

MSTAB: STABILIZED INDUCED DIMENSION REDUCTION FOR KRYLOV SUBSPACE RECYCLING*

MARTIN P. NEUENHOFEN[†] AND CHEN GREIF[‡]

Abstract. We introduce *Mstab*, a Krylov subspace recycling method for the iterative solution of sequences of linear systems, where the system matrix is fixed and is large, sparse, and nonsymmetric, and the right-hand-side vectors are available in sequence. *Mstab* utilizes the short-recurrence principle of induced dimension reduction-type methods, adapted to solve sequences of linear systems. Using *IDRstab* for solving the linear system with the first right-hand side, the proposed method then recycles the Petrov space constructed throughout the solution of that system, generating a larger initial space for subsequent linear systems. The richer space potentially produces a rapidly convergent scheme. Numerical experiments demonstrate that *Mstab* often enters the superlinear convergence regime faster than other Krylov-type recycling methods.

Key words. sequence of linear systems, iterative solvers, Krylov subspace recycling, short recurrences, BiCGStab, IDRstab, Mstab, Sonneveld spaces

AMS subject classifications. 93C05, 65F10, 93A15, 65F50, 65N22

DOI. 10.1137/16M1092465

1. Introduction. We consider iterative methods for the solution of sequences of large sparse nonsymmetric linear systems

$$(1.1) \quad \mathbf{Ax}^{(\iota)} = \mathbf{b}^{(\iota)}, \quad \iota = 1, \dots, n_{\text{Systems}},$$

with fixed nonsingular $\mathbf{A} \in \mathbb{C}^{N \times N}$, where the right-hand sides $\mathbf{b}^{(\iota)} \in \mathbb{C}^N$ are provided in sequence. Such situations occur, for example, when applying an implicit time stepping scheme to numerically solve a transient partial differential equation (PDE). Relevant applications are, e.g., topology optimization [9], model reduction [6], structural dynamics [18], quantum chromodynamics [7], electrical circuit analysis [29], fluid dynamics [15], and optical tomography [13]. In all the aforementioned references a technique called Krylov subspace recycling (KSSR) is used.

It is useful to start our discussion by establishing the notation and a few basic principles for solving a single linear system, $\mathbf{Ax} = \mathbf{b}$. Suppose \mathbf{x}_0 is an initial guess of the solution, and let $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$ be the initial residual. We define, as usual, the *Krylov subspace of degree* $k \in \mathbb{N}$ associated with \mathbf{A} and \mathbf{r}_0 as

$$\mathcal{K}_k(\mathbf{A}; \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}.$$

Standard Krylov subspace methods solve a single linear system by searching in the k th iteration an approximate solution $\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$ such that the residual \mathbf{r}_k

*Submitted to the journal's Computational Methods in Science and Engineering section September 6, 2016; accepted for publication (in revised form) January 16, 2018; published electronically April 12, 2018.

<http://www.siam.org/journals/sisc/40-2/M109246.html>

Funding: The second author's work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

[†]MartinNeuenhofen@googlemail.com, www.MartinNeuenhofen.de.

[‡]Department of Computer Science, University of British Columbia, Vancouver, BC, Canada (greif@cs.ubc.ca).

is perpendicular to a Petrov space \mathcal{C}_k of a growing dimension. For example, Galerkin methods use the actual Krylov subspace, namely, $\mathcal{C}_k = \mathcal{K}_k(\mathbf{A}; \mathbf{r}_0)$. Typical optimality criteria for constructing the iterates imply that it is desirable to enrich the Petrov space throughout the iteration.

In general, Krylov subspace methods can simultaneously achieve optimality (such as residual norm minimization) and use short recurrences only if the matrix is Hermitian or symmetric; the most well-known example here is the conjugate gradient method for symmetric positive definite matrices, which minimizes the \mathbf{A} -norm of the error using three-term recurrence relations. When nonsymmetric linear systems are considered, we typically need to give up either optimality or short recurrences. GMRES [19], for example, is based on seeking residual norm minimization at the cost of using long-term recurrence relations. In contrast, the family of BiCG methods is based on bi-Lanczos tridiagonalization, maintaining the short recurrences while giving up optimality. The BiCG method accomplishes this by constructing Krylov subspaces associated with \mathbf{A} and with \mathbf{A}^H . Several methods have been developed, which improve upon BiCG by more efficiently using \mathbf{A}^H -information. For example, BiCGStab [25] combines the polynomials underlying BiCG with a minimal residual step.

The induced dimension reduction (IDR) method [28, 23, 11] is strongly related to BiCG-type methods and can be interpreted in a similar way in terms of the polynomials that are generated throughout the iteration. But it is based on a different point of view for constructing the search space. Methods of this class compute a sequence of numerical solutions whose residuals reside in a sequence of shrinking spaces. After finitely many iterations these spaces vanish, resulting in termination with the exact solution (in the absence of roundoff errors).

When it comes to solving systems with multiple right-hand sides, as in (1.1), the above mentioned methods serve as the basis for developing new techniques. For the purpose of illustration, suppose that we need to solve two linear systems, where the right-hand-side vectors are given in sequence:

$$(1.2) \quad \mathbf{Ax}^{(1)} = \mathbf{b}^{(1)}, \quad \mathbf{Ax}^{(2)} = \mathbf{b}^{(2)}.$$

A straightforward (but not necessarily optimal) approach is to solve both systems subsequently with a conventional Krylov subspace method, applied successively to the linear systems in hand. That is, the first system $\mathbf{Ax}^{(1)} = \mathbf{b}^{(1)}$ is solved for $\mathbf{x}^{(1)}$ with a Krylov subspace method in m iterations. When solving the first system, the Krylov subspace method iteratively builds a Petrov space $\mathcal{C}_m^{(1)}$, starting from $\mathcal{C}_0^{(1)} = \{\mathbf{0}\}$. Subsequently, the second system is solved for $\mathbf{x}^{(2)}$ by applying a conventional Krylov subspace method to $\mathbf{Ax}^{(2)} = \mathbf{b}^{(2)}$. Since no information from the first system is reused, this means the initial Petrov space for the second system is $\mathcal{C}_0^{(2)} = \{\mathbf{0}\}$. This seems like a waste, because $\mathcal{C}_m^{(1)}$ may hold useful information on powers of \mathbf{A} (and hence its spectrum). It is thus desirable to develop methods that give rise to a better exploitation of the subspace $\mathcal{C}_m^{(1)}$ while solving the second system. This principle easily generalizes to situations where more than two linear systems need to be solved.

The KSSR approach [6, 18, 29] aims to take advantage of the information that has been accumulated during the iterative solution of the first system: $\mathbf{Ax}^{(1)} = \mathbf{b}^{(1)}$ is solved for $\mathbf{x}^{(1)}$ with a Krylov subspace method in m iterations, as before, and at that point a compressed basis matrix comprising a few search directions is stored in memory. Subsequently, the second system is solved for $\mathbf{x}^{(2)}$ by applying any

Krylov subspace method to a modified linear system whose coefficient matrix is a projected matrix associated with \mathbf{A} and the aforementioned compressed basis matrix. At their core, KSSR methods are based on periodically truncating a basis of deflation vectors against which the updated residual is orthogonalized in a long-recursive way. This is true even when the original Krylov subspace method is based on short recurrences.

Our proposed new method is based on applying IDRstab as a starting point. The approach we take is similar to deflation in the sense that the residual is maintained in a smaller-dimensional space, but in contrast to KSSR methods, the computations required to maintain the residual in the smaller space are based on the short recurrences of the IDR method itself. This, in turn, introduces a potential computational advantage, as long as the construction of the subsequent space is sufficiently effective. Our approach is inspired by the work of Miltenberger [14], who performed numerical experiments with recycled auxiliary vectors in IDR(s).

An outline of the remainder of this paper follows. In section 2 we review KSSR methods. In section 3 we present our new KSSR method $\mathcal{M}(s)\text{stab}(\ell)$ and discuss its convergence properties in comparison to IDR(s) $\text{stab}(\ell)$. $\mathcal{M}\text{stab}$ is a parameter-dependent method; section 4 provides examples of how the performance of $\mathcal{M}\text{stab}$ changes with the parameter and lays out an algorithmic way of finding good choices for it. In section 5 we present numerical experiments based on discretized elliptic and parabolic PDEs, where $\mathcal{M}\text{stab}$ outperforms GMRES, BiCGStab, IDRstab, RBiCGStab, and GCRO-DR. Finally, in section 6 we offer some brief concluding remarks.

2. Krylov subspace recycling methods. As mentioned in the introduction, KSSR methods aim to take advantage of the information that has been accumulated during the iterative solution of a previous system. Going back to (1.2), below we further illustrate this in more detail. The first system $\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}^{(1)}$ is solved for $\mathbf{x}^{(1)}$ with a Krylov subspace method in m iterations, and during the process a compressed basis matrix $\mathbf{U}_c \in \mathbb{C}^{N \times c}$, $c \ll m$, of a few search directions is stored in memory. Afterward, the second system is solved for $\mathbf{x}^{(2)}$ by applying any Krylov subspace method to the system

$$(2.1) \quad \underbrace{(\mathbf{I} - \mathbf{A}\mathbf{U}_c(\mathbf{A}\mathbf{U}_c)^\dagger)\mathbf{A}}_{=:\tilde{\mathbf{A}}} \tilde{\mathbf{x}}^{(2)} = \underbrace{(\mathbf{I} - \mathbf{A}\mathbf{U}_c(\mathbf{A}\mathbf{U}_c)^\dagger)\mathbf{b}^{(2)}}_{=:\tilde{\mathbf{b}}^{(2)}}$$

to solve for $\tilde{\mathbf{x}}^{(2)}$. The original solution is

$$\mathbf{x}^{(2)} = \tilde{\mathbf{x}}^{(2)} + \mathbf{U}_c(\mathbf{A}\mathbf{U}_c)^\dagger(\mathbf{b}^{(2)} - \mathbf{A}\tilde{\mathbf{x}}^{(2)}).$$

The matrix $(\mathbf{I} - \mathbf{A}\mathbf{U}_c(\mathbf{A}\mathbf{U}_c)^\dagger)$ is a projector onto $\mathcal{N}(\mathbf{A}\mathbf{U}_c)$. Applying it accomplishes a deflation effect: small eigenvalues are removed from the spectrum in (2.1), which in turn may accelerate convergence. For more on deflated Krylov subspace methods, see, for example, [10, 12].

Matrix-vector products with $\tilde{\mathbf{A}}$ are more computationally costly than matrix-vector products with \mathbf{A} but are not prohibitive: matrix-vector products with $(\mathbf{A}\mathbf{U}_c)^\dagger$ can be facilitated by applying once and for all a stable orthogonalization scheme (such as modified or iterated Gram-Schmidt, Givens, or Householder) for $(\mathbf{A}\mathbf{U}_c)$. The advantage of this approach is that the initial Petrov space for (2.1) is now $\mathcal{C}_0^{(2)} = \text{range}(\mathbf{A}\mathbf{U}_c)$, which is c -dimensional upon the start of the iteration. Depending on the space spanned by the columns of \mathbf{U}_c , this may yield faster convergence.

Let us briefly review a selection of various KSSR methods from the literature. In the GCRO framework [6, 29], a matrix \mathbf{C}_c with orthonormal columns is explicitly formed: $\mathbf{C}_c = \mathbf{A}\mathbf{U}_c$, where \mathbf{U}_c is a basis of search directions. Here $\mathbf{U}_c, \mathbf{C}_c \in \mathbb{C}^{N \times c}$ have a small size of c , e.g., $c = 20$ columns. All the methods start with empty or recycled matrices for \mathbf{U}_c and \mathbf{C}_c and during the solution process they apply periodically a heuristic to update the columns of these matrices. New columns are overwritten, aiming to accelerate convergence over the conventional Krylov method applied to (2.1). All the methods discussed below differ almost only in the heuristic that they use to update the columns of \mathbf{U}_c and \mathbf{C}_c .

GMRES-DR [16] applies a restarted GMRES variant on (2.1), where in each restart the matrices $\mathbf{U}_c, \mathbf{C}_c$ are updated by estimated Ritz vectors. This strategy can be applied for a single system, as well as for a sequence of systems. For the case of Hermitian systems the nested GMRES procedure is replaced by MINRES, yielding the recycling method R-MINRES [9]. Although R-MINRES is a short-recurrence method, the columns for \mathbf{C}_c are still periodically updated. Thus, regardless of the short-recursion nature of the applied Krylov method, the recycled Petrov space is still truncated.

In GCRO-DR [18] the recycled basis matrix \mathbf{U}_c is chosen as a basis of approximate Ritz vectors. GCRO-OT [8] chooses \mathbf{U}_c by another heuristic that is called optimal truncation. There also exist recycling variants of BiCG and BiCGStab [5, 2, 1] that effectively solve (2.1) with the respective short-recurrence method, optionally under restarts combined with updates of \mathbf{C}_c .

The above described KSSR methods have been shown to be very effective for many applications; see, for example, [16, 9, 6, 5, 2, 1, 13, 24]. They can all be described as methods for constructing and solving the system (2.1), where a full basis of $\mathcal{C}_0^{(2)}$ must be stored. However, a potential disadvantage of these methods is that they require additional storage resources and entail a nontrivial computational cost, since fundamentally, they orthogonalize the residuals against the deflation space by using (long) recurrences of length c .

3. Mstab: An IDR method for Krylov subspace recycling. $\mathcal{M}(s)\text{stab}(\ell)$ is a new IDR KSSR method. It employs data that is computed by $\text{IDR}(s)\text{stab}(\ell)$.

As mentioned in the introduction, the IDR framework [11] is related to BiCG-type methods but it is based on a different point of view. Methods based on the IDR concept compute a sequence of numerical solutions \mathbf{x}_j with residuals \mathbf{r}_j , where $\mathbf{r}_j \in \mathcal{G}_j$, $j = 0, 1, 2, \dots$. The spaces \mathcal{G}_j are iteratively shrinking, $\mathcal{G}_0 \supset \mathcal{G}_1 \supset \mathcal{G}_2 \supset \dots$, such that there exists $j \in \mathbb{N}$ where $\dim(\mathcal{G}_j) = 0$. IDR methods terminate after finitely many iterations with the exact solution [28, 23, 11], since $\mathbf{r}_j \in \mathcal{G}_j = \{\mathbf{0}\} \Rightarrow \mathbf{r}_j = \mathbf{0}$.

At its core, an IDR method has two components:

- Initialization. An initial guess \mathbf{x}_0 with $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is constructed such that $\mathbf{x}_0, \mathbf{r}_0 \in \mathbb{C}^N$, $\mathbf{U}_0 \in \mathbb{C}^{N \times s}$, $\mathbf{V}_0 = \mathbf{A}\mathbf{U}_0$, where $\mathbf{r}_0 \in \mathcal{G}_0$ and $\text{range}(\mathbf{V}_0) \subset \mathcal{G}_0$.
- Iterative scheme. Given input $\mathbf{x}_j, \mathbf{r}_j \in \mathbb{C}^N$, $\mathbf{U}_j, \mathbf{V}_j \in \mathbb{C}^{N \times s}$, $\mathbf{V}_j = \mathbf{A}\mathbf{U}_j$, where $\mathbf{r}_j \in \mathcal{G}_j$, $\text{range}(\mathbf{V}_j) \subset \mathcal{G}_j$, the iterative scheme returns data $\mathbf{x}_{j+\ell}, \mathbf{r}_{j+\ell} = \mathbf{b} - \mathbf{A}\mathbf{x}_{j+\ell} \in \mathbb{C}^N$, $\mathbf{U}_{j+\ell}, \mathbf{V}_{j+\ell} = \mathbf{A}\mathbf{U}_{j+\ell} \in \mathbb{C}^{N \times s}$, such that $\mathbf{r}_{j+\ell} \in \mathcal{G}_{j+\ell}$, $\text{range}(\mathbf{V}_{j+\ell}) \subset \mathcal{G}_{j+\ell}$. The scheme uses short recurrences and stores $\mathcal{O}(N s \ell)$ values.

The algorithm consists of performing the initialization, followed by repeatedly calling the iterative scheme until $\mathbf{x}_j, \mathbf{r}_j$ satisfy a termination criterion. The iterates \mathbf{x}_j are constructed such that their associated residuals reside in successively shrinking spaces. A concrete method can now be constructed if we can generate

- a sequence of spaces $\{\mathcal{G}_j\}$ that satisfy a (yet undiscussed) recursive property;
- an initial guess whose residual resides in \mathcal{G}_0 and s additional vectors that also live in \mathcal{G}_0 .

3.1. IDRstab. With a recipe in hand that requires defining a sequence of spaces and an initialization procedure, we now discuss one particular instance of an IDR-type method: $\text{IDR}(s)\text{stab}(\ell)$. It uses the following spaces.

DEFINITION 1 (IDR spaces). *Given $\mathbf{A} \in \mathbb{C}^{N \times N}$, $\mathbf{r}_0 \in \mathbb{C}^N$, $\mathbf{P} \in \mathbb{C}^{N \times s}$, $\{\omega_j\}_{j \in \mathbb{N}} \in \mathbb{C} \setminus \{0\}$, where $s < N$ and $s, N \in \mathbb{N}$, $\text{rank}(\mathbf{P}) = s$, define recursively*

$$(3.1) \quad \begin{aligned} \mathcal{G}_0 &:= \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0), \\ \mathcal{G}_{j+1} &:= (\mathbf{I} - \omega_{j+1} \cdot \mathbf{A}) \cdot (\mathcal{G}_j \cap \text{range}(\mathbf{P})^\perp) \quad \forall j \in \mathbb{N}_0. \end{aligned}$$

The vector space \mathcal{G}_j is called *IDR space of degree j* .

THEOREM 1 (induced dimension reduction). *For the above IDR spaces, under the mild assumption that $\text{range}(\mathbf{P})$ and $\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0)$ do not share a nontrivial invariant subspace of \mathbf{A} , the following holds:*

- (a) $\mathcal{G}_j \subset \mathcal{G}_{j-1} \quad \forall j \in \mathbb{N}$,
- (b) $\dim(\mathcal{G}_j) \leq \max\{0, \dim(\mathcal{G}_{j-1}) - s\} \quad \forall j \in \mathbb{N}$.

Proof. See [26, 11] and [23, p. 1037]. □

Next, an initialization has to be designed to construct $\mathbf{x}_0, \mathbf{r}_0, \mathbf{U}_0, \mathbf{V}_0$. Since \mathcal{G}_0 is the full Krylov subspace of \mathbf{r}_0 , choosing \mathbf{U}_0 as a basis for the Krylov subspace $\mathcal{K}_s(\mathbf{A}; \mathbf{r}_0)$ serves an initialization.

IDRstab [22] combines the recurrence (3.1) with the stabilization approach of BiCGStab(ℓ). The scalars $\omega_{j+1}, \dots, \omega_{j+\ell}$ are computed so that $\|\mathbf{r}_{j+\ell}\|_2$ is minimized. That is, IDRstab is a clever method that performs a higher-dimensional minimization that defines $\omega_{j+1}, \dots, \omega_{j+\ell}$ values after the next vector in the space has been computed. The scheme is described in detail in [22, section 5]; see also [17, Chapter 4].

At this point the method $\text{IDR}(s)\text{stab}(\ell)$ has been fully defined. Efficient implementations can be found in [3, 4, 17]. From the property $\mathbf{r}_j \in \mathcal{G}_j$ and Theorem 1 we can determine the dimension of the Petrov spaces of $\text{IDR}(s)\text{stab}(\ell)$. As is common for conventional Krylov methods, the Petrov spaces \mathcal{C}_j are the spaces against which the residuals \mathbf{r}_j are iteratively orthogonalized. This definition holds also for IDR-type methods [20]. The restriction $\mathbf{r}_j \in \mathcal{G}_j$ is equivalent to $\mathbf{r}_j \perp \mathcal{C}_j$ when defining $\mathcal{C}_j := \mathcal{G}_j^\perp$. Since $\mathbf{r}_j \in \mathcal{G}_j \cap \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0)$ holds by construction, we remove $(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0))^\perp$ from \mathcal{C}_j . In what follows we use the term *associated Petrov space* of \mathcal{G}_j to refer to the Petrov space against which a residual is orthogonal when it lives in \mathcal{G}_j .

In the lemma that follows, the symbol \setminus denotes the set-minus operator.

LEMMA 1 (Petrov spaces of IDR spaces). *Consider the associated Petrov spaces $\{\mathcal{C}_j\}_{j \in \mathbb{N}_0} \subset \mathbb{C}^N$ to the IDR spaces $\{\mathcal{G}_j\}$:*

$$\mathcal{C}_j := \mathcal{G}_j^\perp \setminus (\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0))^\perp.$$

Let the requirement of Theorem 1 hold. Then

$$\dim(\mathcal{C}_j) \geq j s.$$

Proof. Since \mathcal{C}_j is the subtraction of two orthogonal complements of vector spaces, it is a vector space. The proof follows by induction over $j \in \mathbb{N}_0$, using $\dim(\mathcal{C}_0) = 0$ and $\dim(\mathcal{G}_j) - \dim(\mathcal{G}_{j+1}) \geq s$. □

After $j(s + 1)$ matrix-vector products $\text{IDR}(s)\text{stab}(\ell)$ has generated the residual $\mathbf{r}_j \in \mathcal{G}_j$, i.e., $\mathbf{r}_j \perp \mathcal{C}_j$, where \mathcal{C}_j is of about the dimension of the number of matrix-vector products, i.e., the dimension of the search space.

3.2. $\mathcal{M}(s)\text{stab}(\ell)$. During the computational work of solving the linear system, IDRstab repeatedly applies an iterative scheme in which data $\mathbf{x}_j, \mathbf{r}_j, \mathbf{U}_j, \mathbf{V}_j$ is replaced by $\mathbf{x}_{j+\ell}, \mathbf{r}_{j+\ell}, \mathbf{U}_{j+\ell}, \mathbf{V}_{j+\ell}$. Therefore, for a user-supplied value $p \in \mathbb{N} \cdot \ell$ we could modify IDRstab such that it returns a matrix

$$(3.2) \quad \hat{\mathbf{U}} := \mathbf{U}_p.$$

\mathcal{Mstab} adapts IDRstab in a way that allows for efficiently utilizing data that has been accumulated throughout the solution process. Figure 3.1 shows how the method is applied. The first linear system is solved using IDRstab . An additional input number $p \in \mathbb{N}$ specifies an IDR space \mathcal{G}_p of which data $\hat{\mathbf{U}} \in \mathbb{C}^{N \times s}$ with $\text{range}(\mathbf{A}\hat{\mathbf{U}}) \in \mathcal{G}_p$ is returned together with \mathbf{P} when $\text{IDR}(s)\text{stab}(\ell)$ terminates. Then, $\mathcal{M}(s)\text{stab}(\ell)$ is fed with the inputs $\mathbf{A}, \mathbf{b}^{(2)}, \mathbf{x}_0^{(2)}, \mathbf{P}, \hat{\mathbf{U}}$; see Algorithm 1.

PROPOSITION 1. Consider $p \in \mathbb{N}$, where $\mathcal{G}_p^{(1)} \neq \{\mathbf{0}\}$ is the IDR space of degree p of IDRstab when solving the first linear system. Under mild assumptions, the Petrov spaces of $\mathcal{M}(s)\text{stab}(\ell)$ for the second linear system have the dimension

$$\dim(\mathcal{C}_j^{(2)}) = j s + p(s - 1),$$

Algorithm 1. $\mathcal{M}(s)\text{stab}(\ell)$.

```

1: procedure MSTAB( $\mathbf{A}, \mathbf{b}, \mathbf{x}_0, \ell, \text{tol}, \mathbf{P}, \hat{\mathbf{U}}$ )
2:   // 1) Initialization
3:    $\mathbf{x} := \mathbf{x}_0, \mathbf{r} := \mathbf{b} - \mathbf{A}\mathbf{x}$  //  $\mathbf{r} \in \mathcal{M}_p$ 
4:    $\mathbf{U} := \hat{\mathbf{U}}, \mathbf{V} := \mathbf{A}\mathbf{U}$  //  $\text{range}(\mathbf{V}) \subset \mathcal{M}_p$ 
5:   // 2) Repetition
6:    $j := 0$ 
7:   while  $\|\mathbf{r}\|/\|\mathbf{b}\| > \text{tol}$  do
8:      $[\mathbf{x}, \mathbf{r}, \mathbf{U}, \mathbf{V}] := \text{ITERATIVEScheme}(\mathbf{A}, \mathbf{P}, \mathbf{b}, \ell; \mathbf{x}, \mathbf{r}, \mathbf{U}, \mathbf{V})$ 
9:      $j := j + \ell$ 
10:    //  $\mathbf{r} \in \mathcal{M}_{p+j}, \text{range}(\mathbf{V}) \subset \mathcal{M}_{p+j}$ 
11:  end while
12:  return  $\mathbf{x}$ 
13: end procedure

```

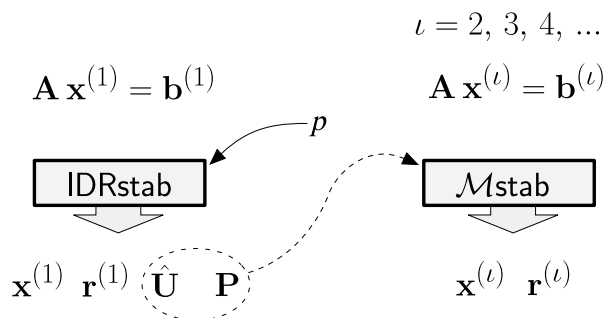


FIG. 3.1. Calling \mathcal{Mstab} to solve a sequence of linear systems.

whereas if we apply $\text{IDR}(s)\text{stab}(\ell)$ to solve the second system, the dimension of its Petrov space is only

$$\dim(\mathcal{C}_j^{(2)}) = j s.$$

Both methods have the same computational cost per iteration.

We have discussed why a larger-dimensional Petrov space may increase the prospects of faster convergence. Thus, the Petrov space associated with $\mathcal{M}(s)\text{stab}(\ell)$ is larger than the corresponding one for IDRstab , which illustrates the potential of $\mathcal{M}(s)\text{stab}(\ell)$. Let us characterize the sequence of nested spaces that $\mathcal{M}\text{stab}$ utilizes.

DEFINITION 2 (\mathcal{M} -spaces). Given $\mathbf{A} \in \mathbb{C}^{N \times N}$, $\mathbf{r}_0^{(1)}, \mathbf{r}_0^{(2)} \in \mathbb{C}^N$, $\mathbf{P} \in \mathbb{C}^{N \times s}$, $\{\omega_j\}_{j \in \mathbb{N}} \in \mathbb{C} \setminus \{0\}$, $p \in \mathbb{N}$, where $s < N$ and $s, N \in \mathbb{N}$, $\text{rank}(\mathbf{P}) = s$, define recursively

$$\begin{aligned} \mathcal{M}_0 &:= \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}) + \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(2)}), \\ (3.3) \quad \mathcal{M}_{j+1} &:= (\mathbf{I} - \omega_{j+1} \cdot \mathbf{A}) \cdot (\mathcal{M}_j \cap \text{range}(\mathbf{P})^\perp) + \text{span}\{\mathbf{r}_0^{(2)}\} \quad \forall j = 0, \dots, p-1, \\ \mathcal{M}_{j+1} &:= (\mathbf{I} - \omega_{j+1} \cdot \mathbf{A}) \cdot (\mathcal{M}_j \cap \text{range}(\mathbf{P})^\perp) \quad \forall j = p, p+1, p+2, \dots \end{aligned}$$

The vector space \mathcal{M}_j is called the \mathcal{M} -space of degree j .

One can show that \mathcal{M} -spaces have similar properties to IDR spaces.

THEOREM 2 (properties of \mathcal{M} -spaces). Consider the IDR spaces $\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_p$ that arise from \mathbf{A} , $\mathbf{r}_0^{(1)}$, \mathbf{P} , $\{\omega_j\}$ and the \mathcal{M} -spaces that arise from \mathbf{A} , $\mathbf{r}_0^{(1)}$, $\mathbf{r}_0^{(2)}$, \mathbf{P} , $\{\omega_j\}$, p . If $\text{range}(\mathbf{P})$ and \mathcal{M}_0 do not share a nontrivial invariant subspace of \mathbf{A} , then

- (a) $\mathcal{M}_j \subset \mathcal{M}_{j-1} \quad \forall j \in \mathbb{N}$,
- (b) $\dim(\mathcal{M}_j) \leq \max\{0, \dim(\mathcal{M}_{j-1}) - s\} \quad \forall j = p+1, p+2, \dots$,
- (c) $\mathcal{G}_p \subset \mathcal{M}_p$, $\mathbf{r}_0^{(2)} \in \mathcal{M}_p$

Proof. A complete proof is given in the first author’s master’s thesis [17, Theorem 2, p. 97]. Let us provide a sketch of the proof, with the main details. Item (a) can be shown by induction over $j = 0, 1, 2, \dots$

Proving item (b) relies on making use of the following superspace:

$$\mathcal{M}_{j+d} \subset \left(\prod_{k=1}^d (\mathbf{I} - \omega_{j+k} \cdot \mathbf{A}) \right) \cdot (\mathcal{M}_j \cap \mathcal{K}_d^\perp(\mathbf{A}^H; \mathbf{P})) + \mathcal{K}_d(\mathbf{A}; \mathbf{r}_0^{(2)}).$$

The above relation is shown in [17, Lemma 4, p. 99]. It follows by induction over $d = 0, 1, 2, \dots$ for every $j \in \mathbb{N}_0$, where [11, equation (3.11)] is used in the induction step. Using the nontrivial invariant subspace requirement, a bound on the dimension of the superspace is readily available. \mathcal{M}_p does not share a nontrivial invariant subspace with $\text{range}(\mathbf{P})$ since $\mathcal{M}_0 \supset \mathcal{M}_p$ did not. Bounds for the dimension of \mathcal{M}_j for $j \geq p$ follow from the equivalence of (3.3) to (3.1).

Item (c) can be proved by induction, where $\mathcal{G}_j \subset \mathcal{M}_j$ is shown for $j = 0, 1, 2, \dots, p$. □

We note that the property (a) and (3.3) are identical to the properties of IDR spaces. The reason for defining the \mathcal{M} -spaces in this particular way is that with the help of the data $\hat{\mathbf{U}}$ and by applying $\text{IDR}(s)\text{stab}(\ell)$ for solving the first system, we can design an initialization procedure for $\mathcal{M}(s)\text{stab}(\ell)$ that finds initial data $\mathbf{x}_0, \mathbf{r}_0, \mathbf{U}_0, \mathbf{V}_0$ which in fact lives in \mathcal{M}_p . To achieve this, we set

$$\begin{aligned} \mathbf{x}_0 &:= \mathbf{x}_0^{(2)}, \\ \mathbf{r}_0 &:= \mathbf{r}_0^{(2)} \equiv \mathbf{b}^{(2)} - \mathbf{A}\mathbf{x}_0^{(2)}, \\ \mathbf{U}_0 &:= \hat{\mathbf{U}}, \\ \mathbf{V}_0 &:= \mathbf{A}\hat{\mathbf{U}}. \end{aligned}$$

Since $\text{range}(\mathbf{A}\hat{\mathbf{U}}) \subset \mathcal{G}_p$ and $\mathcal{G}_p \subset \mathcal{M}_p$ it follows that $\mathbf{r}_0 \in \mathcal{M}_p$ and $\text{range}(\mathbf{V}_0) \subset \mathcal{M}_p$; thus we have a feasible initialization.

Thus, in comparison to `IDRstab`, the initial data of `Mstab` lives not only in \mathcal{M}_0 but even in \mathcal{M}_p . This is advantageous because the dimension of \mathcal{M}_p is smaller than the dimension of \mathcal{M}_0 (which is comparable to the space \mathcal{G}_0). In particular, we can show the following result for the associated Petrov spaces of these \mathcal{M} -spaces.

LEMMA 2 (Petrov spaces of \mathcal{M} -spaces). *Consider the associated Petrov spaces $\{\mathcal{C}_j\}_{j \in \mathbb{N}_0} \subset \mathbb{C}^N$ to the \mathcal{M} -spaces $\{\mathcal{M}_j\}$:*

$$\mathcal{C}_j := \mathcal{M}_j^\perp \setminus (\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(2)}) + \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}))^\perp.$$

Let the requirement of Theorem 2 hold. Then

$$\dim(\mathcal{C}_j) \geq \begin{cases} j(s-1), & j \leq p, \\ p(s-1) + (j-p)s & \text{otherwise.} \end{cases}$$

Proof. The proof follows by induction over $j \in \mathbb{N}_0$, using $\dim(\mathcal{C}_0) = 0$ and the lower estimate for $\dim(\mathcal{M}_j) - \dim(\mathcal{M}_{j+1})$ from Theorem 2. \square

Since $\mathcal{M}(s)\text{stab}(\ell)$ does not start from \mathcal{M}_0 but from \mathcal{M}_p , the Petrov space does not start from \mathcal{C}_0 but rather from \mathcal{C}_p , so the initial Petrov space of $\mathcal{M}(s)\text{stab}(\ell)$ has $p(s-1)$ dimensions. The Petrov space of `IDR(s)stab`(ℓ) at iteration p had p s dimensions. Thus, a fraction $\vartheta = (s-1)/s$ of the overall Petrov dimensions available at that moment during `IDRstab` is recycled by `Mstab`, where s is a method parameter.

4. Practical choice of p . As explained in section 3.2, `Mstab` is initiated by calling `IDRstab` with an additional input parameter p , which determines the matrix $\hat{\mathbf{U}}$ defined in (3.2). As Lemma 2 shows, the larger p is, the larger the associated Petrov spaces are, which is beneficial. However, p cannot be arbitrarily large because \mathbf{U}_0 in the initialization (for both `IDRstab` and `Mstab`) must have full rank. Since the columns of $\hat{\mathbf{U}}$ are necessarily in the space $\mathbf{A}^{-1}\mathcal{G}_p$ and $\dim(\mathcal{G}_p) \leq \dim(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)})) - ps$, it follows that the requirement $\text{rank}(\hat{\mathbf{U}}) = s \Rightarrow \dim(\mathcal{G}_p) \geq s$ and the grade of $\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)})$ imply

$$(4.1) \quad p \leq \frac{\dim(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}))}{s} - 1.$$

Clearly, this cannot typically be a practical requirement, as it would necessitate an accurate prediction of $\nu := \dim(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}))$, the grade of the full Krylov subspace of the first linear system. In some practical settings we have observed that replacing ν by the number of matrix-vector products where `IDR(s)stab`(ℓ) for the first system

seems to be entering a superlinear phase, and taking p to be the upper bound in (4.1), yields a value (for p) that gives rise to a good performance of $\mathcal{M}(s)\text{stab}(\ell)$.

From a practical point of view, we need to have an algorithmic recipe for selecting p . We go about it in the following way. For a given threshold $\tau \in (0, 1)$, IDRstab is adapted so that p is the maximal integer value that satisfies

$$(4.2) \quad \|\mathbf{r}_p\|_2 / \|\mathbf{b}\|_2 \geq \tau.$$

Taking p as large as possible would maximize the dimension of $\mathcal{C}_0^{(2)}$, which is desirable for obtaining speed-up. At the same time, the threshold makes sure in a quantitative way that $\mathbf{r}_p \neq \mathbf{0} \Rightarrow \mathcal{G}_p \neq \{\mathbf{0}\}$ holds and thus (4.1) is satisfied. In practice, we select the value for p within IDRstab in a greedy fashion. We start with $p := 0$, $\hat{\mathbf{U}} := \mathbf{0}_{N \times s}$. Each time a new iterate $\mathbf{x}_j, \mathbf{r}_j, \mathbf{U}_j, \mathbf{V}_j$ satisfies the constraint (4.2) we overwrite $p := j$, $\hat{\mathbf{U}} := \mathbf{U}_j$. The threshold τ should not be chosen too small, to avoid a damaging effect of roundoff errors in $\hat{\mathbf{U}}$. Values in the range $[10^{-2}, 10^{-4}]$ seem reasonable.

In what follows we illustrate some theoretical and practical convergence properties of $\mathcal{M}(s)\text{stab}(\ell)$. Suppose we need to solve two linear systems in sequence. If the two corresponding Krylov spaces have the grades $\nu := \dim(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}))$, $\hat{\nu} := \dim(\mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(1)}) + \mathcal{K}_N(\mathbf{A}; \mathbf{r}_0^{(2)}))$, then $\text{IDR}(s)\text{stab}(\ell)$ terminates for the second system after at most $\lceil \hat{\nu}/s \rceil (s + 1)$ matrix-vector products. Choosing p that satisfies (4.1), $\mathcal{M}(s)\text{stab}(\ell)$ in contrast converges for the second system after at most

$$\left\lceil \frac{\hat{\nu} - p(s - 1)}{s} \right\rceil (s + 1)$$

matrix-vector products. For $p = \lfloor \nu/s - 1 \rfloor$ the expression can be bounded as

$$\left\lceil \frac{\hat{\nu} - p(s - 1)}{s} \right\rceil (s + 1) \leq \left\lceil \frac{\nu}{s(s - 1)} + \frac{\hat{\nu} - \nu}{s} \right\rceil (s + 1).$$

This is potentially interesting theoretically, because we are not aware of other Krylov-type methods that, when restricted to $\mathcal{O}(Ns)$ storage, are capable of terminating in $\mathcal{O}(N/s)$ matrix-vector products.

In the following experiments we show the influence of various choices of the parameter p on the convergence behavior of $\mathcal{M}\text{stab}$. The first experiment investigates the termination behavior, whereas the second experiment provides a showcase for superlinear convergence.

Example 4.1. We apply $\text{IDR}(4)\text{stab}(1)$ and $\mathcal{M}(4)\text{stab}(1)$ to solve two systems as in (1.2), where

$$(4.3) \quad \mathbf{A} = \text{tridiag}(2, 3, 1) \in \mathbb{R}^{100 \times 100}, \quad \mathbf{b}^{(1)} = \mathbf{e}_1, \quad \mathbf{b}^{(2)} = \mathbf{e}_2, \quad \mathbf{x}_0^{(1)} = \mathbf{x}_0^{(2)} = \mathbf{0}.$$

We solve $\mathbf{b}^{(1)}$ with GMRES and $\text{IDR}(4)\text{stab}(1)$; $\mathbf{b}^{(2)}$ is solved with $\mathcal{M}(4)\text{stab}(1)$ for three different values of p . In Figure 4.1 we show the results. The residuals \mathbf{r}_p are marked in the figure for each value of p by +, ×, □.

From GMRES's curve we see $\dim \mathcal{K}_N(\mathbf{A}; \mathbf{b}^{(1)}) = 100$, and thus $\text{IDR}(4)\text{stab}(1)$ terminates after 125 MV-s; cf. Theorem 1(b).

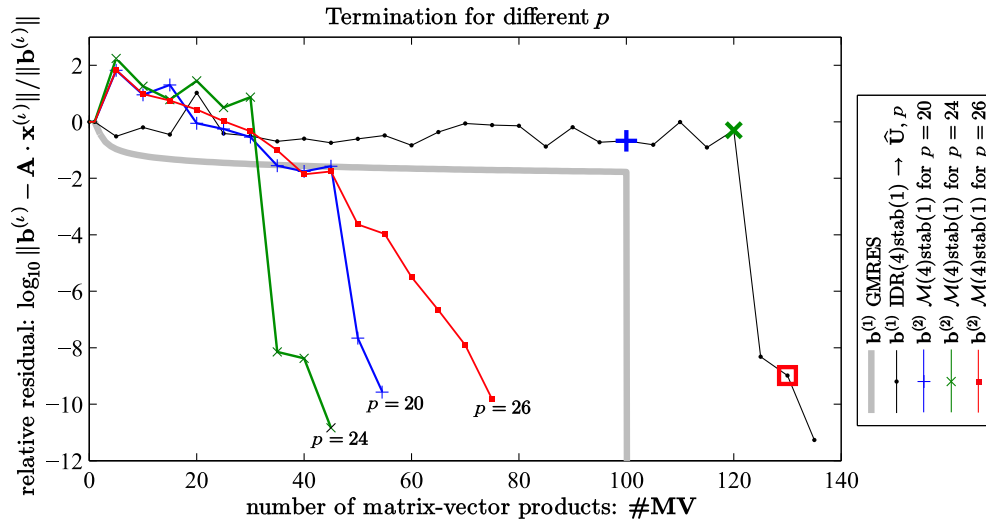


FIG. 4.1. Convergence graphs of GMRES, IDR(4)stab(1), and M(4)stab(1) for the sequence from Example 4.1 for different choices of p . The residuals \mathbf{r}_p in IDRstab’s convergence graph are marked by +, x, □.

For $p = 20$ Theorem 2(b) says that Mstab must terminate after 50 MV-s (because then the M-space has dimension zero), which it does. For $p = 24$ Theorem 2(b) guarantees convergence after at most 35 MV-s, which is better. $p = 24$ is the largest p that does not violate (4.1). For $p = 26$ performance degrades. The convergence curve of GMRES for $\mathbf{b}^{(2)}$ is roughly identical to that for $\mathbf{b}^{(1)}$.

From the example we see that larger values for p are likely to achieve earlier termination and that as a rule of thumb, a value of p that is large enough but at the same time complies with the upper bound (4.1) may give rise to a good convergence behavior. That said, in practice even if the optimal value of p is not determined precisely, nearby values still give rise to decent performance.

In contrast to the smooth convergence graph of GMRES, the residuals of IDRstab and Mstab converge in a nonmonotonic fashion. This is expected; it is inherited from BiCG-type methods. In comparing IDRstab to Mstab, we observe that the residual norm for the latter initially spikes. Eventually Mstab converges faster than IDRstab, but it takes a few iterations for the residual norm to settle. A possible explanation for this, in this particular instance, may be that since Mstab starts iterating from $\mathbf{r}_0 \in \mathcal{M}_p$, only the scalars $\omega_{p+1}, \omega_{p+2}, \dots$ are available, whereas for IDRstab the scalars $\omega_1, \omega_2, \dots, \omega_p, \omega_{p+1}, \omega_{p+2}, \dots$ can be chosen. Mstab then chooses $\omega_{p+1}, \omega_{p+2}, \dots$ to stabilize the residual norms, which compensates for the initial spike in subsequent iterations. We note, however, that in other numerical experiments we have not observed a similar initial spike of the residual norm for Mstab. Obtaining a bound on the residual norm for early iterations is desirable but would require further assumptions, since convergence theory for Mstab relies on convergence theory for BiCG-type methods in the general case.

Example 4.2. Let us consider the two linear systems

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_I & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{II} \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{pmatrix} \mathbf{b}_I \\ \mathbf{b}_{II} \end{pmatrix}, \quad \mathbf{b}^{(2)} \equiv \mathbf{b}^{(1)}, \quad \mathbf{x}_0^{(1)} = \mathbf{x}_0^{(2)} = \mathbf{0},$$

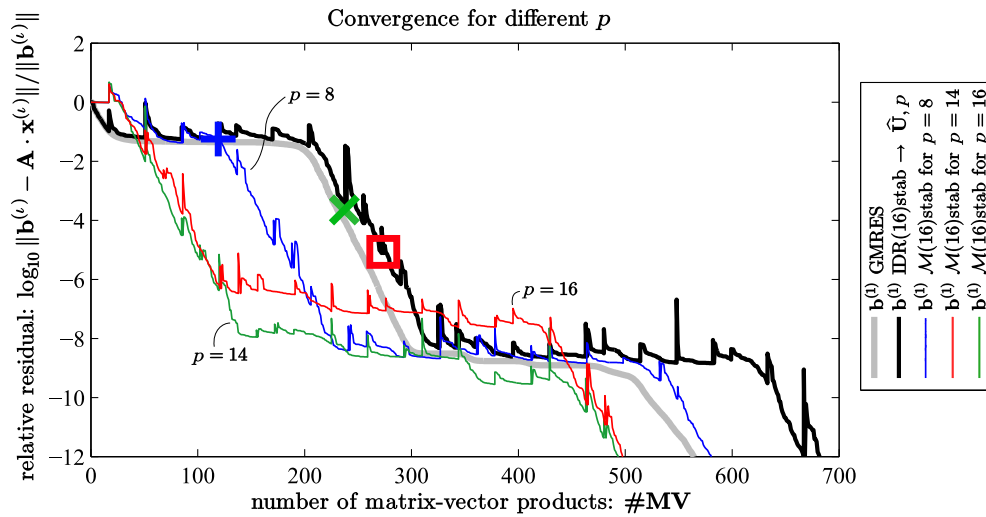


FIG. 4.2. Convergence graphs of GMRES, IDR(16)stab(2) and M(16)stab(2) for the sequence from Example 4.2 for different values of p . The residuals \mathbf{r}_p in IDRstab’s convergence graph are marked by +, \times , \square .

where $\mathbf{A}_I \in \mathbb{R}^{4096 \times 4096}$ and $\mathbf{b}_I \in \mathbb{R}^{4096}$ are from [22, section 6.2], which is a discretization of a two-dimensional (2D) convection-dominated elliptic PDE, and

$$\mathbf{A}_{II} = \text{tridiag}(2, 3, 1) \in \mathbb{R}^{3 \times 3}, \quad \mathbf{b}_{II} = \delta \mathbf{e}_1 \in \mathbb{R}^3.$$

In Figure 4.2 we consider this case for $\delta = 10^{-3}$. Even though the matrix is block diagonal with a small 3×3 second diagonal block and the blocks are obviously decoupled, the effect of the small parameter δ is detrimental and potentially global, because the optimal polynomial that is constructed throughout the iteration of a minimum residual solver (such as GMRES) attempts to resolve very small roots. This, in turn, has an effect on the overall speed of convergence; it is associated with spectral structure, independent of the second diagonal block size. As a result, stagnation occurs at a relative residual which we observe to be approximately $10^{-5} \delta$. We use that to model a stagnation phase and test the ability of our recycling method to overcome the stagnation and yield convergence.

In contrast to the previous example, where there was essentially no progress in residual norm reduction until the last iteration and the grade was equal to the dimension of the linear system, in this example we cannot easily find an estimate for the grade to apply (4.1). We therefore opt for the algorithmic choice of p that we have described in the discussion that followed (4.2). For τ between 10^{-2} and 10^{-5} we obtain values for p between 14 and 16.

In Figure 4.2 we see the convergence of Mstab for three different choices of p , namely, $p = 8, 14, 16$. The residuals \mathbf{r}_p are marked in the figure by +, \times , \square . We observe that for $p = 8$ the method yields superlinear convergence at a later iterate than for $p = 14$. For $p = 14$ and $p = 16$ the methods converge in a similar way. The method for $p = 14$ stagnates at a slightly smaller residual ($\approx 10^{-8}$) than the variant with $p = 16$ ($\approx 10^{-6}$).

5. Large-scale numerical experiments. We compare Mstab against the Krylov recycling methods RBiCGStab [1] and GCRO-DR [18]. We also include

TABLE 5.1

Computational cost of methods and parameters used in the numerical experiments: Number of DOTs and AXPYs (BLAS-1 operations) per matrix-vector product (MV). RBiCGStab is BiCGStab applied to (2.1) with $c = 20$ deflation vectors. The last column lays out how much storage in total the respective methods require in the number of vectors of length N .

Method	DOTs/MV	AXPYs/MV	Stored N -vectors
RBiCGStab ($c = 20$)	22	33	47
$\mathcal{M}(6)\text{stab}(2)$	27.8	29.4	29
$\mathcal{M}(8)\text{stab}(2)$	35.7	36.7	37
GCRO-DR(15, 10)	65.7	92.8	42
GCRO-DR(30, 20)	124.5	176.5	72

comparisons with the standard Krylov methods GMRES, PCG, IDRstab. Since those methods do not utilize a recycling strategy, we expect them to be outperformed by the recycling methods in the case of multiple right-hand sides. Our code is written in MATLAB, which has its limitations in terms of providing a precise picture of computational cost in large-scale settings but is nonetheless capable of generating useful evidence on the computational merits of the various methods that are tested.

For the iterative scheme in $\mathcal{M}\text{stab}$ and IDRstab we use the robust implementation [17, Algorithm 26] with $\ell \in \{1, 2\}$ adaptively.¹ The parameter choice $m = 30, k = 20$ for GCRO-DR is as recommended in [18]. Of this method we were provided with an implementation by the second author of [18]. In RBiCGStab we use $c = 20$ deflation vectors. We note that GCRO-DR and RBiCGStab are methods for a more general setting, namely, sequences where also the system matrix changes. We study the relative residuals with respect to the number of computed matrix-vector products ($\#\text{MV}$) and the computation times.

Table 5.1 shows the average computational cost and the required storage of each tested method. PCG and BiCGStab are cheaper per iteration than RBiCGStab, IDR(s)stab(ℓ) and $\mathcal{M}(s)\text{stab}(\ell)$. GCRO-DR entails additional orthogonalizations and requires an additional deflation basis, compared to $\mathcal{M}\text{stab}$. In general, for larger values of s, ℓ and m, k the methods require more computation time for each MV but potentially a smaller $\#\text{MV}$ in total.

In each test case we solve a sequence (1.1) with at least two right-hand sides $\mathbf{b}^{(1)}, \mathbf{b}^{(2)}$. As described at the beginning of section 3.2 and illustrated in Figure 3.1, IDR(s)stab(ℓ) generates $\hat{\mathbf{U}}$ when it solves the first system. Then $\mathcal{M}(s)\text{stab}(\ell)$ uses it when solving the subsequent linear systems. In each convergence figure the residual \mathbf{r}_p on IDRstab's convergence graph is marked by a circle. In all the experiments p is algorithmically chosen using the threshold $\tau = 10^{-3}$.

5.1. A 2D elliptic convection-diffusion-reaction problem. The following problem is from [22, section 6.4(b)], but with a finer mesh. We consider the unit square $\Omega = (0, 1)^2$ and apply a centered finite difference discretization of the following elliptic PDE on a Cartesian mesh of 350×350 interior points, resulting in a real nonsymmetric linear system of $N = 122,500$ equations:

$$\begin{aligned}
 -\Delta u + \frac{1000}{\sqrt{2}}(u_x + u_y) - 1000u &= f && \text{in } \Omega, \\
 u &= g && \text{on } \partial\Omega.
 \end{aligned}$$

¹In the legends of our figures we state the maximum value of ℓ that was used.

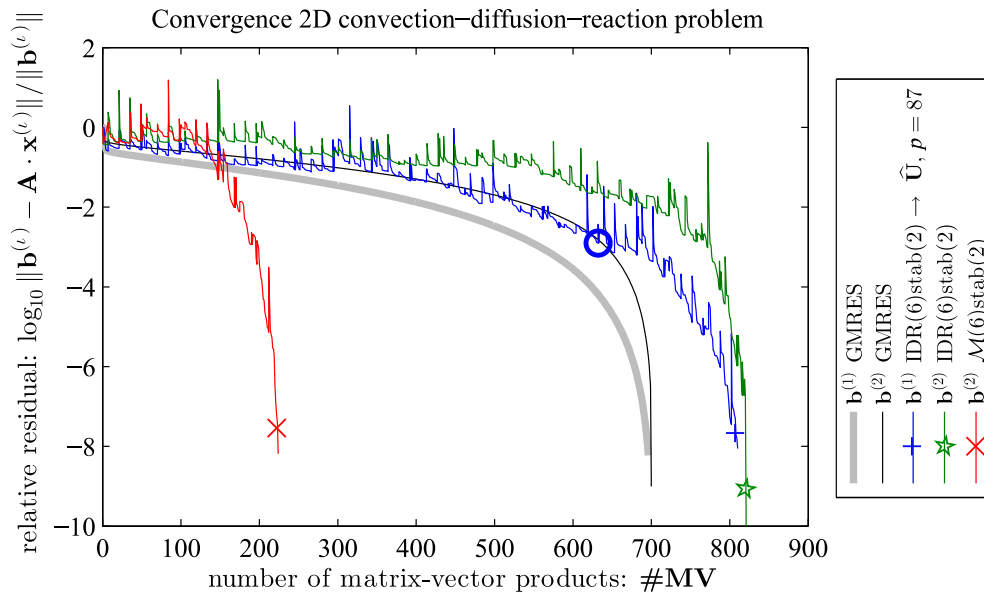


FIG. 5.1. Convergence graphs of GMRES, IDR(6)stab(2), and $\mathcal{M}(6)$ stab(2), showing the number of matrix-vector products versus normalized residuals for the 2D convection-diffusion-reaction problem. The circle signifies \mathbf{r}_p .

We consider a sequence of two right-hand sides $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ that are chosen such that the discrete solutions on the mesh points are, respectively,

$$\begin{aligned} u^{(1)}(x, y) &= \sin(x) \sin(y) \exp(x + y); \\ u^{(2)}(x, y) &= \sin(x + y) \exp(x + y). \end{aligned}$$

The initial guesses are $\mathbf{x}_0^{(1)} = \mathbf{0}$ and $\mathbf{x}_0^{(2)} = \alpha \mathbf{x}_0^{(1)}$, where α is a scalar so that $\|\mathbf{b}^{(2)} - \mathbf{A} \mathbf{x}_0^{(2)}\|_2$ is minimal; we say it is a *residual-optimal initial guess*. Choosing the second initial guess in this way is reasonable because it is in line with the spirit of modern iterative methods, which typically aim to reduce the norm of the residual.

The convergence graphs are shown in Figure 5.1. IDRstab takes approximately 88 seconds to solve each of the two linear systems. \mathcal{M} stab achieves a speed-up over IDRstab of a factor of approximately 3.6 in #MV and a similar factor in computation time: approximately 24 seconds.

5.2. A preconditioned 2D elliptic surface problem. In [27], a finite element discretization of an elliptic PDE model for the ocean of planet earth is presented. This discretization leads to one linear system of $N = 169,850$ equations for each month with a nonchanging mildly nonsymmetric system matrix and changing right-hand-side vectors. The right-hand sides arise from a time-dependent wind-field model. The triangulation uses uniform spherical coordinates, resulting in bad conditioning of the system matrix.

For preconditioning we use ILUTP with drop tolerance 10^{-2} . The ratio of non-zeros in the preconditioner over the system matrix is 1.62. We apply GMRES, BiCGStab, IDR(8)stab(2), RBiCGStab, and $\mathcal{M}(8)$ stab(2) to the split preconditioned system, where the permutation matrix is merged into the left factor. For the deflation

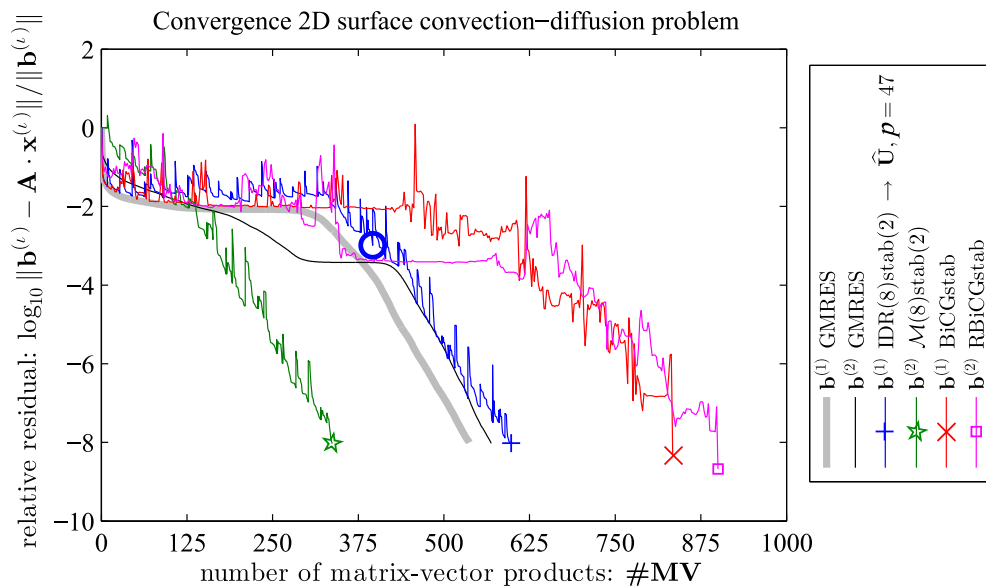


FIG. 5.2. Convergence graphs of GMRES, BiCGStab, RBiCGStab with $c = 20$ deflation vectors, IDR(8)stab(2), and \mathcal{M} (8)stab(2), showing the number of matrix-vector products versus normalized residuals for the preconditioned 2D surface problem. The circle signifies \mathbf{r}_p .

basis in RBiCGStab we use $c = 20$ Ritz vectors, which we compute from the absolute smallest eigenvalues of the Arnoldi basis of GMRES.

Figure 5.2 shows the numerical results. We observe that IDR(8)stab(2) is close to GMRES in terms of speed of convergence. \mathcal{M} (8)stab(2) yields a speed-up over IDRstab of 1.8 in #MV and computation time (approximately 68 second versus 127 seconds). The deflation in RBiCGStab initially yields a speed-up compared to BiCGStab. However, at a relative residual of 10^{-4} the method stagnates, and its overall computational time is approximately 230 seconds. The deflation introduces overhead (cf. Table 5.1) of ≥ 10 AXPYs and DOTs per MV, leading to a longer computation time in total. GCRO-DR(30, 20) did not converge for this problem.

5.3. A 3D parabolic convection-diffusion-reaction problem. This problem is an instationary extension of [22, section 6.2]. We solve on $\Omega = (0, 1)^3$ a discretization of the following parabolic PDE:

$$\begin{aligned} u_t - \Delta u + 1000(x u_x + y u_y + z u_z) + 10 u &= 1 && \text{in } \Omega \times (0, T), \\ u &= 0 && \text{on } \partial\Omega \times (0, T), \\ u(t = 0, \cdot) &= 0 && \text{in } \Omega. \end{aligned}$$

For the spatial discretization we construct finite volumes on a $65 \times 65 \times 65$ grid. The temporal discretization is done by implicit Euler with time step $\Delta t = 0.1$. This results in a sequence of nonsymmetric linear systems of $N = 262,144$ equations. $\mathbf{x}^{(l)}$ holds the temporal increment of the solution at the l th time step. Residual-optimal initial guesses are used.

Figure 5.3 shows the convergence results of GMRES, IDRstab, and the recycling methods \mathcal{M} stab and GCRO-DR for the first three systems. Both of the recycling methods improve upon GMRES and have a similar convergence behavior in terms of residual

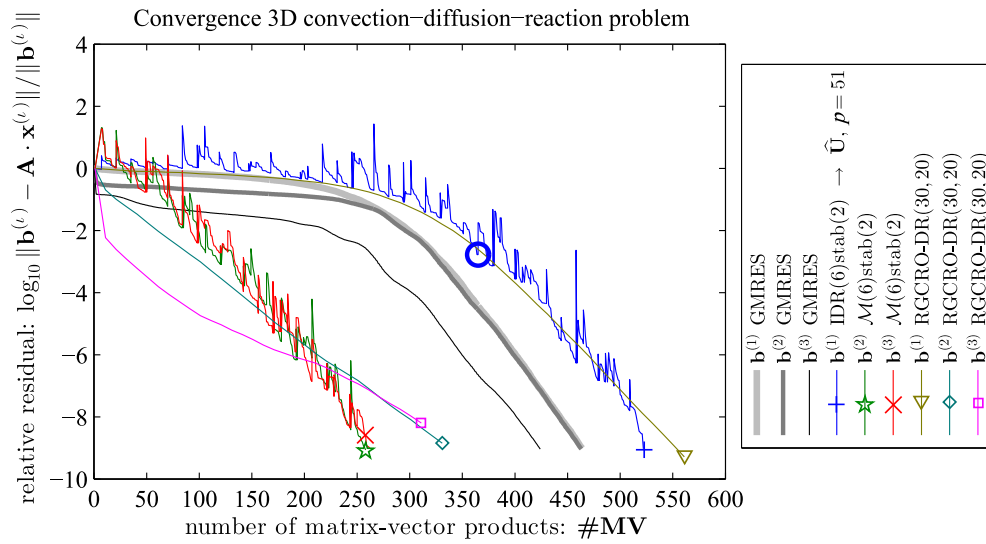


FIG. 5.3. Convergence graphs of GMRES, IDR(6)stab(2), M(6)stab(2), and GCRO-DR(30,20), showing the number of matrix-vector products versus normalized residuals for the 3D parabolic convection-diffusion-reaction problem with a positive shift. The circle signifies r_p .

reduction over the number of matrix-vector products. However, Mstab is faster in computation time because it entails less computational overhead; cf. Table 5.1. Indeed, in this case Mstab takes approximately 76 seconds to converge, whereas IDRstab takes approximately 166 seconds. GCRO-DR is significantly outperformed in this case in terms of computation times. The matrices in this example are very sparse (pentadiagonal), and thus #AXPY and #DOT play an important role in computation time; see Table 5.1. If the matrices were denser, then computation time would be dominated by #MV, which is similar for both methods. Consequently, for denser problems the overall performance of GCRO-DR and Mstab is likely to be similar.

5.4. A 2D parabolic diffusion problem. We discretize the following parabolic PDE, where Ω is shown in Figure 5.4:

$$\begin{aligned} u_t - \Delta u &= 1 && \text{in } \Omega \times (0, T), \\ u &= 0 && \text{on } \partial\Omega \times (0, T), \\ u(t = 0, \cdot) &= 0 && \text{in } \Omega. \end{aligned}$$

For the temporal discretization we choose the implicit Euler method with time step $\Delta t = 0.1$. For the spatial discretization we use piecewise linear finite elements on the tetrahedral mesh given in Figure 5.4. The mesh has $N = 26,398$ nodes and 53,510 finite elements. The systems are preconditioned by incomplete Cholesky factorization with fill-in of first level. Residual-optimal initial guesses are used.

The linear systems here are symmetric positive definite, and it may be natural to restrict our attention to PCG and its variants. But since Mstab is a short-term recurrence method, too, its ability to recycle data makes it superior in this case to PCG. In Figure 5.5 we observe that Mstab yields convergence within about half of the number of iterations of PCG for all but the first right-hand side.

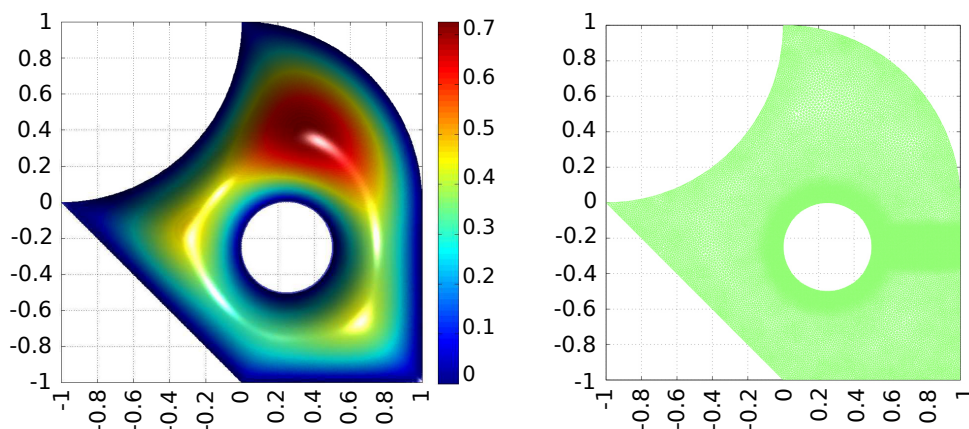


FIG. 5.4. Scaled stationary solution (left) and tetrahedral mesh (right) for the finite elements discretization of the parabolic diffusion problem.

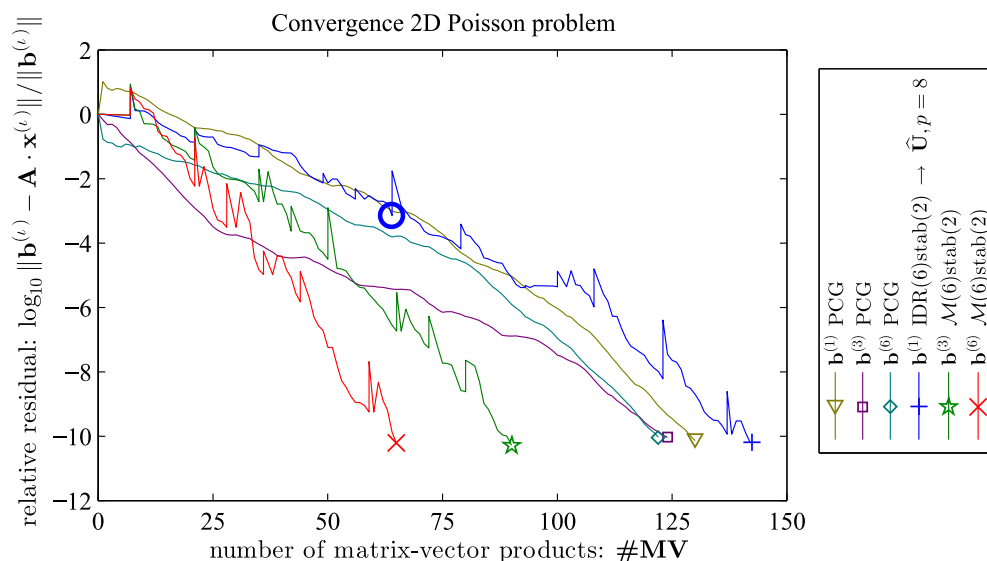


FIG. 5.5. Convergence graphs of PCG, IDR(6)stab(2), and M(6)stab(2), showing the number of matrix-vector products versus normalized residuals for the 2D parabolic diffusion problem. The circle signifies r_p .

6. Concluding remarks. \mathcal{Mstab} is a new IDR-type method that is based on a generalized theory of IDR. We have presented theoretical results that provide some insight into the merits of this method and presented test cases in which the method converges faster than competing methods.

\mathcal{Mstab} of degree ℓ inherits the properties of BiCGStab(ℓ). Recent advances in the theoretical and experimental understanding of BiCGStab(ℓ) and IDRstab indicate that these methods can handle strongly nonsymmetric systems very well, and we observe that \mathcal{Mstab} has similar properties. Our numerical experiments indicate that for sparse and highly nonsymmetric problems arising from discretized elliptic and parabolic PDE with convection terms, the approach taken in this paper is effective.

There are a few possible directions for future work. The choice of the parameter p requires careful attention, and it is desirable to provide additional evidence that the method is not overly sensitive to it. Another question to investigate is the ability of \mathcal{Mstab} to handle strongly indefinite linear systems. We suspect that the answer to this question is tied to robustness of bi-Lanczos methods at large, and the key may be the design of robust and highly efficient preconditioners.

In contrast to \mathcal{Mstab} , GCRO-DR has the advantage that it is suitable for treating the more general problem where not only the right-hand sides but also the system matrices change. It would be highly desirable to adapt \mathcal{Mstab} to situations not currently covered.

Acknowledgments. We thank Martin van Gijzen for helpful comments, for pointing out a few useful references, and for providing us with a larger version of the numerical experiment in section 5.2. The first author would like to thank Eric de Sturler for sharing and discussing his GCRO-DR implementation during the preparation of an earlier version of this manuscript. Finally, we thank Karen Willcox for her helpful editorial advice.

REFERENCES

- [1] K. AHUJA, P. BENNER, E. DE STURLER, AND L. FENG, *Recycling BiCGSTAB with an application to parametric model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. S429–S446.
- [2] K. AHUJA, E. DE STURLER, S. GUGERCIN, AND E. R. CHANG, *Recycling BiCG with an application to model reduction*, SIAM J. Sci. Comput., 34 (2012), pp. A1925–A1949.
- [3] K. AIHARA, K. ABE, AND E. ISHIWATA, *A variant of IDRstab with reliable update strategies for solving sparse linear systems*, J. Comput. Appl. Math., 259 (2014), pp. 244–258.
- [4] K. AIHARA, K. ABE, AND E. ISHIWATA, *Preconditioned IDRStab algorithms for solving non-symmetric linear systems*, IAENG Int. J. Appl. Math., 45, 2015.
- [5] A. AMRITKAR, E. DE STURLER, K. ŚWIRYDOWICZ, D. TAFTI, AND K. AHUJA, *Recycling Krylov subspaces for CFD applications and a new hybrid recycling solver*, J. Comput. Phys., 303 (2015), pp. 222–237.
- [6] P. BENNER AND L. FENG, *Recycling Krylov subspaces for solving linear systems with successively changing right-hand sides arising in model reduction*, in Model Reduction for Circuit Simulation, Lect. Notes Electr. Eng. 74, Springer, New York, 2011, pp. 125–140.
- [7] J. BOLTEN, N. BOZOVIC, AND A. FROMMER, *Preconditioning of Krylov-subspace methods using recycling in lattice QCD computations*, Proc. Appl. Math. Mech., 13 (2013), pp. 413–414.
- [8] E. DE STURLER, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.
- [9] L. S. DUARTE, W. CELES, A. PEREIRA, I. F. M. MENEZES, AND G. H. PAULINO, *PolyTop++: An efficient alternative for serial and parallel topology optimization on CPUs & GPUs*, Struct. Multidiscip. Optim., 52 (2015), pp. 845–859.
- [10] 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.
- [11] M. H. GUTKNECHT, *IDR explained*, Electron. Trans. Numer. Anal., 36 (2009/10), pp. 126–148.
- [12] M. H. GUTKNECHT, *Deflated and augmented Krylov subspace methods: A framework for deflated BiCG and related solvers*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1444–1466.
- [13] M. E. KILMER AND E. DE STURLER, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.
- [14] M. MILTENBERGER, *Die IDR(s)-Methode zur Lösung von parametrisierten Gleichungssystemen*, Diploma thesis, Technische Universität Berlin, Fakultät II - Naturwissenschaften und Mathematik, Institut für Mathematik, Berlin, Germany, 2009.
- [15] K. MOHAMED, S. NADARAJAH, AND M. PARASCHIVOIU, *Krylov recycling techniques for unsteady simulation of turbulent aerodynamic flows*, in Proceedings of the 26th International Congress of the Aeronautical Sciences, 2008.
- [16] R. B. MORGAN, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
- [17] M. P. NEUENHOFEN, *A Restarted GMRES-Based Implementation of IDR(s)stab(L) to Yield Higher Robustness*, Master thesis, RWTH Aachen University, Aachen Institute for Advanced Study in Computational Engineering Science, Aachen, Germany, 2017.

- [18] 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.
- [19] 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.
- [20] V. SIMONCINI AND D. B. SZYLD, *Interpreting IDR as a Petrov-Galerkin method*, SIAM J. Sci. Comput., 32 (2010), pp. 1898–1912.
- [21] G. L. G. SLEIJPEN, P. SONNEVELD, AND M. B. VAN GIJZEN, *Bi-CGSTAB as an induced dimension reduction method*, Appl. Numer. Math., 60 (2010), pp. 1100–1114.
- [22] G. L. G. SLEIJPEN AND M. B. VAN GIJZEN, *Exploiting BiCGstab(ℓ) strategies to induce dimension reduction*, SIAM J. Sci. Comput., 32 (2010), pp. 2687–2709.
- [23] P. SONNEVELD AND M. B. VAN GIJZEN, *IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Comput., 31 (2009), pp. 1035–1062.
- [24] 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.
- [25] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
- [26] M. B. VAN GIJZEN AND P. SONNEVELD, *An elegant IDR(s) variant that efficiently exploits biorthogonality properties*, ACM Trans. Math. Software, 38 (2011), 19.
- [27] M. B. VAN GIJZEN, C. B. VREUGDENHIL, AND H. OKSUZOGLU, *The finite element discretization for stream-function problems on multiply connected domains*, J. Comput. Phys., 140 (1998), pp. 30–46.
- [28] P. WESSELING AND P. SONNEVELD, *Numerical experiments with a multiple grid and a preconditioned Lanczos type method*, in Approximation Methods for Navier-Stokes Problems, Lecture Notes in Math. 771, Springer, Berlin, 1980, pp. 543–562.
- [29] Z. YE, Z. ZHU, AND J. R. PHILLIPS, *Generalized Krylov recycling methods for solution of multiple related linear equation systems in electromagnetic analysis*, in Proceedings of the Design Automation Conference, 2008, pp. 682–687.