Noname manuscript No.
(will be inserted by the editor)

Some observations on weighted GMRES

Stefan Güttel · Jennifer Pestana

Received: date / Accepted: date

Abstract We investigate the convergence of the weighted GMRES method for solving linear systems. Two different weighting variants are compared with unweighted GMRES for three model problems, giving a phenomenological explanation of cases where weighting improves convergence, and a case where weighting has no effect on the convergence. We also present new alternative implementations of the weighted Arnoldi algorithm which may be favourable in terms of computational complexity, and examine stability issues connected with these implementations. These implementations of weighted GMRES are compared for a large number of examples. We find that weighted GMRES may outperform unweighted GMRES for some problems, but more often this method is not competitive with other Krylov subspace methods like GMRES with deflated restarting or BICGSTAB, in particular when a preconditioner is used.

Keywords weighted GMRES · linear systems · Krylov subspace method · harmonic Ritz values

S. G. was supported by Deutsche Forschungsgemeinschaft Fellowship No. GU 1244/1-1. This publication was based on work supported in part by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

S. Güttel
Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford OX1 3LB, UK.
Current address: School of Mathematics, The University of Manchester, Alan Turing Building, Manchester M13 9PL, UK, E-mail: stefan.guettel@manchester.ac.uk

J. Pestana
Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford OX1 3LB, UK,
E-mail: pestana@maths.ox.ac.uk

1 Introduction

The GMRES method of Saad and Schultz [32] is one of the most popular Krylov subspace methods for solving a non-Hermitian system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{C}^{N\times N}$ is invertible and $\boldsymbol{b} \in \mathbb{C}^N$. Given an initial guess $\boldsymbol{x}^{(0)}$, GMRES computes successive iterates $\boldsymbol{x}^{(k)}$, $k = 1, 2, \ldots$, so that

$$\|\boldsymbol{r}^{(k)}\|_2 = \min_{\substack{p\in\mathcal{P}_k \\ p(0)=1}} \|p(A)\boldsymbol{r}^{(0)}\|_2,$$

where $\mathcal{P}_k$ denotes the linear space of polynomials of degree at most k, and $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$ is the k-th residual.

Since GMRES uses the Arnoldi algorithm, its computational cost increases with each iteration. An alternative is to restart GMRES after m iterations [32], taking the last computed residual as the next initial residual. We call the original method *full* GMRES and the latter *restarted* GMRES or GMRES(m). The set of m Arnoldi iterations between successive restarts will be called a *cycle*.

Although in exact arithmetic full GMRES is guaranteed to terminate with the exact solution in at most N steps, the restarted version may stagnate [6, 13, 32, 39]. Even if stagnation does not occur, convergence can be extremely slow [5, 40, 41]. The behaviour of restarted GMRES has been well studied and a number of remedies for slow convergence have been proposed [12, 24, 31, 35–37].

One such remedy is the *weighted* GMRES method of Essai [14], shortly denoted as WGMRES(m), that aims to improve the convergence of GMRES(m) by using a weighted inner product that changes at each cycle. This weighted inner product will be called a *D-inner product*, and is defined for any Hermitian positive definite matrix $D \in \mathbb{C}^{N\times N}$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^N$ as

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_D = \boldsymbol{y}^H D\boldsymbol{x},$$

where $\boldsymbol{y}^H$ represents the Hermitian conjugate of $\boldsymbol{y}$. The associated D-norm of a vector $\boldsymbol{x} \in \mathbb{C}^N$ is

$$\|\boldsymbol{x}\|_D = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle_D}.$$

The WGMRES(m) method also starts from an initial guess $\boldsymbol{x}^{(0)}$, and then computes successive approximations $\boldsymbol{x}^{(k)}$ at each cycle $k = 1, 2, \ldots$, such that at the end of the k-th cycle

$$\|\boldsymbol{r}^{(k)}\|_D = \min_{\substack{p\in\mathcal{P}_m \\ p(0)=1}} \|p(A)\boldsymbol{r}^{(k-1)}\|_D.$$

For further details we refer to Essai [14].

The essential ingredient of weighted GMRES is the weighted Arnoldi algorithm [14] that, after m iterations, generates basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ of the Krylov space

$$\mathcal{K}_m(A, \boldsymbol{r}) = \operatorname{span}\{\boldsymbol{r}, A\boldsymbol{r}, \ldots, A^{m-1}\boldsymbol{r}\}.$$

If one collects the Krylov basis vectors in a matrix $V_m = \big[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\big] \in \mathbb{C}^{N\times m}$, one can write down an Arnoldi decomposition

$$AV_m = V_{m+1}\underline{H_m} = V_m H_m + \boldsymbol{v}_{m+1} h_{m+1,m} \boldsymbol{e}_m^T, \tag{1}$$

where $\underline{H_m} \in \mathbb{C}^{(m+1)\times m}$ is the upper Hessenberg matrix

$$\underline{H_m} = \begin{bmatrix} H_m \\ h_{m+1,m}\boldsymbol{e}_m^T \end{bmatrix},$$

and $\boldsymbol{e}_m \in \mathbb{R}^m$ is the m-th canonical unit vector. The matrix $V_{m+1} = [V_m, \boldsymbol{v}_{m+1}]$ is D-orthonormal, i.e., $V_{m+1}^H D V_{m+1} = I_{m+1}$, the identity matrix of dimension $m+1$. The weighted Arnoldi algorithm requires more computation per iteration than standard Arnoldi in the Euclidean inner product and, consequently, one cycle of WGMRES(m) is computationally more expensive than one cycle of GMRES(m). However, convergence may occur more quickly.

We would like to emphasize that *weighting* must not be confused with *preconditioning*. From a (left) preconditioner P one expects that a Krylov subspace method using the matrix $P^{-1}A$ for the solution of the preconditioned linear system $P^{-1}A\boldsymbol{x} = P^{-1}\boldsymbol{b}$ converges faster, and typically this means that the eigenvalues of $P^{-1}A$ are clustered. (Right preconditioning has an analogous effect.) Weighting, on the other hand, does not change the Krylov space at all. It merely affects the inner product that is used to extract an approximation from the Krylov space built with the original matrix A.

Essai [14] considered the particular weight matrix

$$D = \frac{1}{\sqrt{N}\|\boldsymbol{r}^{(k-1)}\|_2}\mathrm{diag}\big(|r_1^{(k-1)}|, |r_2^{(k-1)}|, \ldots, |r_N^{(k-1)}|\big), \tag{2}$$

where the $r_j^{(k-1)}$ are the entries of the residual vector $\boldsymbol{r}^{(k-1)}$, so that greater emphasis is given to large components of the residual at each cycle. Note that $D = D^{(k)}$ changes at each cycle, but to keep notation simple, we typically omit the superindex k. The matrix D may be poorly conditioned if the diagonal entries vary too much in magnitude. In such cases, adding a small multiple of the identity will improve the conditioning of D.

For a number of test problems, WGMRES(m) with the weight matrix (2) required fewer cycles and less CPU time than the standard GMRES(m) method [14]. Application of WGMRES(m) to systems left-preconditioned by ILU(0) [23] also resulted in a slight reduction in the number of cycles required for convergence when compared with GMRES(m) [7]. However, the CPU time for WGMRES(m) was greater, with the additional CPU time a consequence of the computation of nonstandard inner products and norms. The weighted GMRES method has also been used to solve shifted linear systems [21], and systems with multiple right-hand sides [19]. We remark that Niu *et al.* [27] showed that WGMRES(m) can be accelerated by augmenting the Krylov space at cycle k with the ℓ most recent error approximations $\boldsymbol{z}^{(i)}$, $i = k-\ell, \ldots, k-1$, where $\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)}$ when $i > 0$ and $\boldsymbol{0}$ otherwise.

Although intuitively it seems sensible to emphasize those entries of the residual vector that are large in magnitude, the convergence behaviour of WGMRES(m) is not well understood. We attempt to remedy this here by examining the harmonic Ritz values associated with WGMRES(m). In view of the limited understanding even of the convergence of full GMRES, it seems unlikely at this stage that a simple and complete convergence theory for WGMRES can be developed. However, some insight can be gained by studying several model problems. We also propose three alternative implementations of the weighted Arnoldi algorithm and compare their cost and stability.

The outline of this paper is as follows. An analysis of the harmonic Ritz values associated with GMRES(m) and WGMRES(m) is given in Section 2. In Section 3 we describe Essai's implementation of the weighted Arnoldi algorithm and propose alternative implementations. We additionally discuss the merits of each. Finally, in Section 4, the different implementations are tested on a number of problems and compared with standard GMRES(m), GMRES(m) with deflated restarting, and BICGSTAB.

2 Harmonic Ritz values and the convergence of weighted GMRES

In this section we try to shed some light on the convergence behaviour of weighted GMRES and explain why this method may converge faster than unweighted GMRES in some cases, or why weighting may have no effect on the convergence at all. It should be emphasized that GMRES(m) and WGMRES(m) after k cycles yield residuals $\boldsymbol{r}^{(k)}$ from the *same* Krylov space $\mathcal{K}_{km+1}(A, \boldsymbol{r}^{(0)})$ but the harmonic Ritz values that uniquely determine the residual polynomials may exhibit considerably different behaviour. In other words, the approximation spaces of both methods are the same but the extractions from these spaces may be different. This very property makes weighting quite different from what is typically achieved by a (left) preconditioner P, which aims at improving the space $\mathcal{K}_{km+1}(P^{-1}A, P^{-1}\boldsymbol{r}^{(0)})$ for faster convergence.

The convergence of GMRES (and its restarted and weighted variants) is generally very difficult to analyse, if not impossible, as in theory any non-increasing convergence curve can be obtained with any choice of eigenvalues and Ritz values [3, 17, 11]. Additionally, restarted GMRES may exhibit any admissible cycle-convergence behaviour, where the two admissible situations are that the residuals decrease strictly monotonically at each cycle or that there is complete stagnation, i.e., that there is some index s for which $\boldsymbol{r}^{(i)} = \boldsymbol{r}^{(s)}$ for all $i > s$ [37]. Nevertheless, we still consider it instructive to make clear the relations between the unweighted and weighted (harmonic) Ritz values in the following. At the end of this section we will study three (unpreconditioned) model problems. As no set of examples can be exhaustive, our primary aim must be to illustrate and analyse some effects that may cause the difference in the convergence of GMRES(m) and WGMRES(m) observed in practical examples.

Facts about harmonic Ritz values. Let us start by reviewing some well-known facts about harmonic Ritz values, see [16,38,42]. First of all, the weighted harmonic Ritz values θ_j with corresponding Ritz vectors $\boldsymbol{u}_j$ satisfy

$$\boldsymbol{u}_j \in \mathcal{K}_m(A,\boldsymbol{r}), \quad (A^{-1}\boldsymbol{u}_j - \theta_j \boldsymbol{u}_j) \perp_D A\mathcal{K}_m(A,\boldsymbol{r}), \tag{3}$$

where $\boldsymbol{u} \perp_D \mathcal{V}$ means that $\langle \boldsymbol{u}, \boldsymbol{v}\rangle_D = 0$ for every $\boldsymbol{v} \in \mathcal{V}$. The condition $\boldsymbol{u}_j \in \mathcal{K}_m(A,\boldsymbol{r})$ is equivalent to $\boldsymbol{u}_j = V_m \boldsymbol{z}_j$, where V_m has as its columns the D-orthonormal basis vectors of $\mathcal{K}_m(A,\boldsymbol{r})$ in (1), and $\boldsymbol{z}_j \in \mathbb{C}^m$. It follows from (1) and (3) that

$$\left(H_m + |h_{m+1,m}|^2 \boldsymbol{f}_m \boldsymbol{e}_m^T\right) \boldsymbol{z}_j = \theta_j \boldsymbol{z}_j,$$

where $\boldsymbol{f}_m = H_m^{-H}\boldsymbol{e}_m$. It is also well known that the harmonic Ritz values $\theta_1^{(k)}, \ldots, \theta_m^{(k)}$ associated with cycle k are the zeros of the residual polynomial $p_m^{(k)} \in \mathcal{P}_m$, $p_m^{(k)}(0) = 1$, which is uniquely determined by the condition

$$\begin{aligned} \|\boldsymbol{r}^{(k)}\|_{D^{(k)}} &= \|p_m^{(k)}(A)p_m^{(k-1)}(A)\cdots p_m^{(1)}(A)\boldsymbol{r}^{(0)}\|_{D^{(k)}} \\ &= \min_{\substack{p\in\mathcal{P}_m \\ p(0)=1}} \|p(A)p_m^{(k-1)}(A)\cdots p_m^{(1)}(A)\boldsymbol{r}^{(0)}\|_{D^{(k)}}. \end{aligned}$$

If A is normal, then with $\widetilde{p}^{(k)} := p_m^{(k)}p_m^{(k-1)}\cdots p_m^{(1)}$ we have

$$\|\boldsymbol{r}^{(k)}\|_{D^{(k)}} \le \|\boldsymbol{r}^{(0)}\|_{D^{(k)}} \max_{\lambda\in\Lambda(A)} |\widetilde{p}^{(k)}(\lambda)|,$$

so that the convergence of (restarted) GMRES in the 2-norm can be understood in terms of the uniform convergence of residual polynomials on the discrete set of eigenvalues $\Lambda(A)$.

Relationship between Ritz values with and without weighting. Relationships between the Hessenberg matrices generated by weighted and unweighted GMRES were obtained by Essai [14], and we briefly recall them here.

Starting from a given vector, the Arnoldi method in the D-inner product builds D-orthogonal vectors that satisfy (1). With the same starting vector, the Arnoldi method in the Euclidean inner product computes orthogonal vectors $\widehat{\boldsymbol{v}}_i$ such that

$$A\widehat{V}_m = \widehat{V}_{m+1}\underline{\widehat{H}_m}, \quad \widehat{V}_m = [\widehat{\boldsymbol{v}}_1, \ldots, \widehat{\boldsymbol{v}}_m], \quad \underline{\widehat{H}_m} = \begin{bmatrix} \widehat{H}_m \\ \widehat{h}_{m+1,m}\boldsymbol{e}_m^T \end{bmatrix}, \tag{4}$$

where again $\underline{\widehat{H}_m}$ is an upper Hessenberg matrix. Additionally, the matrices V_m and $\widehat{V}_m$ are linked by

$$V_m = \widehat{V}_m S_m, \tag{5}$$

where S_m is upper triangular and is nonsingular in the absence of breakdown. This matrix S_m allows the Hessenberg matrices H_m and $\widehat{H}_m$ to be related by [14, Corollary 1]

$$\widehat{H}_m = S_m^{-1}H_m S_m + \frac{\widehat{h}_{m+1,m}}{s_{m+1,m+1}} \boldsymbol{s}_{m+1}\boldsymbol{e}_m^T, \tag{6}$$

where $\boldsymbol{s}_{m+1}$ is the vector obtained from the first m elements of the last column of S_{m+1} and $s_{m+1,m+1}$ is the $(m+1)$-th element of this column. This shows that the two Hessenberg matrices differ by a similarity transformation and a rank-one modification.

Examining the effect of the rank-one difference on the harmonic Ritz values is difficult. An easier task is to compare the weighted and unweighted Ritz values, θ_i and $\widehat{\theta}_i$, $i = 1, \ldots, m$, respectively. By applying the Bauer–Fike theorem [20, Theorem 6.3.2] to (6), we obtain

$$\min_j |\theta_j - \widehat{\theta}_i| \leq \kappa(S_m^{-1} X_m) \frac{|\widehat{h}_{m+1,m}|}{|s_{m+1,m+1}|} \|\boldsymbol{s}_{m+1}\|_2, \ i = 1, \ldots, m,$$

where X_m is an eigenvector matrix of H_m and $\kappa(S_m^{-1} X_m)$ is the 2-norm condition number of $S_m^{-1} X_m$.

The influence of the weighting matrix D is not obvious from the above inequality. However, we can obtain a (typically pessimistic) upper bound that displays the effect of weighting more clearly. Using (5) and the D-orthogonality of V_m we find that $S_m^{-H} S_m^{-1} = \widehat{V}_m^H D \widehat{V}_m$, from which it follows that $\kappa(S_m^{-1})^2 = \kappa(\widehat{V}_m^H D \widehat{V}_m) \leq \kappa(D)$. Thus,

$$\min_j |\theta_j - \widehat{\theta}_i| \leq \kappa(D)\kappa(X_m) \frac{|\widehat{h}_{m+1,m}|}{|s_{m+1,m+1}|} \|\boldsymbol{s}_{m+1}\|_2. \tag{7}$$

This shows that the difference between the weighted and unweighted Ritz values depends on the nonnormality of H_m (through X_m) and the conditioning of the change of basis matrix S_m. The latter term is bounded by the condition number of the diagonal weight matrix D, which in the case of Essai's weighting (2) is given by the ratio of the largest and smallest values of $\boldsymbol{r}^{(k-1)}$ in magnitude. Consequently, we obtain a smaller bound (7) when the entries of the residual vector have similar magnitudes.

The nearness of the Ritz values additionally depends on the quantities $\widehat{h}_{m+1,m}$, $s_{m+1,m+1}$ and $\boldsymbol{s}_{m+1}$. Within GMRES, $\widehat{h}_{m+1,m}$ is small in magnitude when the residual norm is small [32, Proposition 1], while the terms involving entries of S_{m+1} are small when the angle between $\boldsymbol{v}_{m+1}$ and $\widehat{\boldsymbol{v}}_{m+1}$ is small, since then $|s_{m+1,m+1}|$ is large relative to the remaining entries in the $(m+1)$-th column of S_{m+1}, i.e., the entries of $\boldsymbol{s}_{m+1}$.

Relationship between WGMRES and GMRES residuals. If the GMRES and WGMRES residuals coincide at the $(k-1)$-th cycle, then at the end of the k-th cycle the residuals can differ by at most [28, Lemma 1]

$$\frac{1}{\sigma_1(D^{\frac{1}{2}} \widehat{V}_m)} \|\boldsymbol{r}_{WGMRES}^{(k)}\|_D \leq \|\boldsymbol{r}_{GMRES}^{(k)}\|_2 \leq \frac{1}{\sigma_m(D^{\frac{1}{2}} \widehat{V}_m)} \|\boldsymbol{r}_{WGMRES}^{(k)}\|_D,$$

where σ_1 and σ_m represent the largest and smallest singular values, respectively. Since $\widehat{V}_m$ is orthogonal, $\sigma_m(D^{\frac{1}{2}}) \leq \sigma_m(D^{\frac{1}{2}} \widehat{V}_m) \leq \sigma_1(D^{\frac{1}{2}} \widehat{V}_m)$ and

$$\|\boldsymbol{r}_{GMRES}^{(k)}\|_2 \leq \|\boldsymbol{r}_{WGMRES}^{(k)}\|_2 \leq \sqrt{\kappa(D)} \|\boldsymbol{r}_{GMRES}^{(k)}\|_2.$$

From this we see that, given the same residual at the start of a cycle, weighted GMRES cannot reduce the Euclidean norm of the residual by more than GMRES during the cycle. Thus, if WGMRES has an advantage over GMRES this must be caused by a different starting residual.

Three model problems. We now compare the harmonic Ritz values generated by GMRES(m) with the weighted harmonic Ritz values of WGMRES(m) for three model problems. As well as Essai's weight matrix (2) we consider an alternative, proposed by Najafi and Zareamoghaddam [26], who were concerned that as the magnitudes of the entries of the residual became smaller it would be difficult to compute with (2); D_{rand} is a diagonal matrix with random uniformly distributed entries in (0.5,1.5). Although the following three examples have little practical relevance, we believe that they serve the purpose of giving insight into how weighting can possibly improve the convergence of restarted GMRES (Examples 1 and 3), or how weighting can have absolutely no effect on the convergence (Example 2).

Example 1 (interval) *We examine the harmonic Ritz values of a diagonal matrix with diagonal entries (and, hence, eigenvalues)* $1, 2, \ldots, 100$. *The right-hand side* $\boldsymbol{b}$ *is a vector of all ones, scaled to unit length. As one can see in Figure 1 (a), it appears that for unweighted GMRES(m) with* $m = 5$ *the harmonic Ritz values of every second cycle have* m *accumulation points, giving asymptotically* $2m$ *accumulation points* $\theta_1^*, \ldots, \theta_{2m}^*$ *in total. In this example these accumulation points are approximately*

$$\begin{aligned} &3.348,\ 22.208,\ 51.510,\ 79.318,\ 96.908,\\ &3.453,\ 20.616,\ 49.477,\ 79.784,\ 98.155. \end{aligned}$$

It should be noted that a similar 2-cyclic behaviour has been observed and analysed for the so-called optimum gradient method in the 1950's [15, 2] (this method can be interpreted as restarted FOM), and in the context of matrix function approximations in [1]. This restarted GMRES behaviour may also be related to the asymptotic orthogonality of successive initial residuals $\boldsymbol{r}^{(k-1)}$ *and* $\boldsymbol{r}^{(k)}$*, proven for the case* $m = N-1$ *in [5, Theorem 2]. A detailed investigation of this phenomenon is beyond the scope of this paper, but we expect that tools similar to those used in the mentioned papers can be applied.*

In Figure 1 (b) we show the level lines of the modulus of the nodal polynomial $q_{2m}(z) = \prod_{j=1}^{2m}(z - \theta_j^*)$. *These level lines are also known as* lemniscates, *see also the discussion in [1]. One can read off from this plot that the level line* $10^{14.535}$ *is the smallest one containing* $\Lambda(A)$ *in its interior, and the level line* $10^{14.909}$ *passes through the origin. By the normalization condition of residual polynomials, the modulus of* $q_{2m}(\lambda)/q_{2m}(0)$ *is at most* $10^{-0.3740} \approx 0.4227$ *for all* $\lambda \in \Lambda(A)$. *This residual polynomial is the result of two restart cycles, hence the expected convergence rate of restarted unweighted GMRES(m) in this example is approximately* $\sqrt{0.4227} \approx 0.6502$. *This rate is shown in Figure 1 (c) as the black dashed line, and it coincides well with the observed linear convergence of unweighted GMRES(m) (black curve with + markers).*

The convergence of the weighted GMRES(m) variants under consideration appears much less regular. The harmonic Ritz values associated with Essai's weighting appear to cover the spectral interval of A more evenly, and this is also indicated by the histogram in Figure 1 (d), which shows the distribution of harmonic Ritz values over the spectral interval of A. This "randomization" of interpolation nodes causes the method to converge faster than linearly. A similar effect is achieved by random weighting with D_{rand}. The fact that these harmonic Ritz values are spread out over the spectral interval of A makes visually clear that weighting does not attempt to cluster the spectrum (and thereby the harmonic Ritz values) as we might expect a preconditioner to do. We have not attempted to plot the lemniscates associated with the harmonic Ritz values produced by the weighted GMRES variants as, due to the observed irregular behaviour, these lemniscates cannot be described by just a few accumulation points. Therefore the evaluation of the residual polynomials at zero would not give more information than the computed residual norms.

Example 2 (circle) *Our second example is a diagonal matrix with $N = 100$ diagonal elements (eigenvalues) $\beta \cdot e^{2i\pi j/N} + 1$ on a circle of radius $\beta = 0.9$ centered at $z = 1$, $j = 1, 2, \ldots, N$. The right-hand side $\boldsymbol{b}$ is a vector of all ones, scaled to unit length, and the restart length is $m = 5$. As can be seen from Figure 2 (a), the harmonic Ritz values "spiral" towards the point $z = 1$, being almost evenly spaced on concentric circles. This effect appears for all types of weighting under investigation, and the convergence shown in Figure 2 (b) seems to be unaffected by whatever weighting method we use. To explain this observation, assume that at some cycle the harmonic Ritz values are unit roots of order m shifted and scaled to a circle of radius $\alpha < \beta$ centered at $z = 1$. The corresponding nodal polynomial is $q_m(z) = (z-1)^m - \alpha^m$. As can be verified easily, the maximal modulus of q_m on the circle of radius β centered at $z = 1$ is attained at points "in the middle" of two neighboring eigenvalues, for example, at the point $z^* = 1 + \beta e^{\pi i/N}$. Hence, the modulus of the residual polynomial is bounded by*

$$\left|\frac{q_m(z^*)}{q_m(0)}\right| = \left|\frac{q_m(1+\beta e^{\pi i/N})}{q_m(0)}\right| = \left|\frac{\beta^m e^{\pi i m/N} - \alpha^m}{(-1)^m - \alpha^m}\right| \approx \beta^m$$

for sufficiently small α. This explains why we see convergence with rate β in Figure 2 (b), indicated by the black dashed line, and weighting has essentially no effect on the convergence here.

Example 3 (Jordan block) *Our next example is an upper triangular Jordan block J of size $N = 100$ with eigenvalue 1. The right-hand side $\boldsymbol{b}$ is a vector of all ones, scaled to unit length. As one can see in Figure 3 (b), the unweighted GMRES(5) method will stagnate except in the first cycle, where a little progress is made. The corresponding harmonic Ritz values reappear at 10 points on a circle of radius one around the eigenvalue 1, with the real point $\theta = 2$ being counted twice due to symmetry, see Figure 3 (a). The harmonic Ritz values associated with Essai's weighted GMRES(5) method move closer*

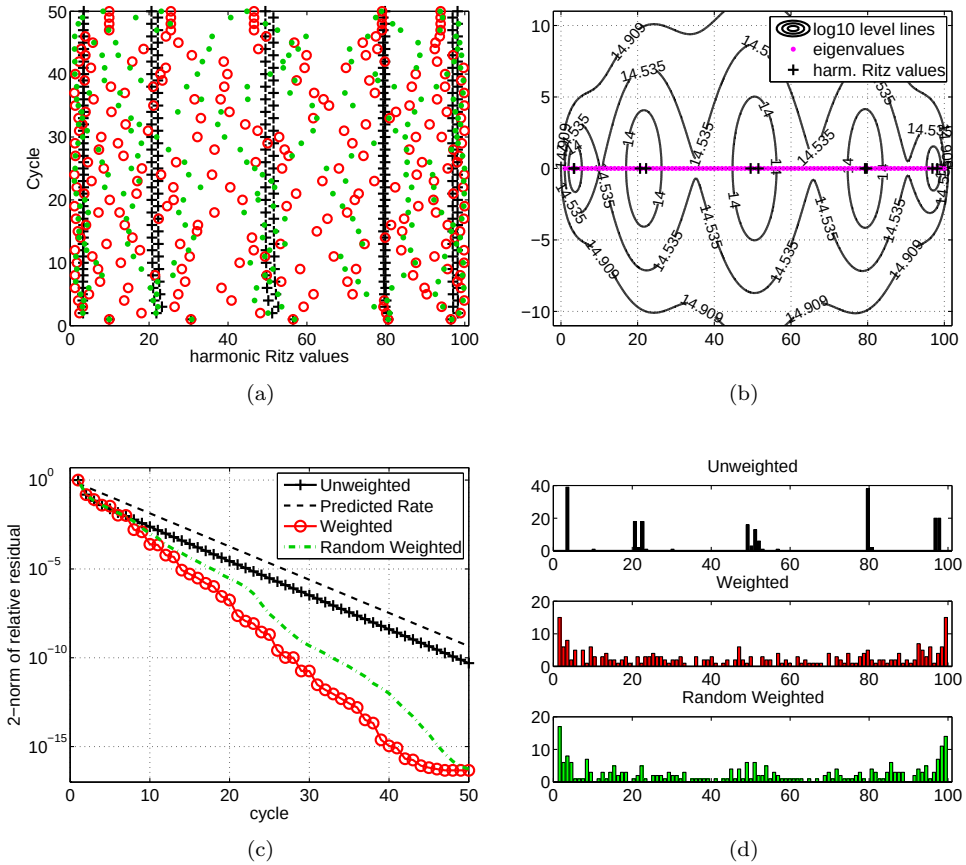
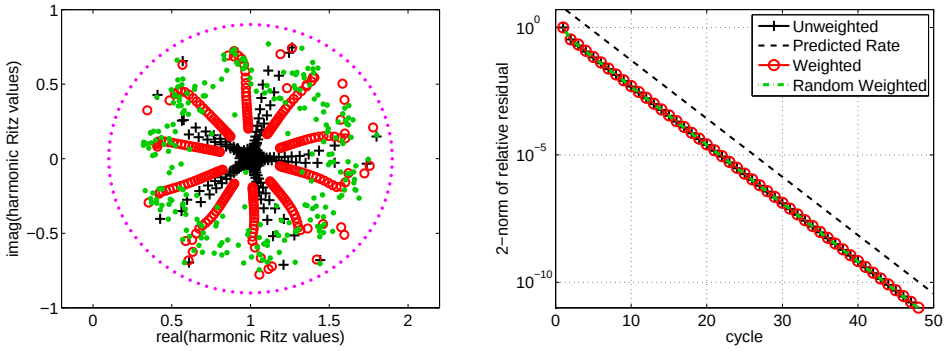


Fig. 1: (a) Harmonic Ritz values for GMRES(5) (black +), WGMRES(5) with (2) (red ∘), and WGMRES(5) with D_{rand} (green dots) for a diagonal matrix with equispaced eigenvalues on $[1, 100]$. The harmonic Ritz values are shown at the end of each of 50 cycles. (b) Lemniscates associated with unweighted GMRES(5). (c) Relative 2-norm residuals for the considered GMRES(5) variants. (d) Histogram indicating the distribution of harmonic Ritz values.

towards the eigenvalue 1 *with each cycle. After* 23 *cycles, weighted GMRES(5) has found the exact solution of* $J\boldsymbol{x} = \boldsymbol{b}$. *The random weighting matrix* D_{rand} *leads to stagnation just as unweighted GMRES(5). As opposed to the previous two examples, the harmonic Ritz values shown in Figure 3 (a) do not explain the convergence curves in Figure 3 (b) for this (highly) nonnormal example.*

To still give some insight into the different behaviour of unweighted GMRES and Essai's weighted GMRES, we visualize in Figure 3 (c) and (d) the entries of the residual vector after each cycle. The special structure of the Jordan matrix results in large residual entries being shifted up the vector with each



(a) (b)

Fig. 2: (a) Harmonic Ritz values for GMRES(5) (black +), WGMRES(5) with (2) (red ∘), and WGMRES(5) with D_{rand} (green dots) for a 100×100 matrix with eigenvalues distributed on the shifted unit circle. The harmonic Ritz values are shown at the end of each of 50 cycles. (b) Relative 2-norm residuals for the considered GMRES(5) variants. (All three convergence curves are visually indistinguishable.)

cycle. With unweighted GMRES(5) the residual vector is initially largest in its last entries. This phenomenon is fully described by Theorem 2.1 in [22], which is stated in terms of the transpose of the Jordan block, J^T, but is also valid for J. Some intuition is gained by observing that if $S = \left[\mathbf{0}, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_{N-1}\right]$ is the noncircular shift matrix, then $\mathcal{K}_m(J, \boldsymbol{r}) = \mathcal{K}_m(S, \boldsymbol{r})$ for any vector $\boldsymbol{r}$. It follows that the first two basis vectors, $\boldsymbol{r}$ and $S\boldsymbol{r}$ differ only in the last component and this affects the weight of the residual. More generally, any two basis vectors $S^j\boldsymbol{r}$ and $S^{j+1}\boldsymbol{r}$ differ in the $(N-j)$-th component only. Thus the "support" of nonzero entries in the residual vector at the end of the first cycle is in the last five components of the residual. At later cycles, this "support" forms a band that gets wider with each cycle, eventually polluting all entries of the residual vector and causing the method to stagnate. WGMRES(5) with Essai's weighting initially has largest entries at the bottom of the residual vector, although the other components also have some weight. Weighted GMRES then "cleans up" the entries of the residual vector which were large in the previous cycle because more weight is placed at those entries. Eventually, this WGMRES(5) variant finds the exact solution in the 24-th cycle.

Note that this Jordan example also explains why the Krylov space and hence the residual at the end of a cycle may depend sensitively on the initial residual for that cycle: instead of working with the matrix J and right-hand side vector $\boldsymbol{b}$ we could as well run restarted weighted or unweighted GMRES with $A = XJX^{-1}$ and $\widetilde{\boldsymbol{b}} = X\boldsymbol{b}$, where $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ is an arbitrary invertible matrix. Since $\mathcal{K}_m(A, \widetilde{\boldsymbol{b}}) = X\mathcal{K}_m(J, \boldsymbol{b})$, each column of Figure 3 (c) and

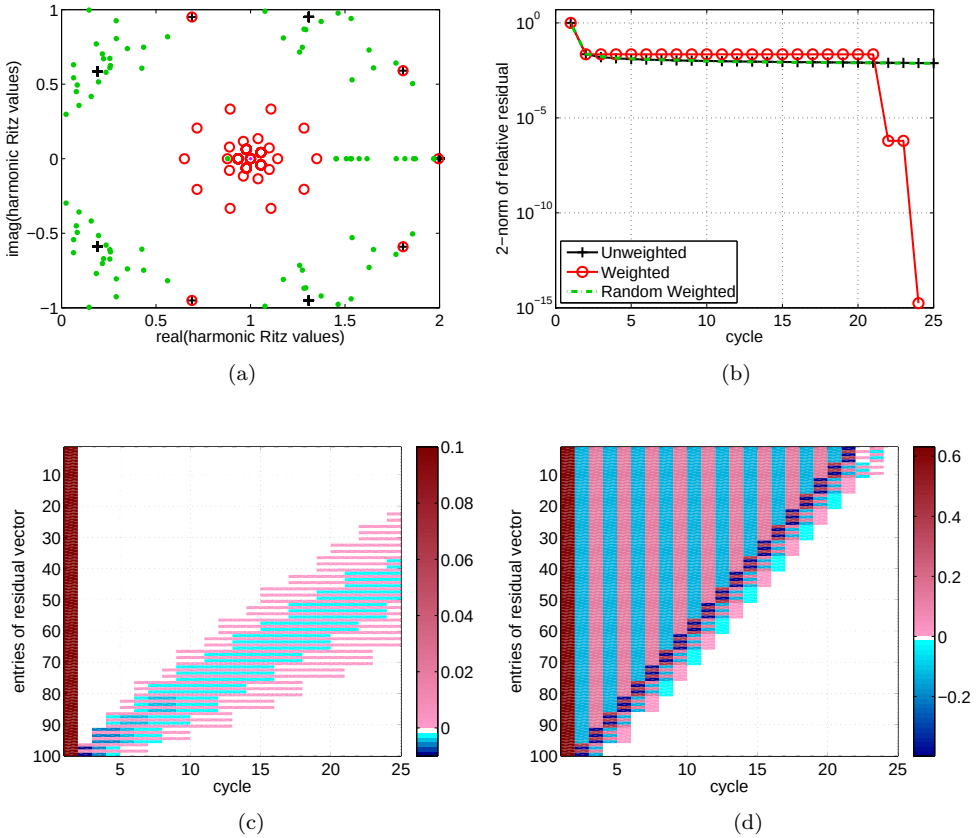


Fig. 3: (a) Harmonic Ritz values for GMRES(5) (black +), WGMRES(5) with (2) (red ∘), and WGMRES(5) with D_{rand} (green dots) for a Jordan block with eigenvalue 1. The harmonic Ritz values are shown at the end of each of the 25 cycles. The eigenvalue at 1 is plotted as an orange dot. (b) Relative 2-norm residuals for the considered GMRES(5) variants. (The convergences curves of WGMRES(5) with D_{rand} and GMRES(5) are visually hard to distinguish.) (c) Entries of the residual vectors after each cycle of unweighted GMRES(5). (d) Entries of the residual vectors after each cycle of GMRES(5) with Essai's weighting.

(d) can now be interpreted as the components of a residual vector in the basis of generalized eigenvectors of A. If a zero component r_j of a residual vector $\boldsymbol{r} = [r_1, \ldots, r_N]^T$ is altered from 0 to $\epsilon > 0$, for example by finite precision arithmetic, then this can cause a change of an eigenvector component in $A\boldsymbol{r}$ (and the following Krylov subspace vectors) of order $\epsilon\|\boldsymbol{x}_j\|$, which can be arbitrarily large depending on $\|\boldsymbol{x}_j\|$.

We conclude this section by remarking on the observations of Cao and Yu [7], who found little benefit in using WGMRES(m) on ILU-preconditioned problems unless the restart length was short. In particular, they found that although the number of cycles of WGMRES(m) was slightly lower than that required by GMRES(m) the CPU time of WGMRES(m) was greater. It is known that the application of ILU preconditioners often clusters eigenvalues. In such cases weighting—the effect of which is typically to shift the harmonic Ritz values from those obtained by GMRES(m)—appears to offer little benefit. We have found that, for a number of linear systems right-preconditioned by ILU(0), WGMRES(m) requires a number of cycles that is similar to, or slightly greater than, that required by GMRES(m) (see Example 9). However, WGMRES(m) may be of some benefit when the restart length is short or when the spectrum of the preconditioned matrix is not nicely clustered.

3 The weighted Arnoldi algorithm

In this section we discuss four variants of the weighted Arnoldi algorithm for constructing the D-orthonormal basis required by WGMRES. All variants are mathematically equivalent, but Algorithms 2, 3 and 4 are, to our knowledge, new implementations.

3.1 Variants of the algorithm

The most straightforward implementation of the weighted Arnoldi algorithm replaces Euclidean inner products in a standard Arnoldi algorithm with D-inner products (see, e.g., Essai [14], Sarkis and Szyld [33]). The j-th iteration of such a weighted Arnoldi algorithm with modified Gram–Schmidt orthogonalization (MGS) requires the computation of j D-inner products (see Algorithm 1). If D is a diagonal matrix such as (2), the inner products in Algorithm 1 can be efficiently implemented as $(\boldsymbol{v} \circ \boldsymbol{d})^H \boldsymbol{u}$ or $\boldsymbol{v}^H(\boldsymbol{d} \circ \boldsymbol{u})$, where $\circ$ represents the Hadamard product and $\boldsymbol{d}$ the vector of diagonal elements of D. For a general matrix D, each nonstandard inner product consists of a matrix-vector product and an Euclidean inner product. Algorithm 2 shows that when classical Gram–Schmidt orthogonalization (CGS) is used instead, these nonstandard inner products can be replaced by Euclidean inner products and only two additional matrix-vector products with D in the whole cycle; these can be computed as Hadamard products when D is diagonal. As a result, Algorithm 2 may be much cheaper than Algorithm 1. In addition, the CGS method is more easily parallelizable and may be faster in practice, as it can make use of higher level BLAS routines. Even so, each step of weighted Arnoldi with Algorithms 1 or 2 is more expensive than a step of the Arnoldi algorithm in the Euclidean inner product because of the D-inner products and D-norms. The number of multiplications by D is fixed for the CGS version but increases as k increases in the MGS algorithm. Note that Algorithms 1 and 2 can be used in combination with a preconditioner in a straightforward manner.

Variants of the weighted Arnoldi algorithm:

Inputs: Matrix $A \in \mathbb{C}^{N\times N}$, diagonal positive definite weight matrix $D \in \mathbb{C}^{N\times N}$, vector $\boldsymbol{r} \in \mathbb{C}^N$, number of Arnoldi iterations m

Outputs: D-orthonormal Arnoldi vectors $\{\boldsymbol{v}_1 \ldots, \boldsymbol{v}_m\}$ of $\mathcal{K}_m(A, \boldsymbol{r})$ and upper Hessenberg matrix $\underline{H_m} = [h_{ij}] \in \mathbb{C}^{(m+1)\times m}$

Algorithm 1: Explicit D-inner products and MGS orthogonalization

$\boldsymbol{v}_1 = \boldsymbol{r}/\|\boldsymbol{r}\|_D$
for $k = 1, 2, \ldots, m$ **do**
  $\boldsymbol{w} = A\boldsymbol{v}_k$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{jk} = \boldsymbol{v}_j^* D\boldsymbol{w}$
    $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{v}_j h_{jk}$
  end
  $h_{k+1,k} = \|\boldsymbol{w}\|_D$
  if $h_{k+1,k} = 0$ **then**
    Stop
  end
  $\boldsymbol{v}_{k+1} = \boldsymbol{w}/h_{k+1,k}$
end

Algorithm 2: Explicit D-inner products and CGS orthogonalization

$\boldsymbol{v}_1 = \boldsymbol{r}/\|\boldsymbol{r}\|_D$
for $k = 1, 2, \ldots, m$ **do**
  $\boldsymbol{w} = A\boldsymbol{v}_k$
  $\boldsymbol{y} = D\boldsymbol{w}$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{jk} = \boldsymbol{v}_j^* \boldsymbol{y}$
    $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{v}_j h_{jk}$
  end
  $h_{k+1,k} = \|\boldsymbol{w}\|_D$
  if $h_{k+1,k} = 0$ **then**
    Stop
  end
  $\boldsymbol{v}_{k+1} = \boldsymbol{w}/h_{k+1,k}$
end

Algorithm 3: Implicit D-inner products and MGS orthogonalization

$\widetilde{A} = D^{\frac{1}{2}} A D^{-\frac{1}{2}}$
$\boldsymbol{w} = D^{\frac{1}{2}}\boldsymbol{r}$
$\widetilde{\boldsymbol{v}}_1 = \boldsymbol{w}/\|\boldsymbol{w}\|_2$
for $k = 1, 2, \ldots, m$ **do**
  $\boldsymbol{w} = \widetilde{A}\widetilde{\boldsymbol{v}}_k$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{jk} = \widetilde{\boldsymbol{v}}_j^* \boldsymbol{w}$
    $\boldsymbol{w} = \boldsymbol{w} - \widetilde{\boldsymbol{v}}_j h_{jk}$
  end
  $h_{k+1,k} = \|\boldsymbol{w}\|_2$
  if $h_{k+1,k} = 0$ **then**
    Stop
  end
  $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{w}/h_{k+1,k}$
end
$[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] = D^{-\frac{1}{2}}[\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_m]$

Algorithm 4: Implicit D-inner products and CGS orthogonalization

$\widetilde{A} = D^{\frac{1}{2}} A D^{-\frac{1}{2}}$
$\boldsymbol{w} = D^{\frac{1}{2}}\boldsymbol{r}$
$\widetilde{\boldsymbol{v}}_1 = \boldsymbol{w}/\|\boldsymbol{w}\|_2$
for $k = 1, 2, \ldots, m$ **do**
  $\boldsymbol{w} = \widetilde{A}\widetilde{\boldsymbol{v}}_k$
  $\boldsymbol{y} = \boldsymbol{w}$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{jk} = \widetilde{\boldsymbol{v}}_j^* \boldsymbol{y}$
    $\boldsymbol{w} = \boldsymbol{w} - \widetilde{\boldsymbol{v}}_j h_{jk}$
  end
  $h_{k+1,k} = \|\boldsymbol{w}\|_2$
  if $h_{k+1,k} = 0$ **then**
    Stop
  end
  $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{w}/h_{k+1,k}$
end
$[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] = D^{-\frac{1}{2}}[\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_m]$

As an alternative to computing D-inner products at each step of the weighted Arnoldi algorithm, we can perform the Arnoldi algorithm in the Euclidean inner product on a row- and column-scaled matrix and a scaled starting vector. More precisely, we can apply the Arnoldi algorithm in the Euclidean inner product to the transformed matrix $\widetilde{A} = D^{\frac{1}{2}} A D^{-\frac{1}{2}}$ and starting vector $\widetilde{\boldsymbol{r}} = D^{\frac{1}{2}}\boldsymbol{r}$. Doing so gives matrices $\widetilde{V}_m$ and $\underline{\widetilde{H}_m}$ that satisfy

$$\widetilde{A}\widetilde{V}_m = \widetilde{V}_{m+1}\underline{\widetilde{H}_m}, \tag{8}$$

where $\underline{\widetilde{H}_m}$ is an upper Hessenberg matrix, $\widetilde{V}_m^H \widetilde{V}_m = I_m$, and the columns of $\widetilde{V}_m$ form a basis of $\mathcal{K}_m(\widetilde{A}, \widetilde{\boldsymbol{r}})$. Premultiplying both sides of (8) by $D^{-\frac{1}{2}}$ gives the Arnoldi decomposition

$$AV_m = V_{m+1}\underline{H_m},$$

where $V_m = D^{-\frac{1}{2}}\widetilde{V}_m$ and $\underline{H_m} = \underline{\widetilde{H}_m}$. The columns of V_m are D-orthonormal since $V_m^H D V_m = \widetilde{V}_m^H \widetilde{V}_m = I_m$. Moreover, since the columns of $\widetilde{V}_m$ span $\mathcal{K}_m(\widetilde{A}, \widetilde{\boldsymbol{r}}) \equiv \mathcal{K}_m(D^{\frac{1}{2}} A D^{-\frac{1}{2}}, D^{\frac{1}{2}}\boldsymbol{r})$, the columns of V_m span $\mathcal{K}_m(A, \boldsymbol{r})$.

The resulting algorithm with MGS orthogonalization is given in Algorithm 3; Algorithm 4 is the CGS version. These algorithms differ from preconditioned Arnoldi methods since the basis vectors are always scaled by the matrix $D^{-\frac{1}{2}}$. Again, the CGS version is more amenable to parallelization and can utilize higher level BLAS, although the number of (Euclidean) inner products is identical to that required by the MGS version. We note that orthogonalizing a transformed matrix with respect to the Euclidean inner product, to obtain a basis that is orthogonal with respect to a nonstandard inner product, was considered in the context of rounding error analysis in [30]. Additionally, Heyouni and Essai [19] considered the use of matrix square roots for enforcing D-orthogonality when solving systems with multiple right-hand sides, although they still computed D-inner products at each iteration of their weighted Arnoldi algorithm. It does not seem feasible to use Algorithms 3 and 4 in combination with a preconditioner.

In the remainder of this manuscript, we use Algorithms 1–4 to refer to both the different weighted Arnoldi variants and the corresponding WGMRES(m) methods. The meaning will be clear from the context. In all weighted GMRES variants, the stopping criterion that is easiest to measure is the reduction in $\|\boldsymbol{r}^{(k)}\|_{D^{(k)}}$. Throughout this manuscript, however, we measure the reduction of the residual $\|\boldsymbol{r}^{(k)}\|_2$ in the Euclidean norm and stop the algorithms when $\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{b}\|_2$ has decreased below a prescribed tolerance. It is possible to use $\|\boldsymbol{r}^{(k)}\|_{D^{(k)}}$ and the elements of $D^{(k)}$ to bound the reduction of the residual in the Euclidean norm (see [4]), but we do not consider this here.

3.2 Operation counts

In all four variants of the Arnoldi algorithm discussed here, the number of matrix-vector products is m, and their computation requires $2m \times \texttt{Nnz}$ arithmetic operations, where `Nnz` is the number of nonzero elements in the matrix A

or $\widetilde{A}$, respectively. Furthermore, the successive orthogonalization of m Krylov basis vectors requires $m(m+1)/2$ inner products and vector updates of the form $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{v}_j h_{jk}$. One such vector update requires $2N$ arithmetic operations. Computing a single inner product requires $2N$ or $3N$ arithmetic operations in the unweighted or weighted case, respectively. The row and column[1] scaling of the matrix A to form $\widetilde{A}$ and vector $\boldsymbol{r}$ to form $\widetilde{\boldsymbol{r}}$ in Algorithms 3 and 4 requires $3N + 2 \times \texttt{Nnz}$ operations, counting the computation of a square root or a division as a single arithmetic operation. Algorithms 3 and 4 also require Nm multiplications to scale the basis vectors at the end of each cycle of the weighted Arnoldi algorithm.

In Section 4 we will compare different variants of WGMRES(m) with GMRES-DR(m, ℓ) [25] on various numerical examples, but it easy to discuss the computational cost theoretically as well. In each cycle, the GMRES-DR(m, ℓ) method augments the Krylov basis computed by GMRES(m) with the Schur vectors corresponding to the ℓ smallest harmonic Ritz values. These Schur vectors, and the initial residual of the new cycle k, form the columns of a matrix $P_{\ell+1} \in \mathbb{C}^{(m+1)\times(\ell+1)}$, from which the first $\ell + 1$ basis vectors of the new cycle are obtained via

$$V_{\ell+1}^{(k)} = V_{m+1}^{(k-1)} P_{\ell+1}. \tag{9}$$

The remaining columns of $V_{\ell+m}^{(k)}$ are then computed by the Arnoldi algorithm. The dense matrix-matrix multiplication (9) requires $2N(m+1)(\ell+1)$ arithmetic operations and so dominates the cost of augmenting the Krylov basis. Depending on the restart length m and the number of Schur vectors ℓ, the cost of this matrix-vector product can make each cycle of GMRES-DR(m, ℓ) more expensive than a cycle of WGMRES(m).

We summarize the operation counts for a single cycle of Algorithms 1–4 and GMRES-DR in Table 1.

Table 1: Operation counts for a one cycle of Algorithms 1–4 and GMRES-DR.

Algorithm 1	$2m \times \texttt{Nnz} + \frac{5}{2}Nm^2$
Algorithm 2	$2m \times \texttt{Nnz} + 2Nm^2 + 2Nm$
Algorithm 3	$2m \times \texttt{Nnz} + 2Nm^2 + 3N + 2 \times \texttt{Nnz} + Nm$
Algorithm 4	$2m \times \texttt{Nnz} + 2Nm^2 + 3N + 2 \times \texttt{Nnz} + Nm$
GMRES-DR	$2m \times \texttt{Nnz} + 2Nm^2 + 2Nm\ell$

Although the performance of each algorithm is machine dependent, Algorithms 3 and 4, for which the cost of the weighted inner product is independent

[1] As pointed out by one of the referees, it is possible to eliminate the column scaling in Algorithms 3 and 4 at the expense of storing m additional vectors. This is achieved by computing each Arnoldi vector $\boldsymbol{v}_k$ as soon as the corresponding scaled vector $\widetilde{\boldsymbol{v}}_k$ is available. Specifically, assume that $\widetilde{\boldsymbol{v}}_k$ is available and compute $\boldsymbol{v}_k = D^{-\frac{1}{2}}\widetilde{\boldsymbol{v}}_k$. Then at each iteration of Algorithm 3 and 4 the vector $\boldsymbol{w}$ can be formed as $\boldsymbol{w} = D^{-\frac{1}{2}}AD^{\frac{1}{2}}\widetilde{\boldsymbol{v}}_k = D^{-\frac{1}{2}}AD^{\frac{1}{2}}D^{-\frac{1}{2}}\boldsymbol{v}_k = D^{-\frac{1}{2}}A\boldsymbol{v}_k$.

of the restart length, can become more efficient than Algorithms 1 and 2 in exact arithmetic when the restart length increases and the number of nonzeros `Nnz` is sufficiently small. However, the numerical stability of these four variants can be quite different, as discussed in the next subsection.

3.3 Numerical stability

In finite precision the quantities computed by Algorithms 1–4 typically differ. In particular, the initial residual at each cycle, that defines both the Krylov space and the weight matrix, may change. Since the choice of weight matrix determines the way in which iterates are extracted from the Krylov space, any change may alter significantly subsequent iterates and residuals.

Differences between the algorithms may also be caused by the orthogonalization procedure. It is well known that classical Gram–Schmidt orthogonalization in the Euclidean inner product is less stable than modified Gram–Schmidt orthogonalization (see, for example, [29]). Our experience is that D-orthogonalization, when D is given by (2), also becomes less stable by CGS orthogonalization than by MGS orthogonalization, i.e., Algorithms 2 and 4 are less stable than Algorithms 1 and 3. These observations are supported by bounds on

$$\|I_m - V_m^H D V_m\|_2 \tag{10}$$

obtained by Rozložník *et al.* [30]. The MGS bound depends on κ, the 2-norm condition number of $\left[D^{\frac{1}{2}}\boldsymbol{r}, D^{\frac{1}{2}}AV_m\right]$, while the CGS bound depends on κ^2 and, consequently is much worse than the MGS bound when κ is large.

In WGMRES(m) this loss of orthogonality principally affects the small least squares problem that must be solved. By the minimization property of WGMRES(m), at the k-th cycle the residual vector satisfies

$$\begin{aligned}\|\boldsymbol{r}^{(k)}\|_D &= \min_{\boldsymbol{y}\in\mathbb{C}^m} \left\|V_{m+1}\left(\underline{H_m}\boldsymbol{y} - \|\boldsymbol{r}^{(k-1)}\|_D \boldsymbol{e}_1\right)\right\|_D \\ &= \min_{\boldsymbol{y}\in\mathbb{C}^m} \left\|\underline{H_m}\boldsymbol{y} - \|\boldsymbol{r}^{(k-1)}\|_D \boldsymbol{e}_1\right\|_{V_m^H D V_m},\end{aligned}$$

where, although all matrices change with each cycle, we have dropped the superscript k for readability. If the columns of V_m are D-orthogonal, then $\boldsymbol{r}^{(k)} = \boldsymbol{r}^{(k-1)} + V_m\boldsymbol{y}_m$, where

$$\boldsymbol{y}_m = \arg\min_{\boldsymbol{y}\in\mathbb{C}^m} \left\|\underline{H_m}\boldsymbol{y} - \|\boldsymbol{r}^{(k-1)}\|_D \boldsymbol{e}_1\right\|_2$$

and it is this small least squares problem that is solved in WGMRES(m) to update the iterate and residual vector. However, if $V_m^H D V_m \approx I_m$, then $\widehat{\boldsymbol{r}}^{(k)} = \boldsymbol{r}^{(k-1)} + V_m\boldsymbol{y}_m$, with

$$\boldsymbol{y}_m = \arg\min_{\boldsymbol{y}\in\mathbb{C}^m} \left\|\underline{H_m}\boldsymbol{y} - \|\boldsymbol{r}^{(k-1)}\|_D \boldsymbol{e}_1\right\|_2,$$

is only an approximation of $\boldsymbol{r}^{(k)}$.

However, for any $\boldsymbol{r}^{(k)} \in \mathbb{C}^m$,

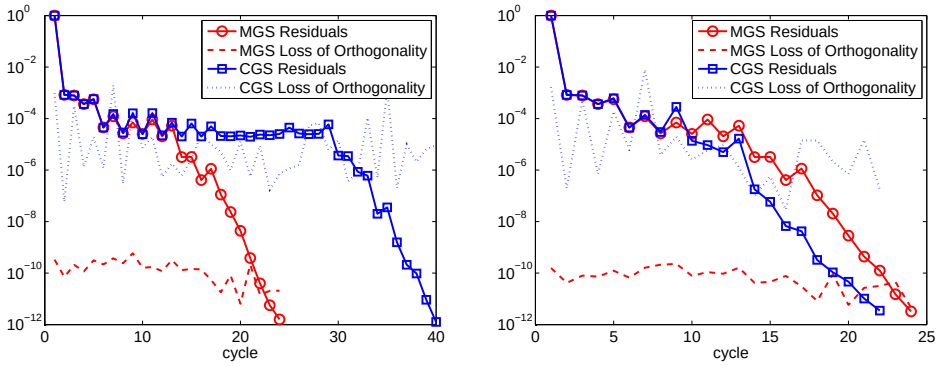
$$\sigma_{\min}\big(D^{\frac{1}{2}}V_m\big)\|\boldsymbol{r}^{(k)}\|_2 \leq \|\boldsymbol{r}^{(k)}\|_{V_m^H D V_m} \leq \sigma_{\max}\big(D^{\frac{1}{2}}V_m\big)\|\boldsymbol{r}^{(k)}\|_2,$$

where $\sigma_{\min}\big(D^{\frac{1}{2}}V_m\big)$ and $\sigma_{\max}\big(D^{\frac{1}{2}}V_m\big)$ are the smallest and largest singular values of $D^{\frac{1}{2}}V_m$. Assuming that at some cycle k the initial residuals agree, $\boldsymbol{r}^{(k-1)} = \widehat{\boldsymbol{r}}^{(k-1)}$, this shows that $\boldsymbol{r}^{(k)}$ and $\widehat{\boldsymbol{r}}^{(k)}$ can only differ significantly when the singular values of $D^{\frac{1}{2}}V_m$ start to differ from 1. In practice, the loss of orthogonality in $D^{\frac{1}{2}}V_m$ over a single cycle is typically modest, in particular when m is small, but the differences in the residuals $\boldsymbol{r}^{(k)}$ and $\widehat{\boldsymbol{r}}^{(k)}$ may build up over several cycles, leading to different convergence histories.

A detailed analysis of the effect of loss of orthogonality on WGMRES(m) convergence appears complicated due to the dynamic nature of the weighting and is beyond the scope of this manuscript. In most examples we have tested, the orthogonalization procedure did not seem to have a great impact on the convergence of WGMRES(m), and the convergence was not significantly delayed by loss of orthogonality, similarly to what is known about GMRES in the Euclidean inner product (see, for example, Simoncini and Szyld [34], Greenbaum *et al.* [18] and Drkošová *et al.* [10] for examples and analysis of the convergence of GMRES in finite precision). To give a (rare) example where the orthogonalization procedure indeed seems to have a pronounced effect on the convergence we apply WGMRES(70) to the matrix `fs_541_2` of size $N = 541$, which is available from the University of Florida Sparse Matrix Collection [9]. We choose $\boldsymbol{b}$ such that the solution is $\boldsymbol{x} = [1, 1, \ldots, 1]^T/\sqrt{N}$, and use the initial guess $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. Figure 4 shows the relative residuals $\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{b}\|_2$ at the end of each cycle, and the loss of D-orthogonality of V_m in (1) at the end of each weighted Arnoldi run, measured by (10). We note that the difference between $\sigma_{\min}\big(D^{\frac{1}{2}}V_m\big)$ and 1, and between $\sigma_{\max}\big(D^{\frac{1}{2}}V_m\big)$ and 1, is approximately equal to (10) for this problem.

Figure 4 (a) shows the residual history and loss of orthogonality when WGMRES(m) with explicitly computed nonstandard inner products is applied using Algorithm 1 (that uses MGS orthogonalization) and Algorithm 2 (that uses CGS orthogonalization). Similarly, Figure 4 (b) compares the convergence and loss of orthogonality of WGMRES(m) computed with the row- and column-scaled linear system using MGS orthogonalization (Algorithm 3) and CGS orthogonalization (Algorithm 4). We see that regardless of whether inner products are computed explicitly or implicitly, the MGS basis vectors are much closer to orthogonal than the CGS basis vectors.

In Figure 4 (a) we observe that when the inner products are computed directly the MGS version (Algorithm 1) converges significantly faster than the CGS variant (Algorithm 2). Indeed, the CGS version stagnates, although this stagnation is not associated with the complete loss of orthogonality of the basis vectors. Instead, it may be attributable to finite precision errors in the weight matrix (and, therefore, residual vector), introduced by the different orthogonalization procedures. Regardless of the cause, the stagnation highlights the



(a) (b)

Fig. 4: Relative residuals at the end of each cycle of WGMRES(70) using (a) Algorithms 1 and 2 and (b) Algorithms 3 and 4 with MGS (solid red with ∘) and CGS (solid blue with □) D-orthogonalization. Also plotted is $\|I_m - V_m^H D V_m\|_2$ at the end of each weighted Arnoldi cycle with MGS (dashed red) and CGS (dotted blue) orthogonalization.

large difference in convergence that rounding errors in the orthogonalization procedure can have on the computed WGMRES solution.

In contrast to Algorithms 1 and 2, it is clear from Figure 4 (b) that when the inner products are computed implicitly the MGS version (Algorithm 3) converges more slowly than the CGS version (Algorithm 4). This again shows that the implementation of the orthogonalization procedure can have a bearing on the convergence of weighted GMRES.

For this problem, it appears that neither orthogonalization procedure is definitively better than the other; instead performance is affected by rounding errors caused by the orthogonalization procedure in combination with the inner product computation (i.e., whether nonstandard inner products are computed explicitly, as in Algorithms 1 and 2, or implicitly, as in Algorithms 3 and 4).

In the following section we compare the numerical stability of MGS and CGS orthogonalization for a number of examples. We find that the performance of each depends on the problem and that neither is a clear winner.

4 Numerical experiments

In this section we compare different variants of WGMRES(m) with each other and with more established Krylov subspace methods, such as GMRES with deflated restarting (GMRES-DR) and BICGSTAB. The first five examples are those used by Essai [14] and are available from the University of Florida Sparse Matrix Collection [9]. For these examples we compare the four variants of WGMRES(m) in Algorithms 1–4 with each other and with GMRES-

m	GMRES-DR$(m,5)$	GMRES-DR$(m,10)$	Alg 1	Alg 2	Alg 3	Alg 4
40	20(23)	18(20)	21(35)	21(35)	21(35)	21(35)
50	14(44)	14(5)	15(32)	15(32)	15(32)	15(32)
60	11(49)	11(26)	12(43)	12(43)	12(43)	12(43)
70	10(36)	10(15)	11(12)	11(12)	11(12)	11(12)
80	9(12)	8(69)	10(5)	10(3)	10(5)	10(5)

Table 2: Number of cycles, with the number of iterations in the last cycle given in parentheses, for `add20`.

DR(m,ℓ). The last example compares two WGMRES(m) variants, unweighted GMRES(m), GMRES-DR(m,ℓ) and BICGSTAB over a large set of right-preconditioned test problems that is described below.

In all examples we take $\boldsymbol{x}^{(0)} = \mathbf{0}$ as the initial guess. A method is stopped when the relative residual $\|\boldsymbol{r}_j^{(k)}\|_2/\|\boldsymbol{b}\|_2$ has decreased to below 10^{-10}. If this tolerance is not reached after 100 cycles we terminate the method, which we denote by '—'. Iteration counts are given in the form $it_{\text{out}}(it_{\text{in}})$, where it_{out} is the number of cycles and it_{in} is the number of steps in the last cycle. In our notation GMRES-DR(m,ℓ) augments a Krylov subspace of dimension m with ℓ approximate Schur vectors. This makes sure that all GMRES variants require exactly m matrix-vector products with A in a cycle.

Example 4 (`add20`) *Our first test problem is* ***`add20`****, a matrix from a circuit simulation problem of dimension* 2 395 *with* 13 151 *nonzeros. The right-hand side* $\boldsymbol{b}$ *is a random vector. Table 2 shows the number of cycles required by GMRES-DR(m,ℓ) with* $\ell = 5, 10$ *as well as WGMRES(m) with Algorithms 1–4. We first observe that GMRES-DR requires slightly fewer cycles than WGMRES for all choices of* m *(*$m = 40, 50, \ldots, 80$*). Comparing the WGMRES variants, we find that for this example all algorithms require the same number of cycles, except for* $m = 80$ *for which Algorithm 2 converges slightly faster. Overall, however, it appears that for this example all methods behave stably.*

Example 5 (`bfwa782`) *The matrix* ***`bfwa782`*** *comes from an electromagnetics problem and is of dimension* 782 *with* 7 514 *nonzeros. Again* $\boldsymbol{b}$ *is a random vector. The number of cycles for the different methods are given in Table 3. We see that the difference between GMRES-DR and WGMRES is more pronounced, with the former requiring significantly fewer cycles for convergence. When* $m = 40$*, Algorithm 1 (with MGS orthogonalization) requires* 4 *fewer cycles than the other WGMRES variants. As the restart length increases, however, the different WGMRES implementations behave similarly and require identical numbers of cycles to converge.*

Example 6 (`fs_541_2`) *Our next example* ***`fs_541_2`*** *has dimension 541 and 4 282 nonzeros, with* $\boldsymbol{b}$ *a random vector. Table 4 again shows that GMRES-DR converges faster than WGMRES, with the latter failing to converge after 100 cycles when* $m = 40, 80$*. For restart lengths* $m = 50, 60$*, Algorithm 2 (with CGS orthogonalization) requires fewest matrix-vector products among the WGMRES variants. However, when* $m = 70$ *convergence for the same*

m	GMRES-DR$(m,5)$	GMRES-DR$(m,10)$	Alg 1	Alg 2	Alg 3	Alg 4
40	17(13)	9(2)	65(34)	69(20)	69(24)	69(24)
50	11(34)	7(23)	34(26)	34(26)	34(27)	34(27)
60	8(33)	6(18)	28(58)	28(58)	28(58)	28(58)
70	7(42)	5(36)	22(5)	22(5)	22(5)	22(5)
80	6(46)	4(70)	14(56)	14(56)	14(56)	14(56)

Table 3: Number of cycles, with the number of iterations in the last cycle given in parentheses, for `bfwa782`.

m	GMRES-DR$(m,5)$	GMRES-DR$(m,10)$	Alg 1	Alg 2	Alg 3	Alg 4
40	35(27)	29(21)	—	—	—	—
50	22(18)	20(49)	23(47)	23(33)	23(42)	23(42)
60	18(49)	16(53)	25(60)	24(24)	25(60)	25(60)
70	13(48)	12(62)	21(65)	37(60)	22(9)	22(9)
80	10(78)	9(76)	—	—	—	—

Table 4: Number of cycles, with the number of iterations in the last cycle given in parentheses, for `fs_541_2`.

m	GMRES-DR$(m,5)$	GMRES-DR$(m,10)$	Alg 1	Alg 2	Alg 3	Alg 4
40	60(40)	48(32)	50(24)	52(5)	54(29)	54(29)
50	42(27)	33(41)	35(33)	36(44)	33(2)	33(2)
60	33(9)	25(54)	28(31)	28(22)	27(60)	27(60)
70	25(51)	21(50)	26(26)	24(34)	26(37)	26(37)
80	20(51)	18(2)	20(49)	21(66)	20(9)	20(9)

Table 5: Number of cycles, with the number of iterations in the last cycle given in parentheses, for `memplus`.

algorithm is significantly delayed (see also Figure 4). Although CGS orthogonalization can give slightly faster convergence, it does not appear to be as stable as MGS orthogonalization.

Example 7 (`memplus`) *The matrix `memplus` is another circuit simulation problem, having dimension 17 758 and 99 147 nonzeros. The vector* $\boldsymbol{b}$ *is chosen at random. We see from Table 5 that for restart lengths smaller than* $m = 80$*, WGMRES(m) converges in fewer cycles than GMRES-DR(*$m, 5$*). GMRES-DR(*$m, 10$*) outperforms WGMRES, except when Algorithms 3 and 4 are applied with* $m = 50$*.*

Considering now the different WGMRES implementations, we see that there is some variation between them. For this problem WGMRES with Algorithms 3 and 4 require fewer cycles than WGMRES with Algorithms 1 or 2 when $m = 50, 60, 80$ *and more when* $m = 40, 70$*.*

Example 8 (`orsirr_1`) *A matrix from oil reservoir simulation, `orsirr_1` has dimension* $1\,030$ *and* $6\,858$ *nonzeros. We choose* $\boldsymbol{b}$ *to be a random vector. This is the one example for which WGMRES outperforms GMRES-DR(m,10) (see Table 6). For smaller restart lengths, WGMRES(m) also converges faster than GMRES-DR(m,5).*

m	GMRES-DR$(m,5)$	GMRES-DR$(m,10)$	Alg 1	Alg 2	Alg 3	Alg 4
40	70(9)	—	61(35)	61(21)	58(2)	69(37)
50	55(13)	—	46(46)	45(32)	48(11)	48(41)
60	34(55)	—	35(41)	34(42)	34(32)	35(30)
70	25(12)	—	28(39)	27(54)	28(18)	27(41)
80	—	—	22(32)	24(31)	23(8)	24(13)

Table 6: Number of cycles, with the number of iterations in the last cycle given in parentheses, for `orsirr_1`.

Different implementations of WGMRES again require different numbers of cycles for convergence, particularly for smaller restart lengths. The best performing WGMRES(m) variant changes with the restart length, so that overall none outperforms the others consistently. We note in particular that when $m = 40$ *Algorithm 3 converges fastest while Algorithm 4, which uses CGS orthogonalization, requires 11 more cycles, again highlighting the impact of the orthogonalization method on convergence.*

Example 9 (performance profile) *In order to get a general idea of whether WGMRES is a practical method when preconditioners are used, we compare GMRES, WGMRES with modified and classical Gram–Schmidt orthogonalization (see Algorithms 1 and 2), as well as GMRES-DR and BICGSTAB on a large number of problems from the University of Florida Sparse Matrix Collection. In contrast to Cao and Yu [7] we use right preconditioning, which minimizes the norm of the residual vector, rather than left preconditioning, which minimizes the norm of the preconditioned residual. The reason is that typically the residual vector is of greater interest than the preconditioned residual.*

Our method of comparison is as follows. We first retrieve all nonsymmetric matrices A *of sizes between* 10^4 *and* 10^6 *having no more than 15 nonzero elements per row on average. There are 220 such matrices. We then apply sparse reverse Cuthill–McKee reordering as implemented in Matlab's* `symrcm`*. Next we scale the columns of* A *to have unit Euclidean norm, followed by a scaling of the rows of* A *to unit norm. Our aim is to compute an ILU preconditioner with thresholding and pivoting via Matlab's* `ilu`*. For stability reasons we compute an ILU factorization of* $A+\sigma I$*, where* $\sigma = 10^{-12}$ *if all diagonal elements of* A *are zero, or* $\sigma = 10^{-12}\max\{|a_{ii}|\}$ *if some but not all diagonal elements* a_{ii} *of* A *zero, or* $\sigma = 0$ *otherwise. This procedure follows some recommendations given in [8]. We successively use a drop tolerance of* $10^{-3}, 10^{-4}, \ldots, 10^{-8}$*, and stop when the* U *factor of the factorization has a condition number below* 10^{15}*, so that it can be assumed numerically nonsingular. If this condition is not satisfiable with a drop tolerance of* 10^{-8}*, then the matrix* A *is skipped. The right-hand side vector* $\boldsymbol{b}$ *either comes with the matrix from the collection, or is generated randomly.*

We now run the Krylov methods GMRES(10), the two WGMRES(10) variants, GMRES-DR(10,5), BICGSTAB on the test matrices. A method is marked as failed if more than 50 restart cycles or 250 BICGSTAB iterations

are required to obtain a relative residual norm of 10^{-8}*. Note that in our notation GMRES-DR(10,5) requires* 10 *matrix-vector products (MVP) per cycle, exactly like GMRES(10) and WGMRES(10), and BICGSTAB requires 2 MVP per iteration, so that the maximal number of MVP is 500 for all methods. If all methods fail on a matrix, then this matrix is excluded from the test. Since the collection also contains many singular matrices, related to eigenvalue or least-squares problems, only 109 out of the 220 retrieved matrices are finally included in our test.*

The performance profile in Figure 5 allows us to compare the number of MVP needed for each method to converge across all test problems. More specifically, if for each linear system the performance ratio measures the number of MVP for the k-th method to converge to the number of MVP for the best performing method to converge, then the function $f_k(\alpha)$ *measures the fraction of problems in the test set for which the performance ratio of method* k *is less than or equal to* α*. Thus,* $\alpha = 1$ *shows the fraction of problems for which the k-th method requires the fewest MVP of all methods. Also,* $\lim_{\alpha\to\infty} f_k(\alpha)$ *indicates the number of failures.*

It is somewhat disappointing that WGMRES(10) is generally outperformed by GMRES(10). At least we observe that the CGS version of WGMRES does not perform considerably worse than the MGS version and, considering the potential computational savings outlined in Section 3, it seems that one should generally favour the CGS version when using WGMRES. We note that GMRES(10) and both variants of WGMRES(10) fail on at least 16 matrices from our test set; this failure rate is considerably higher than for GMRES-DR and BICGSTAB. Overall, GMRES-DR(10,5) requires fewest MVP in general, and thereby outperforms all other methods under consideration. It is, however, less robust than BICGSTAB, the latter of which fails for 3 matrices only but typically requires the most MVP. To summarize, we believe that WGMRES should not be used in combination with preconditioners, although we are aware that for some examples it may perform satisfactorily.

5 Conclusions

The weighted GMRES variant presented by Essai has recently gained interest for solving linear systems. This method is justified by a heuristic that emphasizes large residual components via a weighted inner product. With the help of simple model problems we have given insight into how weighting affects the distribution of harmonic Ritz values, or how it affects entries in the residual vector after each cycle. For example, in one case where the harmonic Ritz values appeared in cyclic pairs on the spectral interval of a matrix, weighting had the effect of "randomizing" these harmonic Ritz values and thereby covering the spectral interval more evenly. This led to an improved convergence of WGMRES compared to the linear convergence observed for GMRES on the same example.

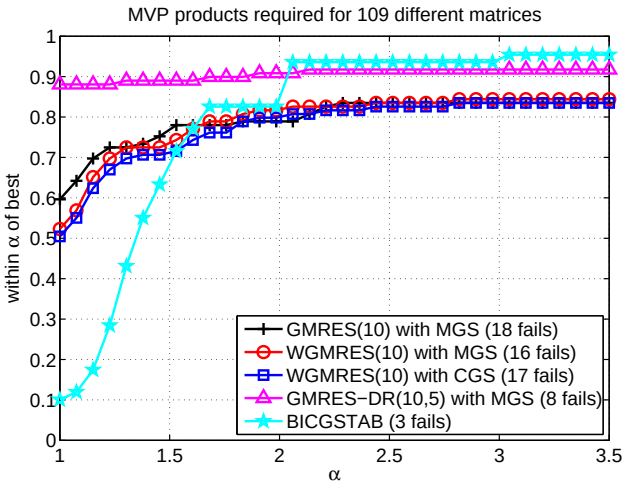


Fig. 5: Performance profile of matrix vector products required by various Krylov methods applied to 109 matrices from the University of Florida Sparse Matrix Collection with an ILU preconditioner.

We presented four different implementations of weighted GMRES and compared their numerical cost and stability properties. Our numerical results suggest that these variants generally converge similarly to Essai's original implementation, but may require fewer arithmetic operations. However, for at least one matrix and restart length each of Algorithms 1–4 outperforms the others, so that no method is definitively better. Occasionally these differences are significant, indicating the presence of instabilities. Our investigations suggest that these are not due to a rapid loss of orthogonality. However, even a small loss of orthogonality can alter the residual obtained, so that differences build up over several cycles. The problem is exacerbated by the sensitivity of the Krylov subspace itself to perturbations in the residual. We stress, however, that these difficulties with finite precision and loss of orthgonality are inherent to the orthogonalization procedure and not weighting itself, and hence are equally relevant to unweighted (restarted) GMRES.

When applied to some unpreconditioned problems, WGMRES(m) can outperform GMRES(m). A test run with a large number of matrices from the University of Florida Sparse Matrix Collection revealed, similarly to observations made in [7], that weighted GMRES is typically outperformed by GMRES if a preconditioner is used. In addition, we have compared these methods with other state-of-the-art Krylov methods like GMRES-DR (GMRES with deflated restarting) and BICGSTAB. GMRES-DR required fewest matrix-vector products, whereas BICGSTAB appeared to be the most robust method in our test, at the cost of requiring the most matrix-vector products. One advantage of

WGMRES(m) over GMRES-DR(m, ℓ) is that there is no parameter ℓ to be chosen.

We find that, although weighted GMRES may outperform unweighted GMRES for some examples, in general this method is not competitive with other Krylov subspace methods like BICGSTAB or deflated GMRES, in particular when preconditioners are used. If, for whatever reason, weighted GMRES needs to be used with a moderate restart length, then the CGS version (Algorithm 2) presented in Section 3 is recommended.

Acknowledgements We are grateful to Andy Wathen and the two anonymous referees for their valuable comments and suggestions.

References

1. Afanasjew, M., Eiermann, M., Ernst, O.G., Güttel, S.: A generalization of the steepest descent method for matrix functions. Electron. Trans. Numer. Anal. **28**, 206–222 (2008)
2. Akaike, H.: On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. Ann. Inst. Statist. Math. **11**, 1–16 (1959)
3. Arioli, M., Pták, V., Strakoš, Z.: Krylov sequences of maximal length and convergence of GMRES. BIT **38**, 636–643 (1998)
4. Ashby, S.F., Holst, M.J., Manteuffel, T.A., Saylor, P.E.: The role of the inner product in stopping criteria for conjugate gradient iterations. BIT **41**, 26–52 (2001)
5. Baker, A.H., Jessup, E.R., Manteuffel, T.: A technique for accelerating the convergence of restarted GMRES. SIAM J. Matrix Anal. Appl. **26**, 962–984 (2005)
6. Brown, P.: A theoretical comparison of the Arnoldi and GMRES algorithms. SIAM J. Sci. Stat. Comput. **12**, 58–78 (1991)
7. Cao, Z.H., Yu, X.Y.: A note on weighted FOM and GMRES for solving nonsymmetric linear systems. Appl. Math. Comput. **151**, 719–727 (2004)
8. Chow, E., Saad, Y.: Experimental study of ILU preconditioners for indefininte matrices. J. Comput. Appl. Math. **86**, 387–414 (1997)
9. Davis, T.A., Hu, Y.: The University of Florida Sparse Matrix Collection. ACM Trans. Math. Softw. **38**, 1:1–1:25 (2011)
10. Drkošová, J., Greenbaum, A., Rozložník, M., Strakoš, Z.: Numerical stability of GMRES. BIT **35**, 309–330 (1995)
11. Duintjer Tebbens, J., Meurant, G.: Any Ritz value behavior is possible for Arnoldi and for GMRES with any convergence curve. SIAM J. Matrix Anal. Appl. **33**(3), 958–978 (2012)
12. Eiermann, M., Ernst, O.G., Schneider, O.: Analysis of acceleration strategies for restarted minimal residual methods. J. Comput. Appl. Math. **123**, 261–292 (2000)
13. Embree, M.: The tortoise and the hare restart GMRES. SIAM Rev. **45**, 259–266 (2003)
14. Essai, A.: Weighted FOM and GMRES for solving nonsymmetric linear systems. Numer. Algorithms **18**, 277–292 (1998)
15. Forsythe, G.E., Motzkin, T.S.: Asymptotic properties of the optimum gradient method (abstract). Bull. Amer. Math. Soc. **57**, 183 (1951)
16. Goossens, S., Roose, D.: Ritz and harmonic Ritz values and the convergence of FOM and GMRES. Numer. Linear Algebra Appl. **6**, 281–293 (1997)
17. Greenbaum, A., Pták, V., Strakoš, Z.: Any nonincreasing convergence curve is possible for GMRES. SIAM J. Matrix Anal. Appl. **17**, 465–469 (1996)
18. Greenbaum, A., Rozložník, M., Strakoš, Z.: Numerical behaviour of the modified Gram-Schmidt GMRES implementation. BIT **37**, 706–719 (1997)
19. Heyouni, M., Essai, A.: Matrix Krylov subspace methods for linear systems with multiple right-hand sides. Numer. Algorithms **40**, 137–156 (2005)

20. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, New York, NY (1990)
21. Jing, Y.F., Huang, T.Z.: Restarted weighted full orthogonalization method for shifted linear systems. Comput. Math. Appl. **57**, 1583–1591 (2009)
22. Liesen, J., Strakoš, Z.: Convergence of GMRES for tridiagonal Toeplitz matrices. SIAM J. Matrix Anal. Appl. **26**, 233–251 (2004)
23. Meijerink, J.A., van der Vorst, H.A.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. Comput. **31**, 148–162 (1977)
24. Morgan, R.B.: A restarted GMRES method augmented with eigenvectors. SIAM J. Matrix Anal. Appl. **16**, 1154–1171 (1995)
25. Morgan, R.B.: GMRES with deflated restarting. SIAM J. Sci. Comput. **24**, 20–37 (2002)
26. Najafi, H.S., Zareamoghaddam, H.: A new computational GMRES method. Appl. Math. Comput. **199**, 527–534 (2008)
27. Niu, Q., Lu, L., Zhou, J.: Accelerate weighted GMRES by augmenting error approximations. Int. J. Comput. Math. **87**, 2101–2112 (2010)
28. Pestana, J., Wathen, A.J.: On the choice of preconditioner for minimum residual methods for non-hermitian matrices. J. Comput. Appl. Math. **249**(0), 57–68 (2013). DOI 10.1016/j.cam.2013.02.020. URL http://www.sciencedirect.com/science/article/pii/S0377042713000903
29. Rice, J.R.: Experiments on Gram-Schmidt orthogonalization. Math. Comput. **20**, 325–328 (1966)
30. Rozložník, M., Tůma, M., Smoktunowicz, A., Kopal, J.: Numerical stability of orthogonalization methods with a non-standard inner product. BIT Numer. Math. **52**, 1035–1058 (2012)
31. Saad, Y.: Analysis of augmented Krylov subspace methods. SIAM J. Matrix Anal. Appl. **18**, 435–449 (1997)
32. 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)
33. 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)
34. Simoncini, V., Szyld, D.B.: The effect of non-optimal bases on the convergence of Krylov subspace methods. Numer. Math. **100**, 711–733 (2005)
35. Simoncini, V., Szyld, D.B.: On the occurence of superlinear convergence of exact and inexact Krylov subspace methods. SIAM Rev. **47**, 247–272 (2005)
36. Vecharynski, E., Langou, J.: The cycle-convergence of restarted GMRES for normal matrices is sublinear. SIAM J. Sci. Comput. **32**, 186–196 (2010)
37. Vecharynski, E., Langou, J.: Any admissible cycle-convergence behavior is possible for restarted GMRES at its initial cycles. Numer. Linear Algebra Appl. **18**, 499–511 (2011)
38. Wang, Z., Qi, J., Liu, C., Li, Y.: Weighted harmonic Arnoldi method for large interior eigenproblems. World Acad. Sci. Eng. Technol. **56**, 1476–1479 (2011)
39. Zavorin, I., O'Leary, D.P., Elman, H.: Complete stagnation of GMRES. Linear Algebra Appl. **367**, 165–183 (2003)
40. Zhong, B., Morgan, R.B.: Complementary cycles of restarted GMRES. Numer. Linear Algebra Appl. **15**, 559–571 (2008)
41. Zhong, B.J.: A product hybrid GMRES algorithm for nonsymmetric linear systems. J. Comput. Math. **23**, 83–92 (2005)
42. Zhong, H.X., Wu, G.: Thick restarting the weighted harmonic Arnoldi algorithm for large interior eigenproblems. Int. J. Comput. Math. **88**, 994–1012 (2011)