performance of all four methods (in terms of timings and matrix-vector products) for solving all seven systems from Group 2. Right-hand sides were generated as in previous experiments, and the shifts were {.0001,.0002,.01,.02}. In addition to comparing block iteration counts versus single iteration counts, we also provide block iteration counts multiplied by a cost multiplier of 3.3, derived from timing experiments comparing matrix single-vector products and matrix block-vector products for matrices of the same size and sparsity as those used in this experiment. Using this metric, the shifted GMRES method [14] and shifted FOM method [34] outperform our methods.

Table 6.2
Comparison of block sFOM and block sGMRES with their sequentially applied counterparts when each initial residuals of the shifted systems are not collinear for one matrix from Group 2. Both in terms of iteration counts and when multiplying block iterations by the ratio of block matvec cost to single matvec cost, we see that our proposed methods outperform their single-vector counterparts.

| Method              | Matvecs | Block Matvecs ×3.3 |
|---------------------|---------|--------------------|
| Shifted Block GMRES | 525     | 1732               |
| Sequential GMRES    | 2836    | *                  |
| Shifted Block FOM   | 602     | 1987               |
| Sequential FOM      | 3126    | *                  |

6.7. Mat-Vec counts for unrelated initial residuals. In Table 6.2 we compare the performance of our two methods when the right-hand sides are unrelated, i.e., a situation in which shifted FOM and shifted GMRES are not applicable. We also compare both methods against simply applying GMRES and FOM sequentially to each shifted system. The right-hand side for each system is generated randomly, and again the shifts were {.0001,.0002,.01,.02}. In terms of matrix-vector product counts, our methods are clearly superior, and this remains the case when we use the block iteration cost multiplier. For unrelated right-hand sides, it is apparent that great speedups can be attained.

6.8. Effect of random block in initial cycle of block sFOM. Observe that if we begin with collinear residuals, we described a method in Section 4.1.2 in which we apply a block FOM cycle in which the block Krylov subspace is generated by one residual and \( s - 1 \) random vectors. The question arises: how variable is the performance of this method for different sets of random vectors. To shed light on the answer, for the first matrix from Group 2, the same shifts as in Experiment 6.7
same right-hand side generated randomly for all shifted systems, and a zero vector initial approximation for all systems, we applied the shifted block FOM algorithm implemented with this random vector strategy to these shifted systems 200 times. Since the initial residuals are collinear, an initial cycle of block FOM with random vectors is executed in this situation. We recorded the number of iterations to convergence for each experiment, each with a different set of random vectors being generated in that initial cycle. In Figure 6.6 we plot a histogram for the 200 iteration counts. As one can appreciate, there is a large variation in performance ($\approx$ 60 iterations), though a plurality of the iteration counts are clustered near the mean.

7. Conclusions. We have presented two new Krylov subspace methods for simultaneously solving shifted linear systems. By building a block Krylov subspace from the initial residual for each system, we eliminate the collinear residual restriction, both that we must begin with collinear initial residuals and that the residuals must be collinear at restart. Furthermore, by basing our new methods on block Krylov subspaces, we realize the benefits in data movement costs associated to block sparse matrix operations. Numerical experiments demonstrate both the validity of the methods and that they can outperform their single-vector counterparts.

Acknowledgments. We thank Jen Pestana for closely reading an earlier version of this manuscript and offering suggestions and corrections.

REFERENCES

[1] The Trilinos Project website., Accessed February 24, 2012. http://trilinos.sandia.gov/packages/.

[2] Mian Ilyas Ahmad, Daniel B. Szyld, and Martin B. van Gijzen, Preconditioned multishift BiCG for $H_2$-optimal model reduction, Tech. Report 12-06-15, Department of Mathematics, Temple University, June 2012.

[3] José I. Aliaga, Daniel L. Boley, Roland W. Freund, and Vicente Hernández, A Lanczos-type method for multiple starting vectors, Mathematics of computation, 69 (2000), pp. 1577–1602.

[4] James Baglama, Dealing with linear dependence during the iterations of the restarted block Lanczos methods, Numerical Algorithms, 25 (2000), pp. 23–36.
[5] Manuel Baumann and Martin B. van Gijzen, *Nested krylov methods for shifted linear systems*, Tech. Report 14-01, Delft University of Technology, 2014.

[6] Sebastian Birk and Andreas Frommer, *A deflated conjugate gradient method for multiple right-hand sides and multiple shifts*, (In preparation).

[7] Peter N. Brown, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

[8] Dean Darnell, Ronald B. Morgan, and Walter Wilcox, *Deflated GMRES for systems with multiple shifts and multiple right-hand sides*, Linear Algebra and its Applications, 429 (2008), pp. 2145–2143.

[9] Timothy A. Davis and Yifan Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Softw., 38 (2011), pp. 1:1–1:25.

[10] Augustin A. Dubrulle, *Retooling the method of block conjugate gradients*, Electronic Transactions on Numerical Analysis, 12 (2001), pp. 216–233 (electronic).

[11] Roland W. Freund and Manish Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, in Proceedings of the Fifth Conference of the International Linear Algebra Society (Atlanta, GA, 1995), vol. 254, 1997, pp. 119–157.

[12] Roland W. Freund and Noël M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numerische Mathematik, 60 (1991), pp. 315–339.

[13] Andreas Frommer, *BiCGStab(l) for families of shifted linear systems*, Computing, 70 (2003), pp. 87–109.

[14] Andreas Frommer and Uwe Glässner, *Restarted GMRES for shifted linear systems*, SIAM Journal on Scientific Computing, 19 (1998), pp. 15–26.

[15] Andreas Frommer, Stephan Güskens, Thomas Lippert, Bertold Nöckel, and Klaus Schilling, *Many masses on one stroke: Economic computation of quark propagators*, International Journal of Modern Physics C, 6 (1995), pp. 627–638.

[16] Martin H. Gutknecht and Thomas Schmelzer, *A QR–decomposition of block tridiagonal matrices generated by the block lanczos process*, in Proceedings of the 17th IMACS World Congress, Paris, 2005, Ecole Central de Lille, Villeneuve d’Ascq, France, July 2005, pp. 1–8, on CD only.

[17] Martin H. Gutknecht and Thomas Schmelzer, *Updating the QR decomposition of block tridiagonal and block Hessenberg matrices*, Applied Numerical Mathematics, 58 (2008), pp. 871–883.

[18] The block grade of a block Krylov space, Linear Algebra and its Applications, 430 (2009), pp. 174–185.

[19] Mark Hoemmen, *Communication-avoiding Krylov subspace methods*, PhD thesis, University of California Berkeley, 2010.

[20] Beat Jegerlehner, *Krylov space solvers for sparse linear systems*, Tech. Report IUHET-353, Indiana University, 1996.

[21] Misha E. Kilmer and Eric de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM Journal on Scientific Computing, 27 (2006), pp. 2140–2166.

[22] Sabrina Kirchner, *IDR-Verfahren zur Lösung von Familien geshifteter linearer Gleichungssysteme*, master’s thesis, Bergische Universität Wuppertal, Department of Mathematics, Wuppertal, Germany, 2011.

[23] Daniel Kressner and Christine Tobler, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.

[24] Julian Langou, *Iterative methods for solving linear systems with multiple right-hand sides*, PhD thesis, CERFACS, France, 2003.

[25] Karl Meerbergen and Zhaolin Bai, *The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides*, SIAM J. Matrix Anal. Appl., 31 (2009/10), pp. 1642–1662.

[26] Marko Hoymuddin, Mark Hoemmen, James Demmel, and Katherine Yelick, *Minimizing communication in sparse matrix solvers*, in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC ’09, New York, NY, USA, 2009, ACM, pp. 36:1–36:12.

[27] Diann P. O’Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra and its Applications, 29 (1980), pp. 293–322.

[28] Michael L. Parks, Kirk M. Soodhalter, and Daniel B. Szyld, *Block krylov subspace recycling*, In Preparation.

[29] Mickael Robbe and Miloud Sadkane, *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra and its Applications, 419 (2006), pp. 265–285.

[30] Yousef Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Mathematics of Computation, 37 (1981), pp. 105–126.
[31] Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, Second ed., 2003.

[32] Yousef Saad and Martin H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

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

[34] Valeria Simoncini, Restarted full orthogonalization method for shifted linear systems, BIT. Numerical Mathematics, 43 (2003), pp. 459–466.

[35] Valeria Simoncini and Efstratios Gallopoulos, An iterative method for nonsymmetric systems with multiple right-hand sides, SIAM Journal on Scientific Computing, 16 (1995), pp. 917–933.

[36] Valeria Simoncini and Efstratios Gallopoulos, Convergence properties of block GMRES and matrix polynomials, Linear Algebra and its Applications, 247 (1996), pp. 97–119.

[37] Valeria Simoncini and Daniel B. Szyld, Recent computational developments in Krylov subspace methods for linear systems, Numerical Linear Algebra with Applications, 14 (2007), pp. 1–59.

[38] Kirk M. Soodhalter, A block MINRES algorithm based on the banded lanczos method, Numerical Algorithms, (2014), pp. 1–22.

[39] Kirk M. Soodhalter, Stagnation of block GMRES and its relationship to block FOM, arXiv 1411.7801, 2014.

[40] Kirk M. Soodhalter, Daniel B. Szyld, and Fei Xue, Krylov subspace recycling for sequences of shifted linear systems, Applied Numerical Mathematics, 81C (2014), pp. 105–118.

[41] Brigitte Vital, Etude de quelques méthodes de résolution de problèmes lineaires de grande taille sur multiprocesseur, PhD thesis, Université de Rennes, 1990.

[42] Gang Wu, Yan-Chun Wang, and Xiao-Qing Jin, A preconditioned and shifted GMRES algorithm for the PageRank problem with multiple damping factors, SIAM J. Sci. Comput., 34 (2012), pp. A2558–A2575.