SeMA (2017) 74:213–231
DOI 10.1007/s40324-016-0088-7

GMRES with multiple preconditioners

Chen Greif[1] · **Tyrone Rees**[2] · **Daniel B. Szyld**[3]

Received: 9 March 2016 / Accepted: 30 May 2016 / Published online: 17 June 2016

Abstract We propose a variant of GMRES, where multiple (two or more) preconditioners are applied simultaneously, while maintaining minimal residual optimality properties. To accomplish this, a block version of Flexible GMRES is used, but instead of considering blocks associated with multiple right hand sides, we consider a single right-hand side and grow the space by applying each of the preconditioners to all current search directions, minimizing the residual norm over the resulting larger subspace. To alleviate the difficulty of rapidly increasing storage requirements, we present a heuristic limited-memory selective algorithm, and demonstrate the effectiveness of this approach.

Keywords Iterative methods · Linear systems · Preconditioning · Krylov subspaces · GMRES

Mathematics Subject Classification 65F10 · 65N22 · 15A06

The work of Chen Greif was supported in part by Discovery Grant 261539 from the Natural Sciences and Engineering Research Council of Canada (NSERC). The work of Tyrone Rees was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The work of Daniel B. Szyld was supported in part by the U.S. National Science Foundation under grant DMS-1418882.

✉ Daniel B. Szyld
szyld@temple.edu

Chen Greif
greif@cs.ubc.ca

Tyrone Rees
tyrone.rees@stfc.ac.uk

[1] Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada

[2] Department of Scientific Computing, STFC Rutherford Appleton Laboratory, Chilton Didcot, Oxfordshire OX11 0QX, UK

[3] Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, PA 19122-6094, USA

1 Introduction

We are interested in the iterative solution of a linear system of the form

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \tag{1.1}$$

where $\mathcal{A} \in \mathbb{R}^{n\times n}$ is a large and sparse, possibly nonsymmetric or indefinite, matrix. We would like to use a modern iterative method, such as those associated with a Krylov subspace [34,43], and in particular GMRES [35]. One ingredient in the successful application of these methods is the use of a (nonsingular) preconditioner, $\mathcal{P}$. Here we consider only right preconditioning, i.e., we consider the equivalent system

$$\mathcal{A}\mathcal{P}^{-1}\mathbf{u} = \mathbf{b}, \tag{1.2}$$

where $\mathbf{u} = \mathcal{P}\mathbf{x}$.

It is common to have two or more different candidate preconditioners for the same linear system, each possessing different properties. For example, in the case of saddle-point problems, block diagonal preconditioners and constraint preconditioners provide such a choice; see, e.g., [4,21,23,29,39]. What we propose in this paper is a way to use more than one preconditioner simultaneously. We accomplish this by employing what one may think of as a block version of Flexible GMRES (FGMRES) [33], whereby at each step we add to the space multiple directions based on the application of all the preconditioned operators. The new iterate is optimal in this subspace in the minimum residual least-squares sense. We mention that Calandra et al. [7] have recently described a block version of FGMRES which—while sharing some similarities with the work described here—differs both in scope and details with the GMRES algorithm with multiple preconditioners we propose.

Block methods are used either for the solution of linear systems with multiple right hand sides, or to enrich the space with additional directions; see, e.g., the pioneering paper [24], the survey [16], or [9,30], and references therein. One of the new ingredients in our proposed technique is the use of different preconditioning directions for each component of the block method.

Another element that distinguishes our proposal from standard block methods is that, since all preconditioners are applied to the existing basis at each step, the dimension of the subspace may increase very rapidly. This has the advantage of generating a very rich space (with information coming from all the preconditioners) where the solution is sought; indeed, our method might yield significantly faster convergence by combining the preconditioners compared to methods that force the user to pick just one direction (or even one at a time). On the other hand, this approach has the disadvantage of excessive memory requirements. Thus, we consider also a heuristic variant where a selective set of directions from this rich space are used. We mention that there are cases where the selective and the complete versions coincide, i.e., the problems are such that the growth in storage is only linear; see, e.g., Sect. 2.5 and [2].

While the new GMRES with multiple preconditioners (MPGMRES) we propose here (both in its complete and selective versions) can be used with any number of preconditioners, we expect that it will be most useful with a small number of preconditioners, typically two, and our study is in part driven by this consideration.

We present numerical experiments in practical situations where our method with multiple preconditioners converges faster than alternative formulations involving GMRES with any one of the preconditioners alone, and faster than FGMRES where one simply cycles through the available preconditioners, as done in [32] for a particular application.

This work is inspired in part by previous work involving one of the authors [6], where such multiple preconditioning is used for the conjugate gradient method (CG) for symmetric positive definite linear systems. The algorithm in [6] is called *multipreconditioned conjugate gradients* (MPCG), and it combines the preconditioners while aiming to preserve the optimality criterion of minimizing the energy norm of the error. However, a drawback of MPCG is that, while being designed for symmetric positive definite matrices, it does not in general retain the short-term recurrence of CG. By contrast, since GMRES for general non-symmetric matrices does not use short recurrences, their absence in the method discussed in this paper is not a problem.

The remainder of the paper is organized as follows. In Sect. 2 we derive and study our proposed MPGMRES algorithm. We develop a complete algorithm and then present a selective version of it. In Sect. 3 we give a few details on the computational cost and implementation issues, including pointers to our software, which is available for the interested readers. In Sect. 4 we present some numerical experiments illustrating the effectiveness of the proposed method.

2 MPGMRES

We start this section with a brief description of the GMRES algorithm and its flexible version. We do this in part to fix the notation and establish some concepts which we then use to construct GMRES with multiple preconditioners (MPGMRES). We then move on to discuss the complete and selective MPGMRES algorithms and some of their properties.

2.1 GMRES and flexible GMRES

GMRES is often the solution method of choice for non-symmetric linear systems of the form (1.1). Starting from some initial vector $\mathbf{x}^{(0)}$, and the corresponding initial residual $\mathbf{r}^{(0)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(0)}$, the k^{th} step of GMRES consists of computing the vector $\mathbf{x}^{(k)}$ in the space $\mathbf{x}^{(0)} + \mathcal{K}_k(\mathcal{A}, \mathbf{r}^{(0)})$, where

$$\mathcal{K}_k(\mathcal{A}, \mathbf{r}^{(0)}) := \text{span}\{\mathbf{r}^{(0)}, \mathcal{A}\mathbf{r}^{(0)}, \ldots, \mathcal{A}^{k-1}\mathbf{r}^{(0)}\} = \{p(\mathcal{A})\mathbf{r}^{(0)}, \deg p < k\}$$

is the standard Krylov subspace (p here denotes a polynomial), such that the residual corresponding to $\mathbf{x}^{(k)}$ has the minimal norm among all vectors in $\mathbf{x}^{(0)} + \mathcal{K}_k(\mathcal{A}, \mathbf{r}^{(0)})$. Consider the preconditioned system (1.2); preconditioned GMRES finds $\mathbf{u}^{(k)}$ which minimizes the 2-norm of the residual over $\mathbf{u}^{(0)} + \mathcal{K}_k(\mathcal{A}\mathcal{P}^{-1}, \mathbf{r}^{(0)})$. Written in terms of $\mathbf{x}$, since $\mathbf{x} = \mathcal{P}^{-1}\mathbf{u}$, we therefore find

$$\begin{aligned} \mathbf{x}^{(k)} &\in \mathbf{x}^{(0)} + \mathcal{P}^{-1}\mathcal{K}_k(\mathcal{A}\mathcal{P}^{-1}, \mathbf{r}^{(0)}) \\ &\in \mathbf{x}^{(0)} + \mathcal{K}_k(\mathcal{P}^{-1}\mathcal{A}, \mathcal{P}^{-1}\mathbf{r}^{(0)}). \end{aligned} \tag{2.1}$$

Note that the latter space is the same as in left-preconditioned GMRES, but when using left or right preconditioning, the functional which is minimized is different; see, e.g., [34, p. 272], [38].

An orthonormal basis for the Krylov subspace $\mathcal{K}_k(\mathcal{A}\mathcal{P}^{-1}, \mathbf{r}^{(0)})$ is computed using the Arnoldi algorithm. This generates a decomposition of the form

$$\mathcal{A}Z_k := \mathcal{A}\mathcal{P}^{-1}V_k = V_{k+1}\widetilde{H}_k,$$

where $V_k \in \mathbb{R}^{n\times k}$ has orthonormal columns with the first being $\mathbf{r}^{(0)}/\|\mathbf{r}^{(0)}\|_2$, and $\widetilde{H}_k \in \mathbb{R}^{(k+1)\times k}$ is upper Hessenberg. Since the columns of V_k span the space $\mathcal{K}_k(\mathcal{A}\mathcal{P}^{-1}, \mathbf{r}^{(0)})$, the iterate $\mathbf{x}^{(k)}$ therefore must have the form

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathcal{P}^{-1}V_k\mathbf{y}^{(k)} = \mathbf{x}^{(0)} + Z_k\mathbf{y}^{(k)}$$

for some vector $\mathbf{y}^{(k)} \in \mathbb{R}^k$. GMRES finds this vector $\mathbf{y}^{(k)}$ by solving a least-squares problem

$$\min_{\mathbf{y}\in\mathbb{R}^k} \|\|\mathbf{r}^{(0)}\|_2\mathbf{e}_1 - \widetilde{H}_k\mathbf{y}\|_2, \tag{2.2}$$

since $\|\mathbf{b} - \mathcal{A}\mathbf{x}^{(k)}\|_2 = \|\|\mathbf{r}^{(0)}\|_2\mathbf{e}_1 - \widetilde{H}_k\mathbf{y}^{(k)}\|_2$.

The Flexible GMRES method of Saad [33] allows us to use a different preconditioner at each iteration.The key idea of FGMRES is to store the application of the preconditioner at the j^{th} step, $\mathcal{P}_j^{-1}v_j$, in the j^{th} column of Z_k, so that we still have

$$\mathcal{A}Z_k = V_{k+1}\widetilde{H}_k, \tag{2.3}$$

and the same least-squares problem (2.2) is solved. Thus, the approximation at the kth step is still in $\mathbf{x}^{(0)} + \mathcal{R}(Z_k)$, where $\mathcal{R}(\cdot)$ denotes the range of a linear operator. Note that $\mathcal{R}(Z_k)$ is not, strictly speaking, a Krylov subspace, but it is nonetheless the space where the approximation is sought [42,43]. We also note that in FGMRES, one usually stores both V_k and Z_k, i.e., effectively doubling the storage requirements of GMRES.

2.2 Derivation of MPGMRES

Suppose that instead of just a single preconditioner we have t (nonsingular) preconditioners, $\mathcal{P}_1, \ldots, \mathcal{P}_t$, $t \geq 2$. We have in mind $t = 2$ for most applications, and to simplify notation, we will in the sequel occasionally discuss certain issues specifically for two preconditioners. For a fixed preconditioner, say $\mathcal{P}_j$, the GMRES algorithm would commence with the product $\mathcal{P}_j^{-1}\mathbf{r}^{(0)}$ as in (2.1). In our proposed method, we collect all these vectors in

$$Z^{(1)} = [\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \ldots, \mathcal{P}_t^{-1}\mathbf{r}^{(0)}] \in \mathbb{R}^{n\times t},$$

and consider the first iterate $\mathbf{x}^{(1)}$, which minimizes the residual norm of vectors over

$$\mathbf{x}^{(0)} + \text{span}\{\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \ldots, \mathcal{P}_t^{-1}\mathbf{r}^{(0)}\}. \tag{2.4}$$

We can therefore define the first iterate of a new method as

$$\mathbf{x}^{(1)} := \mathbf{x}^{(0)} + Z^{(1)}\mathbf{y}^{(1)},$$

where the vector $\mathbf{y}^{(1)} \in \mathbb{R}^t$ is chosen to minimize the residual in the 2-norm. Thus, $\mathbf{x}^{(1)}$ is the best approximation which incorporates information from all preconditioners. It follows from (2.4) that

$$\mathbf{r}^{(1)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(1)} \in \mathbf{r}^{(0)} + \text{span}\left\{\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \ldots, \mathcal{A}\mathcal{P}_t^{-1}\mathbf{r}^{(0)}\right\}. \tag{2.5}$$

Our proposed method is in part based on block GMRES, which was first introduced in [47] for multiple right hand sides. It was studied, e.g., in [9,22,30,41], and applied recently, e.g., in [7]; see also [16] and references therein. We also mention block Arnoldi as a necessary important ingredient; see, e.g., [36,37], [34, §6.12].

In our Arnoldi-type block procedure to obtain an orthonormal basis of the search space, we start by orthogonalizing every column of $W := \mathcal{A}Z^{(1)}$ with respect to $V^{(1)} := \mathbf{r}^{(0)}/\|\mathbf{r}^{(0)}\|_2$,

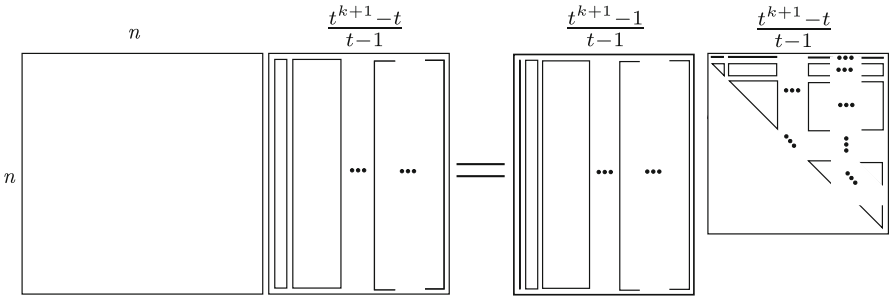


Fig. 1 Schematic diagram of the Arnoldi-type decomposition (2.7)

and among themselves (using a reduced QR factorization), and storing the coefficients in the matrices $H^{(j,1)}$, $j = 1, 2$, which are part of the upper Hessenberg matrix $\widetilde{H}_k$ (see Fig. 1); thus obtaining $V^{(2)}$. Then we increase the space by applying the multiple preconditioners, i.e., at step k, compute

$$Z^{(k)} = [\mathcal{P}_1^{-1} V^{(k)} \dots \mathcal{P}_t^{-1} V^{(k)}], \tag{2.6}$$

and repeat the process. As with block Arnoldi, we obtain the relation

$$\mathcal{A}\widetilde{Z}_k = \widetilde{V}_{k+1}\widetilde{H}_k, \tag{2.7}$$

where

$$\widetilde{Z}_k = \left[\, Z^{(1)} \,\dots\, Z^{(k)} \,\right], \qquad \widetilde{V}_{k+1} = \left[\, V^{(1)} \,\dots\, V^{(k+1)} \,\right]$$

and

$$\widetilde{H}_k = \begin{bmatrix} H^{(1,1)} & H^{(1,2)} & \cdots & H^{(1,k)} \\ H^{(2,1)} & H^{(2,2)} & & H^{(2,k)} \\ & \ddots & & \vdots \\ & & H^{(k,k-1)} & H^{(k,k)} \\ & & & H^{(k+1,k)} \end{bmatrix}.$$

We have implicitly assumed that $Z^{(k)}$ and $\widetilde{Z}_k$ have full rank. We keep this assumption throughout this section and address the situation when this assumption fails to hold, and deflation is needed, in Sect. 3.

We call the columns of $Z^{(k)}$ *search directions*, and the columns of $V^{(k)}$ *basis vectors*.

Observe that $Z^{(1)}$ and $V^{(2)}$ have t columns, while $Z^{(2)}$ and $V^{(3)}$ have t^2 columns, and in general $Z^{(i)}$ and $V^{(i+1)}$ have t^i columns. Therefore $\widetilde{V}_{k+1}$ has

$$\tau_k := \sum_{i=0}^{k} t^i = \frac{t^{k+1} - 1}{t - 1} \tag{2.8}$$

columns, while $\widetilde{Z}_k$ has $\tau_k - 1 = (t^{k+1} - t)/(t - 1)$ columns. Thus, in the common case of $t = 2$ the dimension of the search space is $2(2^k - 1)$. Note also that $H^{(1,i)} \in \mathbb{R}^{1\times t}$ for all i, and since the blocks $H^{(i+1,i)}$ on the sub-diagonal come from the QR factorization, they are all upper triangular. Therefore, as in this Arnoldi-type algorithm, the matrix $\widetilde{H}_k$ above is upper Hessenberg, here of order $(\tau_k - 1) \times \tau_k$. Figure 1 is a schematic diagram of the Arnoldi-type decomposition (2.7) showing the dimensions of the matrices involved.

As in block GMRES,since the columns of the basis matrix $\widetilde{V}_{k+1}$ are orthogonal, we have

$$\arg \min_{\hat{\mathbf{x}} \in \mathbf{x}^{(0)} + \mathcal{R}(\widetilde{Z}_k)} \|\mathbf{b} - \mathcal{A}\hat{\mathbf{x}}\|_2 = \arg \min_{\mathbf{y}} \|\beta \mathbf{e}_1 - \widetilde{H}_k \mathbf{y}\|_2, \tag{2.9}$$

where we have used $\hat{\mathbf{x}} = \mathbf{x}^{(0)} + \widetilde{Z}_k \mathbf{y}$ and $\beta = \|\mathbf{r}^{(0)}\|_2$. The proposed method, complete MPGMRES, is given as Algorithm 1.

Algorithm 1 Complete MPGMRES

Choose $\mathbf{x}^{(0)}$, $\mathbf{r}^{(0)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(0)}$
$\beta = \|\mathbf{r}^{(0)}\|$, $V^{(1)} = \mathbf{r}^{(0)}/\beta$
for $k = 1, \ldots,$ **until** convergence **do**
  $Z^{(k)} = [\mathcal{P}_1^{-1} V^{(k)} \ldots \mathcal{P}_t^{-1} V^{(k)}]$
  $W = \mathcal{A} Z^{(k)}$
  for $j = 1, \ldots, k$ **do**
    $H^{(j,k)} = (V^{(j)})^T W$
    $W = W - V^{(j)} H^{(j,k)}$
  end for
  $W = V^{(k+1)} H^{(k+1,k)}$ (reduced QR factorization)
  $\mathbf{y}^{(k)} = \text{argmin} \|\beta \mathbf{e}_1 - \widetilde{H}_k \mathbf{y}\|_2$
  $\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + [Z^{(1)} \ldots Z^{(k)}] \mathbf{y}^{(k)}$
end for

Note that Algorithm 1 contains the variant of block Arnoldi and reduced QR factorizations. Both of these algorithms can be thought of in terms of Gram-Schmidt orthogonalization, and we can perform the QR step explicitly in an Arnoldi-style algorithm. This is known as the band Arnoldi algorithm, and was first proposed (for the block-Lanczos case) by Ruhe [31]; see, e.g., [12, §6], [16, §9] [34, p. 209], for the block-Arnoldi case. In our experience taking advantage of blocking techniques (e.g., by using Level 3 BLAS calls), rather than the vector-based calculations needed in a Ruhe-style implementation, make Algorithm 1 the most efficient style of implementation. On the other hand, an implementation of the Ruhe version may mitigate the effects of dealing with loss of rank, which we discuss in some detail in Sect. 3.

2.3 The search and residual spaces

We describe the spaces where the iterates, and more importantly, the associated residuals computed in Algorithm 1 reside. We illustrate first the case $t = 2$. The first two iterates satisfy

$$\begin{aligned}
\mathbf{x}^{(1)} - \mathbf{x}^{(0)} &\in \text{span}\{\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{P}_2^{-1}\mathbf{r}^{(0)}\};\\
\mathbf{x}^{(2)} - \mathbf{x}^{(0)} &\in \text{span}\{\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{P}_2^{-1}\mathbf{r}^{(0)}, \mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)},\\
&\qquad \mathcal{P}_2^{-1}\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{P}_2^{-1}\mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)}\},
\end{aligned}$$

and the rest follows the same pattern. It follows (cf. (2.5)) that

$$\begin{aligned}
\mathbf{r}^{(1)} - \mathbf{r}^{(0)} &\in \text{span}\{\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)}\};\\
\mathbf{r}^{(2)} - \mathbf{r}^{(0)} &\in \text{span}\{\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)}, \mathcal{A}\mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{A}\mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)},\\
&\qquad \mathcal{A}\mathcal{P}_2^{-1}\mathcal{A}\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{A}\mathcal{P}_2^{-1}\mathcal{A}\mathcal{P}_2^{-1}\mathbf{r}^{(0)}\},\\
&= \{p(\mathcal{A}\mathcal{P}_1^{-1}, \mathcal{A}\mathcal{P}_2^{-1})\mathbf{r}^{(0)}\},
\end{aligned}$$

where $p = p(z_1, z_2)$ is a multivariate second degree polynomial in two (non-commuting) variables with $p(0, 0) = 1$.

Using the same observation, it is not hard to see that in the case of t preconditioners,

$$\begin{aligned}\mathbf{r}^{(k)} &= \mathbf{r}^{(0)} + p(\mathcal{A}\mathcal{P}_1^{-1}, \ldots, \mathcal{A}\mathcal{P}_t^{-1})\mathbf{r}^{(0)} \\ &\in \mathcal{R}(AZ^{(k)}) = \mathcal{R}(V^{(k+1)}) = \{p(\mathcal{A}\mathcal{P}_1^{-1}, \ldots, \mathcal{A}\mathcal{P}_t^{-1})\mathbf{r}^{(0)}, p(0, \ldots, 0) = 1\},\end{aligned}$$

where $p = p(z_1, \ldots, z_t)$ is a multivariate polynomial of degree k in t (non-commuting) variables. We call this space $\mathbb{P}_k = \mathbb{P}_k[z_1, \ldots, z_t]$. Note that what makes this space so rich is the presence of not only the powers of $\mathcal{A}\mathcal{P}_j^{-1}$, but also the cross terms, say of the form $\mathcal{A}\mathcal{P}_i^{-1}\mathcal{A}\mathcal{P}_j^{-1}$.

Therefore, from (2.9), we have that

$$\|\mathbf{r}^{(k)}\| = \min_{\substack{p \in \mathbb{P}_k[z_1, \ldots, z_t] \\ p(0, \ldots, 0) = 1}} \|p(\mathcal{A}\mathcal{P}_1^{-1}, \ldots, \mathcal{A}\mathcal{P}_t^{-1})\mathbf{r}^{(0)}\|. \tag{2.10}$$

Notice that (2.10) reduces back to the standard minimal residual characterization of GMRES in the case of a single preconditioner ($t = 1$).

It follows from (2.10) that the known GMRES convergence bounds can be generalized to this situation, e.g., by replacing the univariate polynomial bounds with those using multivariate polynomials (on the spectra of each preconditioned matrix $\mathcal{A}\mathcal{P}_j^{-1}$).

2.4 Selective MPGMRES

As we saw in Sect. 2.2, since $Z^{(k)} \in \mathbb{R}^{n \times t^k}$, the dimension of the search space in MPGMRES grows exponentially fast. It is natural to consider an approximation of the entire space by selective directions, and thus find an appropriate subspace whose dimension would grow more slowly, ideally only linearly. We call this algorithm selective MPGMRES (sMPGMRES). There is an inexhaustible list of potential selection strategies, and we describe some possibilities below:

- First we describe our heuristic method of choice for modifying complete MPGMRES to get a practical algorithm. At step k, apply the preconditioners only to certain columns of $V^{(k)}$. As a default we apply $\mathcal{P}_1$ to the first column of $V^{(k)}$, $\mathcal{P}_2$ to the second, and so on. Using this method the matrix $Z^{(k)}$ is in $\mathbb{R}^{n \times t}$ for all k, and we just have to perform a QR factorization of a matrix whose number of columns is equal to the (typically small) number of preconditioners. This process can be represented by replacing the second to last line in Algorithm 1, namely (2.6), by

$$Z^{(k)} = [\mathcal{P}_1^{-1}V_1^{(k)} \ldots \mathcal{P}_t^{-1}V_t^{(k)}]. \tag{2.11}$$

  Of course, it is not necessary to associate the ith preconditioner with the ith column, and other alternatives are possible. In general, given some permutation π (which may or may not change with each iteration), we can compute the i^{th} column of $Z^{(k)}$ as $\mathcal{P}_i^{-1}V_{\pi(i)}^{(k)}$.
- A viable alternative is to use all the columns of $V^{(k)}$ simultaneously by applying the multipreconditioning step to $V^{(k)}\boldsymbol{\alpha}_k$ for any vector $\boldsymbol{\alpha}_k$ of appropriate size. Specifically, giving equal weight to all columns, i.e., using the vector $V^{(k)}\mathbf{1}$, where $\mathbf{1}$ denotes the vector of ones, may be an appropriate choice.
- One could choose the "best" possible subspace in some sense, as was done by de Sturler [8] in another context; see also [13]. Note that such choices still keep $Z^{(k)} \in \mathbb{R}^{n \times t}$ for all k.

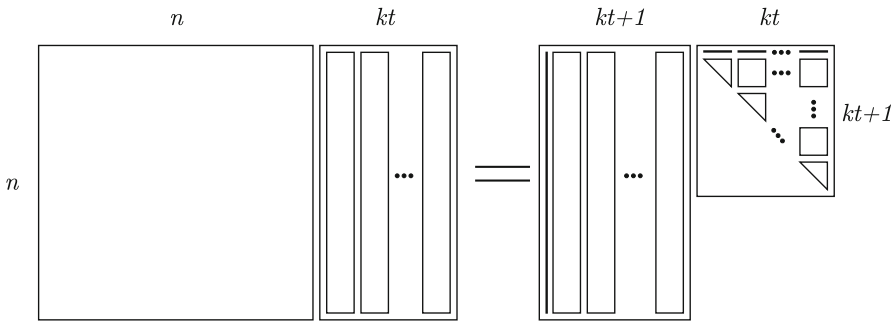


Fig. 2 Schematic diagram of Arnoldi-type decomposition for selective MPGMRES

Applying any of the selective schemes described above produces an Arnoldi-type decomposition, which we show schematically in Fig. 2. For the rest of this section, and in our numerical experiments, we will use the choice (2.11) as our selective version of MPGMRES; see also Sect. 4.

In terms of the optimality condition (2.10), selective MPGMRES minimizes the residual in a tk-dimensional subspace of the space

$$\{p(\mathcal{A}\mathcal{P}_1^{-1}, \ldots, \mathcal{A}\mathcal{P}_t^{-1})\mathbf{r}^{(0)},\ \ p \in \mathbb{P}_k[z_1, \ldots, z_t],\ \ p(0, \ldots, 0) = 1\},$$

which is itself $(t^k - 1)$ dimensional.

The heuristic choice (2.11), or any other choice of selected columns, determines which subspace one is using.

2.5 A case of linear growth of the search space

Let us consider the special case where we have two (nonsingular) preconditioners that add up to the coefficient matrix, i.e., $\mathcal{A} = \mathcal{P}_1 + \mathcal{P}_2$. This situation represents, for example, the ADI preconditioning approach; see, e.g., [5,19]. We begin with a general lemma which we find of interest beyond its use in this paper: If the sum of two preconditioners equals the matrix of coefficients of the original problem, then the preconditioned operators commute, and their product equals its sum. The converse also holds.

Lemma 2.1 *Let $\mathcal{A}, \mathcal{P}_1, \mathcal{P}_2$ be nonsingular matrices, and let $\mathcal{M}_1 = \mathcal{A}\mathcal{P}_1^{-1}$, $\mathcal{M}_2 = \mathcal{A}\mathcal{P}_2^{-1}$. Then, $\mathcal{A} = \mathcal{P}_1 + \mathcal{P}_2$ if and only if*

$$\mathcal{M}_1\mathcal{M}_2 = \mathcal{M}_2\mathcal{M}_1 = \mathcal{M}_1 + \mathcal{M}_2.$$

Proof Assume first that $\mathcal{A} = \mathcal{P}_1 + \mathcal{P}_2$. Then,

$$\mathcal{M}_1 = \mathcal{A}\mathcal{P}_1^{-1} = I + \mathcal{P}_2\mathcal{P}_1^{-1}, \quad \text{and} \quad \mathcal{M}_2 = \mathcal{A}\mathcal{P}_2^{-1} = I + \mathcal{P}_1\mathcal{P}_2^{-1}$$

It follows that

$$\mathcal{M}_1\mathcal{M}_2 = (I + \mathcal{P}_2\mathcal{P}_1^{-1})(I + \mathcal{P}_1\mathcal{P}_2^{-1}) = 2I + \mathcal{P}_2\mathcal{P}_1^{-1} + \mathcal{P}_1\mathcal{P}_2^{-1} = \mathcal{M}_1 + \mathcal{M}_2 = \mathcal{M}_2\mathcal{M}_1.$$

For the converse, we have that $\mathcal{M}_1 + \mathcal{M}_2 = \mathcal{M}_1\mathcal{M}_2$ implies

$$0 = \mathcal{A}(\mathcal{P}_1^{-1} + \mathcal{P}_2^{-1} - \mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1}) = \mathcal{A}\mathcal{P}_1^{-1}(\mathcal{P}_2 + \mathcal{P}_1 - \mathcal{A})\mathcal{P}_2^{-1},$$

and the lemma follows. □

Observe that Lemma 2.1 also applies to the matrices $\mathcal{P}_1^{-1}\mathcal{A}$ and $\mathcal{P}_2^{-1}\mathcal{A}$. The proof is essentially the same. The lemma is useful because it implies that the mixed terms do not bring any additional information to the space. Specifically, we can see in the case of polynomials of degree 2 that this lemma implies

$$\mathbb{P}_2[\mathcal{M}_1, \mathcal{M}_2] = \text{span}\left\{I, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_1^2, \mathcal{M}_2^2\right\} = \mathbb{P}_2[\mathcal{M}_1] + \mathbb{P}_2[\mathcal{M}_2].$$

More generally, we have that the polynomials in two variables decouple into the sum of polynomials in one variable of the same degree.

Proposition 2.2 *Let the variables z_1, z_2 be such that $z_1 z_2 = z_2 z_1 = z_1 + z_2$. Then, $\mathbb{P}_k[z_1, z_2] = \mathbb{P}_k[z_1] + \mathbb{P}_k[z_2]$, $k = 1, 2, \ldots$*

Proof We use induction on k. For $k = 1$, there is nothing to prove since

$$\mathbb{P}_1[z_1, z_2] = \text{span}\,\{I, z_1, z_2\} = \mathbb{P}_1[z_1] + \mathbb{P}_1[z_2].$$

We then assume that the statement of the proposition is true for k, and prove it for $k+1$. All we need to do is to show that any mixed terms of order $k+1$ can be written as a sum of terms in $\mathbb{P}_k[z_1, z_2]$. Indeed, for $1 \le i \le k$,

$$\begin{aligned} {z_1}^{k+1-i}{z_2}^{i} &= {z_1}^{k-i}(z_1 + z_2){z_2}^{i-1} \\ &= {z_1}^{k+1-i}{z_2}^{i-1} + {z_1}^{k-i}{z_2}^{i}. \end{aligned}$$

□

One of the direct consequences of Proposition 2.2 is that the search space has dimension $2k$ instead of $\mathcal{O}(2^k)$.

While the case described here is indeed special, it indicates that in some cases MPGMRES may perform very well without necessarily incurring an exponential growth of the dimension of the search space; see, e.g., the very recent paper [2] presenting another case where the search space has linear growth. This special case is also exploited to obtain short recurrences in MPCG; see [6, Theo. 3.3]

2.6 Related algorithms

Recently, *combination* preconditioners of the form

$$\mathcal{P}^{-1} = \alpha_1 \mathcal{P}_1^{-1} + \alpha_2 \mathcal{P}_2^{-1} \tag{2.12}$$

have been proposed and explored for saddle-point problems, where the constants α_1, α_2 are fixed; see in particular [27,28,45]. One can interpret the first step of MPGMRES, for $t = 2$, as choosing the optimal values of α_1, α_2 in (2.12) so that the residual $\mathbf{r}^{(1)}$ is minimal. In the second step, though, MPGMRES changes the values of α_1, α_2 in (2.12) to minimize the residual in a richer space, as described in Sect. 2.3.

In the recent paper [1], linear combinations of two preconditioners are used as in (2.12), and the coefficients are computed at each step to minimize the residual, as in MPGMRES. The difference is that only linear combinations (2.12) and not polynomials of higher order are sought.

In terms of the multipreconditioning paradigm considered in this paper, one could cycle with FGMRES through the available preconditioners in some prescribed order. This strategy was in fact proposed by Rui, Yong, and Chen in the context of electromagnetic wave scattering problems [32] in a method they termed 'multipreconditioned GMRES'. They show numerically that for their applications, the convergence of this method is never better than the best preconditioner applied by itself, although of course one may not know which preconditioner will

perform best *a priori*. This is in contrast to the selective MPGMRES method described here, which is derived to work in a potentially richer subspace and may beat single preconditioners applied separately (and therefore FGMRES with cycling); see the experiments in Sect. 4.

In general, we can say that (complete) MPGMRES is an extension of a block version of FGMRES, using an Arnoldi-type process, where at each step the space grows with application of all preconditioners to the current basis vectors. On the other hand, if one interprets FGMRES as any method having the property (2.3), then complete MPGMRES would be a special case.

Finally, we mention multi-splitting methods. Given a set of preconditioners $\mathcal{P}_i$, $i = 1, \ldots, t$, and a corresponding set of positive semi-definite diagonal weighting matrices $\mathcal{D}_i$ such that $\sum_i \mathcal{D}_i = I$, the multi-splitting algorithm [25] is defined as the iterative method governed by the stationary iteration $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \sum_{i=1}^{t} \mathcal{D}_i \mathcal{P}_i^{-1} \mathbf{r}^{(k)}$, where as usual $\mathbf{r}^{(k)} = \mathbf{b} - \mathcal{A}\mathbf{x}^{(k)}$. The difference is that in this method one has to define a weighting of the preconditioners *a priori*, whereas in MPGMRES a weighting which is in some sense optimal is computed as part of the algorithm.

3 Implementation details and computational cost

We first discuss the possibility of breakdowns and provide a few details on implementation. It is well known that all breakdowns in GMRES—i.e., cases where the last sub-diagonal entry of the matrix $\widetilde{H}_k$ is zero—are 'lucky' in that they occur only when the algorithm has converged to the exact solution. This is not the case with an algorithm that uses multiple preconditioners, as there are cases—e.g., if $\mathcal{P}_1 = \mathcal{P}_2$ or $\mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1} = \mathcal{P}_2^{-1}\mathcal{A}\mathcal{P}_1^{-1}$—where the matrix defined in (2.6) will not be of full rank, and hence we may have a zero on the sub-diagonal of $\widetilde{H}_k$ before reaching the exact solution.

To remove linear dependence we need to detect the rank of the matrix W in Algorithm 1; we do this by employing a rank-revealing QR factorization, e.g., the LAPACK routine `xGEQP3`. Such a factorization applies column pivoting to find a factorization of W of the form

$$WP = Q \begin{bmatrix} R & \hat{R} \\ 0 & 0 \end{bmatrix},$$

where P is a permutation matrix and $R \in \mathbb{R}^{r \times r}$ is upper triangular, where r is the (numerical) rank of W. We can then set $V^{(k+1)}$ to the first r columns of Q, and $H^{(k+1,k)}$ to R. Note that, since we have a QR factorization of the permuted WP (not W) we must also change the other blocks $H^{(i,k)}$ in the last column of H; we keep the first r columns of the matrix $H^{(i,k)}P$. The ordering of $Z^{(k)}$ also changes to $Z^{(k)}P$. Note that these changes of ordering can be done by storing an index vector, and no copying or deleting of vectors need be performed. This process is related to *deflation* in block GMRES algorithms; see, e.g., [16,22,30]. In block methods a vector is removed when a linear combination of the right hand side vectors has converged. Here, in contrast, the linear dependence is not an inherent issue, but is purely a result of the redundancy of the user-provided preconditioners.

Since the matrix $\widetilde{H}_k$ in MPGMRES is upper Hessenberg, the Givens rotations used in the solution of the least-squares problem (2.9) are applied in the same way as in GMRES.[1] The main difference in the implementation for MPGMRES is that—as described above—it is possible to have a sub-diagonal entry of $\widetilde{H}_k$ that is zero while the algorithm has not

[1] As with GMRES, other implementations are possible, e.g., using Householder transformations.

converged to the exact solution. There can be two reasons why the matrix W is deemed to be rank deficient. Either

- one or more of its columns have contributed nothing to the space; or
- we have a lucky breakdown, and the algorithm has converged.

We now establish a condition for distinguishing between these two situations. Suppose, without loss of generality, that pivoting was not required. The difference between an actual breakdown and a lucky breakdown is that in the former the corresponding column of $Z^{(k)}$ is linearly dependent on the previous columns, whereas in the latter the linear dependence is a result of convergence, and so the relevant columns of $Z^{(k)}$ are full rank.

Let $\bar{Z}_k$ denote $\tilde{Z}_{k-1}$ with the first r columns of $Z^{(k)}$ appended, and similarly $\bar{V}_{k+1}$ for $\tilde{V}_k$. Let $\bar{H}_k$ be the upper Hessenberg matrix formed by appending $[(H^{(1,k)})^T \ldots (H^{(r,k)})^T]^T$ to $\tilde{H}_{k-1}$. Then for any vector $\mathbf{v}$ which is orthogonal to the columns of $\bar{V}_{k+1}$ we have

$$\begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} [\bar{V}_{k+1}\ \mathbf{v}]^T \mathcal{A}[\bar{Z}_k\ (Z^{(k)})_i] = \begin{bmatrix} G\bar{H}_k & G\mathbf{h} \\ 0 & 0 \end{bmatrix}$$

for some vector $\mathbf{h}$, where G is the matrix corresponding to the previously defined Givens rotations, and hence $G\bar{H}_k$ is an upper triangular matrix.

If $[\bar{Z}_k\ (Z^{(k)})_i]$ has rank s, say, then since $\bar{V}_{k+1}$ and $\mathcal{A}$ have full rank, we must have that $\begin{bmatrix} \bar{H}_k & \mathbf{h} \\ 0 & 0 \end{bmatrix}$ also has rank s. In particular, if $(Z^{(k)})_i$ is a linear combination of the columns of $\bar{Z}_k$, then $G\mathbf{h} \in \mathcal{R}(G\bar{H}_k)$, and hence the final entry of $G\mathbf{h}$ will vanish. On the other hand, if $[\bar{Z}_k\ (Z^{(k)})_i]$ is full rank, then the rectangular matrix $\begin{bmatrix} G\bar{H}_k & G\mathbf{h} \\ 0 & 0 \end{bmatrix}$ must be of full rank too, and so the final entry of $G\mathbf{h}$ must be non-zero. Therefore, we can say that the zero in the (i, i) position of $H^{(k+1,k)}$ corresponds to a lucky breakdown if and only if the entry in the ith column of the final row of $GH^{(k,k)}$ is non-zero, where $GH^{(k,k)}$ denotes the matrix $H^{(k,k)}$ after the previous Givens rotations have been applied to it.

We remark that in order to calculate the rank numerically we must prescribe a tolerance on the diagonal entries of R. In our experience the performance of MPGMRES is not very sensitive to the choice of this parameter—if a column is nearly linearly dependent, then it may be removed and, in the case of selective MPGMRES, we can find an alternative vector which enriches the space in a more meaningful way. In our MATLAB and Fortran 95 codes we currently set this tolerance to $\sqrt{\epsilon_{\text{machine}}}$. Note that this is different than the situation for deflation in block GMRES for multiple right hand sides, where there are good numerical reasons for keeping the deflated columns available for further calculation; see, e.g., [22,30].

We have made available two codes which implement MPGMRES. There is a MATLAB implementation available on the Mathworks File Exchange[2], and a Fortran 95 version, `HSL_MI29`, which is part of HSL [18] and is freely available to academics. Of these, the HSL code uses reverse communication for the user to apply the preconditioner and matrix-vector products, and this can be done in parallel.

We now summarize the computational cost. In Table 1, we compare the number of matrix-vector products, inner products, and preconditioner solves for Algorithm 1 in its complete and selective versions, and Flexible GMRES with cycling preconditioners. While complete MPGMRES offers the possibility of rapid convergence due to a very rich space, its applicability with a large number of preconditioners, is limited by the explosion of its storage requirements. On the other hand, the selective version remains viable for small t.

[2] http://www.mathworks.com/matlabcentral/fileexchange/34562-multi-preconditioned-gmres

Table 1 Number of matrix-vector products, inner products and preconditioner solves at the k^{th} iteration when using t preconditioners, for complete and selective versions of MPGMRES, and FGMRES with cycling multiple preconditioners

	Matrix-vector products	Inner products	Preconditioner solves
MPGMRES	t^k	$\frac{t^{2k+1}+t^{2k}+t^{k+1}-3t^k}{2(t-1)}$	t^k
sMPGMRES	t	$(k-\frac{1}{2})t^2+\frac{3}{2}t$	t
FGMRES	1	$k+1$	1

Finally, we mention the potential for parallelization. Given a distributed memory architecture, each preconditioner solve can be performed on a separate processor, and by taking advantage of this one can obtain significant savings.

4 Applications and numerical experiments

In this section we apply the proposed algorithms to two numerical examples: the solution of a problem from PDE-constrained optimization, and preconditioners for the Navier–Stokes equations. See also [15] for a domain decomposition example.

We present convergence graphs, as well as computational times. In one set of examples we provide timings in a parallel setting with a Fortran code, and in other examples the timings are based on running a serial MATLAB implementation. We are aware that in many instances MATLAB times may be unreliable, but in this case we are comparing very similar codes, and therefore the computational times are representative of the performance of the method.

4.1 PDE-constrained optimization

Many real-world problems can be formulated as PDE-constrained optimization problems; see, e.g., [17,46], and references therein. Consider the following model problem

$$\begin{aligned} \min_{y,u} \frac{1}{2}||y-\hat{y}||_2^2 \quad &+\frac{\beta}{2}||u||_2^2 \\ \text{s.t.} \qquad -\nabla^2 y \quad &= u \text{ in } \Omega \\ y \quad &= f \text{ on } \partial\Omega. \end{aligned} \tag{4.1}$$

Here $\hat{y}$ is some pre-determined optimal state, and we want the system to get to a state y as close to the optimal state as possible—in the sense of minimizing the given cost functional—while satisfying Poisson's equation in some domain Ω. The mechanism we have of changing y is by varying the right-hand side of the PDE, u, which is called the control in this context. Note that the norm of the control appears in the cost functional, along with a Tikhonov regularization parameter, β, to ensure that the problem is well-posed.

If we discretize the problem using finite elements, then the minimum of the discretized cost functional is found by solving the linear system of equations

$$\begin{bmatrix} \beta Q & 0 & -Q \\ 0 & Q & K \\ -Q & K & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix}, \tag{4.2}$$

where Q is a mass matrix, K is a stiffness matrix and $\mathbf{u}$, $\mathbf{y}$ and $\mathbf{p}$ represent the discretized control, state and Lagrange multipliers respectively [29,39]. This matrix is typically very large and sparse, and the system (4.2) is generally solved iteratively. It was shown in [29] that two preconditioners that are optimal in terms of the mesh size taken are

$$\mathcal{P}_{bd} := \begin{bmatrix} \beta Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & KQ^{-1}K \end{bmatrix} \text{ and } \mathcal{P}_{con} := \begin{bmatrix} 0 & 0 & -Q \\ 0 & \beta KQ^{-1}K & K \\ -Q & K & 0 \end{bmatrix}.$$

Although these preconditioners perform well for moderately small values of β—say $\beta > 10^{-4}$—the clustering of the generalized eigenvalues of the preconditioned system deteriorates as $\beta \to 0$ [29, Corollary 3.3 and Corollary 4.4].

We remark that since this is a symmetric problem, for single preconditioners we would typically use a short-term recurrence Krylov method, such as MINRES in the case of $\mathcal{P}_{bd}$ or projected CG in the case of $\mathcal{P}_{con}$. Using such a method would result in smaller storage costs and computational time than with GMRES, although care must be taken if a large number of iterations are needed as MINRES can become unstable; see [44]. Note also that other preconditioners have been developed which may be better suited for small β, for example a block triangular preconditioner [3], or preconditioners that are β independent [26,39]. Here our aim is not to argue that this is the way one should solve such control problems—justifying such a claim would require more exhaustive tests and domain-specific theory, and as such is beyond the scope of this work. Rather, we use it to highlight a real-world case where combining two preconditioners using MPGMRES gives a solution faster than solving using GMRES with either preconditioner alone.

Example 4.1 We discretize the control problem (4.1) on the domain $\Omega = [0, 1]^2$ using Q_1 finite elements with a uniform mesh size of $h = 2^{-7}$. We take the desired state as

$$\hat{y} = \begin{cases} (2x_1 - 1)^2(2x_2 - 1)^2 & \text{if } (x_1, x_2) \in [0, \frac{1}{2}]^2 \\ 0 & \text{otherwise} \end{cases}.$$

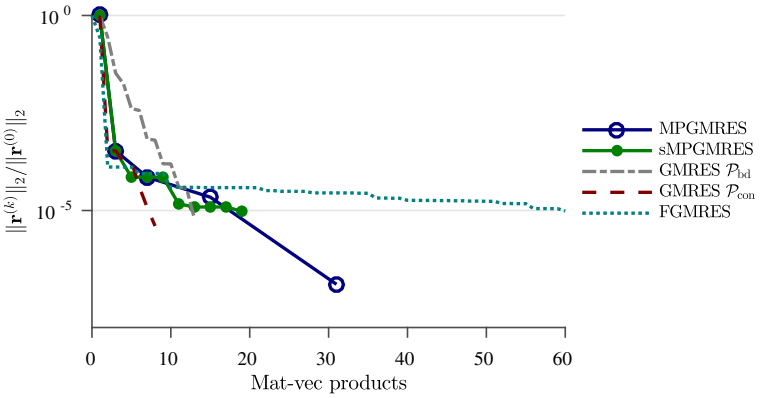
We apply the preconditioners $\mathcal{P}_{bd}$, $\mathcal{P}_{con}$ exactly (using MATLAB's `backslash` command), with MPGMRES (selective and complete), GMRES, and FGMRES with cycling. We used the MATLAB implementation of MPGMRES (available on the Mathworks File Exchange), and applied $\mathcal{P}_{bd}$ and $\mathcal{P}_{con}$ to the first and second columns of $Z^{(k)}$ respectively on odd iterations, reversing the order on even iterations. For all other options we used the defaults. The results are given in Fig. 3.

Figure 3 and the accompanying Table 2[3] show that, although neither of the preconditioners $\mathcal{P}_{bd}$ or $\mathcal{P}_{con}$ are effective for small β, their combination generates an effective solution procedure. In this table, and those that follow, bold face indicates the fastest algorithm. For $\beta > 10^{-4}$—the range in which the preconditioners were designed to be effective—there is no benefit to using MPGMRES. We see that FGMRES with cycling preconditioners is not competitive for this example. In Fig. 3, we show relative residual norms vs. (preconditioned) matrix-vector products, i.e., vs. the dimension of the search space. This measure represents most of the work performed for each method (especially for $t = 2$), and thus it gives a fair comparison.

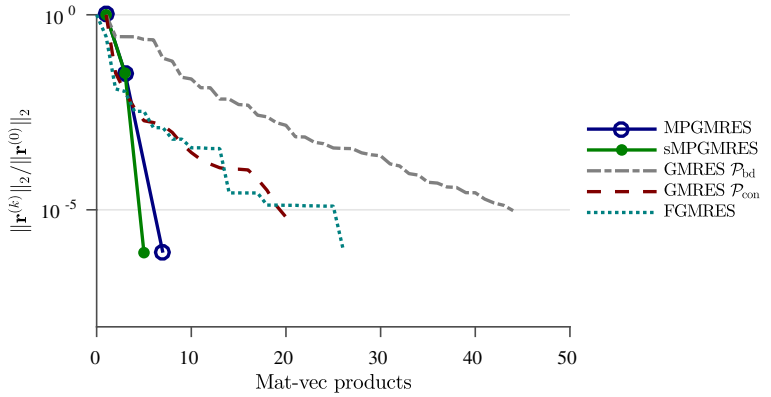
We note that $\mathcal{P}_{bd} + \mathcal{P}_{con} = \mathcal{A} + \mathcal{E}$, where

$$\mathcal{E} = \begin{bmatrix} 0 & Q & 0 \\ 0 & \beta KQ^{-1}K & 0 \\ 0 & 0 & KQ^{-1}K \end{bmatrix}.$$

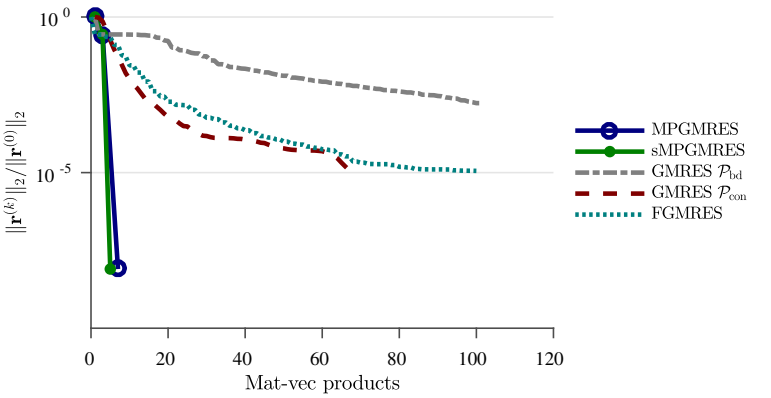
[3] These experiments were ran on a machine with an Intel Core i5-2500S CPU @ 2.70GHz with 8GB RAM.



(a) $\beta = 10^{-4}$



(b) $\beta = 10^{-6}$



(c) $\beta = 10^{-8}$

Fig. 3 Convergence curves showing number of matrix-vector products vs. relative residuals for solving the optimal control problem with MPGMRES, GMRES and FGMRES with cycling

Table 2 Timings (s) for Example 4.1

	$\beta = 10^{-4}$	$\beta = 10^{-6}$	$\beta = 10^{-8}$
Complete MPGMRES	6.5	1.7	1.6
Selective MPGMRES	4.1	**1.3**	**1.2**
GMRES, $\mathcal{P}_{bd}$	2.1	6.9	16.2
GMRES, $\mathcal{P}_{con}$	**1.9**	4.7	17.4
FGMRES (cycling)	13.6	5.3	26.3

Bold face indicates the fastest algorithm

In other words, especially for small β, these two preconditioners add up to the coefficient matrix, except in two blocks, one of which is small in magnitude if the regularization parameter β is small. Thus, we are close, in a structural sense, to the special case analyzed in Sect. 2.5. This observation may provide some intuitive insight as to why the complete and selective versions of MPGMRES perform almost identically. The two preconditioners seem to complement each other in terms of the eigenvectors associated with the eigenvalue 1. Both of the associated preconditioned matrices have an eigenvalue 1 of high multiplicity and the eigenvectors associated with one of these preconditioned matrices are orthogonal to the eigenvectors associated with the other preconditioned matrix; cf. the analyses in [14,42].

4.2 Navier–Stokes equations

The steady-state Navier–Stokes equations describe the flow of an incompressible Newtonian fluid, and are given by

$$\begin{aligned} -\nu \nabla^2 \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned}$$

plus appropriate boundary conditions. A common method to solve these equations numerically is to use a Picard iteration, which requires the solution of a linearized version of the Navier–Stokes equations—the Oseen problem—at each iteration. The Oseen problem takes the form

$$\begin{aligned} -\nu \nabla^2 \mathbf{u} + \mathbf{w} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned}$$

where, in a Picard method, the vector $\mathbf{w}$ is the computed velocity from the previous iteration.

An important quantity when dealing with the Navier–Stokes equations is the Reynolds number, $R = UL/\nu$, where U denotes some reference value (e.g., the maximum magnitude of the inflow velocity) and L denotes a characteristic length scale. Reynolds numbers $R > 1$ correspond to convection-dominated flows, which can be challenging to compute numerically.

Discretization of the Oseen equations by finite elements leads to a saddle-point system of the form

$$\begin{bmatrix} \mathbf{A} + \mathbf{N} & B^T \\ B & 0 \end{bmatrix},$$

where $\mathbf{A}$, $\mathbf{N}$ and B denote the vector Laplacian, vector convection and the divergence matrices respectively. For further details, see, e.g., [11, Chapter 7.3]. A number of preconditioners have been proposed for this linear system. Two of the most successful have been the

Table 3 Iteration numbers, with wall-clock times in parentheses, for solving an Oseen problem with R = 100

N	dim	PCD		LSC		sMPGMRES	
2^2	162	21	(0.03)	16	(0.02)	**14**	**(<0.01)**
2^3	578	30	(0.04)	**21**	**(<0.01)**	27	(0.07)
2^4	2178	37	**(0.03)**	**25**	(0.04)	35	(0.22)
2^5	8450	39	(0.19)	31	(0.17)	**22**	**(0.13)**
2^6	33,282	38	(0.50)	37	(0.60)	**24**	**(0.30)**
2^7	1,32,098	36	(1.92)	44	(2.62)	**25**	**(1.05)**
2^8	5,26,338	35	(6.27)	52	(9.60)	**24**	**(3.84)**

Bold face indicates the fastest algorithm

Table 4 Iteration numbers, with wall-clock times in parentheses, for solving an Oseen problem with $h = 2^{-8}$

R	PCD		LSC		sMPGMRES	
1	18	(2.86)	44	(8.18)	**13**	**(2.07)**
10	22	(3.81)	49	(9.49)	**15**	**(2.32)**
100	35	(5.99)	52	(10.16)	**24**	**(3.92)**

Bold face indicates the fastest algorithm

pressure-convection-diffusion (PCD) [20,40] and least-squares-commutator [10] (LSC) preconditioners.

The ideal PCD preconditioner, derived using a commutator argument, is given by the matrix

$$\begin{bmatrix} \mathbf{A}+\mathbf{N} & B^T \\ 0 & A_p F_p^{-1} Q \end{bmatrix},$$

where A_p denotes the discrete Laplacian in the pressure space, $F_p = A_p + N_p$ denotes the convection-diffusion equation in the pressure space, and Q denotes the pressure mass matrix. Alternatively, the ideal LSC preconditioner is given by the matrix

$$\begin{bmatrix} \mathbf{A}+\mathbf{N} & B^T \\ 0 & (B\mathbf{Q}^{-1}B^T)(B\mathbf{Q}^{-1}\mathbf{F}\mathbf{Q}^{-1}B^T)^{-1}(B\mathbf{Q}^{-1}B^T) \end{bmatrix},$$

where $\mathbf{Q}$ is the velocity mass matrix, and $\mathbf{F} = \mathbf{A} + \mathbf{N}$.

We approximate the mass matrix by its diagonal, and other systems are solved using a direct solver, either `HSL_MA48` (for non-symmetric systems) or `HSL_MA57` (for the symmetric systems). In practice these solves may be replaced by a spectrally equivalent preconditioner, e.g., a multigrid V-cycle.

As in Example 4.1, using MPGMRES here seems to be a promising alternative for solving matrices from Navier–Stokes problems, but more focused testing on a range of problems—which is beyond the scope of this manuscript—needs to be done before we can justify such a claim. Our primary aim here is to demonstrate two preconditioners that, when combined using MPGMRES, allow us to solve a non-artificial problem faster than if we employed any single preconditioner alone. Tables 3 and 4 give iteration counts and timings[4] for solving an Oseen problem posed on a unit square. We apply boundary conditions corresponding to a leaky cavity problem [11, Example 7.1.3] with an advection field given by

[4] These experiments were ran on a two-socket machine, each with Intel Xeon CPU E5-2687W 0 @ 3.10 GHz (i.e. 2 × 8 cores), and with 64GB memory total.

$$\left[2x_2(1-x_1^2),\ 2x_1(1-x_2^2)\right].$$

The system is discretized with a uniform mesh using Q2-Q1 (Taylor-Hood) finite elements. The resulting linear system is solved using GMRES and selective MPGMRES, with either the PCD preconditioner, the LSC preconditioner, or both. We use the Fortran 95 implementation of MPGMRES—`HSL_MI29`—and apply the preconditioners inside a simple openMP parallel 'do' loop. In the case of selective MPGMRES, the selection strategy used is to apply the LSC preconditioner to the first column of $Z^{(k)}$, and the PCD preconditioner to the second, otherwise we use the default options in `HSL_MI29`.

First, we highlight the fact that MPGMRES has greater memory requirements than GMRES, in this case requiring two columns to be added to V_k in the decomposition (2.7) per iteration, compared with the one added in standard GMRES (see Sect. 2.4, and note that Z_k can be recovered from V_k). Therefore, if memory is limited, using a single preconditioner may give a solution where MPGMRES may run out of memory.

Provided the problem can be solved using both methods, the wall clock time tells us which method is better. Neither the PCD nor the LSC preconditioner is the clear winner here, the best alternative depending on the mesh size. MPGMRES, on the other hand, almost always has the fewest iterations. Although each iteration of MPGMRES requires two preconditioner solves—and hence is more expensive—these are done in parallel here, and so the cost in terms of wall-clock time is not much greater (see the final column of the tables). The overhead of invoking the openMP machinery is amortized as the matrix size increases.

Acknowledgements We thank the two anonymous referees for their questions and comments, which helped us improve our presentation.

References

1. Ayuso de Dios, B., Barker, A.T., Vassilevski, P.S.: A combined preconditioning strategy for nonsymmetric systems. SIAM J. Sci. Comput. **36**, A2533G–A2556 (2014)
2. Bakhos, T., Ladenheim, S., Kitanidis, P.K., Saibaba, A.K., Szyld D.B.: Multipreconditioned GMRES for shifted systems. In: Research Report (2016), Department of Mathematics, Temple University (2016). Accessed 31 Mar 2016
3. Bai, Z.-Z.: Block preconditioners for elliptic PDE-constrained optimization problems. Computing **91**, 379–395 (2011)
4. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 418–477 (2002)
5. Birkhof, G., Varga, R.S., Young Jr., D.M.: Alternating direction implicit methods. Adv. Comput. **3**, 189–273 (1962)
6. Bridson, R., Greif, C.: A multipreconditioned conjugate gradient algorithm. SIAM J. Matrix Anal. Appl. **27**, 1056–1068 (2006)
7. Calandra, H., Gratton, S., Langou, J., Pinel, X., Vasseur, X.: Flexible variants of block restarted GMRES methods with application to geophysics. SIAM J. Sci. Comput. **34**, A714–A736 (2012)
8. de Sturler, E.: Nested Krylov methods based on GCR. J. Comput. Appl. Math. **67**, 15–41 (1996)
9. Elbouyahyaoui, L., Messaoudi, A., Sadok, H.: Algebraic properties of the block GMRES and block Arnoldi methods. Electr. Trans. Num. Anal. **33**, 207–220 (2008–2009)
10. Elman, H., Howle, V.E., Shadid, J., Shuttleworth, R., Tuminaro, R.: Block preconditioners based on approximate commutators. SIAM J. Sci. Comput. **27**, 1651–1668 (2006)
11. Elman, H., Silvester, D., Wathen, A.: Finite elements and fast iterative solvers, Second Edition, Oxford University Press, Oxford (2014)
12. Freund, R.W.: Model reduction methods based on Krylov subspaces. Acta Num. **12**, 267–319 (2003)
13. Gaul, A., Gutknecht, M.H., Liesen, J., Nabben, R.: A framework for deflated and augmented Krylov subspace methods. SIAM J. Matrix Anal. Appl. **34**, 495–518
14. Greenbaum, A., Pták, V., Strakoš, Z.: Any nonincreasing convergence curve is possible for GMRES. SIAM J. Matrix Anal. Appl. **17**, 465–469 (1996)

15. Greif, C., Rees, T., Szyld, D.B.: Additive Schwarz with variable weights, domain decomposition methods in science and engineering XXI. In: Erhel, J., Gander, M., Halpern, L., Pichot, G., Sassi, T., Widlund, O. (eds.) Proceedings of the 21st international conference on domain decomposition methods. Lecture Notes in Computer Science and Engineering, vol. 98, pp. 655–662. Springer, Berlin (2014)
16. Gutknecht, M.H.: Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. In: Siddiqi, A.H., Duff, I.S., Christensen, O. (eds.) Modern mathematical models, methods and algorithms for real world systems, pp. 420–447. Anamaya, New Delhi (2007)
17. Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: Optimization with PDE constraints. Mathematical modelling: theory and applications. Springer, New York (2008)
18. HSL (2013). A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk
19. Johnsson, S.L., Saad, Y., Schultz, M.H.: Alternating direction methods on multiprocessors. SIAM J. Stat. Sci. Comput. **8**, 686–700 (1987)
20. Kay, D., Loghin, D., Wathen, A.J.: A preconditioner for the steady-state Navier–Stokes equations. SIAM J. Sci. Comput. **24**, 237–256 (2002)
21. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indefinite linear systems. SIAM J. Matrix Anal. Appl. **21**, 1300–1317 (2000)
22. Langou, J.: Iterative methods for solving linear systems with multiple right hand sides. Ph.D. thesis, INSA, Toulouse, June 2003. CERFACS Report TH/PA/03/24 (2003)
23. Lukšan, L., Vlček, J.: Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. Num. Linear Algebra Appl. **5**, 219–247 (1998)
24. O'Leary, D.P.: The block conjugate gradient algorithm and related methods. Linear Algebra Appl. **29**, 293–322 (1980)
25. O'Leary, D.P., White, R.E.: Multi-splittings of matrices and parallel solution of linear systems. SIAM J. Algebraic Discret. Methods **6**, 630–640 (1985)
26. Pearson, J.W., Wathen, A.J.: A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Linear Algebra Appl. **19**, 816–829 (2012)
27. Pestana, J.: Nonstandard inner products and preconditioned iterative methods. D.Phil. thesis, University of Oxford (2011)
28. Pestana, J., Wathen, A.J.: Combination preconditioning of saddle point systems for positive definiteness. Num. Linear Algebra Appl. **20**, 785–808 (2013)
29. Rees, T., Dollar, H.S., Wathen, A.J.: Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. **32**, 271–298 (2010)
30. Robbé, M., Sadkane, M.: Exact and inexact breakdowns in the block GMRES method. Linear Algebra Appl. **419**, 265–285 (2006)
31. Ruhe, A.: Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. Math. Comput. **33**, 680–687 (1979)
32. Rui, P.-L., Yong, H., Chen, R.-S.: Multipreconditioned GMRES method for electromagnetic wave scattering problems. Microw. Opt. Technol. Lett. **50**, 150–152 (2008)
33. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM J. Sci. Comput. **14**, 461–469 (1993)
34. Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia (2003)
35. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
36. Sadkane, M.: A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices. Num. Math. **23**, 181–193 (2009)
37. Sadkane, M.: Block Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. Num. Math. **64**, 687–706 (1993)
38. Sarkis, M., Szyld, D.B.: Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms. Comput. Methods Appl. Mech. Eng. **196**, 1612–1621 (2007)
39. Schöberl, J., Zulehner, W.: Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. SIAM J. Matrix Anal. Appl. **29**, 752–773 (2007)
40. Silvester, D., Elman, H., Kay, D., Wathen, A.: Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. J. Comput. Appl. Math. **128**, 261–279 (2001)
41. Simoncini, V., Gallopouolos, E.: Convergence properties of block GMRES and matrix polynomials. Linear Algebra Appl. **47**, 97–119 (1996)
42. Simoncini, V., Szyld, D.B.: On the occurrence of superlinear convergence of exact and inexact Krylov subspace methods. SIAM Rev. **47**, 247–272 (2005)
43. Simoncini, V., Szyld, D.B.: Recent computational developments in Krylov subspace methods for linear systems. Num. Linear Algebra Appl. **14**, 1–59 (2007)

44. Sleijpen, G.L., van der Vorst, H.A., Modersitzki, J.: Differences in the effects of rounding errors in Krylov solvers for symmetric indefinite linear systems. SIAM J. Matrix Anal. Appl. **22**, 726–751 (2000)
45. Stoll, M., Wathen, A.J.: Combination preconditioning and the Bramble-Pasciak$^+$ preconditioner. SIAM J. Matrix Anal. Appl. **30**, 582–608 (2008)
46. Tröltzsch, F.: Optimal control of partial differential equations: Theory, methods and applications. American Mathematical Society, Providence (2010)
47. Vital, B.: Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocessor. Ph. D. thesis, Université de Rennes (1990)