LOW-RANK-MODIFIED GALERKIN METHODS FOR THE LYAPUNOV EQUATION[*]

KATHRYN LUND[†] AND DAVIDE PALITTA[‡]

Abstract. Of all the possible projection methods for solving large-scale Lyapunov matrix equations, Galerkin approaches remain much more popular than minimal residual ones. This is mainly due to the different nature of the projected problems stemming from these two families of methods. While a Galerkin approach leads to the solution of a low-dimensional matrix equation per iteration, a matrix least-squares problem needs to be solved per iteration in a minimal residual setting. The significant computational cost of these least-squares problems has steered researchers towards Galerkin methods in spite of the appealing properties of minimal residual schemes. In this paper we introduce a framework that allows for modifying the Galerkin approach by low-rank, additive corrections to the projected matrix equation problem with the two-fold goal of attaining monotonic convergence rates similar to those of minimal residual schemes while maintaining essentially the same computational cost of the original Galerkin method. We analyze the well-posedness of our framework and determine possible scenarios where we expect the residual norm attained by two low-rank-modified variants to behave similarly to the one computed by a minimal residual technique. A panel of diverse numerical examples shows the behavior and potential of our new approach.

Key words. Lyapunov equation, matrix equation, block Krylov subspace, model order reduction

AMS subject classifications. 65F45, 65F50, 65F10, 65N22, 65J10

1. Introduction. We are interested in the numerical solution of large-scale Lyapunov equations of the form

$$(1.1) \qquad AX + XA^* + \boldsymbol{C}\boldsymbol{C}^* = 0,$$

where $A \in \mathbb{C}^{n \times n}$ is large and sparse and $\boldsymbol{C} \in \mathbb{C}^{n \times r}$, $r \ll n$, has low rank. Throughout the paper, we use plain uppercase italic letters to refer to square matrices (e.g., A), boldface uppercase italic letters to refer to *block vectors* (e.g., $\boldsymbol{C}$), and boldface lowercase italic letters for column vectors (e.g., $\boldsymbol{c}$). When a square matrix has block structure, it is denoted by a script (calligraphic) letter (e.g., $\mathcal{H}$), and the concatenation of block vectors into a basis is denoted with a bold script (calligraphic) letter (e.g., $\boldsymbol{\mathcal{V}}$). Matrices with Kronecker structure are formatted with upright bold sans serif font (e.g., $\mathbf{H}$).

The Lyapunov equation (1.1) is encountered in many applications. For instance, in some model reduction [1] and robust/optimal control strategies [37], a Lyapunov equation, or a sequence of such equations, has to be solved. Moreover, the discretization of certain elliptic partial differential equations (PDEs) leads to an algebraic problem that can be often represented in terms of a Lyapunov equation; see, e.g., [29]. We refer the interested reader to the survey papers [7, 33] and the references therein for more details about the aforementioned applications and further research areas where Lyapunov equations play an important role.

We assume the matrix A to be stable: its spectrum is contained in the left-half open complex plane $\mathbb{C}^-$. Therefore, the solution X to (1.1) is Hermitian positive semidefinite [35]. Moreover, it is well known that the singular values of X rapidly decay to zero, if certain further assumptions on A are considered; see, e.g., [3, 31]. In this case, the solution X can be well approximated by a low-rank matrix $\boldsymbol{Z}\boldsymbol{Z}^* \approx X$, $\boldsymbol{Z} \in \mathbb{C}^{n \times t}$, $t \ll n$, and the computation of the

---

[†]Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (lund@mpi-magdeburg.mpg.de), ORCID: 0000-0001-9851-6061.

[‡]Università di Bologna, Centro AM², Dipartimento di Matematica, Piazza di Porta S. Donato 5, 40127 Bologna, Italy (davide.palitta@unibo.it), ORCID:0000-0002-6987-4430.

low-rank factor $\boldsymbol{Z}$ is the task of the so-called *low-rank methods*. Numerous, diverse algorithms belong to this broad class of solvers. Some examples are projection methods [10, 32], low-rank alternating direction implicit (ADI) methods [25, 26], sign function methods [4, 5], and Riemannian optimization methods [36]. A more complete list of low-rank solvers can be found in the surveys [7, 33].

We focus on projection methods, and we propose a novel framework for solving (1.1). Given a suitable subspace $\mathscr{K}$ and a matrix $\boldsymbol{\mathcal{V}}_m \in \mathbb{C}^{n \times mr}$, $mr < n$, whose columns represent an orthonormal basis of $\mathscr{K}$, projection methods seek an approximate solution X_m to (1.1) of the form $X_m = \boldsymbol{\mathcal{V}}_m Y_m \boldsymbol{\mathcal{V}}_m^*$. The square matrix $Y_m \in \mathbb{C}^{mr \times mr}$ can be computed in different ways. Most of the schemes available in the literature compute Y_m by imposing a Galerkin condition on the residual matrix $R_m = AX_m + X_m A^* - \boldsymbol{C}\boldsymbol{C}^*$. Imposing such a condition, namely $\boldsymbol{\mathcal{V}}_m^* R_m \boldsymbol{\mathcal{V}}_m = 0$, is equivalent to computing Y_m as the solution of the *projected* Lyapunov equation

$$(1.2) \qquad \mathcal{H}_m Y + Y \mathcal{H}_m^* + \boldsymbol{E}_1 \Gamma \Gamma^* \boldsymbol{E}_1^* = 0,$$

where $\mathcal{H}_m = \boldsymbol{\mathcal{V}}_m^* A \boldsymbol{\mathcal{V}}_m$, $\boldsymbol{V}_1 \Gamma$ is the economic QR factorization of $\boldsymbol{C}$, and $\boldsymbol{E}_1 = e_1 \otimes I_r$; see, e.g., [10, 32].

A less explored alternative consists of computing Y_m by imposing a minimal residual (MR) condition, namely, we compute Y_m as follows:

$$(1.3) \qquad Y_m = \arg\min_Y \|\underline{\mathcal{H}}_m Y [I_m \ \ 0] + [I_m \ \ 0]^* Y \underline{\mathcal{H}}_m^* + \underline{\boldsymbol{E}}_1 \Gamma \Gamma^* \underline{\boldsymbol{E}}_1 \|_{\mathrm{F}},$$

where $\underline{\mathcal{H}}_m = \boldsymbol{\mathcal{V}}_{m+1}^* A \boldsymbol{\mathcal{V}}_m$ and $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm;[1] see, e.g., [19, 27].

In spite of their appealing minimization property, MR methods for matrix equations are not commonly adopted. The solution of the matrix least-squares problem in (1.3) can be remarkably more expensive than solving (1.2); see [27], as well as our own numerical results in Section 5. Moreover, the matrix Y_m computed by (1.3) may be indefinite (in floating-point as well as in exact arithmetic), meaning that the computed approximation $X_m = \boldsymbol{\mathcal{V}}_m Y_m \boldsymbol{\mathcal{V}}_m^*$ is as well, even though the solution X to (1.1) is semidefinite. For further discussion about this peculiar drawback of MR methods for Lyapunov equations, see [30, Section 5], and for a similar approach for algebraic Riccati equations with global Krylov subspace methods, see [20].

Instead, we propose to compute Y_m as the solution of a low-rank modification (LRM) of (1.2), whereby the coefficient matrix $\mathcal{H}_m$ is replaced by $\mathcal{H}_m + \mathcal{M}$ for a certain low-rank matrix $\mathcal{M}$. This approach is inspired by a similar technique for univariate matrix functions [13]. In particular, we consider the so-called "harmonic modification" $\mathcal{M}$, which addresses some computational issues of projection methods while maintaining interesting theoretical features. We refer to the resulting method as a pseudo-minimal residual (PMR) method, given that it appears to closely approximate the MR approximation for (1.1) with symmetric A.

In Section 2 we introduce a bivariate LRM framework that generalizes the univariate one from [13]. We precisely define the PMR method in Section 3, quantify how close it is to MR via an eigenvalue analysis, and propose another LRM that minimizes the residual over a structured space of Kronecker sums. We show how general LRM approaches can be combined with the compress-and-restart strategy of [24] in Section 4. Results of numerical experiments are presented in Section 5, and we summarize our findings and contributions in Section 6.

---

[1]Computing Y_m as in (1.3) is equivalent to imposing a Petrov–Galerkin condition on the residual for a specific choice of the test space; see, e.g., [30, Section 5].

2. Low-rank modification framework. An LRM framework for block Krylov subspace methods (KSMs) has been previously developed for matrix functions; see, in particular, [13]. While much of the framework transfers easily to matrix equations, we must take some care, since we are moving from a univariate framework to a bivariate one.

We begin by defining the mth *block Krylov subspace* for A and $\boldsymbol{C}$ in terms of the block span, denoted here as $\mathrm{span}^\square$:

$$\mathscr{K}_m(A, \boldsymbol{C}) := \mathrm{span}^\square\{\boldsymbol{C}, A\boldsymbol{C}, \dots, A^{m-1}\boldsymbol{C}\} = \left\{ \sum_{j=0}^{m-1} A^j \boldsymbol{C}\, \Gamma_j : \{\Gamma_j\}_{j=0}^d \subset \mathbb{C}^{r \times r} \right\}.$$

Note that $\mathscr{K}_m(A, \boldsymbol{C}) \subset \mathbb{C}^{n \times r}$, i.e., its elements are *block vectors*. This is in contrast to the column span treatment of block KSMs, which is often used for solving linear systems with a column right-hand side. For a foundational resource on the different interpretations of block KSMs and how they relate to each other, see [15].

Generating $\mathscr{K}_m(A, \boldsymbol{C})$ via the block Arnoldi process gives rise to the *block Arnoldi relation*

$$(2.1) \qquad A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_{m+1}\underline{\mathcal{H}}_m = \boldsymbol{\mathcal{V}}_m\mathcal{H}_m + \boldsymbol{V}_{m+1}H_{m+1,m}\boldsymbol{E}_m^*,$$

where $\boldsymbol{\mathcal{V}}_{m+1} \in \mathbb{C}^{n \times (m+1)r}$ is orthonormal,

$$\underline{\mathcal{H}}_m = \begin{bmatrix} \mathcal{H}_m \\ H_{m+1,m}\boldsymbol{E}_m^* \end{bmatrix}$$

is block upper Hessenberg, and $\boldsymbol{E}_m = \boldsymbol{e}_m \otimes I_r$ is a unit block vector.

With the block Arnoldi decomposition, we can directly write the Galerkin approximation to (1.1) over $\mathscr{K}_m(A, \boldsymbol{C})$. First, we project (1.1) down and solve the $mr \times mr$ problem

$$(2.2) \qquad \mathcal{H}_m Y + Y\mathcal{H}_m^* + \boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^* = 0.$$

Letting $Y_m^{\mathsf{G}} \in \mathbb{C}^{mr \times mr}$ denote the solution to (2.2), we define the Galerkin approximation to (1.1) as

$$X_m^{\mathsf{G}} := \boldsymbol{\mathcal{V}}_m Y_m^{\mathsf{G}} \boldsymbol{\mathcal{V}}_m^*.$$

We can make LRMs to the Galerkin solution in much the same way as for matrix functions by simply replacing $\mathcal{H}_m$ in (2.2) with $\mathcal{H}_m + \mathcal{M}$, where $\mathcal{M} = \boldsymbol{M}\boldsymbol{E}_m^*$ for some matrix $\boldsymbol{M} \in \mathbb{C}^{mr \times r}$. It is possible to choose $\mathcal{M}$ so that $\mathcal{H}_m + \mathcal{M}$ has harmonic Ritz values – as in the generalized minimal residual method – or so that certain Ritz values are prescribed; see, e.g., [11, 12, 13] and Section 3. Most importantly, it is possible to choose $\mathcal{M}$ so that the resulting approximation is still in the product of the Krylov subspaces $\mathscr{K}_m(A, \boldsymbol{C}) \times \mathscr{K}_m(A, \boldsymbol{C})$, which Lemma 2.2 demonstrates.

Letting Y_m^{mod} denote the solution to the modified problem,

$$(2.3) \qquad (\mathcal{H}_m + \mathcal{M})Y + Y(\mathcal{H}_m + \mathcal{M})^* + \boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^* = 0,$$

leads to an approximate solution $X_m^{\mathsf{mod}} := \boldsymbol{\mathcal{V}}_m Y_m^{\mathsf{mod}} \boldsymbol{\mathcal{V}}_m^*$, whose residual norm can be cheaply computed, as shown in the next proposition.

PROPOSITION 2.1. *Let Y_m^{mod} be the solution to the Lyapunov equation* (2.3). *Then the residual matrix $R_m^{\mathsf{mod}} = AX_m^{\mathsf{mod}} + X_m^{\mathsf{mod}}A^* + \boldsymbol{C}\boldsymbol{C}^*$ can be written as*

$$(2.4) \qquad R_m^{\mathsf{mod}} = \boldsymbol{\mathcal{V}}_{m+1}G_m \begin{bmatrix} & I_r & \\ I_r & & -I_r \\ & -I_r & \end{bmatrix} G_m^* \boldsymbol{\mathcal{V}}_{m+1}^* \in \mathbb{C}^{n \times n},$$

Algorithm 1 LRM Galerkin approach for Lyapunov equations.

1: **input** $A \in \mathbb{C}^{n \times n}$, $\boldsymbol{C} \in \mathbb{C}^{n \times r}$, $m_{\max}$, ε
2: **output** $\boldsymbol{\mathcal{V}}_m = \begin{bmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 & \cdots & \boldsymbol{V}_m \end{bmatrix} \in \mathbb{C}^{n \times mr}$, $Y_m^{\text{mod}} \in \mathbb{C}^{mr \times mr}$
3: Compute an economic QR factorization of $\boldsymbol{C} = \boldsymbol{V}_1 \Gamma$
4: **for** $m = 1, \ldots, m_{\max}$ **do**
5:     Compute the next basis block $\boldsymbol{V}_{m+1}$ and update $\mathcal{H}_m$
6:     Compute Y_m^{mod} as the solution of (2.3)
7:     Compute $\|R_m^{\text{mod}}\|_{\text{F}}$
8:     **if** $\|R_m^{\text{mod}}\|_{\text{F}} \leq \varepsilon \cdot \|\boldsymbol{C}^* \boldsymbol{C}\|_{\text{F}}$ **then**
9:         **return** $\boldsymbol{\mathcal{V}}_m$ and Y_m^{mod}
10:     **end if**
11: **end for**

---

where

$$G_m = \begin{bmatrix} \boldsymbol{E}_{m+1} H_{m+1,m} & \underline{\mathcal{I}} Y_m^{\text{mod}} \boldsymbol{E}_m & \underline{\mathcal{I}} \boldsymbol{M} \end{bmatrix} \in \mathbb{C}^{n \times 3r}$$

and

$$\underline{\mathcal{I}} = \begin{bmatrix} I_{mr} \\ 0 \end{bmatrix} \in \mathbb{R}^{(m+1)r \times mr}.$$

Moreover,

(2.5)
$$\begin{aligned} \|R_m^{\text{mod}}\|_F^2 = 2(\|Y_m^{\text{mod}} \boldsymbol{E}_m H_{m+1,m}^*\|_F^2 + \|Y_m^{\text{mod}} \boldsymbol{E}_m \boldsymbol{M}^*\|_F^2 \\ + 2 \cdot \text{trace}((\boldsymbol{E}_m^* Y_m^{\text{mod}} \boldsymbol{M})^2)). \end{aligned}$$

Proof. Thanks to the block Arnoldi relation (2.1) and since $X_m^{\text{mod}} := \boldsymbol{\mathcal{V}}_m Y_m^{\text{mod}} \boldsymbol{\mathcal{V}}_m^*$,

$$\begin{aligned} R_m^{\text{mod}} &= A X_m^{\text{mod}} + X_m^{\text{mod}} A^* - \boldsymbol{C}\boldsymbol{C}^* \\ &= \boldsymbol{\mathcal{V}}_m((\mathcal{H}_m + \mathcal{M})Y_m^{\text{mod}} + Y_m^{\text{mod}}(\mathcal{H}_m + \mathcal{M})^* - \boldsymbol{E}_1 \Gamma\Gamma^* \boldsymbol{E}_1)\boldsymbol{\mathcal{V}}_m^* \\ &\quad + \boldsymbol{V}_{m+1} H_{m+1,m} \boldsymbol{E}_m^* Y_m^{\text{mod}} \boldsymbol{\mathcal{V}}_m^* + \boldsymbol{\mathcal{V}}_m Y_m^{\text{mod}} \boldsymbol{E}_m H_{m+1,m}^* \boldsymbol{V}_{m+1}^* \\ &\quad - \boldsymbol{\mathcal{V}}_m \mathcal{M} Y_m^{\text{mod}} \boldsymbol{\mathcal{V}}_m^* - \boldsymbol{\mathcal{V}}_m Y_m^{\text{mod}} \mathcal{M}^* \boldsymbol{\mathcal{V}}_m^* \\ &= \boldsymbol{\mathcal{V}}_{m+1}(\boldsymbol{E}_{m+1} H_{m+1,m} \boldsymbol{E}_m^* Y_m^{\text{mod}} \underline{\mathcal{I}}^* + \underline{\mathcal{I}} Y_m^{\text{mod}} \boldsymbol{E}_m H_{m+1,m}^* \boldsymbol{E}_{m+1}^* \\ &\quad - \underline{\mathcal{I}} \mathcal{M} Y_m^{\text{mod}} \underline{\mathcal{I}}^* - \underline{\mathcal{I}} Y_m^{\text{mod}} \mathcal{M}^* \underline{\mathcal{I}}^*)\boldsymbol{\mathcal{V}}_{m+1}^*. \end{aligned}$$

The results follow by plugging the low-rank form $\mathcal{M} = \boldsymbol{M}\boldsymbol{E}_m^*$ into the expression above. Similarly, by recalling that $\boldsymbol{E}_{m+1}^* \underline{\mathcal{I}} = 0$, $\underline{\mathcal{I}}^* \underline{\mathcal{I}} = I_{mr}$, and $\boldsymbol{E}_{m+1}^* \boldsymbol{E}_{m+1} = I_r$, a direct computation shows (2.5). □

In Algorithm 1 the low-rank-modified Galerkin approach for (1.1) is outlined, where $m_{\max}$ denotes the maximum number of block basis vectors and $\varepsilon > 0$ is the desired relative residual tolerance. Note that the algorithm reduces to the standard Galerkin approach whenever $\mathcal{M} = 0$.

The connection between matrix equations and bivariate functions of matrices has been well established by Kressner [22, 23]. In fact, the LRM framework holds for bivariate matrix functions in general, and thus a wide array of other applications, such as Fréchet derivatives, the Stein equation, and time-limited and frequency-limited balanced truncation model reduction [6, 14].

Let $\mathbb{P}_{k,\ell}(\mathbb{C}, \mathbb{C})$ denote the space of bivariate matrix polynomials of degree k in the first variable and degree ℓ in the second. Like univariate matrix functions [17], bivariate matrix

functions can be defined as Hermite interpolating polynomials evaluated on the matrices A and B (of compatible dimension). In more detail,

$$f\{A,B\}(\boldsymbol{CD}^*) := p_{k,l}\{A,B\}(\boldsymbol{CD}^*) := \sum_{i=0}^{k}\sum_{j=0}^{\ell}\alpha_{ij}A^i\boldsymbol{C}(B^j\boldsymbol{D})^*,$$

for some degrees k and ℓ and scalars α_{ij}, as long as assumptions on the eigenvalues of A and B are met; see [22, Definition 2.3]. In particular, the solution X to the Lyapunov equation (1.1) can be expressed as the evaluation of the function $f(x,y) = 1/(x+y)$, i.e., $X = f\{A,A^*\}(\boldsymbol{CC}^*)$.

We state an alternative to [23, Lemma 2] that accounts for LRMs for Lyapunov equations.

LEMMA 2.2. *Let $\mathcal{H}$ and $\boldsymbol{\mathcal{V}}$ denote the Arnoldi matrix and basis from (2.1). For all bivariate polynomials q of degrees $m-1, m-1$ or less, it holds that*

$$q(A,A^*)(\boldsymbol{CC}^*) = \boldsymbol{\mathcal{V}} \cdot q\{\mathcal{H}+\mathcal{M},(\mathcal{H}+\mathcal{M})^*\}(\boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1) \cdot \boldsymbol{\mathcal{V}}^*,$$

as long as $\mathcal{M} = \boldsymbol{M}\boldsymbol{E}_m^*$ for some $\boldsymbol{M} \in \mathbb{C}^{n\times r}$.

The proof follows from a straightforward combination of [23, Lemma 2] and [13, Theorem 2.7]. The extension to Sylvester equations is straightforward; one just has to be careful with the dimensions for the second Krylov subspace.

In work being developed in parallel, it has been shown that equations of the form (2.2) arise in Krylov subspace techniques combined with sketching; see [28]. However, there is a fundamental difference between this setting and our novel low-rank-modified Galerkin (G) method. Indeed, while in the former the form of $\boldsymbol{M}$ is dictated solely by the use of sketching, here we choose $\boldsymbol{M}$ to try to meet a target behavior in our solver.

3. A pseudo-minimal residual method. The generalized minimal residual method (GMRES) is a popular approach for linear systems, precisely because of the guaranteed monotonic behavior of the residual [16]. It can be easily shown that (block) GMRES is equivalent in exact arithmetic to an LRM of the full orthogonalization method (FOM); see [34, Theorem 3.3] or [11, 13]. By assuming $\mathcal{H}_m$ is non-singular and choosing

$$\boldsymbol{M}^{\text{PMR}} := \mathcal{H}_m^{-*}\boldsymbol{E}_m H_{m+1,m}^* H_{m+1,m}$$

and $\mathcal{M}^{\text{PMR}} := \boldsymbol{M}^{\text{PMR}}\boldsymbol{E}_m^*$, it holds that

$$\boldsymbol{\mathcal{V}}_{m+1}\underline{\mathcal{H}}_m^\dagger\boldsymbol{E}_1 = \boldsymbol{\mathcal{V}}_{m+1}(\mathcal{H}_m+\mathcal{M}^{\text{PMR}})^{-1}\boldsymbol{E}_1,$$

which is also the GMRES solution for the linear system $A\boldsymbol{X} = \boldsymbol{C}$.

The modification $\mathcal{M}^{\text{PMR}}$ in (2.3) does not guarantee a minimal residual approximation to (1.1), but it comes close in many cases, leading to our moniker "pseudo-minimal residual" (PMR). The well-posedness of (2.3) is difficult to prove in general, since predicting the impact of the low-rank modification $\mathcal{M}$ on the spectral properties of $\mathcal{H}_m$ is seldom doable. In the next proposition, we show that this is possible in the case of $\mathcal{M}^{\text{PMR}}$.

PROPOSITION 3.1. *Let A have its field of values in the left half of the complex plane, denoted here as $\mathbb{C}^-$.[2] Then for all m such that $mr < n$ and such that the block Arnoldi algorithm does not break down, $\mathcal{H}_m + \mathcal{M}^{\text{PMR}}$ is stable (i.e., its eigenvalues have negative real part), implying in particular that equation (2.3) has a unique, positive semidefinite solution.*

---

[2]This property has been denoted as "negative definiteness" or "negative realness" in the literature, but with varying consistency, and we spell it out to avoid confusion.

Proof. The proof is essentially a special case of part of the proof of [13, Corollary 4.4]. For ease of reading, we reproduce it in our setting here.

Fix m such that $mr < n$ and assume that the Arnoldi algorithm does not break down. It follows from (2.1) and the assumption on A that $\mathcal{H}_m$ also has its field of values in $\mathbb{C}^-$. Now let $(\lambda, \boldsymbol{x})$ be an eigenpair of $\mathcal{H}_m + \mathcal{M}^{\text{PMR}}$, namely,

$$(3.1) \qquad (\mathcal{H}_m + \mathcal{M}^{\text{PMR}})\boldsymbol{x} = \lambda\boldsymbol{x}.$$

Recall that $\mathcal{M}^{\text{PMR}} = \boldsymbol{M}^{\text{PMR}}\boldsymbol{E}_m^* = \mathcal{H}_m^{-*}\boldsymbol{E}_m H_{m+1,m}^* H_{m+1,m}\boldsymbol{E}_m^*$. Left-multiplying (3.1) by $\mathcal{H}_m^*$ leads to

$$(\mathcal{H}_m^*\mathcal{H}_m + \boldsymbol{E}_m H_{m+1,m}^* H_{m+1,m}\boldsymbol{E}_m^*)\boldsymbol{x} = \lambda\mathcal{H}_m^*\boldsymbol{x}.$$

Therefore,

$$(3.2) \qquad \lambda = \frac{\boldsymbol{x}^*\mathcal{H}_m^*\mathcal{H}_m\boldsymbol{x} + \boldsymbol{x}^*\boldsymbol{E}_m H_{m+1,m}^* H_{m+1,m}\boldsymbol{E}_m\boldsymbol{x}}{\boldsymbol{x}^*\mathcal{H}_m^*\boldsymbol{x}}.$$

Since the numerator in (3.2) is a positive real number whereas the denominator has negative real part, λ has negative real part as well; consequently, $\mathcal{H}_m + \mathcal{M}^{\text{PMR}}$ is stable. To conclude, the solution of (2.3) for PMR is positive semidefinite, as it is the solution to a Lyapunov equation with a stable coefficient matrix and a positive semidefinite constant term given by $\boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^*$. ☐

We have already mentioned that, in the MR framework, the matrix Y_m is not necessarily semidefinite even in the case of A with field of values in $\mathbb{C}^-$, as the former is the solution to the least-squares problem (1.3). The semidefiniteness of Y_m can be included in the formulation of the least-squares problem as an additional constraint; see, e.g., [30]. However, this constraint increases the already expensive computational cost of the solution of (1.3). Proposition 3.1 shows that the semidefiniteness of Y_m^{PMR} is guaranteed in our novel setting.

In the following, we explore connections between PMR and actual minimal residual methods.

3.1. Connections to the minimal residual method. Recall that in the minimal residual framework, Y_m solves the matrix least-squares problem (1.3). By considering the Kronecker representations of the solutions to (1.3) and (2.3), we can more directly relate the two approaches.

Define

$$\underline{\mathbf{H}}_m := \underline{\mathcal{H}}_m \otimes \underline{\mathcal{I}} + \underline{\mathcal{I}} \otimes \underline{\mathcal{H}}_m$$

and

$$\underline{\boldsymbol{g}}_m = \text{vec}(\underline{\boldsymbol{E}}_1\Gamma\Gamma^*\underline{\boldsymbol{E}}_1^*).$$

The non-singularity of the Kronecker matrix $\mathbf{H}_m$ follows from that of $\mathcal{H}_m$. Note that we work with these quantities only theoretically, as $\underline{\mathbf{H}}_m$ has dimensions $(m+1)^2 r^2 \times m^2 r^2$ and is dense, making it prohibitively expensive, in practice, to form for even moderate m and r. The vector $\underline{\boldsymbol{g}}_m$ has length $(m+1)^2 r^2$. We can therefore define $\boldsymbol{y}_m^{\text{MR}} := -\underline{\mathbf{H}}^\dagger\underline{\boldsymbol{g}}_m$, which minimizes $\|\underline{\mathbf{H}}_m\boldsymbol{y} + \underline{\boldsymbol{g}}_m\|_{\text{F}}$. In fact, due to the equivalence between (1.3) and the Kronecker problem $\|\underline{\mathbf{H}}_m\boldsymbol{y} + \underline{\boldsymbol{g}}_m\|_{\text{F}}$, it holds that $\boldsymbol{y}_m^{\text{MR}} = \text{vec}(Y_m^{\text{MR}})$; see, e.g., [18, Chapter 4].

The PMR solution can be expressed in vectorized form as $\boldsymbol{y}_m^{\text{PMR}} := -(\mathbf{H}_m + \mathbf{M}_m^{\text{PMR}})^{-1}\boldsymbol{g}_m$, where

$$(3.3) \quad \mathbf{M}_m^{\text{PMR}} := \mathcal{H}_m^{-*}\mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{H}_m^{-*}\mathcal{J}_m \quad \text{and} \quad \mathcal{J}_m := \boldsymbol{E}_m H_{m+1,m}^* H_{m+1,m}\boldsymbol{E}_m^*.$$

Note as well that, for $\mathbf{M}^{\mathsf{G}} := 0$, we recover the vectorized Galerkin solution $\boldsymbol{y}_m^{\mathsf{G}}$. Furthermore, the following result holds true.

PROPOSITION 3.2. *Define*

$$\mathbf{M}_m^{\mathsf{MR}} := \mathbf{H}_m^{-*}(\mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m). \tag{3.4}$$

Then

$$\boldsymbol{y}_m^{\mathsf{MR}} = -(\mathbf{H}_m + \mathbf{M}_m^{\mathsf{MR}})^{-1}\boldsymbol{g}_m.$$

Proof. The Moore–Penrose inverse (and $\underline{\mathbf{H}}_m$ having full column rank) gives

$$\boldsymbol{y}_m^{\mathsf{MR}} = -(\underline{\mathbf{H}}_m^*\underline{\mathbf{H}}_m)^{-1}\underline{\mathbf{H}}_m^*\underline{\boldsymbol{g}}_m. \tag{3.5}$$

Expanding $\underline{\mathbf{H}}_m^*\underline{\mathbf{H}}_m$ results in

$$
\begin{aligned}
\underline{\mathbf{H}}_m^*\underline{\mathbf{H}}_m &= (\underline{\mathcal{H}}_m^* \otimes \underline{\mathcal{I}}^* + \underline{\mathcal{I}}^* \otimes \underline{\mathcal{H}}_m^*)(\underline{\mathcal{H}}_m \otimes \underline{\mathcal{I}} + \underline{\mathcal{I}} \otimes \underline{\mathcal{H}}_m) \\
&= \underline{\mathcal{H}}_m^*\underline{\mathcal{H}}_m \otimes \underline{\mathcal{I}}^*\underline{\mathcal{I}} + \underline{\mathcal{I}}^*\underline{\mathcal{I}} \otimes \underline{\mathcal{H}}_m^*\underline{\mathcal{H}}_m + \underline{\mathcal{H}}_m^*\underline{\mathcal{I}} \otimes \underline{\mathcal{I}}^*\underline{\mathcal{H}}_m + \underline{\mathcal{I}}^*\underline{\mathcal{H}}_m \otimes \underline{\mathcal{H}}_m^*\underline{\mathcal{I}} \\
&= \mathcal{H}_m^*\mathcal{H}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{H}_m^*\mathcal{H}_m + \mathcal{H}_m \otimes \mathcal{H}_m^* + \mathcal{H}_m^* \otimes \mathcal{H}_m + \mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m \\
&= \mathbf{H}_m^*\mathbf{H}_m + \mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m. \tag{3.6}
\end{aligned}
$$

By [18, Lemma 4.3.1], it follows that

$$
\begin{aligned}
\underline{\mathbf{H}}_m^*\underline{\boldsymbol{g}}_m &= ((\underline{\mathcal{H}}_m^* \otimes \underline{\mathcal{I}}^*) + (\underline{\mathcal{I}}^* \otimes \underline{\mathcal{H}}_m^*))\operatorname{vec}(\underline{\boldsymbol{E}}_1\Gamma\Gamma^*\underline{\boldsymbol{E}}_1^*) \\
&= \operatorname{vec}(\underline{\mathcal{I}}^*\underline{\boldsymbol{E}}_1\Gamma\Gamma^*\underline{\boldsymbol{E}}_1^*\bar{\underline{\mathcal{H}}}_m) + \operatorname{vec}(\underline{\mathcal{H}}_m^*\underline{\boldsymbol{E}}_1\Gamma\Gamma^*\underline{\boldsymbol{E}}_1^*\underline{\mathcal{I}}) \\
&= \operatorname{vec}(\boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^*\bar{\mathcal{H}}_m) + \operatorname{vec}(\mathcal{H}_m^*\boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^*) \\
&= ((\mathcal{H}_m^* \otimes \mathcal{I}) + (\mathcal{I} \otimes \mathcal{H}_m^*))\operatorname{vec}(\boldsymbol{E}_1\Gamma\Gamma^*\boldsymbol{E}_1^*) \\
&= \mathbf{H}_m^*\boldsymbol{g}_m, \tag{3.7}
\end{aligned}
$$

where $\bar{\mathcal{H}} = (\mathcal{H}^*)^T$ denotes the element-wise complex conjugate of $\mathcal{H}$. Combining (3.6) and (3.7) with (3.5) leads to the desired result:

$$
\begin{aligned}
\boldsymbol{y}_m^{\mathsf{MR}} &= -(\mathbf{H}_m^*\mathbf{H}_m + \mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m)^{-1}\mathbf{H}_m^*\boldsymbol{g}_m \\
&= -(\mathbf{H}_m + \mathbf{H}_m^{-*}(\mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m))^{-1}\boldsymbol{g}_m \\
&= -(\mathbf{H}_m + \mathbf{M}_m^{\mathsf{MR}})^{-1}\boldsymbol{g}_m. \quad \square
\end{aligned}
$$

The modification matrices $\mathbf{M}_m^{\mathsf{PMR}}$ and $\mathbf{M}_m^{\mathsf{MR}}$ are indeed closely related.

LEMMA 3.3. *With the notation above, we have*

$$\mathbf{M}_m^{\mathsf{PMR}} = \mathbf{M}_m^{\mathsf{MR}} + \mathbf{H}_m^{-*}(\mathcal{H}_m^{-*}\mathcal{J}_m \otimes \mathcal{H}_m^* + \mathcal{H}_m^* \otimes \mathcal{H}_m^{-*}\mathcal{J}_m).$$

Proof. Multiplying $\mathbf{M}_m^{\mathsf{PMR}}$ and $\mathbf{H}_m^*$ gives

$$\mathbf{H}_m^*\mathbf{M}_m^{\mathsf{PMR}} = \mathcal{J}_m \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{J}_m + \mathcal{H}_m^{-*}\mathcal{J}_m \otimes \mathcal{H}_m^* + \mathcal{H}_m^* \otimes \mathcal{H}_m^{-*}\mathcal{J}_m.$$

Inverting $\mathbf{H}_m^*$ immediately gives the desired result. $\square$

Lemma 3.3 implies that whenever the term $\mathbf{H}_m^{-*}(\mathcal{H}_m^{-*}\mathcal{J}_m \otimes \mathcal{H}_m^* + \mathcal{H}_m^* \otimes \mathcal{H}_m^{-*}\mathcal{J}_m)$ is small, the PMR solution is close to the MR one. This is the case, for instance, when $\|\mathcal{J}_m\|_{\mathrm{F}} \ll 1$. However, another interesting scenario where $\mathbf{M}_m^{\mathsf{PMR}} \approx \mathbf{M}_m^{\mathsf{MR}}$ is described in the following theorem.

THEOREM 3.4. *Let* $\mathcal{H}_m = \mathcal{Q}_m \Lambda_m \mathcal{Q}_m^{-1}$, $\Lambda_m = \text{diag}(\lambda_1, \ldots, \lambda_{mr})$ *and* $\mathcal{Q}_m \in \mathbb{C}^{mr \times mr}$, *be the eigendecomposition of* $\mathcal{H}_m$. *Then*

$$\|\mathbf{M}_m^{\text{MR}} - \mathbf{M}_m^{\text{PMR}}\|_F^2 \leq 2m^3 r^4 q_m \bar{c_m} \max_{i,j=1,\ldots,mr} \frac{(|\lambda_j|^2 + |\lambda_i|^2)^2}{|\lambda_j|^2 |\lambda_i|^2 |\lambda_i + \lambda_j|^2},$$

where $\bar{c}_m > 0$ *and* $q_m = \|\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*}\|_F^2$.

Proof. Again applying [18, Lemma 4.3.1], the pth column of $\mathbf{M}^{\text{PMR}}$ (cf. (3.3)) can be rewritten as

$$\begin{aligned}
\mathbf{M}_m^{\text{PMR}} e_p &= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*})(\Lambda_m^{-*} \mathcal{Q}_m^* \mathcal{J}_m \otimes \mathcal{Q}_m^* + \mathcal{Q}_m^* \otimes \Lambda_m^{-*} \mathcal{Q}_m^* \mathcal{J}_m) e_p \\
&= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*})(\Lambda_m^{-*} \mathcal{Q}_m^* \mathcal{J}_m \otimes \mathcal{Q}_m^* + \mathcal{Q}_m^* \otimes \Lambda_m^{-*} \mathcal{Q}_m^* \mathcal{J}_m) \text{vec}(e_\ell e_k^T) \\
&= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*}) \text{vec}(\mathcal{Q}_m^* e_\ell e_k^T \mathcal{J}_m^T \bar{\mathcal{Q}}_m \Lambda_m^{-*} + \Lambda_m^{-*} \mathcal{Q}_m^* \mathcal{J}_m e_\ell e_k^T \bar{\mathcal{Q}}),
\end{aligned}$$

where $p = (\ell - 1)mr + k$, for some $\ell, k \in \{1, \ldots, mr\}$,[3] and $\bar{\mathcal{Q}}_m$ denotes the element-wise complex conjugate of $\mathcal{Q}_m$.

Similarly for the p-th column of $\mathbf{M}_m^{\text{MR}}$ (cf. (3.4)):

$$\begin{aligned}
&\mathbf{M}_m^{\text{MR}} e_p \\
&= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*})(\Lambda_m \otimes \mathcal{I} + \mathcal{I} \otimes \Lambda_m)^{-*}(\mathcal{Q}_m^* \mathcal{J}_m \otimes \mathcal{Q}_m^* + \mathcal{Q}_m^* \otimes \mathcal{Q}_m^* \mathcal{J}_m) e_p \\
&= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*})(\Lambda_m \otimes \mathcal{I} + \mathcal{I} \otimes \Lambda_m)^{-*}(\mathcal{Q}_m^* \mathcal{J}_m \otimes \mathcal{Q}_m^* + \mathcal{Q}_m^* \otimes \mathcal{Q}_m^* \mathcal{J}_m) \text{vec}(e_\ell e_k^T) \\
&= (\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*}) \text{vec}(\mathcal{L}_m \circ (\mathcal{Q}_m^* e_\ell e_k^T \mathcal{J}_m^T \bar{\mathcal{Q}}_m + \mathcal{Q}_m^* \mathcal{J}_m e_\ell e_k^T \bar{\mathcal{Q}}_m)),
\end{aligned}$$

where $\mathcal{L}_m$ is a Cauchy matrix whose (i,j)-th element is given by $\mathcal{L}_{i,j} = (\bar{\lambda}_i + \bar{\lambda}_j)^{-1}$, and $\circ$ denotes the Hadamard product. Note that $\mathcal{L}_m$ is well defined, due to $\mathcal{H}_m$ being non-singular.

Thanks to the expression above, we first notice that the indexes of some of the zero columns of $\mathbf{M}_m^{\text{MR}}$ and $\mathbf{M}_m^{\text{PMR}}$ coincide. Indeed, by recalling that $\mathcal{J}_m = \mathbf{E}_m H_{m+1,m}^* H_{m+1,m} \mathbf{E}_m^*$, $\mathcal{J}_m e_\ell e_k^T = 0$ if both ℓ and k belong to $\{1, \ldots, (m-1)r\}$.

As a consequence, $\mathbf{M}_m^{\text{PMR}} e_p$ and $\mathbf{M}_m^{\text{MR}} e_p$ may be different from the zero vector if and only if $p = (\ell - 1)mr + k$ is such that either ℓ or k is in $\{(m-1)r + 1, \ldots, mr\}$. We thus have to consider only these mr^2 columns for achieving an upper bound on the column-wise differences between $\mathbf{M}_m^{\text{PMR}}$ and $\mathbf{M}_m^{\text{MR}}$. Let $\mathcal{D}_m := \mathcal{Q}_m^* \mathcal{J}_m e_\ell e_k^T \bar{\mathcal{Q}}_m$ for either ℓ or $k \in \{(m-1)r + 1, \ldots, mr\}$. Then

$$\|\mathbf{M}_m^{\text{MR}} e_p - \mathbf{M}_m^{\text{PMR}} e_p\|_{\text{F}}^2 \leq \|\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*}\|_{\text{F}}^2 \|\mathcal{L}_m \circ (\mathcal{D}_m^T + \mathcal{D}_m) - (\mathcal{D}_m^T \Lambda^{-*} + \Lambda^{-*} \mathcal{D}_m)\|_{\text{F}}^2,$$

otherwise, we are sure that $\mathbf{M}_m^{\text{MR}} e_p = \mathbf{M}_m^{\text{PMR}} e_p = 0$.

A direct inspection of the entries of $\mathcal{L}_m \circ (\mathcal{D}_m^T + \mathcal{D}_m)$ and $\mathcal{D}_m^T \Lambda^{-*} + \Lambda^{-*} \mathcal{D}_m$ shows that

$$e_i^T (\mathcal{L}_m \circ (\mathcal{D}_m^T + \mathcal{D}_m)) e_j = \frac{(\mathcal{D}_m)_{j,i} + (\mathcal{D}_m)_{i,j}}{\bar{\lambda}_i + \bar{\lambda}_j},$$

and

$$e_i^T (\mathcal{D}_m^T \Lambda^{-*} + \Lambda^{-*} \mathcal{D}_m) e_j = \frac{(\mathcal{D}_m)_{j,i}}{\bar{\lambda}_i} + \frac{(\mathcal{D}_m)_{i,j}}{\bar{\lambda}_j}.$$

---

[3]Notice the abuse of notation here: $e_p \in \mathbb{R}^{m^2 r^2}$, whereas $e_\ell, e_k \in \mathbb{R}^{mr}$.

We thus have

$$
\left\| \mathcal{L}_m \circ (\mathcal{D}_m^T + \mathcal{D}_m) - (\mathcal{D}_m^T \Lambda^{-*} + \Lambda^{-*} \mathcal{D}_m) \right\|_{\mathrm{F}}^2
$$

$$
= \sum_{i=1}^{mr} \sum_{j=1}^{mr} \left| \frac{(\mathcal{D}_m)_{j,i} + (\mathcal{D}_m)_{i,j}}{\bar{\lambda}_i + \bar{\lambda}_j} - \frac{(\mathcal{D}_m)_{j,i}}{\bar{\lambda}_i} - \frac{(\mathcal{D}_m)_{i,j}}{\bar{\lambda}_j} \right|^2
$$

$$
= \sum_{i=1}^{mr} \sum_{j=1}^{mr} \left| \frac{\bar{\lambda}_j \bar{\lambda}_i ((\mathcal{D}_m)_{j,i} + (\mathcal{D}_m)_{i,j}) - (\bar{\lambda}_i + \bar{\lambda}_j)(\bar{\lambda}_j (\mathcal{D}_m)_{j,i} + \bar{\lambda}_i (\mathcal{D}_m)_{i,j})}{\bar{\lambda}_j \bar{\lambda}_i \cdot (\bar{\lambda}_i + \bar{\lambda}_j)} \right|^2
$$

$$
\leq m^2 r^2 c_m(\ell, k) \max_{i,j=1,\ldots,mr} \frac{\left| 2\bar{\lambda}_j \bar{\lambda}_i - (\bar{\lambda}_i + \bar{\lambda}_j)^2 \right|^2}{\left| \bar{\lambda}_j \bar{\lambda}_i \cdot (\bar{\lambda}_i + \bar{\lambda}_j) \right|^2}
$$

$$
\leq m^2 r^2 c_m(\ell, k) \max_{i,j=1,\ldots,mr} \frac{\left| \bar{\lambda}_i^2 + \bar{\lambda}_j^2 \right|^2}{\left| \bar{\lambda}_j \bar{\lambda}_i \cdot (\bar{\lambda}_i + \bar{\lambda}_j) \right|^2},
$$

where $c_m(\ell, k) = \|\mathcal{D}_m\|_{\max}^2$. Notice that we can drop the conjugation in the expression above thanks to the presence of the absolute value. Moreover, by recalling that $\|\mathbf{M}_m^{\mathrm{MR}} e_p - \mathbf{M}_m^{\mathrm{PMR}} e_p\|_{\mathrm{F}} \neq 0$ for (at most) mr^2 columns, we have

$$
\|\mathbf{M}^{\mathrm{MR}} - \mathbf{M}^{\mathrm{PMR}}\|_{\mathrm{F}}^2 = \sum_{p=1}^{m^2 r^2} \|\mathbf{M}^{\mathrm{MR}} e_p - \mathbf{M}^{\mathrm{PMR}} e_p\|_{\mathrm{F}}^2
$$

$$
\leq 2m^3 r^4 q_m \bar{c}_m \max_{i,j=1,\ldots,mr} \frac{\left| \lambda_i^2 + \lambda_j^2 \right|^2}{\left| \lambda_j \lambda_i \cdot (\lambda_i + \lambda_j) \right|^2},
$$

where $\bar{c}_m = \max_{\ell,k} c_m(\ell, k)$ and $q_m = \|\mathcal{Q}_m^{-*} \otimes \mathcal{Q}_m^{-*}\|_{\mathrm{F}}$. □

Theorem 3.4 shows that the distance between $\mathbf{M}_m^{\mathrm{PMR}}$ and $\mathbf{M}_m^{\mathrm{MR}}$ can be related to the magnitude of the function

$$
f(x, y) := \frac{\left| x^2 + y^2 \right|^2}{\left| xy \cdot (x + y) \right|^2}, \quad x, y \in \mathrm{spec}(\mathcal{H}_m).
$$

Since $\mathrm{spec}(\mathcal{H}_m) \subseteq W(A)$, where $W(A) := \{z^* A z \in \mathbb{C}, \|z\| = 1\}$ is the field of values of A, one can compute an approximation of $W(A)$ and study $\max_{x,y} f(x, y)$ in $W(A)$. If this value is small, then we can expect the PMR and MR methods to achieve similar results.

COROLLARY 3.5. *Let $A = A^*$. With the notation of Theorem 3.4, it holds that*

$$
\|\mathbf{M}_m^{\mathrm{MR}} - \mathbf{M}_m^{\mathrm{PMR}}\|_F^2 \leq 2m^3 r^4 \bar{c}_m \max_{i,j=1,\ldots,mr} \frac{(\lambda_j^2 + \lambda_i^2)^2}{\lambda_j^2 \lambda_i^2 (\lambda_i + \lambda_j)^2}.
$$

Proof. The proof follows from Theorem 3.4 by noticing that having an Hermitian A implies $\mathcal{H}_m$ to be Hermitian so that $\mathcal{Q}_m$ is unitary and $\mathrm{spec}(\mathcal{H}_m)$ is real. □

3.2. A residual-minimizing Kronecker sum approach. We introduce $\oplus$ to denote the *Kronecker sum* between two matrices $B, C \in \mathbb{C}^{p \times p}$:

$$
B \oplus C := B \otimes I_p + I_p \otimes C.
$$

Define $\mathbf{J}_m := \mathcal{J}_m \oplus \mathcal{J}_m$. It follows from Proposition 3.2 and the definitions in (3.3) that

$$
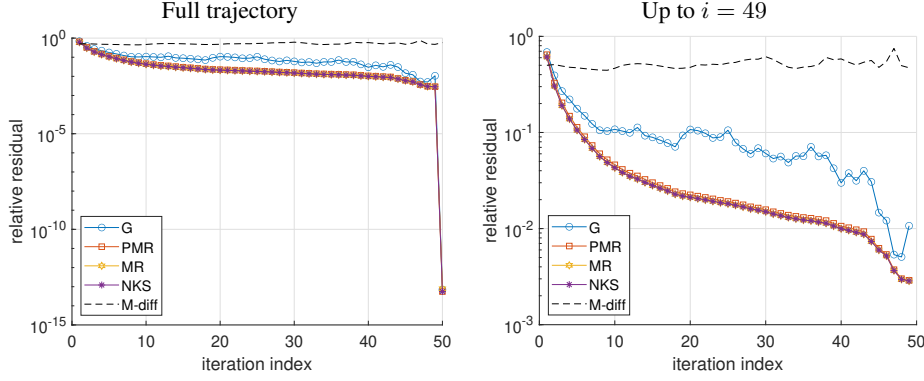\boldsymbol{y}_m^{\mathrm{MR}} = -(\mathbf{H}_m + \mathbf{H}_m^{-*} \mathbf{J}_m)^{-1} \boldsymbol{g}_m
$$

FIG. 3.1. *Relative residuals with respect to iteration index for various methods and A as a 1D Laplacian. "M-diff" is $\|\boldsymbol{M}^{\text{PMR}} - \boldsymbol{M}^{\text{NKS}}\|_2/\|\boldsymbol{M}^{\text{PMR}}\|_2$.*

and

$$\boldsymbol{y}_m^{\text{PMR}} = -((\mathcal{H}_m + \mathcal{H}_m^{-*}\mathcal{J}_m) \oplus (\mathcal{H}_m + \mathcal{H}_m^{-*}\mathcal{J}_m))^{-1}\boldsymbol{g}_m.$$

It is tempting to conjecture that $(\mathcal{H}_m + \mathcal{H}_m^{-*}\mathcal{J}_m) \oplus (\mathcal{H}_m + \mathcal{H}_m^{-*}\mathcal{J}_m)$ is the closest Kronecker sum to $\mathsf{H}_m + \mathsf{H}_m^{-*}\mathsf{J}_m$, under the Frobenius norm. In particular, it is possible to find $\boldsymbol{M}_{\text{NKS}} \in \mathbb{C}^{mr \times r}$, where NKS stands for "nearest Kronecker sum," such that

$$\boldsymbol{y}_m^{\text{NKS}} = -((\mathcal{H}_m + \boldsymbol{M}^{\text{NKS}}\boldsymbol{E}_m^*) \oplus (\mathcal{H}_m + \boldsymbol{M}^{\text{NKS}}\boldsymbol{E}_m^*))^{-1}\boldsymbol{g}_m$$

and $\boldsymbol{M}_{\text{NKS}}$ minimizes the Frobenius norm of (2.4), thereby potentially achieving a much lower residual for the problem (1.1) than the PMR approach. More explicitly,

$$(3.8) \qquad \boldsymbol{M}_{\text{NKS}} := \arg\min \left\| G_m(\boldsymbol{M}) \begin{bmatrix} & & I_r \\ I_r & & -I_r \\ & -I_r & \end{bmatrix} G_m(\boldsymbol{M})^* \right\|_{\text{F}},$$

where

$$G_m(\boldsymbol{M}) :=$$
$$\begin{bmatrix} \boldsymbol{E}_{m+1}H_{m+1,m} - \underline{\mathcal{I}}\operatorname{vec}(((\mathcal{H}_m + \boldsymbol{M}\boldsymbol{E}_m^*) \oplus (\mathcal{H}_m + \boldsymbol{M}\boldsymbol{E}_m^*))^{-1}\boldsymbol{g}_m)\boldsymbol{E}_m & \underline{\mathcal{I}}\boldsymbol{M} \end{bmatrix}$$

and

$$\underline{\mathcal{I}} := \begin{bmatrix} I_{mr} \\ 0 \end{bmatrix}.$$

Unfortunately, this conjecture is not true, as is demonstrated by the following script, `test_kron_sum_conj.m`, in our toolbox, `LowRank4Lyap`, described in further detail in Section 5. In this script, A is a one-dimensional discretization of the Laplace operator, and the optimization problem (3.8) is solved via a general unconstrained optimization routine, `fminunc`, with $\boldsymbol{M}^{\text{PMR}}$ as an initial guess. See Figure 3.1 for a plot of the relative residuals alongside the relative difference $\|\boldsymbol{M}^{\text{PMR}} - \boldsymbol{M}^{\text{NKS}}\|_2/\|\boldsymbol{M}^{\text{PMR}}\|_2$ and Table 3.1 for the final iterations of the plot. Although NKS, PMR, and MR appear to overlap, there are small visible differences when one zooms in.

Although the NKS approach fits the LRM framework, as it is currently implemented, it is not scalable in practice. Furthermore, for Hermitian A, NKS and PMR appear to be very close, meaning that PMR may be sufficient for such scenarios. We also examine NKS for a non-symmetric problem; see Example 5.6 in Section 5.

TABLE 3.1
Relative residuals for final iterations in Figure 3.1.

	Galerkin	PMR	MR	NKS
46	1.2120e-02	5.3767e-03	5.1983e-03	5.1996e-03
47	5.3607e-03	3.7351e-03	3.6560e-03	3.6565e-03
48	5.0699e-03	3.0104e-03	2.9629e-03	2.9632e-03
49	1.0686e-02	2.8980e-03	2.8543e-03	2.8548e-03
50	7.0699e-14	5.5865e-14	7.8170e-14	5.5956e-14

4. Compress-and-restart strategy. Following earlier work with Kressner and Massei [24], we note that a compress-and-restart strategy is also viable for the LRM framework in Section 2. The key is to express the residual in a factored form with a "core" matrix. For the following, we drop the mod superscript to allow space for restart cycle indices, expressed as (k), $k = 1, \ldots, k_{\max}$. The cycle length and block size are denoted as m_k and r_k, respectively, and in the case of redundancies, they may be dropped in favor of just the cycle index.

From Proposition 2.1, recall that

$$R^{(1)} := \boldsymbol{\mathcal{V}}_{m_1+1} G^{(1)} L^{(1)} (G^{(1)})^* \boldsymbol{\mathcal{V}}_{m_1+1}^* \in \mathbb{C}^{n \times n},$$

where, for all $k \in \{1, \ldots, k_{\max}\}$,

$$L^{(k)} := \begin{bmatrix} & I_{r_k} & \\ I_{r_k} & & -I_{r_k} \\ & -I_{r_k} & \end{bmatrix}.$$

Denote $\boldsymbol{C}^{(2)} := \boldsymbol{\mathcal{V}}_{m_1+1} G^{(1)} \in \mathbb{C}^{n \times r_2}$, where here $r_2 = 3r_1$. To restart, we compute the next Krylov subspace $\mathscr{K}_{m_2}(A, \boldsymbol{C}^{(2)})$ and obtain the basis $\boldsymbol{\mathcal{V}}_{m_2+1} \in \mathbb{C}^{n \times (m_2+1)r_2}$ and block Hessenberg $\underline{\mathcal{H}}_{m_2+1} \in \mathbb{C}^{(m_2+1)r_2 \times m_2 r_2}$. To compute an update to the solution from $k = 1$, we then solve the projected problem

$$(4.1) \quad (\mathcal{H}_{m_2} + \boldsymbol{M}^{(2)} \boldsymbol{E}_{m_2}^*) Y + Y (\mathcal{H}_{m_2} + \boldsymbol{M}^{(2)} \boldsymbol{E}_{m_2}^*)^* + \boldsymbol{E}_1 \Gamma^{(2)} L^{(2)} (\Gamma^{(2)})^* \boldsymbol{E}_1^* = 0,$$

where $\boldsymbol{M}^{(2)} \in \mathbb{C}^{m_2 r_2 \times r_2}$ is an LRM and here $\boldsymbol{E}_1 \in \mathbb{R}^{n \times m_2 r_2}$. Letting $Y^{(2)}$ denote the solution to (4.1), we can then compute $Z^{(2)} := \boldsymbol{\mathcal{V}}_{m_2} Y^{(1)} \boldsymbol{\mathcal{V}}_{m_2}^*$ and add it back to $X^{(1)}$. Doing this iteratively leads to a final solution of the form

$$X = X^{(1)} + \sum_{k=2}^{k_{\max}} Z^{(k)},$$

where each $Z^{(k)} = \boldsymbol{\mathcal{V}}_{m_k} Y^{(k)} \boldsymbol{\mathcal{V}}_{m_k}^*$, $Y^{(k)}$ is the solution of the projected problem

$$(\mathcal{H}_{m_k} + \boldsymbol{M}^{(k)} \boldsymbol{E}_{m_k}^*) Y + Y (\mathcal{H}_{m_k} + \boldsymbol{M}^{(k)} \boldsymbol{E}_{m_k}^*)^* + \boldsymbol{E}_1 \Gamma^{(k)} L^{(k)} (\Gamma^{(k)})^* \boldsymbol{E}_1^* = 0,$$

$$\boldsymbol{\mathcal{V}}_{m_k}^* \boldsymbol{C}^{(k)} = \boldsymbol{E}_1 \Gamma^{(k)},$$

$$\boldsymbol{C}^{(k)} := \boldsymbol{\mathcal{V}}_{m_{(k-1)}+1} G^{(k-1)} \in \mathbb{C}^{n \times r_{(k-1)}},$$

and

$$G^{(k)} := \begin{bmatrix} \boldsymbol{E}_{m_k+1} H_{m_k+1, m_k} & \underline{\mathcal{I}} Y^{(k)} \boldsymbol{E}_{m_k} & \underline{\mathcal{I}} \boldsymbol{M}^{(k)} \end{bmatrix} \in \mathbb{C}^{n \times 3r_k}.$$

Although the residual at each iteration can be computed as in Proposition 2.1,

$$R^{(k)} := \boldsymbol{\mathcal{V}}_{m_k+1} G^{(k)} L^{(k)} (G^{(k)})^* \boldsymbol{\mathcal{V}}_{m_k+1}^* \in \mathbb{C}^{n \times n},$$

one clear downside to this approach is that the rank of $R^{(k)}$ triples each restart cycle. As discussed in more detail in [24], we therefore need a compression strategy of the starting vector $C^{(k)}$ from one cycle to the next. We do this in a symmetric fashion via [24, Algorithm 3]. We also employ a $\mathrm{mem}_{\max}$ parameter that dictates how many column vectors can be stored per cycle and toggle the block size r_k and maximum basis size m_k accordingly per cycle. See [24, Algorithm 4] for more details.

5. Numerical results. All numerical tests were written and run in MATLAB 2022a and can be found in the repository `LowRankMod4Lyap`,[4] hosted on GitLab. Every test was run on a single, standard node of the compute cluster Mechthild,[5] housed at the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany. A standard node consists of two Intel Xeon Silver 4110 (Skylake) CPUs, each with eight cores, 64 kB L1 cache, and 1024 kB L2 cache at a clock rate of 2.1 GHz, as well as 12 MB of shared L3 cache. We set `maxNumCompThread` in MATLAB to 4.

We consider a wide variety of numerical tests in this section. In Section 5.1 we compare the convergence results among the G, MR, and PMR methods for both Hermitian and non-Hermitian matrices A and plot the bound from Theorem 3.4 for problems with Hermitian A. We study the performance of these methods for varying ranks r in Section 5.2. We see how the compress-and-restart strategy works for PMR compared to G in Section 5.3.

Test matrices have either been generated by our own code or taken from the SuiteSparse Matrix Collection [9] (and, in particular, originally from the Oberwolfach Benchmark Collection [21]) or the SLICOT Benchmark Collection [8].[6] We provide descriptions for each matrix below.

- `bad_cond_diag`: A is a diagonal matrix with logarithmically spaced values ranging from 1 to 10^{12}.
- `conv_diff_3d`: Central finite-differences stencil of three-dimensional convection–diffusion operator. The matrix A has size $n = N^3$, where N is the number of discretization points in each direction. The parameter ε controls the viscosity dominance, whereby smaller ε correlates with low viscosity and high convection.
- `laplacian_2d`: Central finite-differences stencil of the two-dimensional Laplacian operator. The matrix A has size $n = N^2$, where N is the number of discretization points in each direction.
- `log_diag`: Example 4 from [30]; A is non-normal but diagonalizable with logarithmically spaced eigenvalues from 1 to 100.
- `rail_1357`: A symmetric, heat transfer, steel profile cooling matrix of size 1357×1357 in the Oberwolfach Collection; C is determined by the problem.[7]
- `iss`: Component 1r of the International Space Station problem in the SLICOT Collection; non-symmetric; C is determined by the problem.

Unless otherwise noted, the constant term CC^* is built from uniformly distributed random numbers (i.e., `rand` in MATLAB).

---

[4]https://gitlab.com/katlund/LowRankMod4Lyap
[5]https://www.mpi-magdeburg.mpg.de/cluster/mechthild
[6]The collection can currently be found at
https://github.com/SLICOT/Benchmark-ModelReduction.
[7]Note that the data is provided as a generalized Lyapunov equation, which we transform into a standard one. For details, please see the code.

5.1. Convergence behavior. For all examples in this section, we restrict ourselves to the non-restarted version of each algorithm. In addition to the convergence results, we also plot the eigenvalues of the solution Y_m to the small problem (1.2), (1.3), or (2.3), as well as a Ritz-value function based on the bound from Theorem 3.4,

$$f(x,y) := \frac{\left|x^2 + y^2\right|^2}{\left|xy \cdot (x+y)\right|^2},$$

where the coefficients have been dropped. In each example, the function is evaluated on the spectrum of $\mathcal{H}_m$. All algorithms are halted after surpassing a relative residual of 10^{-6}.

EXAMPLE 5.1. For the first example, we consider a `log_diag` problem with $n = 1000$ and $r = 3$. Notice that, in this example, A has field of values in the right half of the complex plane $\mathbb{C}^+$, so the exact solution X to (1.1) is negative semidefinite.

The results are shown in Figure 5.1 and all methods perform similarly. However, MR suffers from a slightly positive solution eigenvalue. Neither the Galerkin nor PMR approaches produce positive eigenvalues, as expected from our theoretical analysis. The Ritz-valued function shows that the difference between PMR and MR should be relatively small, confirming what we see for the residual behavior.
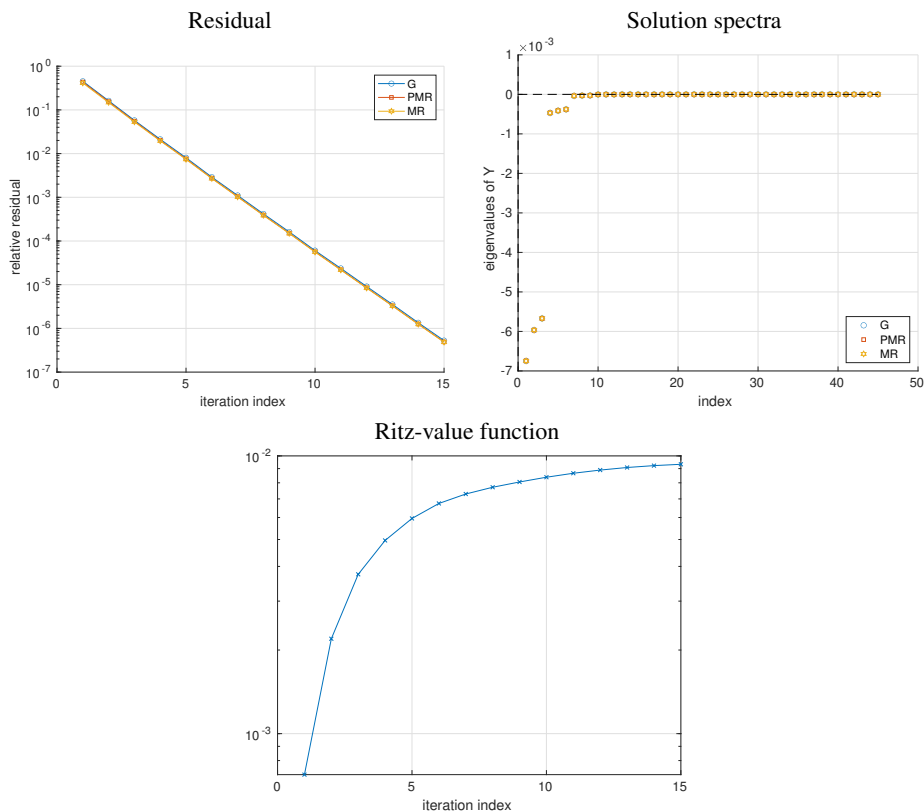
FIG. 5.1. *Convergence results for Example 5.1.*

EXAMPLE 5.2. We now consider a `bad_cond_diag` problem with $n = 500$ and $r = 3$, with results shown in Figure 5.2. Both MR and PMR achieve nearly the same relative

residual in the early iteration and improve over the Galerkin approach until later residuals, at which point all methods overlap. The cluster of eigenvalues of A near zero poses numerical challenges for all algorithms, leading to some positive eigenvalues in Y_m. However, the PMR approach minimizes the positive eigenvalues the best, with the largest having magnitude $\mathcal{O}(10^{-16})$, while that of the Galerkin approach is $\mathcal{O}(10^{-11})$ and that of MR is $\mathcal{O}(10^{-7})$. As for the Ritz-value function, it remains relatively small until near convergence, at which point $\mathcal{J}_m$ is close to zero, causing a spike.
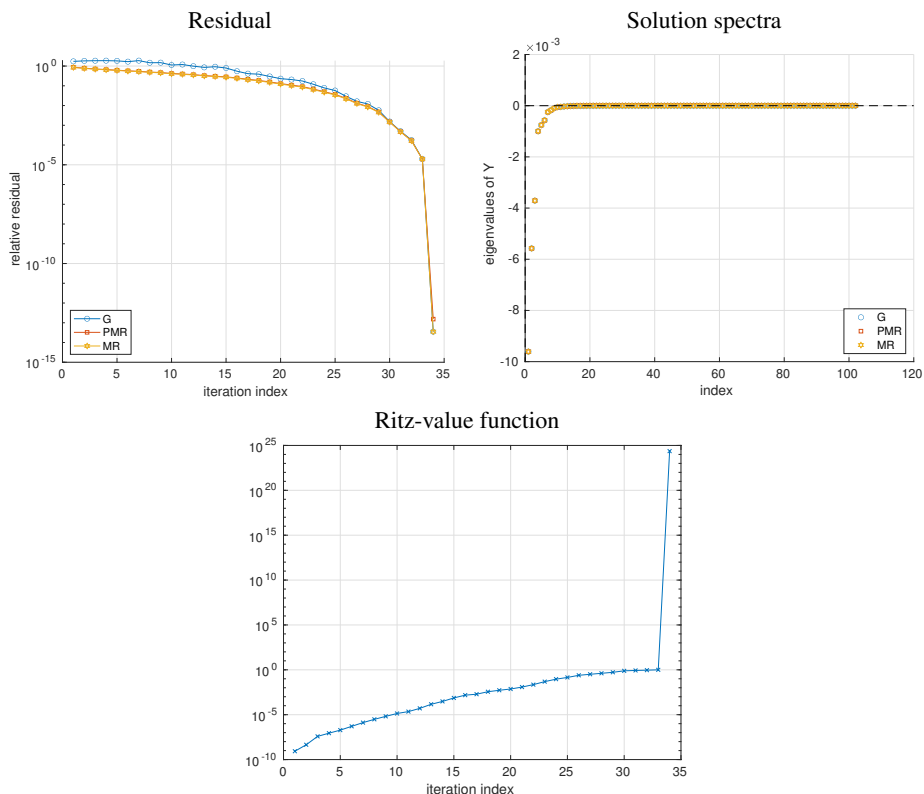
FIG. 5.2. *Convergence results for Example 5.2.*

EXAMPLE 5.3. We begin to see a clearer advantage of the PMR approach for the `laplacian_2d` problem with $n = 10^4$ and $r = 3$ in Figure 5.3. The PMR method overlaps with MR very closely and reaches 10^{-6} ten iterations before the Galerkin method. As for the solution spectra, there are fewer eigenvalues for MR and PMR due to the earlier convergence, but we can see that all methods closely approximate the same spectra overall. The Ritz-value function also remains relatively small for all iterations.

EXAMPLE 5.4. The `conv_diff_3d` problem, with $n = 25^3$ and $r = 3$ allows us to see the effect that non-normality has on the behavior of each method. For a problem with strong diffusion ($\varepsilon = 10^{-2}$, left panel in Figure 5.4), we see that PMR overlaps with MR relatively well and converges one iteration before the Galerkin method. As diffusion becomes weaker ($\varepsilon = 10^{-3}$, right panel in Figure 5.4), PMR drifts from MR and is unable to capture its more drastic reduction in iterations (nine fewer versus three fewer, for MR and PMR, respectively). This confirms our intuition that PMR better approximates MR for symmetric matrices A.
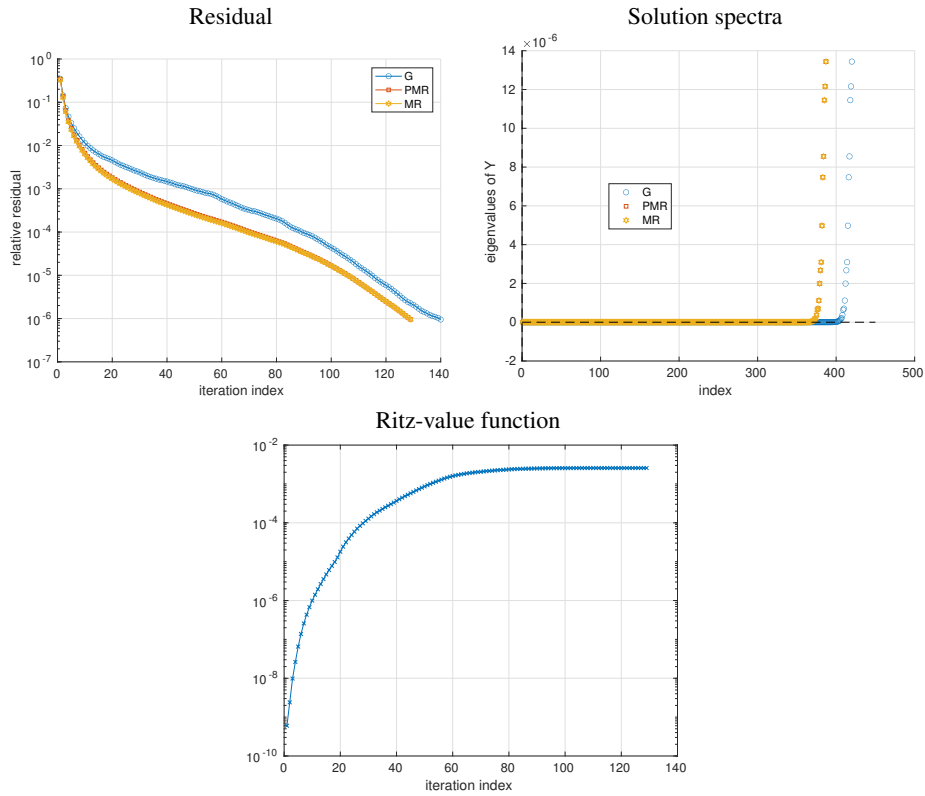
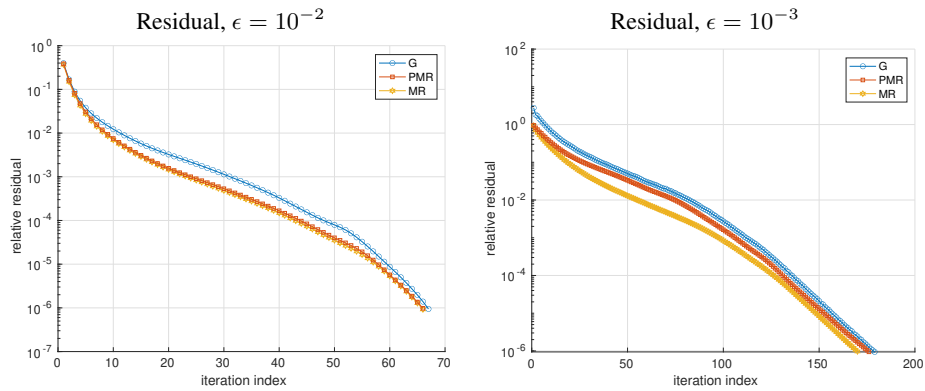Fig. 5.3. *Convergence results for Example 5.3.*

Fig. 5.4. *Convergence results for Example 5.4.*

EXAMPLE 5.5. With the `rail_1357` problem, we get a look at a more practical application. Although not reported in the plots in Figure 5.5, we first note that the Galerkin and PMR methods both took about 2 minutes, while MR needed more than 35 minutes, to achieve the same relative residual tolerance. We can also clearly see that the Galerkin method does not produce a monotonic residual, while PMR follows the minimal residual of MR very closely. In terms of iteration counts, there is practically no improvement over the Galerkin

method, but residual monotonicity for the same computational cost is incredibly useful in practice. The Ritz-value function is, however, counter-intuitive, given how well PMR visually overlaps with MR. This suggests that a more precise metric might be needed to better measure when PMR could be relied on to approximate MR. No method produces erroneous solution eigenvalues.
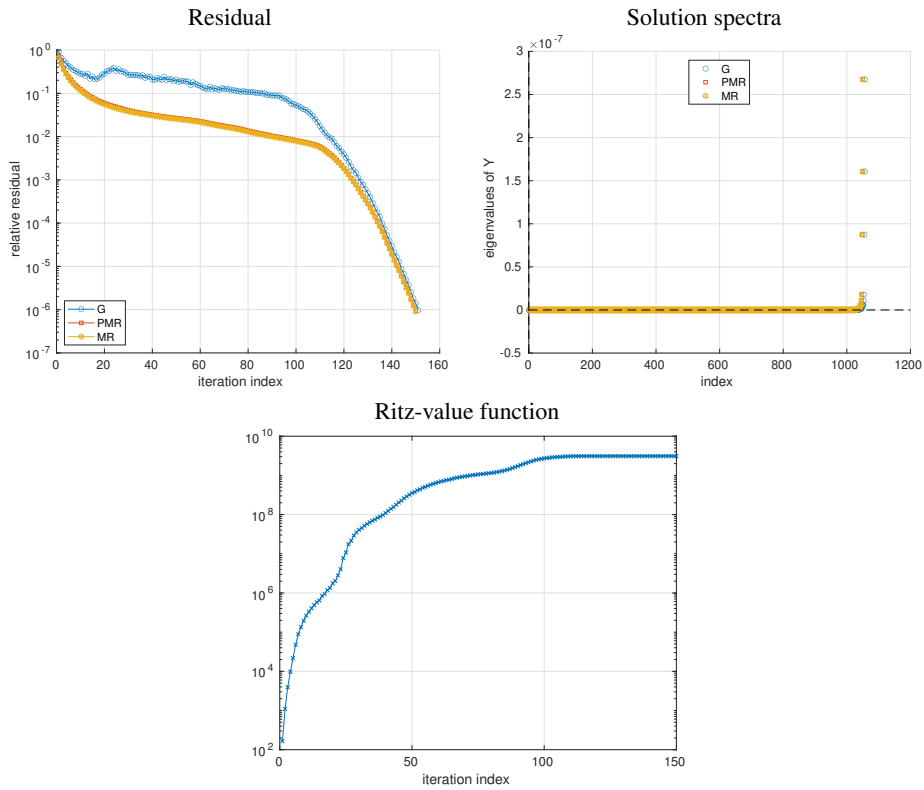
FIG. 5.5. *Convergence results for Example 5.5.*
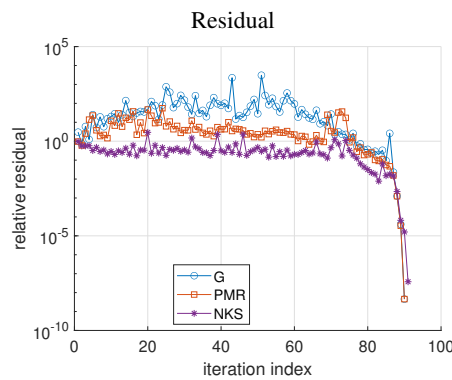
FIG. 5.6. *Convergence results for Example 5.6.*

EXAMPLE 5.6. For the final example in this section, in Figure 5.6 we demonstrate how badly PMR can behave when A comes from the `iss` problem and is far from symmetric. Although PMR is in some ways less bad than Galerkin, both are far from monotonic, especially with respect to the MR approach. We also compare both methods with NKS as described in Section 3.2. As implemented, NKS is far from practicable, because an optimization problem has to be solved each iteration. However, it is clear that it achieves a much lower residual with fewer large jumps.

5.2. Performance relative to block size. We now examine how effective each approach is relative to varying rank sizes. We consider just the `laplacian_2d` and `conv_diff_3d` problems. All results are presented as heatmaps scaled by the results from the Galerkin approach, to facilitate comparisons. We consider $r \in \{1, 2, 4, 8, 16\}$. The blue-green heatmaps display timings ratios, while the orange-pink ones display a ratio of iteration counts. All tests are run on a standard node of Mechthild as described at the beginning of Section 5.

EXAMPLE 5.7. Results for the `laplacian_2d` problem can be found in Figure 5.7. For small r, both MR and PMR appear to have a strong advantage over the Galerkin approach in terms of both timings and iteration counts. In fact, MR and PMR always have fewer iteration counts. However, MR clearly begins to suffer for $r = 4$, and its overall timings become increasingly worse as r increases. PMR in contrast remains competitive, always requiring fewer iterations than Galerkin and being slightly faster in terms of timings, although the gains decrease as r increases.
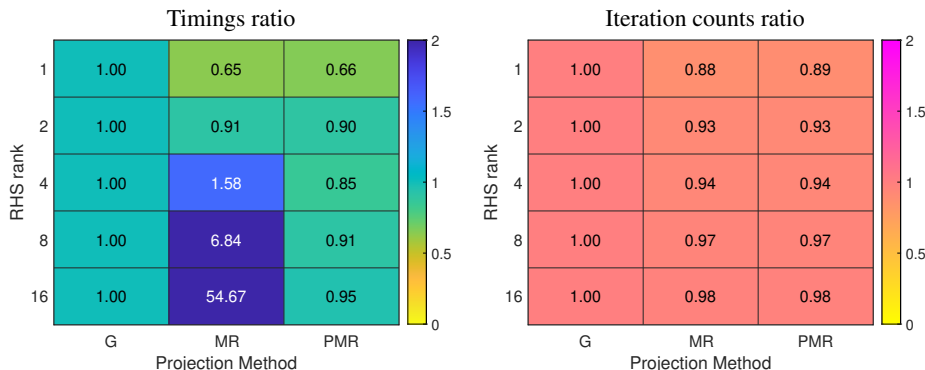

FIG. 5.7. *Performance results for Example 5.7.*

EXAMPLE 5.8. For this `conv_diff_3d` problem we have fixed $\varepsilon = 10^{-2}$. The results in Figure 5.8 are similar to those for Example 5.7, with both MR and PMR outperforming the Galerkin approach for $r = 1$. However, as r increases, MR becomes slower. Meanwhile, PMR continues to demonstrate a slight advantage over Galerkin, with the advantage decreasing as r increases.

5.3. Restarting. For the final set of examples, we study the behavior of the compress-and-restart strategy and compare between the Galerkin and PMR approaches. We no longer consider the MR approach, as it is ill-suited for a compress-and-restart strategy, due to it lacking the appropriate type of low-rank modification formulation.

Our primary goal in this section is to study restarts and their effect on the monotonicity of the residual. We therefore set the compression tolerance to machine epsilon (i.e., approximately 10^{-16} for IEEE `int64`) to effectively turn off compression.
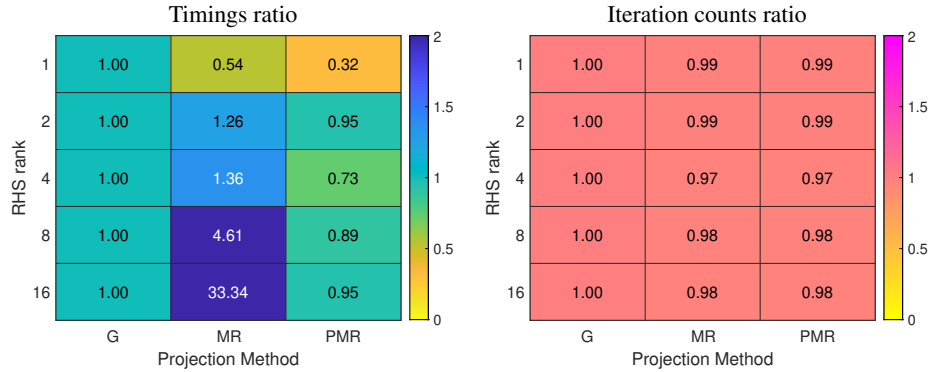
FIG. 5.8. *Performance results for Example 5.8.*

EXAMPLE 5.9. We again consider the `bad_cond_diag` problem with $n = 100$ and $r = 3$ and residual plot in Figure 5.9. The maximum number of columns to be stored is set to 100. Although PMR only improves over the Galerkin approach by a few iterations, its monotonic residual makes it more reliable in practice.
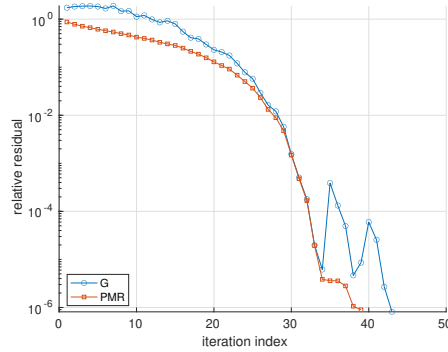

FIG. 5.9. *Convergence results for Example 5.9.*

EXAMPLE 5.10. We also look at the `laplacian_2d` once again, with $n = 10^4$ and $r = 3$ and maximum number of columns set to 96. The results are shown in Figure 5.10. Although the convergence of PMR is almost perfectly monotonic, it is notably slower than that of the Galerkin method. It may therefore be reasonable to combine and switch methods after a certain point – perhaps after the residual hits 10^{-4} – but determining a reliable heuristic in practice remains an open challenge.

EXAMPLE 5.11. Finally, we return to the `rail_1357` problem, whose results for different maximum memory limits are shown in Figure 5.11. For the lower memory limit in the left panel, we see a situation similar to that of Example 5.10, whereby the PMR residual is smooth and monotonic but ultimately converges more slowly than that of the Galerkin approach. In the right panel, we see how the behavior changes with a higher memory tolerance: the PMR method is able to improve slightly and remain monotonic.

6. Conclusions and outlook. In this paper we have presented a new general framework for solving large-scale Lyapunov matrix equations. Low-rank-modified Galerkin methods, which include the Galerkin method as a special case with the zero modification, introduces
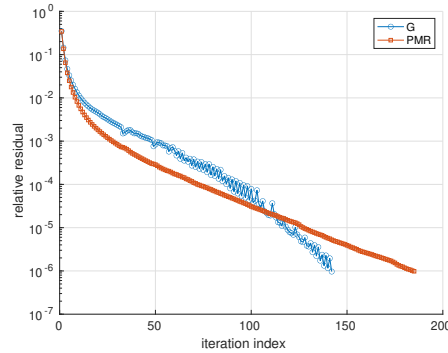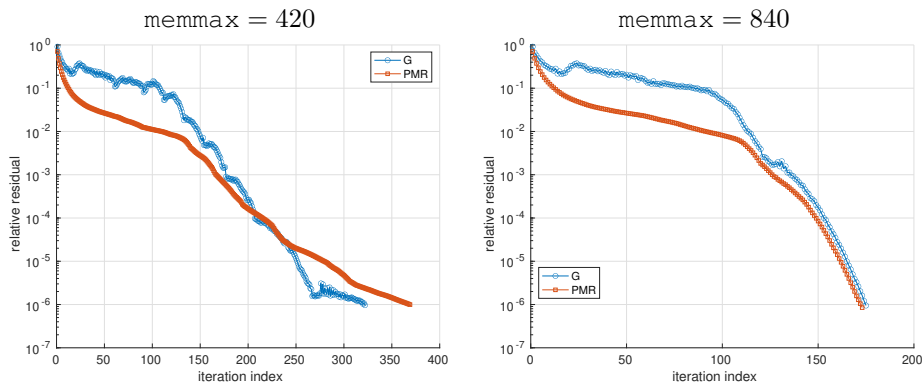
FIG. 5.10. *Convergence results for Example 5.10.*

FIG. 5.11. *Convergence results for Example 5.11.*

a suitable low-rank correction in the projected equation aimed at achieving a certain target behavior. Driven by the goal of designing a computationally affordable scheme able to show a convergence rate similar to a minimal residual approach, we have proposed two non-trivial options for such a low-rank correction.

The first one, PMR, is defined by taking inspiration from the relation between FOM and GMRES in the linear system setting. We showed that, under certain hypotheses on A, adopting this correction leads to a projection method that is well defined. Moreover, we have depicted possible scenarios where we expect the performance of our new solver to be close to that achieved by MR. Our numerical results have confirmed such findings. On the other hand, further analysis is needed to fully understand the relation between the two approaches and other situations where PMR may be applicable.

The second low-rank correction we have proposed, NKS, is computed by minimizing the current residual norm at each iteration. In spite of its appealing theoretical features, computing this low-rank correction directly is unreasonably expensive, making its use limited in practice. Designing ad-hoc, more efficient, optimization procedures is an avenue worth pursuing to make this approach affordable in terms of computational cost. To this end, sketch-and-solve methods like, e.g., the Blendenpik algorithm [2] could be a valid option.

Our new framework is sufficiently flexible to handle low-rank corrections different from the ones we have proposed, and it would also be useful in designing low-rank corrections targeting other goals as, e.g., a certain spectral distribution of $\mathcal{H}_m + \mathcal{M}$. We have also showed

that, thanks to the low-rank format of the residual matrix computed by our novel low-rank-modified Galerkin method, the latter can be successfully integrated in a compress-and-restart scheme for matrix equations.

For the sake of simplicity, we have restricted our analysis to the use of polynomial Krylov subspaces. However, generalizing our approach to the case of more sophisticated approximation spaces like extended and rational Krylov subspaces is just a technical exercise.

As a possible outlook, we envision our new low-rank-modified Galerkin framework could be applied to the solution of other matrix equations like, e.g., generalized Lyapunov equations or algebraic Riccati equations.

REFERENCES

[1] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.

[2] H. AVRON, P. MAYMOUNKOV, AND S. TOLEDO, *Blendenpik: supercharging Lapack's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236.

[3] J. BAKER, M. EMBREE, AND J. SABINO, *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 656–668.

[4] U. BAUR, *Low rank solution of data-sparse Sylvester equations*, Numer. Linear Algebra Appl., 15 (2008), pp. 837–851.

[5] U. BAUR AND P. BENNER, *Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic*, Computing, 78 (2006), pp. 211–234.

[6] P. BENNER, P. KÜRSCHNER, AND J. SAAK, *Frequency-limited balanced truncation with low-rank approximations*, SIAM J. Sci. Comput., 38 (2016), pp. A471–A499.

[7] P. BENNER AND J. SAAK, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM-Mitt., 36 (2013), pp. 32–52.

[8] Y. CHAHLAOUI AND P. VAN DOOREN, *Benchmark examples for model reduction of linear time-invariant dynamical systems*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., Lect. Notes Comput. Sci. Eng., 45, Springer, Berlin, 2005, pp. 379–392.

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

[10] V. DRUSKIN AND V. SIMONCINI, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems Control Lett., 60 (2011), pp. 546–560.

[11] A. FROMMER, S. GÜTTEL, AND M. SCHWEITZER, *Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1602–1624.

[12] A. FROMMER, K. LUND, AND D. B. SZYLD, *Block Krylov subspace methods for functions of matrices*, Electron. Trans. Numer. Anal., 47 (2017), pp. 100–126.
    https://etna.ricam.oeaw.ac.at/vol.47.2017/pp100-126.dir/pp100-126.pdf

[13] ———, *Block Krylov subspace methods for functions of matrices II: Modified block FOM*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 804–837.

[14] W. GAWRONSKI AND J.-N. JUANG, *Model reduction in limited time and frequency intervals*, Internat. J. Systems Sci., 21 (1990), pp. 349–376.

[15] M. H. GUTKNECHT, *Block Krylov space methods for linear systems with multiple right-hand sides: an introduction*, in Modern Mathematical Models, Methods and Algorithms for Real World Systems, A. H. Siddiqi, I. S. Duff, and O. Christensen, eds., Anamaya, New Delhi, 2007, pp. 420–447.

[16] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436 (1953).

[17] N. J. HIGHAM, *Functions of Matrices*, SIAM, Philadelphia, 2008.

[18] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[19] D. Y. HU AND L. REICHEL, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.

[20] K. JBILOU AND M. RAYDAN, *Nonlinear least-squares approach for large-scale algebraic Riccati equations*, SIAM J. Sci. Comput., 41 (2019), pp. A2193–A2211.

[21] J. G. KORVINK AND E. B. RUDNYI, *Oberwolfach Benchmark Collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., Lect. Notes Comput. Sci. Eng., 45, Springer, Berlin, 2005, pp. 311–315.

[22] D. KRESSNER, *Bivariate matrix functions*, Oper. Matrices, 8 (2014), pp. 449–466.

[23] ———, *A Krylov subspace method for the approximation of bivariate matrix functions*, in Structured Matrices in Numerical Linear Algebra, D. A. Bini, F. Di Benedetto, E. Tyrtyshnikov, and M. Van Barel, eds., Springer INdAM Ser., 30, Springer, Cham, 2019, pp. 197–214.

[24] D. KRESSNER, K. LUND, S. MASSEI, AND D. PALITTA, *Compress-and-restart block Krylov subspace methods for Sylvester matrix equations*, Numer. Linear Algebra Appl., 28 (2021), Art. e2339, 17 pages.

[25] P. KÜRSCHNER, *Efficient Low-Rank Solution of Large-Scale Matrix Equations*, PhD. Thesis, Faculty of Mathematics, Otto-von-Guericke-University, Magdeburg, 2016.

[26] J.-R. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[27] Y. LIN AND V. SIMONCINI, *Minimal residual methods for large scale Lyapunov equations*, Appl. Numer. Math., 72 (2013), pp. 52–71.

[28] D. PALITTA, M. SCHWEITZER, AND V. SIMONCINI, *Sketched and truncated polynomial Krylov subspace methods: matrix equations*, Preprint on arXiv, 2023. https://arxiv.org/abs/2311.16019

[29] D. PALITTA AND V. SIMONCINI, *Matrix-equation-based strategies for convection–diffusion equations*, BIT, 56 (2016), pp. 751–776.

[30] ———, *Optimality properties of Galerkin and Petrov–Galerkin methods for linear matrix equations*, Vietnam J. Math., 48 (2020), pp. 791–807.

[31] T. PENZL, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.

[32] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[33] ———, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441.

[34] V. SIMONCINI AND E. GALLOPOULOS, *Convergence properties of block GMRES and matrix polynomials*, Linear Algebra Appl., 247 (1996), pp. 97–119.

[35] J. SNYDERS AND M. ZAKAI, *On nonnegative solutions of the equation $AD + DA = -C$*, SIAM J. Appl. Math., 18 (1970), pp. 704–714.

[36] B. VANDEREYCKEN AND S. VANDEWALLE, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2553–2579.

[37] K. ZHOU, J. C. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, 1996.