A NEW BLOCK KRYLOV SUBSPACE FRAMEWORK WITH APPLICATIONS TO FUNCTIONS OF MATRICES ACTING ON MULTIPLE VECTORS

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN (DR. RER. NAT.)

an der Fakultät Mathematik und Naturwissenschaften der
Bergischen Universität Wuppertal vorgelegte und genehmigte

DISSERTATION
von
Kathryn Lund

Betreut durch Prof. Dr. Andreas Frommer und Prof. Dr. Daniel B. Szyld

Dissertation eingereicht am: 23. Februar 2018
Tag der Disputation: 26. März 2018

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20180613-120530-8
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20180613-120530-8]

ABSTRACT

We propose a new framework for understanding block Krylov subspace methods, which hinges on a matrix-valued inner product. We can recast the "classical" block Krylov methods, such as O'Leary's block conjugate gradients, global methods, and loop-interchange methods, within this framework. Leveraging the generality of the framework, we develop an efficient restart procedure and error bounds for the shifted block full orthogonalization method (Sh-BFOM(m)). Regarding BFOM as the prototypical block Krylov subspace method, we propose another formalism, which we call modified BFOM, and show that block GMRES and the new block Radau-Lanczos method can be regarded as modified BFOM. In analogy to Sh-BFOM(m), we develop an efficient restart procedure for shifted BGMRES with restarts (Sh-BGMRES(m)), as well as error bounds.

Using this framework and shifted block Krylov methods with restarts as a foundation, we formulate block Krylov subspace methods with restarts for matrix functions acting on multiple vectors $f(A)\boldsymbol{B}$. We obtain convergence bounds for B(FOM)^2 (BFOM for Functions Of Matrices) and block harmonic methods (i.e., BGMRES-like methods) for matrix functions.

With various numerical examples, we illustrate our theoretical results on Sh-BFOM and Sh-BGMRES. We also analyze the matrix polynomials associated to the residuals of these methods. Through a variety of real-life applications, we demonstrate the robustness and versatility of B(FOM)^2 and block harmonic methods for matrix functions. A particularly interesting example is the tensor t-function, our proposed definition for the function of a tensor in the tensor t-product formalism. Despite the lack of convergence theory, we also show that the block Radau-Lanczos modification can reduce the number of cycles required to converge for both linear systems and matrix functions.

ZUSAMMENFASSUNG

In dieser Arbeit stellen wir einen allgemeinen Rahmen für die Formulierung und theoretische Betrachtung von Block-Krylow-Unterraumverfahren vor, welcher sich die Definition eines matrixwertigen Innen-Produktes zunutze macht. Innerhalb dieses Rahmens formulieren wir die "klassischen" Block-Krylow-Unterraum-Verfahren, wie O'Learys Block-Conjugate-Gradient-Verfahren, globale Krylow-Unterraumverfahren und Schleifentausch-Verfahren. Die Allgemeinheit unseres Ansatzes ermöglicht es uns, das Shifted-Block-Full-Orthogonalization-Verfahren (Sh-BFOM(m)) um effiziente Neustarts zu erweitern und Fehlerschranken anzugeben. Darüber hinaus geben wir eine Modifikation des BFOM-Verfahrens an, welche wir Modified-BFOM nennen. Das BFOM-Verfahren lässt sich als Prototyp vieler Block-Krylow-Unterraumverfahren ansehen. In gleicher Weise zeigen wir, dass sich das Block-GMRES- sowie das neue Block-Radau-Lanczos-Verfahren als Modified-BFOM auffassen lassen. Analog zur Konstruktion von Sh-BFOM(m) entwickeln wir eine effiziente Neustart-Prozedur und Fehlerschranken für das Shifted-BGMRES-Verfahren (Sh-BGMRES(m)).

Unter Zurhilfenahme unseres allgemeinen Ansatzes und auf Basis von Neustarts verwendenden Shifted-Block-Krylow-Verfahren entwickeln wir Block-Krylow-Unterraum-Verfahren mit Neustarts für die Berechnung von $f(A)\boldsymbol{B}$, d.h. für die Berechnung des Produktes einer Matrix mit mehreren rechten Seiten, wobei die Matrix durch die Anwendung einer Matrixfunktion gegeben ist. Des Weiteren leiten wir Schranken für die Konvergenzgeschwindigkeit der B(FOM)^2 (BFOM for Functions

of Matrices) und Block-Harmonic-Verfahren (d.h. Verfahren ähnlich zu BGMRES) für Matrixfunktionen her.

Anhand von diversen numerischen Beispielen bestätigen wir unsere theoretischen Vorhersagen über Sh-BFOM und Sh-BGMRES. Außerdem analysieren wir die matrixwertigen Polynome, welche mit den Residuenverläufen der Verfahren in Beziehung stehen. Indem wir das B(FOM)^2- und die Block-Harmonic-Verfahren in diversen praxisrelevanten Anwendungen testen, zeigen wir, dass die Verfahren robust und vielseitig einsetzbar sind. Ein besonders interessantes Beispiel, welches wir betrachten, ist die Tensor-t-Funktion. Sie ist der von uns vorgeschlagene Weg, Funktionen auf Tensoren, welche mit der durch das t-Produkt gegebenen Struktur versehen sind, zu definieren. Da wir keine theoretischen Belege haben, zeigen wir mittels Experimenten, dass die Block-Radau-Lanczos-Modifikation die Anzahl der Durchläufe sowohl für die iterative Lösung linearer Systeme als auch für die Berechnung von Matrixfunktionen reduziert.

ACKNOWLEDGEMENTS

This thesis was written under the terms of the Agreement for a Doctoral Thesis Co-tutorship between Temple University – Of the Commonwealth System of Higher Education and Bergische Universität Wuppertal, Germany, School of Mathematics and Natural Sciences.

I am painfully aware of my short-comings, as well as my privilege.[1] This work comprises not only the past few years of research, but many personal mountains and valleys along the way. I cannot possibly thank everyone who has helped me, but as a compulsive list-maker, I have to try.

If anyone challenges the mathematician stereotype, it is Daniel Szyld, an intelligent man with heart and style. I have not always understood his words of encouragement (I recall something about *The Magic Flute*), but they uplifted me anyway. Tú no has sido solamente un consejero, sino también como un padre.

It is not often that I go up to a speaker after a talk, but I took that risk with Andreas Frommer and then fell in love with matrix functions. I was lucky that Daniel and Andreas are good friends, and lucky again that Andreas was willing to host me for at first 3 months, and then 6 more months, in Wuppertal. His commitment to my work inspired me to learn new things (and break old things). Und ich bin so dankbar, dass du alle meine viele Emails immer geantwortet hast.

[1] Lots of people tell me to quit being so self-deprecating... really, I'm trying. It's just, sometimes I'm trying to be funny. Guess I'm not even good at that...

Benjamin Seibold was the first to encourage me to pursue a career in applied mathematics. With him, I got my first taste of real research via traffic modeling and learned to expect more of myself than I thought possible.

I am also thankful to Gillian Queisser, who encouraged me to attend an interesting multigrid conference in Bruchsal, and Matthias Bolten, who took an early interest in my work and supported me in the job application process.

Maria Lorenz has been a cheerleader and coach throughout my time in grad school. She often checked in on me and met for tea to talk about life.

My first math teacher was my mother, and she made sure I never fell behind, despite attending three different high schools. I first learned about matrices and Gaussian elimination from her Algebra II textbook. I am also very thankful to DeAnn Scherer– my high-school calculus teacher– and David Zitarelli– my college calculus professor– who both saw something in me early on.

Melena, mi amor, mi tesoro! Por tí, casi perdí los primeros años en el programa. Pero sin tí, estaría perdida. Te quiero más de lo que tú me quieres a mí.

My Temple peeps: you should be thanking **me** for making grad school such a disruptively vibrant place.

Meine Wupperleute: Hannah for Kürbissuppe and for translating my abstract into German (Isa auch!); Artur for not stealing the 50 euros I gave him to mail me my residence permit; Sarah for rehearsal Oktoberfest; and many others for helping me around Wuppertal when I first arrived on Krücken. I am especially thankful to Teodor for introducing me to Mendeley and asynchronous methods and for helping me slay the beast.

Grace, Rayan, Max, Matty, Damaris, Kristen, and Alisa: you all have been with me for a while. Thanks for sticking around and calling out my BS.

To all the benevolent postdocs and older "siblings" and "cousins": thank you for your sage wisdom.

Many thanks to various YouTube channels: Ambient, Arctic Empire, SuicideSheep, Fluidified, Pandora Journey, and some relaxation channel that keeps changing its name because of copyright infringement. My soul thanks the artists whose albums often provided sweet resonance: Crywolf, Natalia Lafourcade, The Glitch Mob, Tom Day, Chevelle, Relient K, Switchfoot, Hozier, Joy Williams, Thousand Foot Krutch, and Lucius.

To my felines: Fluffy, you were my best friend since I was four. Chloe, you were a spark of life. Shayera, you were the most beautiful demon-possessed cat I have ever met. Dove, get off my keyboard. Auri, stop eating Dove's food.

To my mother, who was discouraged from pursuing computer science and higher level mathematics due the sexism of her time: whenever I thought of giving up, I would think about you, and renew my resolve to be a gentle and persistent thorn in the side of a culture that often lacks compassion. Thank you for trying to understand what matrix functions are.

To my father, who has always supported me despite not understanding what I do: I appreciate the jokes about real analysis vs. fake analysis, and partial vs. full differential equations. Also, yes, there are definitely polynomials in the equations of the graphs of the logarithm of the chart of the quadratic of the C-file of your Mustang. But you're right, I can't hover.

I finally thank Bao, a man full of love and patience, for proving me wrong.

My parents and Bao were often much needed sources of funding. Other sources of funding include teaching assistantships from the Department of Mathematics of Temple University, the Deutsche Forschungsgemeinschaft through Collaborative Research Centre SFB TRR55 "Hadron Physics from Lattice QCD;" the National Science Foundation via grant DMS-1418882; and the U.S. Department of Energy under grant DE-SC 0016578. I would also like to thank Temple, SIAM, ILAS, and AMS for travel funding throughout the years to attend various conferences.

Efolóh thah efoláı.

Belóh ail efol.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

FIGURE 1. A bas-relief of the sirrush, a chimera-like creature composed of different parts of a dragon, a lion, an eagle, a snake, and other animals. The sirrush is prominently featured throughout the reconstructed Ishtar Gate at the Pergamon Museum in Berlin. The photograph is the author's own work.

CHAPTER 1

INTRODUCTION

Functions of matrices are a bit like the sirrush (see Figure 1). In a naive sense, they should not exist: how is it possible to insert a matrix into a scalar function and obtain anything meaningful? Their definition is the fusion of disparate pieces of complex analysis and linear algebra, and they appear in applications as varied as differential equations, measures of connectedness in networks, and operators in theoretical particle physics. The methods for approximating matrix functions, whether one wants to compute the matrix function $f(A)$ itself or the matrix function acting on a vector $f(A)\boldsymbol{b}$, are also diverse and combine techniques from numerical linear algebra, complex analysis, and polynomial and rational function approximations.

This dissertation focuses on new methods for computing

$$f(A)\boldsymbol{B},$$

where f has a Cauchy-Stieltjes integral representation, $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$, and $\boldsymbol{B} \in \mathbb{C}^{n\times s}$. When possible, we consider general A which are linear operators from $\mathbb{C}^n$ to $\mathbb{C}^n$, and where the action of A on elements of $\mathbb{C}^{n\times s}$ is column-wise. In particular,

we consider scenarios with memory limitations, i.e., only a certain number of $n \times s$ matrices can be stored at a given time.

There are three kinds of approaches established in the literature for computing $f(A)\boldsymbol{B}$, when f is the exponential, A is represented by a large and sparse matrix, and $\boldsymbol{B}$ has multiple columns. The first is to find a polynomial p such that $f \approx p$ and compute $p(A)\boldsymbol{B}$ instead. This is effectively what Al-Mohy and Higham do in [5], where the polynomial is chosen as a truncated Taylor series and combined with scaling and an efficient way of computing powers of A times $\boldsymbol{B}$. Another approach is to find a rational approximation r to f. Then a series of linear systems $(A+\tau_i I)^{-1}\boldsymbol{B}$ must be approximated. Birk proposes a deflated shifted conjugate gradients algorithm for this approach in his thesis [15], using Padé approximations; and Wu, Pang, and Sun propose an "alternatively" shifted block full orthogonalization method with restarts in [141], using Carathéodory-Fejér approximations. The third approach is to generate a block Krylov subspace and project and restrict $f(A)\boldsymbol{B}$ onto this subspace, meaning that only $f(\mathcal{H})$ for a much smaller matrix $\mathcal{H}$ must be computed. Lopez and Simoncini take this approach in [96], where they further modify the algorithm so that special properties of A are forced upon $\mathcal{H}$.

We aim to fill in many left by these methods by developing methods that

- are defined for general functions f that do not require a priori approximations;
- are suitable for general matrices A, in particular large and sparse matrices that may be impossible to store or access directly; and
- treat $\boldsymbol{B}$ as a matrix, instead of as separate columns, so that matrix-matrix products and Level 3 BLAS may be used (see the introduction to Chapter 3).

We also consider scenarios with memory limitations, and we want to obtain convergence results for a broad class of functions and matrices.

The methods we propose are in the vein of the approach of Frommer, Güttel, and Schweitzer [53, 52] and Frommer, Lund, Schweitzer, and Szyld [54], which combine Krylov subspaces and quadrature to approximate $f(A)\boldsymbol{b}$ for a single vector $\boldsymbol{b}$. While such an approach is in many ways similar to that of approximating f by a polynomial or rational function, there are some key differences. For one, we do not need an a priori polynomial or rational approximation to f. In fact, we do not need to know anything about f, except that it is defined on the field of values of A. (See Lemma 3.13; the spectrum of $\mathcal{H}$ is contained in the field of values of A.) In our particular analysis, we do require that f have a Cauchy-Stieltjes form, but this is not such a stringent restriction for many functions of interest. Secondly, we do not compute $p(A)$ or $r(A)$– the computation is reduced to polynomials of $\mathcal{H}$ (or in the case of rational Krylov methods, rational functions of $\mathcal{H}$), which is a much smaller matrix for which computations are cheap and we can use direct methods.

Foundational for our matrix function methods is a new, comprehensive framework for block Krylov methods. This framework encompasses decades' worth of literature and, in effect, describes all the reasonable variations of block Krylov methods. We use this framework to generalize many well-known results, discover as-yet unexplored variations on block Krylov methods, and derive error bounds for our matrix function methods. We devote particular attention to block full orthogonalization methods (BFOM) with restarts for shifted linear systems, the results of which are an end goal by themselves as well as a components in the analysis of the matrix

function methods. We also develop a formalism for modifying BFOM by low-rank matrices, examples of which include block GMRES and a new method called "block Radau-Lanczos," and analyze these methods with matrix polynomials.

In addition to a number of traditional matrix function applications, we study the behavior of our methods on a newly proposed definition for functions of tensors. This definition is based on the t-product formalism of [88] and possesses analogous versions of the key properties of matrix functions. What's more, computing the so-called "tensor t-function" reduces to computing a function of a block circulant matrix times a block vector.

1.1 Musings on scientific computing

To make sense of the breadth of research encountered in the preparation of this thesis, particularly in numerical linear algebra and high-performance computing, we have developed a modest theory for tracking the possible sources of error a problem encounters as it journeys from "real life" down into the underbelly of a computer processor, and back up to reality.

(1) A **real-life** problem is discovered, formulated, and perhaps simplified. A simplification introduces error, since it is not the actual problem a human wants to solve.[1]

(2) **Data** is collected and stored. The collection process may introduce noise, and large data sets may only be stored by some smaller representation. Both aspects introduce error.

[1] Here we pay homage to Plato's *Allegory of the Cave*.

(3) A **model** is proposed. This model introduces error by making assumptions and oversimplifying certain complex aspects of reality.[2]

(4) A **numerical algorithm** is proposed in exact arithmetic to run simulations of or compute solutions to the model. For problems requiring the solution of linear systems of equations,[3] such algorithms are often iterative, generating a sequence of approximations that converge theoretically to a numerical solution. As the iterative process must be limited to a finite number of steps, an approximation error is introduced.

(5) The algorithm is **implemented** on a machine in floating point arithmetic– yet another source of error. The implementation accounts for the choice of programming language, memory management, parallelization, load balancing, and much more. All of these components have spawned fields in and of themselves and lead to additional sources of error, if not merely sources of annoyance for humans, who consistently introduce bugs.

(6) A solution is obtained and then **translated** back into reality. Error can arise in the translation between mathematical language to human language.[4]

These stages are not disjoint, and researchers do not necessarily move through them linearly. There are often feedback loops between different stages, e.g., realizing that the first round of data is insufficient and going back to collect more, running experiments to find the model is not good enough, drawing conclusions from running an

[2]https://xkcd.com/669/

[3]i.e., all of them.

[4]See season 3, episode 11 of *Last Week Tonight with John Oliver*. Everything causes and cures cancer.

algorithm and then finding a bug, and so forth. Ideally, an exhaustive scientific work journeys through all stages, but as problems these days have become more complicated, individual scientists have become more specialized and often work in only one or two stages at a time.

The bulk of this thesis lives in stage 4, but one of the secondary goals of the framework proposed in Chapters 3 and 4 is to bridge the gap between stages 4 and 5, to encourage that algorithmic development keep an eye towards implementation. Block Krylov methods in particular pose great potential for using hardware more efficiently. Our framework comprises a wide array of options– choice of block inner product and norm, additive low-rank modifications to adjust the eigenvalues of the projection and restriction of A onto the block Krylov subspace, multiple interpretations of the approximation, etc.– all at once. The associated software (see Section 7.1) is written to be transparent and versatile, so that future users can easily switch settings to determine which is best for their applications. All of this is extended even more generally to matrix functions, and much of the code and theory can be recycled and recast for eigenvalue solvers and matrix equations.

1.2 Outline

Various tools from numerical analysis and linear algebra are collected in Chapter 2. A summary of Krylov subspace methods for a single right-hand side and a detailed development of the comprehensive block Krylov framework, including theory for shifted block full orthgonalization methods (Sh-BFOM), is contained in Chapter 3. Chapter 4 embellishes the themes of Chapter 3 by exploring a polynomial version of the block Arnoldi relation with low-rank additive modifications, and posing shifted block

GMRES (Sh-BGMRES) as a modified Sh-BFOM. A block version of the Radau-Lanczos method is also recast in this modified BFOM framework. In both Chapters 3 and 4, we devote attention to formulating efficient restart techniques, to mitigate well understood issues with storage limitations for block Krylov methods, and to proving error bounds for the restarted, shifted approximations. Throughout, we also make use of the interpolating matrix polynomials associated to the BFOM and BGMRES approximations, as well as their residuals. The block framework is applied to approximate $f(A)\boldsymbol{B}$ in Chapter 5, and error bounds on Sh-BFOM and Sh-BGMRES allow us to prove error bounds for the new block Krylov matrix function methods.

The versatility and robustness of our matrix function methods are demonstrated in the numerical experiments of Chapter 7, along with small-scale analyses of shifted block Krylov methods and visualizations of matrix polynomials. In addition to a variety of academic examples useful for understanding the numerical properties of the methods, real-life applications are also studied, whose background is expounded in Chapter 6. Therein we also propose a new definition for a function acting on a tensor, which appears to be the first of its kind in multilinear algebra.

A summary and outlook is provided at the end of Chapters 3-7. In Chapter 8 we discuss our main contributions, recommendations, and some directions for future work.

CHAPTER 2

NECESSARY TOOLS

Here we aggregate disparate definitions, tools, and notions needed to understand the rest of the work. There is no particular order to these concepts, i.e., the ordering does not reflect a ranking of difficulty or importance.

This chapter contains some novel results, or in the least, results that we have proven anew, as we could not find them in the literature in the form that we require. Parts of Lemma 2.7 identify properties of Hermitian positive definite and positive real matrices that are maintained through products and congruence-like transformations. Theorem 2.15 demonstrates a relationship between the block eigendecomposition of an operator A and that of $f(A)$. Section 2.8 examines matrix derivatives. Other concepts, like the notions of interpolating matrix polynomials and block eigenvalues, are dusted off and cast in a new light. We use these concepts to prove a nontrivial result on the factorization of a special interpolating matrix polynomial (Theorem 2.24). The rest of this chapter contains results that are well established in numerical analysis and linear algebra, so the seasoned researcher may feel at ease skipping ahead and looking back here when necessary.

TABLE 2.1. Descriptions and examples of notation used throughout this work.

typeface description	usage	examples
uppercase blackboard bold	spaces	$\mathbb{S}$, $\mathbb{C}$
lowercase plain	scalars, constants, scalar-valued functions	α, f
uppercase plain	square matrices without block structure	H, Λ
lowercase bold	vectors	$\boldsymbol{b}$, $\boldsymbol{x}$
uppercase bold	block vectors	$\boldsymbol{Y}$, $\boldsymbol{\Gamma}$
uppercase calligraphy	block matrices, tensors	$\mathcal{H}$, $\mathcal{A}$
uppercase bold calligraphy	concatenation of block vectors	$\boldsymbol{\mathcal{V}}$

2.1 Notation

We utilize various scripts and boldface settings to distinguish between different objects. We try to remain strict with these choices, so that it is possible to determine from an object's typeface what its usage is. See Table 2.1. A particular exception to these rules is Krylov subspace $\mathscr{K}$, which uses an uppercase script typeface, different from the calligraphy typeface in Table 2.1. Per tradition, we also use Γ to denote a curve in the complex plane.

Standard linear algebra notation is used throughout: spec(A) denotes the spectrum of the operator A; trace (A) denotes the trace of A; and diag$(A_{11}, \ldots, A_{mm})$ creates a block diagonal matrix with $A_{11}, \ldots, A_{mm}$ as the block diagonal entries. We additionally define some special objects as "standard block unit vectors." Let $\otimes$ denote the Kronecker product between two matrices, i.e., with $A = (a_{ij})$ and $B = (b_{ij})$,

$$A \otimes B = \begin{bmatrix} a_{11}B & \ldots & a_{1,n}B \\ a_{21}B & \ldots & a_{2,n}B \\ \vdots & \ddots & \vdots \\ a_{n,1}B & \ldots & a_{n,n}B \end{bmatrix}.$$

Let I_d denote the $d \times d$ identity matrix. Then the kth *standard unit vector* $\widehat{\boldsymbol{e}}_k^d \in \mathbb{C}^d$ is the kth column of I_d, and the kth *standard block unit vector* is $\widehat{\boldsymbol{E}}_k^{d\times s} = \widehat{\boldsymbol{e}}_k^d \otimes I_s \in \mathbb{C}^{d\times s}$. We often drop the superscripts when the dimensions are clear from context. See equation (2.1) provides various ways of expressing $\widehat{\boldsymbol{E}}_1^{d\times s}$:

$$\widehat{\boldsymbol{E}}_1^{d\times s} = \begin{bmatrix} I_{s\times s} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes I_{s\times s}. \tag{2.1}$$

2.2 Matrix functions

Following [57, 74, 125], we concern ourselves with the three main matrix function definitions, based on the Jordan canonical form, Hermite interpolating polynomials, and the Cauchy-Stieltjes integral form. In each case, the validity of the definition boils down to the differentiability of f on the spectrum of A. When f is analytic on the spectrum of A, all the definitions are equivalent, and we can switch between them freely.

Let $A \in \mathbb{C}^{n\times n}$ be a matrix with spectrum $\text{spec}(A) := \{\lambda_j\}_{j=1}^N$, where $N \leq n$ and the λ_j are distinct. An $m \times m$ Jordan block $J_m(\lambda)$ of an eigenvalue λ has the form

$$J_m(\lambda) = \begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in \mathbb{C}^{m\times m}.$$

Suppose that A has Jordan canonical form

$$A = XJX^{-1} = X^{-1}\,\text{diag}(J_{m_1}(\lambda_{j_1}), \ldots, J_{m_p}(\lambda_{j_\ell}))X, \tag{2.2}$$

with p blocks of sizes m_i such that $\sum_{i=1}^p m_i = n$, and where the values $\{\lambda_{j_k}\}_{k=1}^\ell \in \text{spec}(A)$. Note that eigenvalues may be repeated in the sequence $\{\lambda_{j_k}\}_{k=1}^\ell$. Let n_j denote the *index* of λ_j, or the size of the largest Jordan block associated to λ_j.

A function is *defined on the spectrum of* A if all the following values exist:

$$f^{(k)}(\lambda_j), \qquad k = 0, \ldots, n_j - 1, \qquad j = 1, \ldots, N.$$

Definition 2.1: Suppose $A \in \mathbb{C}^{n\times n}$ has Jordan form (2.2) and that f is defined on the spectrum of A. Then we define

$$f(A) := Xf(J)X^{-1},$$

where $f(J) := \text{diag}(f(J_{m_1}(\lambda_{j_1})), \ldots, f(J_{m_p}(\lambda_{j_\ell})))$, and

$$f(J_{m_i}(\lambda_{j_k})) := \begin{bmatrix} f(\lambda_{j_k}) & f'(\lambda_{j_k}) & \frac{f''(\lambda_{j_k})}{2!} & \ldots & \frac{f^{(n_{j_k}-1)}(\lambda_{j_k})}{(n_{j_k}-1)!} \\ 0 & f(\lambda_{j_k}) & f'(\lambda_{j_k}) & \ldots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{f''(\lambda_{j_k})}{2!} \\ \vdots & & \ddots & \ddots & f'(\lambda_{j_k}) \\ 0 & \ldots & \ldots & 0 & f(\lambda_{j_k}) \end{bmatrix} \in \mathbb{C}^{m_i\times m_i}$$

Note that when A is diagonalizable with $\text{spec}(A) = \{\lambda_j\}_{j=1}^n$ (possibly no longer distinct), Definition 2.1 reduces to

$$f(A) = X \,\text{diag}(f(\lambda_1), \ldots, f(\lambda_n))X^{-1}.$$

Matrix powers are well defined, so a scalar polynomial evaluated on a matrix is naturally defined. Given $p(z) = \sum_{k=0}^m z^k c_k$, for some $\{c_k\}_{k=1}^m \subset \mathbb{C}$, we have that $p(A) := \sum_{k=1}^m A^k c_k$. Based on this, we can define non-polynomial functions of matrices by using again derivatives as we did in Definition 2.1.

Definition 2.2: Suppose that f is defined on $\text{spec}(A)$, and let p with $\deg p \leq \sum_{j=1}^N n_j$ be the unique Hermite interpolating polynomial satisfying

$$p^{(k)}(\lambda_j) = f^{(k)}(\lambda_j), \text{ for all } k = 0, \ldots, n_{j-1}, \quad j = 1, \ldots, N.$$

We then define $f(A) := p(A)$.

Crucial for our methods and analysis is the Cauchy-Stieltjes integral definition.

Definition 2.3: Let $\mathbb{D} \subset \mathbb{C}$ be a region, and suppose that $f : \mathbb{D} \to \mathbb{C}$ is analytic with integral representation

$$f(z) = \int_\Gamma \frac{g(t)}{t - z} \,\mathrm{d}t, \quad z \in \mathbb{D}, \tag{2.3}$$

with a path $\Gamma \subset \mathbb{C} \setminus \mathbb{D}$ and function $g : \Gamma \to \mathbb{C}$. Further suppose that the spectrum of A is contained in $\mathbb{C} \setminus \mathbb{D}$. Then we define

$$f(A) := \int_\Gamma g(t)(tI - A)^{-1} \,\mathrm{d}t.$$

When f is analytic, $g = \frac{1}{2\pi i} f$, and Γ is a contour enclosing the spectrum of A, then Definition 2.3 reduces to the usual Cauchy integral definition.

Various matrix function properties will prove useful throughout our analysis.

Theorem 2.4 (Theorem 1.13 in [74]): Let $A \in \mathbb{C}^{n\times n}$ and let f be defined on the spectrum of A. Then

(i) $f(A)A = Af(A)$;

(ii) $f(A^T) = f(A)^T$;

(iii) $f(XAX^{-1}) = Xf(A)X^{-1}$;

(iv) $f(\lambda) \in \text{spec}(f(A))$ for all $\lambda \in \text{spec}(A)$;

(v) $XA = AX$ implies that $Xf(A) = f(A)X$;

(vi) if A is block triangular, then $f(A)$ is block triangular, and $f(A)_{ii} = f(A_{ii})$, where A_{ii} is a block diagonal element of A;

(vii) if $A = \text{diag}(A_{11}, \ldots, A_{mm})$, then $f(A) = \text{diag}(f(A_{11}), \ldots, f(A_{mm}))$;

(viii) $f(I_m \otimes A) = I_m \otimes f(A)$; and

(ix) $f(A \otimes I_m) = f(A) \otimes I_m$.

2.3 Hermitian positive definite and positive real operators

We call an operator $A : \mathbb{C}^n \to \mathbb{C}^n$ *Hermitian positive definite* (HPD) if $A = A^*$, where A^* denotes the adjoint operator, and has positive eigenvalues. We say that A is positive real in the standard Euclidean sense or *Euclidean positive real* (EPR) if for all $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{x} \neq 0$, $\text{Re}(\boldsymbol{x}^* A\boldsymbol{x}) > 0$. Similar definitions hold with non-strict inequality: A is *Hermitian positive semi-definite* (HPSD) if $A = A^*$ and has nonnegative eigenvalues, while A is *Euclidean nonnegative real* (ENNR) if for all $\boldsymbol{x} \in \mathbb{C}^n$, $\text{Re}(\boldsymbol{x}^* A\boldsymbol{x}) \geq 0$.

A useful notion for understanding the positive realness of an operator is its field of values.

Definition 2.5: Let A be an operator on a Hilbert space $\mathbb{H}$ with inner product $\langle \cdot, \cdot \rangle$. The *field of values* $\mathbb{F}_{\langle \cdot, \cdot \rangle}(A)$ is defined as

$$\mathbb{F}_{\langle \cdot, \cdot \rangle}(A) := \left\{ \frac{\langle Ax, x \rangle}{\langle x, x \rangle} : x \in \mathbb{H}, x \neq 0 \right\}.$$

Note that the field of values contains the spectrum of the operator A; additionally, the spectrum of A is defined independently of the inner product $\langle \cdot, \cdot \rangle$.

Lemma 2.6: The following properties are equivalent to the operator $A : \mathbb{C}^n \to \mathbb{C}^n$ being Euclidean positive (nonnegative) real:

(i) A^* is also EPR (ENNR);

(ii) $A + A^*$ is Hermitian positive (semi)definite;

(iii) $\mathbb{F}_{\langle \cdot, \cdot \rangle_2}(A)$ lies in the right half of the complex plane (including the imaginary axis if A is ENNR).

Proof: Let $\boldsymbol{x} \in \mathbb{C}^n$ be nonzero. Part (i) follows by noting that $\boldsymbol{x}^* A^* \boldsymbol{x} = (A\boldsymbol{x})^* \boldsymbol{x} = \overline{\boldsymbol{x}^* A \boldsymbol{x}}$ and thus $\mathrm{Re}\big(\overline{\boldsymbol{x}^* A \boldsymbol{x}}\big) = \mathrm{Re}(\boldsymbol{x}^* A \boldsymbol{x}) \geq 0$. This also gives that $\boldsymbol{x}^*(A + A^*)\boldsymbol{x} = 2\mathrm{Re}(\boldsymbol{x}^* A \boldsymbol{x})$ and thus part (ii). Part (iii) is a straightforward application of Definition 2.5 and the definition for EPR (ENNR). □

Lemma 2.7: Let $A, B \in \mathbb{C}^{n \times n}$.

(i) (congruence) If A is Hermitian and $Q \in \mathbb{C}^{n \times n}$ is nonsingular, then $Q^* A Q$ is Hermitian with the same number of positive, negative, and zero eigenvalues.

(ii) (similarity) If $Q \in \mathbb{C}^{n \times n}$ is nonsingular, then $Q^{-1} A Q$ has the same spectrum as A.

(iii) (semi-congruence) If A is HPSD and $Q \in \mathbb{C}^{n\times m}$ with $m \leq n$, then Q^*AQ is HPSD.

(iv) Let B be EPR (ENNR) and $Q \in \mathbb{C}^{n\times m}$ with $m \leq n$. If Q has full rank, then Q^*BQ is EPR (ENNR). If Q does not have full rank, then we can only guarantee that Q^*BQ is ENNR.

(v) If B is EPR (ENNR), then the real part of its spectrum is positive (nonnegative).

(vi) If A is HPD (HPSD) and B is EPR, then the real part of the spectrum of AB is positive (nonnegative).

(vii) If B is normal and has spectrum with positive (nonnegative) real part, then B is EPR (ENNR).

Proof: Part (i) is well established and otherwise known as "Sylvester's Law of Inertia;" see, e.g., [94, Section 5.5] for a proof. Part (ii) is also well known and simple to prove using the Jordan canonical form of a matrix.

For part (iii), note that $\boldsymbol{y}^*Q^*AQ\boldsymbol{y} = (Q\boldsymbol{y})^*A(Q\boldsymbol{y})$, where $\boldsymbol{y} \in \mathbb{C}^m$. If $\boldsymbol{y}$ is nonzero and Q has full rank, then $Q\boldsymbol{y}$ is nonzero; if Q does not have full rank, then there is a $\boldsymbol{y} \neq 0$ such that $Q\boldsymbol{y} = 0$. Either way, since A is HPSD, $(Q\boldsymbol{y})^*A(Q\boldsymbol{y}) \geq 0$, meaning Q^*AQ is HPSD.

The proof for part (iv) is analogous to that of part (iii); simply consider $\mathrm{Re}(\boldsymbol{y}^*Q^*AQ\boldsymbol{y})$ instead.

Part (v) follows immediately from the definition of positive realness, since the field of values contains the spectrum.

Part (vi) follows from everything before. We first assume that A is HPD; it then has a well-defined and nonsingular square root, allowing us to write

$$AB = A^{1/2}A^{1/2}B = A^{1/2}A^{1/2}BA^{1/2}A^{-1/2}.$$

Part (i) implies that $A^{1/2}BA^{1/2}$ has a spectrum with positive real part, and the similarity transformation $A^{1/2}A^{1/2}BA^{1/2}A^{-1/2}$ preserves the spectrum.

Now suppose that A is HPSD; i.e., it may have a zero eigenvalue and is therefore not invertible. Letting $\epsilon > 0$, we have that $A + \epsilon I$ is HPD and invertible. Following the discussion of the previous paragraph, $(A + \epsilon I)B$ has spectrum with positive real part. Since the spectrum depends continuously on the entries of the matrix, letting $\epsilon \to 0$ gives that AB also has nonnegative real part.

Lastly, if B is normal, then there exists $X \in \mathbb{C}^{n\times n}$ unitary and $D \in \mathbb{C}^{n\times n}$ diagonal such that $B = XDX^*$. Then $B + B^* = X(D + D^*)X^*$. D is trivially EPR (ENNR) since its diagonals are the eigenvalues of B, which have positive (nonnegative) real part. Therefore, the eigenvalues of the diagonal matrix $D + D^*$ are all real and positive, i.e. $D + D^*$ is Hermitian positive definite. Part (vii) then follows by Lemma 2.6. □

Remark 2.8: Note that Lemma 2.7 holds for a general inner product, as long as one replaces the notion of "Hermitian" with "self-adjoint" and regards normality with respect to the general inner product.

2.4 Block notions

2.4.1 Disambiguation

There are two main uses of the term "block" in numerical linear algebra. The first is with an eye towards high-performance computing and the motivation of making algorithms more parallelizable. Given a matrix $A \in \mathbb{C}^{n\times n}$, one can partition A into a number of sub-matrices or "blocks," depending on the given computer architecture and memory movement strategies; A may then be regarded as a *block matrix*. Similarly, given a number of vectors $\{\boldsymbol{b}_i\}_{i=1}^{s}$ on which A or $f(A)$ must act, one can batch or "block" these vectors into tall and skinny matrices or *block vectors* to help with load balancing.

The second comes from the viewpoint of building matrices and vectors over a space $\mathbb{S} \subset \mathbb{C}^{s\times s}$. So that the resulting block objects have convenient structure, we take $\mathbb{S}$ to be a *-subalgebra with identity;[1] examples of such $\mathbb{S}$ are given in Table 3.1. This means that $\mathbb{S}$ is a vector subspace containing I_s that is also closed under matrix multiplication and conjugate transposition. The nature of $\mathbb{S}$ leads to a few importance consequences. For one, being a vector subspace with identity means that $\mathbb{S}$ always contains the subspace $\mathbb{C}I_s$. Additionally, any scalar polynomial of an element in $\mathbb{S}$ is also an element in $\mathbb{S}$. Then the inverses of all nonsingular elements of $\mathbb{S}$ are also members of $\mathbb{S}$, since the inverse of a matrix can be expressed as a scalar polynomial of that matrix (see Definition 2.2 or consequences of the Cayley-Hamilton Theorem).

[1] We thank Michele Benzi for suggesting the use of a *-subalgebra.

The space $\mathbb{S}^m$ denotes the $ms \times s$ block vectors whose $s \times s$ block elements are taken from $\mathbb{S}$, and likewise the space $\mathbb{S}^{m\times m}$ denotes $ms \times ms$ block matrices. In this way, we can regard such block matrices as "flattened" fourth-order tensors; see, e.g., [114]. For further thoughts on block matrix operations, see [65, Section 1.3].

2.4.2 Block eigenvalues

The earliest mention of block eigenvalues in the literature is a technical report by Dennis, Traub, and Weber in 1971 [30]. Throughout this subsection, we rely on this report and a subsequent publication [31].

Definition 2.9: For $A \in \mathbb{C}^{n\times n}$, $(\Lambda, \boldsymbol{Q}) \in \mathbb{C}^{s\times s} \times \mathbb{C}^{n\times s}$ is a *block eigenpair* if s divides n, $\boldsymbol{Q}$ has full rank, and

$$A\boldsymbol{Q} = \boldsymbol{Q}\Lambda.$$

We call Λ a *block eigenvalue* and $\boldsymbol{Q}$ a *block eigenvector*.

It is important to keep in mind that block eigenpairs are merely another way of describing certain kinds of invariant subspaces. If $(\Lambda, \boldsymbol{Q})$ is a block eigenpair of A, then the columns of $\boldsymbol{Q}$ form a basis for an invariant subspace of A, on which the action of A can be represented by Λ. Consequently, the spectrum of block eigenvalues of A are contained in the spectrum of A.

Theorem 2.10 (Theorem 8.1 in [30]): If $(\Lambda, \boldsymbol{Q}) \in \mathbb{C}^{s\times s} \times \mathbb{C}^{n\times s}$ is a block eigenpair of $A \in \mathbb{C}^{n\times s}$, and $\boldsymbol{Q}$ has full rank, then the eigenvalues of Λ are also eigenvalues of A.

In analogy to results for scalar eigenvalues, we consider whether the action of A can be described by a block eigendecomposition. We state the theorems without proof, since they are rather technical, but they can be found, e.g., in [30].

Definition 2.11: A set of block eigenvalues of A is a *complete set of block eigenvalues* of A if the set of all the eigenvalues of the block eigenvalues is the same as the set of eigenvalues of A.

Theorem 2.12 (Theorem 8.2 in [30]): Suppose that $A \in \mathbb{C}^{n\times n}$, and that s divides n. Then A has a complete set of $s \times s$ block eigenvalues.

Note that Theorem 2.12 does not place a restriction on what type of matrix A is.

Definition 2.13: Two block vectors $\boldsymbol{Q}_1, \boldsymbol{Q}_2 \in \mathbb{C}^{n\times s}$ are block orthogonal if $\boldsymbol{Q}_1^* \boldsymbol{Q}_2 = 0$.

In Chapter 3, we restate Definition 2.13 within the setting of a general matrix-valued inner product. For now, the classical definition with the Euclidean inner product is enough to obtain a block eigendecomposition result.

Theorem 2.14 (Theorem 8.4 in [30]): If $A \in \mathbb{C}^{n\times n}$ has $s \times s$ block eigenvalues $\Lambda_1, \ldots, \Lambda_m$, $m = \frac{n}{s}$, with respective block eigenvectors $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_m$ that are pairwise block orthogonal, and if Λ is a block eigenvalue of A, then Λ is a block eigenvalue of $\operatorname{diag}(\Lambda_1, \ldots, \Lambda_m)$. Furthermore,

$$[\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_m]^{-1} A [\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_m] = \operatorname{diag}(\Lambda_1, \ldots, \Lambda_m). \tag{2.4}$$

With the notion of a block eigendecomposition, we can obtain a result similar to Theorem 2.4 (iv) for block eigenvalues.

Theorem 2.15: Let $A \in \mathbb{C}^{n\times n}$ have a block eigendecomposition as in the hypotheses of Theorem 2.14, and let $f : \mathbb{C} \to \mathbb{C}$ be defined on the spectrum of A. Then

$$f(A)\boldsymbol{Q}_j = \boldsymbol{Q}_j f(\Lambda_j), \text{ for all } j = 1, \ldots, m. \tag{2.5}$$

In other words, we have a block eigendecomposition for $f(A)$.

Proof: By equation (2.4) and parts (iii) and (vii) of Theorem 2.4,

$$[\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_m]^{-1} f(A)[\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_m] = \operatorname{diag}(f(\Lambda_1), \ldots, f(\Lambda_m)). \tag{2.6}$$

Note that for $j = 1, \ldots, m$, $f(\Lambda_j)$ is well defined, since by Theorem 2.10 the eigenvalues of a block eigenvalue are a subset of the eigenvalues of A, and f is defined on the spectrum of A. Consequently, equation (2.6) gives the desired result. □

2.5 Matrix polynomials

Following in the footsteps of many others [43, 46, 73, 86, 127, 129, 130], we wish to utilize matrix polynomials to enhance our understanding of block Krylov methods. The thesis by Kent [86] is particularly inspirational, as one of the first works to describe elements of a block Krylov subspace in terms of matrix-valued polynomials. Matrix polynomials are analyzed extensively in the book by Gohberg, Lancaster, and Rodman [62], as well as in a series of papers by Dennis, Traub, and Weber [30, 31, 32]. Although we draw many definitions and foundational results from these two wells of sources, we recast them for our own specific purposes.

Let $\mathbb{P}_d(\mathbb{K})$ denote the space of polynomials of degree d over some space of matrices $\mathbb{K}$; in particular, we mean that the polynomials of $\mathbb{P}_d(\mathbb{K})$ take their coefficients from $\mathbb{K}$ and refer to these polynomials as *matrix polynomials*. It is helpful to

think of an element P of $\mathbb{P}_d(\mathbb{K})$ as an ordered list of coefficients $(\Gamma_0, \ldots, \Gamma_d) \subset \mathbb{K}^{d+1}$, where $P(Z) = \sum_{k=0}^{d} Z^k \Gamma_k$. Note that we assume the argument Z is right-multiplied by the coefficients Γ_k. Left multiplication is also possible, but we do not consider it here for reasons that will soon be clear.

We focus on polynomials over the *-subalgebra $\mathbb{S} \subset \mathbb{C}^{s\times s}$ defined in Section 2.4. Elements of $\mathbb{P}_d(\mathbb{S})$ are perhaps most naturally regarded as operators from $\mathbb{C}^{s\times s}$ to $\mathbb{C}^{s\times s}$, but just as scalar polynomials can be defined to act on matrices, these *matrix polynomials* can be defined to act on other objects as well. We consider the following three interpretations of $P \in \mathbb{P}_d(\mathbb{S})$ and let context determine which particular interpretation is in use:

- $P : \mathbb{C}^{s\times s} \to \mathbb{C}^{s\times s}$, $P(Z) = \sum_{k=0}^{d} Z^k \Gamma_k$;
- $P : \mathbb{C} \to \mathbb{C}^{s\times s}$, $P(z) = \sum_{k=0}^{d} z^k \Gamma_k$, known in the literature as a *λ-matrix* [30, 31, 32, 62, 93]; and
- $P : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$, $P(A) \circ \boldsymbol{V} = \sum_{k=0}^{d} A^k \boldsymbol{V} \Gamma_k$, where the $\circ$ notation is introduced in the thesis by Kent [86].

The final interpretation is the reason we only consider multiplying by matrix coefficients on the right; such polynomials arise in our description of block Krylov subspaces in Section 3.1.

Two polynomials $P(z) = \sum_{k=0}^{d} z^k \Gamma_k$ and $\widetilde{P}(z) = \sum_{k=0}^{d} z^k \widetilde{\Gamma}_k$ are equal as long as $\Gamma_k = \widetilde{\Gamma}_k$ for all $k = 0, \ldots, d$. Dividing a matrix polynomial by an element of $\mathbb{S}$ is defined as right-dividing each coefficient, i.e., for some nonsingular $S \in \mathbb{S}$, $P(z)S^{-1} = \sum_{k=0}^{d} z^k \Gamma_k S^{-1}$. Note the distinction in notation when the $\circ$ operator is

involved:

$$(P(\mathcal{H})S^{-1}) \circ \boldsymbol{V} = \sum_{k=0}^{d} \mathcal{H}^k \boldsymbol{V} \Gamma_k S^{-1} = (P(\mathcal{H}) \circ \boldsymbol{V})S^{-1}$$

vs.

$$P(\mathcal{H}) \circ (\boldsymbol{V}S^{-1}) = \sum_{k=0}^{d} \mathcal{H}^k \boldsymbol{V} S^{-1} \Gamma_k$$

One can also divide one matrix polynomial by another.

Definition 2.16: (i) A λ-matrix D is *regular* if there exists z such that $\det(D(z)) \neq 0$.

(ii) Let P, K, R and D be λ-matrices, where P has degree d and D is regular with degree less than d. K is defined as the *left quotient* and R as the *left remainder* of P on division by D if

$$P(z) = D(z)K(z) + R(z).$$

(iii) We say that P is *left divisible* by D if $R \equiv 0$.

All λ-matrices we consider are regular, unless otherwise noted. Consequently, the division process is unique; see, e.g., [58, Chapter 4, Section 2].

Theorem 2.17: The λ-matrix $P(z)$ is divisible on the left by $zI - S$ if and only if $P(S) = 0$.

Proof: This theorem originates as Theorem 3.3 and its corollary in [93]. We reproduce the proof here, as the book may be difficult to find, and the proof itself is

illustrative. It is enough to show that the remainder of $P(z)$ divided by $zI-S$ is $P(S)$:

$$\begin{aligned}
P(z) &= z^d\Gamma_d + z^{d-1}\Gamma_{m-1} + \cdots + \Gamma_0 \\
&= (zI - S)z^{d-1}\Gamma_d + z^{d-1}(S\Gamma_d + \Gamma_{d-1}) + z^{d-2}\Gamma_{d-2} + \cdots + \Gamma_0 \\
&= (zI - S)\big(z^{d-1}\Gamma_d + z^{d-2}(S\Gamma_d + \Gamma_{d-1})\big) \\
&\quad + z^{d-2}(S^2\Gamma_d + S\Gamma_{d-1} + \Gamma_{d-2}) + z^{d-3}\Gamma_{d-3} + \cdots + \Gamma_0 \\
&= (zI - S)\big(z^{d-1}\Gamma_d + z^{d-2}(S\Gamma_d + \Gamma_{d-1}) + \cdots \\
&\quad + (S^{d-1}\Gamma_d + S^{d-2}\Gamma_{d-1} + \cdots + \Gamma_1)\big) \\
&\quad + S^d\Gamma_d + S^{d-1}\Gamma_{d-1} + \cdots + \Gamma_0 \\
&= (zI - S)Q(z) + P(S), \text{ where}
\end{aligned}$$

$$Q(z) := z^{d-1}\Gamma_d + z^{d-2}(S\Gamma_d + \Gamma_{d-1}) + \cdots + (S^{d-1}\Gamma_d + S^{d-2}\Gamma_{d-1} + \cdots + \Gamma_1). \qquad \square$$

2.5.1 Solvents and latent roots

As with scalar polynomials, there exists a notion of roots for matrix polynomials; but unlike scalar polynomials, there is no Fundamental Theorem of Algebra. However, under special assumptions, we can factor matrix polynomials.

Definition 2.18: Let $P \in \mathbb{P}_d(\mathbb{S})$. A matrix $S \in \mathbb{S}$ is called a *left solvent* of P if $P(S) = 0$.

From now onward, we omit "left" when referring to quotients, divisibility, and solvents.

Given a set of solvents for a matrix polynomial, one might want to inductively apply the technique from Theorem 2.17 to write the polynomial as the product of

linear factors. In general, this is not possible. However, we can obtain something useful if we make a few assumptions.

Theorem 2.19: Let $P \in \mathbb{P}_d(\mathbb{S})$ and let $\{S_j\}_{j=1}^d \subset \mathbb{C}^{s\times s}$ denote the set of solvents of P. Suppose the pairwise differences of the solvents are nonsingular. Then

$$P(z) = (zI - S_1)\cdots(zI - S_d)Q_0, \tag{2.7}$$

for some $Q_0 \in \mathbb{S}$.

Proof: From the proof of Theorem 2.17, we obtain a quotient $Q_{d-1} \in \mathbb{P}_{d-1}$ such that

$$P(z) = (zI - S_1)Q_{d-1}(z).$$

Since $P(S_2) = 0$, we can conclude that $Q_{d-1}(S_2) = 0$ as well since $S_2 - S_1$ is invertible. Then we obtain the next quotient $Q_{d-2} \in \mathbb{P}_{d-2}$ such that

$$Q_{d-1}(z) = (zI - S_2)Q_{d-2}(z).$$

Since $S_3 - S_2$ is invertible, we obtain the next quotient, and so forth. Inductively,

$$P(z) = (zI - S_1)\cdots(zI - S_d)Q_0,$$

where $Q_0 \in \mathbb{S}$. □

Theorem 2.19 is sufficient for our purposes in Section 4.2.5. However, for the sake of context, we mention that there are other scenarios in which one can guarantee a priori that a matrix polynomial can be decomposed into linear factors, in particular with monic polynomials, for which the leading coefficient is the identity I.

Definition 2.20: Let $P \in \mathbb{P}_d(\mathbb{S})$.

(i) A scalar $\lambda \in \mathbb{C}$ is a *latent root* if $P(\lambda)$ is singular. Equivalently, the latent roots of a matrix polynomial P are precisely the zeros of the scalar polynomial $\det(P(\lambda))$.

(ii) The set $\{S_j\}_{j=1}^{d} \subset \mathbb{S}$ is a *complete set of solvents* if P is monic and has ds distinct latent roots, and if the ds eigenvalues of $\{S_j\}_{j=1}^{d}$ match the latent roots exactly.

Theorem 2.21: If the latent roots of $P \in \mathbb{P}_d(\mathbb{S})$ are distinct and if P is monic, then P has a complete set of solvents $\{S_j\}_{j=1}^{d} \subset \mathbb{S}$, and can be written as the product of *linear factors*

$$P(z) = (zI - S_1)(zI - S_2)\cdots(zI - S_d).$$

Proof: See [31, Theorem 4.1, Corollary 4.2, and Corollary 4.3], as well as [62, Theorem 3.21], of which this result is a particular case. □

To illustrate these concepts, consider the polynomial

$$P(z) = z^2 I - z\begin{bmatrix} -2 & 4 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix}.$$

One can check that $S_1 = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$ and $S_2 = \begin{bmatrix} -3 & 2 \\ 0 & -1 \end{bmatrix}$ are solvents of P. Furthermore, $-3, -1, 1$, and 3 are latent roots of P. Therefore, $\{S_1, S_2\}$ is a complete set of solvents of P, and P can be factored as

$$P(z) = \left(zI - \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\right)\left(zI - \begin{bmatrix} -3 & 2 \\ 0 & -1 \end{bmatrix}\right).$$

2.5.2 Interpolating matrix polynomials

In the same vein as Definition 2.2, we may want to match a matrix function to a matrix polynomial, in which case, we need the notion of interpolating matrix polynomials.

There are different, albeit related, notions for interpolating matrix polynomials. Given a set of solvents, Dennis, Traub, and Weber define a set of fundamental matrix polynomials and a block Vandermonde matrix in a fashion analogous to the scalar case [31]. As long as the block Vandermonde matrix is nonsingular, the set of fundamental matrix polynomials is well defined and interpolation can be carried out. Our notion of interpolating matrix polynomials is closer to the development of [86, Section 4.2].

Definition 2.22: Given $\mathcal{H} \in \mathbb{S}^{m\times m}$, $\boldsymbol{V} \in \mathbb{S}^m$, and f such that $f(\mathcal{H}) : \mathbb{S}^m \to \mathbb{S}^m$ is defined, we say that $Q \in \mathbb{P}_{m-1}(\mathbb{S})$ *interpolates* f *on the pair* $(\mathcal{H}, \boldsymbol{V})$ if

$$Q(\mathcal{H}) \circ \boldsymbol{V} = f(\mathcal{H})\boldsymbol{V}.$$

Define the block Vandermonde matrix

$$\boldsymbol{\mathcal{V}} := [\boldsymbol{V}|\mathcal{H}\boldsymbol{V}|\cdots|\mathcal{H}^{m-1}\boldsymbol{V}]. \tag{2.8}$$

If $\boldsymbol{\mathcal{V}}$ is nonsingular, then the system

$$\boldsymbol{\mathcal{V}}\boldsymbol{\Gamma} = f(\mathcal{H})\boldsymbol{V}$$

has a solution $\boldsymbol{\Gamma} = \begin{bmatrix} \Gamma_0 \\ \vdots \\ \Gamma_{m-1} \end{bmatrix} \in \mathbb{S}^m$, which is equivalent to the existence of an interpolating matrix polynomial $Q(z) = \sum_{j=0}^{m-1} z\Gamma_j$ for f and $\boldsymbol{V}$. Note that Q is not

necessarily equivalent to the scalar interpolating polynomial associated to the function f acting on $\mathcal{H}$ guaranteed by Definition 2.2. For our purposes, when we invoke an interpolating matrix polynomial characterization of $f(\mathcal{H})\boldsymbol{V}$, we assume it exists.

Definition 2.23: A *block characteristic polynomial* of $\mathcal{H} \in \mathbb{S}^{m\times m}$ with respect to $\boldsymbol{V} \in \mathbb{S}^m$ is a matrix polynomial $P \in \mathbb{P}_m(\mathbb{S})$ such that

$$P(\mathcal{H}) \circ \boldsymbol{V} = 0.$$

By letting $Q \in \mathbb{P}_{m-1}(\mathbb{S})$ be the matrix polynomial interpolating z^{-1} on $(\mathcal{H}, \boldsymbol{V})$, then $P(z) := I - zQ(z)$ is a block characteristic polynomial for $(\mathcal{H}, \boldsymbol{V})$. Under certain assumptions, the block characteristic polynomial takes on the "spirit" of $\mathcal{H}$, in the sense that its latent roots match the eigenvalues of $\mathcal{H}$, and its solvents are similarity transformations of the block eigenvalues of $\mathcal{H}$. The following theorem makes this more precise.

Theorem 2.24: Suppose $\mathcal{H} \in \mathbb{S}^{m\times m}$ has a block eigendecomposition,

$$\mathcal{H} = \mathcal{U}\mathcal{T}\mathcal{U}^{-1},$$

where $\mathcal{U} \in \mathbb{S}^{m\times m}$ is invertible, $\mathcal{T} = \operatorname{diag}(\{\Theta_j\}_{j=1}^m) \in \mathbb{S}^{m\times m}$, and $\{\Theta_j\}_{j=1}^m \subset \mathbb{S}$ are diagonalizable and the block eigenvalues of $\mathcal{H}$. Let $\boldsymbol{V} \in \mathbb{S}^m, \boldsymbol{V} \neq 0$, be such that each block entry W_j of $\boldsymbol{W} := [W_1 \cdots W_m] := \mathcal{U}^{-1}\boldsymbol{V}$ is invertible, and let $Q \in \mathbb{P}_{m-1}(\mathbb{S})$ interpolate z^{-1} on $(\mathcal{H}, \boldsymbol{V})$. Defining $P \in \mathbb{P}_m(\mathbb{S})$ as $P(z) := I - zQ(z)$, it holds that

(i) $S_j := W_j^{-1}\Theta_j W_j$ are solvents of P, for all $j = 1, \ldots, m$;

(ii) the eigenvalues of $\mathcal{H}$ exactly match the latent roots of P; and

(iii) if $S_i - S_j$ is nonsingular for all $i \neq j$, then

$$P(z) = (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_m^{-1}), \tag{2.9}$$

where $\widetilde{S}_j := (S_1 \cdots S_{j-1}) S_j (S_1 \cdots S_{j-1})^{-1}$.

Proof: (i) Let $\{\Gamma_k\}_{k=0}^{m-1} \subset \mathbb{S}$ denote the coefficients of Q. Then

$$Q(\mathcal{H}) \circ \boldsymbol{V} = \mathcal{U}\mathcal{T}^{-1}\mathcal{U}^{-1}\boldsymbol{V} = \mathcal{U}\mathcal{T}^{-1}\boldsymbol{W}, \tag{2.10}$$

while

$$Q(\mathcal{H}) \circ \boldsymbol{V} = \sum_{k=0}^{m-1} \mathcal{H}^k \boldsymbol{V}\Gamma_k = \sum_{k=0}^{m-1} \mathcal{U}\mathcal{T}^k\mathcal{U}^{-1}\boldsymbol{V}\Gamma_k = \mathcal{U}Q(\mathcal{T}) \circ \boldsymbol{W}. \tag{2.11}$$

Combining equalities (2.10) and (2.11) and multiplying both sides by $\mathcal{U}^{-1}$ gives that

$$Q(\mathcal{T}) \circ \boldsymbol{W} = \mathcal{T}^{-1}\boldsymbol{W}.$$

The block structure of $\mathcal{T}$ further implies that

$$Q(\Theta_j) \circ W_j = \Theta_j^{-1} W_j, \tag{2.12}$$

and multiplying both sides of equation (2.12) by W_j^{-1} finally results in

$$W_j^{-1} Q(\Theta_j) \circ W_j = W_j^{-1} \sum_{k=0}^{m-1} \Theta_j^k W_j \Gamma_k = \sum_{k=0}^{m-1} W_j^{-1}\Theta_j^k W_j \Gamma_k = W_j^{-1}\Theta_j^{-1} W_j.$$

In summary, for $S_j = W_j^{-1}\Theta_j W_j$,

$$Q(S_j) = S_j^{-1}, \text{ for all } j = 1, \ldots, m. \tag{2.13}$$

By the definition of P, $P(S_j) = 0$.

(ii) Fix j. Since each Θ_j is diagonalizable, so is S_j, i.e.,

$$S_j = X_j T_j X_j^{-1},$$

where $X_j \in \mathbb{C}^{s\times s}$ and $T_j := \text{diag}(\theta_{1,j}, \ldots, \theta_{s,j})$, with $\{\theta_{ij}\}_{i=1}^s$ being the eigenvalues of Θ_j. Let $\{\widetilde{\Gamma}_k\}_{k=0}^m$ denote the coefficients of P. Consequently,

$$\begin{aligned} 0 = P(S_j) &= X_j \sum_{k=0}^m T_j^k X_j^{-1} \widetilde{\Gamma}_k \\ &= X_j \sum_{k=0}^m \begin{bmatrix} \theta_{1j}^k & & \\ & \ddots & \\ & & \theta_{sj}^k \end{bmatrix} X_j^{-1} \widetilde{\Gamma}_k \end{aligned}$$

Then multiplying on both sides by X_j^{-1} gives that

$$\sum_{k=0}^m \begin{bmatrix} \theta_{1j}^k & & \\ & \ddots & \\ & & \theta_{sj}^k \end{bmatrix} X_j^{-1} \widetilde{\Gamma}_k = 0,$$

implying that the ith row of $\sum_{k=0}^m \theta_{ij}^k X_j^{-1} \widetilde{\Gamma}_k = X_j^{-1} P(\theta_{ij})$ is all zero. Therefore, $X_j^{-1} P(\theta_{ij})$ has determinant zero, but since $\det(X_j^{-1}) \neq 0$, then $\det(P(\theta_{ij}) = 0$, for all $i = 1, \ldots, s$. Furthermore, the above holds for all $j = 1, \ldots, m$, so $\bigcup_{j=1}^m \{\theta_{ij}\}_{i=1}^s$ are precisely the latent roots of P. From Theorem 2.10, $\bigcup_{j=1}^m \{\theta_{ij}\}_{i=1}^s$ are also the eigenvalues of $\mathcal{H}$, thus concluding the proof.

(iii) By Theorem 2.19, we can write

$$P(z) = (zI - S_1) \cdots (zI - S_m) K_0,$$

where $K_0 = (-1)^m (S_1 \cdots S_m)^{-1}$, since $P(0) = I$. Define

$$\widetilde{P}_k(z) := (zI - S_1)(zI - S_2) \cdots (zI - S_k),$$

for each $k = 1, \ldots, m$. Then, in particular,

$$\widetilde{P}_1(z) = zI - S_1 = (I - zS_1^{-1})(-S_1),$$

which serves as our base case. Suppose that for all $1 \leq k \leq m-1$,

$$\widetilde{P}_k(z) = (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_k^{-1})(-1)^k (S_1 \cdots S_k),$$

where for each $j = 1, \ldots, k$, $\widetilde{S}_j = (S_1 \cdots S_{j-1}) S_j (S_1 \cdots S_{j-1})^{-1}$. Then

$$\begin{aligned}
\widetilde{P}_m(z) &= \widetilde{P}_{m-1}(zI - S_m) \\
&= (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_{m-1}^{-1})(-1)^{m-1}(zS_1 \cdots S_{m-1} - S_1 \cdots S_{m-1}S_m) \\
&= (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_{m-1}^{-1})(S_1 \cdots S_m - zS_1 \cdots S_{m-1})(-1)^m \\
&= (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_{m-1}^{-1})(I - z(S_1 \cdots S_{m-1})(S_1 \cdots S_m)^{-1})(-1)^m (S_1 \cdots S_m).
\end{aligned}$$

Noting that $(S_1 \cdots S_{m-1})(S_1 \cdots S_m)^{-1} = (S_1 \cdots S_{m-1}) S_m^{-1} (S_1 \cdots S_{m-1})^{-1}$ and defining $\widetilde{S}_m := (S_1 \cdots S_{m-1}) S_m (S_1 \cdots S_{m-1})^{-1}$ concludes the induction process. Then we have that

$$P(z) = (I - z\widetilde{S}_1^{-1}) \cdots (I - z\widetilde{S}_m^{-1})(-1)^m (S_1 \cdots S_m) K_0.$$

With $(-1)^m (S_1 \cdots S_m) K_0 = I$, we obtain the final result. □

2.6 Stieltjes functions

A Stieltjes function is a function $f : \mathbb{C} \setminus (-\infty, 0] \to \mathbb{C}$ that can be written as a Riemann-Stieltjes integral:

$$f(z) = \int_0^\infty \frac{1}{z+t} \, \mathrm{d}\mu(t), \tag{2.14}$$

where μ is monotonically increasing and nonnegative on $[0, \infty)$ and $\int_0^\infty \frac{1}{t+1} \, \mathrm{d}\mu(t) < \infty$. We note that $f(A)$ is defined as long as the spectrum of A, which could be complex, contains no non-positive real values.

See, e.g., [72] for more information about Stieltjes functions. We focus on the family of functions given as follows, for $\alpha \in (0, 1)$:

$$z^{-\alpha} = \frac{\sin((1-\alpha)\pi)}{\pi} \int_0^\infty \frac{1}{z+t} \, \mathrm{d}\mu(t), \qquad \text{with} \quad \mathrm{d}\mu(t) = t^{-\alpha} \, \mathrm{d}t. \tag{2.15}$$

We pay special attention to $z^{-1/2}$ and its role in the sign function, which can be written as $\operatorname{sign}(z) = (z^2)^{-1/2}$; see Section 6.2 and the numerical examples in Chapter 7.

2.7 Gauss quadrature rules

Quadrature, or numerical integration, is a standard method for approximating the value of finite integrals. Let μ be a measure on an interval $[a, b]$ whose moments are finite, and let the function $f : \mathbb{C} \to \mathbb{C}$ be such that its Riemann-Stieltjes integral exists. Then an N-point quadrature rule for approximating $\int_a^b f(t) \, \mathrm{d}\mu(t)$ is a set of weights $\{w_i\}_{i=1}^N$ and nodes $\{t_i\}_{i=1}^N$ such that

$$\int_a^b f(t) \, \mathrm{d}\mu(t) = \sum_{i=1}^N w_i f(t_i) + R_N[f],$$

where $R_N[f]$ is the remainder. The rule is of degree d if $R_N[p] = 0$ for all polynomials of degree d; it is exact of degree d if in addition there exists q of degree $d+1$ such that $R[q] \neq 0$. For more on quadrature rules, including their relationship to interpolation, see, e.g., [28].

The idea behind Gauss quadrature rules is to solve for weights and nodes so that the resulting rule has high degree. In particular, an N Gauss rule is exact of degree $2N-1$. The monograph by Golub and Meurant [63] expounds the fascinating relationship among Gauss quadrature rules, orthogonal polynomials, and the Lanczos procedure.

Variations of standard Gauss rules are also possible. In particular, the Gauss-Radau rule determines the weights and remaining $N-1$ nodes when one node is fixed as one of the end points of integration. The Gauss-Lobatto rule is similar, except that both endpoints of the interval are fixed so that only $N-2$ nodes must be determined. Procedures for both are contained in [63].

Block Gauss quadrature rules can also be defined, where the given measure μ is matrix-valued. In this case, the weights and nodes are also matrix-valued. Analogous to the scalar case, there is a connection to matrix polynomials and the block Lanczos procedure (see Algorithm 3.1.1). We consider block Gauss quadrature rules in the context of the block framework developed in Chapter 3. See [63] for an introduction via 2×2 blocks, and [116] for a treatment of general $k \times k$ blocks.

2.8 Matrix derivatives

Suppose $A : \mathbb{C} \to \mathbb{C}^{n\times n}$ is a matrix of scalar functions whose ijth entry is given as $a_{ij}(z)$. We define $\frac{d}{dz}[A(z)]$ to be the matrix of derivatives, i.e., the matrix whose ijth entry is given as $\frac{d}{dz}[a_{ij}(z)]$. A number of intuitive results hold for matrix derivatives.

Lemma 2.25: Let $A, B : \mathbb{C} \to \mathbb{C}^{n\times n}$ be such that each of their entries is a function of z. Let $C \in \mathbb{C}^{n\times n}$ be a constant matrix. Assume that A and B are differentiable, i.e., that

$$\lim_{h\to 0} \frac{A(z+h) - A(z)}{h} =: \frac{d}{dz}[A(z)],$$

and likewise for B. Then

(i) $\frac{d}{dz}[A(z)B(z)] = \frac{d}{dz}[A(z)]B(z) + A(z)\frac{d}{dz}[B(z)]$;

(ii) $\frac{d}{dz}[CA(z)] = C\frac{d}{dz}[A(z)]$;

(iii) $\frac{d}{dz}[A(z)^*] = \frac{d}{dz}[A(z)]^*$; and

(iv) $\frac{d}{dz}[A(z)^{-1}] = -A(z)^{-1}\frac{d}{dz}[A(z)]A(z)^{-1}$.

Proof: Parts (i)-(iii) follow by basic calculus. The proof for part (iv) is a bit more subtle. Note that

$$\begin{aligned}\frac{A(z+h)^{-1} - A(z)^{-1}}{h} &= \frac{A(z+h)^{-1}\left(A(z) - A(z+h)\right)A(z)^{-1}}{h} \\ &= -A(z+h)^{-1}\frac{A(z+h) - A(z)}{h}A(z)^{-1},\end{aligned}$$

whose limit as $h \to 0$ is the desired result. □

Note that if A is Hermitian, then by part (iii) of Lemma 2.25, the derivative is also Hermitian.

2.9 List of abbreviations

In a work this size, it is easy to lose track of where an acronym is first defined. We collect them here for reference.

- CG: Conjugate Gradients
- ENNR: Euclidean nonnegative real
- EPR: Euclidean positive real
- FOM: Full Orthogonalization Method
- $(\text{FOM})^2$: Full Orthogonalization Method for Functions Of Matrices
- GMRES: Generalized Minimal RESidual
- HPSD: Hermitian positive semi-definite
- HPD: Hermitian positive definite
- MINRES: MINimal RESidual

- RL: Radau-Lanczos
- SA: self-adjoint

The prefix of "B" to an acronym denotes "block." With this in mind, the meaning of acronyms such as "BCG," "BPR," "BFOM," and "BGMRES" should be clear.

CHAPTER 3

BLOCK KRYLOV SUBSPACES WITH RESTARTS FOR SHIFTED LINEAR SYSTEMS

Krylov subspace methods are among the most popular iterative methods for solving linear systems $A\boldsymbol{x} = \boldsymbol{b}$ [122]. Their success hinges on the efficient computation of matrix-vector products $A\boldsymbol{b}$, where $A : \mathbb{C}^n \to \mathbb{C}^n$ is sparse and often too large to store or compute with directly, and $\boldsymbol{b} \in \mathbb{C}^n$. With the advent of Level 3 Sparse Basic Linear Algebra Subprogram (BLAS3) standards for matrix-matrix products $A\boldsymbol{B}$, where $\boldsymbol{B} \in \mathbb{C}^{n\times s}$, block Krylov subspace methods have gained more popularity. (For details about these standards see, e.g., [37].) Indeed, it is well known that $A\boldsymbol{B}$ can be computationally more advantageous than computing each column $A\boldsymbol{b}_i$ separately and concatenating them; see, e.g., [11, 15, 115, 111]. Furthermore, the demands of high-performance computing have pushed for cheaper matrix-matrix product kernels, so that operations like $A\boldsymbol{B}$ can be made extremely fast relative to the time needed for communication between parallel processors, thus making them a common technique for "communication avoidance" [33, 66, 80].

Block Krylov methods have additional benefits from a numerical algebraic point of view. Because a block Krylov subspace utilizes information from multiple columns at once, one might expect fewer iterations required for convergence. Block methods also allow for computing multiple eigenvalues at once [24, 82, 85], and they are a natural choice for solving matrix equations [45, 73].

The end goal for this chapter and the next is to develop a general framework and theory for block Krylov methods in order to solve the family of shifted linear systems

$$(A + tI)\boldsymbol{X}(t) = \boldsymbol{B}, \tag{3.1}$$

where $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$ is large and sparse, $\boldsymbol{B} \in \mathbb{C}^{n\times s}$, and t is such that $A + tI$ is nonsingular.

3.0.1 Disambiguation and history

Despite the abundance of block methods in the literature, there is some ambiguity regarding what a "block" method is. For example, one of the earliest appearances of the term "block" in the context of Krylov basis building is by Ruhe in [118]; see Algorithm 3.0.1 for Ruhe's variant of the block Arnoldi method. In our formalism, we require that a block Krylov method be dominated by matrix-matrix products $A\boldsymbol{B}$, so that sparse BLAS3 can be exploited. In which case, we do not regard Ruhe's variant as a block method.

Another issue is that there are multiple ways of blocking and building a block basis, and each has a significant effect on the implementation of the algorithm and resulting algebraic properties of the solution. Gutknecht [69] and Elbouyahyaoui,

Algorithm 3.0.1: Ruhe's Arnoldi

Given A, s orthonormal vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_s$, m

Compute $B = \|\boldsymbol{B}\|_2$ and $\boldsymbol{V}_1 = \boldsymbol{B}B^{-1}$

Set $\boldsymbol{V}_0 = 0$, $H_{0,1} = B$

for $j = s, s+1, \ldots, s+m-1$ **do**

  Set $k = j - s + 1$

  Compute $\boldsymbol{w} = A\boldsymbol{v}_k$

  for $i = 1, 2, \ldots, j$ **do**

    $h_{i,k} = \boldsymbol{w}^* \boldsymbol{v}_i$

    $\boldsymbol{w} = \boldsymbol{w} - h_{i,k}\boldsymbol{v}_i$

  end for

  Compute $h_{j+1,k} = \|\boldsymbol{w}\|_2$ and $\boldsymbol{v}_{j+1} = \boldsymbol{w}/h_{j+1,k}$

end for

return B, $\boldsymbol{\mathcal{V}}_m = [\boldsymbol{V}_1 | \ldots | \boldsymbol{V}_m]$, $\mathcal{H}_m = (H_{j,k})_{j,k=1}^m$, $\boldsymbol{V}_{m+1}$, and $H_{m+1,m}$

Messaoudi, and Sadok [46] are some of the first authors to begin differentiating between the two main blocking techniques, which we refer to as "classical" and "global." The work by these authors is our main source of inspiration for extracting and formalizing the underlying mechanisms of block Krylov methods. In the following, we present a brief historical overview and categorization of block methods present in the literature; precisely what the block inner product is will be made clear in Section 3.1. Though thorough, the sources listed are not exhaustive: block methods are found in many shapes and sizes and are used for eigenvalue solvers, linear systems, and matrix equations alike.

The classical block inner product. Block Krylov methods are first used for eigenvalue solvers in 1974 by Cullum and Donath [24] and in 1977 by Golub and Underwood [64]. We call the underlying block inner product (see Section 3.1) for these methods the "classical" block inner product. In 1980, O'Leary applies block methods in the form of block conjugate gradients (BCG) to solve linear systems with multiple right-hand sides [108]. Since then, BCG has been explored thoroughly in the literature. Much attention has been paid to the issue of linear dependence among columns of the Krylov basis vectors (see Remark 3.10 and [16, 36, 105, 106]) or breakdown scenarios [20]. Other methods relying on the block Krylov basis have also been developed, such as block Lanczos [119], two-side block Lanczos [7], and block FOM (BFOM) [133, 134].

Classical Block GMRES (BGMRES) comprises a field by itself. Simoncini and Gallopoulos first propose BGMRES, along with a thorough theoretical analysis [127, 129, 130]. Other work has dealt with acceleration techniques [102], block grade

and block QR factorizations [69, 70], performance [11], breakdown scenarios [117], and preconditioning and deflation techniques [21].

Many other methods rely on classical block Krylov subspaces. Recycling shares many properties with block methods [111, 133]. Block SYMMLQ and MINRES are explored in [124, 132], and block QMR in [50]. The block Arnoldi procedure underlying all of these methods is also used for matrix equations [2, 45, 110] and model reduction [1].

The global block inner product. Jbilou, Messaoudi, and Sadok debut global FOM and GMRES for matrix equations in 1999 [83], but the nomenclature is introduced by Saad in the first edition of [122]. Thorough convergence and algebraic studies are conducted in [18, 73, 46], and it is precisely these works which begin to uncover an underlying framework between the classical and global Krylov methods.

As with classical methods, variations and different uses of global methods have been developed. They are used for model reduction in [1] and matrix equations in [13]. Bi-conjugate gradients, QMR, BiCGStab, and other variations are developed in [115, 143].

Loop-interchange block inner product. These methods involve "exchanging" a loop in the implementation of Arnoldi for multiple vectors so that block operations are used instead; another term more commonly used in computer science is "batching." This technique has likely been considered many times before, but it is first formally named and studied by Rashedi, Ebadi, Birk, and Frommer in 2016 [115].

3.0.2 Krylov subspace methods

We recapitulate some of the basic details of (non-block) Krylov spaces. Given $A: \mathbb{C}^n \to \mathbb{C}^n$, and $\boldsymbol{b} \in \mathbb{C}^n$, the mth Krylov subspace of A and $\boldsymbol{b}$ is defined as

$$\mathscr{K}_m(A, \boldsymbol{b}) := \text{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}.$$

An orthonormal basis can be generated with the Arnoldi or Lanczos procedure (see Algorithm 3.0.2), leading to the *Arnoldi (Lanczos) relation*

$$A\boldsymbol{V}_m = \boldsymbol{V}_m H_m + \boldsymbol{v}_{m+1} h_{m,m+1} \widehat{\boldsymbol{e}}_m^* = \boldsymbol{V}_{m+1} \underline{H}_m, \tag{3.2}$$

where $\underline{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m}\boldsymbol{e}_m^* \end{bmatrix}$, $\boldsymbol{v}_{m+1}$ and the columns of $\boldsymbol{V}_m$ are orthonormal, and H_m is an $m \times m$ upper Hessenberg matrix. We assume the procedure does not breakdown, and regardless of whether Arnoldi or Lanczos is used, we insist that the entire basis $\boldsymbol{V}_m$ be returned, for reasons related to matrix function approximations; see Chapter 5 and [78]. Naturally, when the Lanczos procedure is used and A is thus Hermitian, H_m reduces to a tridiagonal, Hermitian matrix, for which less storage can be used. For other details and variations regarding Krylov subspace methods for a single vector, we direct the reader to [38, 122, 131].

For comparison with the block case, we recall some well-known results for FOM and GMRES. Let the FOM and GMRES approximations to $A\boldsymbol{x} = \boldsymbol{b}$ be given as

$$\boldsymbol{x}_m^F := \boldsymbol{V}_m H_m^{-1} \widehat{\boldsymbol{e}}_1 \beta, \text{ and}$$

$$\boldsymbol{x}_m^G := \boldsymbol{V}_{m+1} \underline{H}_m^+ \widehat{\boldsymbol{e}}_1 \beta,$$

respectively, where the superscript $+$ denotes the Moore-Penrose inverse. Let $\boldsymbol{x}_*$ denote the exact solution to $A\boldsymbol{x} = \boldsymbol{b}$. Assume that A is HPD and let $0 < \lambda_{\min} \leq \lambda_{\max}$

Algorithm 3.0.2: Arnoldi and Lanczos procedures

Given A, $\boldsymbol{b}$, $\langle \cdot, \cdot \rangle$, induced norm $\|\cdot\|$, m

Compute $\beta = \|\boldsymbol{b}\|$ and $\boldsymbol{v}_1 = \boldsymbol{b}/\beta$

if A is Hermitian with respect to $\langle \cdot, \cdot \rangle$ **then**

  Set $\boldsymbol{v}_0 = 0$, $h_{0,1} = \beta$

  for $k = 1, \ldots, m$ **do**

    $\boldsymbol{w} = A\boldsymbol{v}_k - \boldsymbol{v}_{k-1} h_{k-1,k}$

    $h_{k,k} = \langle \boldsymbol{v}_k, \boldsymbol{w} \rangle$

    $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{v}_k h_{k,k}$

    Compute $h_{k+1,k} = \|\boldsymbol{w}\|$ and $\boldsymbol{v}_{k+1} = \boldsymbol{w} h_{k+1,k}^{-1}$

    Set $h_{k,k+1} = \overline{h_{k+1,k}}$

  end for

else

  for $k = 1, \ldots, m$ **do**

    Compute $\boldsymbol{w} = A\boldsymbol{v}_k$

    for $j = 1, \ldots, k$ **do**

      $h_{j,k} = \langle \boldsymbol{v}_j, \boldsymbol{w} \rangle$

      $\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{v}_j h_{j,k}$

    end for

    Compute $h_{k+1,k} = \|\boldsymbol{w}\|$ and $\boldsymbol{v}_{k+1} = \boldsymbol{w} h_{k+1,k}^{-1}$

  end for

end if

return β, $\boldsymbol{V}_m = [\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_m]$, $H_m = (h_{j,k})_{j,k=1}^{m}$, $\boldsymbol{v}_{m+1}$, and $h_{m+1,m}$

denote the smallest and largest eigenvalues of A, respectively. Define the constants

$$\kappa := \frac{\lambda_{\max}}{\lambda_{\min}}, \quad c := \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}, \quad \text{and} \quad \xi_m := \frac{1}{\cosh(m \ln c)} = \frac{2}{c^m + c^{-m}}. \tag{3.3}$$

If $\kappa = 1$, then set $\xi_m = 0$. Indeed, when A is HPD, FOM reduces to CG.

Theorem 3.1: The FOM error $\boldsymbol{e}_m := \boldsymbol{x}_* - \boldsymbol{x}_m^F$ at step m satisfies

$$\|\boldsymbol{e}_m\|_A = \min_{\boldsymbol{x} \in \mathscr{K}_m(A,\boldsymbol{b})} \|\boldsymbol{x}_* - \boldsymbol{x}\|_A \leq \xi_m \|\boldsymbol{e}_0\|_A \leq 2c^m \|\boldsymbol{e}_0\|_A .$$

Proof: See, e.g., [97, Ch. 8] and [122, Ch. 6]. □

Suppose now that A is positive real, and define

$$\rho := \min\left\{ \frac{\operatorname{Re}(\boldsymbol{v}^* A^{-1} \boldsymbol{v})}{\boldsymbol{v}^* \boldsymbol{v}} : \boldsymbol{v} \in \mathbb{C}^n, \boldsymbol{v} \neq 0 \right\};$$

$$\gamma := \min\left\{ \frac{\operatorname{Re}(\boldsymbol{v}^* A \boldsymbol{v})}{\boldsymbol{v}^* \boldsymbol{v}} : \boldsymbol{v} \in \mathbb{C}^n, \boldsymbol{v} \neq 0 \right\}; \text{ and}$$

$$\nu_{\max} := \max\left\{ \frac{(A\boldsymbol{v})^* A \boldsymbol{v}}{\boldsymbol{v}^* \boldsymbol{v}} : \boldsymbol{v} \in \mathbb{C}^n, \boldsymbol{v} \neq 0 \right\}.$$

Theorem 3.2: The GMRES residual $\boldsymbol{r}_m^G := A\boldsymbol{x}_m^G - \boldsymbol{b}$ at step m can be bounded as

$$\|\boldsymbol{r}_m\| \leq \left(1 - \frac{\gamma^2}{\nu_{\max}}\right)^{m/2} \|\boldsymbol{b}\|_2 . \tag{3.4}$$

Proof: See, e.g., [42, Theorem 3.3]. □

3.1 A comprehensive block Krylov subspace framework

We take $\mathbb{S} \subset \mathbb{C}^{s \times s}$ as the *-subalgebra with identity defined in Section 2.4. The key behind the framework is pairing $\mathbb{S}$ with a matrix-valued inner product and scaling quotient mapping elements from $\mathbb{C}^{n \times s}$ to $\mathbb{S}$. In this way, we regard $\mathbb{S}$ as a kind of generalized field, and the following notions as generalizations of an inner product

and norm. Matrix-valued norms and inner products have been considered before with similar purposes; see, e.g., the right bilinear form of [43], the $\diamond$ product of [18], the block inner product of [60], or the matrix-valued inner products used to define orthogonal polynomials in [25, 68]. The bulk of this section is taken from [55].

Definition 3.3: A mapping $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ from $\mathbb{C}^{n\times s} \times \mathbb{C}^{n\times s}$ to $\mathbb{S}$ is called a *block inner product onto* $\mathbb{S}$ if it satisfies the following conditions for all $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{C}^{n\times s}$ and $C \in \mathbb{S}$:

(i) $\mathbb{S}$-*linearity*: $\langle\!\langle \boldsymbol{X} + \boldsymbol{Y}, \boldsymbol{Z}C \rangle\!\rangle_{\mathbb{S}} = \langle\!\langle \boldsymbol{X}, \boldsymbol{Z} \rangle\!\rangle_{\mathbb{S}} C + \langle\!\langle \boldsymbol{Y}, \boldsymbol{Z} \rangle\!\rangle_{\mathbb{S}} C$;

(ii) *symmetry*: $\langle\!\langle \boldsymbol{X}, \boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}} = \langle\!\langle \boldsymbol{Y}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}^{*}$;

(iii) *definiteness*: $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}$ is positive definite if $\boldsymbol{X}$ has full rank, and $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}} = 0$ if and only if $\boldsymbol{X} = 0$.

Definition 3.4: A mapping N which maps all $\boldsymbol{X} \in \mathbb{C}^{n\times s}$ with full rank on a matrix $N(\boldsymbol{X}) \in \mathbb{S}$ is called a *scaling quotient* if for all such $\boldsymbol{X}$ there exists $\boldsymbol{Y} \in \mathbb{C}^{n\times s}$ such that $\boldsymbol{X} = \boldsymbol{Y}N(\boldsymbol{X})$ and $\langle\!\langle \boldsymbol{Y}, \boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}} = I$.

Remark 3.5: Let $\boldsymbol{X} \in \mathbb{C}^{n\times s}$. Some consequences of Definitions 3.3 and 3.3 include the following:

(i) Condition (ii) implies that $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}$ is always Hermitian. Then along with condition (iii), $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}$ is HPD when $\boldsymbol{X}$ has full rank, and HPSD otherwise.

(ii) By the previous comment, $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}$ is at least HPSD. Then the Cholesky factorization can induce a scaling quotient, provided the resulting factors belong to $\mathbb{S}$.

(iii) Let $f(z) = z^{1/2}$. By Definition 2.2, $f(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})$ can be expressed as $p(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})$ for some scalar polynomial p; consequently $f(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}) \in \mathbb{S}$. Then f can induce a scaling quotient as $\boldsymbol{X} = \boldsymbol{Y} f(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})$, since for $\boldsymbol{X}$ full rank, $\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}^{-1/2}$ is nonsingular and

$$\begin{aligned} \langle\!\langle \boldsymbol{Y}, \boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}} &= \langle\!\langle \boldsymbol{X} f(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})^{-1}, \boldsymbol{X} f(\langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})^{-1} \rangle\!\rangle_{\mathbb{S}} \\ &= \langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}^{-1/2} \langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}} \langle\!\langle \boldsymbol{X}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}^{-1/2} = I_s. \end{aligned}$$

(iv) Although the scaling quotient N is only formally defined for full-rank $\boldsymbol{X}$, it can be extended in practice to rank-deficient $\boldsymbol{X}$ via a deflation routine. For example, with a rank-revealing QR the linearly dependent columns of $\boldsymbol{X}$ can be removed so that a skinnier $\widehat{\boldsymbol{X}} \in \mathbb{C}^{n \times \widehat{s}}$ is considered instead, with N then being the analogous map from $\mathbb{C}^{n \times \widehat{s}} \to \mathbb{C}^{\widehat{s} \times \widehat{s}}$. See Remark 3.10 for details regarding deflation in implementation, and the thesis [15] for a thorough theoretical consideration of deflation in the classical inner product.

(v) If $\boldsymbol{X}$ has full rank, then $\langle\!\langle \boldsymbol{X} N(\boldsymbol{X})^{-1}, \boldsymbol{X} N(\boldsymbol{X})^{-1} \rangle\!\rangle_{\mathbb{S}} = I$.

Blocked-based notions of orthogonality and normalization also play important roles.

Definition 3.6: (i) $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n \times s}$ are *block orthogonal*, if $\langle\!\langle \boldsymbol{X}, \boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}} = 0$.

(ii) $\boldsymbol{X} \in \mathbb{C}^{n \times s}$ is *block normalized* if $N(\boldsymbol{X}) = I$.

(iii) $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m\} \subset \mathbb{C}^{n \times s}$ is *block orthonormal* if $\langle\!\langle \boldsymbol{X}_i, \boldsymbol{X}_j \rangle\!\rangle_{\mathbb{S}} = \delta_{ij} I$, where δ_{ij} is the Kronecker delta.

Remark 3.7: When we need to distinguish between inner products, we may write, e.g., "$\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-orthogonal" instead of "block orthogonal." The reader may find it helpful to read "$\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$" as "block" in such instances.

We say that a set of vectors $\{\boldsymbol{X}_j\}_{j=1}^m \subset \mathbb{C}^{n\times s}$ $\mathbb{S}$-*spans* a space $\mathscr{K} \subset \mathbb{C}^{n\times s}$ and write $\mathscr{K} = \text{span}^{\mathbb{S}}\{\boldsymbol{X}_j\}_{j=1}^m$, where

$$\text{span}^{\mathbb{S}}\{\boldsymbol{X}_j\}_{j=1}^m := \left\{ \sum_{j=1}^m \boldsymbol{X}_j \Gamma_j : \Gamma_j \in \mathbb{S} \text{ for all } j = 1, \ldots, m \right\}.$$

The set $\{\boldsymbol{X}_j\}_{j=1}^m$ constitutes a block orthonormal basis for $\mathscr{K}$, $\mathscr{K} = \text{span}^{\mathbb{S}}\{\boldsymbol{X}_j\}_{j=1}^m$, and $\{\boldsymbol{X}_j\}_{j=1}^m$ are orthonormal. Furthermore, the $\mathbb{S}$-span forms a proper subspace of $\mathbb{C}^{n\times s}$, so that the *mth block Krylov subspace for* A *and* $\boldsymbol{B}$ is well defined as

$$\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}) = \text{span}^{\mathbb{S}}\{\boldsymbol{B}, A\boldsymbol{B}, \ldots, A^{m-1}\boldsymbol{B}\}.$$

The $\mathbb{S}$-span notation is particularly important here. In some of the literature, authors confuse notation, defining the classical block Krylov subspace as $\mathscr{K}_m^{\texttt{Cl}}(A, B) = \text{span}\{\boldsymbol{B}, A\boldsymbol{B}, \ldots, A^{m-1}\boldsymbol{B}\}$. Since span denotes a linear combination, i.e., with scalars, this definition is usually not what authors intend; it would, however, be the correct definition for the global Krylov subspace. What is often intended is not span but rather colspan,

$$\mathscr{K}_m^{col}(A, \boldsymbol{B}) := \text{colspan}\{\boldsymbol{B}, A\boldsymbol{B}, \ldots, A^{m-1}\boldsymbol{B}\} := \mathscr{K}_m(A, \boldsymbol{b}_1) + \cdots + \mathscr{K}_m(A, \boldsymbol{b}_s).$$

The subspace $\mathscr{K}_m^{\texttt{Cl}}(A, B)$ is related to $\mathscr{K}_m^{col}(A, \boldsymbol{B})$ in that the columns of every element of $\mathscr{K}_m^{\texttt{Cl}}(A, B)$ are elements of $\mathscr{K}_m^{col}(A, \boldsymbol{B})$. But $\mathscr{K}_m^{col}(A, \boldsymbol{B}) \subset \mathbb{C}^n$; given the block-focused nature of our framework, we want to stick with a formulation of block Krylov subspaces that are subsets of $\mathbb{C}^{n\times s}$.

Table 3.1 summarizes combinations of $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, and N that lead to established or feasible block Krylov subspaces. The classical and global block products are first identified in [46], and the hybrid case is first considered in [55] as a combination

of the classical and loop-interchange methods. To ensure that the block-diagonal sparsity structure is retained, taking the matrix square root as the scaling quotient is recommended for the hybrid case; an economic QR is also possible, though.

The choice of $\mathbb{S}$ directly affects what the elements of $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ look like. In some sense, the structure and nature of $\mathbb{S}$ controls how much information is shared among the columns of the block vectors in $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$, as emphasized by the visualizations in the first column of Table 3.1. More precisely, consider the classical, loop-interchange, and global Krylov subspaces:

$$\mathscr{K}_m^{\mathtt{Cl}}(A, \boldsymbol{B}) = \left\{ \sum_{k=0}^{m-1} A^k \boldsymbol{B} C_k : C_k \in \mathbb{C}^{s\times s} \right\};$$

$$\mathscr{K}_m^{\mathtt{Li}}(A, \boldsymbol{B}) = \mathscr{K}_m(A, \boldsymbol{b}_1) \times \cdots \times \mathscr{K}_m(A, \boldsymbol{b}_s) = \left\{ \sum_{k=0}^{m-1} A^k \boldsymbol{B} D_k : D_k \in \mathbb{C}^{s\times s} \text{ is diagonal} \right\},$$

$$\text{where } \mathscr{K}_m(A, \boldsymbol{b}_i) := \operatorname{span}\{\boldsymbol{b}_i, A\boldsymbol{b}_i, \ldots, A^{m-1}\boldsymbol{b}_i\} \subset \mathbb{C}^n;$$

$$\mathscr{K}_m^{\mathtt{Gl}}(A, \boldsymbol{B}) = \operatorname{span}\{\boldsymbol{B}, A\boldsymbol{B}, \ldots, A^{m-1}\boldsymbol{B}\} = \left\{ \sum_{k=0}^{m-1} A^k \boldsymbol{B} c_k : c_k \in \mathbb{C} \right\}.$$

In fact, these different spaces are nested as

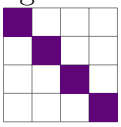
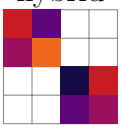
$$\mathscr{K}_m^{\mathtt{Gl}}(A, \boldsymbol{B}) \subset \mathscr{K}_m^{\mathtt{Li}}(A, \boldsymbol{B}) \subset \mathscr{K}_m^{\mathtt{Cl}}(A, \boldsymbol{B}).$$

Hybrid block Krylov subspaces would be nested between $\mathscr{K}_m^{\mathtt{Cl}}(A, \boldsymbol{B})$ and $\mathscr{K}_m^{\mathtt{Li}}(A, \boldsymbol{B})$. In this sense, then, the global approach allows for the least interaction between columns, and the classical approach the most.

Notions of self-adjointness and positive realness can also be represented in our block framework.

Definition 3.8: Let A be an operator mapping $\mathbb{C}^{n\times s}$ to $\mathbb{C}^{n\times s}$.

TABLE 3.1. Depictions and descriptions of block inner products used in numerical examples.

	$\mathbb{S}$	$\langle\!\langle \boldsymbol{X}, \boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}}$	$N(\boldsymbol{X})$
classical 	$\mathbb{C}^{s\times s}$	$\boldsymbol{X}^*\boldsymbol{Y}$	R, where $\boldsymbol{X} = \boldsymbol{Q}R$, and $\boldsymbol{Q} \in \mathbb{C}^{n\times s}$
global 	$\mathbb{C}I_s$	$\frac{1}{s}(\boldsymbol{X}^*\boldsymbol{Y})I_s$	$\frac{1}{s}\left\|\boldsymbol{X}\right\|_{\mathrm{F}} I_s$
loop-interchange 	$I_s \otimes \mathbb{C}$	$\operatorname{diag}(\boldsymbol{X}^*\boldsymbol{Y})$	$\operatorname{diag}(\left\|\boldsymbol{x}_1\right\|_2, \ldots, \left\|\boldsymbol{x}_s\right\|_2)$
hybrid 	$I_p \otimes \mathbb{C}^{q\times q}$, $s = qp$	$\operatorname{diag}(\boldsymbol{X}_1^*\boldsymbol{Y}_1, \ldots, \boldsymbol{X}_p^*\boldsymbol{Y}_p)$	$(\boldsymbol{X}^*\boldsymbol{X})^{1/2}$ or $N^{\mathtt{Cl}}(\boldsymbol{X})$

(i) A is *block self-adjoint* (BSA) if for all $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n\times s}$,

$$\langle\!\langle A\boldsymbol{X}, \boldsymbol{Y}\rangle\!\rangle_{\mathbb{S}} = \langle\!\langle \boldsymbol{X}, A\boldsymbol{Y}\rangle\!\rangle_{\mathbb{S}}.$$

(ii) A block self-adjoint A is *block positive definite* (BPD), if a) for all full rank $\boldsymbol{X} \in \mathbb{C}^{n\times s}$, the matrix $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}} \in \mathbb{C}^{s\times s}$ is self-adjoint and positive definite, and b) for all rank-deficient $\boldsymbol{X} \neq 0$, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ is self-adjoint, positive semi-definite, and nonzero.

(iii) A block self-adjoint A is *block positive semi-definite* (BPSD), if for $\boldsymbol{X} \neq 0$, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ is self-adjoint and positive semi-definite.

(iv) A is *block positive real* (BPR), if a) for all full rank $\boldsymbol{X} \in \mathbb{C}^{n\times s}$ the matrix $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}} \in \mathbb{C}^{s\times s}$ is EPR, and b) for all rank-deficient $\boldsymbol{X} \neq 0$, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ is ENNR and nonzero.

(v) A is *block nonnegative real* (BNNR), if for all $\boldsymbol{X} \neq 0$, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ is ENNR.

As in the non-block case, an operator may be self-adjoint with respect to one inner product, but not another, and likewise for other inner-product-based properties. We examine how block properties of an operator translate into more familiar scalar properties in Section 3.1.2.

3.1.1 The Block Arnoldi relation

To be useful in practice, we need a way to compute a block orthonormal basis of $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. Algorithm 3.1.1 is the generalization of the Arnoldi and Lanczos procedures within our framework. Issues of breakdowns and eigenvalue deflation are discussed for different $\langle\!\langle \cdot, \cdot\rangle\!\rangle_{\mathbb{S}}$ in, e.g., [10, 15, 16, 70, 84, 124, 134]; we do not go

into such details here. We assume that Algorithm 3.1.1 runs to completion without breaking down, i.e., that we obtain

(i) a block orthonormal basis $\{\boldsymbol{V}_k\}_{k=1}^{m+1} \subset \mathbb{C}^{n\times s}$, such that each $\boldsymbol{V}_k$ has full rank and $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}) = \operatorname{span}^{\mathbb{S}}\{\boldsymbol{V}_k\}_{k=1}^{m}$, and

(ii) a block upper Hessenberg matrix $\mathcal{H}_m \in \mathbb{S}^{m\times m}$ and $H_{m+1,m} \in \mathbb{S}$,

all satisfying the *block Arnoldi relation*

$$A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m\mathcal{H}_m + \boldsymbol{V}_{m+1}H_{m+1,m}\widehat{\boldsymbol{E}}_m^*, \tag{3.5}$$

where $\boldsymbol{\mathcal{V}}_m = [\boldsymbol{V}_1|\ldots|\boldsymbol{V}_m] \in \mathbb{C}^{n\times ms}$, and

$$\mathcal{H}_m = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,m} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,m} \\ & \ddots & \ddots & \vdots \\ & & H_{m,m-1} & H_{m,m} \end{bmatrix}.$$

An alternative form of equation (3.5) will at times be more convenient:

$$A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_{m+1}\underline{\mathcal{H}}_m, \tag{3.6}$$

where $\boldsymbol{\mathcal{V}}_{m+1} := [\boldsymbol{\mathcal{V}}_m|\boldsymbol{V}_{m+1}]$ and $\underline{\mathcal{H}}_m := \begin{bmatrix} \mathcal{H}_m \\ H_{m+1,m}\widehat{\boldsymbol{E}}_{m+1}^* \end{bmatrix}$.

A schematic of the block Arnoldi relation is given in Figure 3.1. Note that the block entries of $\mathcal{H}_m$ are elements of $\mathbb{S}$, so it is natural to say that $\mathcal{H}_m \in \mathbb{S}^{m\times m}$ and maps elements of $\mathbb{S}^m$, which is a proper subspace of $\mathbb{C}^{ms\times s}$, to $\mathbb{S}^m$.

Since the block upper Hessenberg matrix $\mathcal{H}_m$ is the restriction and projection of A onto $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$, it is insightful to consider the structure and sparsity of $\mathcal{H}_m$ in different paradigms, which reveal how much of A is "captured" by $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. In the classical case (top left of Figure 3.2), $\mathcal{H}_m$ is dense. The lower diagonal blocks are only triangular because the traditional $\boldsymbol{Q}R$ is used; if a rank-revealing $\boldsymbol{Q}R$ or

Algorithm 3.1.1: Block Arnoldi and Block Lanczos procedures

Given: A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, m

Compute $B = N(\boldsymbol{B})$ and $\boldsymbol{V}_1 = \boldsymbol{B}B^{-1}$

if *A is block self-adjoint* **then**

  Set $\boldsymbol{V}_0 = 0$, $H_{0,1} = B$

  for $k = 1, \ldots, m$ **do**

    $\boldsymbol{W} = A\boldsymbol{V}_k - \boldsymbol{V}_{k-1}H_{k-1,k}$

    $H_{k,k} = \langle\!\langle \boldsymbol{V}_k, \boldsymbol{W} \rangle\!\rangle_{\mathbb{S}}$

    $\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{V}_k H_{k,k}$

    Compute $H_{k+1,k} = N(\boldsymbol{W})$ and $\boldsymbol{V}_{k+1} = \boldsymbol{W}H_{k+1,k}^{-1}$

    Set $H_{k,k+1} = H_{k+1,k}^{*}$

else

  for $k = 1, \ldots, m$ **do**

    Compute $\boldsymbol{W} = A\boldsymbol{V}_k$

    for $j = 1, \ldots, k$ **do**

      $H_{j,k} = \langle\!\langle \boldsymbol{V}_j, \boldsymbol{W} \rangle\!\rangle_{\mathbb{S}}$

      $\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{V}_j H_{j,k}$

    Compute $H_{k+1,k} = N(\boldsymbol{W})$ and $\boldsymbol{V}_{k+1} = \boldsymbol{W}H_{k+1,k}^{-1}$

Return B, $\boldsymbol{\mathcal{V}}_m = [\boldsymbol{V}_1 | \ldots | \boldsymbol{V}_m]$, $\mathcal{H}_m = (H_{j,k})_{j,k=1}^{m}$, $\boldsymbol{V}_{m+1}$, and $H_{m+1,m}$

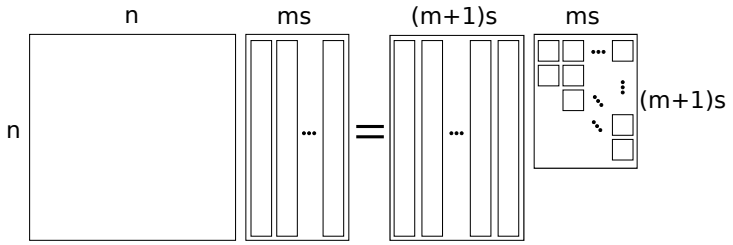


FIGURE 3.1. Illustration of the block Arnoldi relation.

matrix square root were used as the scaling quotient, they would have a different structure. In terms of density, the hybrid case (bottom) is next in line, with a sparsity between that of the classical and global or loop-interchange cases. In fact, the global case produces an $\mathcal{H}_m$ with a special Kronecker structure: $\mathcal{H}_m = H_m \otimes I_s$, where $H_m \in \mathbb{C}^{m\times m}$ and is upper Hessenberg.

Some properties will prove helpful for for understanding how the orthonormality of the Krylov basis vectors works with respect to the block inner product. The proofs are a straightforward exercise, using Definition 3.3 and properties of the Krylov basis $\boldsymbol{\mathcal{V}}_m$.

Lemma 3.9: Let $\boldsymbol{Y}, \boldsymbol{W} \in \mathbb{S}^{m+1} \subseteq \mathbb{C}^{(m+1)s\times s}$, with $\mathbb{S}$ entries denoted as Y_j and W_j, respectively, for $j = 1, \ldots, m+1$. Then

(i) $\langle\!\langle \boldsymbol{V}_j, \boldsymbol{\mathcal{V}}_{m+1}\boldsymbol{Y} \rangle\!\rangle_{\mathbb{S}} = Y_j$

(ii) $\langle\!\langle \boldsymbol{\mathcal{V}}_{m+1}\boldsymbol{Y}, \boldsymbol{V}_j \rangle\!\rangle_{\mathbb{S}} = Y_j^*$

(iii) $\langle\!\langle \boldsymbol{\mathcal{V}}_{m+1}\boldsymbol{Y}, \boldsymbol{\mathcal{V}}_{m+1}\boldsymbol{W} \rangle\!\rangle_{\mathbb{S}} = \boldsymbol{Y}^*\boldsymbol{W}$

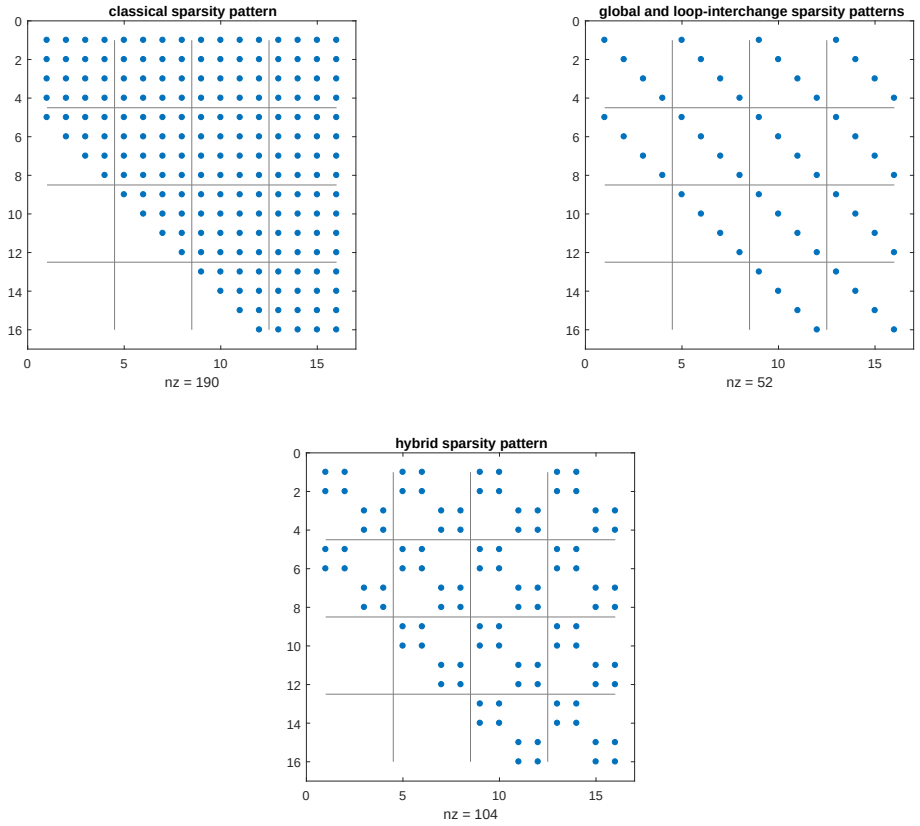


FIGURE 3.2. Sparsity patterns of $\mathcal{H}_4$ for different block inner products and $s = 4$, with $q = 2$ for the hybrid example.

Remark 3.10: When implementing Algorithm 3.1.1, it is important to account for linear dependence among columns of the basis vectors. The classical inner product is most problematic with linear dependence; see techniques for BCG described in [7, 15, 16, 36, 108] and for BGMRES in [69].

We make a particular choice for how to deal with linear dependence in the classical and hybrid versions of Algorithm 3.1.1. There are methods for short-term recurrences [36, 108], which are not appropriate for our purposes, since we focus

on FOM-like and GMRES-like methods and need the full Krylov basis for matrix functions in Chapter 5. Other methods require breaking up the columns of the block vectors and using matrix-vector multiplication [7], which is also not preferable, since the resulting algorithm would not take advantage of Level 3 sparse BLAS. Algorithm 7.3 of [15] is the only viable column deflation routine that both features matrix-matrix operations and retains the entire Krylov basis. We employ this routine when using Algorithm 3.1.1 for matrix functions.

3.1.2 Preserving properties of A in $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$

The motivation behind Krylov methods is to reduce A to a small matrix with similar properties that is computationally cheaper to work with. In this section, we discuss how properties of A are transferred to $\mathcal{H}_m$ via the block framework, but to do so, we introduce a scalar inner product and norm induced by $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, as well as block analogues of a number of linear algebra notions.

By taking the trace, we can convert $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ into a scalar inner product $\langle \cdot, \cdot \rangle_{\mathbb{S}} : \mathbb{C}^{n\times s} \times \mathbb{C}^{n\times s} \to \mathbb{C}$:

$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_{\mathbb{S}} := \operatorname{trace}\left(\langle\!\langle \boldsymbol{Y}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}} \right). \tag{3.7}$$

Properties of $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ guarantee that (3.7) is a true inner product on $\mathbb{C}^{n\times s}$. Naturally, it induces the norm

$$\|\boldsymbol{X}\|_{\mathbb{S}} := \langle \boldsymbol{X}, \boldsymbol{X} \rangle_{\mathbb{S}}^{\frac{1}{2}}.$$

Note that the classical, global, loop-interchange, and hybrid paradigms all reduce to the Frobenius norm. Weighted versions of $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ and $\|\cdot\|_{\mathbb{S}}$ can be easily defined for operators $W : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$ which are self-adjoint and positive definite with respect

to $\langle \cdot, \cdot \rangle_{\mathbb{S}}$:

$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_{W\text{-}\mathbb{S}} := \langle \boldsymbol{X}, W\boldsymbol{Y} \rangle_{\mathbb{S}}; \text{ and}$$

$$\|\boldsymbol{X}\|_{W\text{-}\mathbb{S}} := \langle \boldsymbol{X}, \boldsymbol{X} \rangle_{W\text{-}\mathbb{S}}^{\frac{1}{2}}.$$

The following is a combination and generalization of [55, Lemmas 3.5 and 3.6].

Lemma 3.11: Let p be a scalar-valued polynomial with real coefficients and let $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$ be self-adjoint with respect to $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$. Then

$$\|p(A)\|_{W\text{-}\mathbb{S}} = \max_{\lambda \in \text{spec}(A)} |p(\lambda)| .$$

Proof: Since A is self-adjoint with respect to $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$ on $\mathbb{C}^{n\times s}$, it admits an $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$-orthonormal basis of eigenvectors from $\mathbb{C}^{n\times s}$. Then the operator norm $\|A\|_{W\text{-}\mathbb{S}}$ is given as

$$\|A\|_{W\text{-}\mathbb{S}} = \max_{\lambda \in \text{spec}(A)} |\lambda| .$$

Since p has real coefficients, $p(A)$ is also $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$-self-adjoint, and $\text{spec}(p(A)) = \{p(\lambda) : \lambda \in \text{spec}(A)\}$, where the spectrum of A is precisely its eigenvalues. As a direct consequence, we have that $\|p(A)\|_{W\text{-}\mathbb{S}} = \max_{\lambda \in \text{spec}(A)} |p(\lambda)|$. □

The scalar inner product $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ induces a traditional, scalar notion of orthogonality, and similarly for weighted versions. Trivially, block orthogonality implies scalar orthogonality, since $\langle\!\langle \boldsymbol{Y}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}} = 0_s$ implies $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_{\mathbb{S}} = \text{trace}\left(\langle\!\langle \boldsymbol{Y}, \boldsymbol{X} \rangle\!\rangle_{\mathbb{S}}\right) = 0$, and likewise for weighted cases.

Another useful result bounds a weighted norm from above and below; the original statement and proof are presented as [55, Lemma 4.4].

Lemma 3.12: Let A be $\langle\cdot,\cdot\rangle_{\mathbb{S}}$-positive-definite, and let $g : (0,\infty) \to (0,\infty)$ be a scalar rational function. Also, let $g_{\min}$ and $g_{\max}$ denote the minimum and maximum values of g on $\mathrm{spec}(A)$, respectively. Then

$$\sqrt{g_{\min}}\,\|\boldsymbol{V}\|_{\mathbb{S}} \leq \|\boldsymbol{V}\|_{g(A)\text{-}\mathbb{S}} \leq \sqrt{g_{\max}}\,\|\boldsymbol{V}\|_{\mathbb{S}}\,.$$

Proof: Since A is $\langle\cdot,\cdot\rangle_{\mathbb{S}}$-positive definite, its spectrum is positive and A has a $\langle\cdot,\cdot\rangle_{\mathbb{S}}$-orthonormal eigenbasis, i.e., there exist $\{\beta_j\}_{j=1}^{ns} \subset \mathbb{C}$ and $\{\boldsymbol{Q}_j\}_{j=1}^{ns} \subset \mathbb{C}^{n\times s}$ such that $A\boldsymbol{Q}_j = \lambda_j \boldsymbol{Q}_j$ and $\langle \boldsymbol{Q}_j, \boldsymbol{Q}_k\rangle_{\mathbb{S}} = \delta_{jk}$. Given any $\boldsymbol{V} \in \mathbb{C}^{n\times s}$, we expand it in terms of this basis as $\boldsymbol{V} = \sum_{j=1}^{ns} \beta_j \boldsymbol{Q}_j$. Then

$$\|\boldsymbol{V}\|^2_{g(A)\text{-}\mathbb{S}} = \langle g(A)\boldsymbol{V}, \boldsymbol{V}\rangle_{\mathbb{S}} = \langle \sum_{j=1}^{ns} g(\lambda_j)\beta_j \boldsymbol{Q}_j, \sum_{j=1}^{ns} \beta_j \boldsymbol{Q}_j\rangle_{\mathbb{S}} = \sum_{j=1}^{ns} g(\lambda_j)|\beta_j|^2,$$

and thus

$$g_{\min} \sum_{j=1}^{ns} |\beta_j|^2 \leq \|\boldsymbol{V}\|^2_{g(A)\text{-}\mathbb{S}} \leq g_{\max} \sum_{j=1}^{ns} |\beta_j|^2.$$

Noting that $\sum_{j=1}^{ns} |\beta_j|^2 = \|\boldsymbol{V}\|^2_{\mathbb{S}}$ leads to the desired result. □

Recall that the spectrum of an operator is defined independently of the inner product. This independence is crucial, since we need to take advantage of an unusual inner product $\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}$ defined as

$$\langle \boldsymbol{X}, \boldsymbol{Y}\rangle_{\boldsymbol{\mathcal{V}}_m} := \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, \boldsymbol{\mathcal{V}}_m \boldsymbol{Y}\rangle_{\mathbb{S}}, \;\; \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{S}^m,$$

where $\boldsymbol{\mathcal{V}}_m$ is the matrix of the block Arnoldi vectors as in the relation (3.5). It follows that the field of values of $\mathcal{H}_m$ with respect to $\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}$ is contained in the field of values of A with respect to $\langle\cdot,\cdot\rangle_{\mathbb{S}}$; see Definition 2.5 for the definition of the field of values with respect to a given inner product.

Lemma 3.13 (Lemma 4.1 of [55]): It holds that

$$\mathbb{F}_{\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}}(\mathcal{H}_m) \subset \mathbb{F}_{\langle\cdot,\cdot\rangle_{\mathbb{S}}}(A).$$

Proof: By the Arnoldi relation (3.5) it holds that

$$\begin{aligned}\langle \boldsymbol{X}, \mathcal{H}_m \boldsymbol{X}\rangle_{\boldsymbol{\mathcal{V}}_m} &= \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{X}\rangle_{\mathbb{S}} \\ &= \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, A\boldsymbol{\mathcal{V}}_m \boldsymbol{X}\rangle_{\mathbb{S}} - \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{X}\rangle_{\mathbb{S}} \\ &= \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, A\boldsymbol{\mathcal{V}}_m \boldsymbol{X}\rangle_{\mathbb{S}}.\end{aligned}$$

The last equality holds since $\langle\!\langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{X}\rangle\!\rangle_{\mathbb{S}} = 0$, which can be seen by breaking $\boldsymbol{\mathcal{V}}_m \boldsymbol{X}$ into components and applying Lemma 3.9. We conclude the proof by noting that $\boldsymbol{\mathcal{V}}_m \boldsymbol{X} \in \mathbb{C}^{n\times s}$ and $\langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, \boldsymbol{\mathcal{V}}_m \boldsymbol{X}\rangle_{\mathbb{S}} = \langle \boldsymbol{X}, \boldsymbol{X}\rangle_{\boldsymbol{\mathcal{V}}_m}$. □

The block properties from Definition 3.8 carry over to their scalar analogues.

Lemma 3.14: Let A be an operator mapping $\mathbb{C}^{n\times s}$ to $\mathbb{C}^{n\times s}$.

(i) If A is BSA, then A is self-adjoint with respect to $\langle\cdot,\cdot\rangle_{\mathbb{S}}$.

(ii) If A is BSA and BP(S)D, then A is self-adjoint and positive (semi-)definite with respect to $\langle\cdot,\cdot\rangle_{\mathbb{S}}$.

(iii) If A is BPR (BNNR), then A is positive real (nonnegative real) with respect to $\langle\cdot,\cdot\rangle_{\mathbb{S}}$.

Proof: Part (i) follows by straightforward application of the definition of $\langle\cdot,\cdot\rangle_{\mathbb{S}}$. For part (ii), we note that whenever $\boldsymbol{X} \neq 0$ and A is BPSD, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ is positive semi-definite and therefore has nonnegative eigenvalues, so $\operatorname{trace}(\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}) \geq 0$. When A is BPD, $\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}$ has at least one positive eigenvalue, so $\operatorname{trace}(\langle\!\langle \boldsymbol{X}, A\boldsymbol{X}\rangle\!\rangle_{\mathbb{S}}) > 0$.

An analogous argument holds for part (iii) by looking instead at the real part of trace $(\langle\!\langle \boldsymbol{X}, A\boldsymbol{X} \rangle\!\rangle_{\mathbb{S}})$. $\square$

Lemma 3.15: If A possesses one of the block properties of Definition 3.8 with respect to $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, then $\mathcal{H}_m$ possess the same property with respect to $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\boldsymbol{\mathcal{V}}_m}$. Furthermore, $\mathcal{H}_m$ possesses the scalar version of the property with respect to $\langle \cdot, \cdot \rangle_{\boldsymbol{\mathcal{V}}_m}$.

Proof: Since Algorithm 3.1.1 switches to the block Lanczos procedure when A is BSA, $\mathcal{H}_m$ inherits the same property by construction. Suppose now that A is BSA and BP(S)D. By the proof of Lemma 3.13, $\langle \boldsymbol{X}, \mathcal{H}_m \boldsymbol{X} \rangle_{\boldsymbol{\mathcal{V}}_m} = \langle \boldsymbol{\mathcal{V}}_m \boldsymbol{X}, A\boldsymbol{\mathcal{V}}_m \boldsymbol{X} \rangle_{\mathbb{S}}$. Then by Definition 3.8, $\mathcal{H}_m$ is also BSA and BP(S)D. Likewise when A is BPR or BNNR. Lemma 3.14 applied to $\mathcal{H}_m$ and $\langle \cdot, \cdot \rangle_{\boldsymbol{\mathcal{V}}_m}$ concludes the proof. $\square$

3.1.3 Block orthogonal projectors

Let $\boldsymbol{\mathcal{P}}$ denote the $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$-orthogonal projector onto a subspace $\mathscr{K}$ of $\mathbb{C}^{n \times s}$. The following results are generalizations of [122, Theorem 1.38 and Corollary 1.39].

Theorem 3.16: Given $\boldsymbol{Y} \in \mathbb{C}^{n \times s}$, $\mathscr{K} \subset \mathbb{C}^{n \times s}$, and a $\langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}$-orthogonal $\boldsymbol{\mathcal{P}}$,

$$\|\boldsymbol{Y} - \boldsymbol{\mathcal{P}}\boldsymbol{Y}\|_{W\text{-}\mathbb{S}} = \min_{\boldsymbol{X} \in \mathscr{K}} \|\boldsymbol{Y} - \boldsymbol{X}\|_{W\text{-}\mathbb{S}}\,.$$

Corollary 3.17: Let $\boldsymbol{Y} \in \mathbb{C}^{n \times s}$ and $\mathscr{K} \subset \mathbb{C}^{n \times s}$ be given. Then $\boldsymbol{Z} \in \mathbb{C}^{n \times s}$ satisfies

$$\|\boldsymbol{Y} - \boldsymbol{Z}\|_{W\text{-}\mathbb{S}} = \min_{\boldsymbol{X} \in \mathscr{K}} \|\boldsymbol{Y} - \boldsymbol{X}\|_{W\text{-}\mathbb{S}}$$

if and only if

$$\boldsymbol{Z} \in \mathscr{K} \text{ and } \boldsymbol{Y} - \boldsymbol{Z} \text{ is } \langle \cdot, \cdot \rangle_{W\text{-}\mathbb{S}}\text{-orthogonal to } \mathscr{K}\,.$$

3.1.4 Cospatiality vs. collinearity

The notion of collinearity is well established for vectors. Two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$ are *collinear* if there exists a nonzero $c \in \mathbb{C}$ such that $\boldsymbol{x} = c\boldsymbol{y}$. This notion also makes sense for block vectors, if one regards the space $\mathbb{C}^{n\times s}$ as a vector space, i.e., $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n\times s}$ are collinear if there exists a $c \in \mathbb{C}$ such that $\boldsymbol{X} = c\boldsymbol{Y}$.

A more general notion proves useful in our analysis, that of cospatiality, which is first coined in [55]. We say that two block vectors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n\times s}$ are *cospatial* if there exists a $C \in \mathbb{S}$ such that $\boldsymbol{X} = \boldsymbol{Y}C$. Another way to regard cospatiality is to think of the columns of $\boldsymbol{X}$ and $\boldsymbol{Y}$ as spanning the same subspace in a way specified by the zero-nonzero structure of elements of $\mathbb{S}$.

3.1.5 Characterizations of block Krylov subspaces

For reference throughout the rest of this chapter, it is helpful to summarize the various characterizations of block Krylov subspaces up to this point:

$$\begin{aligned}
\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}) &= \operatorname{span}^{\mathbb{S}}\{\boldsymbol{B}, A\boldsymbol{B}, \ldots, A^{m-1}\boldsymbol{B}\} \\
&= \left\{ \sum_{k=1}^{m} A^{k-1}\boldsymbol{B}C_k : C_k \in \mathbb{S} \right\} \\
&= \{P(A) \circ \boldsymbol{B} : P \in \mathbb{P}_{m-1}(\mathbb{S})\} \\
&= \{\boldsymbol{\mathcal{V}}_m \boldsymbol{Y} : \boldsymbol{Y} \in \mathbb{S}^m\},
\end{aligned}$$

where P is a matrix polynomial as defined in Section 2.5, $\mathbb{S}^m$ is the subspace of $\mathbb{C}^{ms\times s}$ whose elements take their block entries from $\mathbb{S}$, and $\boldsymbol{\mathcal{V}}_m$ is the matrix of basis vectors from the block Arnoldi relation (3.5).

3.2 Block FOM

The essential Krylov method is the Full Orthogonalization Method (FOM) [120]. While not the most popular method for solving linear systems, it serves as an important starting point for Krylov methods for matrix functions [34, 35, 39, 52, 53, 54, 78, 81, 90, 121]. This section is largely taken from [55] and treats block FOM (BFOM). Throughout we assume that the initial approximation to the system is $\boldsymbol{X}_0 = 0$.

We continue in the vein of the previous section, with quantities from Section 3.1.1. Note that $\boldsymbol{B} \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$, and $\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \widehat{\boldsymbol{E}}_1 B = \boldsymbol{V}_1 B$, where $B = N(\boldsymbol{B})$. We define the mth BFOM approximation to the linear system $A\boldsymbol{X} = \boldsymbol{B}$ as $\boldsymbol{X}_m \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ satisfying the *block Galerkin condition*

$$\boldsymbol{R}_m := \boldsymbol{B} - A\boldsymbol{X}_m \perp_{\langle\!\langle \cdot,\cdot \rangle\!\rangle_{\mathbb{S}}} \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}). \tag{3.8}$$

Theorem 3.18 (Theorem 3.1 in [55]): Assume that $\mathcal{H}_m : \mathbb{S}^m \to \mathbb{S}^m$ is nonsingular and let $\boldsymbol{Y}_m = \mathcal{H}_m^{-1} \widehat{\boldsymbol{E}}_1 B$. Then $\boldsymbol{X}_m := \boldsymbol{\mathcal{V}}_m \boldsymbol{Y}_m$ belongs to $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ and satisfies the block Galerkin condition (3.8).

Proof: Since $\boldsymbol{Y}_m \in \mathbb{S}^m$, it follows that $\boldsymbol{X}_m \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ and

$$\begin{aligned}
\boldsymbol{R}_m = \boldsymbol{B} - A\boldsymbol{X}_m &= \boldsymbol{\mathcal{V}}_m \widehat{\boldsymbol{E}}_1 B - A\boldsymbol{\mathcal{V}}_m \boldsymbol{Y}_m \\
&= \boldsymbol{\mathcal{V}}_m \widehat{\boldsymbol{E}}_1 B - (\boldsymbol{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^*) \boldsymbol{Y}_m \\
&= -\boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \\
&= \boldsymbol{V}_{m+1} C_m \text{ with } C_m := -H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \in \mathbb{S}.
\end{aligned} \tag{3.9}$$

Since $\boldsymbol{V}_{m+1}$ is block orthogonal to $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ by construction, so is $\boldsymbol{V}_{m+1} C_m$, implying that $\boldsymbol{X}_m$ satisfies condition (3.8). $\square$

Definition 3.19: Two block vectors $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{C}^{n\times s}$ are $\mathbb{S}$-*cospatial* if there exists $C \in \mathbb{S}$ such that

$$\boldsymbol{X} = \boldsymbol{Y}C. \tag{3.10}$$

The proof of Theorem 3.18 demonstrates that $\boldsymbol{R}_m$ and $\boldsymbol{V}_{m+1}$ are $\mathbb{S}$-cospatial; in particular, they $\mathbb{S}$-span the same subspace. We often drop the prefix $\mathbb{S}$ when it is clear from context.

3.2.1 Error bounds

It is possible to interpret $\boldsymbol{X}_m$ and $\boldsymbol{R}_m$ from the point of view of matrix polynomials. Since $\boldsymbol{X}_m \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$, there exists a matrix polynomial $Q_{m-1} \in \mathbb{P}_{m-1}(\mathbb{S})$ such that $\boldsymbol{X}_m = Q_{m-1}(A) \circ \boldsymbol{B}$, and another matrix polynomial $P_m(z) := I - zQ_{m-1}(z)$ satisfying $\boldsymbol{R}_m = P_m(A) \circ \boldsymbol{B}$. Denoting the space of mth degree matrix polynomials P with $P(0) = I$ as $\bar{\mathbb{P}}_m(\mathbb{C})$, it is clear that $P_m \in \bar{\mathbb{P}}_m(\mathbb{S})$. We can derive some useful results on the error of BFOM, for matrices A that are BSA and BPD, in the spirit of the conjugate gradient results of Theorem 3.1. We denote the BFOM error for the mth approximation $\boldsymbol{X}_m = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m^{-1} \widehat{\boldsymbol{E}}_1 B$ as

$$\boldsymbol{E}_m := \boldsymbol{X}_* - \boldsymbol{X}_m,$$

where $\boldsymbol{X}_*$ is the exact solution to $A\boldsymbol{X} = \boldsymbol{B}$. The following theorem is originally [55, Theorem 3.7].

Theorem 3.20: Let $A \in \mathbb{C}^{n\times n}$ be BSA and BPD with respect to $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, and $\boldsymbol{B} \in \mathbb{C}^{n\times s}$ be a block right-hand-side vector. Then the BFOM error $\boldsymbol{E}_m$ satisfies

$$\|\boldsymbol{E}_m\|_{A\text{-}\mathbb{S}} = \min_{\boldsymbol{X}\in\mathscr{K}_m^{\mathbb{S}}(A,\boldsymbol{B})} \|\boldsymbol{X}_* - \boldsymbol{X}\|_{A\text{-}\mathbb{S}} \leq \xi_m \|\boldsymbol{B}\|_{A\text{-}\mathbb{S}}, \tag{3.11}$$

with ξ_m from (3.3).

Proof: By the block Galerkin condition (3.8), $\boldsymbol{R}_m$ is $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-orthogonal and consequently $\langle \cdot, \cdot \rangle_{\mathbb{S}}$-orthogonal to $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. Then for all $\boldsymbol{V} \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$,

$$0 = \langle \boldsymbol{R}_m, \boldsymbol{V} \rangle_{\mathbb{S}} = \langle A\boldsymbol{E}_m, \boldsymbol{V} \rangle_{\mathbb{S}} = \langle \boldsymbol{E}_m, \boldsymbol{V} \rangle_{A\text{-}\mathbb{S}}.$$

Applying Corollary 3.17 then gives the equality in (3.11).

To prove the inequality in (3.11), recall that $\boldsymbol{R}_m = P_m \circ \boldsymbol{B}$, for some matrix polynomial $P_m \in \mathbb{P}_m(\mathbb{S})$. Since A^{-1} commutes with A,

$$\begin{aligned} \boldsymbol{E}_m &= A^{-1}\boldsymbol{R}_m = A^{-1}P_m(A) \circ \boldsymbol{B} \\ &= P_m(A) \circ A^{-1}\boldsymbol{B} = P_m(A) \circ \boldsymbol{E}_0. \end{aligned}$$

It follows that

$$\|P_m(A) \circ \boldsymbol{E}_0\|_{A\text{-}\mathbb{S}} = \min_{P \in \bar{\mathbb{P}}_m(\mathbb{S})} \|P(A) \circ \boldsymbol{E}_0\|_{A\text{-}\mathbb{S}}\,. \tag{3.12}$$

By embedding scalar polynomials $\rho(\lambda) = 1 + \sum_{i=1}^m \gamma_i \lambda^i$ in $P_\rho(\lambda) = I_s + \sum_{i=1}^m (\gamma_i I_s)\lambda^i$, $\bar{\mathbb{P}}_m(\mathbb{C})$ can be regarded as a subspace of $\bar{\mathbb{P}}_m(\mathbb{S})$, with $P_\rho(A) \circ \boldsymbol{X} = \rho(A)\boldsymbol{X}$. Along with equation (3.12) and Lemma 3.11, this gives that

$$\|P_m(A) \circ \boldsymbol{E}_0\|_{A\text{-}\mathbb{S}} \leq \|\rho(A)\boldsymbol{E}_0\|_{A\text{-}\mathbb{S}} \leq \max_{\lambda \in \mathrm{spec}(A)} |\rho(\lambda)| \cdot \|\boldsymbol{E}_0\|_{A\text{-}\mathbb{S}} \text{ for any } \rho \in \bar{\mathbb{P}}_m(\mathbb{C}).$$

Taking ρ as the (scaled) Chebyshev polynomial of degree m for the interval $[\lambda_{\min}, \lambda_{\max}]$ (as in, e.g., [122, Chapter 6]), then $\max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |\rho(\lambda)| \leq \xi_m$. □

Remark 3.21: Theorem 3.20 is not a novel result for the classical, global, and loop-interchange block inner products. In the classical and loop interchange cases, $\|\boldsymbol{X}\|_{A\text{-}\mathbb{S}} = \|\boldsymbol{X}\|_{A\text{-F}} = \sqrt{\mathrm{trace}\,(\boldsymbol{X}^* A \boldsymbol{X})}$, and in the global case, $\|\boldsymbol{X}\|_{A\text{-}\mathbb{S}} = s\,\|\boldsymbol{X}\|_{A\text{-F}}$.

Consequently, Theorem 3.20 reduces in all three cases to

$$\|\boldsymbol{E}_m\|_{A\text{-F}} \leq \xi_m \|\boldsymbol{E}_0\|_{A\text{-F}}. \tag{3.13}$$

For the classical case, this result is contained in unpublished work by Eisenstat [41], who rewrites results from [108] in terms of the A-weighted Frobenius norm. In the loop interchange case, we can use the standard CG error bound from Theorem 3.1 for each column as an alternative way to arrive at inequality (3.13). In the global case, the estimate (3.13) can also be obtained as follows. Solving the block linear system $A\boldsymbol{X} = \boldsymbol{B}$ with global BFOM is identical to solving $(I_s \otimes A)\,\mathrm{vec}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{B})$ with FOM [73, Theorem 1], where vec is the operator that reshapes an $n \times s$ block vector into an $ns \times 1$ vector by stacking the columns. Since $I_s \otimes A$ and A have identical spectra, κ, c, and ξ_m are just as in (3.3). Applying Theorem 3.1 we obtain that

$$\|\mathrm{vec}(\boldsymbol{E}_m)\|_{I_s\otimes A} \leq \xi_m \|\mathrm{vec}(\boldsymbol{E}_0)\|_{I_s\otimes A}.$$

Converting everything back to block form gives inequality (3.13).

The power of Theorem 3.20 is the generality of the result. It holds for all scalar inner products and weighted block inner products that satisfy Definition 3.3. It also allows one to see how the error associated to different inner products relate to each other.

Theorem 3.22: Let $\boldsymbol{E}_m^{\square}$ denote the error for the mth BFOM approximation $\boldsymbol{X}_m^{\square}$, where $\square$ denotes the choice of one of the paradigms from Table 3.1, i.e., $\square$ is a placeholder for `Cl` (classical), `Gl` (global), `Li` (loop-interchange), or `Hy` (hybrid). Then

$$\left\|\boldsymbol{E}_m^{\texttt{Cl}}\right\|_{A\text{-F}} \leq \left\|\boldsymbol{E}_m^{\texttt{Hy}}\right\|_{A\text{-F}} \leq \left\|\boldsymbol{E}_m^{\texttt{Li}}\right\|_{A\text{-F}} \leq \left\|\boldsymbol{E}_m^{\texttt{Gl}}\right\|_{A\text{-F}}. \tag{3.14}$$

Proof: The result follows immediately upon noting that $\mathbb{S}^{\texttt{Gl}} \subseteq \mathbb{S}^{\texttt{Li}} \subseteq \mathbb{S}^{\texttt{Hy}} \subseteq \mathbb{S}^{\texttt{Cl}}$. $\square$

3.2.2 Shifted BFOM with restarts: Sh-BFOM(m)

Consider again the family of shifted linear systems

$$(A + tI)\boldsymbol{X}(t) = \boldsymbol{B}, \qquad \text{(3.1 revisited)}$$

where the admissible shifts t are bounded away from the spectrum of $-A$. We now develop a shifted BFOM with restarts. The main use of this method in our work is for matrix functions with Cauchy-Stieltjes integral expressions (see Chapter 5). Shifted BFOM with restarts is obviously also useful in and of itself for solving equation (3.1); see, e.g., [141], where a modified version of shifted BFOM with restarts is used on rational approximations to the matrix exponential.

Restarting is often paired with Krylov subspace methods to mitigate storage limitations, i.e., the number of basis vectors that can be stored on one's machine at a time. Short-term recurrences are another common technique, but since we need the entire Krylov basis for the matrix function approximations considered in Chapter 5, we do not consider short-term recurrences here. Throughout this section, we assume the number of Krylov basis vectors m, also referred to as the restart cycle length, is fixed a priori. It is also possible to implement restarts where the number of basis vectors changes per cycle, but since m usually reflects a hardware memory limitation, it is reasonable to assume it is fixed. We use a superscript in parentheses (k) to denote the cycle index, with $k = 0, 1, 2$, and so forth.

Cospatiality among the shifted and restarted residuals is essential for our convergence analysis and for an efficient algorithm. To that end, we prescribe

$\boldsymbol{X}_m^{(0)}(t) := 0$ and define the restarted BFOM approximation to $\boldsymbol{X}_*(t)$ obtained after the $k+1$st cycle as

$$\boldsymbol{X}_m^{(k+1)}(t) := \boldsymbol{X}_m^{(k)}(t) + \boldsymbol{Z}_m^{(k)}(t), \quad k = 0, 1, \ldots$$

with $\boldsymbol{Z}_m^{(k)}(t)$ defined as the BFOM approximation to $\boldsymbol{Z}_*^{(k)}(t)$ in the block residual equation

$$(A + tI)\boldsymbol{Z}_*^{(k)}(t) = \boldsymbol{R}_m^{(k)}(t), \text{ with } \boldsymbol{R}_m^{(k)}(t) := \boldsymbol{B} - (A + tI)\boldsymbol{X}_m^{(k)}(t).$$

An explicit form for $\boldsymbol{Z}_m^{(k)}(t)$ is determined by the following discussion.

We start with the first cycle, which, by the shift invariance of the Arnoldi relation (3.5) and the definition of the non-shifted BFOM approximation in Theorem 3.18, is described by the quantities

$$\boldsymbol{Y}_m^{(1)}(t) := \left(\mathcal{H}_m^{(1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)};$$

$$\boldsymbol{X}_m^{(1)}(t) := \boldsymbol{\mathcal{V}}_m^{(1)} \boldsymbol{Y}_m(t); \text{ and}$$

$$\boldsymbol{R}_m^{(1)}(t) := \boldsymbol{V}_{m+1}^{(1)} C_m^{(1)}(t), \text{ with } C_m^{(1)}(t) := -H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(1)}(t).$$

Since $\boldsymbol{R}_m^{(1)}(t)$ is cospatial to $\boldsymbol{V}_{m+1}^{(1)}$ for all shifts, we can begin computing the second Krylov basis with $\boldsymbol{V}_{m+1}^{(1)}$. We then obtain the block basis $\{\boldsymbol{V}_1^{(2)} = \boldsymbol{V}_{m+1}^{(1)}, \ldots, \boldsymbol{V}_{m+1}^{(2)}\}$, which $\mathbb{S}$-spans $\mathscr{K}_m^{\mathbb{S}}\left(A, \boldsymbol{V}_1^{(2)}\right)$; note that $B^{(2)} = I_s$ since $\boldsymbol{V}_{m+1}^{(1)}$ is already block orthonormal. The error of the first cycle $\boldsymbol{Z}_m^{(1)}(t)$ can then be approximated as

$$\boldsymbol{Z}_m^{(1)}(t) := \boldsymbol{\mathcal{V}}_m^{(2)} \boldsymbol{Y}_m^{(2)}(t) C_m^{(1)}(t) \text{ with } \boldsymbol{Y}_m^{(2)}(t) := \left(\mathcal{H}_m^{(2)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1.$$

Applying the same logic as in the derivation (3.9), we find that the residual to the equation $(A+tI)\boldsymbol{Z}_*(t) = \boldsymbol{R}_m^{(1)}(t)$ for the approximation $\boldsymbol{Z}_m^{(1)}(t)$,

$$\boldsymbol{R}_m^{(1)}(t) - (A+tI)\boldsymbol{Z}_m^{(1)}(t) = -\boldsymbol{V}_{m+1}^{(2)} H_{m+1,m}^{(2)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(2)}(t) C_m^{(1)}(t),$$

is block orthogonal to $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{V}_1^{(2)})$, thus satisfying the block Galerkin condition (3.8) and affirming that $\boldsymbol{Z}_m^{(1)}(t)$ as defined is indeed the BFOM approximation for the residual equation $(A+tI)\boldsymbol{Z}_*^{(1)}(t) = \boldsymbol{R}_m^{(1)}(t)$. The residual $\boldsymbol{R}_m^{(2)}(t)$ of the updated approximation $\boldsymbol{X}_m^{(2)}(t) = \boldsymbol{X}_m^{(1)}(t) + \boldsymbol{Z}_m^{(1)}(t)$ is then given as

$$\boldsymbol{R}_m^{(2)}(t) = \boldsymbol{R}_m^{(1)}(t) - (A+tI)\boldsymbol{Z}_m^{(1)}(t) = -\boldsymbol{V}_{m+1}^{(2)} H_{m+1,m}^{(2)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(2)}(t) C_m^{(1)}(t).$$

Defining $C_m^{(2)}(t) := -H_{m+1,m}^{(2)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(2)}(t)$ leads to a succinct expression for the cospatiality relationship between $\boldsymbol{R}_m^{(2)}(t)$ and $\boldsymbol{V}_{m+1}^{(2)}$,

$$\boldsymbol{R}_m^{(2)}(t) = \boldsymbol{V}_{m+1}^{(2)} C_m^{(2)}(t) C_m^{(1)}(t).$$

Inductively, if we start the $k+1$st cycle with the $m+1$st block basis vector from the previous cycle, i.e., if we take $\boldsymbol{V}_1^{(k+1)} = \boldsymbol{V}_{m+1}^{(k)}$, we can then describe all cycles with the following quantities:

$$\boldsymbol{Y}_m^{(k+1)}(t) = \left(\mathcal{H}_m^{(k+1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 \tag{3.15}$$

$$\boldsymbol{Z}_m^{(k)}(t) = \boldsymbol{\mathcal{V}}_m^{(k+1)} \boldsymbol{Y}_m^{(k+1)}(t) C_m^{(k)}(t) \cdots C_m^{(1)}(t) \tag{3.16}$$

$$\boldsymbol{X}_m^{(k+1)}(t) = \boldsymbol{X}_m^{(k)}(t) + \boldsymbol{Z}_m^{(k)}(t) \tag{3.17}$$

$$\boldsymbol{R}_m^{(k+1)}(t) = \boldsymbol{R}_m^{(k)}(t) - (A+tI)\boldsymbol{Z}_m^{(k)}(t) = \boldsymbol{V}_{m+1}^{(k)} C_m^{(k)}(t) \cdots C_m^{(1)}(t), \tag{3.18}$$

with

$$C_m^{(j)}(t) = -H_{m+1,m}^{(j)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(j)}(t), \quad j = 1, \ldots, k. \tag{3.19}$$

Shifted BFOM with restarts is summarized by Algorithm 3.2.1. Given a predetermined set of shifts $t \in \{t_i\}_{i=1}^N$, it is possible to make the implementation of Algorthim 3.2.1 very efficient. At each cycle k, a new Krylov basis is generated that can be reused for each t_i, so lines 3-5 and 8-10 can be gathered in loops over the index i. Furthermore, by carefully replacing intermediate quantities, it is possible to predetermine and preallocate precisely how much memory is required for Algorithm 3.2.1. In particular, we compute and store $\widetilde{C}_m^{(k)}(t) = C_m^{(k)}(t) \cdots C_m^{(1)}(t)$ instead of the individual matrices $C_m^{(j)}(t)$. Of course, if the set of shifts is not fixed per cycle, it is more difficult to preallocate memory.

3.2.3 Error bounds for shifted systems with restarts

Let the error to the Sh-BFOM(m) approximation at cycle k be given as

$$\boldsymbol{E}_m^{(k)}(t) := \boldsymbol{X}_*(t) - \boldsymbol{X}_m^{(k)}(t). \tag{3.20}$$

The goal now is to bound (3.20) by a quantity that decreases as k increases. To that end, we consider shifted versions of quantities (3.3) for a BSA and BPD operator A, with $\operatorname{spec}(A) \subset [\lambda_{\min}, \lambda_{\max}]$, $\lambda_{\min} > 0$:

$$\kappa(t) := \frac{\lambda_{\max} + t}{\lambda_{\min} + t}, \quad c(t) := \frac{\sqrt{\kappa(t)} - 1}{\sqrt{\kappa(t)} + 1}, \quad \text{and} \quad \xi_m(t) := \frac{1}{\cosh(m \ln c(t))}. \tag{3.21}$$

Theorem 3.23: Given $A \in \mathbb{C}^{n \times n}$ BSA and BPD and $t \geq 0$, let $\boldsymbol{X}_m^{(k)}(t)$ be the Sh-BFOM(m) approximation to the shifted system (3.1) at cycle k. Then the error can

Algorithm 3.2.1: Sh-BFOM(m): shifted BFOM with restarts

1: Given A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, m, t, `tol`

2: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(1)}$, $\underline{\mathcal{H}}_m^{(1)}$, and $B^{(1)}$

3: Compute $\boldsymbol{Y}_m^{(1)}(t) = \left(\mathcal{H}_m^{(1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)}$

4: Compute and store $\boldsymbol{X}_m^{(1)}(t) = \boldsymbol{\mathcal{V}}_m^{(1)} \boldsymbol{Y}_m^{(1)}(t)$

5: Compute and store $\widetilde{C}_m^{(1)}(t) = H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(1)}(t)$

6: **for** $k = 1, 2, \ldots,$ until convergence **do**

7: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{V}_{m+1}^{(k)}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(k+1)}$, $\underline{\mathcal{H}}_m^{(k+1)}$, and $B^{(k+1)}$ in place of previous cycle

8: Compute $\boldsymbol{Y}_m^{(k+1)}(t) = \left(\mathcal{H}_m^{(k+1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1$

9: Compute $\boldsymbol{X}_m^{(k+1)}(t) := \boldsymbol{X}_m^{(k)}(t) + \boldsymbol{\mathcal{V}}_m^{(k+1)} \boldsymbol{Y}_m^{(k+1)}(t) \widetilde{C}_m^{(k)}(t)$ and replace $\boldsymbol{X}_m^{(k)}(t)$

10: Compute $\widetilde{C}_m^{(k+1)}(t) = H_{m+1,m}^{(k+1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(k+1)}(t) \widetilde{C}_m^{(k)}(t)$ and replace $\widetilde{C}_m^{(k)}(t)$

11: **end for**

12: **return** $\boldsymbol{X}_m^{(k+1)}(t)$

be bounded for all $t \geq 0$ as

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A\text{-}\mathbb{S}} \leq \xi_m(t)^k \sqrt{\frac{\lambda_{\max}}{(\lambda_{\min}+t)(\lambda_{\max}+t)}}\,\|\boldsymbol{B}\|_{\mathbb{S}}\,. \tag{3.22}$$

Proof: Lemma 3.12 provides for any $\boldsymbol{V} \in \mathbb{C}^{n\times s}$ that

$$\begin{aligned}\|\boldsymbol{V}\|^2_{(A+tI)\text{-}\mathbb{S}} &= \langle \boldsymbol{V}, \boldsymbol{V}\rangle_{A\text{-}\mathbb{S}} + t\langle \boldsymbol{V}, \boldsymbol{V}\rangle_{\mathbb{S}} = \|\boldsymbol{V}\|^2_{A\text{-}\mathbb{S}} + t\,\|\boldsymbol{V}\|^2_{\mathbb{S}} \\ &\geq \|\boldsymbol{V}\|^2_{A\text{-}\mathbb{S}} + \tfrac{t}{\lambda_{\max}}\,\|\boldsymbol{V}\|_{A\text{-}\mathbb{S}} = \tfrac{\lambda_{\max}+t}{\lambda_{\max}}\,\|\boldsymbol{V}\|^2_{A\text{-}\mathbb{S}}\,;\end{aligned}$$

consequently, $\|\boldsymbol{V}\|_{A\text{-}\mathbb{S}} \leq \sqrt{\frac{\lambda_{\max}}{\lambda_{\max}+t}}\,\|\boldsymbol{V}\|_{(A+tI)\text{-}\mathbb{S}}$. Then

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A\text{-}\mathbb{S}} \leq \sqrt{\frac{\lambda_{\max}}{\lambda_{\max}+t}}\,\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{(A+tI)\text{-}\mathbb{S}}\,, \tag{3.23}$$

and repeated application of Theorem 3.20 to $\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{(A+tI)\text{-}\mathbb{S}}$ gives that

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{(A+tI)\text{-}\mathbb{S}} \leq \xi_m(t)^k \left\|\boldsymbol{E}_0^{(1)}(t)\right\|_{(A+tI)\text{-}\mathbb{S}}\,. \tag{3.24}$$

Since $\boldsymbol{E}_0^{(1)}(t) = \boldsymbol{X}_*(t)$, we can use Lemma 3.12 again to bound

$$\begin{aligned}\left\|\boldsymbol{E}_0^{(1)}(t)\right\|^2_{(A+tI)\text{-}\mathbb{S}} &= \langle (A+tI)^{-1}\boldsymbol{B}, (A+tI)(A+tI)^{-1}\boldsymbol{B}\rangle_{\mathbb{S}} \\ &= \|\boldsymbol{B}\|^2_{(A+tI)^{-1}\text{-}\mathbb{S}} \leq \frac{1}{\lambda_{\min}+t}\,\|\boldsymbol{B}\|^2_{\mathbb{S}}\,.\end{aligned}$$

□

Remark 3.24: Note that for all $t \geq 0$, $0 \leq \xi_m(t) < 1$, and $\lim_{t\to\infty} \xi_m(t) = 0$; see [52, Proposition 4.2]. This comment in combination with Theorem 3.23 shows that

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A\text{-}\mathbb{S}} \leq \xi_m(0)^k \frac{1}{\sqrt{\lambda_{\min}}}\,\|\boldsymbol{B}\|_{\mathbb{S}}\,,$$

i.e., that the error of the shifted restarted BFOM approximation can be bounded independent of the shift.

3.3 Summary and outlook

The comprehensive framework explored in this chapter expands work begun by [46, 69, 73]. Via the *-subalgebra $\mathbb{S}$, block inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, and scaling quotient $N(\cdot)$, the formalism gives a complete algebraic description for block Krylov subspace methods. We have demonstrated in particular what shifted block FOM with restarts looks like in this framework and have derived error bounds for shifted systems with restarts. The algorithm Sh-BFOM(m) provides a versatile implementation of shifted BFOM with restarts, along with notes on storage to aid with memory preallocation.

Our perspective on block Krylov methods leads to a number of insights which may prove useful in other fields and future work. The generality of $\mathbb{S}$ and the block inner product allows one to titrate the amount of information being communicated among columns of the block vectors, which is important for adapting block methods to different computer architectures. We examine this behavior in more detail in Chapter 7. Having the freedom to choose $\mathbb{S}$ and $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ may also prove to be helpful in problems requiring that certain properties or sparsity patterns be maintained. Weighted block inner products may also lead to improvement in some problems [44, 47], and the generality of our framework provides a head-start on the analysis of such methods.

Some concepts remain to be generalized within our framework. Block grade is defined for the classical case in [69, 70] and block angles between subspaces are explored in [134], but it remains open how one should define these concepts for general block inner products. Doing so could lead to a deeper understanding of stagnation for less standard block inner products, like the hybrid or weighted inner

products. It is also possible to generalize many other types of Krylov methods with the concepts presented in this chapter, e.g., block MINRES, two-sided block Lanczos, quasiminimal residual methods; extended and rational Krylov methods; and so forth. We examine block GMRES and the so-called Radau-Lanczos method in our framework in the next chapter.

CHAPTER 4

MODIFIED BLOCK FULL ORTHOGONALIZATION METHODS

We continue in the framework from the previous chapter. That is, we assume that we have a *-subalgebra $\mathbb{S}$, a block inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, a scaling quotient N, and a block Krylov subspace $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ with block Arnoldi relation (3.5):

$$A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^*. \tag{3.5 revisited}$$

We now consider how to modify the block full orthogonalization method (BFOM) via $\mathcal{H}_m$, but first we must understand how $\mathcal{H}_m$ represents A in the block Krylov subspace. Lemma 3.15 states how the block upper Hessenberg $\mathcal{H}_m$ captures some features of A, and Lemma 3.13 ensures that the spectrum of $\mathcal{H}_m$ is always near, in the sense of field of values, to the spectrum of A. Indeed, it is well known in the non-block and classical block settings that as m approaches the (block) grade of the Krylov space, the eigenvalues of $\mathcal{H}_m$ converge to a subset of the eigenvalues of A [46, 70, 109].

When the number m of Krylov basis vectors that we can store is small, it is possible that $\operatorname{spec}(\mathcal{H}_m)$ may be a poor approximation to $\operatorname{spec}(A)$. Increasing

m would improve the approximation, but since we assume m is bounded by some hardware limitation, this is not necessarily feasible. Another issue with some block inner products, such as the global one, is that only a small subset of spec(A) is captured and then given high multiplicity, thus leaving the method in a kind of rut. Preconditioning is another viable technique, and it is well established for improving the robustness of Krylov subspace methods (see, e.g., [67, 122]); however, traditional preconditioning techniques (i.e., simple left or right preconditioning) are not feasible for general functions of matrices, which is our end goal in Chapter 5.

We propose an approach that can adjust the spectrum of $\mathcal{H}_m$ without having to compute additional basis vectors or recompute the Krylov basis altogether. Quite simply, we add to $\mathcal{H}_m$ a low-rank matrix $\mathcal{M}$ of a particular structure. The requirements on the modification $\mathcal{M}$ ensure that the resulting BFOM-like approximation remains in the Krylov subspace, and are derived by examining a polynomial version of the block Arnoldi relation, which we do in Section 4.1. Furthermore, this approach encompasses the block GMRES method of [127, 129, 130] and, within the block framework of Chapter 3, generalizes a relatively new method known as the Radau-Lanczos method of [54].

4.1 A block Arnoldi polynomial relation

We first consider the non-shifted, non-restarted scenario, i.e., that we have an operator $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$ and $\boldsymbol{B} \in \mathbb{C}^{n\times s}$, and we build a block Krylov subspace satisfying the block Arnoldi relation (3.5):

$$A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m\mathcal{H}_m + \boldsymbol{V}_{m+1}H_{m+1,m}\widehat{\boldsymbol{E}}_m^*. \qquad \text{(3.5 revisited)}$$

Recall that $B \in \mathbb{S}$ is such that $\boldsymbol{B} = \boldsymbol{V}_1 B$, from the first step of Algorithm 3.1.1. Right multiplying both sides of relation (3.5) by $\widehat{\boldsymbol{E}}_1 B$ gives that

$$A\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \widehat{\boldsymbol{E}}_1 B. \tag{4.1}$$

Left multiplying equation (4.1) by A and substituting (3.5) results in

$$\begin{aligned} A^2\boldsymbol{B} &= A\boldsymbol{\mathcal{V}}_m \mathcal{H}_m \widehat{\boldsymbol{E}}_1 B \\ &= \boldsymbol{\mathcal{V}}_m \mathcal{H}_m^2 \widehat{\boldsymbol{E}}_1 B + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \mathcal{H}_m \widehat{\boldsymbol{E}}_1 B. \end{aligned}$$

But $\widehat{\boldsymbol{E}}_m^* \mathcal{H}_m \widehat{\boldsymbol{E}}_1 = \widehat{\boldsymbol{E}}_m^* (H_{11}\widehat{\boldsymbol{E}}_1 + H_{21}\widehat{\boldsymbol{E}}_2) = 0$, so

$$A^2\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m^2 \widehat{\boldsymbol{E}}_1 B. \tag{4.2}$$

Conveniently, $\mathcal{H}_m^j \widehat{\boldsymbol{E}}_1$ has a special structure that we can take advantage of for $j = 1, \ldots, m-2$. From Figure 4.1, we can conclude that for all $j = 1, \ldots, m-2$, $\widehat{\boldsymbol{E}}_m^* \mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 = 0$. Then, by repeatedly left-multiplying equation (4.2) by A and using the block Arnoldi relation (3.5), we find that

$$A^j\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 B, \text{ for all } j = 0, \ldots, m-1,$$

or, more generally, the *block Arnoldi polynomial relation*:

$$Q(A) \circ \boldsymbol{B} = \boldsymbol{\mathcal{V}}_m Q(\mathcal{H}_m) \circ \widehat{\boldsymbol{E}}_1 B, \text{ for all } Q \in \mathbb{P}_j(\mathbb{S}), \quad j = 0, \ldots, m-2. \tag{4.3}$$

The block Arnoldi polynomial relation (4.3) provides a powerful perspective on the representation of elements in $\mathscr{K}_m(A, \boldsymbol{B})$. We know already that any element in $\mathscr{K}_m(A, \boldsymbol{B})$ can be written as a matrix polynomial up to degree $m-1$ acting on A and $\boldsymbol{B}$ (see Section 3.1.5). The relation (4.3) states that this element can also

be represented in terms of the block orthonormal basis $\boldsymbol{\mathcal{V}}_m$ multiplied by the same polynomial acting instead on $\mathcal{H}_m$ and $\widehat{\boldsymbol{E}}_1 B$.

$$\mathcal{H}_m = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}, \quad \widehat{\boldsymbol{E}}_1 = \begin{bmatrix} \times \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{H}_m \widehat{\boldsymbol{E}}_1 = \begin{bmatrix} \times \\ \times \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \mathcal{H}_m^2 \widehat{\boldsymbol{E}}_1 = \mathcal{H}_m(\mathcal{H}_m \widehat{\boldsymbol{E}}_1) = \begin{bmatrix} \times \\ \times \\ \times \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Rightarrow \cdots \Rightarrow \quad \mathcal{H}_m^{m-2} \widehat{\boldsymbol{E}}_1 = \begin{bmatrix} \times \\ \times \\ \times \\ \times \\ \times \\ 0 \end{bmatrix}$$

FIGURE 4.1. The zero-non-zero block structure of successive powers of $\mathcal{H}_m$ for $m = 6$. The symbol $\times$ represents a non-zero block entry.

We wish to know to what kinds of restrictions must be placed on $\mathcal{M} \in \mathbb{S}^{m \times m}$ so that a relation like (4.3) holds for $\mathcal{H}_m + \mathcal{M}$. Such additive modifications are equivalent to altering the matrix polynomial that interpolates the modified Hessenberg matrix $\mathcal{H}_m + \mathcal{M}$. When $s = 1$, it is already known that $\mathcal{H}_m$ can only be modified by a rank-one matrix of the form $\boldsymbol{w}\widehat{\boldsymbol{e}}_m^*$, where m is the dimension of the Krylov subspace; see, e.g., [54, Lemmas 1.3 and 1.4], as well as [34, 52, 109, 121, 137]. In the block case, the modifying matrix takes on a similar form, based on the underlying *-subalgebra $\mathbb{S}$.

Theorem 4.1: Let $\mathcal{M} \in \mathbb{S}^{m\times m}$. The matrix $\mathcal{M}$ has the form $\mathcal{M} = \boldsymbol{M}\widehat{\boldsymbol{E}}_m$ for some $\boldsymbol{M} \in \mathbb{S}^m$ if and only if for all $Q \in \mathbb{P}_{m-1}(\mathbb{S})$

$$Q(A) \circ \boldsymbol{B} = \boldsymbol{\mathcal{V}}_m Q(\mathcal{H}_m + \mathcal{M}) \circ \widehat{\boldsymbol{E}}_1 B. \tag{4.4}$$

Proof: Since addition by $\boldsymbol{M}\widehat{\boldsymbol{E}}_m^*$, for $\boldsymbol{M} \in \mathbb{S}^m$ does not alter the structure of the block upper Hessenberg matrix $\mathcal{H}_m$, the relationship from Figure 4.1 holds for $(\mathcal{H}_m + \boldsymbol{M}\widehat{\boldsymbol{E}}_m^*)^j\widehat{\boldsymbol{E}}_1$; consequently,

$$\widehat{\boldsymbol{E}}_m^*(\mathcal{H}_m + \boldsymbol{M}\widehat{\boldsymbol{E}}_m^*)^j\widehat{\boldsymbol{E}}_1 = 0, \text{ for } j = 1, \ldots, m-2, \tag{4.5}$$

but

$$\widehat{\boldsymbol{E}}_j^*(\mathcal{H}_m + \boldsymbol{M}\widehat{\boldsymbol{E}}_m^*)^j\widehat{\boldsymbol{E}}_1 \neq 0, \text{ for } j = 1, \ldots, m-1. \tag{4.6}$$

We split the rest of the proof into two parts.

1. *Sufficient conditions.* Assume that $\mathcal{M} = \boldsymbol{M}\widehat{\boldsymbol{E}}_m^*$, for some $\boldsymbol{M} \in \mathbb{S}^m$. It is enough to show that $A^j\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B$ for all $j = 0, \ldots, m-1$, which we do by induction. Since $A^0\boldsymbol{B} = \boldsymbol{B} = \boldsymbol{V}_1 B = \boldsymbol{\mathcal{V}}_m\widehat{\boldsymbol{E}}_1 B$, we have that the base case holds. Fixing $j \in \{0, \ldots, m-2\}$, we take as our induction hypothesis that

$$A^j\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B. \tag{4.7}$$

Then

$$\begin{aligned} A^{j+1}\boldsymbol{B} &= AA^j\boldsymbol{B} = A\boldsymbol{\mathcal{V}}_m(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B, \quad \text{by (4.7)} \\ &= (\boldsymbol{\mathcal{V}}_m\mathcal{H}_m + \boldsymbol{V}_{m+1}H_{m+1,m}\widehat{\boldsymbol{E}}_m^*)(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B, \quad \text{by (3.5)} \\ &= \boldsymbol{\mathcal{V}}_m\mathcal{H}_m(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B + \boldsymbol{V}_{m+1}H_{m+1,m}\widehat{\boldsymbol{E}}_m^*(\mathcal{H}_m + \mathcal{M})^j\widehat{\boldsymbol{E}}_1 B. \end{aligned} \tag{4.8}$$

By equation (4.5), the second term in equation (4.8) vanishes. Similarly, since $\mathcal{M} = \boldsymbol{M}\widehat{\boldsymbol{E}}_m^*$,

$\mathcal{M}(\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B = 0$, for all $j = 1, \ldots, m-2$. Then equation (4.8) becomes

$$\begin{aligned}
&\boldsymbol{\mathcal{V}}_m \mathcal{H}_m (\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B \\
&= \boldsymbol{\mathcal{V}}_m \mathcal{H}_m (\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B + \mathcal{M}(\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B \\
&= \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + \mathcal{M})^{j+1} \widehat{\boldsymbol{E}}_1 B,
\end{aligned}$$

concluding the induction process.

2. *Necessary conditions.* Let $\mathcal{M} \in \mathbb{S}^{m\times m}$ and suppose that for all $Q \in \mathbb{P}_{m-1}(\mathbb{S})$, relation (4.4) holds. Then, in particular,

$$A^j \boldsymbol{B} = \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B, \quad \forall j = 1, \ldots, m-1,$$

and by part 1 with modification $\mathcal{M} = 0$,

$$A^j \boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 B, \quad \forall j = 1, \ldots, m-1.$$

Then

$$\boldsymbol{\mathcal{V}}_m \mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 B = \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1 B, \quad \forall j = 1, \ldots, m-1.$$

Since $\boldsymbol{\mathcal{V}}_m$ has full rank and B is non-singular, $\mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 = (\mathcal{H}_m + \mathcal{M})^j \widehat{\boldsymbol{E}}_1$ for all $j = 1, \ldots, m-1$. It then follows that for all $j = 1, \ldots, m-1$,

$$\mathcal{H}_m^j \widehat{\boldsymbol{E}}_1 = (\mathcal{H}_m + \mathcal{M}) \mathcal{H}_m^{j-1} \widehat{\boldsymbol{E}}_1, \quad \forall j = 0, \ldots, m-1,$$

which implies for all $j = 1, \ldots, m-1$ that $\mathcal{M}\mathcal{H}_m^{j-1} \widehat{\boldsymbol{E}}_1 = 0$. Then

$$\mathcal{M}\boldsymbol{\mathcal{R}} = 0, \tag{4.9}$$

where

$$\boldsymbol{\mathcal{R}} = [\widehat{\boldsymbol{E}}_1 | \mathcal{H}_m \widehat{\boldsymbol{E}}_1 | \ldots | \mathcal{H}_m^{m-2} \widehat{\boldsymbol{E}}_1] \in \mathbb{S}^{m \times (m-1)}.$$

However, by equations (4.5) and (4.6),

$$\widehat{\boldsymbol{E}}_j^* \mathcal{H}_m^{j-1} \widehat{\boldsymbol{E}}_1 \neq 0, \quad \forall j = 1, \ldots, m-1,$$

$$\widehat{\boldsymbol{E}}_m^* \mathcal{H}_m^{j} \widehat{\boldsymbol{E}}_1 = 0, \quad \forall j = 1, \ldots, m-1.$$

Therefore, $\boldsymbol{\mathcal{R}}$ is a proper block upper triangular matrix. Relation (4.9) implies that possibly only the mth block column of $\mathcal{M}$ could be nonzero, meaning that $\mathcal{M}$ is of the desired form. □

It is also worth noting that this theorem does not merely generalize results from $s = 1$ to the block case; it also generalizes the results for a variety of inner products. Other polynomial methods for computing matrix functions may be regarded as Krylov subspace methods with the appropriate inner product and starting vector; see, for example, [22].

4.2 Shifted BGMRES with restarts: Sh-BGMRES(m)

Both FOM and GMRES are known to stagnate in certain cases for both block and single-column vectors [38, 52, 126, 134]. For our block Krylov framework to be full-fledged, it is therefore important to develop the analogous block GMRES approximation to equation (3.1). We require that this approximation satisfy a block Petrov-Galerkin condition, as well as some relation ensuring that the shifted residuals are cospatial to the non-shifted one (cf. equation (3.18), the cospatial relationship for

Sh-BFOM(m)). This cospatial relation becomes important for formulating efficient restarts and leads to a succinct convergence analysis.

The theory we develop draws inspiration from [69, 127, 130] for classical GMRES and [83, 73, 46] for global GMRES. In particular, we build off of Theorem 3.3 of Simoncini and Gallopoulos [130], which establishes the form a non-shifted approximation should take for the classical block inner product and points to how BGMRES can be thought of as a modified BFOM.

We additionally provide two approaches for obtaining error bounds, and each approach is valid for different subsets of block positive real operators. The differences hinge on which version of the cospatial relationship we use.

4.2.1 The approximation

Suppose that the approximation $\boldsymbol{X}_m(t)$ to equation (3.1) satisfies

- the block Petrov-Galerkin condition for $t = 0$:

$$\boldsymbol{B} - A\boldsymbol{X}_m(0) \perp_{\mathbb{S}} A\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}), \;\; \boldsymbol{X}_m(0) \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}); \tag{4.10}$$

- and a shifted condition for $t \neq 0$:

$$\boldsymbol{B} - (A + tI)\boldsymbol{X}_m(t) \perp_{\mathbb{S}} A\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}), \;\; \boldsymbol{X}_m(t) \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B}). \tag{4.11}$$

Note that condition (4.11) gives rise to an approximation that is not, strictly speaking, the BGMRES approximation for equation (3.1). A true BGMRES approximation would require that the shifted residual be block orthogonal to $(A+tI)\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$, instead of $A\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. Further note that both residuals, $\boldsymbol{R}_m(0) := \boldsymbol{B} - A\boldsymbol{X}_m(0)$ and $\boldsymbol{R}_m(t) := \boldsymbol{B} - (A + tI)\boldsymbol{X}_m(t)$, lie in the space $\mathscr{K}_{m+1}^{\mathbb{S}}(A, \boldsymbol{B})$, and thus in the

$\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-orthogonal complement of $A\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ in that space, implying that they are cospatial. Let

$$\mathcal{M} := \mathcal{H}_m^{-*} \widehat{\boldsymbol{E}}_m H_{m+1,m}^* H_{m+1,m} \widehat{\boldsymbol{E}}_m^*, \tag{4.12}$$

and define

$$\boldsymbol{X}_m(t) := \boldsymbol{\mathcal{V}}_m \boldsymbol{Y}_m(t), \tag{4.13}$$

with

$$\boldsymbol{Y}_m(t) := (\mathcal{H}_m + \mathcal{M} + tI)^{-1} \widehat{\boldsymbol{E}}_1 B. \tag{4.14}$$

Theorem 4.2: With $\boldsymbol{X}_m(t)$ defined as in equation (4.13), $\boldsymbol{R}_m(0)$ satisfies condition (4.10) and $\boldsymbol{R}_m(t)$ satisfies condition (4.11), for all $t > 0$.

Proof: The block orthogonality condition on $\boldsymbol{R}_m(t)$ is equivalent to

$$\langle\!\langle A\boldsymbol{\mathcal{V}}_m \boldsymbol{Z}, \boldsymbol{B} - (A + tI)\boldsymbol{\mathcal{V}}_m \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}} = 0, \text{ for all } \boldsymbol{Z} \in \mathbb{S}^m,$$

the left-hand side of which can be expanded via the block Arnoldi relation (3.5) as

$$\langle\!\langle (\boldsymbol{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^*) \boldsymbol{Z}, \boldsymbol{B} - (\boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + tI) + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^*) \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}}.$$

This can be further broken into four parts:

$$\langle\!\langle \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{Z}, \boldsymbol{B} - \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + tI) \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}} \tag{4.15}$$

$$- \langle\!\langle \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{Z}, \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}} \tag{4.16}$$

$$+ \langle\!\langle \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z}, \boldsymbol{B} - \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + tI) \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}} \tag{4.17}$$

$$- \langle\!\langle \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z}, \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}}. \tag{4.18}$$

Writing $\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \widehat{\boldsymbol{E}}_1 B$, the term $\boldsymbol{B} - \boldsymbol{\mathcal{V}}_m (\mathcal{H}_m + tI) \boldsymbol{Y}_m(t)$ becomes $\boldsymbol{\mathcal{V}}_m (\widehat{\boldsymbol{E}}_1 B - (\mathcal{H}_m + tI) \boldsymbol{Y}_m(t))$. Then both (4.16) and (4.17) are of the form $\langle\!\langle \boldsymbol{\mathcal{V}}_m \boldsymbol{W}, \boldsymbol{V}_{m+1} W \rangle\!\rangle_{\mathbb{S}}$,

for some $\boldsymbol{W} \in \mathbb{S}^m$ and $W \in \mathbb{S}$, and consequently both terms equal zero by the block orthonormality of the Krylov basis vectors. Using again that $\boldsymbol{B} = \boldsymbol{\mathcal{V}}_m \widehat{\boldsymbol{E}}_1 B$, and also that $(\mathcal{H}_m + \mathcal{M} + tI)\boldsymbol{Y}_m(t) = \widehat{\boldsymbol{E}}_1 B$, the term (4.15) can be rewritten as $\langle\!\langle \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{Z}, \boldsymbol{\mathcal{V}}_m \mathcal{M} \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}}$. The final term (4.18) can be rewritten by taking advantage of properties of the block inner product and the block orthonormality of the Krylov basis:

$$\begin{aligned} &\langle\!\langle \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z}, \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \rangle\!\rangle_{\mathbb{S}} \\ &= (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z})^* (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m). \end{aligned}$$

It therefore suffices to show that

$$\langle\!\langle \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{Z}, \boldsymbol{\mathcal{V}}_m \mathcal{M} \boldsymbol{Y}_m(t) \rangle\!\rangle_{\mathbb{S}} = (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z})^* (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m).$$

Indeed, by Lemma 3.9 and the definition of $\mathcal{M}$, we find that

$$\begin{aligned} \langle\!\langle \boldsymbol{\mathcal{V}}_m \mathcal{H}_m \boldsymbol{Z}, \boldsymbol{\mathcal{V}}_m \mathcal{M} \boldsymbol{Y}_m \rangle\!\rangle_{\mathbb{S}} &= \sum_{j=1}^{m} (\widehat{\boldsymbol{E}}_j^* \mathcal{H}_m \boldsymbol{Z})^* \widehat{\boldsymbol{E}}_j^* (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \mathcal{H}_m^{-1})^* H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \\ &= \sum_{j=1}^{m} \boldsymbol{Z}^* \mathcal{H}_m^* \widehat{\boldsymbol{E}}_j \widehat{\boldsymbol{E}}_j^* \mathcal{H}_m^{-*} (H_{m+1,m} \widehat{\boldsymbol{E}}_m^*)^* H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \\ &= \boldsymbol{Z}^* \mathcal{H}_m^* \mathcal{H}_m^{-*} (H_{m+1,m} \widehat{\boldsymbol{E}}_m^*)^* H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m \\ &= (H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Z})^* H_{m+1,m} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m. \end{aligned}$$ □

By Theorem 3.16 and Corollary 3.16, satisfying condition (4.10) is equivalent to minimizing the residual norm. Note that this equivalence only holds for the non-shifted residual, and is originally stated for the classical paradigm in [130, Section 1] and for the global paradigm in [73, Section 3.2] and [46, Section 2.2].

Corollary 4.3: The BGMRES residual minimizes the $\mathbb{S}$-norm, i.e.,

$$\|\boldsymbol{R}_m(0)\|_{\mathbb{S}} = \min_{\boldsymbol{X}\in\mathscr{K}_m^{\mathbb{S}}(A,\boldsymbol{B})} \|\boldsymbol{B} - A\boldsymbol{X}(0)\|_{\mathbb{S}} = \min_{\substack{P\in\mathbb{P}_m(\mathbb{S})\\ P(0)=I}} \|P(A)\circ\boldsymbol{B}\|_{\mathbb{S}}\,. \tag{4.19}$$

Theorem 3.22 establishes a relationship between the BFOM errors for different block inner products, as a consequence of a minimization property and the fact that $\mathbb{S}^{\mathtt{Gl}} \subseteq \mathbb{S}^{\mathtt{Li}} \subseteq \mathbb{S}^{\mathtt{Hy}} \subseteq \mathbb{S}^{\mathtt{Cl}}$. Thanks to Corollary 4.3, a similar property holds for the BGMRES residuals, compare with [46, Theorem 2.4].

Theorem 4.4: Let $\boldsymbol{R}_m^{\square}$ denote the residual for the mth BGMRES approximation $\boldsymbol{X}_m^{\square}$, where $\square$ denotes the choice of one of the paradigms from Table 3.1. Then

$$\left\|\boldsymbol{R}_m^{\mathtt{Cl}}(0)\right\|_{\mathbb{S}} \le \left\|\boldsymbol{R}_m^{\mathtt{Hy}}(0)\right\|_{\mathbb{S}} \le \left\|\boldsymbol{R}_m^{\mathtt{Li}}(0)\right\|_{\mathbb{S}} \le \left\|\boldsymbol{R}_m^{\mathtt{Gl}}(0)\right\|_{\mathbb{S}}\,. \tag{4.20}$$

4.2.2 Cospatial factors

In [51], the authors develop a restarted shifted GMRES method for a single right-hand side. Their analysis hinges on the shifted residuals being collinear to the non-shifted residual, i.e., $\boldsymbol{r}(t) = \boldsymbol{r}(0)\rho(t)$, for some scalar collinear factor $\rho(t)$, that is derived from polynomials interpolating the *harmonic Ritz values*, or the eigenvalues of $\mathcal{H}_m + \mathcal{M}$; see Remark 4.17. While it is possible to proceed by analogy within our block framework, one has to take great care, because the scalar factor of collinearity is replaced by a matrix-valued factor for cospatiality, now derived from non-commutative matrix polynomials. Without commutativity, such a factor becomes difficult to analyze, especially for deriving error bounds. We therefore take a different approach, showing that $\boldsymbol{R}_m(t)$ is cospatial to some special block vector in

$\mathscr{K}^{\mathbb{S}}_{m+1}(A, \boldsymbol{B})$. An approach that is more similar to that of [51], but also much more technical, can be found in Section 4.2.5.

Lemma 4.5: Denote $\boldsymbol{U} := \mathcal{H}_m^{-*} \widehat{\boldsymbol{E}}_m H^*_{m+1,m} \in \mathbb{S}^m \subset \mathbb{C}^{ms \times s}$, and recall that $\boldsymbol{B} = \boldsymbol{V}_1 B$. Then

$$\boldsymbol{R}_m(t) = \boldsymbol{\mathcal{V}}_{m+1} \begin{bmatrix} \boldsymbol{U} \\ -I \end{bmatrix} \boldsymbol{U}^* \big(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\big)^{-1} \widehat{\boldsymbol{E}}_1 B. \tag{4.21}$$

Proof: Since $\boldsymbol{X}_m(t) \in \mathscr{K}_m(A, \boldsymbol{B})$, there exists $\boldsymbol{G}_m(t) \in \mathbb{S}^m$ such that $\boldsymbol{X}_m(t) = \boldsymbol{\mathcal{V}}_m \boldsymbol{G}_m(t)$. The block Petrov-Galerkin condition (4.11) is then equivalent to

$$0 = \langle\!\langle A\boldsymbol{V}_j, \boldsymbol{B} - (A + tI)\boldsymbol{\mathcal{V}}_m \boldsymbol{G}_m(t) \rangle\!\rangle_{\mathbb{S}}, \text{ for all } j = 1, \ldots, m. \tag{4.22}$$

Using the block Arnoldi relation (3.6) and the fact that $\boldsymbol{B} = \boldsymbol{V}_1 B = \boldsymbol{\mathcal{V}}_{m+1} \widehat{\boldsymbol{E}}_1^{(m+1)} B$, equation (4.22) becomes

$$0 = \left\langle\!\!\left\langle \boldsymbol{\mathcal{V}}_{m+1} \underline{\mathcal{H}}_m \widehat{\boldsymbol{E}}_j, \boldsymbol{\mathcal{V}}_{m+1} (\widehat{\boldsymbol{E}}_1^{(m+1)} B - \left(\underline{\mathcal{H}}_m + t \begin{bmatrix} I \\ 0 \end{bmatrix} \right) \boldsymbol{G}_m(t)) \right\rangle\!\!\right\rangle_{\mathbb{S}}, \text{ for all } j = 1, \ldots, m,$$

which by Lemma 3.9 reduces to

$$0 = (\underline{\mathcal{H}}_m \widehat{\boldsymbol{E}}_j)^* \left(\widehat{\boldsymbol{E}}_1^{(m+1)} B - \left(\underline{\mathcal{H}}_m + t \begin{bmatrix} I \\ 0 \end{bmatrix} \right) \boldsymbol{G}_m(t) \right) \text{ for all } j = 1, \ldots, m. \tag{4.23}$$

Equation (4.23) holding for all $j = 1, \ldots, m$ implies that

$$\begin{aligned} 0 =& \underline{\mathcal{H}}_m^* \left(\widehat{\boldsymbol{E}}_1^{(m+1)} B - \left(\underline{\mathcal{H}}_m + t \begin{bmatrix} I \\ 0 \end{bmatrix} \right) \boldsymbol{G}_m(t) \right) \\ =& \mathcal{H}_m^* \widehat{\boldsymbol{E}}_1^{(m)} B - \underline{\mathcal{H}}_m^* \left(\underline{\mathcal{H}}_m + t \begin{bmatrix} I \\ 0 \end{bmatrix} \right) \boldsymbol{G}_m(t) \\ =& \mathcal{H}_m^* \widehat{\boldsymbol{E}}_1^{(m)} B \\ & - \left(\mathcal{H}_m^* \mathcal{H}_m + (\widehat{\boldsymbol{E}}_m H^*_{m+1,m})(\widehat{\boldsymbol{E}}_m H^*_{m+1,m})^* + t\mathcal{H}_m^* \right) \boldsymbol{G}_m(t) \end{aligned} \tag{4.24}$$

Recalling that $\boldsymbol{U} = \mathcal{H}_m^{-*}\widehat{\boldsymbol{E}}_m H_{m+1,m}^*$, one can show that

$$\begin{aligned}&\left(\mathcal{H}_m^*\mathcal{H}_m + \left(\widehat{\boldsymbol{E}}_m H_{m+1,m}^*\right)\left(\widehat{\boldsymbol{E}}_m H_{m+1,m}^*\right)^* + t\mathcal{H}_m^*\right)^{-1}\\ &= \mathcal{H}_m^{-1}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\mathcal{H}_m^{-*}.\end{aligned}$$

This fact plus equation (4.24) allows us to solve for $\boldsymbol{G}_m(t)$:

$$\boldsymbol{G}_m(t) = \mathcal{H}_m^{-1}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\mathcal{H}_m^{-*}\mathcal{H}_m^*\widehat{\boldsymbol{E}}_1^{(m)}B \tag{4.25}$$

$$= \mathcal{H}_m^{-1}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\widehat{\boldsymbol{E}}_1^{(m)}B. \tag{4.26}$$

We can then write the residual as

$$\begin{aligned}\boldsymbol{R}_m(t) &= \boldsymbol{B} - (A + tI)\boldsymbol{\mathcal{V}}_m\boldsymbol{G}_m(t)\\ &= \boldsymbol{B} - (A + tI)\boldsymbol{\mathcal{V}}_m\mathcal{H}_m^{-1}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\widehat{\boldsymbol{E}}_1^{(m)}B\\ &= \boldsymbol{\mathcal{V}}_{m+1}\widehat{\boldsymbol{E}}_1^{(m+1)}B - \boldsymbol{\mathcal{V}}_{m+1}\left(\underline{\mathcal{H}}_m + t\begin{bmatrix}I\\0\end{bmatrix}\right)\mathcal{H}_m^{-1}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\widehat{\boldsymbol{E}}_1^{(m)}B\\ &= \boldsymbol{\mathcal{V}}_{m+1}\left(\widehat{\boldsymbol{E}}_1^{(m+1)}B - \begin{bmatrix}I + t\mathcal{H}_m^{-1}\\ \boldsymbol{U}^*\end{bmatrix}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)\widehat{\boldsymbol{E}}_1^{(m)}B\right)\\ &= \boldsymbol{\mathcal{V}}_{m+1}\left(I - \begin{bmatrix}I + t\mathcal{H}_m^{-1} & 0\\ \boldsymbol{U}^* & 0\end{bmatrix}\begin{bmatrix}\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1} & 0\\ 0 & I\end{bmatrix}\right)\widehat{\boldsymbol{E}}_1^{(m+1)}B\\ &= \boldsymbol{\mathcal{V}}_{m+1}\begin{bmatrix}\boldsymbol{U}\\-I\end{bmatrix}\boldsymbol{U}^*\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\widehat{\boldsymbol{E}}_1^{(m)}B.\end{aligned}$$

□

Lemma 4.5 shows that all the residuals are cospatial to $\boldsymbol{\mathcal{V}}_{m+1}\begin{bmatrix}\boldsymbol{U}\\-I\end{bmatrix}$ with the cospatiality factor

$$\begin{aligned}G_m(t) &:= \boldsymbol{U}^*\left(I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1}\right)^{-1}\widehat{\boldsymbol{E}}_1^{(m)}B\\ &= H_{m+1,m}\widehat{\boldsymbol{E}}_m^*(\mathcal{H}_m + \mathcal{M} + tI)^{-1}\widehat{\boldsymbol{E}}_1 B.\end{aligned} \tag{4.27}$$

4.2.3 Restarts

We refer to the quantities of Section 4.2.1 (i.e., equations (4.14), (4.13), etc.) as the first cycle, denoting everything with the superscript (1). More explicitly,

$$\begin{aligned}
\boldsymbol{Y}_m^{(1)}(t) &= \left(\mathcal{H}_m^{(1)} + \mathcal{M}^{(1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)} \\
\boldsymbol{X}_m^{(1)}(t) &= \boldsymbol{\mathcal{V}}_m^{(1)} \boldsymbol{Y}_m^{(1)}(t) \\
\boldsymbol{R}_m^{(1)}(t) &= \boldsymbol{\mathcal{V}}_{m+1}^{(1)} \begin{bmatrix} \boldsymbol{U}^{(1)} \\ -I \end{bmatrix} G_m^{(1)}(t) \\
G_m^{(1)}(t) &= H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(1)}(t)
\end{aligned}$$

To restart efficiently, we seek an additive correction to the approximation from the previous cycle. This process is similar to what is done for shifted BFOM with restarts (cf. Section 3.2.2). Suppose we are at cycle k, with $k \geq 1$. To approximate the error $\boldsymbol{E}_m^{(k)}(t) := \boldsymbol{X}(t) - \boldsymbol{X}_m^{(k)}(t)$, we approximate the solution of the residual system

$$(A + tI)\boldsymbol{Z}(t) = \boldsymbol{R}_m^{(k)}(t) = \boldsymbol{\mathcal{V}}_{m+1}^{(k)} \begin{bmatrix} \boldsymbol{U}^{(k)} \\ -I \end{bmatrix} G_m^{(k)}(t) \tag{4.28}$$

with a shifted BGMRES approximation. That is, we normalize $\boldsymbol{\mathcal{V}}_{m+1}^{(k)} \begin{bmatrix} \boldsymbol{U}^{(k)} \\ -I \end{bmatrix}$ as $\boldsymbol{V}_1^{(k+1)} B^{(k+1)}$ to compute the next block Krylov space $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{V}_1^{(k+1)})$ and approximate $\boldsymbol{Z}(t)$ as

$$\boldsymbol{Z}_m^{(k)}(t) := \boldsymbol{\mathcal{V}}_m^{(k+1)} \boldsymbol{Y}_m^{(k+1)}(t), \text{ where} \tag{4.29}$$

$$\boldsymbol{Y}_m^{(k+1)}(t) := \left(\mathcal{H}_m^{(k+1)} + \mathcal{M}^{(k+1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(k+1)} G_m^{(k)}(t). \tag{4.30}$$

Then we update $\boldsymbol{X}_m^{(k)}(t)$ as

$$\boldsymbol{X}_m^{(k+1)}(t) := \boldsymbol{X}_m^{(k)}(t) + \boldsymbol{Z}_m^{(k)}(t). \tag{4.31}$$

Following a similar procedure as in Section 4.2.2, this time for the system (4.28), we find that

$$\boldsymbol{R}_m^{(k+1)}(t) = \boldsymbol{\mathcal{V}}_{m+1}^{(k+1)} \begin{bmatrix} \boldsymbol{U}^{(k+1)} \\ -I \end{bmatrix} G_m^{(k+1)}(t), \text{ and} \tag{4.32}$$

$$G_m^{(k+1)}(t) = H_{m+1,m}^{(k+1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(k+1)}(t). \tag{4.33}$$

The restart procedure is summarized in Algorithm 4.2.1. Note that we never need to compute the shifted residuals, just the cospatial factors and some other intermediate quantities. As with the shifted BFOM approach, we can preallocate storage for some quantities and reuse the block Krylov basis for efficiency.

4.2.4 Error bounds for shifted systems with restarts

We aim to reproduce a result like that of Theorem 3.23 for the shifted BGMRES with restarts on systems where A is a block positive real operator. By Lemma 3.14(i), we have that A and thus also A^{-1} is scalar positive real, implying that all the following quantities are positive and well defined:

$$\begin{aligned} \rho &:= \min \left\{ \frac{\operatorname{Re}(\langle \boldsymbol{V}, A^{-1}\boldsymbol{V} \rangle_{\mathbb{S}})}{\langle \boldsymbol{V}, \boldsymbol{V} \rangle_{\mathbb{S}}} : \boldsymbol{V} \in \mathbb{C}^{n \times s}, \boldsymbol{V} \neq 0 \right\}; \\ \gamma &:= \min \left\{ \frac{\operatorname{Re}(\langle \boldsymbol{V}, A\boldsymbol{V} \rangle_{\mathbb{S}})}{\langle \boldsymbol{V}, \boldsymbol{V} \rangle_{\mathbb{S}}} : \boldsymbol{V} \in \mathbb{C}^{n \times s}, \boldsymbol{V} \neq 0 \right\}; \text{ and} \\ \nu_{\max} &:= \max \left\{ \frac{\langle A\boldsymbol{V}, A\boldsymbol{V} \rangle_{\mathbb{S}}}{\langle \boldsymbol{V}, \boldsymbol{V} \rangle_{\mathbb{S}}} : \boldsymbol{V} \in \mathbb{C}^{n \times s}, \boldsymbol{V} \neq 0 \right\}. \end{aligned} \tag{4.34}$$

We also have some important results on the spectrum of $\mathcal{H}_m + \mathcal{M}$.

Lemma 4.6: Suppose that A is block positive real. Then the matrix $\mathcal{H}_m + \mathcal{M}$ has spectrum with positive real part.

Algorithm 4.2.1: Sh-BGMRES(m): shifted BGMRES with restarts

1: Given A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, m, t, `tol`

2: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(1)}$, $\underline{\mathcal{H}}_m^{(1)}$, and $B^{(1)}$

3: Compute $\boldsymbol{Y}_m^{(1)}(t) = \left(\mathcal{H}_m^{(1)} + \mathcal{M}^{(1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)}$

4: Compute and store $\boldsymbol{X}_m^{(1)}(t) = \boldsymbol{\mathcal{V}}_m^{(1)} \boldsymbol{Y}_m^{(1)}(t)$

5: Compute and store $G_m^{(1)}(t) = H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(1)}(t)$

6: **for** $k = 1, 2, \ldots,$ until convergence **do**

7: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{\mathcal{V}}_{m+1}^{(k)} \begin{bmatrix} \boldsymbol{U}^{(k)} \\ -I \end{bmatrix}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m to obtain $\boldsymbol{\mathcal{V}}_{m+1}^{(k+1)}$, $\underline{\mathcal{H}}_m^{(k+1)}$, and $B^{(k+1)}$

8: Compute $\boldsymbol{Y}_m^{(k+1)}(t) = \left(\mathcal{H}_m^{(k+1)} + \mathcal{M}^{(k+1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 B^{(k+1)} G_m^{(k)}(t)$

9: Update $\boldsymbol{X}_m^{(k+1)}(t) = \boldsymbol{X}_m^{(k)}(t) + \boldsymbol{\mathcal{V}}_m^{(k+1)} \boldsymbol{Y}_m^{(k+1)}(t)$ and replace $\boldsymbol{X}_m^{(k)}(t)$

10: Compute $G_m^{(k+1)}(t) = H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \boldsymbol{Y}_m^{(k)}(t)$ and replace $G_m^{(k)}(t)$

11: **end for**

12: **return** $\boldsymbol{X}_m^{(k+1)}(t)$

Proof: Let $\mathcal{S} := I + \boldsymbol{U}\boldsymbol{U}^*$, and note that it is Hermitian and nonsingular. Further note that

$$\mathcal{H}_m + \mathcal{M} = \mathcal{H}_m + \boldsymbol{U}\boldsymbol{U}^*\mathcal{H}_m = \mathcal{S}\mathcal{H}_m,$$

which is similar to $\mathcal{S}^{1/2}\mathcal{H}_m\mathcal{S}^{1/2}$. Lemma 3.15 implies that $\mathcal{H}_m$ is $\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}$-positive real. Then by Lemma 2.7 (iv)-(v) and Remark 2.8, $\mathcal{S}^{1/2}\mathcal{H}_m\mathcal{S}^{1/2}$ is also $\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}$-positive real and its spectrum has positive real part. Since $\mathcal{S}\mathcal{H}_m$ and $\mathcal{S}^{1/2}\mathcal{H}_m\mathcal{S}^{1/2}$ are only related by similarity, we cannot conclude that $\mathcal{S}\mathcal{H}_m$ is $\langle\cdot,\cdot\rangle_{\boldsymbol{\mathcal{V}}_m}$-positive real, but we can conclude by Lemma 2.7 (i) that $\mathcal{S}\mathcal{H}_m$, and therefore $\mathcal{H}_m + \mathcal{M}$, has spectrum with positive real part. □

A bound like that of [42, Theorem 3.3], relating the unshifted residual to the original right-hand side $\boldsymbol{B}$, holds within our framework.

Theorem 4.7: The non-shifted BGMRES residual $\boldsymbol{R}_m(0)$ can be bounded as

$$\|\boldsymbol{R}_m(0)\|_{\mathbb{S}} \leq \left(1 - \frac{\gamma^2}{\nu_{\max}}\right)^{m/2} \|\boldsymbol{B}\|_{\mathbb{S}}\,. \tag{4.35}$$

Proof: Let $p(z) = 1 - \alpha z$, where α is yet to be determined. By Corollary 4.3, since the $\mathbb{S}$-norm of $\boldsymbol{R}_m = P_m(A) \circ \boldsymbol{B}$ is minimal over polynomials in $\mathbb{P}_m(\mathbb{S})$, we have that

$$\|\boldsymbol{R}_m(0)\|_{\mathbb{S}} \leq \|p(A)^m \boldsymbol{B}\|_{\mathbb{S}} \leq \|p(A)\|_{\mathbb{S}}^m \|\boldsymbol{B}\|_{\mathbb{S}}\,.$$

Since

$$\begin{aligned}\langle p(A)\boldsymbol{V}, p(A)\boldsymbol{V}\rangle_{\mathbb{S}} &= \langle \boldsymbol{V} - \alpha A\boldsymbol{V}, \boldsymbol{V} - \alpha A\boldsymbol{V}\rangle_{\mathbb{S}} \\ &= \langle \boldsymbol{V}, \boldsymbol{V}\rangle_{\mathbb{S}} - 2\alpha \mathrm{Re}(\langle \boldsymbol{V}, A\boldsymbol{V}\rangle_{\mathbb{S}}) + \alpha^2 \langle A\boldsymbol{V}, A\boldsymbol{V}\rangle_{\mathbb{S}},\end{aligned}$$

it holds that

$$\|p(A)\|_{\mathbb{S}}^2 \leq 1 - 2\alpha\gamma + \alpha^2 \nu_{\max}.$$

With $\alpha = \frac{\gamma}{\nu_{\max}}$ minimizing the right-hand side, the inequality (4.35) follows. $\square$

Results similar to those of Theorem 4.7 hold in the case of restarts.

Corollary 4.8: With the restarted residuals $\boldsymbol{R}_m^{(k)}(0)$ defined as in Section 4.2.3,

$$\left\|\boldsymbol{R}_m^{(k)}(0)\right\|_{\mathbb{S}} \leq \left(1 - \frac{\gamma^2}{\nu_{\max}}\right)^{mk/2} \|\boldsymbol{B}\|_{\mathbb{S}}\,. \tag{4.36}$$

Remark 4.9: Let $(\lambda, \boldsymbol{V}) \in \mathbb{C} \times \mathbb{C}^{n\times s}$ be an eigenpair of A such that $\|\boldsymbol{V}\|_{\mathbb{S}} = 1$. Then

$$\gamma^2 \leq \mathrm{Re}(\lambda)^2 \text{ and } |\lambda|^2 = \mathrm{Re}(\lambda)^2 + \mathrm{Im}(\lambda)^2 \leq \nu_{\max},$$

implying that

$$0 < 1 - \frac{\gamma^2}{\nu_{\max}} < 1.$$

Consequently, the bounds of Theorem 4.7 and Corollary 4.8 are indeed decreasing as functions of m and k, albeit possibly slowly, depending on how γ relates to $\nu_{\max}$.

To show that the norms of the shifted residuals are bounded by the norms of the non-shifted ones, we utilize Lemma 4.5, matrix derivatives, and the following auxiliary quantities:

$$\begin{aligned}
\boldsymbol{U} &:= \mathcal{H}_m^{-*} \widehat{\boldsymbol{E}}_m H_{m+1,m}^* \\
\mathcal{T} &:= \boldsymbol{U}(I + \boldsymbol{U}^*\boldsymbol{U})\boldsymbol{U}^* \\
\mathcal{S} &:= I + \boldsymbol{U}\boldsymbol{U}^* \\
\mathcal{S}_t &:= I + \boldsymbol{U}\boldsymbol{U}^* + t\mathcal{H}_m^{-1} = \mathcal{S} + t\mathcal{H}_m^{-1}
\end{aligned} \tag{4.37}$$

Note that $\boldsymbol{U} \in \mathbb{S}^m \subseteq \mathbb{C}^{ms\times s}$ and $\mathcal{T}, \mathcal{S}, \mathcal{S}_t \in \mathbb{S}^{m\times m} \subseteq \mathbb{C}^{ms\times ms}$.

Theorem 4.10: Let $t \geq 0$. Suppose that A is block positive real, and that $\mathcal{H}_m + \mathcal{M}$ is normal. Then the residual for the shifted BGMRES approximation satisfies

$$\|\boldsymbol{R}_m(t)\|_{\mathbb{S}} \leq \|\boldsymbol{R}_m(0)\|_{\mathbb{S}}\,. \tag{4.38}$$

Proof: We first note that since A is BPR, $\mathcal{H}_m$ has spectrum with positive real part, by Lemmas 3.15 and 2.7 (v). Consequently, $\mathcal{H}_m$ is invertible and the quantities $\boldsymbol{U}$, $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{S}_t$ from (4.37) are well defined.

We now reduce $\|\boldsymbol{R}_m(t)\|_{\mathbb{S}}^2$ to an equivalent quantity whose derivative is easy to take. Noting that $G_m(t)$ from equation (4.27) can be expressed as $G_m(t) = \boldsymbol{U}^* \mathcal{S}_t^{-1} \widehat{\boldsymbol{E}}_1 B$, we have that

$$\begin{aligned}
\|\boldsymbol{R}_m(t)\|_{\mathbb{S}}^2 &= \operatorname{trace}\left(\left(\begin{bmatrix}\boldsymbol{U}\\ -I\end{bmatrix} G_m(t)\right)^* \begin{bmatrix}\boldsymbol{U}\\ -I\end{bmatrix} G_m(t)\right) \\
&= \operatorname{trace}\left(G_m(t)^*(I + \boldsymbol{U}^*\boldsymbol{U})G(t)\right) \\
&= \operatorname{trace}\left(B^* \widehat{\boldsymbol{E}}_1^* \mathcal{S}_t^{-*} \boldsymbol{U}(I + \boldsymbol{U}^*\boldsymbol{U})\boldsymbol{U}^* \mathcal{S}_t^{-1} \widehat{\boldsymbol{E}}_1 B\right) \\
&= \operatorname{trace}\left(B^* \widehat{\boldsymbol{E}}_1^* \mathcal{S}_t^{-*} \mathcal{T} \mathcal{S}_t^{-1} \widehat{\boldsymbol{E}}_1 B\right) \\
&= \operatorname{trace}\left(B^* \widehat{\boldsymbol{E}}_1^* \mathcal{R}_t^* \mathcal{R}_t \widehat{\boldsymbol{E}}_1 B\right),
\end{aligned}$$

where $\mathcal{R}_t := \mathcal{T}^{1/2} \mathcal{S}_t^{-1}$. Applying Lemma 2.25, we take the derivative of $\|\boldsymbol{R}_m(t)\|_{\mathbb{S}}^2$ with respect to t. In terms of $\mathcal{R}_t$, this is

$$\begin{aligned}
\frac{d}{dt}\left[\|\boldsymbol{R}_m(t)\|_{\mathbb{S}}^2\right] &= \operatorname{trace}\left(B^* \widehat{\boldsymbol{E}}_1^* \left(\frac{d}{dt}[\mathcal{R}_t]^* \mathcal{R}_t + \mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t]\right) \widehat{\boldsymbol{E}}_1 B\right) \\
&= \operatorname{trace}\left((\widehat{\boldsymbol{E}}_1 B)^* \frac{d}{dt}[\mathcal{R}_t]^* \mathcal{R}_t \widehat{\boldsymbol{E}}_1 B + (\widehat{\boldsymbol{E}}_1 B)^* \mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t] \widehat{\boldsymbol{E}}_1 B\right). \qquad (4.39)
\end{aligned}$$

Our end goal is to show that the argument of the trace of (4.39) is seminegative real for all $t \geq 0$. The trace is of the form $B + B^*$, and $\operatorname{trace}(B + B^*) = 2\sum_{\lambda \in \operatorname{spec}(B)} \operatorname{Re}(\lambda)$, so it is enough to show that $(\widehat{\boldsymbol{E}}_1 B)^* \mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t] \widehat{\boldsymbol{E}}_1 B$ has spectrum with nonpositive real part. In fact, by Lemma (2.7)(iv), it suffices to show that $-\mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t]$ is Euclidean nonnegative real. We proceed by

1. computing $\frac{d}{dt}[\mathcal{R}_t]$, and
2. examining spectral properties of $-\mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t]$.

1. Further application of Lemma 2.25 implies that $\frac{d}{dt}[\mathcal{R}_t] = \mathcal{T}^{1/2} \frac{d}{dt}\left[\mathcal{S}_t^{-1}\right]$ and $\frac{d}{dt}[\mathcal{S}_t] = \mathcal{H}_m^{-1}$, and part (iv) in particular of this lemma leads to

$$\frac{d}{dt}\left[\mathcal{S}_t^{-1}\right] = -\mathcal{S}_t^{-1} \frac{d}{dt}[\mathcal{S}_t] \mathcal{S}_t^{-1} = -\mathcal{S}_t^{-1} \mathcal{H}_m^{-1} \mathcal{S}_t^{-1} = \mathcal{S}_t^{-1} (\mathcal{S}_t \mathcal{H}_m)^{-1}.$$

Thus, $\frac{d}{dt}[\mathcal{R}_t] = -\mathcal{T}^{1/2} \mathcal{S}_t^{-1} (\mathcal{S}_t \mathcal{H}_m)^{-1}$.

2. By step 1, we have that $-\mathcal{R}_t^* \frac{d}{dt}[\mathcal{R}_t] = \mathcal{S}_t^{-*} \mathcal{T} \mathcal{S}_t^{-1} (\mathcal{S}_t \mathcal{H}_m)^{-1}$. Since $\mathcal{T}$ is HPSD, Lemma 2.7(i) implies that $\mathcal{S}_t^{-*} \mathcal{T} \mathcal{S}_t^{-1}$ is also HPSD. Additionally,

$$(\mathcal{S}_t \mathcal{H}_m)^{-1} = (\mathcal{S} \mathcal{H}_m + tI)^{-1} = (\mathcal{H}_m + \mathcal{M} + tI)^{-1},$$

which has positive real spectrum by the hypothesis that A is BPR and Lemma 4.6. Furthermore, since $\mathcal{H}_m + \mathcal{M}$ is normal, then so are $\mathcal{H}_m + \mathcal{M} + tI$ and its inverse. By Lemma 2.7 (vii), $(\mathcal{H}_m + \mathcal{M} + tI)^{-1}$ is also EPR. Finally, we can then apply part vi to $\mathcal{S}_t^{-*} \mathcal{T} \mathcal{S}_t^{-1} (\mathcal{S}_t \mathcal{H}_m)^{-1}$ to conclude that it has spectrum with nonnegative real part. $\square$

Remark 4.11: Determining whether $\mathcal{H}_m + \mathcal{M}$ is normal is not convenient in practice, but the quantity is at least a "naturally occurring" one in BGMRES. Although

a more general version of Theorem 4.10 remains an open problem, the result appears to hold in many practical scenarios (see Section 7.2.2).

Theorem 4.10 extends to the restarted case, thanks to Lemmas 3.15, 2.7(v), and 4.6, which ensure that both $\mathcal{H}_m^{(k)}$ and $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$ have spectrum with positive real part.

Corollary 4.12: Let $t \geq 0$ and suppose that A is block positive real, and that $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$ is normal. Then the residual for the restarted and shifted BGMRES approximation satisfies

$$\left\| \boldsymbol{R}_m^{(k)}(t) \right\|_{\mathbb{S}} \leq \left\| \boldsymbol{R}_m^{(k)}(0) \right\|_{\mathbb{S}}.$$

To translate the residual results Theorem 4.10 and Corollary 4.12 into error bounds, we need a way to go between the shifted and non-shifted norms. The proof is a generalization of [52, Lemma 6.4], as long as one replaces the standard Euclidean inner product and 2-norm with $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ and $\| \cdot \|_{\mathbb{S}}$, respectively.

Lemma 4.13: Let $A \in \mathbb{C}^{n \times n}$ be block positive real and $t \geq 0$. For all $\boldsymbol{V} \in \mathbb{C}^{n \times s}$,

$$\|\boldsymbol{V}\|_{A^*A\text{-}\mathbb{S}}^2 \leq \frac{\nu_{\max}}{(t + \rho\nu_{\max})^2} \|\boldsymbol{V}\|_{(A+tI)^*(A+tI)\text{-}\mathbb{S}}^2.$$

Theorem 4.14: Let $t \geq 0$ and suppose that A is block positive real, and suppose that the hypotheses of Corollary 4.12. Then the error $\boldsymbol{E}_m^{(k)}(t) = \boldsymbol{X}^*(t) - \boldsymbol{X}_m^{(k)}(t)$ of the restarted BGMRES approximation to the shifted system $(A + tI)\boldsymbol{X}(t) = \boldsymbol{B}$ can be bounded as

$$\left\| \boldsymbol{E}_m^{(k)}(t) \right\|_{A^*A\text{-}\mathbb{S}} \leq \sqrt{\frac{\nu_{\max}}{(t + \rho\nu_{\max})^2}} \left(1 - \frac{\gamma^2}{\nu_{\max}} \right)^{mk/2} \|\boldsymbol{B}\|_{\mathbb{S}}, \tag{4.40}$$

where ρ, γ, and $\nu_{\max}$ are as in (4.34).

Proof: By Lemma 4.13,

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A^*A\text{-}\mathbb{S}}^2 \leq \frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{(A+tI)^*(A+tI)\text{-}\mathbb{S}}^2$$
$$= \frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathbb{S}}^2. \tag{4.41}$$

Corollaries 4.8 and 4.12 give the desired result. □

Remark 4.15: The bound from Theorem 4.14 can be made independent of t by noting that $\frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}$ is monotonically decreasing and therefore bounded by

$$\frac{\nu_{\max}}{(\rho\nu_{\max})^2} = \frac{1}{\rho^2\nu_{\max}}\ .$$

4.2.5 A matrix polynomial approach

Several authors consider a matrix polynomial form of the BGMRES residual. Simoncini and Gallopoulos [127, 129, 130] describe properties of the classical BGMRES residual polynomial and use it to accelerate the convergence of BGMRES; Elbouyahyaoui, Messaoudi, and Sadok [46] derive an explicit expression for the global BGMRES residual polynomial. These polynomials expose underlying behavior of the BGMRES approximation encapsulated in their latent roots. We also use them to provide an alternative way to obtain the error bound (4.40).

For now, we consider only the first cycle and discard the cycle superscripts. Recall from Lemma 4.5 that

$$\boldsymbol{R}_m(t) = \boldsymbol{\mathcal{V}}_{m+1}\begin{bmatrix}\boldsymbol{U}\\ -I\end{bmatrix}G_m(t),$$

where $G_m(t)$ is defined in equation (4.27). Assuming that $G_m(0)$ is invertible, we find that

$$\boldsymbol{R}_m(t) = \boldsymbol{R}_m(0)C_m(t) \text{ with } C_m(t) := G_m(0)^{-1}G_m(t). \tag{4.42}$$

Further assume that there exists a family of matrix polynomials $Q_{m-1,t} \in \mathbb{P}_{m-1}(\mathbb{S})$ interpolating z^{-1} on $(\mathcal{H}_m + \mathcal{M} + tI, \widehat{\boldsymbol{E}}_1 B)$, i.e., such that

$$Q_{m-1,t}(\mathcal{H}_m + \mathcal{M} + tI) \circ \widehat{\boldsymbol{E}}_1 B = \boldsymbol{Y}_m(t).$$

Define the family $P_{m,t} \in \mathbb{P}_m(\mathbb{S})$ as

$$P_{m,t}(z) := I - zQ_{m-1,t}(z).$$

Note that

$$\begin{aligned}
\boldsymbol{R}_m(t) &= \boldsymbol{B} - (A + tI)\boldsymbol{\mathcal{V}}_m Q_{m-1,t}(\mathcal{H}_m + \mathcal{M} + tI) \circ \widehat{\boldsymbol{E}}_1 B \\
&= \boldsymbol{B} - (A + tI)Q_{m-1,t}(A + tI) \circ \boldsymbol{B}, \text{ by Theorem 4.1} \\
&= P_{m,t}(A + tI) \circ \boldsymbol{B}.
\end{aligned} \tag{4.43}$$

In fact, $C_m(t)$ can be written directly in terms of $P_{m,t}(z)$, if some additional assumptions are fulfilled.

Theorem 4.16: Suppose that $\boldsymbol{R}_m(0)$ has full rank and that $\mathcal{H}_m + \mathcal{M}$ has a block eigendecomposition $\mathcal{H}_m + \mathcal{M} = \mathcal{U}\mathcal{T}\mathcal{U}^{-1}$, where $\mathcal{U} \in \mathbb{S}^{m\times m}$ is invertible, $\mathcal{T} = \operatorname{diag}(\{\Theta_j\}_{j=1}^m) \in \mathbb{S}^{m\times m}$, and $\{\Theta_j\}_{j=1}^m \subset \mathbb{S}$ are diagonalizable and the block eigenvalues of $\mathcal{H}_m + \mathcal{M}$. Let $\boldsymbol{V} \in \mathbb{S}^m, \boldsymbol{V} \neq 0$, be such that each block entry W_j of $\boldsymbol{W} := [W_1 \cdots W_m] := \mathcal{U}^{-1}\boldsymbol{V}$ is invertible. Noting that $\{\Theta_j + tI\}_{j=1}^m$ are the block eigenvalues of $\mathcal{H}_m + \mathcal{M} + tI$, define $S_j := W_j^{-1}\Theta_j W_j$, for all $j = 1, \ldots, m$. Assume

that $S_i - S_j$ are nonsingular for all $i, j = 1, \ldots, m-1$, $i \neq j$. With $Q_{m-1,t} \in \mathbb{P}_{m-1}(\mathbb{S})$ interpolating z^{-1} on $(\mathcal{H}_m + \mathcal{M} + tI, \boldsymbol{V})$ and $P_{m,t}(z) := I - zQ_{m-1,t}(z)$, it holds that

(i) the latent roots of $P_{m,t}$ match the eigenvalues of $\mathcal{H}_m + \mathcal{M} + tI$;

(ii) $P_{m,t}(z) = P_{m,0}(z-t)P_{m,0}(-t)^{-1}$; and

(iii) $C_m(t) = P_{m,0}(-t)^{-1}$, where $C_m(t)$ is defined in equation (4.42).

If in addition, $\mathcal{H}_m + \mathcal{M}$ is has spectrum with positive real part, $t \geq 0$, and each $\widetilde{S}_j := (S_1 \cdots S_{j-1})S_j(S_1 \cdots S_{j-1})^{-1}$ is positive real, then

$$\lambda_{\max}(C_m(t)C_m(t)^*) \leq 1,$$

where $\lambda_{\max}(A)$ denotes the maximum eigenvalue of the HPD matrix A.

Proof: By part (i) of Theorem 2.24, each $S_j + tI$ is a solvent of $P_{m,t}(z)$. By part (ii) of the same theorem, the latent roots of each $P_{m,t}$ match the eigenvalues of $\mathcal{H}_m + \mathcal{M} + tI$. Combining Theorem 2.19 and the fact that $P_{m,t}(0) = I$, we obtain that

$$\begin{aligned} P_{m,t}(z) &= \big(zI - S_1 - tI\big) \cdots \big(zI - S_m - tI\big)(-1)^m \big((S_1 + tI) \cdots (S_m + tI)\big)^{-1} \\ &= \big((z-t)I - S_1\big) \cdots \big((z-t)I - S_m\big)(-1)^m \big((S_1 + tI) \cdots (S_m + tI)\big)^{-1} \end{aligned}$$

Then

$$\begin{aligned} P_{m,0}(z-t) &= \big((z-t-0)I - S_1\big) \cdots \big((z-t-0)I - S_m\big)(-1)^m (S_1 \cdots S_m)^{-1} \\ &= \big((z-t)I - S_1\big) \cdots \big((z-t)I - S_m\big)(-1)^m (S_1 \cdots S_m)^{-1}, \end{aligned}$$

and

$$\begin{aligned} P_{m,0}(-t) &= \big((-t-0)I - S_1\big)\cdots\big((-t-0)I - S_m\big)(-1)^m(S_1\cdots S_m)^{-1} \\ &= (S_1 + tI)\cdots(S_m + tI)(S_1\cdots S_m)^{-1}. \end{aligned}$$

Multiplying $P_{m,0}(z-t)$ by $P_{m,0}(-t)^{-1}$ gives the desired result for part (ii). Part (iii) follows by noting that

$$\begin{aligned} \boldsymbol{R}_m(0)C_m(t) = \boldsymbol{R}_m(t) &= P_{m,t}(A + tI) \circ \boldsymbol{B} \\ &= (P_{m,0}(A)P_{m,0}(-t)^{-1}) \circ \boldsymbol{B} = (P_{m,0}(A) \circ \boldsymbol{B})P_{m,0}(-t)^{-1} \\ &= \boldsymbol{R}_m(0)P_{m,0}(-t)^{-1}. \end{aligned}$$

Since $\boldsymbol{R}_m(0)$ has full rank, $P_{m,0}(-t)^{-1} = C_m(t)$.

For the final part with additional assumptions, first note that by part (iii) of this theorem and part (iii) of Theorem 2.24,

$$C_m(t) = (I + t\widetilde{S}_m^{-1})^{-1}\cdots(I + t\widetilde{S}_1^{-1})^{-1}.$$

Then

$$\lambda_{\max}(C_m(t)C_m(t)^*) = \|C_m(t)\|_2^2 \le \left\|(I + t\widetilde{S}_m^{-1})^{-1}\right\|_2^2 \cdots \left\|(I + t\widetilde{S}_1^{-1})^{-1}\right\|_2^2. \tag{4.44}$$

Note that for each $j = 1, \ldots, m$,

$$\begin{aligned} \left\|(I + t\widetilde{S}_j^{-1})^{-1}\right\|_2 &= \lambda_{\max}\left(\big((I + t\widetilde{S}_j^{-1})^{-1}(I + t\widetilde{S}_j^{-1})^{-*}\big)^{-1}\right) \\ &= \lambda_{\max}\left(\big(I + t(\widetilde{S}_j^{-*} + \widetilde{S}_j^{-1}) + t^2\widetilde{S}_j^{-*}\widetilde{S}_j^{-1}\big)^{-1}\right). \end{aligned}$$

Since $\widetilde{S}_j$ is positive real, then so are its inverse and transpose, and by Lemma 2.6 ii, $\widetilde{S}_j^{-*} + \widetilde{S}_j^{-1}$ is HPD. The product $\widetilde{S}_j^{-*}\widetilde{S}_j^{-1}$ is also HPD, so $\big((I+t\widetilde{S}_j^{-1})^{-1}(I+t\widetilde{S}_j^{-1})^{-*}\big)^{-1}$

is HPD and of the form $I + tD$ for an HPD matrix D. Since $t \geq 0$, the spectrum of such a matrix is bounded by 1 from below. Consequently,

$$\left\| (I + t\widetilde{S}_j^{-1})^{-1} \right\|_2 \leq 1, \text{ for all } j = 1, \ldots, m. \qquad \square$$

Part(i) of Theorem 4.16 is proven for the classical case in [130, Theorem 3.3] and for the global case in [46, Theorem 3.1].

Remark 4.17: The eigenvalues of $\mathcal{H}_m$ are called the *Ritz values of A*. When $s = 1$, the eigenvalues of $\mathcal{H}_m + \mathcal{M}$ are referred to as the *harmonic Ritz values of A*, since they turn out to be weighted harmonic means of the eigenvalues of A [109]. We adopt the same nomenclature in the block case, noting that the latent roots of the BGMRES residual polynomial give us harmonic Ritz values.

Under similar assumptions, Theorem 4.16 also holds for restarts. Define the restarted cospatial factors

$$C_m^{(k)}(t) := \left(G_m^{(k)}(0)\right)^{-1} G_m^{(k)}(t).$$

Corollary 4.18: Fix $k \geq 1$, and denote $\widetilde{\mathcal{H}}_m^{(\ell)} = \mathcal{H}_m^{(\ell)} + \mathcal{M}^{(\ell)}$, $\ell = 1, \ldots, k$. If each $\widetilde{\mathcal{H}}_m^{(\ell)}$ has spectrum with positive real part and meets the hypotheses of Theorem 4.16, then for all $t \geq 0$,

$$\left\| C_m^{(k)}(t) \right\|_2 \leq 1.$$

For the culminating result, we require the following bound on the trace of a matrix product.

Lemma 4.19: Let $X, Y \in \mathbb{R}^{s\times s}$ be symmetric matrices, and Y nonnegative definite. Then

$$\operatorname{trace}(XY) \le \lambda_{\max}(X) \cdot \operatorname{trace}(Y).$$

Proof: See, e.g., [138, Lemma 1]. □

Theorem 4.20: Let $A \in \mathbb{C}^{n\times n}$ be block positive real. Suppose further that the assumptions of Theorem 2.24 hold on $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$, so that the conclusion of Corollary 4.18 holds. Then

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A^*A\text{-}\mathbb{S}} \le \sqrt{\frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}}\left(1-\frac{\gamma^2}{\nu_{\max}}\right)^{mk/2}\|\boldsymbol{B}\|_{\mathbb{S}},$$

where ρ, γ, and $\nu_{\max}$ are as in (4.34).

Proof: Again Lemma 4.13 gives that

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A^*A\text{-}\mathbb{S}}^2 = \frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathbb{S}}^2.$$

Recalling that $\boldsymbol{R}_m^{(k)}(t) = \boldsymbol{R}_m^{(k)}(0)C_m^{(k)}(t)$, and by employing Lemma 4.19,

$$\begin{aligned}
\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathbb{S}}^2 &= \operatorname{trace}\left(\langle\!\langle \boldsymbol{R}_m^{(k)}(0)C_m^{(k)}(t), \boldsymbol{R}_m^{(k)}(0)C_m^{(k)}(t)\rangle\!\rangle_{\mathbb{S}}\right)\\
&= \operatorname{trace}\left(C_m^{(k)}(t)^*\langle\!\langle \boldsymbol{R}_m^{(k)}(0), \boldsymbol{R}_m^{(k)}(0)\rangle\!\rangle_{\mathbb{S}}C_m^{(k)}(t)\right)\\
&= \operatorname{trace}\left(\langle\!\langle \boldsymbol{R}_m^{(k)}(0), \boldsymbol{R}_m^{(k)}(0)\rangle\!\rangle_{\mathbb{S}}C_m^{(k)}(t)C_m^{(k)}(t)^*\right)\\
&\le \left\|\boldsymbol{R}_m^{(k)}(0)\right\|_{\mathbb{S}}^2 \lambda_{\max}(C_m^{(k)}(t)C_m^{(k)}(t)^*).
\end{aligned}$$

Corollaries 4.8 and 4.18 finish the proof. □

4.3 Block Radau-Lanczos

We have already established the need for restarts when there is a limitation on the number of Krylov basis vectors we can store. Restarts can significantly slow down convergence, however. Techniques such as eigenvalue deflation [102] can reduce the number of cycles needed to converge, and in this section, we propose yet another alternative called the *block Radau-Lanczos* (BRL) method, which is designed particularly for operators $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$ that are $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-SA and $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-PD.

The Radau-Lanczos (RL) method for $s = 1$ is presented in [54], along with a CG-like convergence bound and an in-depth numerical analysis. The main idea behind the method is to prescribe a Ritz value for every cycle, typically one that is slightly larger than the maximum eigenvalue of A. Given that BGMRES itself is equivalent to modifying the Ritz values via an s-rank modification (see Remark 4.17), it is reasonable to consider other Ritz-value-modifying methods and see if they can also be cast as an s-rank modification. We describe the Radau-Lanczos process in the case of blocks, where we now prescribe a block eigenvalue from $\mathbb{S}$ (and consequently s Ritz values). The reader can review Section 2.4.2 for the definition of block eigenvalues.

Since A is $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-SA and $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$-PD, $\mathcal{H}_m$ is $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\boldsymbol{\mathcal{V}}_m}$-SA and $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\boldsymbol{\mathcal{V}}_m}$-PD by Lemma 3.15 and also block tridiagonal, i.e., of the following form,

$$\mathcal{H}_m = \begin{bmatrix} H_{1,1} & H_{2,1}^* & & & & \\ H_{2,1} & H_{2,2} & H_{3,2}^* & & & \\ & \ddots & \ddots & \ddots & & \\ & & H_{m-1,m-2} & H_{m-1,m-1} & H_{m,m-1}^* & \\ & & & H_{m,m-1} & H_{m,m} & \end{bmatrix}$$
$$= \begin{bmatrix} \mathcal{H}_{m-1} & \widehat{\boldsymbol{E}}_{m-1} H_{m,m-1}^* \\ H_{m,m-1} \widehat{\boldsymbol{E}}_{m-1}^* & H_{m,m} \end{bmatrix}. \tag{4.45}$$

We choose a Hermitian $S_0 \in \mathbb{S}$ to be the matrix whose s eigenvalues we want to impose upon the spectrum of $\mathcal{H}_m$. Letting $\widehat{\boldsymbol{E}}_{m-1} \in \mathbb{S}^{m-1}$ and $I_{m-1} \in \mathbb{C}^{(m-1)\times(m-1)}$, we then define

$$\boldsymbol{D} := (\mathcal{H}_{m-1} - I_{m-1} \otimes S_0)^{-1} \widehat{\boldsymbol{E}}_{m-1} \in \mathbb{S}^{m-1}, \tag{4.46}$$

where $D_{m-1} = \widehat{\boldsymbol{E}}_{m-1}^* \boldsymbol{D}$, i.e., the last block entry of $\boldsymbol{D}$. Then the *block Radau-Lanczos (BRL) modification* is given as

$$\mathcal{M}_{\texttt{rad}} := \boldsymbol{E}_m (S_0 + H_{m,m-1} D_{m-1}^* H_{m,m-1}^* - H_{m,m}) \boldsymbol{E}_m^*. \tag{4.47}$$

It has already been shown why the modification (4.47) fixes eigenvalues when $\mathbb{S} = \mathbb{C}$ and $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ is the usual Euclidean inner product in [54]. The explanation in the block case is similar, but technically more challenging, as one must generalize block Gauss quadrature within the framework of [55] and ensure that certain non-commutative operations are tracked correctly. Furthermore, some additional assumptions must be placed on the spectrum of A. To see why $\mathcal{M}_{\texttt{rad}}$ does what we want, we must take a detour into block Gauss quadrature rules.

4.3.1 Block Gauss quadrature

Block quadrature rules are not new. One can find a nice introduction to them in [63] for approximating expressions of the form $\boldsymbol{B}^* f(A)\boldsymbol{B}$, where $s = 2$; they are extended to general s in [116]. While we are not necessarily concerned with computing $\boldsymbol{B}^* f(A)\boldsymbol{B}$, it is related to the computation of $f(A)\boldsymbol{B}$, and it is precisely this connection that we want to exploit. However, to remain within our framework, we instead consider computing $\langle\!\langle \boldsymbol{B}, f(A)\boldsymbol{B} \rangle\!\rangle_{\mathbb{S}}$. Suppose that A has a block eigendecomposition in the sense of Theorem 2.14, i.e., that s divides n and

$$A\boldsymbol{Q}_j = \boldsymbol{Q}_j \Lambda_j, \text{ where } \boldsymbol{Q}_j \in \mathbb{C}^{n\times s}, \Lambda_j \in \mathbb{S}, \text{ and } \langle\!\langle \boldsymbol{Q}_j, \boldsymbol{Q}_k \rangle\!\rangle_{\mathbb{S}} = \delta_{jk} I_s,$$

for all $j = 1, \ldots, m = \frac{n}{s}$, with δ_{jk} denoting the Kronecker delta function. When $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$ is the classical block inner product, it is possible to choose $(\Lambda_j, \boldsymbol{Q}_j)$ so that $\boldsymbol{Q}_j = [\boldsymbol{q}_j | \cdots | \boldsymbol{q}_j]$ and $\Lambda_j = \lambda_j I_s$, where $(\lambda_j, \boldsymbol{q}_j) \in \mathbb{C} \times \mathbb{C}^n$ is an eigenpair of A, and $\langle \boldsymbol{q}_j, \boldsymbol{q}_k \rangle_2 = \delta_{jk}$. For other block inner products, however, we may not have such a convenient relationship.

Since the block eigenvectors of A are block orthonormal, there exists some $\{C_j\}_{j=1}^n \subset \mathbb{S}$ such that

$$\boldsymbol{B} = \sum_{j=1}^{n} \boldsymbol{Q}_j C_j,$$

which allows us to rewrite $\langle\!\langle \boldsymbol{B}, f(A)\boldsymbol{B} \rangle\!\rangle_{\mathbb{S}}$ as

$$\begin{aligned}
\langle\!\langle \boldsymbol{B}, f(A)\boldsymbol{B} \rangle\!\rangle_{\mathbb{S}} &= \left\langle\!\!\left\langle \sum_{j=1}^{n} \boldsymbol{Q}_j C_j, f(A) \sum_{k=1}^{n} \boldsymbol{Q}_k C_k \right\rangle\!\!\right\rangle_{\mathbb{S}} \\
&= \sum_{j=1}^{n}\sum_{k=1}^{n} \langle\!\langle \boldsymbol{Q}_j C_j, f(A)\boldsymbol{Q}_k C_k \rangle\!\rangle_{\mathbb{S}} \\
&= \sum_{j=1}^{n}\sum_{k=1}^{n} C_j^* \langle\!\langle \boldsymbol{Q}_j, \boldsymbol{Q}_k \rangle\!\rangle_{\mathbb{S}} f(\Lambda_k) C_k, \text{ by Theorem 2.15} \qquad (4.48) \\
&= \sum_{j=1}^{n} C_j^* f(\Lambda_j) C_j. \qquad (4.49)
\end{aligned}$$

Since A is BSA and BPD, each Λ_j is HPD, so there exist $U_j, D_j \in \mathbb{C}^{s\times s}$ such that $\Lambda_j = U_j D_j U_j^*$, where $D_j = \operatorname{diag}(d_{j1}, \ldots, d_{js})$. Then equation (4.49) becomes

$$\sum_{j=1}^{n}\sum_{i=1}^{s} f(d_{ji}) \boldsymbol{w}_{ji} \boldsymbol{w}_{ji}^*, \qquad (4.50)$$

where

$$\boldsymbol{w}_{ji} := \widehat{\boldsymbol{e}}_i^* C_j^* U_j.$$

Let $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum eigenvalues of A, respectively, and let $H : \mathbb{R} \to \mathbb{R}$ denote the Heaviside step function

$$H(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}.$$

From equation (4.50), we define a matrix of measures $\mathrm{d}\alpha(t)$ based on

$$\alpha(t) := \sum_{j=1}^{n}\sum_{i=1}^{s} H(t - d_{ji}) \boldsymbol{w}_{ji} \boldsymbol{w}_{ji}^*, \qquad (4.51)$$

and a bilinear form on pairs of matrix polynomials P, Q taking coefficients in $\mathbb{S}$:

$$(P, Q) := \int_{\lambda_{\min}}^{\lambda_{\max}} P^* Q \, \mathrm{d}\alpha(t). \qquad (4.52)$$

Remark 4.21: Line (4.48) in the preceding discussion merits a few comments. We are making use of the fact that each $C_k \in \mathbb{S}$, and that, due to the scalar polynomial definition of f and that $\Lambda_k \in \mathbb{S}$, $f(\Lambda_k)$ is an element of $\mathbb{S}$ as well. These observations are key, because only elements of $\mathbb{S}$ can move in and out of $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$.

The bilinear form (4.52) gives rise to a set of orthogonal matrix polynomials $\{P_j\}_{j=-1}^{n}$ with a three-term recurrence, given by

$$zP_{j-1}(z) = P_j(z)\Gamma_j + P_{j-1}(z)\Omega_j + P_{j-2}(z)\Gamma_{j-1}^*, \tag{4.53}$$

$$P_0(z) \equiv I_s, \quad P_{-1}(z) \equiv 0_s.$$

See, e.g., [25, 68] for more information on orthogonal matrix polynomials. Defining

$$\boldsymbol{P}_m(z) := \begin{bmatrix} P_0(z)^* \\ \vdots \\ P_{m-1}(z)^* \end{bmatrix},$$

and

$$\mathcal{J}_m := \begin{bmatrix} \Omega_1 & \Gamma_1^* & & & \\ \Gamma_1 & \Omega_2 & \Gamma_2^* & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{m-2} & \Omega_{m-1} & \Gamma_{m-1}^* \\ & & & \Gamma_{m-1} & \Omega_m \end{bmatrix},$$

the three-term recurrence (4.53) can be rewritten as

$$\mathcal{J}_m \boldsymbol{P}_m(z) = z\boldsymbol{P}_m(z) - \begin{bmatrix} 0_s \\ \vdots \\ 0_s \\ \Gamma_m^* P_m(z)^* \end{bmatrix}. \tag{4.54}$$

It is not difficult to show that due to the definition of α and the bilinear form (4.52), $\mathcal{J}_m = \mathcal{H}_m$.

4.3.2 Block Gauss-Radau quadrature

As mentioned earlier, we wish to alter the last block entry of $\mathcal{H}_m$ in order to change s of its eigenvalues to those of some $S_0 = S_0^* \in \mathbb{S}$. This is equivalent to generating a new polynomial $\widetilde{P}_m$ such that $\widetilde{P}_m(S_0) = 0_s$. By equation (4.54), we have that

$$\widetilde{\mathcal{J}}_m \begin{bmatrix} \boldsymbol{P}_{m-1}(S_0) \\ 0_s \end{bmatrix} = (I_m \otimes S_0) \begin{bmatrix} \boldsymbol{P}_{m-1}(S_0) \\ 0_s \end{bmatrix}, \tag{4.55}$$

where

$$\widetilde{\mathcal{J}}_m := \begin{bmatrix} \mathcal{J}_{m-1} & \widehat{\boldsymbol{E}}_{m-1}\Gamma_{m-1}^* \\ \Gamma_{j-1}\widehat{\boldsymbol{E}}_{m-1}^* & \widetilde{\Omega}_m \end{bmatrix}.$$

Then (perhaps seen more easily from the recurrence (4.53))

$$S_0 P_{m-1}(S_0) = P_{m-1}(S_0)\widetilde{\Omega}_m + P_{m-2}(S_0)\Gamma_{m-1}^*;$$

and, assuming that $P_{m-1}(S_0)$ is invertible,

$$\widetilde{\Omega}_m = P_{m-1}(S_0)^{-1} S_0 P_{m-1}(S_0) - P_{m-1}(S_0)^{-1} P_{m-2}(S_0)\Gamma_{m-1}^*.$$

The trick now is to solve for each term using only $\mathcal{J}_m$ and its entries, and S_0. We solve for the second term first. Equation (4.55) reduces to the following, which can also be seen by looking at equation (4.54) for $m-1$:

$$\mathcal{J}_{m-1} \begin{bmatrix} P_0(S_0)^* \\ \vdots \\ P_{m-2}(S_0)^* \end{bmatrix} = (I_{m-1} \otimes S_0) \begin{bmatrix} P_0(S_0)^* \\ \vdots \\ P_{m-2}(S_0)^* \end{bmatrix} - \widehat{\boldsymbol{E}}_{m-1}\Gamma_{m-1}^* P_{m-1}(S_0)^*.$$

Right-multiplying by $-P_{m-1}(S_0)^{-*}$ and combining both terms multiplied by matrices results in that

$$(\mathcal{J}_m - I_{m-1} \otimes S_0)\boldsymbol{D} = \widehat{\boldsymbol{E}}_{m-1}\Gamma_{m-1}^*,$$

with

$$\boldsymbol{D} := -\begin{bmatrix} P_0(S_0)^* P_{m-1}(S_0)^{-*} \\ \vdots \\ P_{m-2}(S_0)^* P_{m-1}(S_0)^{-*} \end{bmatrix}.$$

Noting that $\widehat{\boldsymbol{E}}_{m-1}^* \boldsymbol{D} = -P_{m-2}(S_0)^* P_{m-1}(S_0)^{-*}$ means all we need to do is solve for $\boldsymbol{D}$ to find the second term in $\widetilde{\Omega}_m$. Indeed, this is precisely the $\boldsymbol{D}$ in the (4.46). If $S_0 = \theta_0 I_s$, then the first term in $\widetilde{\Omega}_m$ remains S_0. When S_0 takes any other form, however, then we need to compute $\widetilde{S}_0 := P_{m-1}(S_0)^{-1} S_0 P_{m-1}(S_0)$. Working with the unmodified polynomials and matrix $\mathcal{J}_m$, we have from the relation (4.53) that

$$S_0 P_{m-1}(S_0) = P_m(S_0)\Gamma_m + P_{m-1}(S_0)\Omega_m + P_{m-2}(S_0)\Gamma_{m-1}^*,$$

implying that (and since $\Omega_m = \Omega_m^*$)

$$\Gamma_m^* P_m(S_0)^* = P_{m-1}(S_0)^* S_0^* - \Omega_m P_{m-1}(S_0)^* - \Gamma_{m-1} P_{m-2}(S_0)^*.$$

Then, making use of equation (4.54),

$$\begin{aligned} \mathcal{J}_m \begin{bmatrix} P_0(S_0)^* \\ \vdots \\ P_{m-2}(S_0)^* \\ P_{m-1}(S_0)^* \end{bmatrix} =& (I_m \otimes S_0) \begin{bmatrix} P_0(S_0)^* \\ \vdots \\ P_{m-2}(S_0)^* \\ P_{m-1}(S_0)^* \end{bmatrix} \\ & - \widehat{\boldsymbol{E}}_m (P_{m-1}(S_0)^* S_0^* - \Omega_m P_{m-1}(S_0)^* - \Gamma_{m-1} P_{m-2}(S_0)^*). \end{aligned}$$

Right-multiplying by $-P_{m-1}(S_0)^{-*}$ and rearranging terms gives that

$$(\mathcal{J}_m - I_m \otimes S_0) \begin{bmatrix} \boldsymbol{D} \\ -I \end{bmatrix} = \widehat{\boldsymbol{E}}_m (\widetilde{S}_0^* - \Omega_m + \Gamma_{m-1} D_{m-1}). \tag{4.56}$$

Simplifying the left-hand side of equation (4.56), we find

$$(\mathcal{J}_m - I_m \otimes S_0)\begin{bmatrix}\boldsymbol{D}\\ -I\end{bmatrix} = \begin{bmatrix}(\mathcal{J}_{m-1} - I_{m-1}\otimes S_0)\boldsymbol{D} - \widehat{\boldsymbol{E}}_{m-1}\Gamma_{m-1}^*\\ \Gamma_{m-1}D_{m-1} - \Omega_m + S_0\end{bmatrix}$$
$$= \widehat{\boldsymbol{E}}_m(\Gamma_{m-1}D_{m-1} - \Omega_m + S_0),$$

but by comparing with the right-hand side of equation (4.56), it finally follows that $S_0 = \widetilde{S}_0^*$. Since $S_0 = S_0^*$, we can conclude that

$$\widetilde{\Omega}_m = S_0 + D_{m-1}^*\Gamma_{m-1}^*. \tag{4.57}$$

In other words,

$$\widetilde{\mathcal{J}}_m = \mathcal{J}_m + \widehat{\boldsymbol{E}}_m(S_0 + D_{m-1}^*\Gamma_{m-1}^* - \Omega_m)\widehat{\boldsymbol{E}}_m^*.$$

Replacing $\mathcal{J}_m$ with $\mathcal{H}_m$ gives the expressions in equations (4.46) and (4.47).

4.3.3 Block Radau-Lanczos as a solver

The RL method for $s = 1$ is studied for Stieltjes functions of matrices in [54] and compared with a FOM-based method for matrix functions. (See Chapter 5 for how FOM is used for matrix functions.) Unfortunately, there are some flaws with the conclusions of the RL paper. Due to the way the method is derived, the authors compare the $m+1$st RL approximation with the mth FOM approximation and demonstrate improvement by several measures, mainly the reduction in the number of cycles needed to converge. Re-running these tests but comparing instead the mth approximations of both methods indicates that the RL modification by itself does not lead to improvement in these scenarios; rather, the inclusion of an additional basis vector is the source of improvement noted in the paper [54]. Our tests for BRL as a linear solver can be found in Section 7.2.4. Additional tests on BRL as a matrix

function method can be found in Section 7.5 and suggest that BRL may prove useful with matrix functions in scenarios not considered in [54].

The BRL modification also has potential in many theoretical situations as a tool for devising worst-case scenarios. For example, since this method forces a subset of spec($\mathcal{H}_m + \mathcal{M}$) to match given values, it is possible to choose very "bad" values and yet still have an approximation in the Krylov subspace with otherwise "good" properties. This bolsters the importance of methods like BFOM and BGMRES, which ensure that the spectrum of $\mathcal{H}_m + \mathcal{M}$ is still close to that of A.

4.4 Summary and outlook

By regarding BFOM as the standard Krylov subspace method, we have formulated other block Krylov methods, in particular BGMRES and BRL, and interpreted them as modified versions of BFOM. Block GMRES with restarts for shifted systems has been completely described, and two different proofs for error bounds for shifted systems with restarts have been presented. Matrix polynomials feature strongly in our approach to block Krylov methods, and they are specifically used to describe all possible low-rank modifications to BFOM such that the resulting approximation still lies in the Krylov subspace. Overall, the analysis for BGMRES serves as a template in some sense for other modified methods.

Our work raises a number of open questions and directions for future work. The shifted error bounds of Section 4.2 are not known to hold for all block positive real matrices. (Compare the assumptions for Theorems 4.14 and 4.20.) A different approach must be taken, perhaps by examining a generalized notion of angles between block subspaces (see, e.g., [134]).

Regarding the BRL method, work by Frommer and Schweitzer for the case $s = 1$ suggests that it may prove useful as an error estimator in the Lanczos method for matrix functions [56]. Having a "built-in" error estimator is important for making matrix function algorithms cheaper and more accurate than simply checking the sequence of differences between approximations. At the same time, the BRL method requires more theoretical research, especially within the context of our generalized framework. Orthogonal matrix polynomials (see, e.g., [61]) would feature prominently in such analysis.

At this point, we have a variety of tools for "customizing" Krylov methods: block inner products, additive modifications, shifting, and restarting. More work in the vein of high-performance computing is needed to determine which combinations lead to robust solvers in which situations.

CHAPTER 5

BLOCK KRYLOV METHODS FOR MATRIX FUNCTIONS ACTING ON MULTIPLE VECTORS

Interest in matrix functions has been increasing the past few decades. The monograph by Higham [74] describes many of the key results in the field, particularly for computing $f(A)$ directly for general f as well as for common functions like the sign function, square root, pth root, exponential, logarithm, sine, and cosine. We do not recapitulate results on direct methods here, since many of them are well established and even preprogrammed in current MATLAB distributions [75]. We do briefly discuss some of the iterative methods for computing $f(A)\boldsymbol{b}$, many of which implicitly use direct methods after projecting and restricting A onto a smaller subspace. Then we present our block Krylov methods for $f(A)\boldsymbol{B}$, within the framework from Chapters 3 and 4.

5.1 An overview of iterative methods for $f(A)\boldsymbol{b}$

When A is large and even when it is sparse, it may be impossible to compute and store $f(A)$, because $f(A)$ is often dense. (For why this is the case, simply examine

one of the definitions of $f(A)$ in Section 2.2.) Iterative methods requiring only the action of A on a vector or block vector are necessary in such cases.

It is not unreasonable to use a polynomial approximation to f to compute $f(A)\boldsymbol{b}$. Perhaps the simplest technique is a truncated Taylor series approximation [74, section 4.3]. Other feasible polynomial methods are based on Chebyshev polynomials, Fejér polynomials, least-squares approximations, or Faber series expansions [22, 34, 99, 100, 101, 107].

Another common technique is to approximate f by a rational function $r(z) = \sum_j \frac{w_j}{z+t_j}$. Padé approximates are known to have many special properties [6, 8]. Quadrature rules [71] and partial fraction expansions [15] may also be used to generate rational approximations. A necessary tool for such methods is an efficient solver for shifted systems, such as our block-featured methods from Chapters 3 and 4, or other methods that employ techniques like eigenvalue deflation, subspace recycling, and preconditioning [12, 26, 51, 123, 128, 133, 135, 141, 142].

Krylov methods have long been established as a viable method for computing $f(A)\boldsymbol{b}$ [34, 35, 39, 52, 53, 54, 78, 81, 90, 121]. The earliest results by Druskin and Knizhnerman [34] and Saad [121] focused on the exponential and cosine, but the basic principle holds for general functions. Recall the Arnoldi relation for $s = 1$:

$$A\boldsymbol{V}_m = \boldsymbol{V}_m H_m + \boldsymbol{v}_{m+1} h_{m,m+1} \widehat{\boldsymbol{e}}_m^*. \qquad \text{(3.2 revisited)}$$

Suppose that f is defined on the field of values of A, which contains the eigenvalues of both A and H_m (cf. Lemma 3.13), then the Krylov approximation to $f(A)\boldsymbol{b}$ is given by

$$\boldsymbol{f}_m := \boldsymbol{V}_m f(H_m) \boldsymbol{V}_m^* \boldsymbol{b} = \boldsymbol{V}_m f(H_m) \widehat{\boldsymbol{e}}_1 \beta. \qquad (5.1)$$

Such an approximation is reasonable: $f(H_m)$ is equivalent to a polynomial on H_m interpolating the Ritz values (see Definition 2.2), which are close to the eigenvalues of A; and by Theorem 4.1, the approximation (5.1) is equivalent to this polynomial acting on A. Since $f(A)$ is a polynomial interpolating spec(A), the polynomial from spec(H_m) should be close to the one from spec(A). In this sense, Krylov methods for $f(A)\boldsymbol{b}$ could be thought of as a polynomial approximation to f combined with model order reduction. See [77] for additional ways of deriving the approximation (5.1).

A variety of names have been proposed for the approximation (5.1) depending on whether the Arnoldi or Lanczos method is used to generate the Krylov basis. Since the Arnoldi and Lanczos methods are, strictly speaking, methods for computing this basis; and since just about any Krylov method for linear systems requires the generation of a basis, but uses it differently (e.g., FOM vs. GMRES); we do not think it is appropriate to name methods for matrix functions based on Arnoldi or Lanczos. Instead, we propose the nomenclature (FOM)^2: Full Orthogonalization Method for Functions Of Matrices. For modified FOM or (FOM)^2, we add a suffix to indicate which modification, and for block-featured methods, we add the prefix "B."

(FOM)^2 require access to each block vector of the Krylov basis, even if A is Hermitian and a Lanczos-based procedure has used to compute $\boldsymbol{V}_m$. The reason is that while short-term recurrences may hold for H_m, they do not necessarily hold for $f(H_m)$. As a result, storage is an inherent issue for (FOM)^2, so we consider restarts.

Restarted (FOM)^2 are explored in [3, 39, 40, 53, 52, 81]. The main issue with formulating restarts for matrix functions is that there is no natural notion for defining the residual, which is what is used for restarts for linear systems. Instead, the error

$f(A)\boldsymbol{b} - \boldsymbol{f}_m$ must be approximated. In [125], different methods for estimating the error are summarized, including divided differences and rational approximations, but only the quadrature-based method of [53] is known to be stable.

5.2 Block methods for $f(A)\boldsymbol{B}$

A number of direct methods, i.e., non-iterative methods requiring full access to A, have been proposed for computing the exponential and trigonometric functions. Al-Mohy and Higham [5] use scaling and squaring techniques and a truncated Taylor series approximation to compute $\exp(A)\boldsymbol{B}$. Their method is dominated by matrix-block-vector multiplications $A\boldsymbol{B}$, and they demonstrate its superiority to $(\text{FOM})^2$ without restarts acting on each column of $\boldsymbol{B}$. Higham and Kandolf [76] modify the algorithm of [5] for computing $f(A)\boldsymbol{B}$, where f is a trigonometric or hyperbolic trigonometric function. In [4], Al-Mohy uses scaling techniques, truncated Taylor series, and Chebyshev polynomials to devise a method for $f(tA)\boldsymbol{B}$ and $f(tA^{1/2})\boldsymbol{B}$, where f is cosine, sine, or sinc, or a hyperbolic version of one of those. In all the methods from this group, the authors perform extensive forward error analysis.

Iterative block methods have also been considered, mainly for the exponential function and some classes of matrices. Lopez and Simoncini [96] developed a block Krylov method for $\exp(A)\boldsymbol{B}$ that ensures that certain geometric properties of $\boldsymbol{B}$ are preserved when A is skew-symmetric or Hamiltonian. They do so via a classical block Krylov subspace and a BFOM-like approximation that utilizes properties of the matrix exponential and its action on special matrices. Wu, Pang, and Sun [141] explore an alternatively shifted BFOM method for computing $\exp(A)\boldsymbol{B}$. They approximate the exponential a priori with a Carathéodory-Fejér rational approximation and use

an inexact approximation to A^{-1}, which has a preconditioning-like effect on shifted BFOM. The method shows great improvement over state-of-the-art methods when one can store an LU factorization of A.

There are many benefits to such methods, but they do not cover all possible situations. Our aim is to provide methods that are defined for more general functions and that take advantage of the sparsity of A, especially when A is not known explicitly or there are memory limitations. Iterative methods with restarts are essential in such situations, and with the versatile framework for restarted block Krylov methods from Chapters 3 and 4, we have many tools for building new methods for matrix functions.

Again, the question of reasonableness arises, i.e., does it make sense to look for approximations to $f(A)\boldsymbol{B}$ in $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$? To answer this question, we utilize interpolating matrix polynomials (Section 2.5.2) and the block Arnoldi polynomial relation (Section 4.1). Suppose f is defined and differentiable "enough" on the field of values of A (see Definition 2.5); then $f(A)$ is defined and, by Definition 2.2, there exists a $p \in \mathbb{P}_n(\mathbb{C})$ such that $f(A) = p(A)$. Regarding the coefficients of p as square matrices with constant diagonal– as in the embedding used in the proof of Theorem 3.20– and recalling that $\mathbb{C}I_s \subset \mathbb{S}$ for all $\mathbb{S}$ (see Section 2.4), we then have a matrix polynomial $P \in \mathbb{P}_n(A)$ such that $f(A)\boldsymbol{B} = P(A) \circ \boldsymbol{B}$. Keep in mind that because $\mathbb{S}$ is possibly larger than $\mathbb{C}I_s$, the polynomial P is likely not unique; what is important is that we know such a $P \in \mathbb{P}_n(\mathbb{S})$ exists. Recall now that the block Arnoldi relation

$$A\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{V}_{m+1} H_{m+1,m} \widehat{\boldsymbol{E}}_m^*, \qquad \text{(3.5 revisited)}$$

and let $\mathcal{M} = \boldsymbol{M}\widehat{\boldsymbol{E}}_m^* \in \mathbb{S}^{m\times m}$, with $\boldsymbol{M} \in \mathbb{S}^m$. As long as $f(\mathcal{H}_m + \mathcal{M})$ is defined, we can define the (modified) block FOM approximation to $f(A)\boldsymbol{B}$ as

$$\boldsymbol{F}_m := \boldsymbol{\mathcal{V}}_m f(\mathcal{H}_m + \mathcal{M})\widehat{\boldsymbol{E}}_1 B, \tag{5.2}$$

with $B = N(\boldsymbol{B})$. Indeed, with f defined on $\mathbb{F}_{\langle\cdot,\cdot\rangle_{\mathbb{S}}}(A)$, $f(\mathcal{H}_m+\mathcal{M})$ is also well defined in several important scenarios:

- when $\mathcal{M} = 0$, by Lemma 3.13;
- when A is block positive real, $\mathbb{F}_{\langle\cdot,\cdot\rangle_{\mathbb{S}}}(A) \subset \mathbb{C}^+$, and $\mathcal{M}$ is as in BGMRES, by Lemma 4.6; and
- when A is block self-adjoint and block positive definite, and $\mathcal{M}$ is chosen in a block Radau-Lanczos fashion to ensure that $\mathcal{H}_m + \mathcal{M}$ has eigenvalues in $\mathbb{F}_{\langle\cdot,\cdot\rangle_{\mathbb{S}}}(A)$.

Note that $\boldsymbol{F}_m \in \mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. By a similar argument as for $f(A)$, there exists $Q_{m-1} \in \mathbb{P}_{m-1}(\mathbb{S})$ such that $f(\mathcal{H}_m + \mathcal{M})\widehat{\boldsymbol{E}}_1 B = Q_{m-1}(\mathcal{H}_m + \mathcal{M}) \circ \widehat{\boldsymbol{E}}_1 B$. By Theorem 4.1, we then know that $\boldsymbol{F}_m = Q_{m-1}(A)\boldsymbol{B}$. Consequently, as $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$ approaches $\mathbb{C}^{n\times s}$, Q_{m-1} approaches a polynomial $\widetilde{P} \in \mathbb{P}_n(\mathbb{C})$ and $\mathcal{H}_m$ approaches A, so that $f(A)\boldsymbol{B} = \widetilde{P}(A) \circ \boldsymbol{B}$.

When $\mathcal{M} = 0$, we call (5.2) the B(FOM)^2 approximation, where B(FOM)^2 stands for Block Full Orthogonalization Method for Functions Of Matrices. When $\mathcal{M}$ comes from BGMRES, we call (5.2) the B(FOM)^2+`har` approximation, where "har" stands for "harmonic" (see Remark 4.17); and when $\mathcal{M}$ is related to BRL, B(FOM)^2+`rad`. For unspecfied $\mathcal{M}$, we abbreviate "modified B(FOM)^2 " as B(FOM)^2+`mod`.

The approximation (5.2) does not require that f have any particular kind of representation, i.e., the approximation is defined as long as $f(A)$ and $f(\mathcal{H}_m + \mathcal{M})$ are. For the rest of this chapter, however, we assume that f is a Stieltjes function as in equation (2.14). The integral form of f makes an efficient restart procedure viable.

5.2.1 B(FOM)^2 with restarts: B(FOM)$^2(m)$

The approximation (5.2) requires full access to $\boldsymbol{\mathcal{V}}_m$ and thus encounters the usual memory limitations for large m. With linear systems, we could compute an update by using the residual (see, e.g., Sections 3.2 and 4.2), but with matrix functions there is no natural notion of residual. However, the integral definition of f allows us to formulate a similar update via an error function. The development in this section is taken from [55].

It will be useful to extend the definition of $\circ$ to integrals with matrix-valued coefficients. Recall that for a matrix polynomial $P \in \mathbb{P}_d(\mathbb{S})$, $P(A) \circ \boldsymbol{B} = \sum_{k=0}^{d} A^k \boldsymbol{V} C_k$, for some coefficients $\{C_k\}_{k=0}^{d} \subset \mathbb{S}$. Let a function $F : \mathbb{C} \to \mathbb{S}$ be given as

$$F(z) = \int_\Gamma (t - z)^{-1} G(t) \, \mathrm{d}t,$$

for some $G : \mathbb{C} \to \mathbb{S}$. Then we define $F(A) \circ \boldsymbol{B}$ as

$$F(A) \circ \boldsymbol{B} := \int_\Gamma (tI - A)^{-1} \boldsymbol{B} G(t) \, \mathrm{d}t. \tag{5.3}$$

Even though (5.3) is not of the form "matrix function times a block vector" like $f(A)\boldsymbol{B}$, we can still define a B(FOM)^2 approximation to $F(A) \circ \boldsymbol{B}$, i.e.,

$$\boldsymbol{\mathcal{V}}_m F(\mathcal{H}_m + \mathcal{M}) \circ \widehat{\boldsymbol{E}}_1 B = \int_\Gamma (tI - \mathcal{H}_m - \mathcal{M})^{-1} \widehat{\boldsymbol{E}}_1 B G(t) \, \mathrm{d}t. \tag{5.4}$$

Theorem 5.1 (Theorem 4.3 in [55]): Let f be a Stieltjes function as in (2.14). For $k \geq 1$ and $t \geq 0$ with the cospatial factors $C_m^{(j)}(t) \in \mathbb{S}$ as in (3.19), define the matrix-valued function $\Delta_m^{(k)}(z)$ of the complex variable z as

$$\Delta_m^{(k)}(z) := \int_0^\infty (z+t)^{-1} C_m^{(k)}(t) \cdots C_m^{(1)}(t) \, \mathrm{d}\mu(t). \tag{5.5}$$

Let

$$\boldsymbol{F}_m^{(1)} := \boldsymbol{\mathcal{V}}_m^{(1)} f\big(\mathcal{H}_m^{(1)}\big) \widehat{\boldsymbol{E}}_1 B = \boldsymbol{\mathcal{V}}_m^{(1)} \int_0^\infty \big(\mathcal{H}_m^{(1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B \, \mathrm{d}\mu(t)$$

be the B(FOM)^2 approximation to $f(A)\boldsymbol{B}$ after the first cycle. For $k \geq 1$ set

$$\begin{aligned} \widetilde{\boldsymbol{D}}_m^{(k)} &:= \boldsymbol{\mathcal{V}}_m^{(k+1)} \Delta_m^{(k)}\big(\mathcal{H}_m^{(k+1)}\big) \circ \widehat{\boldsymbol{E}}_1, \text{ and} \\ \boldsymbol{F}_m^{(k+1)} &:= \boldsymbol{F}_m^{(k)} + \widetilde{\boldsymbol{D}}_m^{(k)}. \end{aligned} \tag{5.6}$$

Then for $k = 0, 1, \ldots$ the kth B(FOM)^2 error $\boldsymbol{D}_m^{(k+1)} := f(A)\boldsymbol{B} - \boldsymbol{F}_m^{(k+1)}$ is given as

$$\boldsymbol{D}_m^{(k+1)} = \Delta_m^{(k+1)}(A) \circ \boldsymbol{V}_{m+1}^{(k+1)}. \tag{5.7}$$

Proof: The key to this result is reducing everything to Sh-BFOM(m). Recall the Sh-BFOM(m) approximations $\boldsymbol{X}_m^{(k)}(t)$ to the system (3.1), together with the updates $\boldsymbol{Z}_m^{(k)}(t)$, errors $\boldsymbol{E}_m^{(k)}(t) := \boldsymbol{X}_*(t) - \boldsymbol{X}_m^{(k)}(t)$, and residuals $\boldsymbol{R}_m^{(k)}(t) := \boldsymbol{B} - (A+tI)\boldsymbol{X}_m^{(k)}(t)$ (see (3.16)-(3.18)). Note that for all $k = 0, 1, \ldots$, the error representation for $\boldsymbol{D}_m^{(k+1)}$ from equation (5.7) can be written as

$$\boldsymbol{D}_m^{(k+1)} \;=\; \int_0^\infty (A+tI)^{-1} \boldsymbol{R}_m^{(k+1)}(t) \, \mathrm{d}\mu(t).$$

The exact error (5.7) with $k = 0$ is found via

$$\begin{aligned}\boldsymbol{D}_m^{(1)} = f(A)\boldsymbol{B} - \boldsymbol{F}_m^{(1)} &= \int_0^\infty (A + tI)^{-1}\boldsymbol{B} - \boldsymbol{\mathcal{V}}_m^{(1)}\boldsymbol{Y}_m^{(1)}(t)B \,\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1}\boldsymbol{B} - \boldsymbol{X}_m^{(1)}(t) \,\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1}\boldsymbol{R}_m^{(1)}(t) \,\mathrm{d}\mu(t).\end{aligned}$$

By induction, we can express the exact error for subsequent cycles $k \geq 1$ as

$$\boldsymbol{D}_m^{(k+1)} = f(A)\boldsymbol{B} - \boldsymbol{F}_m^{(k+1)} = f(A)\boldsymbol{B} - (\boldsymbol{F}_m^{(k)} + \widetilde{\boldsymbol{D}}_m^{(k)}) = \boldsymbol{D}_m^{(k)} - \widetilde{\boldsymbol{D}}_m^{(k)}$$

and use equations (3.16) and (3.18) to find that

$$\begin{aligned}\boldsymbol{D}_m^{(k+1)} &= \int_0^\infty (A + tI)^{-1}\boldsymbol{R}_m^{(k)}(t) \,\mathrm{d}\mu(t) \\ &\quad - \boldsymbol{\mathcal{V}}_m^{(k+1)} \int_0^\infty \left(\mathcal{H}_m^{(k+1)} + tI\right)^{-1} \widehat{\boldsymbol{E}}_1 C_m^{(k)}(t) \cdots C_m^{(1)}(t) \,\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1}\boldsymbol{R}_m^{(k)}(t) - \boldsymbol{Z}_m^{(k)}(t) \,\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1} \left(\boldsymbol{R}_m^{(k)}(t) - (A + tI)\boldsymbol{Z}_m^{(k)}(t)\right) \,\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1}\boldsymbol{R}_m^{(k+1)}(t) \,\mathrm{d}\mu(t),\end{aligned}$$

with the last equality holding by equation (3.18). □

Algorithm 5.2.1 employs Theorem 5.1. Since quadrature is used to evaluate the error function $\Delta_m^{(k)}$, if the quadrature nodes are known beforehand, it is possible to preallocate memory for the cospatial factors, much in the same way as for Algorithm 3.2.1. In such a case, not all the $\mathcal{H}_m^{(k)}$ need to be stored from cycle to cycle. In practice, however, adaptive quadrature is necessary to compute the error

to high enough accuracy, meaning that all $\mathcal{H}_m^{(k)}$ must be stored so that $C_m^{(k)}(t)$ can be computed for arbitrary values of t.

Algorithm 5.2.1: B(FOM)$^2(m)$: block full orthogonalization method for functions of matrices with restarts

---

1: Given f, A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, m, t, `tol`

2: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(1)}$, $\underline{\mathcal{H}}_m^{(1)}$, and $B^{(1)}$

3: Compute and store $\boldsymbol{F}_m^{(1)} = \boldsymbol{\mathcal{V}}_m^{(1)} f\big(\mathcal{H}_m^{(1)}\big) \widehat{\boldsymbol{E}}_1 B$

4: Compute and store $C_m^{(1)}(t) = H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \big(\mathcal{H}_m^{(1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)}$ to define $\Delta_m^{(1)}(z)$

5: **for** $k = 1, 2, \ldots,$ until convergence **do**

6: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{V}_{m+1}^{(k)}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(k+1)}$ in place of the previous basis

7: Compute $\widetilde{\boldsymbol{D}}_m^{(k)} := \boldsymbol{\mathcal{V}}_m^{(k+1)} \Delta_m^{(k)}\big(\mathcal{H}_m^{(k+1)}\big) \circ \widehat{\boldsymbol{E}}_1$, where $\Delta_m^{(k)}(z)$ is evaluated via quadrature

8: Compute $\boldsymbol{F}_m^{(k+1)} := \boldsymbol{F}_m^{(k)} + \widetilde{\boldsymbol{D}}_m^{(k)}$ and replace $\boldsymbol{F}_m^{(k)}$

9: Compute $C_m^{(k+1)}(t) = H_{m+1,m}^{(k+1)} \widehat{\boldsymbol{E}}_m^* \big(\mathcal{H}_m^{(k+1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B^{(k+1)} C_m^{(k)}(t)$ and replace $C_m^{(k)}(t)$

10: **end for**

11: **return** $\boldsymbol{F}_m^{(k+1)}$

The convergence of Algorithm 5.2.1 can be shown by generalizing the techniques of [52, Lemma 4.1 and Theorem 4.3] to the block case.

Theorem 5.2 (Theorem 4.5 in [55]): Let f be a Stieltjes function, $A \in \mathbb{C}^{n\times n}$ $\langle\!\langle \cdot,\cdot \rangle\!\rangle_{\mathbb{S}}$-self-adjoint and $\langle \cdot,\cdot \rangle_{\mathbb{S}}$-positive definite, and $\boldsymbol{B} \in \mathbb{C}^{n\times s}$. Let $\boldsymbol{D}_m^{(k)}$ from equation (5.7) be the error of the B(FOM)$^2(m)$ approximation after k cycles. Then, with the quantities defined in (3.21), it holds that

$$\left\|\boldsymbol{D}_m^{(k)}\right\|_{A\text{-}\mathbb{S}} \leq \|\boldsymbol{B}\|_{\mathbb{S}} \sqrt{\lambda_{\max}} \int_0^\infty \frac{\xi_m(t)^k}{\sqrt{\lambda_{\min}+t}\sqrt{\lambda_{\max}+t}}\,\mathrm{d}\mu(t) \leq \gamma \xi_m(0)^k, \tag{5.8}$$

where $\gamma = \|\boldsymbol{B}\|_{\mathbb{S}} \sqrt{\lambda_{\max}} f(\sqrt{\lambda_{\min}\lambda_{\max}})$. In particular, B(FOM)$^2(m)$ converges for all cycle lengths m as $k \to \infty$.

Proof: We write the exact error $\boldsymbol{D}_m^{(k)}$ as an integral over the error of Sh-BFOM(m), i.e.,

$$\boldsymbol{D}_m^{(k)} = \int_0^\infty (A+tI)^{-1}\boldsymbol{R}_m^{(k)}\,\mathrm{d}\mu(t) = \int_0^\infty \boldsymbol{E}_m^{(k)}(t)\,\mathrm{d}\mu(t).$$

Passing the energy norm through the integral (cf. [126, Lemma 2.1]), we obtain

$$\left\|\boldsymbol{D}_m^{(k)}\right\|_{A\text{-}\mathbb{S}} \leq \int_0^\infty \left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A\text{-}\mathbb{S}}\,\mathrm{d}\mu(t). \tag{5.9}$$

By Theorem 3.23,

$$\left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A\text{-}\mathbb{S}} \leq \|\boldsymbol{B}\|_{\mathbb{S}} \sqrt{\lambda_{\max}} \frac{\xi_m(t)^k}{\sqrt{(\lambda_{\min}+t)(\lambda_{\max}+t)}}, \tag{3.22 revisited}$$

which, combined with the inequality (5.9), gives the first inequality in (5.8). Noting that $\xi_m(t)$ is a monotonically decreasing function of t [52, Proposition 4.2], we can bound $\xi_m(t)$ by $\xi_m(0)$. The denominator of $\xi_m(0)$ is the geometric mean $\sqrt{\lambda_{\min}\lambda_{\max}}$, which satisfies

$$\frac{1}{\sqrt{\lambda_{\min}+t}\sqrt{\lambda_{\max}+t}} \leq \frac{1}{\sqrt{\lambda_{\min}\lambda_{\max}}+t},$$

implying that

$$\int_0^\infty \frac{\xi_m(t)^k}{\sqrt{\lambda_{\min}+t}\sqrt{\lambda_{\max}+t}}\,\mathrm{d}\mu(t) \;\le\; \xi_m(0)^k \int_0^\infty \frac{1}{\sqrt{\lambda_{\min}\lambda_{\max}}+t}\,\mathrm{d}\mu(t).$$

The integral on the right is just $f(\sqrt{\lambda_{\min}\lambda_{\max}})$, thus concluding the second inequality in (5.8). □

Note that when $\boldsymbol{B}$ is a column vector, we recover the same results as in [53].

5.2.2 B(FOM)^2+har with restarts: B(FOM)^2+har(m)

To formulate restarts for B(FOM)^2+har, we use the theory developed for Sh-BGMRES(m). The results are similar to B(FOM)^2, with the only differences stemming from the cospatial relationship between the residuals. Recall from Section 4.2 that

$$\boldsymbol{R}_m^{(k)}(t) = \boldsymbol{\mathcal{V}}_{m+1}^{(k)} \begin{bmatrix} \boldsymbol{U}^{(k)} \\ -I \end{bmatrix} G_m^{(k)}(t) \qquad \text{(4.32 revisited)}$$

$$G_m^{(k)}(t) = H_{m+1,m}^{(k)} \widehat{\boldsymbol{E}}_m^* (\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)} + tI)^{-1} \widehat{\boldsymbol{E}}_1 B^{(k)}. \qquad \text{(4.33 revisited)}$$

Theorem 5.3: Let f be a Stieltjes function as in (2.14). For $k \ge 1$ and $t \ge 0$ with the cospatial factors $G_m^{(j)}(t) \in \mathbb{S}$ as in (4.33), define the matrix-valued function $\Delta_m^{(k)}(z)$ of the complex variable z as

$$\Delta_m^{(k)}(z) := \int_0^\infty (z+t)^{-1} G_m^{(k)}(t)\,\mathrm{d}\mu(t). \qquad (5.10)$$

Let

$$\boldsymbol{F}_m^{(1)} := \boldsymbol{\mathcal{V}}_m^{(1)} f\big(\mathcal{H}_m^{(1)} + \mathcal{M}_{\texttt{har}}^{(1)}\big)\widehat{\boldsymbol{E}}_1 B = \boldsymbol{\mathcal{V}}_m^{(1)} \int_0^\infty \big(\mathcal{H}_m^{(1)} + \mathcal{M}_{\texttt{har}}^{(1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)}\,\mathrm{d}\mu(t)$$

be the B(FOM)^2+`har` approximation to $f(A)\boldsymbol{B}$ after the first cycle. For $k \geq 1$ set

$$\widetilde{\boldsymbol{D}}_m^{(k)} := \boldsymbol{\mathcal{V}}_m^{(k+1)} \Delta_m^{(k)} \big(\mathcal{H}_m^{(k+1)} + \mathcal{M}_{\texttt{har}}^{(k+1)} \big) \circ \widehat{\boldsymbol{E}}_1 B^{(k+1)}, \text{ and} \tag{5.11}$$

$$\boldsymbol{F}_m^{(k+1)} := \boldsymbol{F}_m^{(k)} + \widetilde{\boldsymbol{D}}_m^{(k)}.$$

Then for $k = 0, 1, \ldots$, the kth B(FOM)^2 error $\boldsymbol{D}_m^{(k+1)} := f(A)\boldsymbol{B} - \boldsymbol{F}_m^{(k+1)}$ is given as

$$\boldsymbol{D}_m^{(k+1)} = \Delta_m^{(k+1)}(A) \circ \boldsymbol{\mathcal{V}}_{m+1}^{(k+1)} \begin{bmatrix} \boldsymbol{U}^{(k+1)} \\ -I \end{bmatrix}. \tag{5.12}$$

The proof for Theorem 5.3 is nearly identical to that of Theorem 5.1, so we do not present it here.

Algorithm 5.2.2 summarizes the block harmonic method for matrix functions. It encounters the same preallocation issues as Algorithm 5.2.1 in the case that the nodes of the quadrature are not fixed. In addition to storing $\underline{\mathcal{H}}_m^{(k)}$ per cycle, it is also necessary to store $B^{(k)}$ per cycle.

The convergence of Algorithm 5.2.2 depends on bounds for the Sh-BGMRES(m) errors.

Theorem 5.4: Let $A \in \mathbb{C}^{n\times n}$ be block positive real, $\boldsymbol{B} \in \mathbb{C}^{n\times s}$, f a Stieltjes function, and $\boldsymbol{F}_m^{(k)}$ the approximations defined in Theorem 5.3. Take $\nu_{\max}$, ρ, and γ as in (4.34). If the conclusion of either Theorem 4.14 or Theorem 4.20 holds, i.e., if

$$\left\| \boldsymbol{E}_m^{(k)}(t) \right\|_{A^*A\text{-}\mathbb{S}} \leq \sqrt{\frac{\nu_{\max}}{(t+\rho\nu_{\max})^2}} \left(1 - \frac{\gamma^2}{\nu_{\max}} \right)^{mk/2} \|\boldsymbol{B}\|_{\mathbb{S}},$$

then

$$\left\| \boldsymbol{D}_m^{(k)} \right\|_{A^*A\text{-}\mathbb{S}} \leq \sqrt{\nu_{\max}} f(\rho\nu_{\max}) \|\boldsymbol{B}\|_{\mathbb{S}} \left(1 - \frac{\gamma^2}{\nu_{\max}} \right)^{mk/2}.$$

Since $0 < 1 - \frac{\gamma^2}{\nu_{\max}} < 1$, B(FOM)^2+`har` converges for all restart cycle lengths m.

Algorithm 5.2.2: B(FOM)^2+`har`(m): block harmonic method for functions of matrices with restarts

1: Given f, A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, m, t, `tol`

2: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{B}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(1)}$, $\underline{\mathcal{H}}_m^{(1)}$, and $B^{(1)}$

3: Compute and store $\boldsymbol{F}_m^{(1)} = \boldsymbol{\mathcal{V}}_m^{(1)} f\big(\mathcal{H}_m^{(1)} + \mathcal{M}^{(1)}\big)\widehat{\boldsymbol{E}}_1 B^{(1)}$

4: Compute and store $G_m^{(1)}(t) = H_{m+1,m}^{(1)} \widehat{\boldsymbol{E}}_m^* \big(\mathcal{H}_m^{(1)} + \mathcal{M}^{(1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B^{(1)}$ to define $\Delta_m^{(1)}(z)$

5: **for** $k = 1, 2, \ldots,$ until convergence **do**

6: Run Algorithm 3.1.1 with inputs A, $\boldsymbol{V}_{m+1}^{(k)}$, $\mathbb{S}$, $\langle\!\langle \cdot, \cdot \rangle\!\rangle_{\mathbb{S}}$, N, and m and store $\boldsymbol{\mathcal{V}}_{m+1}^{(k+1)}$ in place of the previous basis

7: Compute $\widetilde{\boldsymbol{D}}_m^{(k)} := \boldsymbol{\mathcal{V}}_m^{(k+1)} \Delta_m^{(k)}\big(\mathcal{H}_m^{(k+1)} + \mathcal{M}_{\texttt{har}}^{(k+1)}\big) \circ \widehat{\boldsymbol{E}}_1 B^{(k+1)}$, where $\Delta_m^{(k)}(z)$ is evaluated via quadrature

8: Compute $\boldsymbol{F}_m^{(k+1)} := \boldsymbol{F}_m^{(k)} + \widetilde{\boldsymbol{D}}_m^{(k)}$ and replace $\boldsymbol{F}_m^{(k)}$

9: Compute $G_m^{(k+1)}(t) = H_{m+1,m}^{(k+1)} \widehat{\boldsymbol{E}}_m^* \big(\mathcal{H}_m^{(k+1)} + \mathcal{M}^{(k+1)} + tI\big)^{-1} \widehat{\boldsymbol{E}}_1 B^{(k+1)}$ and replace $G_m^{(k)}(t)$

10: **end for**

11: **return** $\boldsymbol{F}_m^{(k+1)}$

Proof: The proof follows by noting that

$$\left\|\boldsymbol{D}_m^{(k)}\right\|_{A^*A\text{-}\mathbb{S}} \leq \int_0^\infty \left\|\boldsymbol{E}_m^{(k)}(t)\right\|_{A^*A\text{-}\mathbb{S}} \,\mathrm{d}\mu(t), \tag{5.13}$$

applying Theorem 4.14 or 4.20, and noting also that $\int_0^\infty \frac{1}{t+\rho\nu_{\max}} \,\mathrm{d}\mu(t) = f(\rho\nu_{\max})$. □

As in the case of B(FOM)$^2(m)$, the results for B(FOM)^2+$\texttt{har}(m)$ reduce to those of the non-block harmonic method of [52] when $s = 1$ and $\boldsymbol{B}$ is a column vector.

5.2.3 B(FOM)^2+mod with restarts: B(FOM)^2+$\texttt{mod}(m)$

It is, of course, possible to devise algorithms with other modifications, for example, a BRL modification. However, formulating an efficient restart procedure for other modifications is not trivial, because a cospatial relationship between the residuals of each cycle is needed. Furthermore, there is no guarantee of convergence for an arbitrary $\mathcal{M}$. The BRL modification is proof of this, since so-called "bad" eigenvalues can be forced into the spectrum of $\mathcal{H}_m + \mathcal{M}$ so that the approximation is always inaccurate by a certain amount. However, a convergence proof for non-block RL is presented in [54] which suggests that convergence for the BRL method could be obtained as long as the spectrum of A is positive and the eigenvalues fixed by $\mathcal{H}_m + \mathcal{M}_{\texttt{rad}}$ are greater than the largest eigenvalue of A. Certainly similar restrictions must be placed on $\mathcal{M}$ for a feasible algorithm, but deducing precisely what these restrictions are in general is beyond the scope of this work.

5.3 Expressions for the matrix error function for special f

Following the quadrature rules suggested by [53], we work out the explicit expressions for the B(FOM)$^2(m)$ error function $\Delta_m^{(k)}$ for two Cauchy-Stieltjes functions: $z^{-\alpha}$ and $\exp(z)$. The pattern becomes quickly apparent, making it easy to derive such expressions for other functions with integral representation and other B(FOM)^2+$\texttt{mod}(m)$ methods.

5.3.1 $f(z) = z^{-\alpha}$, $0 < \alpha < 1$

Recall from Section 2.6 that for $\alpha \in (0, 1)$,

$$z^{-\alpha} = \frac{\sin((1-\alpha)\pi)}{\pi} \int_0^\infty \frac{1}{z+t} \, \mathrm{d}\mu(t),$$

with $\mathrm{d}\mu(t) = t^{-\alpha} \, \mathrm{d}t$. To approximate this integral, we apply the Cayley transform $t = \delta\frac{1-x}{1+x}$, for some $\delta > 0$ and use N-node Gauss-Jacobi quadrature for the interval $[-1, 1]$ (as in, e.g., [28]). The associated error function can then be written as

$$\widetilde{\boldsymbol{D}}_m^{(k)} \approx -c_{\alpha,\delta} \sum_{j=1}^{N} \frac{w_j}{1+x_j} \boldsymbol{\mathcal{V}}_m^{(k+1)} (\mathcal{H}_m^{(k+1)} + t_j I)^{-1} \widehat{\boldsymbol{E}}_1 C_m^{(k)}(t_j) \ldots C_m^{(1)}(t_j),$$

with the Gauss-Jacobi nodes $\{x_j\}_{j=1}^N$, weights $\{w_j\}_{j=1}^N$, and $\{t_j := \delta\frac{1-x_j}{1+x_j}\}_{j=1}^N$. Be default, we take $\delta = 1$.

5.3.2 $f(z) = \exp(z)$

As a Cauchy-Stieltjes function, the exponential still fits within our framework:

$$\exp(z) = \frac{1}{2\pi i} \int_\Gamma \frac{\exp(t)}{t-z} \, \mathrm{d}t. \tag{5.14}$$

Following [136, 139, 140], we take Γ as a parabolic contour parametrized as

$$\gamma(s) = a + is - cs^2, \quad s \in \mathbb{R}.$$

The parameters a and c are chosen anew for each restart cycle to ensure that Γ encloses the eigenvalues of the matrix $\mathcal{H}_m$. The infinite interval of integration for s is truncated for a given error tolerance `tol` by the truncation parameter $s_0 := \sqrt{a - \log(\texttt{tol})/c}$, so that $|\exp(\gamma(\pm s_0))| = \texttt{tol}$. From the N-point midpoint rule on $[-s_0, s_0]$ we obtain the nodes $s_j := s_0(\frac{2j-1}{N})$, $j = 1, \ldots, N$. Defining $w_j := \exp(\gamma(s_j))\gamma'(s_j)$ and $t_j := -\gamma(s_j)$, we then approximate the error approximation as

$$\widetilde{\boldsymbol{D}}_m^{(k)} \approx \frac{s_0}{N\pi i} \sum_{j=1}^{N} w_j \boldsymbol{\mathcal{V}}_m^{(k+1)} (\mathcal{H}_m^{(k+1)} + t_j I)^{-1} \widehat{\boldsymbol{E}}_1 C_m^{(k)}(t_j) \cdots C_m^{(1)}(t_j).$$

5.4 A note on preconditioning

Preconditioning Algorithms 5.2.1 and 5.2.2 remains an open problem. To understand why, recall the possibilities for preconditioning the linear system $A\boldsymbol{X} = \boldsymbol{B}$, letting $P \approx A^{-1}$:

left	right
$PA\boldsymbol{X} = P\boldsymbol{B}$	$AP\boldsymbol{Y} = \boldsymbol{B}$, with $\boldsymbol{X} = P\boldsymbol{Y}$
$\boldsymbol{X}_m \in \mathscr{K}_m^{\mathbb{S}}(PA, P\boldsymbol{B})$	$\boldsymbol{X}_m \in P\mathscr{K}_m^{\mathbb{S}}(AP, \boldsymbol{B})$

In the case of either left or right preconditioning, we can recover $\boldsymbol{X}$ so that it still solves $A\boldsymbol{X} = \boldsymbol{B}$. Equivalent approaches for matrix functions would mean approximating $f(PA)P\boldsymbol{B}$ in the case of left-preconditioning or $Pf(AP)\boldsymbol{B}$ in the case of right-preconditioning. We would then need a way to recover $f(A)\boldsymbol{B}$, which is not

immediately clear or possible for general f. For Cauchy-Stieltjes functions, it may be possible to exploit to the linear system nature of the integral:

$$\begin{array}{cc} \text{left} & \text{right} \\ f(PA)P\boldsymbol{B} & Pf(AP)\boldsymbol{B} \\ = \int_\Gamma (tI - PA)^{-1} P\boldsymbol{B} g(t)\,\mathrm{d}t & = \int_\Gamma P(tI - AP)^{-1}\boldsymbol{B} g(t)\,\mathrm{d}t \end{array}$$

The difficulty here is that we need an algebraic expression to recover the original integral $\int_\Gamma (tI - A)^{-1}\boldsymbol{B} g(t)\,\mathrm{d}t$. When the preconditioner P depends on t, this may be feasible (see, e.g., [12]). In general, however, the preconditioner should be independent of the integral, thus rendering our situation rather challenging indeed.

5.5 Summary and outlook

We have demonstrated that restarted block Krylov methods are well defined for matrix functions and, when formulated in terms of shifted block Krylov methods with restarts, convergent in a number of important scenarios. Functions with Cauchy-Stieltjes representations are especially suited for these methods. We have also found that functions defined by integrals over matrix-valued coefficients (see equation (5.3)) can be approximated by these methods too.

As the first general-purpose methods proposed for $f(A)\boldsymbol{B}$, B(FOM)^2+`mod`(m) show great potential but still have some remaining issues. Since the error is calculated via quadrature, the choice of quadrature rule plays a crucial role in the efficiency of the algorithm, but we have not included this choice in our error analysis. Future analysis should account for the quadrature rule or determine a priori weights and nodes that give the desired accuracy. It may also be interesting to see how block quadrature rules [63, 116] behave.

B(FOM)^2+`mod` effectively reduces the computation of $f(A)\boldsymbol{B}$ to the computation of a rational function on $\mathcal{H}_m + \mathcal{M}$, thanks to the quadrature rule on the error function. Other common matrix function methods reduce $f(A)\boldsymbol{B}$ to a rational function on A. It would be interesting to compare the theoretical and numerical properties of such approaches. Numerical work to this end has been conducted by Wu, Pang, and Sun [141] for a particular rational approximation to the matrix exponential.

CHAPTER 6

APPLICATIONS

Matrix functions have numerous applications in scientific computing. We aim to demonstrate the efficacy of our methods on a range of problems, whose background is described in this chapter. Recall that we focus on functions with a Cauchy-Stieltjes representation, as in equation (2.3).

6.1 Differential equations

The matrix exponential is the quintessential matrix function problem [98]. Let $A : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}$, and consider the systems of differential equations

$$\frac{d}{dt}[\boldsymbol{Y}] = A\boldsymbol{Y}(t), \tag{6.1}$$

for some $\boldsymbol{Y} : [0, \infty) \to \mathbb{C}^{n\times s}$ with $\boldsymbol{Y}(0) = \boldsymbol{B}$. We can write the solution to equation (6.1) as

$$\boldsymbol{Y}(t) = \exp(At)\boldsymbol{B}.$$

The model equation (6.1) applies in particular when A is the central, second-order finite differences discretization of the Laplace operator in the time-dependent heat

equation

$$\frac{d}{dt}[\boldsymbol{Y}] = \Delta \boldsymbol{Y}(t). \tag{6.2}$$

The size of A depends on the number of discretization points in each dimension. We consider only the two-dimensional case and suppose we have the same number of points N in each dimension, so that $A \in \mathbb{C}^{N^2 \times N^2}$. We generate A via the command `gallery('poisson',N)` in MATLAB.

A simple modification to (6.2) gives the convection-diffusion equation with convection parameter ν:

$$\frac{d}{dt}[\boldsymbol{Y}] = \Delta \boldsymbol{Y}(t) + \nu \boldsymbol{Y}(t). \tag{6.3}$$

A finite-differences matrix can be built for equation (6.3) by modifying the entries of `gallery('poisson',N)` that correspond to the convection term. The value ν determines how close to being symmetric the matrix is; the larger ν is, the more difficult the problem is to solve.

The matrix function $\exp(At)$, where A corresponds to a differential operator, is useful theoretically for differential equations, but it is more often used computationally for exponential integrators, especially when the differential equation is inhomogeneous; see, e.g., [79].

6.2 Lattice QCD

Quantum chromodynamics (QCD) is the theory for describing the strong force interactions between quarks and gluons. The "chromo" part of the theory comes from the need to describe a kind of charge different from binary electric charge; this kind of charge is called "color" and can take on the values of blue, red, or green. (They do

not correspond to color in the sense of the visible light spectrum.) To run simulations on quarks, problems are mapped onto a four-dimensional space-time lattice. Each point of the lattice carries 12 variables, and each variable corresponds to one of the combinations of three colors and four spins a particle can carry.

The overlap Dirac operator (see, e.g., [103]) features prominently in these simulations, an essential component of which is the computation of

$$\operatorname{sign}(Q)\boldsymbol{V},$$

for a matrix Q, which is the Hermitian form of the Wilson-Dirac matrix defined in, e.g., [137]. For additional sources on QCD theory and its connection to numerical linear algebra, see [17, 59].

The sign function does not at first glance have a Cauchy-Stieltjes representation. However,

$$\operatorname{sign}(z) = (z^2)^{-1/2},$$

and the inverted square root is a Stieltjes function. We can therefore apply our methods to $A^{-1/2}$, where $A := Q^2$.

6.3 Functions of tensors

As high-dimensional analogues of matrices, tensors play crucial roles in network analysis [23] and multidimensional differential equations [87]. A variety of decompositions and algorithms have been developed over the years to extract and understand properties of tensors [91]. A natural question is whether the notion of functions of tensors, defined in analogy to functions of matrices as a scalar function taking

a tensor $\mathcal{A}$ as its argument, could prove to be yet another useful tool for studying multidimensional data.

Unfortunately, the definition of such notion is not nearly as straightforward for tensors as it is for matrices. For matrices, the definitions of integration, polynomials, eigendecompositions (ED), and singular value decompositions (SVD) are unique and well established throughout linear algebra, and all of these notions serve as building blocks for definitions of matrix functions, reducing to the same object under reasonable circumstances [9, 74]. Classical decompositions such as Tucker and CANDECOMP/PARAFAC (CP) generalize the SVD in some sense; but many other generalizations of ED and SVD also exist for tensors [29, 88, 91, 92, 95, 104, 112, 113]. Each decomposition is based on maintaining or extracting some inherent structures, which are distinct in high-order settings. That is, a tensor function definition based on the Tucker decomposition would produce a fundamentally different object compared to one based on the CP decomposition.

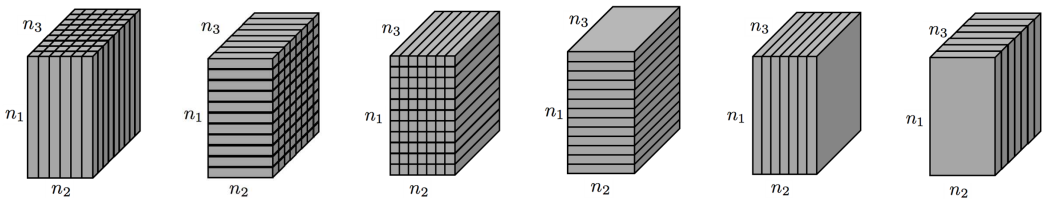
We propose a definition for functions of tensors based on a newer paradigm, the tensor t-product [19, 88, 89]. The beauty of such a definition is that it reduces to the $f(A)\boldsymbol{B}$ problem. One can think of this object in two ways: 1) as a new application of matrix function theory, especially for the $f(A)\boldsymbol{B}$ problem; and 2) as a generalization of such theory to higher-order arrays.

We make a brief comment on syntax and disambiguation: the phrase "tensor function" already has an established meaning in physics; see, e.g., [14]. The most precise phrase for our object of interest would be "a function of a multidimensional array," in analogy to "a function of a matrix." However, since combinations of

prepositional phrases can be cumbersome in English, we risk compounding literature searches by resorting to the term "tensor function."

6.3.1 The tensor t-product and its properties

We direct the reader to Figure 6.1[1] for different "views" of a third-order tensor, which will be useful in visualizing the forthcoming concepts.



(a) (b) (c) (d) (e) (f)

FIGURE 6.1. Different views of a third-order tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. (a) column fibers: $\mathcal{A}(:, j, k)$; (b) row fibers: $\mathcal{A}(i, :, k)$; (c) tube fibers: $\mathcal{A}(i, j, :)$; (d) horizontal slices: $\mathcal{A}(i, :, :)$; (e) lateral slices: $\mathcal{A}(:, j, :)$; (f) frontal slices: $\mathcal{A}(:, :, k)$

In [19, 88, 89], a new concept is proposed for multiplying third-order tensors, based on viewing a tensor as a stack of frontal slices (as in Figure 6.1(f)). We consider a tensor $\mathcal{A}$ of size $m \times n \times p$ and $\mathcal{B}$ of size $n \times s \times p$ and denote their frontal faces respectively as $A^{(i)}$ and $B^{(i)}$, $i = 1, \ldots, p$. We also define the operations `bcirc`,

[1]We thank Misha Kilmer for these images.

`unfold`, `fold`, as

$$\texttt{bcirc}(\mathcal{A}) := \begin{bmatrix} A^{(1)} & A^{(p)} & A^{(p-1)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(p)} & \cdots & A^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(p)} & A^{(p-1)} & \ddots & A^{(2)} & A^{(1)} \end{bmatrix}, \tag{6.4}$$

$$\texttt{unfold}(\mathcal{A}) := \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(p)} \end{bmatrix}, \text{ and } \texttt{fold}(\texttt{unfold}(\mathcal{A})) := \mathcal{A}.$$

The *t-product* of two tensors $\mathcal{A}$ and $\mathcal{B}$ is then given as

$$\mathcal{A} * \mathcal{B} := \texttt{fold}(\texttt{bcirc}(\mathcal{A})\texttt{unfold}(\mathcal{B})).$$

Note that the operators `fold`, `unfold`, and `bcirc` are linear.

The notion of transposition is defined face-wise, i.e., $\mathcal{A}^*$ is the $n \times m \times p$ tensor obtained by taking the conjugate transpose of each frontal slice of $\mathcal{A}$ and then reversing the order of the second through pth transposed slices.

For tensors with $n \times n$ square faces, there is a tensor identity $\mathcal{I}_{n\times n\times p} \in \mathbb{C}^{n\times n\times p}$, whose first frontal slice is the $n \times n$ identity matrix and whose remaining frontal slices are all the zero matrix. Recall from equation (2.1) that $\widehat{\boldsymbol{E}}_1^{np\times n} = \widehat{\boldsymbol{e}}_1^{p} \otimes I_n$; it follows that

$$\widehat{\boldsymbol{E}}_1^{np\times n} = \texttt{unfold}(\mathcal{I}_{n\times n\times p}). \tag{6.5}$$

With $\mathcal{I}_{n\times n\times p}$, it is possible to define the notion of an inverse with respect to the t-product. Namely, $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n\times n\times p}$ are inverses of each other if $\mathcal{A} * \mathcal{B} = \mathcal{I}_{n\times n\times p}$ and $\mathcal{B} * \mathcal{A} = \mathcal{I}_{n\times n\times p}$. The t-product formalism further gives rise to its own notion of polynomials, with powers of tensors defined as $\mathcal{A}^j := \underbrace{\mathcal{A} * \cdots * \mathcal{A}}_{j \text{ times}}$.

Assuming that $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ has diagonalizable faces, we can also define a tensor eigendecomposition. That is, we have that $A^{(k)} = X^{(k)}D^{(k)}(X^{(k)})^{-1}$, for all $k = 1, \ldots, p$, and define $\mathcal{X}$ and $\mathcal{D}$ to be the tensors whose faces are $X^{(k)}$ and $D^{(k)}$, respectively. Then

$$\mathcal{A} = \mathcal{X} * \mathcal{D} * \mathcal{X}^{-1} \text{ and } \mathcal{A} * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * \boldsymbol{d}_i, \tag{6.6}$$

where $\vec{\mathcal{X}}_i$ are the $n \times 1 \times p$ lateral slices of $\mathcal{X}$ (see Figure 6.1(e)) and $\boldsymbol{d}_j$ are the $1 \times 1 \times p$ tubal fibers of $\mathcal{D}$ (see Figure 6.1). We say that $\mathcal{D}$ is *f-diagonal*, i.e., that each of its frontal faces is a diagonal matrix.

6.3.2 The tensor t-exponential

As motivation and in analogy to Section 6.1, we consider the solution to a multidimensional ordinary differential equation. Suppose that $\mathcal{A}$ has square frontal faces, i.e., that $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ and let $\mathcal{B} \in \mathbb{C}^{n\times s\times p}$, whose entries depend on τ. With $\frac{\mathrm{d}}{\mathrm{d}\tau}$ acting element-wise, we consider the differential equation

$$\frac{\mathrm{d}\mathcal{B}}{\mathrm{d}\tau}(\tau) = \mathcal{A} * \mathcal{B}(\tau). \tag{6.7}$$

Unfolding both sides leads to

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{bmatrix} B^{(1)}(\tau) \\ \vdots \\ B^{(n)}(\tau) \end{bmatrix} = \texttt{bcirc}(\mathcal{A}) \begin{bmatrix} B^{(1)}(\tau) \\ \vdots \\ B^{(n)}(\tau) \end{bmatrix},$$

whose solution can be expressed in terms of the matrix exponential as

$$\begin{bmatrix} B^{(1)}(\tau) \\ \vdots \\ B^{(n)}(\tau) \end{bmatrix} = \exp(\texttt{bcirc}(\mathcal{A})\tau) \begin{bmatrix} B^{(1)}(0) \\ \vdots \\ B^{(n)}(0) \end{bmatrix}.$$

Folding both sides again leads to the *tensor t-exponential*,

$$\mathcal{B}(t) = \texttt{fold}(\exp(\mathcal{A}\tau)\texttt{unfold}(\mathcal{B}(0))) =: \exp(\mathcal{A}\tau) * \mathcal{B}(0). \tag{6.8}$$

6.3.3 The tensor t-function and its properties

Using the tensor t-exponential as inspiration, we can define a more general notion for the scalar function f of a tensor $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ multiplied by a tensor $\mathcal{B} \in \mathbb{C}^{n\times s\times p}$ as

$$f(\mathcal{A}) * \mathcal{B} := \texttt{fold}(f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})), \tag{6.9}$$

which we call the *tensor t-function*. Note that $f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})$ is merely a matrix function times a block vector. If $\mathcal{B} = \mathcal{I}_{n\times n\times p}$, then by equation (6.5) the definition for $f(\mathcal{A})$ reduces to

$$f(\mathcal{A}) := \texttt{fold}\Big(f(\texttt{bcirc}(\mathcal{A}))\widehat{\boldsymbol{E}}_1^{np\times n}\Big). \tag{6.10}$$

But does the definition (6.9) behave "as expected" in common scenarios? To answer this question, we require some results on block circulant matrices.

Theorem 6.1 (Theorem 5.6.5 in [27]): Suppose $A, B \in \mathbb{C}^{np\times np}$ are block circulant matrices with $n \times n$ blocks. Let $\{\alpha_j\}_{j=1}^k$ be scalars. Then A^T, A^*, $\alpha_1 A + \alpha_2 B$, AB, $q(A) = \sum_{j=1}^k \alpha_j A^j$, and A^{-1} (when it exists) are also block circulant.

Remark 6.2: From (6.4), we can see that any block circulant matrix $C \in \mathbb{C}^{np\times np}$ can be represented by its first column $C\widehat{\boldsymbol{E}}_1^{np\times n}$. Let $\mathcal{C} \in \mathbb{C}^{n\times n\times p}$ be a tensor whose frontal faces are the block entries of $C\widehat{\boldsymbol{E}}_1^{np\times n}$. Then $\mathcal{C} = \texttt{fold}\Big(C\widehat{\boldsymbol{E}}_1^{np\times n}\Big)$.

Lemma 6.3: Let $\mathcal{A} \in \mathbb{C}^{m\times n\times p}$ and $\mathcal{B} \in \mathbb{C}^{n\times s\times p}$. Then

(i) $\texttt{unfold}(\mathcal{A}) = \texttt{bcirc}(\mathcal{A})\widehat{\boldsymbol{E}}_1^{np\times n}$;

(ii) $\texttt{bcirc}\Big(\texttt{fold}\Big(\texttt{bcirc}(\mathcal{A})\widehat{\boldsymbol{E}}_1^{np\times n}\Big)\Big) = \texttt{bcirc}(\mathcal{A})$;

(iii) $\texttt{bcirc}(\mathcal{A} * \mathcal{B}) = \texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{B})$;

(iv) $\texttt{bcirc}(\mathcal{A})^j = \texttt{bcirc}(\mathcal{A}^j)$, for all $j = 0, 1, \ldots$; and

(v) $(\mathcal{A} * \mathcal{B})^* = \mathcal{B}^* * \mathcal{A}^*$.

Proof: We drop the superscripts on $\widehat{\boldsymbol{E}}_1^{np\times n}$ for ease of presentation. Parts (i) and (ii) follow from Remark (6.2). To prove part (iii), we note by part (i) that

$$\begin{aligned}\texttt{bcirc}(\mathcal{A} * \mathcal{B}) &= \texttt{bcirc}(\texttt{fold}(\texttt{bcirc}(\mathcal{A})\texttt{unfold}(\mathcal{B})))\\ &= \texttt{bcirc}\Big(\texttt{fold}\Big(\texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{B})\widehat{\boldsymbol{E}}_1\Big)\Big).\end{aligned}$$

Note that $\texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{B})$ is a block circulant matrix by Theorem 6.1. Then by part (ii),

$$\texttt{bcirc}\Big(\texttt{fold}\Big(\texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{B})\widehat{\boldsymbol{E}}_1\Big)\Big) = \texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{B}).$$

Part (iv) follows by induction on part (iii). Part (v) is the same as [89, Lemma 3.16]. □

Theorem 6.4: Let $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ and $\mathcal{B} \in \mathbb{C}^{n\times s\times p}$.

(i) If $f \equiv q$, where q is a polynomial, then the tensor t-function definition (6.9) matches the polynomial notion in the t-product formalism, i.e.,

$$\texttt{fold}(q(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})) = \texttt{fold}(\texttt{bcirc}(q(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})).$$

(ii) Let q be the scalar polynomial guaranteed by Definition 2.2 so that $f(\texttt{bcirc}(\mathcal{A})) = q(\texttt{bcirc}(\mathcal{A}))$. Then $f(\mathcal{A}) * \mathcal{B} = q(\mathcal{A}) * \mathcal{B}$.

(iii) If $\mathcal{A}$ is a matrix and $\mathcal{B}$ a block vector (i.e., if $p = 1$), then $f(\mathcal{A}) * \mathcal{B}$ reduces to the usual matrix function definition.

(iv) If $f(z) = z^{-1}$, then $f(\mathcal{A}) * \mathcal{A} = \mathcal{A} * f(\mathcal{A}) = \mathcal{I}_{n\times n\times p}$.

Proof: For part (i), let $q(z) = \sum_{j=1}^{m} c_j z^j$. Then by Lemma 6.3(iv) and the linearity of `fold`, we have that

$$\begin{aligned}
\texttt{fold}(q(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})) &= \texttt{fold}\left(\sum_{j=1}^{m} c_j \texttt{bcirc}(\mathcal{A})^j \cdot \texttt{unfold}(\mathcal{B})\right) \\
&= \sum_{j=1}^{m} c_j \texttt{fold}\big(\texttt{bcirc}\big(\mathcal{A}^j\big) \cdot \texttt{unfold}(\mathcal{B})\big) \\
&= \sum_{j=1}^{m} c_j \texttt{bcirc}\big(\mathcal{A}^j\big) * \mathcal{B} \\
&= \texttt{fold}(\texttt{bcirc}(q(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})).
\end{aligned}$$

Part (ii) is a special case of part (i). As for part (iii), since $p = 1$, we have that $\texttt{fold}(\mathcal{A}) = \texttt{bcirc}(\mathcal{A}) = \mathcal{A} = \texttt{unfold}(\mathcal{A})$, and similarly for $\mathcal{B}$. Then the definition of $f(\mathcal{A}) * \mathcal{B}$ reduces immediately to the matrix function case. Part (iv) follows by carefully unwrapping the definition of $f(\mathcal{A})$:

$$\begin{aligned}
f(\mathcal{A}) * \mathcal{A} &= \texttt{fold}\big(\texttt{bcirc}(\mathcal{A})^{-1}\texttt{unfold}(\mathcal{A})\big) \\
&= \texttt{fold}\Big(\texttt{bcirc}(\mathcal{A})^{-1}\texttt{bcirc}(\mathcal{A})\widehat{\boldsymbol{E}}_1^{np\times n}\Big), \text{by Lemma 6.3(i)} \\
&= \texttt{fold}\Big(\widehat{\boldsymbol{E}}_1^{np\times n}\Big) = \mathcal{I}_{n\times n\times p}.
\end{aligned}$$

Likewise with the other product:

$$\begin{aligned}\mathcal{A} * f(\mathcal{A}) &= \texttt{fold}\big(\texttt{bcirc}(\mathcal{A})\texttt{unfold}\big(\texttt{fold}\big(\texttt{bcirc}(\mathcal{A})^{-1}\texttt{unfold}(\mathcal{I}_{n\times n\times p})\big)\big)\big) \\ &= \texttt{fold}\Big(\texttt{bcirc}(\mathcal{A})\texttt{bcirc}(\mathcal{A})^{-1}\widehat{\boldsymbol{E}}_1^{np\times n}\Big) \\ &= \texttt{fold}\Big(\widehat{\boldsymbol{E}}_1^{np\times n}\Big) = \mathcal{I}_{n\times n\times p}. \qquad \square\end{aligned}$$

We collect further properties of the definition (6.9) that generalize many of the core properties of matrix functions stated in Section 2.2.

Theorem 6.5: Let $\mathcal{A}, \mathcal{D}, \mathcal{X} \in \mathbb{C}^{n\times n\times p}$, so that $\mathcal{X}$ is invertible and $\mathcal{D}$ is f-diagonal with diagonal tubal entries $\boldsymbol{d}_i$, and let $\mathcal{B} \in \mathbb{C}^{n\times s\times p}$. Let $f : \mathbb{C} \to \mathbb{C}$ be defined on a region in the complex plane containing the spectrum of $\texttt{bcirc}(\mathcal{A})$ for (i)-(iii) and on $\texttt{bcirc}(\mathcal{D})$ for (iv). Then it holds that

(i) $f(\mathcal{A})$ commutes with $\mathcal{A}$;

(ii) $f(\mathcal{A}^*) = f(\mathcal{A})^*$;

(iii) $f(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1}) = \mathcal{X} f(\mathcal{A}) \mathcal{X}^{-1}$; and

(iv) $f(\mathcal{D}) * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * f(\boldsymbol{d}_i)$, for all $i = 1, \ldots, n$.

Proof: For all parts, it suffices by Theorem 6.4(ii) to show that the statements hold for $f(z) = \sum_{j=1}^{m} c_j z^j$. Part (i) then follows immediately. To prove part (ii), we need only show that $(\mathcal{A}^j)^* = (\mathcal{A}^*)^j$ for all $j = 0, 1, \ldots$, which follows by induction from Lemma 6.3(v). Part (iii) also follows inductively. The base cases $j = 0, 1$ clearly hold. Assume for some $j = k$, $(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^k = \mathcal{X}(\mathcal{A})^k \mathcal{X}^{-1}$, and then note that

$$\begin{aligned}(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^{k+1} &= (\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^k * (\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1}) \\ &= \mathcal{X} * (\mathcal{A})^k * \mathcal{X}^{-1} * \mathcal{X} * \mathcal{A} * \mathcal{X}^{-1} = \mathcal{X} * (\mathcal{A})^{k+1} * \mathcal{X}^{-1}.\end{aligned}$$

For part (iv), it suffices to show that for all $i = 1, \ldots, n$ and for all $j = 0, 1, \ldots$, $\mathcal{D}^j * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * \boldsymbol{d}_i^j$. For fixed i, the cases $j = 0, 1$ hold, and we assume the statement holds for some $j = k \geq 1$. Then

$$\mathcal{D}^{k+1} * \vec{\mathcal{X}}_i = \mathcal{D} * (\mathcal{D}^k * \vec{\mathcal{X}}_i) = \mathcal{D} * \vec{\mathcal{X}}_i * \boldsymbol{d}_i^k = \vec{\mathcal{X}}_i * \boldsymbol{d}_i^{k+1}. \qquad \square$$

Remark 6.6: Assuming $\mathcal{A}$ has an eigendecomposition $\mathcal{X} * \mathcal{D} * \mathcal{X}^{-1}$ as in (6.6), and assuming f is defined not only on the spectrum of $\texttt{bcirc}(\mathcal{A})$ but also each $\texttt{bcirc}(\boldsymbol{d}_i)$, then by Theorem 6.5 (iii)-(iv) an equivalent definition for $f(\mathcal{A})$ is given as

$$f(\mathcal{A}) = \mathcal{X} * \begin{bmatrix} f(\boldsymbol{d}_1) & & \\ & \ddots & \\ & & f(\boldsymbol{d}_n) \end{bmatrix} * \mathcal{X}^{-1},$$

where the inner matrix takes its elements from the tube fibers (as in Figure 6.1(c)). We further note that the conditions on f are likely redundant, i.e., it seems natural that if f is defined on $\texttt{bcirc}(\mathcal{A})$ then it should also be defined on each $\texttt{bcirc}(\boldsymbol{d}_i)$, but we do not explore this issue further here.

6.3.4 Block diagonalization and the discrete Fourier transform

Per recommendations for tensor computations in [88, 89], we can reduce the computational effort of computing $f(\mathcal{A}) * \mathcal{B}$ by taking advantage of the fact that $\texttt{bcirc}(\mathcal{A})$ can be block diagonalized by the discrete Fourier transform (DFT) along the tubal fibers of $\mathcal{A}$. Let F_p denote the DFT of size $p \times p$. Then we have that

$$(F_p \otimes I_n)\texttt{bcirc}(\mathcal{A})(F_p^* \otimes I_n) = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_p \end{bmatrix} =: D,$$

where D_k are $n \times n$ matrices. Then by Theorem 2.4(iii),

$$f(\texttt{bcirc}(\mathcal{A})) = (F_p^* \otimes I_n) f(D)(F_p \otimes I_n).$$

6.3.5 Communicability of a third-order network

Functions of matrices emerge as measures of centrality and communicability in networks [49]. Given a network– which we regard here as an undirected, unweighted graph with n nodes– we represent the network by its *adjacency matrix* $A \in \mathbb{R}^{n \times n}$. The ijth entry of A is 1 if nodes i and j are connected, and 0 otherwise. As a rule, a node is not connected to itself, so $A_{ii} = 0$. The communicability between nodes i and j is defined as $\exp(A)_{ij}$ [48], and is just one of many ways to measure properties of a network.

These notions can be extended to higher-order situations. Suppose we are concerned instead about triplets, instead of pairs, of nodes. Then it is possible to construct an adjacency tensor $\mathcal{A}$, where a 1 at entry $\mathcal{A}_{ijk}$ indicates that nodes i, j, and k are connected and 0 otherwise. Alternatively, it is not hard to imagine a time-dependent network stored as a tensor, where each frontal face corresponds to a sampling of the network at discrete times. In either situation, we could compute the communicability of a triple as $\exp(\mathcal{A})_{ijk}$, where $\exp(\mathcal{A})$ is our tensor t-exponential.

6.4 Summary and outlook

We have presented only a subset of the abundant applications for matrix functions. Many more are described in, e.g., [74, Chapter 2].

The tensor t-function poses many directions for future work. We present it as a first notion for functions of multidimensional arrays with anticipation that

other definitions are put forth and found to be useful in real-world applications. The tensor t-function $f(\mathcal{A}) * \mathcal{B}$ shows versatility, and the fact that it reduces to a highly structured matrix function problem means that a plethora of tools exist already for understanding its properties.

CHAPTER 7

NUMERICAL EXPERIMENTS

The development of an algorithm is not complete without implementing it and studying its behavior (see Section 1.1). In this chapter, we examine the behavior of the methods developed in the previous chapters. While there are some examples whose results could be used directly in an application, we emphasize that most examples are intentionally devised to allow us to "see inside" the algorithms and understand their properties. We compare different choices for block inner products (reference Table 3.1) and different choices for modifications (i.e., none, harmonic, or Radau-Lanczos). We also look at matrix polynomials explicitly and advocate a way to visualize them that helps elucidate how block Krylov methods interpolate the Ritz values. For the matrix function algorithms, we look at a variety of functions and types of matrices to demonstrate the versatility and robustness of the algorithms, even for functions and matrices that do not satisfy the requirements of the convergence theorems.

7.1 Remarks on implementation

We highlight a number of choices made in the implementation of Algorithms 3.1.1, 3.2.1, 4.2.1, 5.2.1, and 5.2.2, since these choices may affect the observed behavior.

It is well known that breakdowns may occur in the block Arnoldi algorithm (Algorithm 3.1.1). The global inner product has an advantage in this sense. A breakdown (i.e., when $\boldsymbol{W} = 0$ in lines 7 or 15) indicates that the space $\mathscr{K}_m^{\mathtt{Gl}}(A, \boldsymbol{B})$ has reached its maximal size, and the exact solution for $f(A)\boldsymbol{B}$ lies in the space. In the loop-interchange version of Algorithm 3.1.1, a zero in the i-th diagonal position of $\langle\!\langle \boldsymbol{W}, \boldsymbol{W} \rangle\!\rangle_{\mathbb{S}}^{\mathtt{Li}}$ implies that $\mathscr{K}_m(A, \boldsymbol{b}_i)$ has reached its maximal size. We then implement a kind of column deflation so that the ith column is not reused for the next iteration.

Breakdowns occurring with the classical method are more complicated to treat. The scaling quotient of $\boldsymbol{W}$ (see lines 8 or 16) may be exactly or numerically singular, even when the space $\mathscr{K}_m^{\mathtt{Cl}}(A, \boldsymbol{B})$ still has more "room" to grow. The problem is exact or inexact linear dependence among the columns of a basis vector. As discussed in Remark 3.10, we employ [15, Algorithm 7.3], which features block operations and allows us to retain the entire Krylov basis. We also run the classical method without deflation, which is a straightforward implementation of Algorithm 3.1.1.

Adaptive quadrature is used in B(FOM)^2 routines. The quadrature error tolerance is set to be the same as the error tolerance for a given example. Overall error is calculated exactly, since we have access to a machine-accurate solution for each example. In practical scenarios, the approximate error $\widetilde{\boldsymbol{D}}_m^{(k)}$ can be used to check whether the method has converged.

All B(FOM)^2 experiments are run on a Dell desktop with a Linux 64-bit operating system with an Intel®Core$^{\text{TM}}$ i7-4770 CPU @ 3.40 GHz and 32 GB of RAM. In the plots, we abbreviate `clB`(FOM)^2, `glB`(FOM)^2, `liB`(FOM)^2, and (FOM)^2 as `Cl`, `Gl`, `Li`, and `nB`, respectively, where "`nB` " stands for "non-block." All other experiments are run on a Lenovo Thinkpad with a Windows 7 Professional 64-bit operating system with an Intel®Core$^{\text{TM}}$ i7-2760 CPU @ 2.40 GHz and 16 GB of RAM.[1] A package of our routines written in MATLAB can be found at `https://gitlab.com/katlund/bfomfom-main`.[2]

Finally, we remark that we do not consider timings in our experiments, for a number of reasons. On a practical level, it does not make sense, since the experiments have been carried out on two very different machines. Assuming, however, that we had access to the same machine for all experiments, it would still be misleading to report timings. Our code has been written with transparency in mind. That is to say, between the choice of language (MATLAB) and the structure of the code itself, we made little attempt to optimize for speed, instead favoring readability and navigability. The code framework is also relatively general, so that users can easily "plug-in" other functions, quadrature rules, or block inner products.

[1] We would like to take this opportunity to thank the author's laptop for surviving multiple overseas trips, unexpected shutdowns, and everyday abuse.

[2] If the reader happens to be from the not-too-distant future, please contact the author directly for a current version, because who knows where things will be hosted by then.

7.2 Understanding BFOM, BGMRES, and BRL with restarts and shifts

As discussed in Chapters 3 and 4, BFOM and BGMRES are not new methods, but shifted and restarted versions of them are. In this section, we aim to understand their behavior and affirm the developed theory in a variety of different settings. We also take a look at the BRL method as a linear solver and compare it with the BFOM and BGMRES methods.

7.2.1 Diagonal test matrices

We consider a panel of 100×100 diagonal matrices A with different eigenvalue distributions:

1. A is Hermitian positive definite and its spectrum is uniformly spaced in $[10^{-2}, 10^{2}]$;
2. A is Hermitian positive definite and its spectrum is logarithmically spaced in $[10^{-2}, 10^{2}]$;
3. A is Hermitian positive definite with 50 uniformly spaced eigenvalues in $[10^{-2}, 10^{1}]$ and the other 50 in $[10^{1}, 10^{2}]$;
4. A is positive real, its spectrum is symmetric about the real axis, and the real part of its spectrum is uniformly spaced in $[10^{-2}, 10^{2}]$;
5. A is positive real, its spectrum is symmetric about the real axis, and the real part of its spectrum is logarithmically spaced in $[10^{-2}, 10^{2}]$;
6. A is positive real, and the real part of its spectrum is uniformly spaced in $[10^{-2}, 10^{2}]$; and
7. A is positive real, and the real part of its spectrum is logarithmically spaced in $[10^{-2}, 10^{2}]$.

See Figures 7.1 and 7.2 for plots of the eigenvalues of matrix in the complex plane.

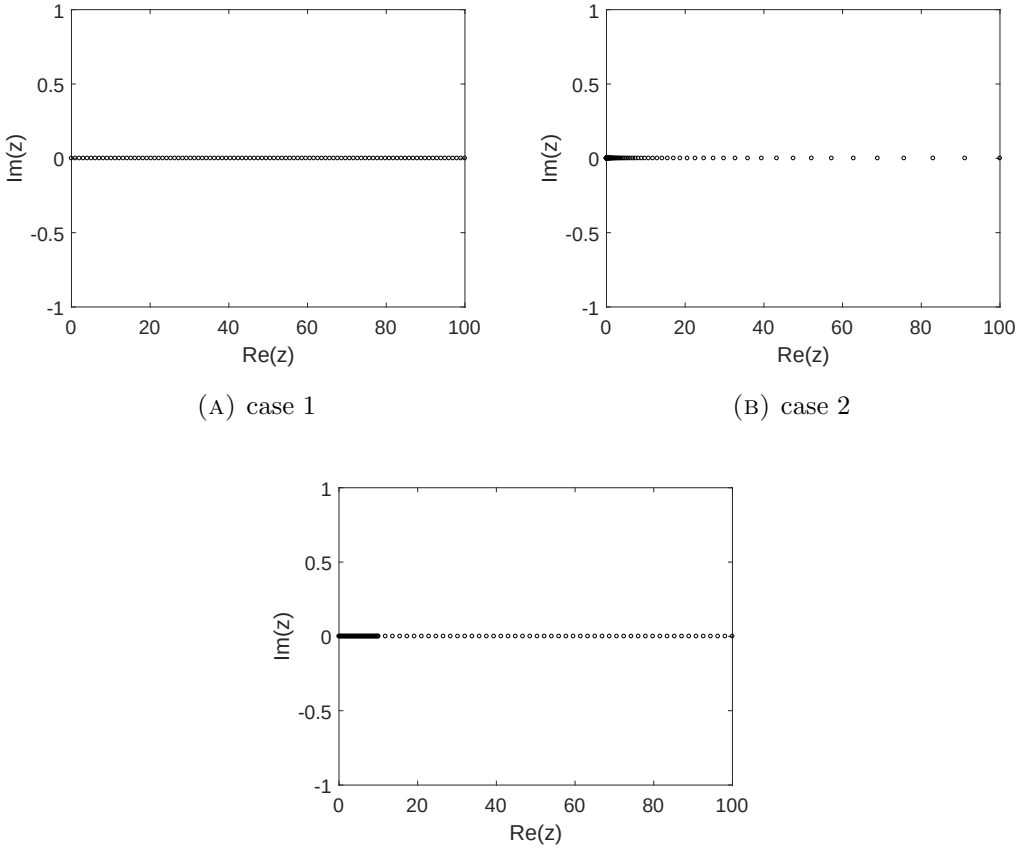
We solve $A\boldsymbol{X} = \boldsymbol{B}$ for a random right-hand side $\boldsymbol{B}$ with $s = 4$ columns; $\boldsymbol{B}$ is the same in every example. We look at the restarted (non-shifted) error $\boldsymbol{E}_m^{(k)} = \boldsymbol{X}^* - \boldsymbol{X}_m^{(k)}$; the restarted shifted residuals $\boldsymbol{R}_m^{(k)}(t) = \boldsymbol{B} - (A+tI)\boldsymbol{X}_m^k(t)$, calculated as $\boldsymbol{B} - (A+tI)\boldsymbol{\mathcal{V}}_m^{(k)}(\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)} + tI)^{-1}\boldsymbol{E}_1 B^k$; and the interpolating matrix polynomials associated to $\boldsymbol{R}_m^{(k)}(0)$ for a specified cycle k.

7.2.2 Shifted residual bounds

We compute the BGMRES and BFOM approximations for cases 1, 2, 3, 5, and 7 with the classical, loop-interchange, and global block inner products to demonstrate that Corollary 4.12 holds. In each of Figures 7.3-7.7, the window on the left shows plots of the error per cycle of the non-shifted approximations, while the one on the right displays the plots of the norm of $\boldsymbol{R}_m^{(k)}(t)$ with respect to t for the third cycle, i.e., $k = 3$. In every case, the cycle length is $m = 10$, and the error tolerance is 10^{-10}.

In every instance, Corollary 4.12 is affirmed, although not as dramatically as one might expect. While this corollary does not lead to very optimistic bounds, we can still be assured that the shifted approximation converges and is bounded by the convergence behavior of the non-shifted approximation. It is also clear that BGMRES does, in fact, generate a minimal residual, at least in comparison to the BFOM residual.

Cases 1 and 3 display the behaviors for uniformly spaced spectra. It is interesting that BGMRES shows so much improvement for the classical method, but not for either the global or loop-interchange methods. However, the addition of more

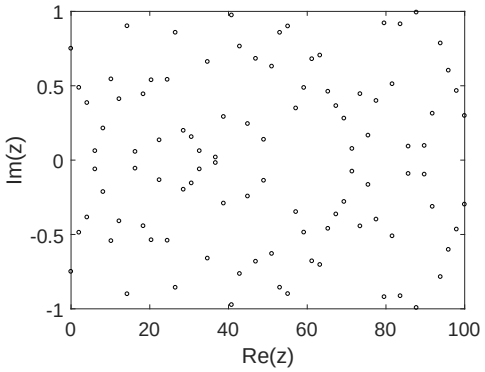


(A) case 1

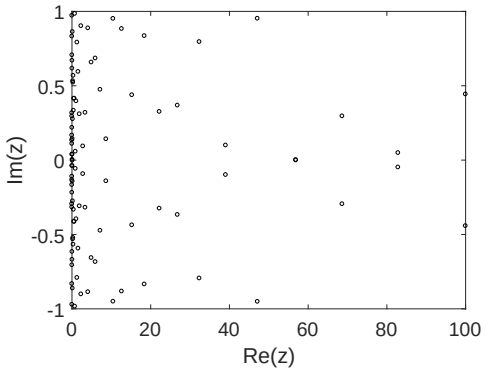
(B) case 2

(C) case 3

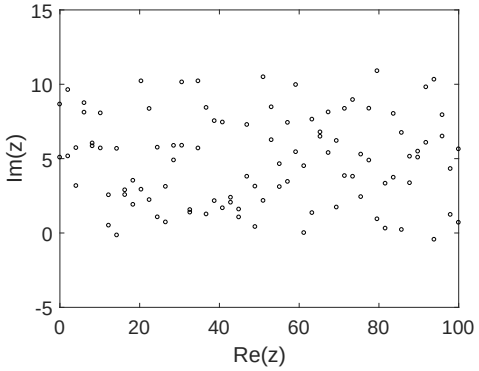
FIGURE 7.1. Eigenvalue distributions for the HPD matrices A. All matrices have condition number 10^4.



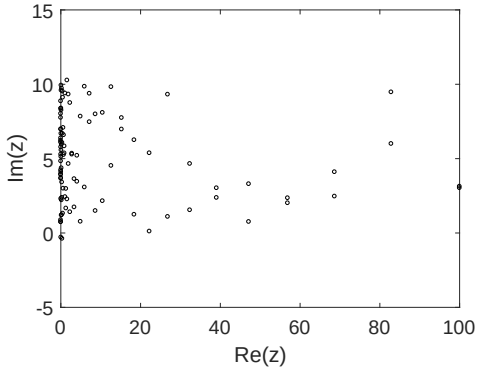
case 4, $\kappa(A) = O(10^2)$



case 5, $\kappa(A) = O(10^3)$



case 6, $\kappa(A) = O(10^2)$



case 7, $\kappa(A) = O(10^3)$

FIGURE 7.2. Eigenvalue distributions for the positive real matrices.

dense eigenvalues near zero (i.e., case 3) makes the problem more challenging for BFOM but does not affect the cycle count as much for BGMRES.

The cases with logarithmically space spectra– case 2 (Figure 7.4), case 5 (Figure 7.6), and case 7 (Figure 7.7)– show how the eigenvalue clustering affects the performance of each method. In case 2, BFOM and BGMRES perform similarly, with BFOM providing only a slight advantage over BGMRES. However, in cases 5 and 7– both of which have nonzero imaginary parts in their spectra–BFOM does not converge at all. Only the classical BGMRES method converges reasonably in either case, since both global and loop-interchange require a high number of restarts.

$\left\|\boldsymbol{E}_m^{(k)}(0)\right\|_{\mathrm{F}}$, vary k $\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathrm{F}}$, vary t

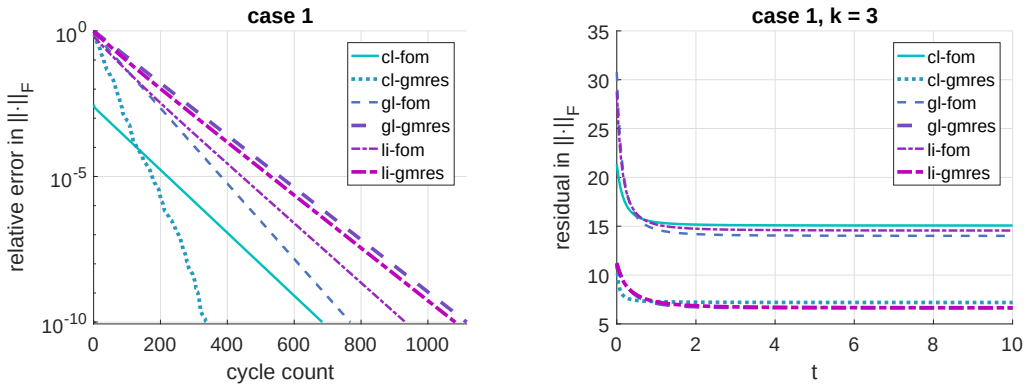


FIGURE 7.3. Convergence plots and shifted residual plots for case 1.

7.2.3 Residual polynomials for BGMRES

The purpose of the examples in this section is to gain insight and intuition for how different block inner products affect the residual polynomials of block Krylov

$\left\|\boldsymbol{E}_m^{(k)}(0)\right\|_{\mathrm{F}}$, vary k $\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathrm{F}}$, vary t

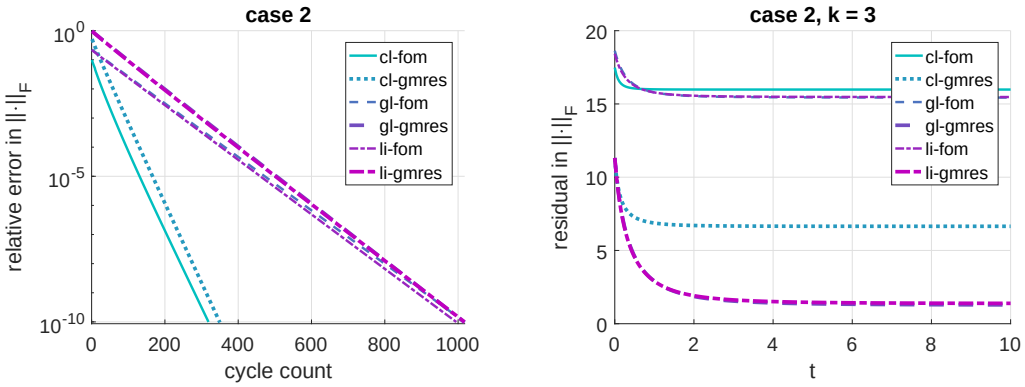


FIGURE 7.4. Convergence plots and shifted residual plots for case 2.

$\left\|\boldsymbol{E}_m^{(k)}(0)\right\|_{\mathrm{F}}$, vary k $\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathrm{F}}$, vary t

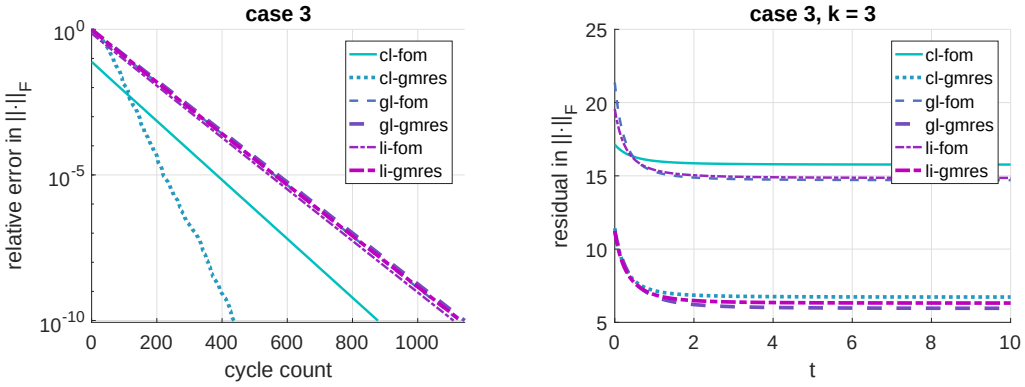


FIGURE 7.5. Convergence plots and shifted residual plots for case 3.

$\left\|\boldsymbol{E}_m^{(k)}(0)\right\|_{\mathrm{F}}$, vary k $\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathrm{F}}$, vary t

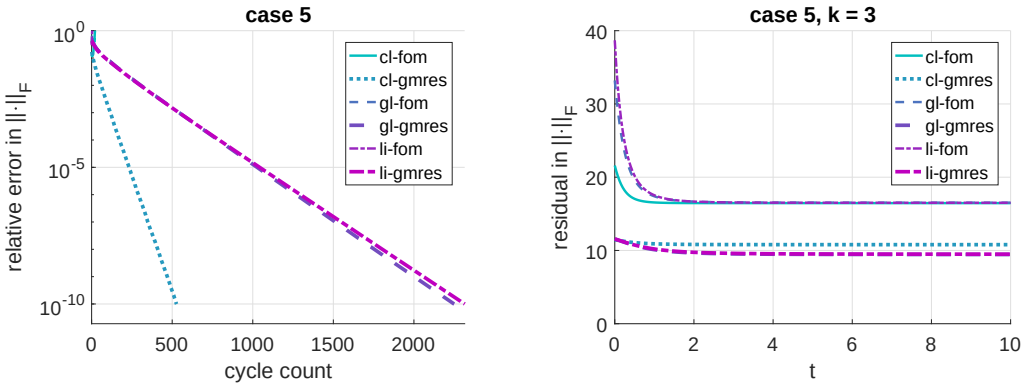


FIGURE 7.6. Convergence plots and shifted residual plots for case 5.

$\left\|\boldsymbol{E}_m^{(k)}(0)\right\|_{\mathrm{F}}$, vary k $\left\|\boldsymbol{R}_m^{(k)}(t)\right\|_{\mathrm{F}}$, vary t

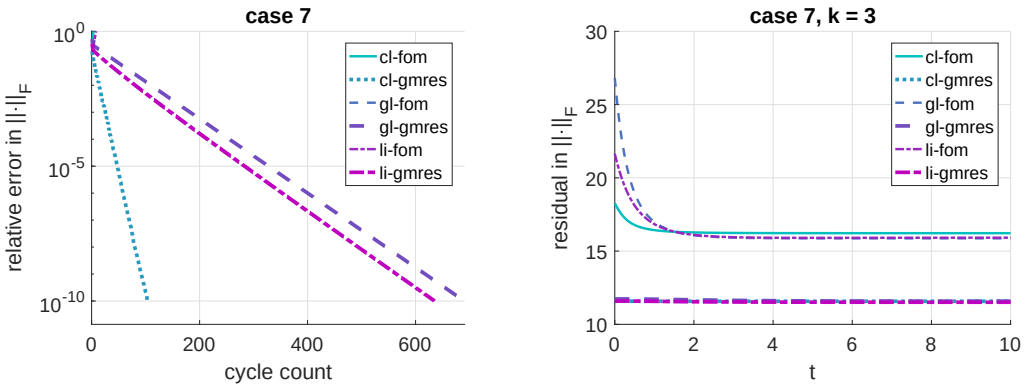


FIGURE 7.7. Convergence plots and shifted residual plots for case 7.

methods. Although we focus on BGMRES polynomials, the insights and conclusions apply to any of the methods considered in Chapters 3 or 4. We do not consider shifts here, and we focus on cases 1, 4, and 6, as they have the simplest spectra. For case 1, we look at the 20th restart cycle; case 4, the 10th cycle; and case 6, the 7th cycle. We again have $m = 10$ and the same right-hand side $\boldsymbol{B}$ as before. There is no particular rationale for the choice of cycle, except that we wanted to go far enough into the restart process to generate polynomials associated to accurate approximations. Recall that by Corollary 4.3, the matrix polynomial associated to the residual is minimal in the $\mathbb{S}$-norm over the space of polynomials of the form $I - zQ(z)$, where $Q \in \mathbb{P}_{m-1}(\mathbb{S})$. It is precisely this minimal polynomial that we look at for each scenario. Also recall from Theorem 4.16 and Remark 4.17 that the latent roots of the residual polynomial coincide with the Ritz values.

In Figure 7.8, we plot the absolute value of the residual polynomial on a range of points containing the spectra of A and of $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$. It is possible to plot in two dimensions, since the spectra of A and $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$ in case 1 are real. In the same window, we plot $|P(\lambda)|$ and $|P(\theta)|$, where $\lambda \in \text{spec}(A)$ and $\theta \in \text{spec}(\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)})$. In the scalar case, it is easy to see that the inequality (4.20) holds.

Cases 4 and 6 necessitate three-dimensional plots, since the matrices are no longer Hermitian and we must therefore plot the polynomial over the complex plane. In Figures 7.9 and 7.10 we plot the absolute value of the determinant of the matrix polynomial, i.e., $|\det(P_m(z))|$, over a set in the complex plane containing the spectra of A and of $\mathcal{H}_m + \mathcal{M}$. Given that the determinant of $P_m(z)$ is itself a polynomial whose roots coincide with the latent roots of $P_m(z)$, $\det(P_m(z))$ is a natural object to

look at. In Figure 7.9, we find that the residual polynomials are symmetric about the real axis, just like the spectrum of A; naturally, in Figure 7.10, there is no symmetry, since spec(A) has none.

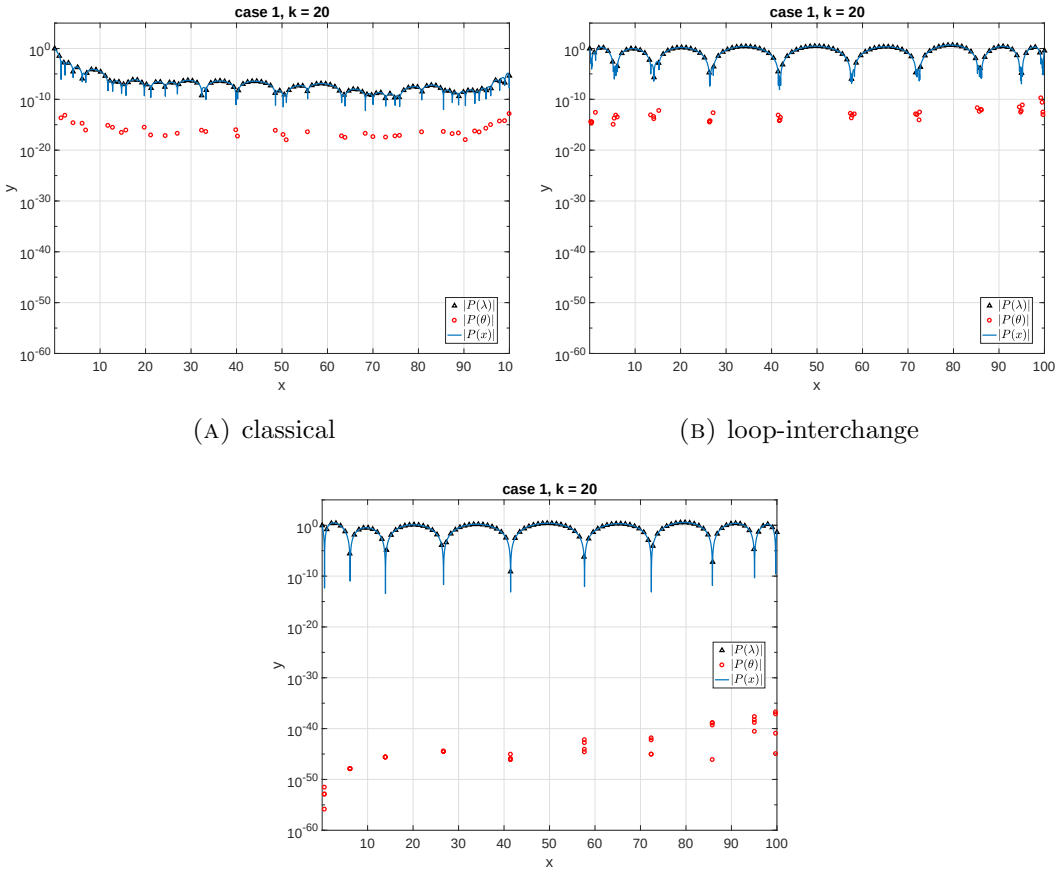
We note too that the scales of the plots in Figures 7.9 and 7.10 differ drastically, and the colorbars on the right only correspond to the plotted surface, not the points. The global matrix polynomials are very accurate on the spectrum of $\mathcal{H}_m^{(k)} + \mathcal{M}^{(k)}$, but overall not as much, especially in comparison to the classical polynomials, which have lower determinant in general.

Comparing Figures 7.8-7.10 across block inner products, we can understand other properties of the residual polynomials. The classical method has the most degrees of freedom and attempts to capture as much of the spectrum of A as possible; for this reason, all the classical polynomials appear to have many zeros. At the other extreme, the global method has the fewest zeros, and each of them has high multiplicity. The loop-interchange method has a mixture of both extremes.

7.2.4 Block Radau-Lanczos as a linear solver

We consider cases 1, 2, and 3, and we choose $S_0 \in \mathbb{S}$ to be $S_0 = \text{diag}(101, 102, 103, 104)$, so that all the eigenvalues fixed by the BRL method are larger than the largest eigenvalue of A. The right-hand side $\boldsymbol{B} \in \mathbb{C}^{n \times s}$ is kept the same as in the previous sections.

The BRL method does appear to improve convergence for small values of m (i.e., the cycle length) when the spectrum of A is uniformly distributed, which can be seen for case 1 in Figure 7.11(A). Plot (B) of the same figure demonstrates that when the spectrum clusters near zero, BRL performs worse for smaller cycle

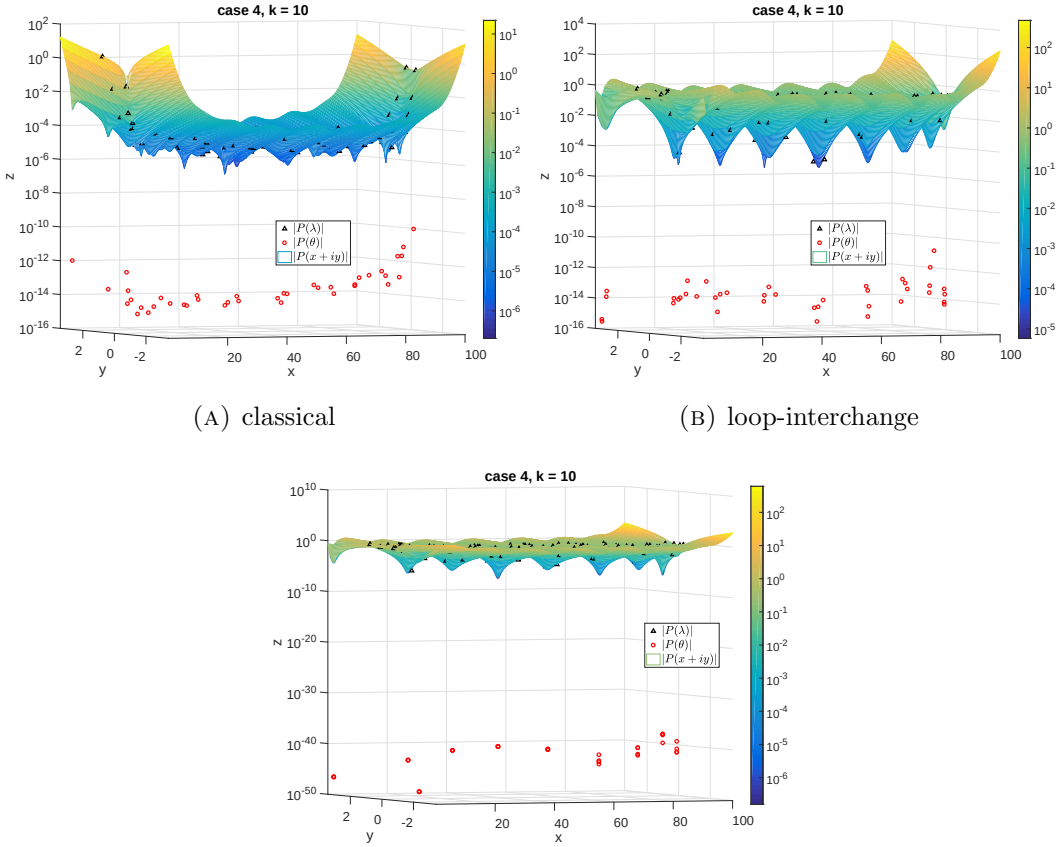


(A) classical

(B) loop-interchange

(C) global

FIGURE 7.8. Case 1 BGMRES residual polynomials

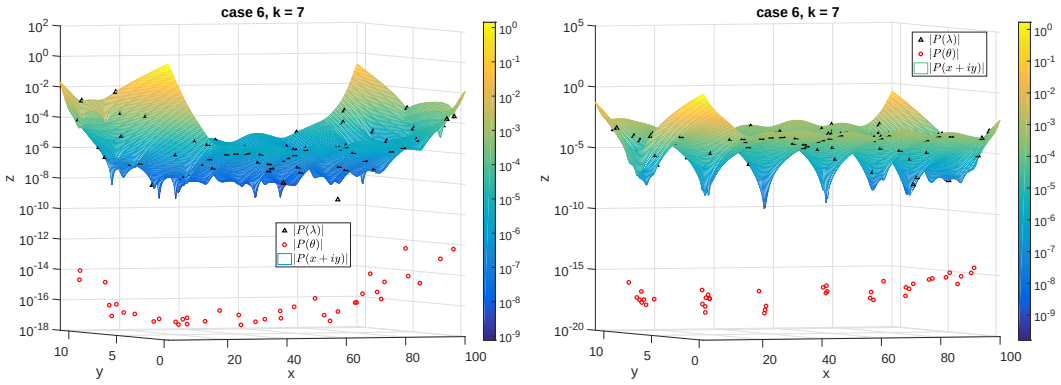


(A) classical

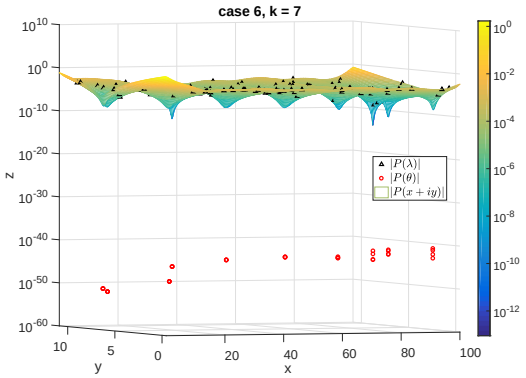
(B) loop-interchange

(C) global

FIGURE 7.9. Case 4 BGMRES residual polynomials



(A) classical (B) loop-interchange



(C) global

FIGURE 7.10. Case 6 BGMRES residual polynomials

lengths. When the spectrum has two sections of differing density, as in case 3, the BRL method performs about the same as BGMRES, shown in plot (C) of the figure. Both plots (B) and (C) challenge the results posited in [54], which shows that the Radau-Lanczos method consistently uses fewer cycles than the FOM-like method to converge. However, our results indicate that the method may sometimes provide benefit when m is small, but otherwise its behavior is similar to that of BGMRES.

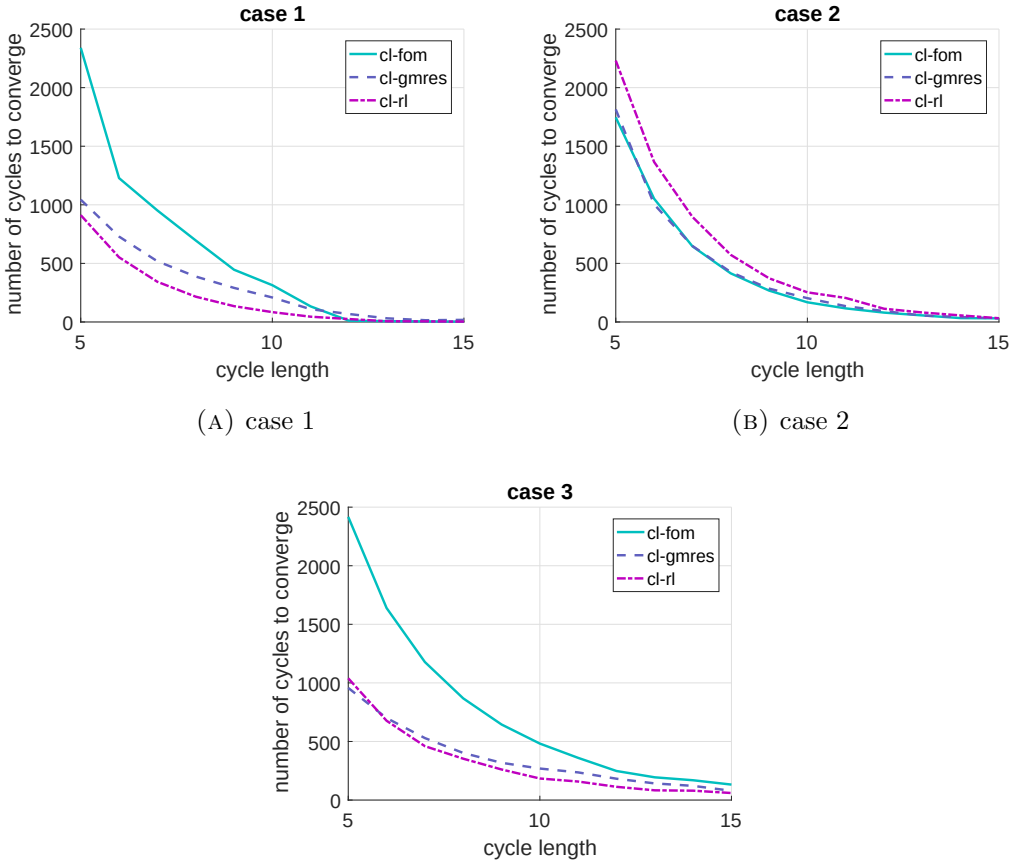
The plots of the residual polynomials in Figure 7.12 shows that the BRL method does fix eigenvalues in the way described in Section 4.3.

7.3 Understanding B(FOM)$^2(m)$

In this section, we recapitulate results from [55] in order to demonstrate the potential for improvement that block methods for matrix functions have over the non-block methods, denoted by (FOM)^2 or the abbreviation "nB." We also examine how the choice of block inner product affects the behavior of B(FOM)$^2(m)$.

7.3.1 B(FOM)^2 on a random tridiagonal HPD matrix

In this example, we compare the bound from Theorem 5.2 with the actual behavior of B(FOM)$^2(m)$ for $f(z) = z^{-1/2}$ acting on a tridiagonal HPD matrix of size 100×100 of condition number $\mathcal{O}(10^2)$. The cycle length m is set to 5, and the error tolerance is set to 10^{-10}. For the hybrid method, $q = 5$; i.e., $\mathbb{S}^{\mathtt{Hy}}$ consists of 10×10 matrices with 5×5 blocks on the diagonal. The solid black line in Figure 7.13 is the theoretical error bound, and it is clear that the bound does a poor job of predicting the true convergence behavior. This is not surprising, since $\xi_m(0)$ is close to 1 in this scenario. We also note that the global, loop-interchange, and non-block versions have nearly

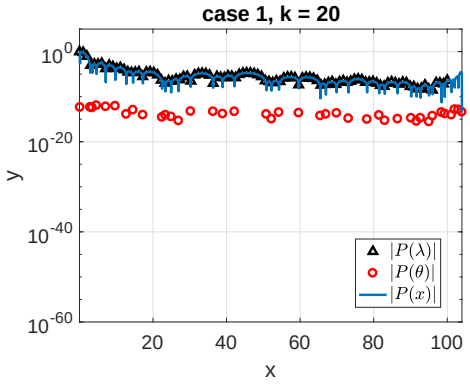


(A) case 1

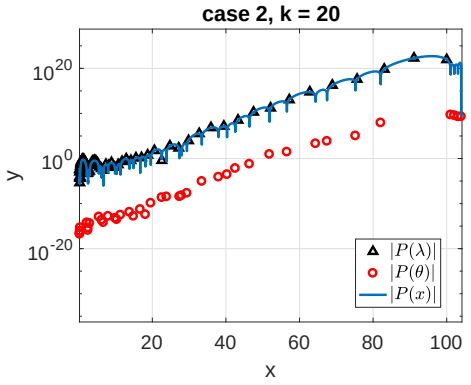
(B) case 2

(C) case 3

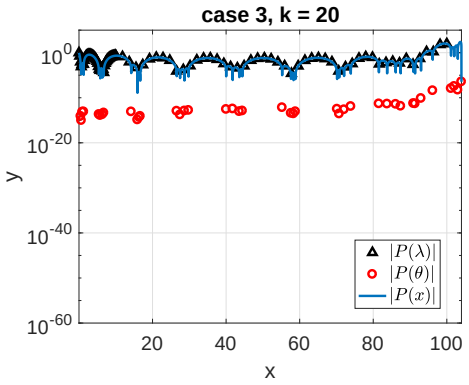
FIGURE 7.11. Cycle length versus number of cycles needed to converge for cases 1-3 and BFOM, BGRMES, and BRL



(A) case 1



(B) case 2



(C) case 3

FIGURE 7.12. Residual polynomial plots for the block Radau-Lanczos method and cases 1-3. Cycle length $m = 10$, and the cycle index $k = 20$.

the same error curves and overlap to form what appears to be one line. (In the case of loop-interchange and non-block, this makes sense, since the two should be identical in exact arithmetic.)

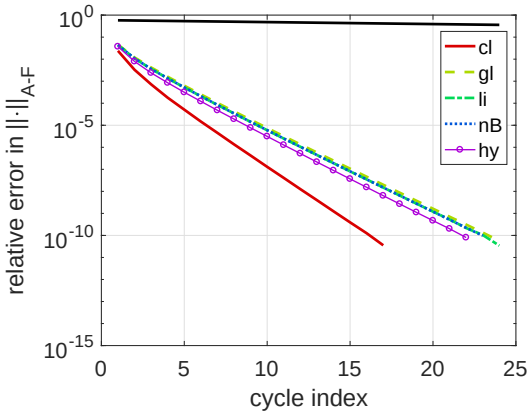


FIGURE 7.13. Convergence history for computing $A^{-1/2}\boldsymbol{B}$, where $A \in \mathbb{C}^{100\times 100}$ is a random tridiagonal HPD matrix, and $\boldsymbol{B} \in \mathbb{C}^{100\times 10}$ is random.

7.3.2 Discretized two-dimensional Laplacian and $f(z) = z^{-1/2}$

We now consider the real, symmetric positive definite matrix arising from the second-order central difference discretization of the negative two-dimensional Poisson equation with 100 grid points in each spatial dimension, so that $n = 10^4$ (see Section 6.1). We compute $f(A)\boldsymbol{B}$ for $f(z) = z^{-1/2}$ and two different $\boldsymbol{B}$ with $s = 10$. The first is given as $\boldsymbol{B} = \boldsymbol{u} \otimes I_{10}$, where $\boldsymbol{u}$ is the vector of dimension 10^3 whose entries are all 1, and I_{10} is the 10×10 identity, making $\boldsymbol{B}$ full rank. The second $\boldsymbol{B}$ is the same as the first, except the first column is a linear combination of the others, leading to linear dependence in the columns of the basis vectors of $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. We also run

two different versions of the classical B(FOM)$^2(m)$, one with deflation as described in Remark 3.10 and the other without. The cycle length $m = 25$, and the error tolerance is set to 10^{-6}. We do not run the hybrid version of B(FOM)$^2(m)$ in either scenario, since it requires an even more complicated deflation routine than classical.

The left plot of Figure 7.14 shows that all methods attain the same accuracy in roughly the same number of cycles. The curves for `cl`B(FOM)^2 with and without deflation overlap, and the remaining three overlap with each other, leading to what appears to be only two curves. Not visible in the plot is that `gl`B(FOM)^2 is slightly less accurate than `li`B(FOM)^2 and (FOM)^2; all are less accurate than either version of `cl`B(FOM)^2, as predicted by the inequalities (3.14).

The right plot of Figure 7.14 displays the results for when linear dependence has been forced into the situation. `cl`B(FOM)^2 without deflation stagnates almost immediately– the code is written so that the process is stopped once the error is no longer decreasing monotonically. We surmise that numerical error allows the process to continue several cycles before an issue is detected. On the other hand, `cl`B(FOM)^2 with deflation converges properly and in much fewer cycles than `gl`B(FOM)^2, `li`B(FOM)^2, or (FOM)^2 (whose curves again appear to overlap).

Although we have promised to avoid discussing runtimes, the extreme nature of the scenario with linear dependence requires commentary. For one, the scenario is highly contrived– it is unlikely that such extreme linear dependence would occur in practice, and in our particular set-up, a user should be able to detect the issue before running the algorithm and adjust accordingly (i.e., by removing the column beforehand and recomputing it afterwards). Secondly, the deflation procedure in our

MATLAB implementation is painfully slow. In the full-rank example, the deflated routine is 20 times slower than the non-deflated routine. Furthermore, neither the global nor the loop-interchange methods require a complicated deflation procedure, and they run in much more reasonable times. In fact, in both scenarios, the global method is the fastest overall: at least 160 times faster than the classical method with deflation in the full-rank case, and over 80 times faster than the classical method with deflation in the rank-deficient case. As such, we do not consider the classical method with deflation in any other example.

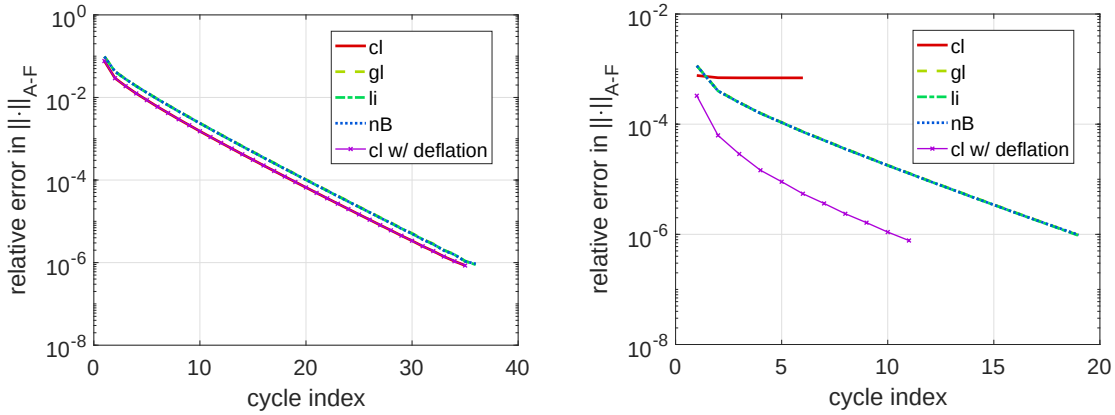


FIGURE 7.14. Convergence history for $A^{-1/2}\boldsymbol{B}$, where $A \in \mathbb{C}^{10^4 \times 10^4}$ is the discretized two-dimensional Laplacian. Left: $\boldsymbol{B} \in \mathbb{C}^{10^4 \times 10}$ has linearly independent columns. Right: the first column of $\boldsymbol{B}$ is a linear combination of other columns.

7.3.3 Overlap Dirac operator and $f(z) = \text{sign}(z)$

We take an 8^4 lattice, such that $n = 12 \cdot 8^4 = 49152$ and $A = Q^2 \in \mathbb{C}^{49152 \times 49152}$.[3] With $\text{sign}(z) = (z^2)^{-1/2}$, we compute $\text{sign}(Q)\widehat{\boldsymbol{E}}_1$ as $A^{-1/2}\boldsymbol{B}$, where $\boldsymbol{B} = Q\widehat{\boldsymbol{E}}_1 \in \mathbb{C}^{49152 \times 12}$. The error tolerance is set to 10^{-6}, and the cycle length is varied, i.e., $m \in \{25, 50, 100, 150\}$. The hybrid method is included in this example, with $q = 4$.

Regarding m as the number of basis vectors that can be stored per cycle, it is not surprising that as m increases, all the methods require fewer cycles to converge, as shown in Figure 7.15. However, it appears that no method particularly benefits from the additional information provided by more basis vectors. In such scenarios, the global method should be preferred.

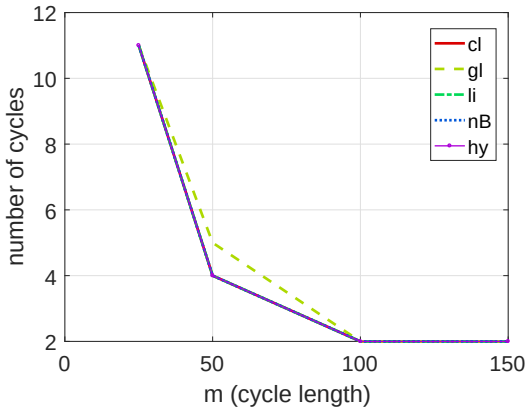


FIGURE 7.15. Number of cycles versus the cycle length for the overlap Dirac operator example.

[3]We acknowledge Marcel Schweitzer for the providing the resulting matrix, which was initially provided to him by Björn Leder. Both are formerly affiliated with the Bergischen Universität Wuppertal.

7.3.4 Convection-diffusion equation and $\exp(z)$

We now consider the action of the exponential on a panel of matrices with different degrees of non-symmetry. The matrices correspond to the standard finite differences discretization of a two-dimensional convection-diffusion equation on $[0,1]\times[0,1]$ with a constant convection field and convection parameter ν (see Section 6.1). We use 350 discretization points in each dimension and a scaling parameter of $2\cdot 10^{-3}$, resulting in matrices of size $350^2 \times 350^2 = 122,500 \times 122,500$. We look at three matrices A_ν, for $\nu \in \{0, 100, 200\}$. When $\nu = 0$, A_ν is real symmetric; otherwise, A_ν is non-symmetric.

Figure 7.16 displays the results for each ν. Although the classical method is slightly more accurate than the others, all require the same number of cycles to converge, with the number of cycles increasing as ν increases.

7.4 Understanding B(FOM)^2+har(m)

In [52], the harmonic modification is shown to rectify some of the convergence problems that the non-modified method has for particular matrices. We reproduce two such examples here, considering also how different block inner products affect the resulting behavior.

7.4.1 A circulant

Let A be a circulant matrix, as in (7.1), of dimension 1001×1001 and with $\alpha = 1.01$. This matrix is of the same type as the one considered in [52, Section 7]. In this paper, the authors consider a 21×21 matrix, which leads to trivial convergence for

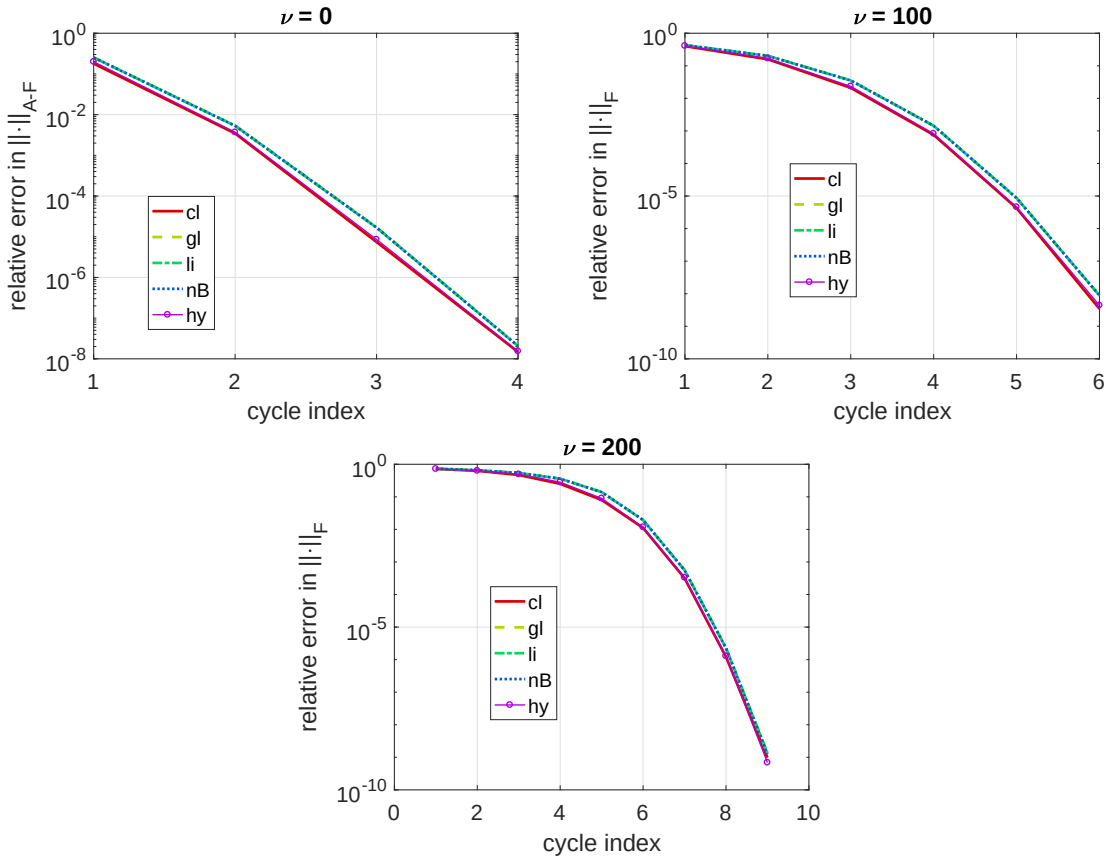


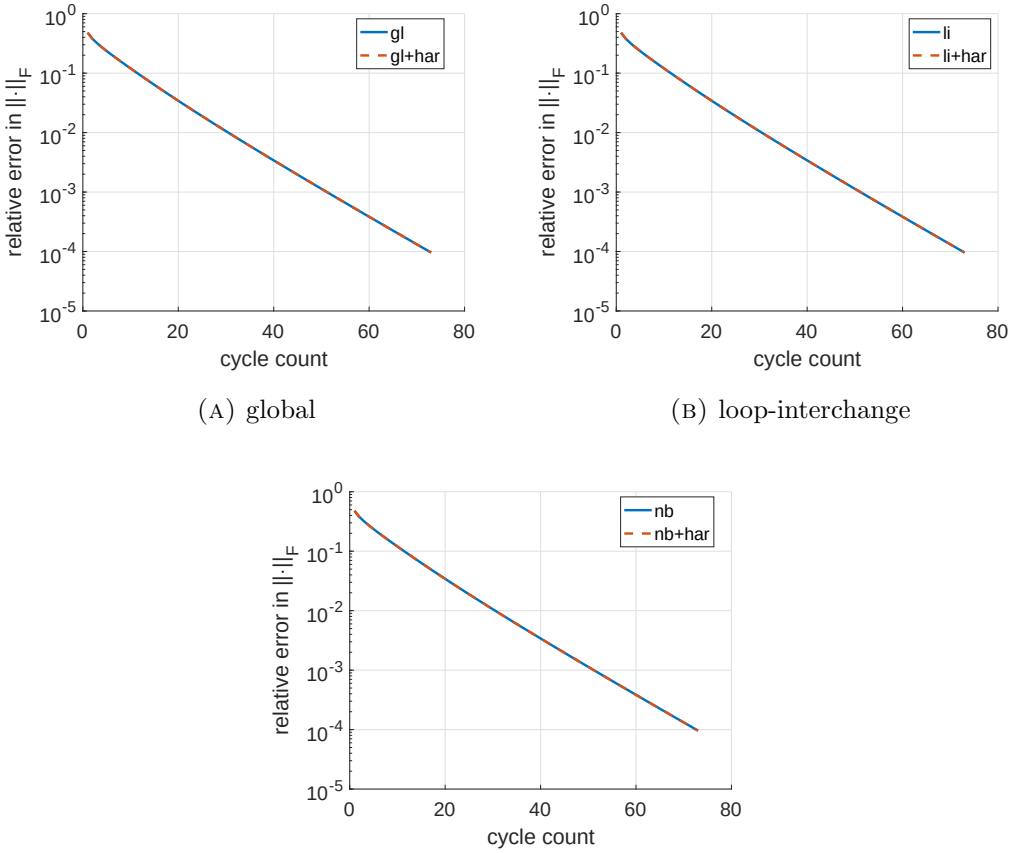
FIGURE 7.16. Convergence histories for computing $\exp(A_\nu)\boldsymbol{B}$, where $A_\nu \in \mathbb{C}^{122,500\times 122,500}$ is the finite differences stencil of a two-dimensional convection-diffusion equation with varying convection parameters $\nu \in \{0, 100, 200\}$, and $\boldsymbol{B} \in \mathbb{C}^{122,500\times 10}$ has random entries.

our block methods. Taking $\alpha > \cos(\frac{2\pi}{n})$ ensures that A is positive real, so that the convergence theory from Section 5.2.2 applies.

$$A = \begin{bmatrix} \alpha & 0 & \cdots & 0 & 1 \\ 1 & \alpha & 0 & \cdots & 0 \\ 0 & 1 & \alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & \alpha \end{bmatrix} \tag{7.1}$$

As discussed in [52] and [126], unmodified FOM-based methods tend to stagnate while approximating $f(A)\boldsymbol{b}$, when f is a Stieltjes function and $\boldsymbol{b} = \widehat{\boldsymbol{e}}_1$; harmonic methods, however, provide a remedy. One might expect similar behavior for our block methods. We only show results for the global and loop-interchange methods, and compare them to the non-block method of computing each column of $A^{-1/2}B$ in serial, where $\boldsymbol{B} \in \mathbb{C}^{1001\times 10}$ is the first ten columns of the identity. We omit results from the classical and hybrid methods, since neither would converge for this problem, even with deflation.

Figure 7.17 displays the convergence results for each method, with $m = 10$ as the cycle length. The harmonic modification provides improvement in no case; in fact, unlike the example in [52], the unmodified method converges in every case. The reason is that the information from additional columns overcomes the anticipated problems with convergence, thus rendering a modification unnecessary. We also point out that all methods require exactly the same number of cycles to converge. In such situations, the global method is the method of choice, since it uses sparse Level 3 BLAS operations, unlike the non-block method, and it is cheaper than the loop-interchange method in terms of storage and floating point operations.



(A) global

(B) loop-interchange

(C) non-block

FIGURE 7.17. Convergence plots for Section 7.4.1, where $A \in \mathbb{C}^{1001\times 1001}$ is a circulant matrix.

7.4.2 A nonnormal and nondiagonalizable

We consider another example from [52, Section 7], matrices of size 1000×1000 with Jordan blocks of the following form on the diagonal:

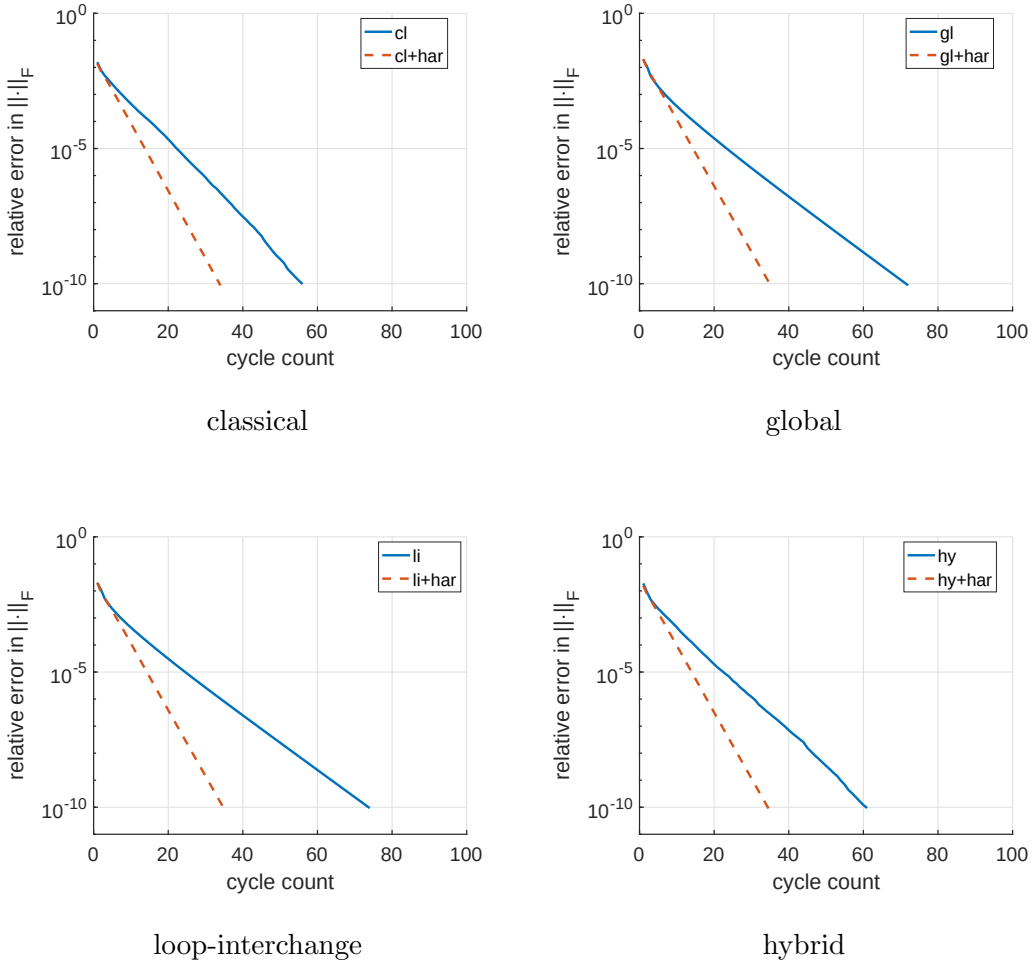
$$\begin{bmatrix} \lambda & 0 \\ 1 & \lambda \end{bmatrix}$$

Such matrices are nonnormal and nondiagonalizable, but they are still positive real if $\mathrm{Re}(\lambda) > 0.5$. We regard λ as a random variable and consider two such A:

- A_1: $\mathrm{Re}(\lambda)$ is uniformly distributed in $[0.6, 0.8]$, and $\mathrm{Im}(\lambda)$ is uniformly distributed in $[-10, 10]$;
- A_2: $\mathrm{Re}(\lambda)$ is uniformly distributed in $[0.5001, 0.5099]$, and $\mathrm{Im}(\lambda)$ is uniformly distributed in $[-10, 10]$.

The second matrix is slightly less well conditioned than the first.

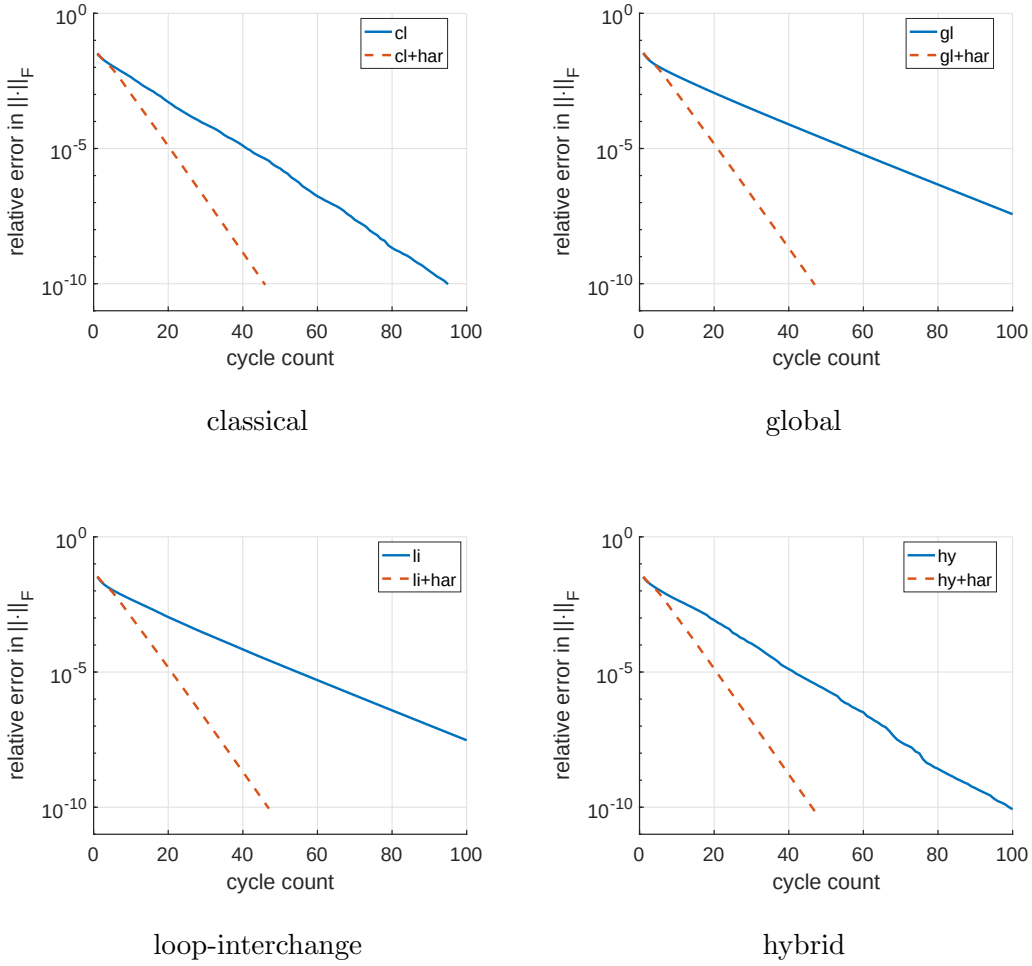
We take $\boldsymbol{B} \in \mathbb{C}^{1000\times 4}$ to be random with $\boldsymbol{B}^*\boldsymbol{B} = I_4$, $m = 10$. As shown in Figures 7.18 and 7.19, the harmonic modification always reduces the total number of cycles. The effect is stronger for A_2, which is has a higher condition number than A_1. Another interesting feature is that the global and loop-interchange methods appear to benefit the most from the modification, in the sense that reduction in the number of cycles is greater for them than for the classical and hybrid methods, in comparison to their respective unmodified versions.



classical global

loop-interchange hybrid

FIGURE 7.18. Convergence plots for A_1 in Section 7.4.2, where A_1 is nonnormal and nondiagonalizable.



classical global

loop-interchange hybrid

FIGURE 7.19. Convergence plots for A_2 in Section 7.4.2, where A_2 is nonnormal and nondiagonalizable.

7.4.3 Tensor t-exponential

We take $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ to be a tensor whose p frontal faces are each adjacency matrices for an undirected, unweighted network, i.e., the frontal faces of $\mathcal{A}$ are symmetric, and the entries are binary. The sparsity structure of this tensor is given in Figure 7.20 for $n = p = 50$. Note that we must actually compute $\exp(\mathcal{A}) * \mathcal{I} = \texttt{fold}\Big(\exp(\texttt{bcirc}(\mathcal{A}))\widehat{\boldsymbol{E}}_1\Big)$ (see Definition (6.10)). With $n = p = 40$, this leads to a 1600×1600 matrix function times a 1600×40 block vector. The sparsity patterns of $\texttt{bcirc}(\mathcal{A})$ and D, where $\mathcal{D}$ is from the eigendecomposition of $\mathcal{A}$, are shown in Figure 7.21. Note that $\texttt{bcirc}(\mathcal{A})$ is not symmetric, but it has a nice banded structure. It should also be noted that while the blocks of $\mathcal{D}$ appear to be structurally identical, they are not numerically equal. This structure is a result of the discrete Fourier transform.

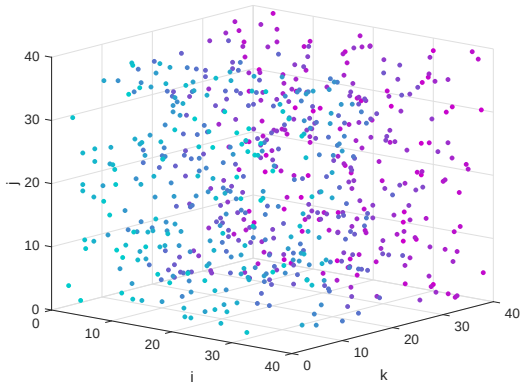
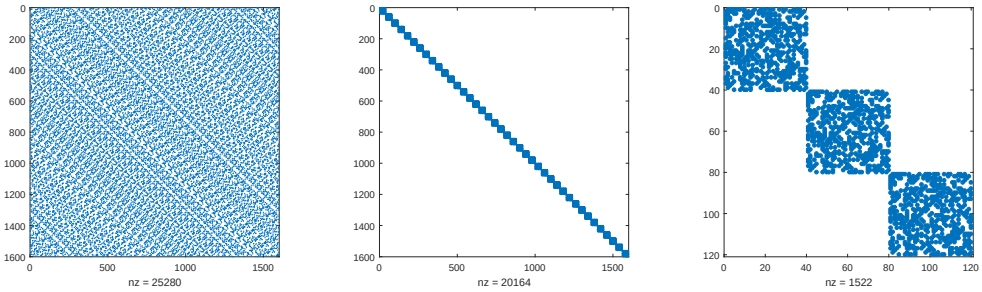


FIGURE 7.20. Sparsity structure for $\mathcal{A}$. Blue indicates that a face is closer to the "front" and pink farther to the "back"; see Figure 6.1(f) for how the faces are oriented.



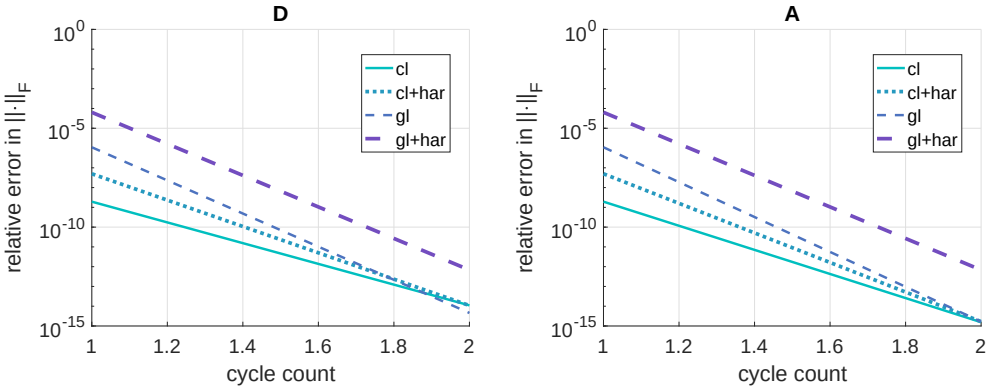
`bcirc`($\mathcal{A}$) D zoom in on D

FIGURE 7.21. Sparsity patterns for block circulants

We compute $\exp(\mathcal{A}) * \mathcal{I}$ with Algorithms 5.2.1 and 5.2.2, both with the classical and global block inner products. The convergence behavior of each version is displayed in Figure 7.22. The restart cycle length is $m = 15$, and the error tolerance is 1e-12. Despite the pathological behavior known to occur with FOM-like methods acting on circulant-type matrices [126], the BFOM methods do not suffer here. In fact, the BFOM methods converge just as well as the block harmonic methods. The methods based on D (case (A)) are only a little less accurate than those based on `bcirc`($\mathcal{A}$) (case (B)), and they require the same number of iterations.

7.5 Understanding B(FOM)^2+`rad`(m)

We run classical B(FOM)^2+`rad`(m) here, with $f(z) = z^{-1/2}$, A as in cases 1 and 3 from Section 7.2.1, and $\boldsymbol{B}$ as the same random block vector from Section 7.2.1. The prescribed block eigenvalue S_0 is the same as in Section 7.2.4. We seek an accuracy of 10^{-4} and limit the maximum number of cycles to 100. While varying the cycle



(A) (B)

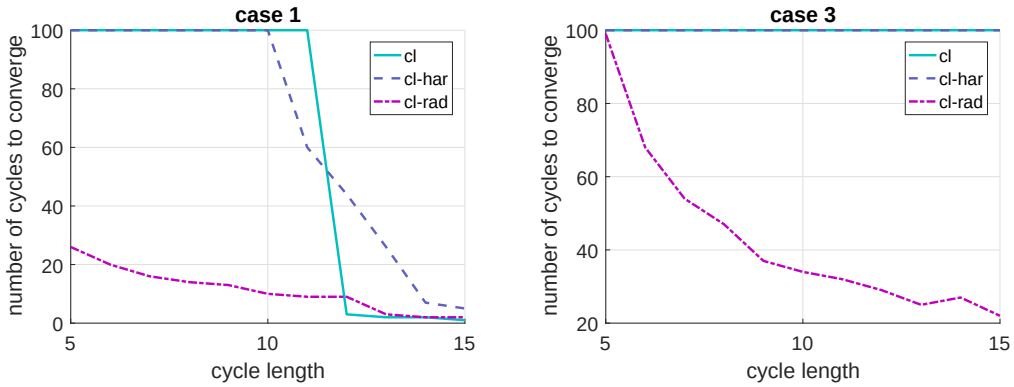
FIGURE 7.22. Convergence plots for (A) classical and global methods on $\exp(D)F_p \otimes I_n \widehat{\boldsymbol{E}}_1$, and (B) classical and global methods on $\exp(\texttt{bcirc}(\mathcal{A}))\widehat{\boldsymbol{E}}_1$

length m from 5 to 15, we compare the convergence behavior of $\texttt{cl}$B(FOM)^2+$\texttt{rad}(m)$ with that of $\texttt{cl}$B(FOM)$^2(m)$ and $\texttt{cl}$B(FOM)^2+$\texttt{har}(m)$.

Figure 7.23 displays the results. The BRL modification leads to significant improvement in both cases, most notably in the second, for which neither the unmodified nor the harmonic methods converges in fewer than 100 cycles for any m.

7.6 Summary and outlook

The experiments of this chapter lead to a number of interesting and surprising results. The main result is an affirmation that theory developed in Chapters 3 and 4 is indeed practical and correct. Our framework provides many variations of BFOM, BGMRES, BRL, B(FOM)^2, and B(FOM)^2+$\texttt{mod}$ that prove to be efficient, robust, and stable in a plethora of situations.



(A) case 1 (B) case 3

FIGURE 7.23. Cycle length versus number of cycles needed to converge for $f(A)\boldsymbol{B}$, where $f(z) = z^{-1/2}$ and A and $\boldsymbol{B}$ from Section 7.2.1.

One of the most surprising results is how effective the global version of these algorithms is. It is the easiest method to implement, requiring no complicated deflation routine nor multiple switches for handling different breakdown scenarios. It is also often the fastest method in terms of runtime, at least in MATLAB on the machines we used. If a user needs to compute $f(A)\boldsymbol{B}$ blindly, `gl`B(FOM)^2 is perhaps the best starting point.

Also surprising is that B(FOM)^2+`rad` outperforms B(FOM)^2+`har` in terms of reducing cycle counts relative to the non-modified method. Such results merit further study for the BRL method for computing matrix functions.

We also highlight the potential that matrix polynomials have as analytical tools. Computing and visualizing matrix polynomials is not computationally intensive and provides insight into how block methods interact with the spectrum of A.

As noted earlier, the software used in this chapter is hosted publicly, and we encourage interested researchers to make their own contributions. A number of implementation issues remain open, but they are beyond the scope of this work. A long-term goal, however, is to provide implementations of B(FOM)^2(m) and modified versions (especially B(FOM)^2+`har`(m)) that are optimized for speed, memory movement, and parallelism, especially now that many attributes of the algorithms are well understood. We also hope that a large-scale comparative study could be conducted with other state-of-the-art methods (as in, e.g., [5, 141]) in order to make recommendations for a wide range of scenarios.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

This dissertation serves two important functions. On the one hand, it proposes a comprehensive and, in many ways, exhaustive framework for understanding block Krylov subspace methods; and on the other hand, it uses this framework as a foundation on which to build analysis for restarted methods for families of shifted systems and general-purpose methods for matrix functions. A number of tools are employed for analyzing the methods and obtaining error bounds, including matrix polynomials, matrix derivatives, lesser known properties of products of Hermitian positive definite and positive real matrices, and so forth. Although the error bounds developed for matrix functions are only shown for Stieltjes functions of matrices, it is not hard to extend these results to functions of the more general Cauchy-Stieltjes form, in analogy to the techniques of [53].

A number of our intermediate results merit further attention. The interpolating matrix polynomials of Section 2.5.2 and the block Arnoldi polynomial relation of Section 4.1 allow us to characterize block Krylov subspace methods from a matrix polynomial viewpoint and look for approximations to $f(A)\boldsymbol{B}$ in terms of matrix polynomials. The block Arnoldi polynomial relation also characterizes all the possible

s-rank modifications to the block upper Hessenberg matrix $\mathcal{H}_m$ so that the associated approximation $\boldsymbol{\mathcal{V}}_m(\mathcal{H}_m + \mathcal{M})^{-1}\widehat{\boldsymbol{E}}_1 B$ lies in $\mathscr{K}_m^{\mathbb{S}}(A, \boldsymbol{B})$. We have shown that block GMRES can be cast as a modified block FOM method, along with the new block Radau-Lanczos method. Both are translated into new methods for computing $f(A)\boldsymbol{B}$.

The general nature of our framework is valuable for describing block Krylov methods not only in a theoretical sense, but also in a computational sense. The choice of the *-subalgebra $\mathbb{S}$ and block inner product directly affects the computational effort, as well as how information is shared between columns of the block vectors of the Krylov basis, and ultimately the accuracy per cycle. The examples we discussed (classical, hybrid, loop-interchange, and global; see Table 3.1) cover a spectrum of possibilities: the classical method shares all the information available among all the columns, is the most computationally expensive, and produces the most accurate approximations per cycle; in contrast, the global method does not allow columns to talk to each other, is the cheapest and most easily implemented, and produces the least accurate approximations per cycle. The hybrid and loop-interchange methods have attributes in between these two extremes. Different applications may benefit from a particular choice of inner product, especially depending on the computational resources available.

The proof techniques leading to the results of Theorems 4.14 and 4.20 are themselves novel and may prove useful for future researchers. Further work could lead to additional insights for shifted BFOM or BGMRES with restarts, and perhaps also for a shifted BRL method with restarts.

The variety of numerical examples in Chapter 7 demonstrates how versatile our methods for matrix functions are, and how visualizing the matrix polynomial associated to a block Krylov method can elucidate its behavior and properties. Methods with the block Radau-Lanczos or block GMRES-like modifications (both for linear systems and matrix functions) are shown to reduce the number of cycles needed to converge in many cases, thus providing viable options when block FOM-like methods are too slow or divergent. We mention again that the block Radau-Lanczos method may prove to be the key to a more accurate error approximation and built-in stopping criteria for the unmodified B(FOM)$^2(m)$ (Algorithm 5.2.1).

Our numerical results also indicate that global methods may be best for computing $f(A)\boldsymbol{B}$, at least in the context of the types of functions and matrices we have considered. As noted already, it is the simplest method to implement and would serve as a good starting point for researchers trying to determine which method is best for their problem, since it is fast and would provide feedback quickly about convergence.

We have also presented a surprising new application for block Krylov methods for matrix functions: the tensor t-function. We have shown that the tensor t-function retains many of the same properties as matrix functions, and its computation reduces to the action of a function of a block circulant matrix on a block vector. A simple generalization of some concepts in network theory points to possible real-life applications of this object. While B(FOM)$^2(m)$ proves to be a viable method for computing the tensor t-function, we emphasize that further study is needed for a larger class of

tensors and functions. For example, there exist t-Krylov methods [88] that may be competitive.

While the primary goal of this dissertation is to establish new methods for computing $f(A)\boldsymbol{B}$, it inadvertently poses many open questions. At this stage in the development of iterative methods for $f(A)\boldsymbol{B}$, many methods exist with many options and parameters, but it is not immediately clear which methods perform better than others. Comprehensive tests have been conducted for the exponential [5, 141] for a subset of methods and matrices, but additional tests are needed for other functions and matrices. A particular challenge is determining the ideal way to incorporate blocking techniques, e.g., how many columns should $\boldsymbol{B}$ have with respect to the size of A, which block inner product will lead to the speediest convergence, etc. Some work for linear systems has already been done [11, 15, 111], but the additional complexity of matrix functions, particularly storing and updating the cospatial factors involved in Algorithms 5.2.1 and 5.2.2, makes this issue all the more difficult. Lastly, thorough investigations of the operation counts of these algorithms and acceleration techniques such as eigenvalue deflation [40] should be conducted to make our methods even better understood and more robust.

We put forth this body of work as a launching point. We have discovered much, but there is yet much to learn!

FIGURE 8.1. The author, deep in reflection.

BIBLIOGRAPHY

[1] Oussama Abidi, Mohammed Heyouni, and Khalide Jbilou. On some properties of the extended block and global Arnoldi methods with applications to model reduction. *Numerical Algorithms*, 75(1):285–304, 2017.

[2] Mohamed Addam, Mohammed Heyouni, and Hassane Sadok. The block Hessenberg process for matrix equations. *Electronic Transactions on Numerical Analysis*, 46:460–473, 2017.

[3] Martin Afanasjew, Michael Eiermann, Oliver G. Ernst, and Stefan Güttel. Implementation of a restarted Krylov subspace method for the evaluation of matrix functions. *Linear Algebra and its Applications*, 429(10):229–314, 2008.

[4] Awad H. Al-Mohy. A new algorithm for computing the actions of trigonometric and hyperbolic matrix functions. Technical report, 2017.

[5] Awad H. Al-Mohy and Nicholas J. Higham. Computing the action of the matrix exponential with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511, 2011.

[6] Awad H. Al-Mohy, Nicholas J. Higham, and Samuel D. Relton. New algorithms for computing the matrix sine and cosine separately or simultaneously. *SIAM Journal on Scientific Computing*, 37(1):A456–A487, 2015.

[7] José I. Aliaga, Daniel L. Boley, Roland W. Freund, and Vicente Hernández. A Lanczos-type method for multiple starting vectors. *Mathematics of Computation*, 69(232):1577–1601, 2000.

[8] Mary Aprahamian and Nicholas J. Higham. Matrix inverse trigonometric and inverse hypoerbolic functions: theory and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 37(4):1453–1477, 2016.

[9] Francesca Arrigo, Michele Benzi, and Caterina Fenu. Computation of generalized matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 37(3):836–860, 2016.

[10] James Baglama. Dealing with linear dependence during the iterations of the restarted block Lanczos methods. *Numerical Algorithms*, 25:23–36, 2000.

[11] Allison H. Baker, John M. Dennis, and Elizabeth R. Jessup. On improving linear solver performance: a block variant of GMRES. *SIAM Journal on Scientific Computing*, 27(5):1608–1626, 2006.

[12] Tania Bakhos, Peter K. Kitanidis, Scott Ladenheim, Arvind K. Saibaba, and Daniel B. Szyld. Multipreconditioned GMRES for shifted systems. *SIAM Journal on Scientific Computing*, 39(5):S222–S247, 2017.

[13] Fatemeh Panjeh Ali Beik and Davod Khojasteh Salkuyeh. On the global Krylov subspace methods for solving general coupled matrix equations. *Computers and Mathematics with Applications*, 62(12):4605–4613, 2011.

[14] Josef Betten. *Creep mechanics*. Springer, Berlin, 3rd edition, 2008.

[15] Sebastian Birk. *Deflated shifted block Krylov subspace methods for Hermitian positive definite matrices*. PhD thesis, Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, 2015.

[16] Sebastian Birk and Andreas Frommer. A deflated conjugate gradient method for multiple right hand sides and multiple shifts. *Numerical Algorithms*, 67(3):507–529, 2014.

[17] Jacques C. R. Bloch, Andreas Frommer, Bruno Lang, and Tilo Wettig. An iterative method to compute the sign function of a non-hermitian matrix and its application to the overlap Dirac operator at nonzero chemical potential. *Computer Physics Communications*, 177:933–943, 2007.

[18] R. Bouyouli, Khalide Jbilou, R. Sadaka, and Hassane Sadok. Convergence properties of some block Krylov subspace methods for multiple linear systems. *Journal of Computational and Applied Mathematics*, 196(2):498–511, 2006.

[19] Karen Braman. Third-order tensors as linear operators on a space of matrices. *Linear Algebra and Its Applications*, 433(7):1241–1253, 2010.

[20] Charles George Broyden. A breakdown of the block CG method. *Optimization Methods and Software*, 7(1):41–55, 1997.

[21] Henri Calandra, Serge Gratton, Julien Langou, Xavier Pinel, and Xavier Vasseur. Flexible variants of block restarted GMRES methods with application to geophysics. *SIAM Journal on Scientific Computing*, 34(2):A714–A736, 2012.

[22] Jie Chen, Mihai Anitescu, and Yousef Saad. Computing $f(A)b$ via least squares polynomial approximations. *SIAM Journal on Scientific Computing*, 33(1):195–222, 2011.

[23] Andrzej Cichocki. Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. Technical Report arXiv:1403.2048v4, 2014.

[24] Jane K. Cullum and William E. Donath. A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In *1974 IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*, volume 13, pages 505–509, 1974.

[25] David Damanik, Alexander Pushnitski, and Barry Simon. The analytic theory of matrix orthogonal polynomials. *Surveys in Approximation Theory*, 4:1–85, 2008.

[26] Dean Darnell, Ronald B. Morgan, and Walter Wilcox. Deflated GMRES for systems with multiple shifts and multiple right-hand sides. *Linear Algebra and its Applications*, 429(10):2415–2434, 2008.

[27] Philip J. Davis. *Circulant Matrices*. AMS Chelsea Publishing, Providence, 2nd edition, 2012.

[28] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, Boston, 2nd edition, 1984.

[29] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[30] John E. Dennis, Joseph F. Traub, and Robert P. Weber. On the matrix polynomial, lambda-matrix and block eigenvalue problems. Technical Report CMU-CS-71-110, Carnegie-Mellon University, 1971.

[31] John E. Dennis, Joseph F. Traub, and Robert P. Weber. The algebraic theory of matrix polynomials. *SIAM Journal on Numerical Analysis*, 13(6):831–845, 1976.

[32] John E. Dennis, Joseph F. Traub, and Robert P. Weber. Algorithms for solvents of matrix polynomials. *SIAM Journal on Numerical Analysis*, 15(3):523–533, 1978.

[33] Richard Dorrance, Fengbo Ren, and Dejan Marković. A scalable sparse matrix-vector multiplication kernel for energy-efficient sparse-blas on FPGAs. *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays - FPGA '14*, pages 161–170, 2014.

[34] Vladimir L. Druskin and Leonid A. Knizhnerman. Two polynomial methods of calculating functions of symmetric matrices. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 29(6):112–121, 1989.

[35] Vladimir L. Druskin and Leonid A. Knizhnerman. Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic. *Numerical Linear Algebra with Applications*, 2(3):205–217, 1995.

[36] Augustin A. Dubrulle. Retooling the method of block conjugate gradients. *Electronic Transactions on Numerical Analysis*, 12:216–233, 2001.

[37] Iain S. Duff, Michael A. Heroux, and Roldan Pozo. An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum. *ACM Transactions on Mathematical Software*, 28(2):239–267, 2002.

[38] Michael Eiermann and Oliver G. Ernst. Geometric aspects of the theory of Krylov subspace methods. *Acta Numerica*, 10:251–312, 2001.

[39] Michael Eiermann and Oliver G. Ernst. A restarted Krylov subspace method for the evaluation of matrix functions. *SIAM Journal on Numerical Analysis*, 44(6):2481–2504, 2006.

[40] Michael Eiermann, Oliver G. Ernst, and Stefan Güttel. Deflated restarting for matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 32(2):621–641, 2011.

[41] Stanley C. Eisenstat. On the rate of convergence of B-CG and BGMRES. Technical report, Unpublished, 2015.

[42] Stanley C. Eisenstat, Howard C. Elman, and Martin H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20(2):345–357, 1983.

[43] A. El Guennouni, Khalide Jbilou, and Hassane Sadok. The block Lanczos method for linear systems with multiple right-hand sides. *Applied Numerical Mathematics*, 51(2-3):243–256, 2004.

[44] Lakhdar Elbouyahyaoui and Mohammed Heyouni. On applying weighted seed techniques to GMRES algorithm for solving multiple linear systems. *Boletim da Sociedade Paranaense de Matemática*, 36(3):155–172, 2018.

[45] Lakhdar Elbouyahyaoui, Mohammed Heyouni, Khalide Jbilou, and Abderrahim Messaoudi. A block Arnoldi based method for the solution of the Sylvester-observer equation. *Electronic Transactions on Numerical Analysis*, 47:18–36, 2017.

[46] Lakhdar Elbouyahyaoui, Abderrahim Messaoudi, and Hassane Sadok. Algebraic properties of the block GMRES and block Arnoldi methods. *Electronic Transactions on Numerical Analysis*, 33:207–220, 2008.

[47] Mark Embree, Ronald B. Morgan, and Huy V. Nguyen. Weighted inner products for GMRES and Arnoldi iterations. Technical Report arXiv:1607.0255v2, 2017.

[48] Ernesto Estrada and Naomichi Hatano. Communicability in complex networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 77(3):1–12, 2008.

[49] Ernesto Estrada and Desmond J. Higham. Network properties revealed through matrix functions. *SIAM Review*, 52(4):696–714, 2010.

[50] Roland W. Freund and Manish Malhotra. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides. *Linear Algebra and its Applications*, 254(1-3):119–157, 1997.

[51] Andreas Frommer and Uwe Glässner. Restarted GMRES for Shifted Linear Systems. *SIAM Journal on Scientific Computing*, 19(1):15–26, 1998.

[52] Andreas Frommer, Stefan Güttel, and Marcel Schweitzer. Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1602–1624, 2014.

[53] Andreas Frommer, Stefan Güttel, and Marcel Schweitzer. Efficient and stable Arnoldi restarts for matrix functions based on quadrature. *SIAM Journal on Matrix Analysis and Applications*, 35(2):661–683, 2014.

[54] Andreas Frommer, Kathryn Lund, Marcel Schweitzer, and Daniel B. Szyld. The Radau-Lanczos method for matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 38(3):710–732, 2017.

[55] Andreas Frommer, Kathryn Lund, and Daniel B. Szyld. Block Krylov subspace methods for functions of matrices. *Electronic Transactions on Numerical Analysis*, 47:100–126, 2017.

[56] Andreas Frommer and Marcel Schweitzer. Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions. *BIT Numerical Mathematics*, 56:865–892, 2016.

[57] Andreas Frommer and Valeria Simoncini. Matrix functions. In Wilhelmus H A Schilders, Henk A van der Vorst, and Joost Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 275–304, Berlin, 2008. Springer.

[58] Felix R. Gantmacher. *The Theory of Matrices*, volume 1. Chelsea, New York, 1959.

[59] Christof Gattringer and Christian B. Lang. *Quantum Chromodynamics on the Lattice*. Springer, Berlin, 2010.

[60] André Gaul. *Recycling Krylov subspace methods for sequences of linear systems - analysis and applications*. PhD thesis, Fakultät für Mathematik und Naturwissenschaften, Technische Universität Berlin, 2014.

[61] Israel Gohberg, editor. *Orthogonal matrix-valued polynomials and applications: seminar on operator theory at the School of Mathematics, Tel Aviv University*. Birkhäuser Verlag, Basel, 1998.

[62] Israel Gohberg, Peter Lancaster, and Leiba Rodman. *Matrix Polynomials*. SIAM, Philadelphia, 2nd edition, 2009.

[63] Gene H. Golub and Gérard Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, Princeton, 2010.

[64] Gene H. Golub and Richard Underwood. The block Lanczos method for computing eigenvalues. In *Mathematical software III: Proceedings of a symposium conducted by the Mathematics Research Center, the University of Wisconsin-Madison*, pages 361–377, New York, 1977. Academic Press.

[65] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4th edition, 2013.

[66] Kazushige Goto and Robert van de Geijn. High-performance implementation of the level-3 BLAS. *ACM Transactions on Mathematical Software*, 35(1):4:1–4:14, 2008.

[67] Anne Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, 1997.

[68] Wolter Groenevelt, Mourad E. H. Ismail, and Erik Koelink. Spectral decomposition and matrix-valued orthogonal polynomials. *Advances in Mathematics*, 244:91–105, 2013.

[69] Martin H. Gutknecht. Block Krylov space methods for linear systems with multiple right-hand sides: An introduction. In A. H. Siddiqi, I. S. Duff, and O. Christensen, editors, *Modern Mathematical Models, Methods and Algorithms for Real World Systems*, pages 420–447, New Delhi, 2007. Anamaya.

[70] Martin H. Gutknecht and Thomas Schmelzer. The block grade of a block Krylov space. *Linear Algebra and its Applications*, 430:174–185, 2009.

[71] Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. Computing A^α, $\log(A)$, and related matrix functions by contour integrals. *SIAM Journal on Numerical Analysis*, 46(5):2505–2523, 2008.

[72] Peter Henrici. *Applied and Computational Complex Analysis, volume 2*. John Wiley & Sons, New York, 1977.

[73] Mohammed Heyouni and Azeddine Essai. Matrix Krylov subspace methods for linear systems with multiple right-hand sides. *Numerical Algorithms*, 40(2):137–156, 2005.

[74] Nicholas J. Higham. *Functions of Matrices*. SIAM, Philadelphia, 2008.

[75] Nicholas J. Higham and Edvin Deadman. A catalogue of software for matrix functions. Version 2.0. Technical Report 2016.3, Manchester Institute for Mathematical Sciences, School of Mathematics, University of Manchester, 2016.

[76] Nicholas J. Higham and Peter Kandolf. Computing the action of trigonometric and hyperbolic matrix functions. *SIAM Journal on Scientific Computing*, 39(2):A613–A627, 2017.

[77] Marlis Hochbruck and Michiel E. Hochstenbach. Subspace extraction for matrix functions. Technical report, 2005.

[78] Marlis Hochbruck and Christian Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 34(5):1911–1925, 1997.

[79] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

[80] Mark Hoemmen. *Communication-avoiding Krylov subspace methods*. PhD thesis, Department of Computer Science, University of California at Berkeley, 2010.

[81] Milos Ilić, Ian W. Turner, and Daniel P. Simpson. A restarted Lanczos approximation to functions of a symmetric matrix. *IMA Journal of Numerical Analysis*, 30:1044–1061, 2010.

[82] Akira Imakura and Tetsuya Sakurai. Block Krylov-type complex moment-based eigensolvers for solving generalized eigenvalue problems. *Numerical Algorithms*, 75(2):413–433, 2017.

[83] Khalide Jbilou, Abderrahim Messaoudi, and Hassane Sadok. Global FOM and GMRES algorithms for matrix equations. *Applied Numerical Mathematics*, 31(1):49–63, 1999.

[84] Hao Ji and Yaohang Li. A breakdown-free block conjugate gradient method. *BIT Numerical Mathematics*, 57(2):379–403, 2017.

[85] Zhongxiao Jia. Generalized block Lanczos methods for large unsymmetric eigenproblems. *Numerische Mathematik*, 80:239–266, 1998.

[86] Mark David Kent. *Chebyshev, Krylov, Lanczos: Matrix relationships and computations*. PhD thesis, Department of Computer Science, Standford University, 1989.

[87] Boris N. Khoromskij. Tensor numerical methods for multidimensional PDES: theoretical analysis and initial applications. *ESAIM: Proceedings and Surveys*, 48(January):1–28, 2015.

[88] Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[89] Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and Its Applications*, 435(3):641–658, 2011.

[90] Leonid A. Knizhnerman. Calculation of functions of unsymmetric matrices using Arnoldi's method. *Computational Mathematics and Mathematical Physics*, 31(1):1–9, 1991.

[91] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2008.

[92] Tamara G. Kolda and Jackson R. Mayo. Shifted power method for computing tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1095–1124, 2011.

[93] Peter Lancaster. *Lambda-matrices and Vibrating Systems*. Pergamon Press, Oxford, 1966.

[94] Peter Lancaster and Miron Tismenetsky. *The Theory of Matrices*. Academic Press, Orlando, 2nd edition, 1985.

[95] Lek-Heng Lim. Singular values and eigenvalues of tensors: a variational approach. In *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP '05)*, volume 3, pages 129–132, 2005.

[96] Luciano Lopez and Valeria Simoncini. Preserving geometric properties of the exponential matrix by block Krylov subspace methods. *BIT Numerical Mathematics*, 46(4):813–830, 2006.

[97] David G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1984.

[98] Cleve Moler and Charles F. van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.

[99] Igor Moret and Paolo Novati. An interpolatory approximation of the matrix exponential based on Faber polynomials. *Journal of Computational and Applied Mathematics*, 131(1-2):361–380, 2001.

[100] Igor Moret and Paolo Novati. The computation of functions of matrices by truncated Faber series. *Numerical Functional Analysis and Optimization*, 22(5-6):697–719, 2001.

[101] Igor Moret and Paolo Novati. Interpolating functions of matrices on zeros of quasi-kernel polynomials. *Numerical Linear Algebra with Applications*, 12(4):337–353, 2005.

[102] Ronald B. Morgan. Restarted block-GMRES with deflation of eigenvalues. *Applied Numerical Mathematics*, 54(2):222–236, 2005.

[103] Herbert Neuberger. A practical implementation of the overlap dirac operator. *Physical Review Letters*, 81(19):4060–4062, 1998.

[104] Michael Ng, Liqun Qi, and Guanglu Zhou. Finding the largest eigenvalue of a nonnegative tensor. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1090–1099, 2009.

[105] Andy A. Nikishin and Alex Yu. Yeremin. Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, I: General iterative scheme. *SIAM Journal on Matrix Analysis and Applications*, 16(4):1135–1153, 1995.

[106] Andy A. Nikishin and Alex Yu. Yeremin. An automatic procedure for updating the block size in the block conjugate gradient method for solving linear systems. *Journal of Mathematics Sciences*, 114(6):1844–1853, 2003.

[107] Paolo Novati. A polynomial method based on Fejér points for the computation of functions of unsymmetric matrices. *Applied Numerical Mathematics*, 44(1-2):201–224, 2003.

[108] Dianne P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications*, 29(1980):293–322, 1980.

[109] Christopher C. Paige, Beresford N. Parlett, and Henk A. van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2(2):115–133, 1995.

[110] Davide Palitta and Valeria Simoncini. Computationally enhanced projection methods for symmetric Sylvester and Lyapunov matrix equations. *Journal of Computational and Applied Mathematics*, 330:648–659, 2018.

[111] Michael L. Parks, Kirk M. Soodhalter, and Daniel B. Szyld. A block recycled GMRES method with investigations into aspects of solver performance. Technical report, Department of Mathematics, Temple University, 2016.

[112] Liqun Qi. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, 40(6):1302–1324, 2005.

[113] Liqun Qi. Eigenvalues and invariants of tensors. *Journal of Mathematical Analysis and Applications*, 325(2):1363–1377, 2007.

[114] Stefan Ragnarsson and Charles F. van Loan. Block tensor unfoldings. *SIAM Journal on Matrix Analysis and Applications*, 33(1):149–169, 2012.

[115] Somaiyeh Rashedi, Ghodrat Ebadi, Sebastian Birk, and Andreas Frommer. On short recurrence Krylov type methods for linear systems with many right-hand sides. *Journal of Computational and Applied Mathematics*, 300:18–29, 2016.

[116] Lothar Reichel, Giuseppe Rodriguez, and Tunan Tang. New block quadrature rules for the approximation of matrix functions. *Linear Algebra and Its Applications*, 502:299–326, 2016.

[117] Mickaël Robbé and Miloud Sadkane. Exact and inexact breakdowns in the block GMRES method. *Linear Algebra and its Applications*, 419(1):265–285, 2006.

[118] Axel Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Mathematics of Computation*, 33(146):680–687, 1979.

[119] Youcef Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of Computation*, 47(178):651–651, 1987.

[120] Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37(155):105–126, 1981.

[121] Yousef Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[122] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, 2nd edition, 2003.

[123] Arvind K. Saibaba, Tania Bakhos, and Peter K. Kitanidis. A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography. *SIAM Journal on Scientific Computing*, 35(6):A3001–A3023, 2013.

[124] Thomas Schmelzer. *Block Krylov methods for Hermitian linear systems*. PhD thesis, Department of Mathematics, University of Kaiserslautern, 2004.

[125] Marcel Schweitzer. *Restarting and error estimation in polynomial and extended Krylov subspace methods for the approximation of matrix functions*. PhD thesis, Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, 2015.

[126] Marcel Schweitzer. Any finite convergence curve is possible in the initial iterations of restarted FOM. *Electronic Transactions on Numerical Analysis*, 45:133–145, 2016.

[127] Valeria Simoncini. Ritz and Pseudo-Ritz values using matrix polynomials. *Linear Algebra and its Applications*, 241-243:787–801, 1996.

[128] Valeria Simoncini. Restarted full orthogonalization method for shifted linear systems. *BIT Numerical Mathematics*, 43:459–466, 2003.

[129] Valeria Simoncini and Efstratios Gallopoulos. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides. *Journal of Computational and Applied Mathematics*, 66:457–469, 1996.

[130] Valeria Simoncini and Efstratios Gallopoulos. Convergence properties of block GMRES and matrix polynomials. *Linear Algebra and its Applications*, 247:97–119, 1996.

[131] Valeria Simoncini and Daniel B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59, 2007.

[132] Kirk M. Soodhalter. A block MINRES algorithm based on the band Lanczos method. *Numerical Algorithms*, 69(3):473–494, 2015.

[133] Kirk M. Soodhalter. Block Krylov subspace recycling for shifted systems with unrelated right-hand sides. *SIAM Journal on Scientific Computing*, 38(1):A302–A324, 2016.

[134] Kirk M. Soodhalter. Stagnation of block GMRES and its relationship to block FOM. *Electronic Transactions on Numerical Analysis*, 46:162–189, 2017.

[135] Dong-lin Sun, Ting-Zhu Huang, Yan-Fei Jing, and Bruno Carpentieri. A block GMRES method with deflated restarting for solving linear systems with multiple shifts and multiple right-hand sides. *Numerical Linear Algebra with Applications*, 2018.

[136] Lloyd N. Trefethen, J. Andre C. Weideman, and Thomas Schmelzer. Talbot quadratures and rational approximations. *BIT Numerical Mathematics*, 46(3):653–670, 2006.

[137] Jasper van den Eshof, Andreas Frommer, Thomas Lippert, Klaus Schilling, and Henk A. van der Vorst. Numerical methods for the QCDd overlap operator: I. Sign-Function and error bounds. *Computer Physics Communications*, 146:203–224, 2002.

[138] Sheng-De Wang, Te-Son Kuo, and Chen-Fa Hsu. Trace bounds on the solution of the algebraic matrix Riccati and Lyapunov equations. *IEEE Transactions on Automatic Control*, AC-31:654–656, 1986.

[139] J. Andre C. Weideman. Optimizing Talbot's contours for the inversion of the Laplace transform. *SIAM Journal on Numerical Analysis*, 44(6):2342–2362, 2006.

[140] J. Andre C. Weideman and Lloyd N. Trefethen. Parabolic and hyperbolic contours for computing the Bromwich integral. *Mathematics of Computation*, 76(259):1341–1356, 2007.

[141] Gang Wu, Hong-kui Pang, and Jiang-li Sun. A shifted block FOM algorithm with deflated restarting for matrix exponential computations. *Applied Numerical Mathematics*, 127:306–323, 2018.

[142] Gang Wu, Yan-chun Wang, and Xiao-Qing Jin. A preconditioned and shifted GMRES algorithm for the PageRank problem with multiple damping factors. *SIAM Journal on Scientific Computing*, 34(5):A2558–A2575, 2012.

[143] Jianhua Zhang, Hua Dai, and Jing Zhao. A new family of global methods for linear systems with multiple right-hand sides. *Journal of Computational and Applied Mathematics*, 236(6):1562–1575, 2011.