Electronic Transactions on Numerical Analysis.
Volume 45, pp. 499–523, 2016.
Copyright © 2016, Kent State University.
ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

TWO RECURSIVE GMRES-TYPE METHODS FOR SHIFTED LINEAR SYSTEMS WITH GENERAL PRECONDITIONING[*]

KIRK M. SOODHALTER[†]

Abstract. We present two minimum residual methods for solving sequences of shifted linear systems, the right-preconditioned shifted GMRES and shifted Recycled GMRES algorithms which use a seed projection strategy often employed to solve multiple related problems. These methods are compatible with a general preconditioning of all systems, and, when restricted to right preconditioning, require no extra applications of the operator or preconditioner. These seed projection methods perform a minimum residual iteration for the base system while improving the approximations for the shifted systems at little additional cost. The iteration continues until the base system approximation is of satisfactory quality. The method is then recursively called for the remaining unconverged systems. We present both methods inside of a general framework which allows these techniques to be extended to the setting of flexible preconditioning and inexact Krylov methods. We present some analysis of such methods and numerical experiments demonstrating the effectiveness of the proposed algorithms.

Key words. Krylov subspace methods, shifted linear systems, parameterized linear systems, quantum chromodynamics

AMS subject classifications. 65F10, 65F50, 65F08

1. Introduction.

We develop techniques for solving a family (or a sequence of families) of linear systems in which the coefficient matrices differ only by a scalar multiple of the identity. There are many applications which warrant the solution of a family of shifted linear systems such as those arising in lattice quantum chromodynamics (QCD) (see, e.g., [14]) as well as other applications such as Tikhonov-Phillips regularization, global methods of nonlinear analysis, and Newton trust region methods [5]. The goal is to develop a framework in which minimum residual methods can be applied to shifted systems in a way that

(a) allows us to exploit the relationships between the coefficient matrices and
(b) is compatible with general (right) preconditioning.

In this paper, we use such a framework to propose two new methods: one which is built on top of the GMRES method [32] for solving a family of shifted systems (cf. (2.1)) and one which is built on top of a GCRO-type augmented Krylov method [10] which, when paired with a harmonic Ritz vector recycling strategy [25, 26], is an extension of the GCRO-DR method [27] to solve a sequence of shifted system families (cf. (2.2)). To do this, we use a seed projection strategy often proposed for use in conjunction with short-term recurrence iterative methods [6, 7, 19, 29].

The rest of this paper is organized as follows. In the next section, we discuss some previous strategies to treat such problems and discuss some of their limitations. In Section 3, we review the minimum residual Krylov subspace method GMRES as well as two GMRES variants, one for shifted linear systems and the other extending GMRES to the augmented Krylov subspace setting, i.e., Recycled GMRES. In Section 4, we present a general framework to perform minimum residual projections of the shifted system residuals with respect to the search space generated for the base system. In Section 4.1 we use this framework to derive our *shifted GMRES* method, and in Section 4.2 we derive a *shifted Recycled GMRES* method. In Section 5, we present some analysis of the expected performance of these methods. In Section 6, we present some numerical results before concluding in Section 7.

[*]Received March 28, 2014. Accepted October 13, 2016. Published online on December 9, 2016. Recommended by A. Frommer.

[†]Johann Radon Institute for Computational and Applied Mathematics, Transfer Group, Altenbergerstraße 69, A-4040 Linz, Austria (kirk.soodhalter@ricam.oeaw.ac.at).

2. Background. Consider a family of shifted linear systems, which we parameterize by ℓ, i.e.,

$$\left(\mathbf{A}+\sigma^{(\ell)}\mathbf{I}\right)\mathbf{x}^{(\ell)}=\mathbf{b} \qquad \text{for } \ell=1,\ldots,L. \tag{2.1}$$

We call the numbers $\left\{\sigma^{(\ell)}\right\}_{\ell=1}^{L}\subset\mathbb{C}$ *shifts*, $\mathbf{A}$ the *base matrix*, and $\mathbf{A}+\sigma\mathbf{I}$ a *shifted matrix*. For notational simplicity, we have assumed here that $\mathbf{b}$ does not depend on $\sigma^{(\ell)}$, but we could just as well discard this assumption as none of the methods developed in this paper require the right-hand sides (or the initial residuals) of the equations in (2.1) to have any relationship. Systems of the form (2.1) are called *shifted linear systems*. Krylov subspace methods have been proposed to simultaneously solve this family of systems; see, e.g., [8, 12, 13, 20, 37]. These methods satisfy requirement (a) but are not compatible with general preconditioning strategies as they rely on the invariance of the Krylov subspace under constant shift of the coefficient matrix; cf. (3.4). Specially chosen polynomial preconditioners, however, have been shown to be compatible with such methods; see, e.g., [1, 3, 4, 18, 23, 34, 43].

We can introduce an additional parameter i, which indexes a sequence of matrices $\{\mathbf{A}_i\}\subset\mathbb{C}^{n\times n}$, and for each i, we solve a family of linear systems of the form

$$\left(\mathbf{A}_i+\sigma_i^{(\ell)}\mathbf{I}\right)\mathbf{x}_i^{(\ell)}=\mathbf{b}_i \qquad \text{for } \ell=1\ldots L_i. \tag{2.2}$$

We consider the case that the right-hand side varies with respect to $\mathbf{A}_i$ but not for each shift, but this assumption is again only for simplicity and could be discarded. Augmented Krylov subspace methods have been proposed for efficiently solving a sequence of linear systems with a slowly changing coefficient matrix, allowing important spectral information generated while solving $\mathbf{A}_i\mathbf{x}_i=\mathbf{b}_i$ to be used to augment the Krylov subspace generated when solving $\mathbf{A}_{i+1}\mathbf{x}_{i+1}=\mathbf{b}_{i+1}$; see, e.g., [27, 33, 42]. In cases such as a Newton iteration, these matrices are available one at a time, while in a case such as an implicit time-stepping scheme, the matrix may not change at all.

In [41], the authors explored solving a family of shifted systems over an augmented Krylov subspace. Specifically, the goal was to develop a method which solves the family of systems simultaneously, using one augmented subspace to extract all candidate solutions, which also had a fixed storage requirement, independent of the number of shifts L. It was shown that in general within the framework of GMRES for shifted systems [13] and subspace recycling [27], such a method *does not exist*. In the context of subspace recycling for Hermitian linear systems, in the absence of preconditioning, Kilmer and de Sturler proposed a MINRES method in a subspace recycling framework which simultaneously solves multiple non-Hermitian systems, which all differ from a real-symmetric system by a complex multiple of the identity [20]. This is done by minimizing the shifted residuals over the augmented Krylov subspace that is built using the symmetric Lanczos process. In this paper, we focus exclusively on problems in which the base coefficient matrices $\mathbf{A}_i$ are *non-Hermitian*.

A conclusion one can draw from [41] is that we should avoid methods relying on the invariance of Krylov subspaces under a constant shift of the identity; cf. (3.4). Relying on this invariance imposes restrictions on our ability to develop an algorithm. Furthermore, relying on this shift invariance means we cannot use arbitrary preconditioners. General preconditioners are unavailable if we want to exploit shift invariance as Krylov subspaces generated by preconditioned systems are not invariant with respect to a shift in the coefficient matrix. In the case that preconditioning is not used, a subspace recycling technique has been proposed [40] that is built on top of the Sylvester equation interpretation of (2.1) observed by Simoncini in [36]. However, this is also not compatible with general preconditioning.

Learning from the results in [41], we focus on methods which do not rely on shift invariance. Rather than focusing on specific Krylov subspace techniques (augmented or not), we instead begin by developing a general framework of minimum residual projection techniques for shifted linear systems. In this framework, we extract candidate solutions for all shifted systems from the augmented Krylov subspace for one linear system, and we select each candidate solution according to a minimum residual Petrov-Galerkin condition. This framework is compatible with arbitrary right preconditioners, and the computational cost for each additional shifted system is relatively small but nontrivial. By specifying subspaces once the framework is developed, we derive minimum residual methods for shifted systems that are compatible with general right preconditioning. Though not considered in this paper, the framework is also compatible with flexible and inexact Krylov methods. These methods descend from the Lanczos-Galerkin seed methods, see, e.g., [6, 7, 19, 29].

In this work, we restrict ourselves to right-preconditioned methods. Doing this allows us to derive methods which require extra storage but no extra applications of the operator or preconditioner, and we minimize the unpreconditioned residual's 2-norm rather than some other norm; see [35] for more details.

3. Preliminaries. We begin with a brief review of Krylov subspace methods as well as techniques of subspace recycling and for solving shifted linear system. Recall that for many Krylov subspace iterative methods for solving

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.1}$$

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

$$\mathcal{K}_j(\mathbf{A}, \mathbf{u}) = \text{span}\left\{\mathbf{u}, \mathbf{A}\mathbf{u}, \ldots, \mathbf{A}^{j-1}\mathbf{u}\right\}$$

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

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_{j+1}\overline{\mathbf{H}}_j \tag{3.2}$$

with $\overline{\mathbf{H}}_j \in \mathbb{C}^{(j+1)\times j}$; see, e.g., [31, Section 6.3] and [38]. Let $\mathbf{x}_0$ be an initial approximation to the solution of a linear system that we wish to solve and $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ be the initial residual. At iteration j, we choose $\mathbf{x}_j = \mathbf{x}_0 + \mathbf{t}_j$, with $\mathbf{t}_j \in \mathcal{K}_j(\mathbf{A}, \mathbf{r}_0)$. In GMRES [32], $\mathbf{t}_j$ satisfies

$$\mathbf{t}_j = \underset{\mathbf{t}\in\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)}{\text{argmin}} \left\|\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{t})\right\|,$$

which is equivalent to solving the smaller minimization problem

$$\mathbf{y}_j = \underset{\mathbf{y}\in\mathbb{C}^j}{\text{argmin}} \left\|\overline{\mathbf{H}}_j\mathbf{y} - \left\|\mathbf{r}_0\right\| \mathbf{e}_1^{(j+1)}\right\|, \tag{3.3}$$

where $\mathbf{e}_\mu^{(i)}$ denotes the μth Cartesian basis vector in $\mathbb{C}^i$. We then set $\mathbf{x}_j = \mathbf{x}_0 + \mathbf{V}_j\mathbf{y}_j$. Recall that in restarted GMRES, often called GMRES(m), we run an m-step cycle of the GMRES method and compute an approximation $\mathbf{x}_m$. We halt the process, discard $\mathbf{V}_m$, and restart with the new residual. This process is repeated until we achieve convergence.

Many methods for the simultaneous solution of shifted systems (see, e.g., [8, 12, 13, 14, 21, 37]) take advantage of the fact that for any shift $\sigma \in \mathbb{C}$, the Krylov subspace generated by $\mathbf{A}$ and $\mathbf{b}$ is invariant under the shift, i.e.,

$$\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \mathcal{K}_j(\mathbf{A} + \sigma\mathbf{I}, \tilde{\mathbf{b}}), \tag{3.4}$$

as long as the starting vectors are collinear, i.e., $\tilde{\mathbf{b}} = \beta\mathbf{b}$ for some $\beta \in \mathbb{C} \setminus \{0\}$, with a shifted Arnoldi relation similar to (3.2)

$$(\mathbf{A} + \sigma\mathbf{I})\mathbf{V}_j = \mathbf{V}_{j+1}\overline{\mathbf{H}}_j(\sigma), \tag{3.5}$$

where $\overline{\mathbf{H}}_j(\sigma) = \overline{\mathbf{H}}_j + \sigma \begin{bmatrix} \mathbf{I}_{j\times j} \\ \mathbf{0}_{1\times j} \end{bmatrix}$. This collinearity must be maintained at restart. In [41], this was shown to be a troublesome restriction when attempting to extend such techniques to augmented Krylov methods. In the case of GMRES, Frommer and Glässner [13] were able to overcome this by minimizing only one residual in the common Krylov subspace and forcing the others to be collinear. This strategy also works in the case of GMRES with deflated restarts [8] because of properties of the augmented space generated using harmonic Ritz vectors. However, it was shown in [41] that residual collinearity cannot be enforced in general. Furthermore, it is not compatible with general preconditioning. The invariance (3.4) can lead to great savings in memory costs but with a loss of algorithmic flexibility. Thus in Section 4, we explore an alternative.

We briefly review Recycled GMRES for non-Hermitian matrices $\mathbf{A}$. Augmentation techniques designed specifically for Hermitian linear systems have also been proposed; see, e.g., [19, 33, 42]. For a more general framework for these types of methods, see [16], the elements of which form a part of the thesis of Gaul [15], which contains a wealth of information on this topic. Gaul and Schlömer describe recycling techniques in the context of self-adjoint operator equations in a general Hilbert space [17].

We begin by clarifying what we mean by Recycled GMRES. We use this expression to describe the general category of augmented GMRES-type methods which are then differentiated by the choice of augmenting subspace. As we subsequently explain, these methods can all be formulated as a GMRES iteration being applied to a linear system premultiplied with a projector. The intermediate solution to this projected problem can then be further corrected yielding a minimum residual approximation for the original problem over an augmented Krylov subspace. GCRO-DR [27] is one method in this category, in which the augmented subspace is built from harmonic Ritz vectors.

The GCRO-DR method represents the confluence of two approaches: those descending from the implicitly restarted Arnoldi method [22], such as Morgan's GMRES-DR [24], and those descending from de Sturler's GCRO method [10]. GMRES-DR is a restarted GMRES algorithm, where at the end of each cycle, harmonic Ritz vectors are computed, and a subset of them is used to augment the Krylov subspace generated at the next cycle. The GCRO method allows the user to select the optimal correction over arbitrary subspaces. This concept is extended by de Sturler in [11], where a framework is provided to optimally reduce convergence rate slowdown due to discarding information upon restart. This algorithm is called GCROT. A simplified version of the GCROT approach, based on restarted GMRES (called LGMRES) is presented in [2]. Parks et al. in [27] combine the ideas of [24] and [11] and extend them to a sequence of slowly-changing linear systems. They call their method GCRO-DR. This method and GCROT are Recycled GMRES methods.

Suppose we are solving (3.1), and we have a k-dimensional subspace $\mathcal{U}$ whose image under the action of $\mathbf{A}$ is $\mathcal{C} = \mathbf{A}\mathcal{U}$. Let $\mathbf{P}$ be the orthogonal projector onto $\mathcal{C}^{\perp}$. Let $\mathbf{x}_0$ be such that $\mathbf{r}_0 \in \mathcal{C}^{\perp}$. At iteration m, the Recycled GMRES method generates the approximation

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{s}_m + \mathbf{t}_m,$$

where $\mathbf{s}_m \in \mathcal{U}$ and $\mathbf{t}_m \in \mathcal{K}_m(\mathbf{PA}, \mathbf{r}_0)$. The corrections $\mathbf{s}_m$ and $\mathbf{t}_m$ are chosen according to the minimum residual Petrov-Galerkin condition over the augmented Krylov subspace, i.e.,

$$\mathbf{r}_m \perp \mathbf{A}\left(\mathcal{U} + \mathcal{K}_m\left(\mathbf{PA}, \mathbf{r}_0\right)\right). \tag{3.6}$$

At the end of the cycle, an updated $\mathcal{U}$ is constructed, the Krylov subspace basis is discarded, and we restart. At convergence, $\mathcal{U}$ is saved to be used when solving the next linear system. This process is equivalent to applying GMRES to the projected problem

$$\mathbf{PA}\left(\widehat{\mathbf{x}}_0 + \mathbf{t}\right) = \mathbf{Pb} \tag{3.7}$$

where $\mathbf{t}_m$ is the mth GMRES correction for (3.7) and the second correction $\mathbf{s}_m \in \mathcal{U}$ is the orthogonal projection of $\mathbf{t}_m$ onto $\mathcal{U}$ where orthogonality is meant with respect to the inner product induced by the positive-definite matrix $\mathbf{A}^*\mathbf{A}$[1]; see, e.g., [15, 16].

Recycled GMRES can be described as a modified GMRES iteration. Let $\mathbf{U} \in \mathbb{C}^{n\times k}$ have columns spanning $\mathcal{U}$, scaled such that $\mathbf{C} = \mathbf{AU}$ has orthonormal columns. Then we can apply $\mathbf{P} = \mathbf{I} - \mathbf{CC}^*$ to $\mathbf{Av}_j$ using k steps of the Modified Gram-Schmidt process. The orthogonalization coefficients are stored in the mth column of $\mathbf{B}_m = \mathbf{C}^*\mathbf{AV}_m$, which is simply $\mathbf{B}_{m-1}$ with one new column appended. Let $\overline{\mathbf{H}}_m$ and $\mathbf{V}_m$ be defined as before but for the projected Krylov subspace $\mathcal{K}_m\left(\mathbf{PA}, \mathbf{r}_0\right)$. Enforcing (3.6) is equivalent to solving the GMRES minimization problem (3.3) for $\mathcal{K}_m\left(\mathbf{PA}, \mathbf{r}_0\right)$ and setting

$$\mathbf{s}_m = -\mathbf{UB}_m\mathbf{y}_m \quad \text{and} \quad \mathbf{t}_m = \mathbf{V}_m\mathbf{y}_m,$$

so that

$$\mathbf{x}_m = \mathbf{x}_0 - \mathbf{UB}_m\mathbf{y}_m + \mathbf{V}_m\mathbf{y}_m = \mathbf{x}_0 + \begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\begin{bmatrix}-\mathbf{B}_m\mathbf{y}_m \\ \mathbf{y}_m\end{bmatrix}.$$

This is a consequence of the fact that the Recycled GMRES least squares problem as stated in [27, Equation 2.13] can be satisfied exactly in the first k rows, which was first observed in [10]. The choice of the subspace $\mathcal{U}$ then determines the actual method.

4. A direct projection framework.

We develop a general framework of minimum residual methods for shifted linear systems which encompasses both unpreconditioned and preconditioned systems. This framework and the methods which follow from it are all based on the *Lanczos-Galerkin projection* scheme for solving multiple linear systems with the same or similar coefficient matrices and different right-hand sides. This was suggested in [28] and further analyzed and extended in, e.g., [6, 7, 19, 29]. In all variants of this scheme, one applies a Krylov subspace iteration to solve one of the linear systems and projects the residuals of the other systems using this already-generated Krylov subspace.

We propose to solve both a single family of shifted systems (2.1) and sequences of shifted system families of the form (2.2). However, it suffices to propose our method in a simpler setting in which we drop the index i and assume that there are only two systems, a base system and a shifted system. Thus for simplicity, we restrict our description to the following two model problems: the unpreconditioned problem

$$\mathbf{Ax} = \mathbf{b} \quad \text{and} \quad (\mathbf{A} + \sigma\mathbf{I})\mathbf{x}(\sigma) = \mathbf{b} \tag{4.1}$$

and the right-preconditioned problem

$$\mathbf{AM}^{-1}\mathbf{w} = \mathbf{b} \quad \text{and} \quad (\mathbf{A} + \sigma\mathbf{I})\mathbf{M}^{-1}\mathbf{w}(\sigma) = \mathbf{b}, \tag{4.2}$$

where $\mathbf{w}_0 = \mathbf{Mx}_0$ and $\mathbf{w}_0(\sigma) = \mathbf{Mx}_0(\sigma)$ and with the setting $\mathbf{x}_m = \mathbf{M}^{-1}\mathbf{w}_m$ and $\mathbf{x}_m(\sigma) = \mathbf{M}^{-1}\mathbf{w}_m(\sigma)$ after m iterations. In this context, we can propose minimum residual Krylov subspace methods in the cases that we do or do not have an augmenting subspace $\mathcal{U}$.

[1] We can write explicitly $\mathbf{s}_m = \mathbf{P}_{\mathcal{U}}\mathbf{t}_m$ where we define $\mathbf{P}_{\mathcal{U}} = \mathbf{U}\left(\mathbf{U}^*\mathbf{A}^*\mathbf{AU}\right)^{-1}\mathbf{U}^*\mathbf{A}^*\mathbf{A}$, which can be rewritten as $\mathbf{P}_{\mathcal{U}} = \mathbf{UC}^*\mathbf{A}$.

We describe the proposed methods in terms of a general sequence of nested subspaces,

$$\mathcal{S}_1 \subset \mathcal{S}_2 \subset \cdots \mathcal{S}_m \subset \cdots .$$

This allows us to cleanly present these techniques as minimum residual projection methods and later to give a clear analysis applicable to any method fitting into this framework. Then we can derive different methods by specifying $\mathcal{S}_m$, e.g., $\mathcal{S}_m = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$.

Let $\{\mathcal{S}_m\}_{i=1}^{m}$ be the nested sequence of subspaces produced by some iterative method for solving (4.1) or (4.2) after m iterations. In the unpreconditioned case (4.1), suppose that we have initial approximations $\mathbf{x}_0$ and $\mathbf{x}_0(\sigma)$ for the base and shifted systems, respectively. For conciseness, let us denote $\mathbf{A}(\sigma) = \mathbf{A} + \sigma\mathbf{I}$. At iteration m, we compute corrections $\mathbf{t}_m$, $\mathbf{t}_m(\sigma) \in \mathcal{S}_m$, which satisfy the minimum residual conditions

$$\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{t}_m) \perp \mathbf{A}\mathcal{S}_m \quad \text{and} \quad \mathbf{b} - \mathbf{A}(\sigma)(\mathbf{x}_0(\sigma) + \mathbf{t}_m(\sigma)) \perp \mathbf{A}(\sigma)\mathcal{S}_m. \tag{4.3}$$

In the preconditioned case (4.2) where we take, e.g., $\mathcal{S}_m = \mathcal{K}_m(\mathbf{AM}, \mathbf{r}_0)$, suppose that we begin with initial approximations $\mathbf{w}_0 = \mathbf{Mx}_0$ and $\mathbf{w}_0(\sigma) = \mathbf{Mx}_0(\sigma)$. Let us denote the preconditioned operators

$$\mathbf{A}_p = \mathbf{AM}^{-1} \quad \text{and} \quad \mathbf{A}_p(\sigma) = (\mathbf{A} + \sigma\mathbf{I})\mathbf{M}^{-1}.$$

At iteration m, we compute corrections $\mathbf{t}_m$, $\mathbf{t}_m(\sigma) \in \mathcal{S}_m$, which satisfy the minimum residual conditions

$$\mathbf{b} - \mathbf{A}_p(\mathbf{w}_0 + \mathbf{t}_m) \perp \mathbf{A}_p\mathcal{S}_m \quad \text{and} \quad \mathbf{b} - \mathbf{A}_p(\sigma)(\mathbf{w}_0(\sigma) + \mathbf{t}_m(\sigma)) \perp \mathbf{A}_p(\sigma)\mathcal{S}_m. \tag{4.4}$$

We emphasize that *the same preconditioner* is used for all systems.

In this framework, we assume that the minimizer for the base case is constructed via a predefined iterative method which generates the sequence $\{\mathcal{S}_m\}$. Therefore, it suffices to describe the residual projection for the shifted system. We can write the update of the shifted system approximation by explicitly constructing the orthogonal projector which is applied during a Petrov-Galerkin projection. Let $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ be a basis for $\mathcal{S}_m$ which we take as the columns of $\mathbf{S}_m \in \mathbb{C}^{n\times m}$. Then we can write this projection and update as

$$\begin{aligned} \mathbf{r}_m(\sigma) &= \mathbf{r}_0(\sigma) - \mathbf{A}(\sigma)\mathbf{S}_m\mathbf{y}_m(\sigma) \quad \text{and} \\ \mathbf{x}_m(\sigma) &= \mathbf{x}_0(\sigma) + \mathbf{S}_m\mathbf{y}_m(\sigma), \end{aligned} \tag{4.5}$$

where $\mathbf{y}_m(\sigma) = \mathbf{N}_m(\sigma)^{-1}\left(\mathbf{A}(\sigma)\mathbf{S}_m\right)^* \mathbf{r}_0(\sigma)$ and $\mathbf{N}_m(\sigma) = (\mathbf{S}_m^*\mathbf{A}(\sigma)^*\mathbf{A}(\sigma)\mathbf{S}_m)$ is the projection scaling matrix since we do not assume that $\mathbf{A}(\sigma)\mathbf{S}_m$ has orthonormal columns. For well-chosen $\mathcal{S}_m$, these projections can be applied using already-computed quantities.

In the following sections, we derive new methods by specifying the subspaces $\{\mathcal{S}_m\}$ and a matrix $\mathbf{S}_m$. This will define $\mathbf{N}_m(\sigma)$. We show that for these choices, $\mathbf{N}_m(\sigma)$ is composed of blocks which can be built from already-computed quantities. Thus, for appropriate choices of $\mathcal{S}_m$, either (4.3) or (4.4) can be applied with manageable additional costs.

We highlight that a strength of this framework is that we can develop methods for shifted systems on top of an existing iterative methods with a few modifications. As the framework only requires a sequence of nested subspaces, it is completely compatible with both standard Krylov subspace methods as well as flexible and inexact Krylov subspace methods.

4.1. A GMRES method for shifted systems.

In the case that we apply the GMRES iteration to the base system, at iteration m, our search space is $\mathcal{S}_m := \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, and the matrix $\mathbf{S}_m := \mathbf{V}_m$ has the first m Arnoldi vectors as columns. The projection and update (4.5) can be simplified due to the shifted Arnoldi relation (3.5). The matrix

$$\mathbf{N}_m(\sigma) := \overline{\mathbf{H}}_m(\sigma)^*\overline{\mathbf{H}}_m(\sigma) \in \mathbb{C}^{m\times m}$$

can be constructed from the already computed upper Hessenberg matrix. Thus the projection (4.3) can be rewritten as

$$\mathbf{x}_m(\sigma) = \mathbf{x}_0(\sigma) + \mathbf{V}_m(\sigma)\mathbf{y}_m(\sigma) \quad \text{and}$$
$$\mathbf{r}_m(\sigma) = \mathbf{r}_0(\sigma) - \mathbf{V}_{m+1}\overline{\mathbf{H}}_m(\sigma)\mathbf{y}_m(\sigma),$$

where $\mathbf{y}_m(\sigma) = \left(\overline{\mathbf{H}}(\sigma)^*\overline{\mathbf{H}}(\sigma)\right)^{-1}\overline{\mathbf{H}}_m(\sigma)^*\mathbf{V}_{m+1}^*\mathbf{r}_0(\sigma)$. As it can be appreciated, applying this is equivalent to solving the least squares problem

$$\mathbf{y}_m(\sigma) = \underset{\mathbf{y}\in\mathbb{C}^i}{\text{argmin}}\left\|\overline{\mathbf{H}}_m(\sigma)\mathbf{y} - \mathbf{V}_{m+1}^*\mathbf{r}_0(\sigma)\right\| \tag{4.6}$$

and setting $\mathbf{x}_m(\sigma) = \mathbf{x}_0(\sigma) + \mathbf{V}_m\mathbf{y}_m(\sigma)$. This method has some similarities with the GMRES method for shifted systems of Frommer and Glässner [13], which is derived from the invariance (3.4), but the methods are distinct. In the method proposed in [13], one must solve small linear systems for each shifted system and it is required that the initial residuals be collinear whereas here one must solve the small least-squares problem (4.6) with no collinearity requirement. Furthermore, what we propose does not guarantee convergence of all systems in one Krylov subspace whereas in [13], this is guaranteed under certain conditions. However, the strengths of the proposed method are that there is no requirement of collinearity for the initial residuals and the ability for preconditioning.

4.1.1. Preconditioning.

Introducing preconditioning into this setting presents complications. No longer can we use the shifted Arnoldi relation (3.5) as we could in the unpreconditioned case. However, by storing some extra vectors as in Flexible GMRES [30], one can enforce (4.4) with no additional application of the operator or preconditioner.

Recall that in right-preconditioned GMRES (see, e.g., [31, Sections 9.3.2 and 9.4.1]), $\mathcal{S}_m := \mathbf{M}^{-1}\mathcal{K}(\mathbf{A}_p, \mathbf{r}_0)$, and $\mathbf{S}_m := \mathbf{M}^{-1}\mathbf{V}_m$. This space is never explicitly constructed though, since if $\mathbf{y}_m$ is the solution to the GMRES least squares problem (3.3) in the preconditioned case, then we simply set $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{M}^{-1}(\mathbf{V}_m\mathbf{y}_m)$. However, in flexible GMRES, one must store this basis. For all $1 \leq i \leq m$, let $\mathbf{z}_i = \mathbf{M}^{-1}\mathbf{v}_i$, and let these vectors be the columns of $\mathbf{Z}_m \in \mathbb{C}^{n\times i}$ so that $\mathbf{Z}_m = \mathbf{M}^{-1}\mathbf{V}_m$.

With these vectors, one can enforce (4.4). Observe that we can write $\mathbf{A}_p(\sigma) = \mathbf{A}_p + \sigma\mathbf{M}^{-1}$. We explicitly project the residual but this time onto $\{\mathbf{A}_p(\sigma)\mathcal{K}_m(\mathbf{A}_p, \mathbf{r}_0)\}^\perp$,

$$\mathbf{r}_m(\sigma) = \mathbf{r}_0(\sigma) - \left(\mathbf{A}_p + \sigma\mathbf{M}^{-1}\right)\mathbf{V}_m\mathbf{N}_m(\sigma)^{-1}\left[\left(\mathbf{A}_p + \sigma\mathbf{M}^{-1}\right)\mathbf{V}_m\right]^*\mathbf{r}_0(\sigma), \tag{4.7}$$

where $\mathbf{N}_m(\sigma) = \left[\left(\mathbf{A}_p + \sigma\mathbf{M}^{-1}\right)\mathbf{V}_m\right]^*\left[\left(\mathbf{A}_p + \sigma\mathbf{M}^{-1}\right)\mathbf{V}_m\right]$. With the right-preconditioned shifted Arnoldi relation

$$\left(\mathbf{A}_p + \sigma\mathbf{M}^{-1}\right)\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma\mathbf{Z}_m$$

we rewrite

$$\mathbf{N}_m(\sigma) = \overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m + \sigma\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{Z}_m + \overline{\sigma}\mathbf{Z}_m^*\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + |\sigma|^2\mathbf{Z}_m\mathbf{Z}_m.$$

Thus, the approximation update and the residual projection (4.7) can be rewritten

$$\begin{aligned}\mathbf{x}_m(\sigma) &= \mathbf{x}_0(\sigma) + \mathbf{Z}_m\mathbf{y}_m(\sigma),\\ \mathbf{r}_m(\sigma) &= \mathbf{r}_0(\sigma) - \left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma\mathbf{Z}_m\right)\mathbf{y}_m(\sigma),\end{aligned}$$

where $\mathbf{y}_m(\sigma) = \mathbf{N}_m(\sigma)^{-1}\left[\left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma\mathbf{Z}_m\right)\right]^*\mathbf{r}_0(\sigma)$. This projection process involves only the precomputed matrices ($\overline{\mathbf{H}}_m$, $\mathbf{V}_{m+1}$, and $\mathbf{Z}_{m+1}$). The matrices $\overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m$, $\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{Z}_m$, and $\mathbf{Z}_m^*\mathbf{Z}_m$ can be computed once, independent of the number of shifted systems. The solution of a dense Hermitian linear system with $\mathbf{N}_m(\sigma)$ must be performed for each σ. This solution of a Hermitian $m \times m$ linear system costs $\mathcal{O}(m^3)$ floating point operations (FLOPS). The right-preconditioned shifted GMRES algorithm (sGMRES) is shown in Algorithm 4.1. Observe that an implementation can rely heavily on an existing GMRES code. It should be noted that all but one step of the shifted residual projections can be formulated in terms of block/BLAS-3 operations so that most computations for all shifts are performed simultaneously.

Algorithm 4.1: Right preconditioned shifted GMRES (`sGMRES()`).

```
Input : $\mathbf{A} \in \mathbb{C}^{n\times n}$; $\mathbf{b} \in \mathbb{C}^n$; $\{\sigma_\ell\}_{\ell=1}^L \subset \mathbb{C}$; initial Approximations $\{\mathbf{x}(\sigma_\ell)\}_{\ell=1}^L$;
        $\varepsilon > 0$; cycle length $m \in \mathbb{N}$
Output: $\{\mathbf{x}(\sigma_\ell)\}_{\ell=1}^L$ such that $\|\mathbf{r}(\sigma_\ell)\| / \|\mathbf{r}_0(\sigma_\ell)\| \leq \varepsilon$ for all $\ell$
for $\ell = 1 \ldots L$ do
    $\mathbf{r}(\sigma_\ell) = \mathbf{b} - (\mathbf{A} + \sigma_\ell\mathbf{I})\mathbf{x}(\sigma_\ell)$
$\gamma_1 = \|\mathbf{r}(\sigma_1)\|$
if $L > 1$ then
    while $\|\mathbf{r}(\sigma_1)\| / \gamma_1 > \varepsilon$ do
        Compute and overwrite $\mathbf{x}(\sigma_1)$, $\mathbf{r}(\sigma_1)$, $\mathbf{V}_{m+1}$, $\mathbf{Z}_m$, $\overline{\mathbf{H}}_m$ by calling GMRES() for
        $\mathbf{A} + \sigma_1\mathbf{I}$, $\mathbf{M}$, $\mathbf{b}$, $\mathbf{x}(\sigma_1)$, and $m$
        Compute and overwrite $\overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m$, $\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{Z}_m$, and $\mathbf{Z}_m^*\mathbf{Z}_m$
        for $\ell = 2 \ldots L$ do
            $\mathbf{N} \leftarrow \overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m + \sigma_\ell\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{Z}_m + \overline{\sigma_\ell}\mathbf{Z}^*\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + |\sigma_\ell|^2\mathbf{Z}_m\mathbf{Z}_m$
            $\mathbf{y} \leftarrow \mathbf{N}^{-1}\left[\left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma_\ell\mathbf{Z}_m\right)\right]^*\mathbf{r}_0(\sigma_\ell)$
            $\mathbf{x}(\sigma_\ell) \leftarrow \mathbf{x}_0(\sigma_\ell) + \mathbf{Z}_m\mathbf{y}$
            $\mathbf{r}(\sigma_\ell) \leftarrow \mathbf{r}_0(\sigma_\ell) - \left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma_\ell\mathbf{Z}_m\right)\mathbf{y}$
    For all $\ell = 2, \ldots L$ compute and overwrite $\mathbf{x}(\sigma_\ell)$ by recursively calling sGMRES()
    for $\mathbf{A}$, $\mathbf{b}$, $\mathbf{M}$, $\{\sigma_\ell\}_{\ell=2}^L$, $\{\mathbf{x}(\sigma_\ell)\}_{\ell=2}^L$, $\varepsilon$, and $m$
else
    while $\|\mathbf{r}(\sigma_1)\| / \gamma_1 > \varepsilon$ do
        Compute and overwrite $\mathbf{x}(\sigma_1)$, $\mathbf{r}(\sigma_1)$ by calling GMRES() for $\mathbf{A} + \sigma_1\mathbf{I}$, $\mathbf{M}$, $\mathbf{b}$,
        $\mathbf{x}(\sigma_1)$, and $m$
```

4.2. An rGMRES method for shifted systems. Suppose now that our iteration for the base system is a Recycled GMRES method.

We begin by projecting the initial residual $\mathbf{r}_{-1}(\sigma)$ associated to the initial approximation $\mathbf{x}_{-1}(\sigma)$ so that we begin with $\mathbf{r}_0(\sigma) \perp \mathbf{A}(\sigma)\mathcal{U}$. This is equivalent to computing the minimum residual correction $\mathbf{t}_0(\sigma) \in \mathcal{U}$ and setting $\mathbf{x}_0(\sigma) = \mathbf{x}_{-1}(\sigma) + \mathbf{t}_0(\sigma)$. In Recycled GMRES, such a projection is necessary to correctly derive the algorithm. For the shifted system, the projection is not necessary, but it does allow for some simplifications later in the derivation. We have then

$$\mathbf{x}_0(\sigma) = \mathbf{x}_{-1}(\sigma) + \mathbf{U}\mathbf{y}_0(\sigma) \quad \text{and} \quad \mathbf{r}_0(\sigma) = \mathbf{r}_{-1}(\sigma) - \mathbf{A}(\sigma)\mathbf{U}\mathbf{y}_0(\sigma), \tag{4.8}$$

where $\mathbf{y}_0(\sigma) = \mathbf{N}_0(\sigma)^{-1}\left(\mathbf{A}(\sigma)\mathbf{U}\right)^*\mathbf{r}_{-1}$ and $\mathbf{N}_0(\sigma) = \left(\mathbf{A}(\sigma)\mathbf{U}\right)^*\left(\mathbf{A}(\sigma)\mathbf{U}\right)$. Since we have $\mathbf{A}(\sigma)\mathbf{U} = \mathbf{C} + \sigma\mathbf{U}$, this projection can be simplified and computed with manageable additional expense,

$$\mathbf{r}_0 = \mathbf{r}_{-1} - (\mathbf{C} + \sigma\mathbf{U})\mathbf{N}_0(\sigma)^{-1}(\mathbf{C} + \sigma\mathbf{U})^*\mathbf{r}_{-1},$$

where we rewrite $\mathbf{N}_0(\sigma) = \mathbf{I}_{k\times k} + \sigma\mathbf{C}^*\mathbf{U} + \overline{\sigma}\mathbf{U}^*\mathbf{C} + |\sigma|^2\,\mathbf{U}^*\mathbf{U}$. The matrices $\mathbf{C}^*\mathbf{U}$ and $\mathbf{U}^*\mathbf{U}$ must only be computed once regardless of the number of shifts, and for each shift we solve $\mathbf{N}_0(\sigma)\mathbf{y}_0(\sigma) = (\mathbf{C} + \sigma\mathbf{U})^*\mathbf{r}_{-1}(\sigma)$.

After a cycle of Recycled GMRES for the base system, (4.3) must be enforced for each shifted system. At iteration m, our search space is $\mathcal{S}_m \ = \mathcal{U} + \mathcal{K}_m(\mathbf{P}\mathbf{A}_p, \mathbf{r}_0)$. The augmented matrix $\mathbf{S}_m := \begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}$ contains as columns the basis for $\mathcal{U}$ and $\mathcal{K}_m(\mathbf{P}\mathbf{A}_p, \mathbf{r}_0)$. In this case, we have $\mathbf{N}_m(\sigma) = \left\{(\mathbf{A} + \sigma\mathbf{I})\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\right\}^*\left\{(\mathbf{A} + \sigma\mathbf{I})\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\right\}$. From [41], we have the identity

$$(\mathbf{A} + \sigma\mathbf{I})\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix} = \begin{bmatrix}(\mathbf{C} + \sigma\mathbf{U}) & \left(\mathbf{C}\mathbf{B}_m + \mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma\mathbf{V}_m\right)\end{bmatrix}.$$

Thus, in the unpreconditioned case, for the augmented Krylov subspace, we can rewrite (4.3) as

$$\begin{aligned}
&\mathbf{r}_m(\sigma) = \mathbf{r}_0(\sigma) - \begin{bmatrix}(\mathbf{C} + \sigma\mathbf{U}) & \left(\mathbf{C}\mathbf{B}_m + \mathbf{V}_{m+1}\overline{\mathbf{H}}_m + \sigma\mathbf{V}_m\right)\end{bmatrix}\mathbf{y}_m(\sigma) \quad \text{and}\\
&\mathbf{x}_m(\sigma) = \mathbf{x}_0(\sigma) + \begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\mathbf{y}_m(\sigma),
\end{aligned}$$

where $\mathbf{y}_m(\sigma) = \mathbf{N}_m(\sigma)^{-1}\left[(\mathbf{C} + \sigma\mathbf{U}) \quad \left(\mathbf{C}\mathbf{B}_m + \mathbf{V}_{m+1}\overline{\mathbf{H}}_m^{(\sigma)}\right)\right]^*\mathbf{r}_0(\sigma)$ and

$$\mathbf{N}_m(\sigma) = \begin{bmatrix}\mathbf{I} + \sigma\mathbf{C}^*\mathbf{U} + \overline{\sigma}\mathbf{U}^*\mathbf{C} + |\sigma|^2\,\mathbf{U}^*\mathbf{U} & \mathbf{B}_m + \overline{\sigma}\mathbf{U}^*\mathbf{C}\mathbf{B}_m + \overline{\sigma}\mathbf{U}^*\mathbf{V}_{m+1}\overline{\mathbf{H}}_m + |\sigma|^2\,\mathbf{U}^*\mathbf{V}_m\\ \mathbf{B}_m^* + \sigma\mathbf{B}_m^*\mathbf{C}^*\mathbf{U} + \sigma\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{U} + |\sigma|^2\,\mathbf{V}_m^*\mathbf{U} & \mathbf{B}_m^*\mathbf{B}_m + \overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m + \sigma\mathbf{H}_m + \overline{\sigma}\mathbf{H}_m^* + |\sigma|^2\,\mathbf{I}\end{bmatrix}.$$

This projection can be performed using already computed quantities, and the matrices $\mathbf{U}^*\mathbf{C}$, $\mathbf{U}^*\mathbf{U}$, $\mathbf{U}^*\mathbf{C}\mathbf{B}_m$, $\mathbf{U}^*\mathbf{V}_{m+1}\overline{\mathbf{H}}_m$, $\overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m$, $\mathbf{H}_m$, and $\mathbf{B}_m^*\mathbf{B}_m$ need only be computed once regardless of the number of shifts. The computation of $\mathbf{y}_m(\sigma)$ must be performed for every shift at a cost of $\mathcal{O}((m+k)^3)$.

4.2.1. Preconditioning. Introducing right preconditioning creates some difficulties which we can again surmount by storing some extra vectors. We note that in the case of preconditioning, we have $\mathbf{C} = \mathbf{A}\mathbf{M}^{-1}\mathbf{U}$. In this case, for right-preconditioned Recycled GMRES, the search space for the base system is $\mathcal{S}_m := \mathbf{M}^{-1}\left\{\mathcal{U} + \mathcal{K}_m(\mathbf{P}\mathbf{A}, \mathbf{r}_0)\right\}$. Let $\mathbf{Z}_{\mathcal{U}} = \mathbf{M}^{-1}\mathbf{U}$ and $\mathbf{Z}_m = \mathbf{M}^{-1}\mathbf{V}_m$ as in Section 4.1.

Using $\mathbf{Z}_{\mathcal{U}}$, we can cheaply perform the initial residual projection,

$$\begin{aligned}
&\mathbf{x}_0(\sigma) = \mathbf{x}_{-1}(\sigma) + \mathbf{U}\mathbf{y}_0(\sigma) \quad \text{and}\\
&\mathbf{r}_0(\sigma) = \mathbf{r}_{-1}(\sigma) - \left(\mathbf{A}_p(\sigma)\mathbf{U}\right)\mathbf{y}_0(\sigma),
\end{aligned} \tag{4.9}$$

where $\mathbf{y}_0(\sigma) = \mathbf{N}_0(\sigma)^{-1}\left(\mathbf{A}_p(\sigma)\mathbf{U}\right)^*\mathbf{r}_{-1}(\sigma)$ and $\mathbf{N}_0(\sigma) = \left(\mathbf{A}_p(\sigma)\mathbf{U}\right)^*\left(\mathbf{A}_p(\sigma)\mathbf{U}\right)$. We can write

$$\mathbf{A}_p(\sigma)\mathbf{U} = \mathbf{C} + \sigma\mathbf{Z}_{\mathcal{U}}.$$

The subspace $\mathcal{U}$ either is available at the start of the algorithm (in which case $\mathbf{U}$ must be scaled so that $\mathbf{A}_p\mathbf{U} = \mathbf{C}$ has orthonormal columns), or it is constructed at the end of a restart cycle. In either case, $\mathbf{Z}_{\mathcal{U}}$ is available in the course of the computation and can be saved. Thus the projection (4.8) can be performed with already computed quantities,

$$\begin{aligned}\mathbf{x}_0(\sigma) &= \mathbf{x}_{-1}(\sigma) + \mathbf{U}\mathbf{y}_0(\sigma) \text{ and}\\ \mathbf{r}_0(\sigma) &= \mathbf{r}_{-1}(\sigma) - \left(\mathbf{C} + \sigma\mathbf{Z}_{\mathcal{U}}\right)\mathbf{y}_0(\sigma),\end{aligned}$$

where we rewrite $\mathbf{y}_0(\sigma) = \mathbf{N}_0(\sigma)^{-1}\left(\mathbf{C} + \sigma\mathbf{Z}_{\mathcal{U}}\right)^* \mathbf{r}_{-1}(\sigma_\ell)$ and

$$\mathbf{N}_0(\sigma) = \mathbf{I} + \sigma\mathbf{C}^*\mathbf{Z}_{\mathcal{U}} + \overline{\sigma}\mathbf{Z}_{\mathcal{U}}^*\mathbf{C} + |\sigma|^2\,\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}.$$

After a cycle of right-preconditioned Recycled GMRES, we must perform the projection (4.4) for each shifted system. We proceed slightly differently in this derivation than in the unpreconditioned case. We have

$$\mathbf{N}_m(\sigma) := \left\{\mathbf{A}_p(\sigma)\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\right\}^* \left\{\mathbf{A}_p(\sigma)\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix}\right\}.$$

Following [27], we define

$$\overline{\mathbf{G}}_m = \begin{bmatrix}\mathbf{I}_{k\times k} & \mathbf{B}_m\\ \mathbf{0}_{(m+1)\times k} & \overline{\mathbf{H}}_m\end{bmatrix},$$

which yields the augmented Arnoldi relation

$$\mathbf{A}_p\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix} = \begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m. \tag{4.10}$$

Using the relation (4.10), an identity for the shifted operator with right preconditioning follows,

$$\mathbf{A}_p(\sigma)\begin{bmatrix}\mathbf{U} & \mathbf{V}_m\end{bmatrix} = \begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m + \sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}. \tag{4.11}$$

We use the relation (4.11) to derive the expansion

$$\begin{aligned}\mathbf{N}_m(\sigma) = \overline{\mathbf{G}}_m^*\overline{\mathbf{G}}_m + |\sigma|^2\begin{bmatrix}\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_m\\ \mathbf{Z}_m^*\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m^*\mathbf{Z}_m\end{bmatrix} + \sigma\overline{\mathbf{G}}_m^*\begin{bmatrix}\mathbf{C}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{C}^*\mathbf{Z}_m\\ \mathbf{V}_{m+1}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{V}_{m+1}^*\mathbf{Z}_m\end{bmatrix}\\ + \overline{\sigma}\begin{bmatrix}\mathbf{Z}_{\mathcal{U}}^*\mathbf{C} & \mathbf{Z}_{\mathcal{U}}^*\mathbf{V}_{m+1}\\ \mathbf{Z}_m^*\mathbf{C} & \mathbf{Z}_m^*\mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m.\end{aligned} \tag{4.12}$$

Thus, the projection can be performed for each shift using already computed quantities. This yields the following updates of the approximation and residual

$$\begin{aligned}\mathbf{x}_m(\sigma) &= \mathbf{x}_0(\sigma) + \begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\mathbf{y}_m(\sigma)\\ \mathbf{r}_m(\sigma) &= \mathbf{r}_0(\sigma) - \left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m + \sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}\mathbf{y}_m(\sigma),\end{aligned}$$

where $\mathbf{y}_m(\sigma) = \mathbf{N}_m(\sigma)^{-1}\left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m + \sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}^*\mathbf{r}_0(\sigma)$. We observe that because of the initial projection of the shifted residual (4.9), we can simplify

$$\begin{aligned}&\left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m + \sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}^*\mathbf{r}_0(\sigma)\\ &\quad= \left\{\begin{bmatrix}\mathbf{C} & \mathbf{C}\mathbf{B}_m + \mathbf{V}_{m+1}\overline{\mathbf{H}}_m\end{bmatrix} + \sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}^*\mathbf{r}_0(\sigma)\\ &\quad= \begin{bmatrix}\mathbf{C}^*\mathbf{r}_0(\sigma)\\ \mathbf{B}_m^*\mathbf{C}^*\mathbf{r}_0(\sigma) + \overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{r}_0(\sigma)\end{bmatrix} + \overline{\sigma}\begin{bmatrix}\mathbf{Z}_{\mathcal{U}}^*\mathbf{r}_0(\sigma)\\ \mathbf{Z}_m^*\mathbf{r}_0(\sigma)\end{bmatrix}\\ &\quad= \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_m^*\mathbf{C}^*\mathbf{r}_0(\sigma) + \overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{r}_0(\sigma) + \overline{\sigma}\mathbf{Z}_m^*\mathbf{r}_0(\sigma)\end{bmatrix},\end{aligned}$$

and thus we can rewrite

$$\mathbf{y}_m(\sigma) = \mathbf{N}_m(\sigma)^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{B}_m^*\mathbf{C}^*\mathbf{r}_0(\sigma)+\overline{\mathbf{H}}_m^*\mathbf{V}_{m+1}^*\mathbf{r}_0(\sigma)+\overline{\sigma}\mathbf{Z}_m^*\mathbf{r}_0(\sigma)\end{bmatrix}.$$

Algorithm 4.2: Right-preconditioned shifted Recycled GMRES (`srGMRES()`).

```
Input  : $\mathbf{A}\in\mathbb{C}^{n\times n}$; $\mathbf{b}\in\mathbb{C}^n$; $\{\sigma_\ell\}_{\ell=1}^L\subset\mathbb{C}$; initial Approximations $\{\mathbf{x}(\sigma_\ell)\}_{\ell=1}^L$;
         $\mathbf{U}\in\mathbb{C}^{n\times k}$; $\varepsilon>0$; cycle length $m\in\mathbb{N}$
Output : $\{\mathbf{x}(\sigma_\ell)\}_{\ell=1}^L$ such that $\|\mathbf{r}(\sigma_\ell)\| / \|\mathbf{r}_0(\sigma_\ell)\|\le\varepsilon$ for all $\ell$
for $\ell=1\ldots L$ do
    $\mathbf{r}(\sigma_\ell)=\mathbf{b}-(\mathbf{A}+\sigma_\ell\mathbf{I})\mathbf{x}(\sigma_\ell)$
$\gamma_1=\|\mathbf{r}(\sigma_1)\|$
$\mathbf{Z}_{\mathcal{U}}=\mathbf{M}^{-1}\mathbf{U}$
$\mathbf{C}=(\mathbf{A}+\sigma_1\mathbf{I})\mathbf{Z}_{\mathcal{U}}$
Compute QR-factorization $\mathbf{QR}=\mathbf{C}$
$\mathbf{C}\leftarrow\mathbf{Q}$, $\mathbf{U}\leftarrow\mathbf{UR}^{-1}$, $\mathbf{Z}_{\mathcal{U}}\leftarrow\mathbf{Z}_{\mathcal{U}}\mathbf{R}^{-1}$
$\mathbf{x}(\sigma_1)\leftarrow\mathbf{x}(\sigma_1)+\mathbf{UC}^*\mathbf{r}(\sigma_1)$ and $\mathbf{r}(\sigma_1)\leftarrow\mathbf{r}(\sigma_1)-\mathbf{CC}^*\mathbf{r}(\sigma_1)$
Compute $\mathbf{C}^*\mathbf{Z}_{\mathcal{U}}$ and $\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}$
for $\ell=2\ldots L$ do
    %%%%% Shifted System Initial Projections %%%%%
    $\mathbf{N}\leftarrow\mathbf{I}+\sigma_\ell\mathbf{C}^*\mathbf{Z}_{\mathcal{U}}+\overline{\sigma_\ell}\mathbf{Z}_{\mathcal{U}}^*\mathbf{C}+|\sigma_\ell|^2\,\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}$
    $\mathbf{y}\leftarrow\mathbf{N}^{-1}\left(\mathbf{C}+\sigma_\ell\mathbf{Z}_{\mathcal{U}}\right)^*\mathbf{r}(\sigma_\ell)$
    $\mathbf{x}(\sigma_\ell)\leftarrow\mathbf{x}(\sigma_\ell)+\mathbf{Uy}$
    $\mathbf{r}(\sigma_\ell)\leftarrow\mathbf{r}(\sigma_\ell)-\left(\mathbf{C}+\sigma_\ell\mathbf{Z}_{\mathcal{U}}\right)\mathbf{y}$
if $L>1$ then
    while $\|\mathbf{r}(\sigma_1)\|/\gamma_1>\varepsilon$ do
        Compute and overwrite $\mathbf{x}(\sigma_1)$, $\mathbf{r}(\sigma_1)$, $\mathbf{V}_{m+1}$, $\mathbf{Z}_m$, $\overline{\mathbf{H}}_m$, $\mathbf{B}_m$ by calling
        rGMRES() for $\mathbf{A}+\sigma_1\mathbf{I}$, $\mathbf{M}$, $\mathbf{b}$, $\mathbf{x}(\sigma_1)$, $\mathbf{U}$, $\mathbf{C}$, and $m$
        Compute and overwrite $\overline{\mathbf{G}}_m$, $\overline{\mathbf{G}}_m^*\overline{\mathbf{G}}_m$, $\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}$, $\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_m$, $\mathbf{Z}_m^*\mathbf{Z}_m$, $\mathbf{C}^*\mathbf{Z}_{\mathcal{U}}$,
        $\mathbf{C}^*\mathbf{Z}_m$, $\mathbf{V}_{m+1}^*\mathbf{Z}_{\mathcal{U}}$, $\mathbf{V}_{m+1}^*\mathbf{Z}_m$
        for $\ell=2\ldots L$ do
            %%%%% Shifted System Projections %%%%%
            $\mathbf{N}\leftarrow\overline{\mathbf{G}}_m^*\overline{\mathbf{G}}_m+|\sigma_\ell|^2\begin{bmatrix}\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_m\\ \mathbf{Z}_m^*\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m^*\mathbf{Z}_m\end{bmatrix}+$
            $\sigma_\ell\overline{\mathbf{G}}_m^*\begin{bmatrix}\mathbf{C}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{C}^*\mathbf{Z}_m\\ \mathbf{V}_{m+1}^*\mathbf{Z}_{\mathcal{U}} & \mathbf{V}_{m+1}^*\mathbf{Z}_m\end{bmatrix}+\overline{\sigma_\ell}\begin{bmatrix}\mathbf{Z}_{\mathcal{U}}^*\mathbf{C} & \mathbf{Z}_{\mathcal{U}}^*\mathbf{V}_{m+1}\\ \mathbf{Z}_m^*\mathbf{C} & \mathbf{Z}_m^*\mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m$
            $\mathbf{y}\leftarrow\mathbf{N}^{-1}\left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m+\sigma_\ell\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}^*\mathbf{r}(\sigma_\ell)$
            $\mathbf{x}(\sigma_\ell)\leftarrow\mathbf{x}_0(\sigma_\ell)+\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\mathbf{y}$
            $\mathbf{r}(\sigma_\ell)\leftarrow\mathbf{r}_0(\sigma_\ell)-\left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m+\sigma_\ell\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}\mathbf{y}$
        Compute updated $\mathbf{U}$, $\mathbf{Z}_{\mathcal{U}}$, and $\mathbf{C}$
    For all $\ell=2,\ldots L$ compute and overwrite $\mathbf{x}(\sigma_\ell)$ by recursively calling
    srGMRES() for $\mathbf{A}$, $\mathbf{b}$, $\mathbf{M}$, $\{\sigma_\ell\}_{\ell=2}^L$, $\{\mathbf{x}(\sigma_\ell)\}_{\ell=2}^L$, $\mathbf{U}$, $\varepsilon$, and $m$
else
    while $\|\mathbf{r}(\sigma_1)\|/\gamma_1>\varepsilon$ do
        Compute and overwrite $\mathbf{x}(\sigma_1)$, $\mathbf{r}(\sigma_1)$ by calling rGMRES() for $\mathbf{A}+\sigma_1\mathbf{I}$, $\mathbf{M}$, $\mathbf{b}$,
        $\mathbf{x}(\sigma_1)$, and $m$
```

The matrices in the sum (4.12) must be computed only once. For each shift, we must compute $\mathbf{y}_m(\sigma)$ at a cost of $\mathcal{O}\big((m+k)^3\big)$. The right-preconditioned shifted Recycled GMRES algorithm (srGMRES) is shown in Algorithm 4.2. Observe that an implementation can rely heavily on an existing Recycled GMRES code. As in the case of Algorithm 4.1, all but one step of the shifted residual projections can be formulated in terms of block/BLAS-3 operations so that almost all computations are performed simultaneously for all shifts. We discuss costs further in Section 5.2.

5. Analysis of direct projection methods.

In this section, we provide some analysis of the direction projection methods. We treat two issues in this section: the quality of the approximations and the costs of the methods.

5.1. The quality of the approximations.

Since all residual corrections are minimum residual projections, we can expect that, at worse, the projection of the shifted residual will achieve no improvement.

We follow the analysis presented in [19]. This analysis proceed from that presented in [7] for the case of a Hermitian positive definite coefficient matrix. In their analysis, the authors assume that a subset of eigenvectors (spanning the subspace $\mathcal{Y}$) have been well-approximated in the underlying Krylov subspace generated by QMR applied to the base matrix (called the seed system in [19]). The authors show that the performance of QMR applied to the non-seed systems with projected residuals can be compared to that of a GMRES iteration in which $\mathcal{Y}$ has been projected away.

In the case of Hermitian positive definite systems, the analysis of the performance of CG-based seed-projection was also extended to the case in which the coefficient matrix varies along with the right-hand side [6]. In this paper, the authors extend the work in [7] by applying a Lanczos-Galerkin-type projection in the case that the system matrices as well as the right-hand sides are changing in some structured way (which we note is also a common assumption motivating subspace recycling methods). One of the special cases considered is the present one, namely that of solving a family of shifted linear systems. In this case, one can again derive CG-based bounds dependent upon the set of eigenvectors well-approximated by Ritz vectors generated by the CG iteration applied to the base system.

In extending this analysis, there are two complications. Algorithm 4.2 does not minimize over a Krylov subspace, and both methods may use preconditioning. In either case, we cannot easily leverage the polynomial approximation analysis. Also of concern is that GMRES-based methods applied to non-Hermitian problems of large dimension often must be restarted, which does not need to be considered for the short-term recurrence-based methods treated in [6, 7, 19]. However, if we restrict our analysis to Algorithm 4.1 without preconditioning (i.e., $\mathbf{M}=\mathbf{I}$) and do not consider restarting, we can analyze performance based on invariant subspace approximation. We follow [6, 7] and specifically use elements of analysis in [19] for the non-Hermitian case.

Let us assume that $\mathbf{A}$ is diagonalizable with eigendecomposition

$$\mathbf{A}=\mathbf{F}\boldsymbol{\Lambda}\mathbf{W} \quad \text{with} \quad \boldsymbol{\Lambda}=\operatorname{diag}\left\{\lambda_1,\lambda_2\ldots,\lambda_n\right\} \quad \text{and} \quad \mathbf{W}=\mathbf{F}^{-1}, \tag{5.1}$$

with $\mathbf{f}_i$ being the ith column of $\mathbf{F}$ and $\mathbf{w}_i^*$ being the ith row of $\mathbf{W}$. Consider the simplified problem (4.1), where for the base system (3.1), we have the initial residual $\mathbf{r}_0$. We first solve the base system using a GMRES iteration terminating in j steps generating the subspace $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ with the associated $\mathbf{V}_j$ and $\overline{\mathbf{H}}_j$. Let $\mathbf{x}_0(\sigma)$ be the initial approximation for the shifted system with the residual $\mathbf{r}_0(\sigma)=\mathbf{b}-(\mathbf{A}+\sigma\mathbf{I})\,\mathbf{x}_0(\sigma)$. Let $\hat{\mathbf{r}}_0(\sigma)\perp(\mathbf{A}+\sigma\mathbf{I})\,\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ be the result of the Lanczos-Galerkin projection of $\mathbf{r}_0(\sigma)$ after the termination of GMRES applied to (3.1). If $\mathbf{P}$ is the projector onto $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ which is orthogonal with respect to the inner product induced by[2] $(\mathbf{A}+\sigma\mathbf{I})^*(\mathbf{A}+\sigma\mathbf{I})$ and $\mathbf{Q}$ is the orthogonal projector onto $(\mathbf{A}+\sigma\mathbf{I})\,\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ with respect to the Euclidean norm[3], then we can then write the Lanczos-Galerkin projection as $\hat{\mathbf{r}}_0(\sigma) = (\mathbf{I}-\mathbf{Q})\,\mathbf{r}_0(\sigma)$, and the associated updated approximation $\hat{\mathbf{x}}_0(\sigma)$ results from the error projection

$$\mathbf{x}(\sigma) - \hat{\mathbf{x}}_0(\sigma) = (\mathbf{I}-\mathbf{P})\,(\mathbf{x}(\sigma)-\mathbf{x}_0(\sigma))\,. \tag{5.2}$$

This can be seen by studying the derivation in Section 4.1 and is a general property of minimum residual projections. With this new starting vector $\hat{\mathbf{x}}_0(\sigma)$, we now consider the performance of GMRES applied to the shifted system.

THEOREM 5.1. *Let $\mathbf{A}\in\mathbb{C}^{n\times n}$ be diagonalizable with eigendecomposition (5.1). Let $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ be the Krylov subspace generated by j iterations of unrestarted GMRES applied to (3.1), and for an indexing set $\mathbb{I}\subsetneq\{1,\ldots,n\}$, let $\mathcal{Y}$ be an invariant subspace of $\mathbf{A}$ spanned by $\{\mathbf{f}_i\}_{i\in\mathbb{I}}$. Let $\mathbf{P}_{\mathcal{Y}}$ be the orthogonal projection onto $\mathcal{Y}$, and let $\overline{\mathbf{x}}_0(\sigma)$ be the result of the error projection*

$$\mathbf{x}(\sigma) - \overline{\mathbf{x}}_0(\sigma) = (\mathbf{I}-\mathbf{P}_{\mathcal{Y}})\,(\mathbf{x}(\sigma)-\hat{\mathbf{x}}_0(\sigma))\,.$$

If we apply Algorithm 4.1 to solve (4.1) with no preconditioning and no restarting, then the residual $\hat{\mathbf{r}}_\ell(\sigma)$ resulting from ℓ iterations of GMRES applied to the shifted system with starting vector $\hat{\mathbf{x}}_\ell(\sigma)$ defined as in (5.2) satisfies the bound

$$\|\hat{\mathbf{r}}_\ell\| \le \|\overline{\mathbf{r}}_\ell(\sigma)\| + \delta,$$

where $\overline{\mathbf{r}}_\ell(\sigma)$ is the residual resulting from applying ℓ iterations of GMRES to the shifted system with starting vector $\overline{\mathbf{x}}_0(\sigma)$ and $\delta = \sum_{i\in\mathbb{I}}(\lambda_i+\sigma)\overline{p}_\ell(\lambda_i+\sigma)\phi_i\mathbf{f}_j$ with $\overline{p}_j$ being the jth GMRES residual polynomial associated to the iteration for starting vector $\overline{\mathbf{x}}_0(\sigma)$ and $\phi_i = \mathbf{w}_i^*\,(\mathbf{I}-\mathbf{P})\,(\mathbf{x}(\sigma)-\mathbf{x}_0(\sigma))$.

Proof. The structure of this proof follows that in [19], but it is also related to the results presented in [6, Section 3.1, Case 1] of CG with Lanczos-Galerkin projection applied to shifted systems.

Because $\mathbf{A}$ is diagonalizable, we can decompose the errors with respect to $\hat{\mathbf{x}}(\sigma)$ and $\overline{\mathbf{x}}(\sigma)$ as

$$\mathbf{x}(\sigma)-\hat{\mathbf{x}}_0(\sigma) = \sum_{i=1}^{n}\phi_i\mathbf{f}_i \quad\text{and}\quad \mathbf{x}(\sigma)-\overline{\mathbf{x}}_0(\sigma) = \sum_{\substack{i=1\\ i\notin\mathbb{I}}}^{n}\phi_i\mathbf{f}_i,$$

which implies that

$$\hat{\mathbf{r}}_0(\sigma) = \sum_{i=1}^{n}\phi_i\,(\lambda_i+\sigma)\,\mathbf{f}_i \quad\text{and}\quad \overline{\mathbf{r}}_0(\sigma) = \sum_{\substack{i=1\\ i\notin\mathbb{I}}}^{n}\phi_i\,(\lambda_i+\sigma)\,\mathbf{f}_i.$$

Because the GMRES residual polynomial $\hat{p}_\ell$ satisfies the minimization condition

$$\hat{p}_\ell = \underset{\substack{p\in\Pi_\ell\\ p(0)=1}}{\text{argmin}}\,\|p(A+\sigma\mathbf{I})\hat{r}_0\| \qquad\text{with}\qquad \Pi_\ell = \{p\,|\,\deg p\le\ell\}\,,$$

[2] i.e., $\mathbf{P} = \mathbf{V}_m\left(\overline{\mathbf{H}}(\sigma)^*\overline{\mathbf{H}}(\sigma)\right)^{-1}\overline{\mathbf{H}}_m(\sigma)^*\mathbf{V}_{m+1}^*$

[3] i.e., $\mathbf{Q} = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m(\sigma)\left(\overline{\mathbf{H}}(\sigma)^*\overline{\mathbf{H}}(\sigma)\right)^{-1}\overline{\mathbf{H}}_m(\sigma)^*\mathbf{V}_{m+1}^*$

we can write

$$\begin{aligned}\|\hat{\mathbf{r}}_\ell\| = \min_{\substack{p\in\Pi_\ell\\ p(0)=1}} \|p(A+\sigma\mathbf{I})\hat{r}_0\| \leq \|\overline{p}_\ell\left(\mathbf{A}+\sigma\right)\hat{\mathbf{r}}_0\| &= \left\|\sum_{i=1}^{n}(\lambda_i+\sigma)\overline{p}_\ell(\lambda_i+\sigma)\phi_i\mathbf{f}_i\right\| \\ &= \left\|\sum_{\substack{i=1\\ i\not\in\mathbb{I}}}^{n}(\lambda_i+\sigma)\overline{p}_\ell(\lambda_i+\sigma)\mathbf{f}_i + \sum_{i\in\mathbb{I}}(\lambda_i+\sigma)\overline{p}_\ell(\lambda_i+\sigma)\phi_i\mathbf{f}_i\right\| \\ &\leq \|\overline{\mathbf{r}}_\ell(\sigma)\| + \underbrace{\left\|\sum_{i\in\mathbb{I}}(\lambda_i+\sigma)\overline{p}_\ell(\lambda_i+\sigma)\phi_i\mathbf{f}_i\right\|}_{\delta}.\end{aligned}$$

From the definitions of $\mathbf{f}_{i_1}$ and $\mathbf{w}_{i_2}$, we know that $\mathbf{w}_{i_2}^*\mathbf{f}_{i_1} = \delta_{i_1,i_2}$. Thus, from the definition of $\mathbf{x}(\sigma) - \hat{\mathbf{x}}_0(\sigma)$ as well as its eigendecomposition, we have that $\phi_i = \mathbf{w}_i^*\left(\mathbf{I}-\mathbf{P}\right)\left(\mathbf{x}(\sigma)-\mathbf{x}_0(\sigma)\right)$. □

Certainly, Theorem 5.1 applies to any invariant subspace $\mathcal{Y}$. However, the interesting case, which is considered in [7, 6, 19], is when $\mathcal{Y}$ is such that the Krylov subspace $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ contains a good approximation of it. If $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$ actually contained $\mathcal{Y}$, then it is straightforward to show that $\phi_i = 0$ for all $i \in \mathbb{I}$, and thus $\delta = 0$. We can then expect that if $\mathcal{Y}$ is well-approximated in $\mathcal{K}_j(\mathbf{A},\mathbf{r}_0)$, then δ would be non-zero but small. In this case, the behavior of GMRES applied to the shifted system with starting vector $\hat{\mathbf{x}}_0$ would mimic GMRES applied to that same system with starting vector $\overline{\mathbf{x}}_0$, in which the iteration is orthogonal to $\mathcal{Y}$. Unfortunately, this theory cannot be easily extended to the case that the correction space is not a Krylov subspace as it relies on the polynomial approximation interpretation of GMRES.

Following [29], we also can analyze the effectiveness of the direct projection by decomposing the residual. This analysis is developed in the general framework setting presented in Section 4 and then interpreted for the individual methods. Here we use the notation that $\mathbf{P}(\cdot)$ denotes the orthogonal projector onto the subspace specified in the argument.

THEOREM 5.2. *Let the sequence of subspaces $\{\mathcal{S}_m\}$ be defined as in Section 4, and additionally let*

$$\mathcal{T}_m = \left(\mathbf{A}+\sigma\mathbf{I}\right)\mathbf{M}^{-1}\mathcal{S}_m. \tag{5.3}$$

If $\mathbf{r}_0(\sigma)$ is the initial residual for the shifted system and $\hat{\mathbf{r}}_0(\sigma)$ is the residual produced by projecting $\mathbf{r}_0(\sigma)$ according to (4.4), then we have that

$$\hat{\mathbf{r}}_0(\sigma) = \left(\mathbf{I}-\mathbf{P}\left(\mathcal{T}_m\right)\right)\mathbf{P}\left(\mathcal{T}_{m+1}\right)\mathbf{r}_0(\sigma) + \left(\mathbf{I}-\mathbf{P}\left(\mathcal{T}_{m+1}\right)\right)\mathbf{r}_0(\sigma). \tag{5.4}$$

Note that in the unpreconditioned case, Theorem 5.2 can be applied by taking $\mathbf{M} = \mathbf{I}$.

Proof. Using the property of projectors, we can decompose

$$\mathbf{r}_0(\sigma) = \mathbf{P}\left(\mathcal{T}_{m+1}\right)\mathbf{r}_0(\sigma) + (\mathbf{I}-\mathbf{P}\left(\mathcal{T}_{m+1}\right))\mathbf{r}_0(\sigma).$$

The minimum residual projection (4.4) can be written as

$$\hat{\mathbf{r}}_0(\sigma) = (\mathbf{I}-\mathbf{P}\left(\mathcal{T}_m\right))\mathbf{P}\left(\mathcal{T}_{m+1}\right)\mathbf{r}_0(\sigma) + (\mathbf{I}-\mathbf{P}\left(\mathcal{T}_m\right))(\mathbf{I}-\mathbf{P}\left(\mathcal{T}_{m+1}\right))\mathbf{r}_0(\sigma).$$

From (5.3) and the definition of $\mathcal{T}_m$, we have that

$$\mathcal{T}_m \subset \mathcal{T}_{m+1},$$

which in turn yields the reverse containment of the orthogonal complements,

$$\mathcal{T}_{m+1}{}^{\perp} \subset \mathcal{T}_m^{\perp}$$

and thus

$$(\mathbf{I} - \mathbf{P}\,(\mathcal{T}_m))(\mathbf{I} - \mathbf{P}\,(\mathcal{T}_{m+1}))\mathbf{r}_0(\sigma) = (\mathbf{I} - \mathbf{P}\,(\mathcal{T}_{m+1}))\mathbf{r}_0(\sigma).$$

This yields the result. □

COROLLARY 5.3. *Let the same assumptions as in* Theorem 5.2 *hold. Then we have the following bound on* $\|\hat{\mathbf{r}}_0(\sigma)\|$*:*

$$\|\hat{\mathbf{r}}_0(\sigma)\| \leq \|(\mathbf{I} - \mathbf{P}\,(\mathcal{T}_m))\,\mathbf{P}\,(\mathcal{T}_{m+1})\,\mathbf{r}_0(\sigma)\| + \|(\mathbf{I} - \mathbf{P}\,(\mathcal{T}_{m+1}))\,\mathbf{r}_0(\sigma)\|\,. \tag{5.5}$$

Proof. We simply take the norm of both sides of (5.4) and apply the triangle inequality. □

From (5.5), we can see that the residual norm bound depends on both the effectiveness of the minimization projection applied to the orthogonal projection of $\mathbf{r}_0(\sigma)$ in $\mathcal{T}_{m+1}$ and the size of the part of the residual which lies in $\mathcal{T}_{m+1}^{\perp}$. As an aside, to connect this analysis back to the two proposed methods, we observe that in the case of the right-preconditioned shifted GMRES algorithm (Algorithm 4.1), we have

$$\begin{aligned}\mathcal{S}_m &= \mathbf{M}^{-1}\mathcal{K}_m(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0) \;\text{ and}\\ \mathcal{T}_m &= \mathbf{A}\mathbf{M}^{-1}\mathcal{K}_m(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0) + \sigma\mathbf{M}^{-1}\mathcal{K}_m(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0).\end{aligned}$$

In the case of the preconditioned rGMRES method for shifted systems (Algorithm 4.2), we have

$$\mathcal{S}_m = \mathbf{M}^{-1}\left\{\mathcal{U} + \mathcal{K}_m((\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)\right\}$$

and

$$\mathcal{T}_m = \mathcal{C} + \mathbf{A}\mathbf{M}^{-1}\mathcal{K}_m((\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0) + \sigma\mathbf{M}^{-1}\left\{\mathcal{U} + \mathcal{K}_m((\mathbf{I} - \mathbf{P})\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)\right\}.$$

As another quick aside, we mention briefly that the matrix $\mathbf{N}_m(\sigma)$ is connected to a generalized eigenvalue approximation problem associated to the computation of the harmonic Ritz values; see, e.g., [25, 26]. This is elaborated upon in the technical report [39].

5.2. The costs of the algorithms.

For Algorithms 4.1 and 4.2, we enumerate the additional per-cycle costs incurred by the proposed algorithms as they are built, respectively, on top of a cycle of GMRES and a cycle of Recycled GMRES.

Let c_{old} denote the cost per iteration of an existing method (here GMRES or Recycled GMRES) and c_{new} the cost per iteration of the modified method (here Algorithm 4.1 or 4.2). Here we do not specify how cost should be measured. It could be by estimating, e.g., FLOPS, amount of data moved, or actual timings of various operations, etc. In our subsequent calculations, though, we estimate costs in FLOPS. In this setting, we have that the new methods cost more per iteration, i.e., $c_{new} = c_{old} + d_{new}$. In Tables 5.1 and 5.2, we list, respectively, the additional costs of each proposed algorithms, allowing us to estimate d_{new}. An important consideration which we don't treat here is the cost of applying the operator, which depends on characteristics such as sparsity. This can dominate the cost per iteration. In judging the effectiveness of these methods, the benefit of iteration reduction is dictated by the matrix-vector product cost (which would also include the cost of applying the preconditioner).

We can similarly define the number of iterations required by both methods to solve all shifted systems, i.e., j_{old} and j_{new}. By assumption, the new method should solve all shifted systems in fewer iterations, i.e., $j_{new} = j_{old} - a_{new}$. Roughly speaking then, the total cost of each method can be estimated by $j_{old} \cdot c_{old}$ and $j_{new} \cdot c_{new}$.

TABLE 5.1
Cost per cycle of extra calculations performed in Algorithm 4.1 when compared to GMRES.

Operations	Alg. Line	FLOPS in $\mathcal{O}(\cdot)$	$\times$ per cycle
$\overline{\mathbf{H}}_m^*\overline{\mathbf{H}}_m$	7	m^3+m^2	1
$\mathbf{V}_{m+1}^*\mathbf{Z}_m$	7	$n\left(m^2+m\right)$	1
$\overline{\mathbf{H}}_m^*\left(\mathbf{V}_{m+1}^*\mathbf{Z}_m\right)$	7	m^3+m^2	1
$\mathbf{Z}_m^*\mathbf{Z}_m$	7	nm^2	1
Sum of 4 $m\times m$ matrices	9	$3m^2$	L
$\left[\left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m+\sigma\mathbf{Z}_m\right)\right]^*\mathbf{r}_0(\sigma)$	10	$2nm$	L
Apply $\mathbf{N}^{-1}$	10	$\frac{2}{3}m^3+m^2$	L
$\mathbf{x}(\sigma_\ell)\leftarrow\mathbf{x}_0(\sigma_\ell)+\mathbf{Z}_m\mathbf{y}$	11	$2mn$	L
$\mathbf{r}(\sigma_\ell)\leftarrow\mathbf{r}_0(\sigma_\ell)-\left(\mathbf{V}_{m+1}\overline{\mathbf{H}}_m+\sigma\mathbf{Z}_m\right)\mathbf{y}$	12	$2mn$	L

TABLE 5.2
Cost per cycle of extra calculations performed in Algorithm 4.2 when compared to Recycled GMRES.

Operations	Alg. Line	FLOPS in $\mathcal{O}(\cdot)$	$\times$per cycle
$\overline{\mathbf{G}}_m^*\overline{\mathbf{G}}_m$	18	$(m+k+1)^2(m+k)$	1
$\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}$	18	k^2n	1
$\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_m$	18	knm	1
$\mathbf{Z}_m^*\mathbf{Z}_m$	18	m^2n	1
$\mathbf{C}^*\mathbf{Z}_{\mathcal{U}}$	18	k^2n	1
$\mathbf{C}^*\mathbf{Z}_m$	18	knm	1
$\mathbf{V}_{m+1}^*\mathbf{Z}_{\mathcal{U}}$	18	$kn(m+1)$	1
$\mathbf{V}_{m+1}^*\mathbf{Z}_m$	18	$nm(m+1)$	1
$\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m+\sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}$	21	$(m+k+1)^2(m+k)$	L
Sum of 4 matrices	21	$3(m+k)$	L
$\left\{\begin{bmatrix}\mathbf{C} & \mathbf{V}_{m+1}\end{bmatrix}\overline{\mathbf{G}}_m+\sigma\begin{bmatrix}\mathbf{Z}_{\mathcal{U}} & \mathbf{Z}_m\end{bmatrix}\right\}^*\mathbf{r}(\sigma_\ell)$	21	$2(m+k)n$	L
Apply $\mathbf{N}^{-1}$	21	$\frac{2}{3}(m+k)^3+(m+k)^2$	L
Update approx.	22	$2(m+k)n$	L
Update resid.	23	$2(m+k)n$	L

5.2.1. Comparison of Algorithm 4.1 to GMRES.

Algorithm 4.1 is built on top of GMRES. In Table 5.1, we list all additional operations and information about their costs. From this, we can estimate the additional per-cycle FLOP cost and then divide by m to estimate $d_{new}^{(4.1)}$. If we simplify, we see that

$$d_{new}^{(4.1)}=\frac{2}{3}(L+3)m^2+2m(2L+n+1)+6Ln+n.$$

5.2.2. Comparison of Algorithm 4.2 to Recycled GMRES.

Algorithm 4.2 is built on top of recycled GMRES. We can compare the costs of a cycle of each algorithm by looking at the additional costs per cycle of Algorithm 4.2. There are also a few initial one-time overhead costs which must be taken into account. Thus in Table 5.2 we display the additional per-cycle costs of Algorithm 4.2 and in Table 5.3 the additional one-time overhead costs.

We use Table 5.2 to estimate $d_{new}^{(4.2)}$, but we must also take into account the onetime costs shown in Table 5.3 by dividing those costs by the total number of iterations $j_{new}^{(4.2)}$. After

TABLE 5.3
One-time overhead costs in Algorithm 4.2 when compared to Recycled GMRES.

Operations	Alg. Line	FLOPS in $\mathcal{O}(\cdot)$	$\times$ per method execution
$\mathbf{Z}_{\mathcal{U}} \leftarrow \mathbf{Z}_{\mathcal{U}}\mathbf{R}^{-1}$	7	k^3	1
$\mathbf{C}^*\mathbf{Z}_{\mathcal{U}}$	9	k^2n	1
$\mathbf{Z}_{\mathcal{U}}^*\mathbf{Z}_{\mathcal{U}}$	9	k^2n	1
Sum of 4 $k \times k$ matrices	11	$3k$	L
$\left(\mathbf{C} + \sigma\mathbf{Z}_{\mathcal{U}}\right)^* \mathbf{r}(\sigma_\ell)$	12	$2kn$	L
Apply $\mathbf{N}^{-1}$	12	$k^3 + \frac{2}{3}k^2$	L
$\mathbf{x}(\sigma_\ell) \leftarrow \mathbf{x}(\sigma_\ell) + \mathbf{U}\mathbf{y}$	13	$2kn$	L
$\mathbf{r}(\sigma_\ell) \leftarrow \mathbf{r}(\sigma_\ell) - \left(\mathbf{C} + \sigma\mathbf{Z}_{\mathcal{U}}\right)\mathbf{y}$	14	$2kn$	L

simplifying we have

$$\begin{aligned} d_{new}^{(4.2)} &= (1 + \frac{5L}{3})m^2 + (2 + 3k + 3L + 5kL + 2n)m \\ &+1 + 4k + 3k^2 + 4L + 6kL + 5k^2L + n + 3kn + 6Ln \\ &+\frac{1}{m}(\frac{5k^3L}{3} + k^3 + 3k^2L + 2k^2n + 2k^2 + 6kLn + 4kL + kn + k) \\ &+\frac{k^3 + 2k^2n + \left(k^3 + \frac{2k^2}{3}\right)L + 6kLn + 3kL}{j_{new}}. \end{aligned} \tag{5.6}$$

5.2.3. Estimating the costs for specific examples. Now we can compare the costs for a specific example. For Algorithm 4.1, let $m = 50$, $L = 5$, and $n = 10^5$. Then we have that $d_{new}^{(4.1)} \approx 1.3 \times 10^6$. For Algorithm 4.2, let us store a small recycled subspace but use the same amount of storage, i.e., $m = 40$ and $k = 5$. This yields $d_{new}^{(4.2)} \approx 1.3 \times 10^6 + \frac{2.0\times10^6}{j_{new}^{(4.2)}}$.

Admittedly, (5.6) is a bit unwieldy and has many parameters. However, if we make an additional assumption on how Algorithm 4.2 is called, we can simplify the associated cost calculation. Let us assume that $k = \frac{1}{2}m$, i.e., that we maintain a recycled subspace half the size of the associated Krylov subspace dimension. Then we see that we can simplify

$$\begin{aligned} d_{new}^{(4.2)} &= m^3\left(\frac{L}{8j_{new}^{(4.2)}} + \frac{1}{8j_{new}^{(4.2)}}\right) + m^2\left(\frac{L}{6j_{new}^{(4.2)}} + \frac{45L}{8} + \frac{n}{2j_{new}^{(4.2)}} + \frac{27}{8}\right) \\ &+m\left(\frac{3Ln}{j_{new}} + \frac{3L}{2j_{new}^{(4.2)}} + \frac{27L}{4} + 4n + \frac{9}{2}\right) + \frac{3}{2} + 6L + \frac{3n}{2} + 9Ln. \end{aligned}$$

Let us assume for Algorithm 4.1 that we have the same values as before. To have approximately equivalent storage for Algorithm 4.2, we set $m = 25$, and we have $d_{new}^{(4.2)} \approx 1.5 \times 10^6 + \frac{6.9\times10^6}{j_{new}^{(4.2)}}$ FLOPS.

In Figure 5.1, we study the growth in the estimated FLOP costs when all but one parameter are held fixed. For srGMRES, we again assume that $k = \frac{1}{2}m$ and that the total number of iterations needed for Algorithm 4.2 to converge is

$$j_{new}^{(4.2)} = \frac{n}{10^{\frac{4}{9} + \frac{2m}{9m+9}} \log_{10} n}.$$

This is somewhat arbitrary, but it qualitatively matches experimental observations. This formula is derived so that for the case that $n = 10^6$ and for $m \to \infty$ we have that $j_{new}^{(4.2)} \to 100$

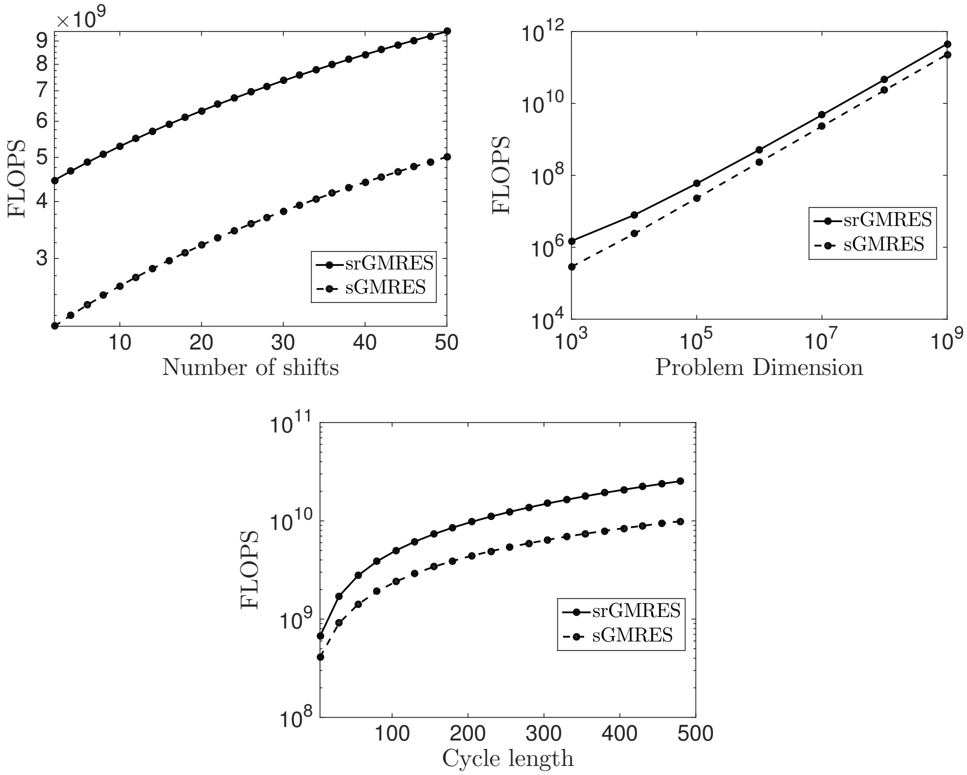

FIG. 5.1. *Estimated cost in FLOPS, respectively, for different numbers of shifts, problem dimensions, and cycle lengths with all other parameters being held constant.*

and the convergence is monotonically decreasing and relatively fast. It is necessary to have some assumption on the value of $j_{new}^{(4.2)}$ since Algorithm 4.2 has some overhead costs which need to be amortized over the total number of iterations. In the three graphs shown, we vary, respectively, the number of shifts (L), the problem dimension (n), and the cycle length (m) with everything else being held constant. For the experiments in which L is held constant, we chose $L = 5$. Similarly, we chose $n = 10^7$ and $m = 100$ in the cases that these parameters were held constant.

We conclude by noting that we consider only one type of costs in this section. In reality, these methods also incur storage costs and data movement costs which are nontrivial for large-scale problems and which must be considered. Furthermore, absent preconditioning, it is clear from the cost calculations that in the case of non-Hermitian shifted systems of the form treated in [20] that the method considered in that paper would be much cheaper than Algorithm 4.2, and absent preconditioning, for general non-Hermitian shifted linear systems satisfying the conditions in [13] (e.g., collinear residuals), that method would outperform Algorithm 4.1. Lastly, both Algorithms 4.1 and 4.2 can be used with flexible preconditioners with *no additional computational or storage costs*.

6. Numerical results.

We performed a series of numerical experiments to demonstrate the effectiveness of our algorithms as well as to compare performance (as measured in both matrix-vector product counts and CPU timings) with other algorithms. All tests were performed in Matlab R2014b (8.4.0.150421) 64-bit running on a Mac Pro workstation with two 2.26 GHz Quad-Core Intel Xeon processors and 12 GB 1066 MHz DDR3 main memory. For these tests, we use *two sets of QCD matrices* downloaded from the University of Florida Sparse Matrix Library [9]. One set of matrices is a collection of seven 3072×3072 complex matrices, and the other is a collection of seven 49152×49152 complex matrices. These matrices, respectively, have $119,808$ nonzero entries (i.e., $\approx 1.3\%$ of the total) and $1,916,928$ nonzero entries (i.e., $\approx 0.08\%$ of the total), and the cost of the matrix-vector product scales linearly with the number of nonzeros. Let us denote the number of nonzero entries n_{nz}. In order for these costs to be able to be related to the quantities in Section 5.2, we express the number of non-zero entries in terms of the dimension n (i.e., the cost of a matrix-vector product is on the order of $\left(\frac{n_{nz}}{n}\right) n$). For both the large and small QCD matrix sets, we have that $\left(\frac{n_{nz}}{n}\right) = 39$ so that the cost of a matrix-vector product is roughly $39n$.

For each matrix $\mathbf{D}$ from the collection, there exists some critical value κ_c such that for $\frac{1}{\kappa_c} < \frac{1}{\kappa} < \infty$, the matrix $\mathbf{A} = \frac{1}{\kappa}\mathbf{I} - \mathbf{D}$ is real-positive. For each $\mathbf{D}$, we took $\mathbf{A} = \left(\frac{1}{\kappa_c} + 10^{-3}\right)\mathbf{I} - \mathbf{D}$ as our base matrix. In our experiments then, each set is taken as the sequence $\{\mathbf{A}_i\}$, and we solve a family of the form (2.2). As described in [9], the matrices $\mathbf{D}$ are discretizations of the Dirac operator used in numerical simulation of quark behavior at different physical temperatures. We note that larger real shifts of $\mathbf{A}_i$ yield better conditioned matrices for all i. For all experiments, we chose the right-hand side $\mathbf{b}_1 = \mathbf{1}$, the vector of ones, and set $\mathbf{b}_i = \mathbf{b}_{i-1} + \mathbf{d}_i$, where $\mathbf{d}_i$ is chosen randomly such that $\|\mathbf{d}_i\| = 10^{-1}$. The requested relative residual tolerance was $\varepsilon = 10^{-8}$. All augmentation was with harmonic Ritz vectors. For all experiments, we preconditioned with an incomplete LU-factorization (ILU) for the system with the smallest shift constructed using the Matlab function `ilu()` called with the default Matlab settings, which is the so-called "no-fill" version of the algorithm also sometimes called `ilu(0)`.

We comment that the usage of ILU was a matter of convenience and effectiveness for these sample problems. Its usage is meant to demonstrate a proof-of-concept rather than as advocating the usage of ILU for large-scale QCD problems. However, we nonetheless calculate the cost of construction and application of this type of ILU preconditioner. Following the description of `ilu(0)` in [31, Algorithm 10.4], if we estimate that the average number of nonzeros per row is $\frac{n_z}{n}$, then the number of FLOPS performed in the triple loop can be bounded by $\frac{n_z}{2}\frac{n_z}{n} = \frac{n_z^2}{2n^2}n$.[3] For both sets of matrices, this rounds up to $761n$, which is a one-time construction cost that is amortized over the entire iteration. Application of the preconditioner involves one forward- and one backward-substitution for the sparse lower- and upper-triangular matrices, respectively. Taken together, these two operations also scale linearly with the number of nonzero elements of $\mathbf{A}$. Here we make the assumption that the cost of both forward- and backward-substitution also scale with the number of nonzero elements of the triangular matrices, each of which has approximately half of the number of nonzero elements as $\mathbf{A}$. Thus the cost of applying the ILU preconditioner can also be estimated as $39n$ for both sets of QCD matrices.

We also comment about methods which we have omitted from testing: the shifted restarted GMRES method [13], the shifted GMRES-DR method [8], and the recursive Recycled GMRES method for shifted systems proposed in [41]. We have omitted these methods from the tests as they do not admit general preconditioning. As such, they require substantially more iterations in many experiments. However, with the methods in [8, 13], there would be some number of

[3] We arrive at this estimate by noting that in the inner loop, one FLOP is performed per cycle. The growth of the outer loop can be bounded by n. For each row visited in the outer loop, the middle loop visits every entry before the diagonal. Thus the growth of the outer and middle loop together can be bounded by $n_z/2$. The inner loop visits every entry in the current row after the current entry visited by the middle loop. The growth of this loop can be estimated by the average number of nonzero entries per row n_z/n.

TABLE 6.1
Matrix-vector product counts for different pairs (m, k) of restart cycle length and recycled subspace dimension for shifted Recycled GMRES. The matrices used in this experiment are the smaller set of QCD matrices from [9]. Experiments were performed for larger values than shown but no further improvement was observed.

$m\backslash k$	5	15	25	35	45	55	65	75	85
5	1566	1295	1205	1161	1146	1131	1126	1116	1111
20	1466	1254	1182	1141	1122	1110	1107	1103	1096
35	1418	1229	1166	1132	1113	1103	1096	1095	1091
50	1363	1223	1158	1128	1114	1105	1099	1097	1090
65	1344	1219	1159	1124	1109	1106	1099	1090	1086
80	1321	1210	1153	1123	1109	1102	1098	1091	1085
95	1321	1210	1153	1124	1108	1100	1097	1093	1084

shifts for which this method would be superior to those presented in this paper, as the cost of recursion in our methods, even with preconditioning, would be greater than simply solving the unpreconditioned problems simultaneously with their shifted GMRES method [13].

Since these experiments involve solving shifted systems with shifts of varying magnitudes, it is useful to have some information about the norms of our test matrices. Therefore, we provide both the one- and two-norms for these matrices (computed, respectively, with the Matlab functions `norm(·, 1)` and `svds(·, 1)`). The 1-norms of these matrices all lie in the interval $(28, 31)$, and their 2-norms lie in the interval $(11, 14)$.

In our first experiment, we tested Algorithm 4.2 with *the set of smaller matrices* for various recycle space dimension sizes and restart cycle lengths. We solve for shifts $\sigma \in \{.01, .02, .03, 1, 2, 3\}$. We report the total required matrix-vector products. We see in Table 6.1 that for these particular QCD matrices, good results can be achieved for a small recycled subspace dimension as long as the cycle length is sufficiently long.

For the remaining tests, we use the *larger set of QCD matrices*. In Table 6.2 we compare time and matrix-vector product counts. For Algorithm 4.2, we chose the cycle-length/recycle subspace dimension pair $(m, k) = (80, 10)$ and use this pair for all experiments with Algorithm 4.2 except for the one shown in Figure 6.1. Parameters for Algorithm 4.1 and other tested methods were chosen in order to have the same per-cycle storage cost of $3k + 2m = 190$ vectors [4]. For each family of linear systems, the experiment was performed ten times and the average time over these ten runs was taken as the run time. We solved for a larger number of shifts of varying magnitudes,

$$\sigma \in \{.001, .002, .003, .04, .05, .06, .07, .8, .9, 1, 1.1, 10, 11, 12\}.$$

We compared four methods (Algorithm 4.1, Algorithm 4.2, sequentially applied GMRES, and sequentially applied Recycled GMRES). We see that for this problem with these shifts, both proposed algorithms outperform the sequential applications of GMRES and Recycled GMRES both in terms of matrix-vector product counts and run times. In this case, the sGMRES algorithm is superior in time to srGMRES but not in terms of matrix-vector products, which demonstrates the difference in overhead costs.

In Figure 6.1, for a total fixed augmented subspace dimension of 100, we investigate how many matrix-vector products are required to solve the same sequence of problems with the same shifts as in the previous experiment for different values of (m, k) such that $m + k = 100$ where m is the dimension of the projected Krylov subspace and k is the dimension of the

[4]for storing $\mathbf{V}_m$, $\mathbf{Z}_m$,$\mathbf{U}$, $\mathbf{C}$, and $\mathbf{Z}_{\mathcal{U}}$

TABLE 6.2
Timing (in seconds) and matrix-vector product (mat-vec) comparisons between preconditioned-shifted rGMRES, shifted GMRES, and sequential applications of rGMRES with cycle length $m = 80$ and recycled subspace dimension $k = 10$ applied to the large QCD matrices. The same preconditioner was used in all experiments.

Method	mat-vecs	time
srGMRES	3117	358.44
sGMRES	4003	322.65
Seq. rGMRES	4379	469.78
Seq. GMRES	5665	489.16

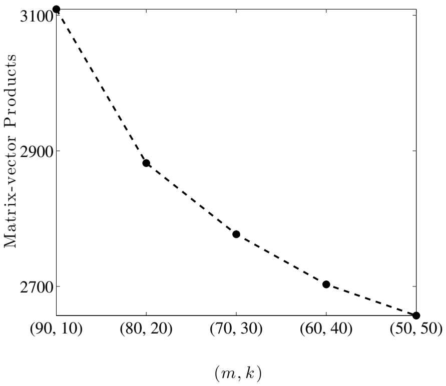

FIG. 6.1. *Matrix-vector product counts for shifted Recycled GMRES for the sequence of large QCD matrices and the same shifts as in Table 6.2 for various pairs (m, k) of Krylov subspace dimension and recycled subspace dimension such that the total augmented subspace Krylov subspace dimension $m + k = 100$.*

recycled subspace. With this we demonstrate a reduction in iterations as we allow more information to be retained in the subspace.

In Table 6.3, we study matrix-vector product counts for different methods for shifts of varying magnitudes. For each shift, we solve just two systems, the base system and one shifted system. Thus we can see how many additional matrix-vector products are required for shifts of different magnitudes. What we observe is that for this set of matrices, the overall performance does not depend on the shift magnitude. For larger shifts, we see that Algorithm 4.2 and sequentially applied rGMRES are comparable *when there is only one shift*. For the QCD matrices, larger real shifts produce better conditioned problems and Table 6.3 illustrates the trade-off between better conditioning and reduced effectiveness of the proposed algorithm for larger shifts. We hypothesize that the smallest values that are attained in the middle of the table are the result of Algorithms 4.1 and 4.2 still being effective for $\mathcal{O}(1)$ shifts where we also observe improved conditioning of the shifted systems.

However, we have seen that for larger numbers of shifts, Algorithm 4.2 can exhibit superior performance. This raises the question, what are the marginal costs of solving each additional linear system for Recycled GMRES and shifted Recycled GMRES, i.e., how many more matrix-vector products does each new shifted system require? This is investigated in Figure 6.2. For two sets of twenty shifts, we calculated the marginal cost of solving each additional shifted system using Algorithm 4.2 as compared to Recycled GMRES. In Figure 6.2 the first set of shifts (left-hand figure) were evenly-spaced points from the interval $[0, 1]$, and the second set of shifts (right-hand figure) were evenly spaced points from the larger interval $[1, 10]$. In Figure 6.2 we see that for the smaller interval, the cost of each new shifted system

TABLE 6.3
Comparison of matrix-vector product counts of 3 methods for different shifts sizes. In each experiment, two systems were solved, the base system and one shifted system with the shift shown in the table column header.

Method$\backslash\Vert\sigma\Vert$	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}
Sh. GMRES Alg. 4.1	1330	1405	1294	967	1067	1265	1306
Sh. rGMRES Alg. 4.2	980	1039	1017	804	908	1105	1144
Seq. rGMRES	1183	1170	1077	812	914	1111	1152

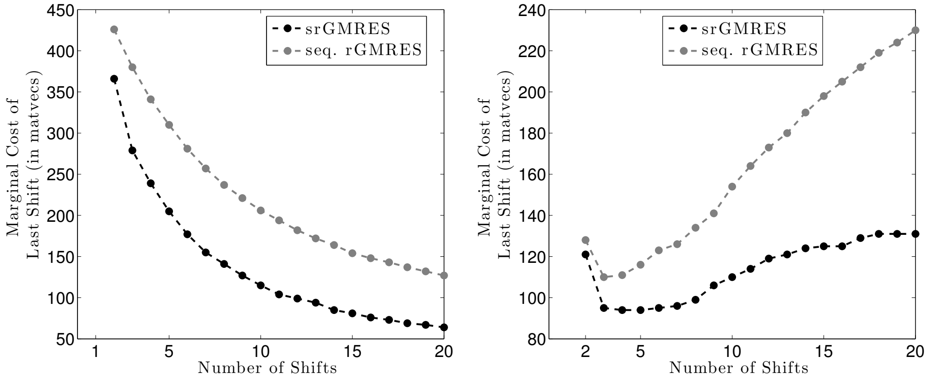

FIG. 6.2. *Comparison of the marginal cost of solving each addition shifted system. For the left-hand figure, the shifts were evenly space points from the interval* $[0,1]$*, and in the right-hand figure, the shifts were evenly spaced points from the larger interval* $[1,10]$.

drops for both algorithms but that Algorithm 4.2 has the lower marginal cost per shift. For the larger set of shifts, we see that the marginal costs for both algorithms actually increases for each new shift. However, the marginal cost of each new shifted system for Algorithm 4.2 becomes more stable (it levels off). For sequentially applied Recycled GMRES, the marginal costs increases steadily for all twenty shifts.

In Figure 6.3, we show the residual histories for systems solved using Algorithm 4.2 for shifts of various magnitudes,

$$\sigma \in \left\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\right\}.$$

From Figure 6.3, we find (in this example) that the amount of improvement for the shifted residuals is somewhat predicted by the shift magnitude, though we again observe that the better conditioning of the systems with larger shifts seems to lead to more rapid convergence but at the expense of reduced effectiveness of the Lanczos-Galerkin projection.

Omitted here is a study of the eigendecomposition of the residuals, which yielded no discernible damping of certain eigenmodes or other interesting observable phenomena after the projection of the shifted residuals in our experiments. Such experiments were to investigate questions of the convergence rates observed in Figure 6.3.

7. Conclusions.

We have presented two new methods for solving a family or a sequence of families of shifted linear systems with general preconditioning. These methods are derived from a general framework, which we also have developed in this paper. These methods use subspaces generated during the minimum residual iteration of the base system to perform the projections for the shifted systems. This technique is fully compatible with right preconditioning, requiring only some additional storage. The strength of methods derived from this framework is that preconditioned methods for shifted systems easily can be built on top

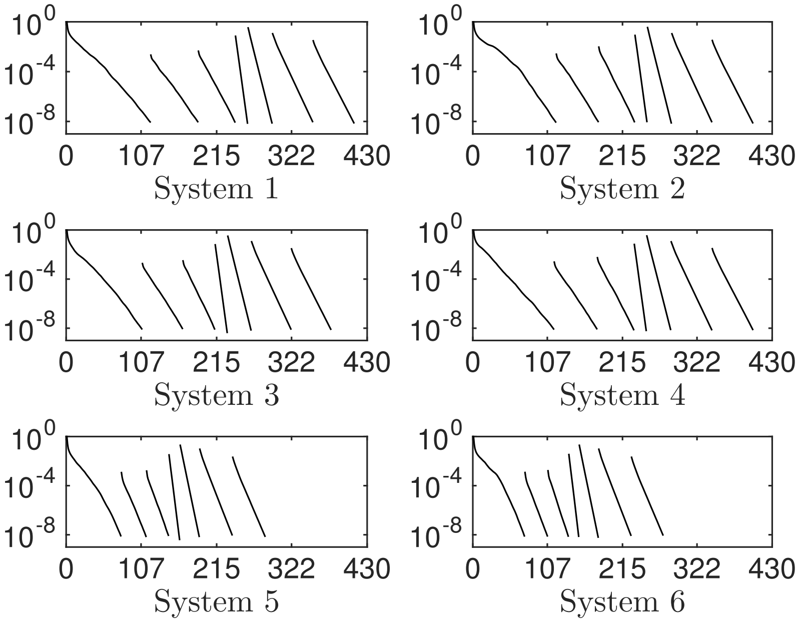

FIG. 6.3. *For the large QCD matrices and* $(m, k) = (100, 5)$*, an illustration of the amount of residual improvement for different magnitude shifts,* $\sigma \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$*. In each subplot, we display the residual curves sequentially to reflect that the algorithm is called for each shifted linear system in sequence. The order in which the systems were solved is the same as the order of the listed shifts.*

of existing minimum residual projection algorithms (and existing codes) with only minor modifications. We developed two algorithms: shifted GMRES and shifted Recycled GMRES. We demonstrated with numerical experiments that both methods can perform competitively.

Finally, we note that our framework is fully compatible with flexible and inexact Krylov subspace methods. As this work all follows from [6, 7, 19, 29], it is also clear that the method is also applicable to the case that we are solving (2.2) but with right-hand sides $\mathbf{b}_{i,\ell}$ which vary both with respect to the coefficient matrix $\mathbf{A}_i$ and the shift σ_ℓ.

Acknowledgments. The author would like to thank Michael Parks who, while reviewing the author's dissertation, made a comment which inspired this work. The author would also like to thank Valeria Simoncini for insightful questions and comments during the author's visit to Bologna and Daniel Szyld for constructive comments. The author further thanks both reviewers for offering comments and criticisms which led to great improvement in the presentation and completeness of this work.

REFERENCES

[1] M. I. AHMAD, D. B. SZYLD, AND M. B. VAN GIJZEN, *Preconditioned multishift BiCG for* $\mathcal{H}_2$*-optimal model reduction*, Tech. Report 12-06-15, Department of Mathematics, Temple University, Philadelphia, June 2012.

[2] A. H. BAKER, E. R. JESSUP, AND T. MANTEUFFEL, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[3] T. BAKHOS, P. K. KITANIDIS, S. LADENHEIM, A. K. SAIBABA, AND D. B. SZYLD, *Multipreconditioned GMRES for shifted systems*, Tech. Report 16-03-31, Department of Mathematics, Temple University,

Philadelphia, March 2016.

[4] M. BAUMANN AND M. B. VAN GIJZEN, *Nested Krylov methods for shifted linear systems*, SIAM J. Sci. Comput., 37 (2015), pp. S90–S112.

[5] M. R. CELIS, J. E. DENNIS, AND R. A. TAPIA, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, 1984 (Boulder, CO, 1984), P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 71–82.

[6] T. F. CHAN AND M. K. NG, *Galerkin projection methods for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[7] T. F. CHAN AND W. L. WAN, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput., 18 (1997), pp. 1698–1721.

[8] 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.

[9] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), Art. 1, (25 pages).

[10] E. DE STURLER, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[11] ———, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.

[12] A. FROMMER, $\mathrm{BiCGStab}(l)$ *for families of shifted linear systems*, Computing, 70 (2003), pp. 87–109.

[13] A. FROMMER AND U. GLÄSSNER, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26.

[14] A. FROMMER, S. GÜSKEN, T. LIPPERT, B. NÖCKEL, AND K. SCHILLING, *Many masses on one stroke: economic computation of quark propagators*, Int. J. Mod. Phys. C, 6 (1995), pp. 627–638.

[15] A. GAUL, *Recycling Krylov Subspace Methods for Sequences of Linear Systems: Analysis and Applications*, Ph.D. Thesis, Fakultät II-Mathematik und Naturwissenschaften, Technischen Universität Berlin, Berlin, 2014.

[16] A. GAUL, M. H. GUTKNECHT, J. LIESEN, AND R. NABBEN, *A framework for deflated and augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 495–518.

[17] A. GAUL AND N. SCHLÖMER, *Preconditioned recycling Krylov subspace methods for self-adjoint problems*, Electron. Trans. Numer. Anal., 44 (2015), pp. 522–547.
http://etna.ricam.oeaw.ac.at/vol.44.2015/pp522-547.dir/pp522-547.pdf

[18] B. JEGERLEHNER, *Krylov space solvers for sparse linear systems.*, Tech. Report IUHET-353, Department of Physics, Indiana University, Blomington, 1996.

[19] M. KILMER, E. MILLER, AND C. RAPPAPORT, *QMR-based projection techniques for the solution of non-Hermitian systems with multiple right-hand sides*, SIAM J. Sci. Comput., 23 (2001), pp. 761–780.

[20] M. E. KILMER AND E. DE STURLER, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.

[21] S. KIRCHNER, *IDR-Verfahren zur Lösung von Familien geshifteter linearer Gleichungssysteme*, Master's thesis, Bergische Universität Wuppertal, Department of Mathematics, Wuppertal, Germany, 2011.

[22] R. B. LEHOUCQ AND D. C. SORENSEN, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

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

[24] R. B. MORGAN, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[25] R. B. MORGAN AND M. ZENG, *Harmonic projection methods for large non-symmetric eigenvalue problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 33–55.

[26] C. C. PAIGE, B. N. PARLETT, AND H. A. VAN DER VORST, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133.

[27] M. L. PARKS, E. DE STURLER, G. MACKEY, D. D. JOHNSON, AND S. MAITI, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.

[28] B. N. PARLETT, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, Linear Algebra Appl., 29 (1980), pp. 323–346.

[29] Y. SAAD, *On the Lánczos method for solving symmetric linear systems with several right-hand sides*, Math. Comp., 48 (1987), pp. 651–662.

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

[31] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[32] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[33] Y. SAAD, M. YEUNG, J. ERHEL, AND F. GUYOMARC'H, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[34] 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.

[35] M. SARKIS AND D. B. SZYLD, *Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1612–1621.

[36] V. SIMONCINI, *On the numerical solution of $AX - XB = C$*, BIT, 36 (1996), pp. 814–830.

[37] ———, *Restarted full orthogonalization method for shifted linear systems*, BIT, 43 (2003), pp. 459–466.

[38] 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.

[39] K. M. SOODHALTER, *Two recursive GMRES-type methods for shifted linear systems with general preconditioning*, Preprint on arXiv, 2014. https://arxiv.org/abs/1403.4428

[40] ———, *Block Krylov subspace recycling for shifted systems with unrelated right-hand sides*, SIAM J. Sci. Comput., 38 (2016), pp. A302–A324.

[41] K. M. SOODHALTER, D. B. SZYLD, AND F. XUE, *Krylov subspace recycling for sequences of shifted linear systems*, Appl. Numer. Math., 81 (2014), pp. 105–118.

[42] S. WANG, E. DE STURLER, AND G. H. PAULINO, *Large-scale topology optimization using preconditioned Krylov subspace methods with recycling*, Internat. J. Numer. Methods Engrg., 69 (2007), pp. 2441–2468.

[43] G. WU, Y.-C. WANG, AND X.-Q. JIN, *A preconditioned and shifted GMRES algorithm for the PageRank problem with multiple damping factors*, SIAM J. Sci. Comput., 34 (2012), pp. A2558–A2575.